Operation Breadcrumbs: Making the Demo for *Bed and BEAKfast*

A Major Qualifying Project Report
submitted to the faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the

Degree of Bachelor of Science
in Computer Science

Degree of Bachelor of Science
in Interactive Media and Game Development

Degree of Bachelor of Arts
in Interactive Media and Game Development

Authors:
Tate Donnelly
Nicholas Frangie
Jade McEvoy
Schuyler Rae Pritchard
Abigail Rauch
Kerri Thornton

Date: April 25, 2024

Dr. Gillian Smith (IMGD/CS)
Adryen Jean Gonzalez (IMGD/HUA)
Ben Schneider (IMGD)

Abstract

Operation Breadcrumbs is the codename for a management and narrative video game about running a bed and breakfast in a community of anthropomorphic birds. Through the character Robin, the narrative explores themes of community and aid, as gameplay focuses on satisfying customers with an intricate cooking system. The goal of this Major Qualifying Project (MQP) was to develop a fleshed-out game concept and release a polished demo to the online distributor, Steam; as of this time, this demo is slated to be released under the name *Bed and BEAKfast* following the completion of this project. This report documents the complete development process, concentrating on the unique challenges of interdisciplinary collaboration faced when developing a game within a year.

Acknowledgments

There are a number of people the team would like to thank for their role and contributions to the Operations Breadcrumbs project; all of our playtesters, social media followers, and the wider IMGD community played an important role in the creation and evolution of our game. Among these groups, we would like to pay special acknowledgement and name several of these supporters.

Firstly, we would like to give thanks to our project advisors, Dr. Gillian Smith, Ben Schneider, and Adryen Jean Gonzalez: over the course of this year, these advisors provided us with their bountiful feedback and input that was invaluable to the team. The care, attention, and consideration they paid to our team ensured that we never strayed too far off course, while still providing us the ability and freedom to find our own way and learn from this experience.

Secondly, the team expresses our heartfelt appreciation to the non-core members who aided in the game's development. Although this MQP is made up of 6 students, several more provided their support throughout the project's development through ISPs; we can safely say the game wouldn't be the same without their special contributions. As such, the MQP would like to thank Ethan Chau for aiding in writing cutscenes and characters; Camille Prats for her work animating characters; Teddy Ker for their additions of flavor text and additional dialogue; and Alistair Gilmour and Jessica Liano for their audio composition and orchestration.

Lastly, we would like to draw special attention to the 2024 PAX IQP team, and share our sincere gratitude for the hard work and effort they expended to plan and coordinate the several events they host for IMGD and its MQPs. Throughout the 2023-2024 academic year, this team was responsible for hosting the internal showcase events Alphafest, JoyArt, and Showfest, along with creating and staffing the IMGD Booth at PAX East 2024. Through their efforts, our team was fortunate enough to be showcased at all four of these events, giving us the opportunity to demo the game to wider audiences; for this, we'd like to acknowledge and thank them for making these events possible.

Table of Contents

1 - Vision

This chapter describes the goal, the initial concept, our design pillars, the experience goals, our target audience, and the values we wanted to express for our project.

1.1 - Goal

Operation Breadcrumbs is the development title for this Major Qualifying Project (MQP), which spans the development of a video game demo originally conceived by the team's six members pursuant to the Interactive Media and Game Development (IMGD) program at Worcester Polytechnic Institute (WPI). The goal of this project was to mimic the development process of a small indie game; however, the team chose early on to not create a full game. Instead, the *Operation Breadcrumbs* would focus on creating a substantial vertical slice featuring the game's introduction arc, intended to be a shorter section of the game featuring mechanics, characters, and gameplay representative of a full title. This gameplay segment would then be released as a polished demo of the idea that could continue into a full game. By creating a short demo of the original game concept, the team could confront many of the core challenges of game development while otherwise utilizing the limited timeline of MQP to deliver high-quality content.

Our game would be a management and narrative game featuring a bed and breakfast run by and for anthropomorphic birds, eventually released under the name *Bed and BEAKfast*. Unlike most games of the management genre — which glorify the high-earning and economic prosperity of the player — we wanted to explore community values in opposition to this profit-centric mindset, often embedded in the capitalist nature of management games. *Bed and BEAKfast* would have a focus on the characters and the community in which the player exists, and not simply the act of achieving greater gains via the player's growth: a game about not optimizing away the very human interactions of running a business.

This report will cover the development process of *Bed and BEAKfast*, from the early concept stages to pre-production to production and, finally, our game's release. Our report will also cover important events at which we showcased our game, such as Protofest, Alphafest, JoyArt, PAX East, and Showfest.

1.2 - Concept

The concept of a game serves not as a blueprint, but as a designed space in which the game is intended to inhabit. The team created a plan for our ideal version of the game, and set it as a faraway goal to hit as we iterated on our ideas, understanding that in the end, though the game might stray from the explicit plans and diagrams laid out in advance, it would remain within the bounds of the vision that we agreed upon at the start of development.

1.2.1 - Game

Our initial concept for the game was a cozy management game set in a world populated by birds. Within this setting, the player was to be incentivized to interact with each customer as a unique character, catering to their individual preferences and needs rather than simply chasing higher numbers and metrics. The envisioned game sought to fulfill those design goals through several interconnected systems.

Every morning, birds request meals from the player character and bed and breakfast (B&B) owner, Robin. Over the course of 15 in-game days, the player learns to satisfy every bird's preferences by experimenting with cooking and interacting with them. In order to gather materials for cooking and furthering your relationships with other characters, the player must forage for them or sometimes purchase them. This sequence of events prompts the player to explore the whole world and its systems (i.e. cooking, reputation, and dialogue systems) in order to reach their goal: giving the residents of the B&B a comfortable and enjoyable experience. The money obtained at the end of a bird's stay is largely a formality. While it can be used to purchase some materials, it is mostly used to pay the B&B's rent and is otherwise supplanted by performing acts of mutual aid to assist other birds and receive help in return.

1.2.2 - World

The world of the game exists to contextualize the message and intended experience in a familiar way: animal allegory. Animal allegories are generally used in stories to help talk about serious topics and teach important lessons.[1] In particular, the game exists in a world where birds and other woodland creatures are sentient, intelligent creatures with their own social environment separate from the world of humans. This particular view into that world focuses exclusively on birds and a singular squirrel in a bird town set approximately in New England. While the game's world references the existence of humans with human produced materials and human-populated areas occasionally being shown, humans would not appear directly in the game or be referenced in dialogue.

In this world, birds lead complex lives in treetop towns where they grow up, work, and interact with their community. The game's world mirrors the real avian world in the behavior of some species. For example, the hometown of Arboro is largely populated by birds that make their homes in the northeast of America. Travelers are made up of birds that often migrate through this region. The squirrel character, a landlord, is designed after the tendency for squirrels to stash away large numbers of nuts only for themselves.

By using animals to portray aspects of our own society, we could convey less familiar, more palatable lessons to the audience through the lens of compelling, non-controversial creatures. In our experience, games seeking to provide players with a lesson or moral will end up

[1] Carolyn L. Burke and Joby G. Copenhaver, "Animals as People in Children's Literature," *Language Arts* 81, no. 3 (January 2004): 205–13, https://doi.org/10.58680/la20042896.

coming across as preaching, pushing away players who don't want a game they have to think too hard about or one they assume will be boring and dry. Animal allegory has historically been used in stories such as Animal Farm[2] and Aesop's Fables[3] which comment on societal issues and talk about more controversial topics. Taking inspiration from these respected stories, rather than dramatically diluting the social commentary aspect, we decided to use animal allegories to make it more digestible to a wider audience.

1.2.3 - Values

One of the main ideas we wished to convey to players was the benefit of mutual aid, an organization theory where individuals in a community support each other without the aid of institutions like states and Non-Governmental Organizations (NGOs).[4] Simply put, it is described as "solidarity, not charity" — a message the game aimed to teach through its story and mechanics.

This game's central story is unabashedly in support of mutual aid, promoting community efforts throughout the story. Additionally, our game criticizes the legitimacy of land ownership as a means to generate wealth. To indicate this, the principal antagonist, Sterling, was designed as a landlord who fervently believes in the tenets and strengths of capitalism, while his influence has almost entirely eroded the social structure of the town. Sterling's short-sighted pursuit of profits leads him to raise the rent on the player's B&B, and contributes to the deterioration of the town's third place[5] prior to the game's opening.

This idea of a third place, another location outside of work or home, is an essential space for people in a community to come together and for outsiders to experience and contribute to the town's social structure. The player's role in the story centers around reestablishing this third place as they grow their B&B from a small tree stump into a blossoming fixture in the community; doing so requires them to disavow Sterling's ideas regarding the natural, money-pursuing tendencies that most management games demand in order to succeed.

Our game also takes inspiration from American political theorist Jodi Dean's "Capitalism is the End of the World,"[6] in which Dean investigates the pervading sense of hopelessness present when facing the problems and fears of capitalism. Dean pinpoints the loss of community and camaraderie as its root cause: even those aware and fearful of the devastation caused by capitalism are paralyzed from attempting any changes, failing to perform collective politics and

[2] George Orwell, *Animal Farm*, (London, England: Secker and Warburg, 1945)
[3] Aesop and Milo Winter, *The Aesop for Children* (Chicago: Rand McNally & Co., 1919).
[4] "What Is Mutual Aid?," Mutual Aid, accessed March 10, 2024, https://www.mutualaid.coop/what-is-mutual-aid/.
[5] Leo W. Jeffres et al., "The Impact of Third Places on Community Quality of Life," *Applied Research in Quality of Life 4*, no. 4 (October 13, 2009): 333–45.
[6] Jodi Dean, "Capitalism Is the End of the World," *Mediations* 33, no. 1–2 (2020), https://mediationsjournal.org/articles/end-of-world.

instead accepting their suffering as individuals. Our game seeks to emphasize the need for communities to come together to find potential solutions.

Several gameplay mechanics were ideated to reinforce this idea such as downplaying the role of money, a staple in traditional management games of this sort. As part of managing the B&B, a rent mechanic was planned to ensure the player was consistently strapped for cash, regardless of their performance; in this vein, money was more symbolic than a gateway to progress. The game would instead provide alternate avenues for running a successful B&B without amassing wealth: getting to know other birds, helping them, and gaining their friendship might lead to their aiding the B&B's growth. In this ideal end-game state for the player, this third place would grow from the contributions and support of the wider community to truly make it into a shared space for all, rather than one controlled by the rent imposed by the game's antagonist.

To accompany this narrative and the political ideologies it expounds, several side stories for birds were planned to be woven into the game to focus on problems tangential to this system. Criticism of unhealthy mindsets and situations spawning from unchecked capitalism — such as work addiction, materialism, cost-cutting, and worker safety, as well as how those problems might be overcome — were planned to receive focus and attention from the later story, further promoting the messages of mutual aid embedded at the heart of this story.

1.3 - Pillars

Pillars are core ideas or statements which are intended to guide a game's development.[7] They must be created early on in the game development process to guide decisions made throughout production. Pillars can be used as metrics to determine whether specific decisions are a right fit for the game, and revising them regularly as a team is important to ensure the game is on the right track.

In our case, we chose our pillars for the overall game and its larger vision. Being that the demo the MQP worked on is meant to set up for the full game — seeding ideas and mechanics that would be expanded in a fuller release — we elected to use the same pillars in this narrow vertical slice.

The pillars for *Operation Breadcrumbs'* game were established in pre-production and then revisited and updated throughout development as the shared vision for the project continued to evolve and change:

1. Management simulator that **prioritizes community** over business.

[7] Max Pears, "Design Pillars – the Core of Your Game," web log, *Game Developer* (blog) (Informa PLC, October 12, 2017), https://www.gamedeveloper.com/design/design-pillars-the-core-of-your-game.

Our highest priority was making sure that the game focused on the birds the player interacted with and the community they were a part of, rather than the business and management mechanics. Since management simulators typically focus on monetary and profit growth, we wanted to ensure the original messaging of the game was upheld, prioritizing systems that propped up and displayed the community and its members at the forefront.

2. Gameplay systems encourage **deliberate experimentation**.

The next pillar sought to codify the types of interaction we wanted players to have within the game. Deliberate experimentation is meant to characterize the way players approach the game systems, with the hope that gameplay would demand a combination of thought and exploration to solve the puzzles it posed the player as part of managing the B&B.

3. Use bird behavior and community as an **allegory for human situations** in late-stage capitalism.

The original vision for the world and its messaging were selected together to allow people to sympathize and agree with the values expressed in these scenarios. As such, acting on this allegory meant ensuring the intended morals and lessons could be interpreted from the story without feeling heavy-handed or obvious.

4. Showcase **visual growth** that reflects thematic and gameplay progression.

Our final pillar was conceived from a desire to engage the player with the game through the expansion of visual elements, such as the physical growth of the player's B&B tree. Even though we were intentionally subverting the role that money serves in most management games, a large part of the genre convention we wanted to pull from was the visual sense of progression and growth the player achieves with success.

1.4 - Experience Goals

A natural counterpart to a game's pillars are its experience goals:, which are meant to guide how the player should feel while playing the game. The team created three experience goals for the player; we want our game to be hopeful, be impactful, and encourage experimentation.

Establishing a sense of hope is critical. Both the demo and overarching story cover heavy topics such as being priced out of one's home, gentrification, economic inequality, and other effects of late-stage capitalism. Despite these serious themes, we didn't want the game to leave players feeling dismayed or sorrowful. Oftentimes when faced with the harsh realities of the world, people already feel overwhelmed and unable to make real change. We intended for the player to feel uplifted by the strength of their in-game community and in turn be inspired to strengthen their own community and combat these real-world problems, rather than feeling dejected and powerless in the face of an abstract evil.

The game should still be impactful for the player however, so that they may recognize the depth and complexity of the socioeconomic problems explored by its story. The team took care to avoid minimizing these very real problems by the hopefulness of the previous experience goal. The tangible issues investigated in the game warrant further consideration and discussion; we wanted the game to compel them to dig deeper than a surface-level comprehension of these problems, to avoid over-simplifying the issues or hand-waving away a solution.

Lastly, players should have the freedom to experiment with gameplay systems to fulfill customer and community needs. *Operation Breadcrumbs* should not consist of formulaic number-crunching; users should not simply be completing perfunctory tasks in order to increase their profits. Rather, they should be actively learning about the characters around them, experimenting with new recipes, and exploring the storylines available to them; this will allow them to best help and serve their community while encouraging growth for all.

1.5 - Target Audience

As part of establishing a vision for the game, the team needed to determine its target audience. The importance of establishing a target audience cannot be understated; knowing who the game is made for is imperative to ensuring that it is tailored to their player experience within an existing demographic that will enjoy and engage with the game[8]. Without this audience in mind, much of the decision making made throughout development can be inconsistent and confusing, resulting in it failing in finding an audience at all. All of the time and effort spent on perfecting a project can be rendered useless if it appeals to no one.

To correctly identify our target audience, we first analyzed the target audiences of our game comparables; these are existing games that share similar gameplay, themes, or experiences that we planned to model in our finished product. In order to align our individual visions, one of the first steps of our development process in December 2022 was to gather these comparables and play them; these games included *Spiritfarer,*[9] *Animal Crossing,*[10] and *Stardew Valley.*[11] While analyzing these games, we also took a look inwards at our own experiences and how they would influence the game's final demographic, comparing elements that we enjoyed about our game comparables to elements that we did not. It was important to focus not just on the mechanics of these games, but also the *experience* of playing them. At the same time, we talked about the story we were trying to tell and to whom we were trying to tell it. We narrowed down our goal: a call to action for people similar to ourselves in our current walks of life, who feel a sense of hopelessness and disjointedness as a result of our current capitalist situation.

[8] Nick Lim, "The Importance of Knowing Your Game's Target Audience," Sonamine, April 4, 2023, https://www.sonamine.com/blog/the-importance-of-knowing-your-games-target-audience.

[9] *Spiritfarer,* (Thunder Lotus Games, 2020), PC, Mac, Switch, PlayStation 4 or later, and Xbox One or later.

[10] *Animal Crossing: New Horizons,* (Nintendo, 2020), Switch.

[11] *Stardew Valley,* v. 1.5 (ConcernedApe, 2016), PC, Mac, Switch, PlayStation 4 or later, Xbox One or later, iOS, and Android.

Our first target audience is young adults. Specifically, we are trying to target those who are in the 18-25 age range and are experiencing the awkward transition from childhood to adulthood through both nostalgia and adult hardships. There are many elements to the game that were created with the intention of attracting this audience. For example, these players will be able to see themselves in the player character of Robin through their immaturity and desire to hold onto childhood and impulsivity. These players may also relate to Robin through the passing of Pop; for many young adults, the death of a grandparent is their first major experience of loss, and can even signify the ending of childhood. Even the act of Robin visiting their old hometown to see how it and its inhabitants have changed since they were gone is something that this demographic has likely experienced in their early adult life.

Our second target audience are those who are sympathetic to or have experienced problems caused by capitalism but don't know how to address them, trapped in a spiral of helplessness against a much larger system. We wanted the game to appeal to these players as our game shows how ideologies such as mutual aid can serve as a potential solution. As explored in Section 1.2.3, the feeling of hopelessness in the face of a highly individualist, capitalist society is pervasive, and we hope to do more for this group by directly imparting lessons about community as a means to offset these issues. This reasoning sums up the target of our game's purpose; by experiencing the story of Robin and their community coming together, establishing their third place, and bettering the lives of birds in and out of their town, we hope that players feel inspired to strengthen ties with those around them and acquire a new sense of hope to rise up against the real and tangible problems created by capitalism.

By specifically targeting both of these audiences, we want to show these players that they are not alone in their experiences, encouraging them to confront problems in their own lives. In determining solutions and considering new perspectives on these issues through gameplay — as well as by encouraging the fostering of relationships and community between characters — these players will be given a new sense of hope for their future.

2 - Background Research

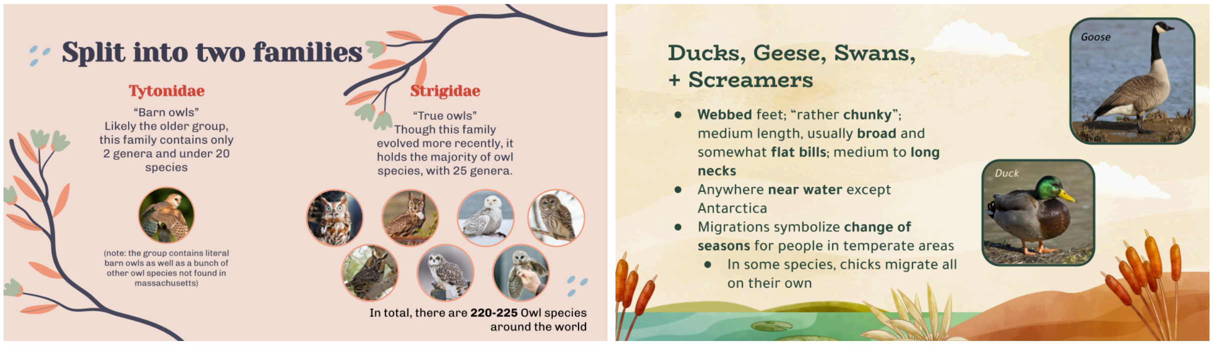
In this chapter, we discuss the research we did prior to and throughout the early stages of the project's development. The topics of this research span birds behavior and ornithology, games we took inspiration from, and socio-economic ideas that we wanted to touch upon as part of the game's story.

2.1 - Birds

Given that we were creating a fairly bird-centric game, we thought it was important early on for our game and its world that we learn more about birds. This research covered a vast array of topics and served as an important base for fleshing out the allegory and political messaging involved in the game. Additionally, we thought it was important that all members of the team participate in the research so that we could collectively use it to make many game decisions such as what birds would appear in the story, what foods could the player make, and what would the characters look like.

We chose to research bird ornithology as it would help the team to create convincing bird inhabitants for the game and gather inspiration for stories based on real-world bird behaviors. The research that was done had three areas of focus: exploring the many possible groups of birds existing throughout New England, acquiring a basic understanding of bird migration, and learning about everyday bird behaviors such as the mating season, diets, and interspecies relationships.

2.1.1 - Species



Split into two families

Tytonidae

"Barn owls"
Likely the older group, this family contains only 2 genera and under 20 species

(note: the group contains literal barn owls as well as a bunch of other owl species not found in massachusetts)

Strigidae

"True owls"
Though this family evolved more recently, it holds the majority of owl species, with 25 genera.

In total, there are **220-225** Owl species around the world

Ducks, Geese, Swans, + Screamers

- **Webbed** feet; "rather **chunky**"; medium length, usually **broad** and somewhat **flat bills**; medium to **long necks**
- Anywhere **near water** except Antarctica
- Migrations symbolize **change of seasons** for people in temperate areas
  - In some species, chicks migrate all on their own

Figure 1: Slides from two presentations on Strigiformes (left) and Anseriformes (right).

The broad nature of this first objective, collecting information on birds in the New England region, necessitated that the team categorize birds into groups that were feasible to research and present amongst the team. To do this, the team borrowed from recognized bird taxonomic classification, and settled on orders as the level at which to investigate the many species. Examples of groups in this category include owls (Strigiformes), hummingbirds

(Apodiformes), and tubenoses (Procellariiformes). The largest group, into which most of the characters in the game would fall, was songbirds; this group is contained in the order Passeriformes, and the order was broken down further into families (such as nuthatches, finches, and tits).

Presentations on each group were made and presented to the full team and kept in the Google Drive to be later referenced by the writers and artists. This was to aid in the process of creating characters, from their personalities to their designs.

A multitude of sources were referenced for this research, including Cornell Lab's "All About Birds" reference website,[12] the National Audubon Society,[13] the Smithsonian's *Birds of North America,*[14] and Fowler's *Zoo and Wild Animal Medicine, Volume 8.*[15] Knowing which birds were present in the area, under what circumstances they inhabited it, and their overarching traits were important first steps in brainstorming characters based in the real world. Information like this guided the character's personality, appearance, and even potential role in the story.

2.1.2 - Migration

We were able to use many of the same resources when researching birds' fall and spring migration habits. Through our research, we found that during the summer season, birds in the northern hemisphere breed and raise their young across many terrains and climates. When fall comes, many of them migrate south to the warm equator – in winter, to an entirely different territory. The arrival of spring brings a new breeding season, and they migrate back to their summer region to claim a territory.

In the past, researchers have assumed that the bird species which summer in boreal (also known as northern) climates get pushed south by increasingly cold winters, where they must wander and scavenge for survival among the residents. However, modern research and bird observation notes that many birds which winter in the tropics have specific niches in which they fit, both in their summer and winter locations.[16] The team paid particular attention to this understanding of migration, in which birds have definite places in their communities in both summering and wintering locations, as well as during migration itself, as it pertained to the story being created. In-game traveling birds are likely birds who migrate through the area, while town birds are often permanent or long-term residents of the area.

[12] All About Birds, accessed April 2, 2024. https://www.allaboutbirds.org/guide/.
[13] Audubon, accessed April 2, 2024. https://www.audubon.org/.
[14] Fred Alsop, *Birds of North America* (New York, NY: Covent Garden Books, 2006).
[15] R. Eric Miller and Murray E. Fowler, *Fowler's Zoo and Wild Animal Medicine: Volume 8* (St. Louis, MO: Elsevier, 2015).
[16] John Hilton Rappole, *Bird Migration: A New Understanding* (Baltimore: Johns Hopkins University Press, 2022).

2.1.3 - Behavior

The team also made it a point to look into bird behavior, both in general and specific to the birds included in the story. The fact that birds form interspecies flocks and coexist across species was of particular interest when forming the town made up of many different species. Additional behaviors that became specifically relevant to the team involved nesting, bird family size, and overall life patterns; these all influenced character decisions. For example, warblers migrate long distances and as a result migrate very early in the season;[17] to depict this behavior in our game, warblers were characterized by being rushed, fast-moving, and energetic.

The research into each of these topics continued until the team felt confident they could create a compelling story with characters whose behaviors pulled details from their nature. Even so, the team felt the extra time invested in studying the overall behaviors and habits of birds was a necessary expense, even if not all the research made its way directly into the game. At the onset of the project, there was little way of knowing what we needed to know that might be relevant to our game and story; despite this, the broad research period carried out as part of pre-production helped establish a shared base of bird knowledge that the team could pull from in the months to come.

2.2 - Game Comparables

Similarly to how the research of birds and ornithology was essential to developing the world and allegory of our game's narrative, a thorough analysis and investigation of similar games was deemed necessary by the team as a matter of providing us with a shared understanding of what game we were making.

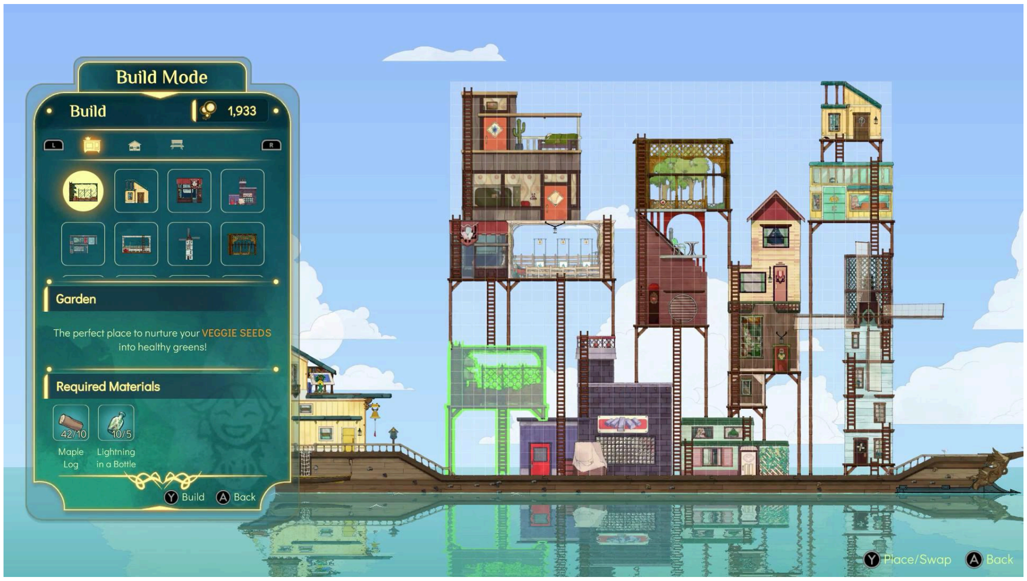
In the industry, this practice of designating and studying games with a similar player experience or target audience one's game concept is an important step to identifying what your game is. Referred to as "game comparables" in this paper, they are those games that are mechanically, artistically, or conceptually similar to the one being developed or discussed. They are often decided early in the brainstorming process and used to demonstrate ideas between team members, influence the art style, and direct the design decisions of the game. They can then be continually referred back to as goals or ideas to adhere to throughout development. Game comparables are also important when introducing and pitching the game to a wider audience, especially when its comparables are more widely known.

Our team made sure to pick comparables that would well illustrate our vision for the game, both internally and externally. As such, our game is unavoidably and intentionally inspired by those other games that the team have played and researched, primarily those with simulation or management gameplay and anti-capitalist messages.

[17] "Yellow Warbler," All About Birds, accessed March 24, 2024, https://www.allaboutbirds.org/guide/Yellow_Warbler/maps-range.

After pre-production, the final comparables selected for our game included *Spiritfarer,*[18] *Animal Crossing,*[19] and *Stardew Valley;*[20] each of these demonstrated a different aspect we aspired to emulate with our game.

2.2.1 - Spiritfarer



The perfect place to nurture your VEGGIE SEEDS into healthy greens!

Figure 2: An image from *Spiritfarer*, developed by Thunder Lotus Games, PC/Mac, 2020.

The first of our comparables, *Spiritfarer,* was chosen for its platforming style, system of vertical and customizable rooms for characters, and dynamic cooking system. In this game, the player assumes the role of the titular “Spiritfarer,” who uses a ship to ferry people who have died to the afterlife, tending to them as their spirits make peace before crossing over in their final days.

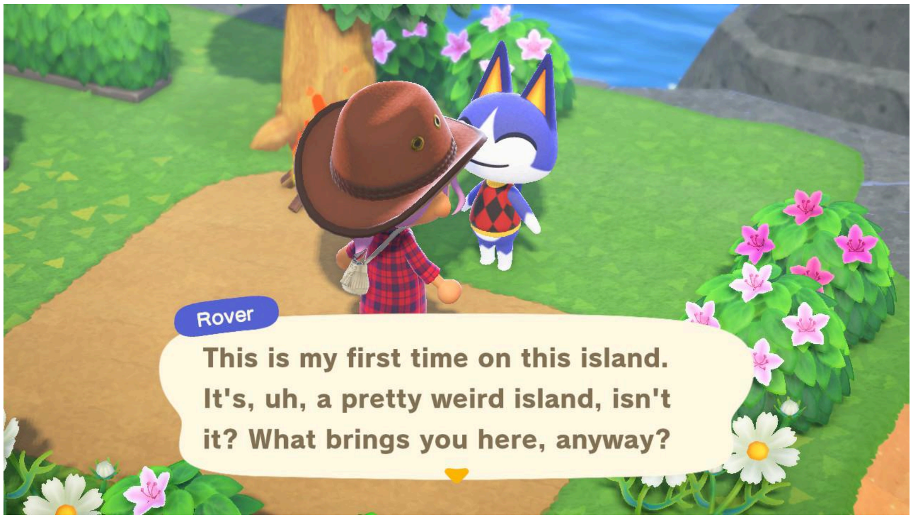
As the player progresses through *Spiritfarer*, they are able to see their progress visually through the additions to their ship; these additions also provide a platforming challenge for the player to master in order to swiftly travel from one location to another. Given that our characters are birds, we took inspiration from this game’s vertical level design for the bed and breakfast that Robin traverses daily. It too grows visually with the player’s progression, and the player must use flight to traverse it easily.

Another piece of inspiration from *Spiritfarer* was the cooking system; in this game, up to two ingredients can be combined to create a dish, with certain dishes being the favorite of particular spirits. If the one or two ingredients do not create a meal, they create a “Questionable

[18] *Spiritfarer,* (Thunder Lotus Games, 2020), PC, Mac, Switch, PlayStation 4 or later, and Xbox One or later.
[19] *Animal Crossing: New Horizons,* (Nintendo, 2020), Switch.
[20] *Stardew Valley,* v. 1.5 (ConcernedApe, 2016), PC, Mac, Switch, PlayStation 4 or later, Xbox One or later, iOS, and Android.

Meal" which is disliked by almost all characters. In addition, those meals are sorted into various types, such as "Breakfast," "Fine Dining," and "Salad." We took a similar approach, where the player inputs up to two ingredients into appliances in the kitchen and gets a meal in return. If that meal does not match a set recipe, it becomes Slop. Those meals are sorted into various types such as "Drink," "Bread," and "Casserole."

2.2.2 - Animal Crossing



This is my first time on this island. It's, uh, a pretty weird island, isn't it? What brings you here, anyway?

Figure 3: An image of *Animal Crossing: New Horizons*, developed by Nintendo, Switch, 2020, screenshot.[21]

Animal Crossing is a well-renowned life sim, where players live and interact in a town populated by anthropomorphic animals. In this series, gameplay revolves around tasks such as fishing, bug catching, gardening, and socializing with the villagers, as the world and events play out in real-time as part of the game's slow-paced and charming appeal.

Animal Crossing was chosen as a representative of how we would create the bird inhabitants of the world, as well as for ideas about physically constructing and supporting a community. It features a wide cast of characters that are all designed after animals, and ones that share a personality type will share potential lines of dialogue. We also designed our characters after a variety of animals, in this case species of birds. Similarly, we ended up creating a comparable dialogue solution for several types of birds, assigning filler characters a personality type and a set of various speech options.

[21] "Dialogue," *Interface In Games*, accessed April 2, 2024, https://interfaceingame.com/screenshots/animal-crossing-new-horizons-dialogue/.

2.2.3 - Stardew Valley

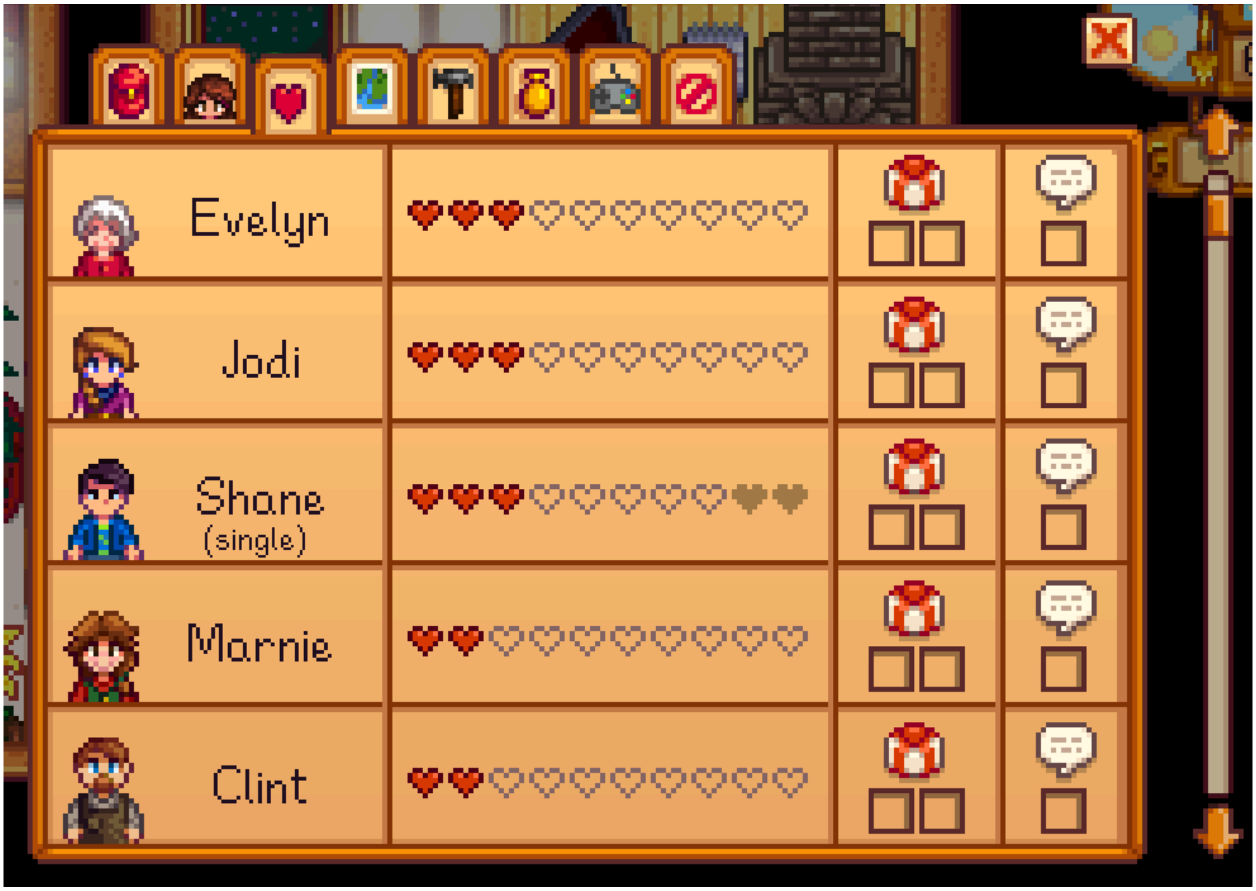


Figure 4: An image of *Stardew Valley*, developed by Eric Barone, PC/Mac, 2016, screenshot.[22]

Among our comparables was the famously successful indie game *Stardew Valley,* a management game about moving to a remote town to run a farm left to the player by their late grandfather. This similarity to the inciting incident of our own story is in no way an accident, with the premise having become a boilerplate introduction for indie games in this genre, thanks in no small part to this game.[23] Beyond this story point, *Stardew Valley* informed the business simulator aspect of our proposal and its game loop, as well as a similar system of befriending characters and viewing cutscenes about their personal story.

The daily gameplay loop gives the player limited time each day to grow crops, explore, interact with others, and manage their farm. There are a specific number of days in a week,

[22] Billie Jo Parker, "The Player Menu: Everything You Need in One Spot," web log, Stardew Valley Blog (blog), March 4, 2021, https://blogs.uww.edu/stardewvalleyblog/2021/03/04/the-player-menu-everything-you-need-in-one-spot/.

[23] Josh Cotts, "The Wave of Stardew-Likes Is Finally Here (& They'll Probably Keep Coming)," CBR, October 20, 2023.

month, and year, with the story progressing as the player does regardless of this time. We took inspiration from this loop, and the original vision for the game closely reflected this. Our initial concept sought to emulate much of the gameplay loop, including the player balancing the management of their B&B with foraging for ingredients in the wild and interacting with townsfolk, all within a strict daily time limit. As our game evolved over the course of the year, its gameplay loop became more distinctive, and as a result it merely borrowed elements from *Stardew Valley* without pulling exactly from its gameplay loop.

The final version of our game's demo encompasses a half-month of playable days. While there is no limit on how much time the player can spend in a day, the player must still complete daily tasks to manage their bed and breakfast. The player also receives a breakdown of their performance at the end of the work period; where *Stardew Valley* rewards the player with money, our game grants points to Robin's relationship with each character.

These relationships are also modeled after *Stardew Valley*'s comparable system, where each character has a total number of relationship points with the player character, represented by the number of hearts in the relationships menu. At certain heart thresholds, characters will mail the player gifts. Each character also effectively has a storyline made of multiple cutscenes that are unlocked at certain heart thresholds and contain dialogue and limited movement. In our game, these cutscenes are unlocked on certain days, but retain a similar format.

Finally, we aimed to tackle some narrative themes that this comparable also explored, but in a different way. *Stardew Valley* is decidedly anti-corporation; its main antagonist is a local supermarket manager, and the encouraged path through the game has the player revitalize a local community center and kick the corporation out of the town with the power of friendship (and optionally a very powerful punch). It is not, however, anti-capitalist. The primary measure of success is money, and the player supports a local business that utilizes unethical sales tactics. Our game focuses instead on making connections with other characters and visual growth of the bed and breakfast as measures of success. It also takes a stance on ethical growth and business: Robin taking shortcuts to potentially increase profit directly harms birds within the community.

2.2.4 - Anti-Comparables

We also referenced games in order to keep in mind aspects we wanted to avoid. Other games featured design decisions that wouldn't fit with our core design decisions, or the overall concept of the game was antithetical to ours despite having some similarities. These anti-comparables are helpful to look at as examples of what not to do, and help to reduce the time spent exploring the potential design space.

One game that we decidedly did not want to use as a comparable was *Bear and Breakfast.*[24] We were immediately wary of the similarities in the project proposal's title and the

[24] *Bear and Breakfast*, (Armor Games Studios, 2022), PC, Switch, PlayStation 4 or later.

concept of animals being served at a bed and breakfast. Our game features exclusively birds as characters existing at a bird-sized scale, and the story focuses on the impact that businesses, public spaces, and capitalism have on communities. *Bear and Breakfast* focuses more on expansion and accelerating growth and features many different species of woodland creatures in a human-scale world. Additionally, it has much more focus on decorating the B&B appropriately for the clients, while our game focuses more on cultivating relationships directly and cooking. Though the overall idea of running a B&B for animals is similar, our gameplay systems and themes are significantly different.

There are also ways that comparable games can be used as anti-comparables when there are unfavorable details of a system that would otherwise fit with our design. For example, while we considered referencing the cooking system of *Stardew Valley* as many other facets of the game were an inspiration, we ultimately did not take many cues from its cooking system. *Stardew Valley* lets the player directly use recipes from the recipe book, preventing the player from inputting whichever ingredients they want. This lack of experimentation is the opposite of our intent with the cooking system; as such, we did not replicate that in our design. There is also the troublingly capitalistic plot of the *Animal Crossing* series that we did not wish to draw from; the premise of each game revolves around Tom Nook selling the player lodging and instructing them to work in order to pay off their newfound debt. In *New Horizons*, it also includes the resource depletion and alleged advancement of uninhabited islands. Needless to say, these were not messages we wished to carry on through our game.

Besides these three, there were considerable other games counted as game comparables before ultimately being dismissed throughout the pre-production and development phases for various reasons. Sometimes the team decided games that we thought would make good comparables were actually in contradiction to our intended player experience upon closer examination. As such, the team used these games to explore additional design spaces, without spending crucial development time to design and test them for ourselves. One such example included plans for a style of cooking similar to *PlateUp!,*[25] a frenzied cooking game that requires players to actively coordinate the preparation and combination of ingredients to make foods. After some prototyping and testing, *PlateUp!* was eventually dismissed from being a game comparable once the team realized that this would not be in line with our plans for a cozy, singleplayer, sidescroller game.

In this manner, many other games would also be examined and compared against our pillars to gain an understanding of what would and wouldn't work in the game. This strategy empowered the team to borrow and reject ideas from the shared base of comparables and anti-comparables we established, and with this important tool, we were able to improve our gameplay systems on top of what we had already envisioned.

[25] *PlateUp!*, (Yogscast Games, 2022), PC, Switch, PlayStation 4 or later, Xbox One or later.

2.3 - Socio-Economics

With the ambitious goal of tackling important and sensitive social issues, the team took the time first to research the topics we felt were most relevant to our game. The two most prominent topics were mutual aid and gentrification.

2.3.1 - Mutual Aid

As one of our core pillars is the prioritization of community and our game touches on themes of community building, the political messaging that arose from early on in development was vocal support for mutual aid. In order to do this topic justice, research was done into understanding the philosophy behind mutual aid. We felt that this was important to establish a baseline for the topic amongst the team, so that we were equipped to approach one of our goals: showing players examples of mutual aid to inspire them to apply the philosophy in their own life.

In his article, *Solidarity Not Charity: Mutual Aid for Mobilization and Survival,* Dean Spade explains that mutual aid is when "people take responsibility for caring for one another and changing political conditions … by actually building new social relations that are more survivable."[26] Importantly, mutual aid is not synonymous with charity; rather, it focuses on "collective care" and building solidarity with others in the mutual aid organization. This distinction is important as charity is support, often conditional, given to marginalized people from the government and wealthy individuals. The patriarchal nature of charity also serves to portray the giver as generous saviors, further legitimizing exploitative economic systems that benefit only a few individuals, while simultaneously stigmatizing the need for such support and dissuading people looking for relief. In contrast, mutual aid focuses on support among peers and is based on empowerment, solidarity, and reciprocity. This is a message that can empower a player and inspire hope, making it an important distinction when approaching the story of our game.

The distinction between charity and mutual aid was important for us to understand as our story follows a protagonist who, through mutual aid efforts, is able to help build their community. Due to us only having one character's point of view – the player character, Robin – it was important that they not be viewed as a "savior," a common trope for video game protagonists to fall into due in part to the cultural monomyth that is *The Hero's Journey.* While the trope can help the player see themselves as the protagonists, it would actively be harmful to the message of our story. In their analysis, Bond and Christensen criticize The Hero's Journey's protagonists for being the prototypical white savior and for its emphasis on individualism, stating it "uses weaker people as instruments" and "has no room for collective action."[27]

[26] Dean Spade, "Solidarity Not Charity: Mutual Aid for Mobilization and Survival," *Social Text* 38, no. 1 (March 1, 2020): 136. https://doi.org/10.1215/01642472-7971139.

[27] Sarah E. Bond and Joel Christensen, "The Man behind the Myth: Should We Question the Hero's Journey?" *Los Angeles Review of Books*, August 12, 2021.

To avoid Robin falling into the savior trope, we decided that rather than a hero, Robin would serve as an agent who, through story arcs, helps to connect members of the community. While Robin would be helping their community, it wouldn't be a one-sided support relationship in which Robin supports everyone without receiving support themselves. Instead, the characters Robin interacts with throughout the game would support Robin when possible and — at the end of the game — come together to help save the B&B. This would also help us prevent presenting the characters Robin helps as only victims of their situations, but rather complex characters in their own right.

One example which demonstrated how these ideas were handled in the narrative involved a character named Ruth and their story arc, in which Robin helps Ruth with various tasks when she is injured in an accident that is tangentially Robin's fault. In return, Ruth teaches Robin sustainable ways to expand the B&B; Ruth insists that she not be viewed as a victim or lesser than for her injury, and — in an act of reciprocity — helps Robin with their problem.

2.3.2 - Gentrification

Since our game is set in a struggling town with a tourist economy and is meant to serve as an allegory for human situations in late-stage capitalism, we knew that our game would touch upon gentrification. Gentrification is a wide-spread, tangible issue — and one that Robin's best friend, Casse, is a clear victim of. We knew that we had to do a substantial amount of research to talk about the issue through our game properly. Given the scope of this project, the New England setting of the game, and our proximity to the area, we decided to focus our research primarily on gentrification in an area of particular interest to the institution behind this MQP: Worcester, Massachusetts.

According to planning activist Dr. Kate Shaw of the University of Melbourne, gentrification is "a generalised middle-class restructuring of place, encompassing the entire transformation from low-status neighbourhoods to upper-middle-class playgrounds."[28] It is the process in which a poorer or otherwise marginalized demographic is forced out of the community they once occupied, typically through economic means, and replaced by wealthier individuals and establishments. This can result in the displacement of former residents, increased risks of poverty, exacerbated social inequalities, and the loss of community and culture.

Oftentimes, gentrification starts a cycle of poverty and homelessness which is hard to escape, even with homeless shelters. For reference, even though Massachusetts has a "right to shelter" law which is supposed to ensure the unhoused have a place to stay, it doesn't apply to "homeless individuals without kids nor to any person with an outstanding arrest warrant, and people with criminal records are routinely denied rentals and leases, affordable or not."[29]

[28] Kate Shaw, "Gentrification: What It Is, Why It Is, and What Can Be Done about It," *Geography Compass* 2, no. 5 (September 2008): 1697–1728, https://doi.org/10.1111/j.1749-8198.2008.00156.x.
[29] Maia Hibbett, "What Happens When Gentrification Comes to a Postindustrial City?," *The Nation*, February 1, 2019.

Homeless shelters can often be dangerous environments; Worcester's main shelter is a "wet shelter" (allows drinking) and has a 9:1 male-female distribution, which can dissuade those in need from using it. Homeless individuals also face a higher risk of criminalization, which only continues the cycle. In fact, "Worcester police made over 2,000 arrests or citations for quality-of-life infractions in a less-than-two-square-mile plot that includes Union Station and Kelley Square,"[30] a place with a high homeless and at-risk population.

While our game is meant to be a cozy and fun game, we aimed to ensure that the tone of the cutscenes where Robin encounters gentrification are serious and treat the topic with the respect and importance it deserves. We weren't able to go in depth about the effects of gentrification in the scope of our demo, though we did set up and plan out relevant storylines in the game. We chose to represent it through members of the community being unable to continue living in the community, such as Casse not being able to afford to stay in Arboro and keep the general store running. We planned for Casse to periodically send letters, in which we could touch upon some of the ramifications of gentrification: loss of community, homelessness, and instability. In addition to Casse's storyline, we see other characters struggle to stay in Arboro and planned for storylines in which Robin interacts with gentrifiers.

[30] Hibbert, *Gentrification*.

3 - Pre-Production

This chapter recounts the pre-production stage of our demo's development. Here, we focus on what each of the subteams did in this stage and how our work led up to showcasing at Protofest.

3.1 - Introduction

Pre-production is the beginning of the game development process, and its importance cannot be understated for its role in establishing the foundation on which the rest of the project relies. During this phase, teams define the scope of the game, study comparables, test prototypes, conduct research, and determine the practices and workflow the team will follow.

Our team was keen on developing our game in a manner that most game studios in the industry would; as such, this included our own period of pre-production. Additionally, since we had less than a year to develop our game's demo, it was imperative that we used our time in this stage as efficiently as possible. In particular, we wanted to use pre-production to create our game's pillars, develop our practices and workflows, prototype, and determine the game's scope.

We started our team's pre-production in A-Term (the first half-semester of WPI's calendar, starting in late August), and one of the first things we decided to do was have a premortem meeting. This entails the entire team coming together to discuss potential problems that may come up during the game's creation and ways to remedy or mitigate those problems. The issues we discussed were regarding team interaction, the game product, the development process, scope, and disaster scenarios such as losing a member for a substantial period of time. Coming up with definitive solutions to these problems was secondary to getting them out in the open early on as part of the process; recognizing potential issues the team might face and addressing them was an essential step in building a culture of clear and honest communication within the team.

During our premortem, we also decided to split into six subteams, with each subteam having two to three members and being responsible for a particular part of the game's development. These subteams were Production, Writing, Design, Art, Programming, and Marketing; the conspicuous absence of an Audio subteam was due in part to the team's plans to rely upon several Independent Study Project (ISP) students to fill the role in the latter half of the year.

Splitting into these subteams allowed us to work somewhat independently, which was important for a team of our size and whose members were actively taking other classes. To ensure the subteams communicated well and stayed on track, we held weekly meetings; to help coordinate the teams, we created and tracked tasks to measure each other's progress between the weeks. We hoped that these plans, in conjunction with the several large milestone events scheduled throughout the year, would allow the team to achieve a balance of independent and cooperative work — a hope that wasn't perfectly realized, instead requiring considerable internal restructuring at the end of pre-production to correct.

3.2 - Subteams

With the subteams decided and assigned, pre-production was the part of development where each group had to prepare for the upcoming production. This involved figuring out processes, tools, and decisions necessary for proper collaboration and unified ideas leading out of this phase. It also involved communicating ideas to make sure each subteam was on the same page.

3.2.1 - Production

When groups of people come together to create a product or creative work, an amount of organization and problem-solving is required. In game development, this role is usually handled by a single or small collection of people, who act as managers of a project in a role designated as producers. Producers are in charge of making sure targets are hit on time, solving problems within the team, and attempting to mitigate future problems that could arise due to poor planning or lack of team cohesion. In our team, two members took on this role: one from the art side and one from the technical side. These team members decided to follow in the footsteps of professional producers working in the industry, choosing a production methodology and adjusting it to fit what we needed from it.

Our team decided that because of the inconsistent nature of deadlines when working alongside college classes, as well as the flexibility necessary in game design in response to player feedback and external complications, we would use an Agile methodology. This is an approach to team and task management that focuses on collaboration and continual improvement of internal strategies. Time is broken into chunks, and after each chunk, the team reflects on the successes and failures leading up to that point, and how to continue and change based on what the product (in our case, the game) needs at that stage. We then decided to go with Agile Scrum[31] specifically.

[31] "What Is Scrum?" Scrum.org, accessed March 31, 2024, https://www.scrum.org/resources/what-scrum-module.

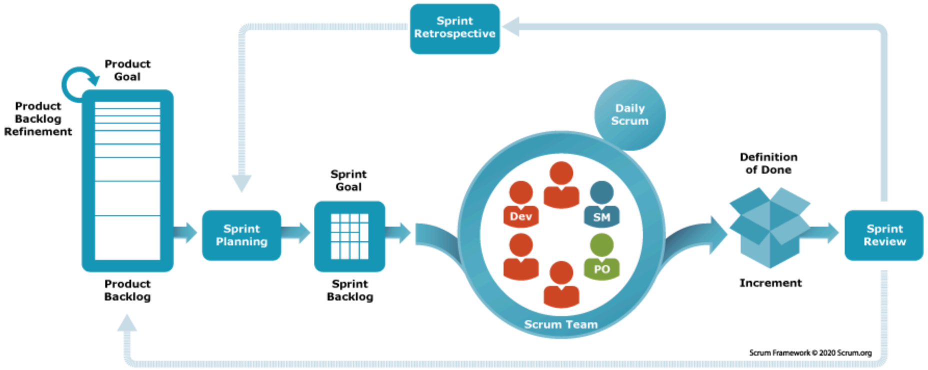
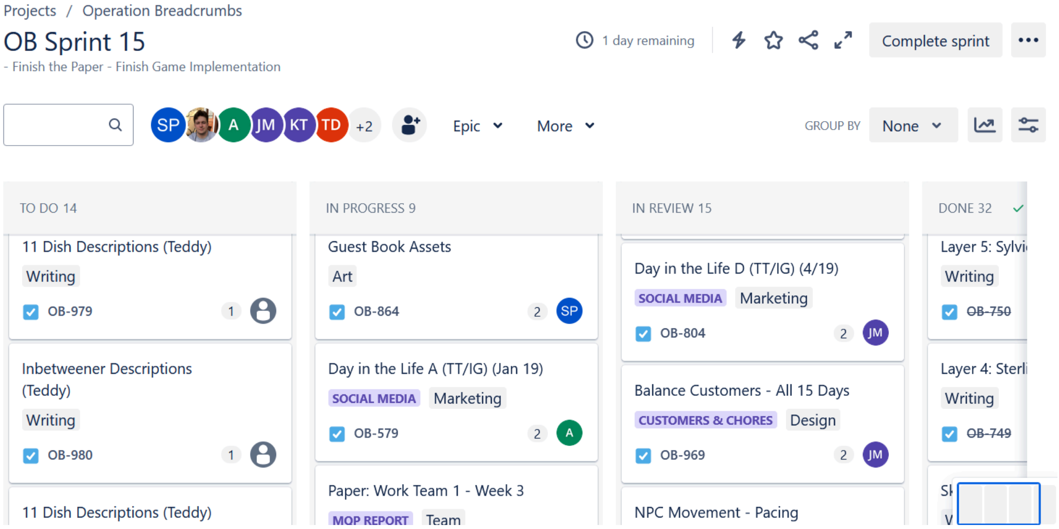


Figure 5: Infographic on the breakdown of Scrum. "What Is Scrum?" Scrum.org, accessed March 31, 2024, https://www.scrum.org/resources/what-scrum-module.

Scrum is a particular implementation of the Agile framework in which time chunks, called sprints, are managed using a variety of processes: tasks are assigned values according to their challenge levels and then picked for sprints from a product backlog by the team in a Sprint Planning meeting, When the sprint ends, the team reflects on what was done in a Sprint Review and what challenges and triumphs were encountered along the way in a Sprint Retrospective. The team used the software Jira[32] to track each of these processes. Though many were inexperienced with Scrum and Jira at the start, the team got the hang of each of these parts and improved them to better fit our needs over the course of pre-production and beyond.



Projects / Operation Breadcrumbs

OB Sprint 15

- Finish the Paper - Finish Game Implementation

1 day remaining

Complete sprint

SP A JM KT TD +2 Epic More GROUP BY None

TO DO 14

11 Dish Descriptions (Teddy)
Writing
☑ OB-979 1

Inbetweener Descriptions (Teddy)
Writing
☑ OB-980 1

11 Dish Descriptions (Teddy)

IN PROGRESS 9

Guest Book Assets
Art
☑ OB-864 2 SP

Day in the Life A (TT/IG) (Jan 19)
SOCIAL MEDIA Marketing
☑ OB-579 2 A

Paper: Work Team 1 - Week 3
MOP REPORT Team

IN REVIEW 15

Day in the Life D (TT/IG) (4/19)
SOCIAL MEDIA Marketing
☑ OB-804 2 JM

Balance Customers - All 15 Days
CUSTOMERS & CHORES Design
☑ OB-969 2 JM

NPC Movement - Pacing

DONE 32 ✓

Layer 5: Sylvi
Writing
☑ ~~OB-750~~

Layer 4: Sterl
Writing
☑ ~~OB-749~~

Sl
V

Figure 6: A later snippet of our Jira board mid-sprint.

[32] Jira, (Atlassian, 2002), PC

At the start of the year, the team's first main struggle was assigning values to tasks. This is an important part of the Scrum process — not just to estimate workload, but also to prioritize tasks, evaluate scope, and keep each developer up to date on the goings-on in parts of the team that they might otherwise be unaware of. The first thing that we tried was assigning time boxes to each task: estimating that this bird portrait would take an hour, while this UI feature would take three. This angle did not work well at all for the team; some had no idea how much time unfamiliar tasks would take, others argued that fun, long tasks were more easy to approach than short, hard tasks. As such, the team pivoted to another strategy often used in Scrum: assigning modified Fibonacci sequence point values (1, 2, 3, 5, 8, 13…) based on how much effort we felt each task should take. This system went over much better with the team; not only did it become easier to plan workloads each two-week sprint, but the team felt more in tune with one another, understanding what other members were doing and how the game was progressing.

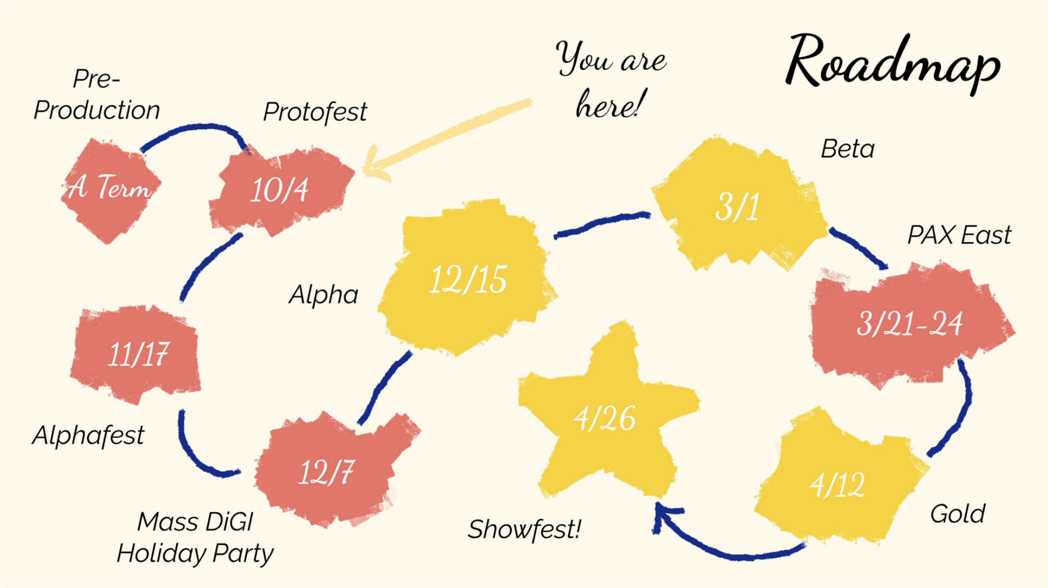


Figure 7: The first interaction of the Alpha-Beta-Gold roadmap developed by the Production team during pre-production.

The final responsibility of the producers in this MQP was to maintain an understanding of the team's progress during development and its intended direction. This was handled in two ways: creating documents to enable communication of each subteam's status, as well as continuous updating of these documents to ensure that time estimates and bumps along the road were accounted for. In hopes of achieving this, the producers created a roadmap at the start of the year to project the progress of the team according to traditional game development milestones such as Alpha, Beta, and Gold — "feature complete," "content complete," and "polish" periods of game development, respectively. Understanding the limited time already available to the team within their one year of development on the demo, the original roadmap planned for a liberal interpretation of these milestones, being “core feature complete,” “placeholder content complete,” and “polish before launch.”

3.2.2 - Writing

The Writing subteam’s job during pre-production was to take the team’s originally agreed upon vision for the game and build upon it. As mentioned before, this basic premise focused on a bird named Robin who inherits their grandfather’s B&B, similar to how the main character of *Stardew Valley*[33] inherits their grandfather’s farm. In keeping with our themes and messaging, we wanted our game to focus less on business optimization and more on the town’s community: rather than having our players grind for collectibles like *Stardew Valley*’s community center quest, we wanted the players to continually interact with and take quests from other characters in the community. To accomplish this, the game’s story would mainly revolve around the other birds in the community. In particular, we wanted it to focus on the power of a supportive community and mutual aid, while criticizing the concept of landlords and consumerism.

The decision to make a demo, rather than a complete game and finished narrative, put the Writing subteam in an odd spot. This meant that we would only be able to complete the introduction arc of the overall story, and then hopefully a few story arcs centering around town characters and travelers staying at the B&B. While this helped to reduce scope, we still wanted to complete the outline for the full game; it was important to have an ending to guide the introduction arc, and it provided a story plan for the game’s potential future development.

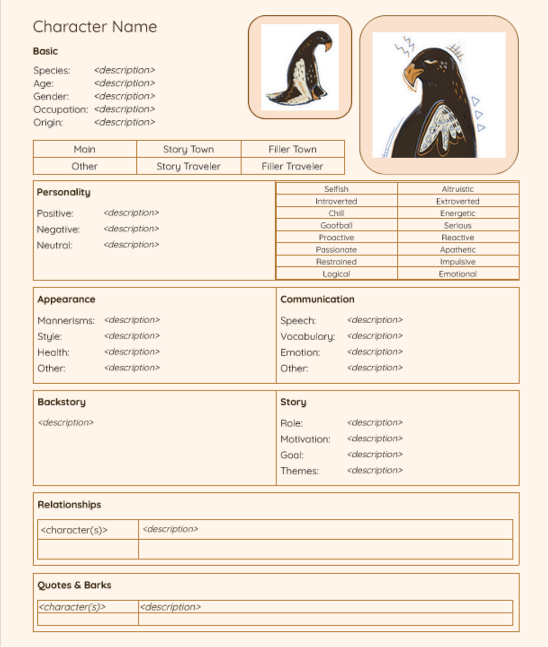
When pre-production officially started in A-Term, we focused on preparing the tools and workflows that we would use throughout development. Our preparation mainly consisted of creating templates for character sheets and story arc outlines and determining the scope of the story. It was important to determine the scope early on as that would inform how many cutscenes, characters, and story arcs we could produce during development. We determined that we could feasibly achieve around five story arcs (not including the introduction arc), each comprising roughly three cutscenes, that would center around five non-playable characters (NPCs). We also created ways to standardize how we would talk about the story arc, relying both on spreadsheets to track the cutscenes necessary for the arc and a template for arc guides — documents that would explain the goals of the arc, how the arc started, how the arc ended, and the characters that were involved.

Early on, we knew we wanted to display the story’s dialogue in a visual novel format, pairing character portraits with text during cutscenes and interactions. We determined that the most efficient way of accomplishing this was with Yarn Spinner,[34] a tool already used by

[33] *Stardew Valley*, v. 1.5 (ConcernedApe, 2016), PC, Mac, Switch, PlayStation 4 or later, Xbox One, iOS, and Android.
[34] Yarn Spinner, (Yarn Spinner, 2015), PC.

published games like *Night In The Woods*,[35] *A Short Hike*,[36] and *Dredge*.[37] For our purposes, Yarn Spinner allowed writers to easily implement and add dialogue in-game, even enabling writers to trigger custom functions made by the programmers, which could be used to move characters, make cinematics, and any number of other effects.

After standardizing our writing format and determining the scope of the story, we moved on to generating characters. To create our characters, we first determined what potential story archetypes we wanted to cover; then, we had several brainstorming meetings where the Writing subteam pitched characters to each other and discussed how, if chosen, their story arcs could play out. To formalize our character generation process, we created template character sheets for each type of character. The character sheets gave each writer an understanding of every character, ensuring that any of the writers would be able to write cutscenes involving any characters.



Character Name

Basic

Species: <description>
Age: <description>
Gender: <description>
Occupation: <description>
Origin: <description>

Main	Story Town	Filler Town
Other	Story Traveler	Filler Traveler

Personality

Positive: <description>
Negative: <description>
Neutral: <description>

Selfish	Altruistic
Introverted	Extroverted
Chill	Energetic
Goofball	Serious
Proactive	Reactive
Passionate	Apathetic
Restrained	Impulsive
Logical	Emotional

Appearance

Mannerisms: <description>
Style: <description>
Health: <description>
Other: <description>

Communication

Speech: <description>
Vocabulary: <description>
Emotion: <description>
Other: <description>

Backstory

<description>

Story

Role: <description>
Motivation: <description>
Goal: <description>
Themes: <description>

Relationships

<character(s)>	<description>

Quotes & Barks

<character(s)>	<description>

Figure 8: Writing subteam's character sheet. See Appendix D for enlarged version.

In order to keep track of all these documents, our workflows, and the game's worldbuilding, we created the Writing Bible, a "living" document linking everything related to the story. It was called a living document as it would continually grow, even after pre-production, as writers would add information such as facts about the town, the characters of the game, and links to the narrative arcs. The Writing Bible also detailed our processes, giving future writers onboarded after pre-production a document they could reference.

[35] *Night in the Woods*, (Finji, 2017), PC, Mac, Linux, Switch, PlayStation 4 or later, Xbox One or later, iOS.
[36] *A Short Hike*, (Adam Robinson-Yu, 2019), PC, Mac, Linux, Switch, PlayStation 4 or later, Xbox One or later.
[37] *Dredge*, (Team17, 2023), PC, Switch, PlayStation 4 or later, Xbox One or later.

At the end of pre-production, the Writing subteam had outlined the major arcs of the game, created documentation that enabled writers to reference key material, and standardized our workflow. We would now be ready to easily onboard new writers and begin writing the cutscenes.

3.2.3 - Design

During pre-production, we had many broad, sweeping discussions about the overall systems of the game in order to explore the depth of the design space and find a version of the game that would be feasible to create, fun to play, and demonstrate each team member's individual skills. At first, this took the form of discussing the balance and timing of kitchen-related tasks, general B&B-related tasks, and exploring outside the B&B. This resulted in the conceptualization of three different "dirty builds," which are rough draft versions of the game we could quickly program and test against each other.

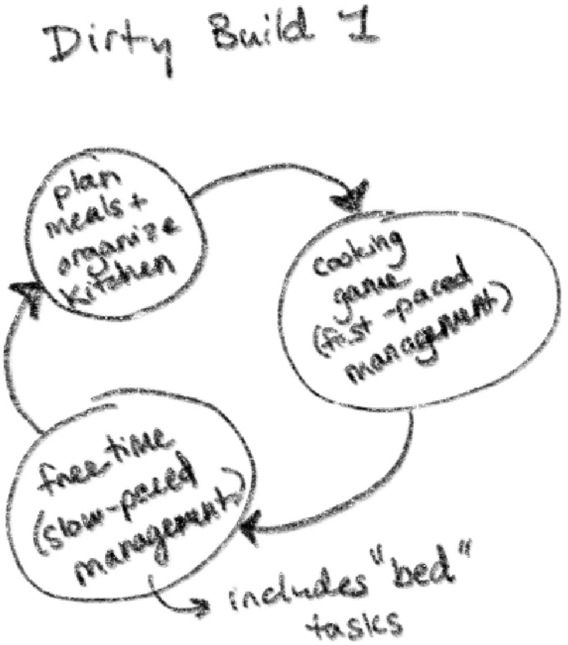

Dirty Build 1
plan meals + organize kitchen
cooking game (fast-paced management)
free-time (slow-paced management)
includes "bed" tasks

Figure 9: Diagram of the first dirty build's core gameplay loop.

The first dirty build focused on kitchen tasks, allowing the player a certain amount of time to focus on cooking food for birds that have ordered specific meals. This is done by directly placing held ingredients into appliances, similar to cooperative cooking games like *PlateUp!*. Following this segment of gameplay is a slow portion in which the player completes other tasks in the B&B like making beds, interacting with birds staying at the B&B and the townspeople, and foraging for ingredients throughout the world. Finally, this build included a portion of time at the end of the day where the player could organize their kitchen, preparing food for the next day.

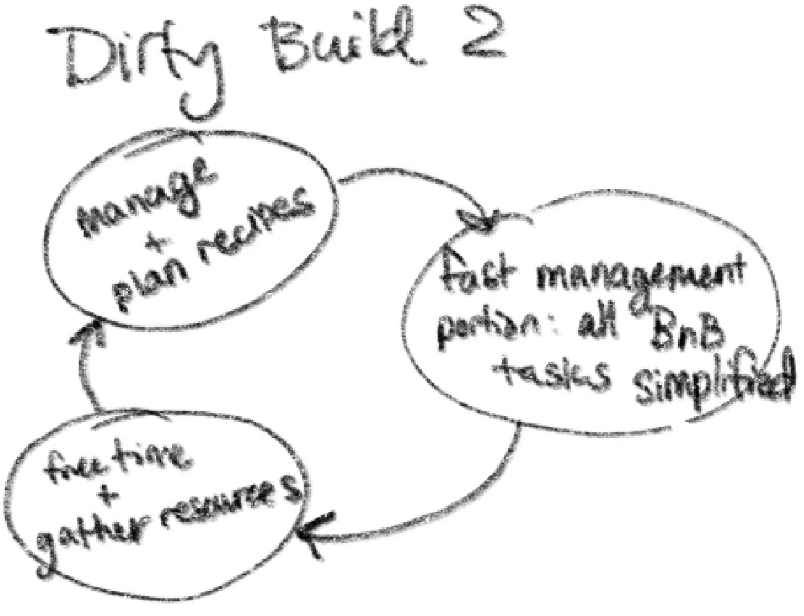

Dirty Build 2
manage + plan recipes
fast management portion: all BnB tasks simplified
free time + gather resources

Figure 10: Diagram of the second dirty build's core gameplay loop.

In the second dirty build idea, the game was far more focused on general B&B upkeep, reducing cooking to pre-set meals that are assigned to birds during the evening of the previous day. There were also a large number of simple tasks to complete throughout the B&B to keep travelers happy. In the next timed section of this version of the game, the player could gather resources and interact with townspeople, continuing the storyline.

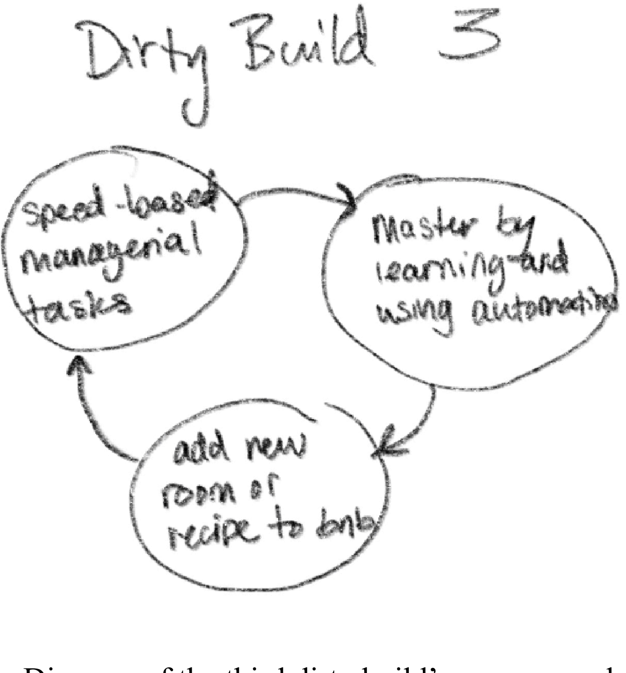

Dirty Build 3
speed-based managerial tasks
Master by learning and using automation
add new room or recipe to bnb

Figure 11: Diagram of the third dirty build's core gameplay loop.

The third dirty build took a different approach to the overall game, placing both complex cooking (though not as complex as the first dirty build) and general B&B tasks during the same timeframe, requiring players to traverse quickly throughout the tree to achieve everything within the time limit. After this, the player would have an unlimited amount of time to explore the town, chat with townspeople, and gather resources; they could also modify their B&B with new rooms and learn new recipes by interacting with others or experimenting in the kitchen.

In the end, we opted instead to pick and choose the team's favorite features from each design plan. From the first dirty build, it was decided to focus primarily on cooking, keeping only a small number of general B&B tasks to complete. Platforming and flight from the third dirty build would be borrowed, without the strict timers and speed-based managerial tasks, keeping more in line with other builds' cozy-game aesthetic. From all these plans the team decided upon the gameplay split, dividing the day into a task management section and following it with B&B construction and story progression.

After making their choice of mechanics and gameplay formula, the Design subteam used the ending of pre-production to begin testing a potential cooking system. Making plans for Protofest, the first event milestone the team planned to use to gauge their progress, Design began creating a card game to demonstrate and test the cooking system we had designed. Over the course of several meetings, we would create a collection of bird, ingredient, and appliance cards. Birds would have flavor preferences, ingredients would have flavors and particular effects that occurred when cooked, and appliances would transform those ingredients in certain ways, preparing them to be served to birds for consumption. Creating this card game involved numerous decisions regarding the number of ingredients, number of flavor types, and how we would try to balance the flavor values. As part of making all these choices, the team began to become wary that some of our initial ideas such as garnishing and cooking multiple meals at the same time might be too complicated for the card game format — an important indicator once we began to test the card game that drove many of our later decisions in developing the cooking mechanic of the actual game.

3.2.4 - Art

The vast majority of current video games utilize visual art of some kind. Save for the early, text-based games like *The Oregon Trail*[38] or the modern interactive fiction community, it is generally agreed upon that visual art is an efficient way to engage players and add detail to a game's world and story. As such, it is common for an art style to be the first thing a prospective player will notice along with the title; it carries the heavy burden of being unique enough to stand out from other games while not being so unfamiliar to players as to turn them off entirely. It also must work well with the tone of the game, supporting the narrative through visual

[38] *The Oregon Trail*, (Minnesota Educational Computing Consortium, 1975), HP 2100 Minicomputer, CDC Cyber 70/73-26 Mainframe Computer.

storytelling, while reinforcing game mechanics to ease the player's understanding. Suffice to say, choosing and implementing an art style that works for the game is crucial for creating a smoother experience for the player**.**

Thus was the pre-production task for the Art subteam made apparent, with artists being responsible for the creation and development of the game's art style during this period, starting with research and initial style tests. Early on, the decision was made to focus on American artists, to match the game's setting of American New England with regionally-accurate birds. Given that, and the fact that the story itself explores issues familiar to New England cities and modern America, we thought it only appropriate that our art style was influenced by American painters and movements. We gathered a selection of artists that we were inspired by and spent time picking which ones we would decide to model our style after.

Many American art movements hold roots in Black American culture/artists as well as African culture and art, but are often not recognized. Thus it was extremely important for us to give equal, if not more, attention to Black American artists. Choosing to test three styles — calling them "impressionist," "scribbly," and "storybook" — we looked to the following artists for inspiration: Childe Hassam and Frank W. Benson for the impressionist style; Jacob Lawrence, Jean-Michel Basquiat, and Henri Matisse for the scribbly style; and Aaron Douglas, along with Disney in general, for the storybook style.

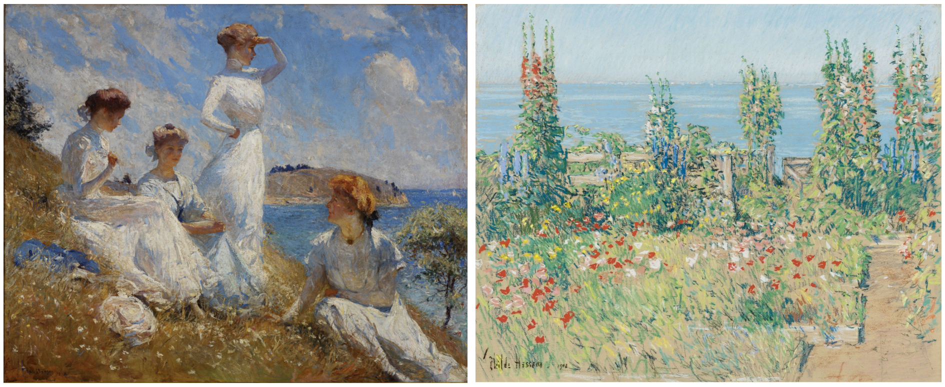


Figure 12: Impressionist artwork by Frank W. Benson, *Summer*, 1909 (left) and Childe Hassam, *Hollyhocks, Isle of Shoals*, 1902 (right).

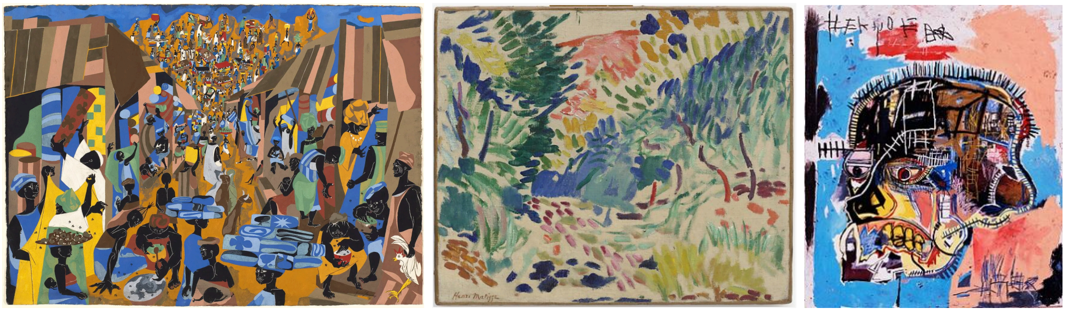


Figure 13: Artworks by Jacob Lawrence, *Street to Mbari*, 1964 (left), Henri Matisse, *Landscape at Collioure*, 1905 (middle), and Jean-Michel Basquiat, *Untitled (Skull)*, 1981 (right).

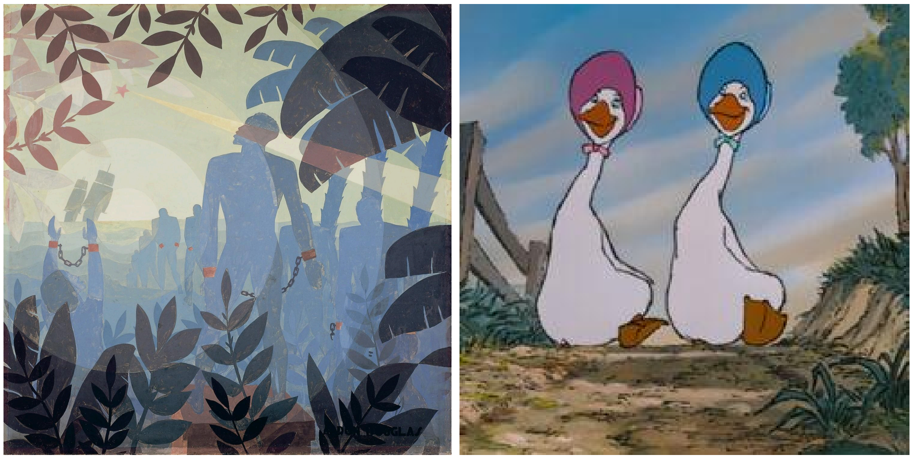


Figure 14: Artworks that inspired the "storybook" style. Aaron Douglas, *Into Bondage*, 1936 (left) and "Geese scene," directed by Wolfgang Reitherman at Disney, *The Aristocats*, 1970 (right).

We also went outside of traditional American artists, taking inspiration for our scribbly style from the use of warped perspective and interesting patterns by Cartoon Saloon, an Irish Animation studio focused on bringing traditional stories to modern audiences, as well as Spry Fox LLC's *Cozy Grove* for our storybook style.

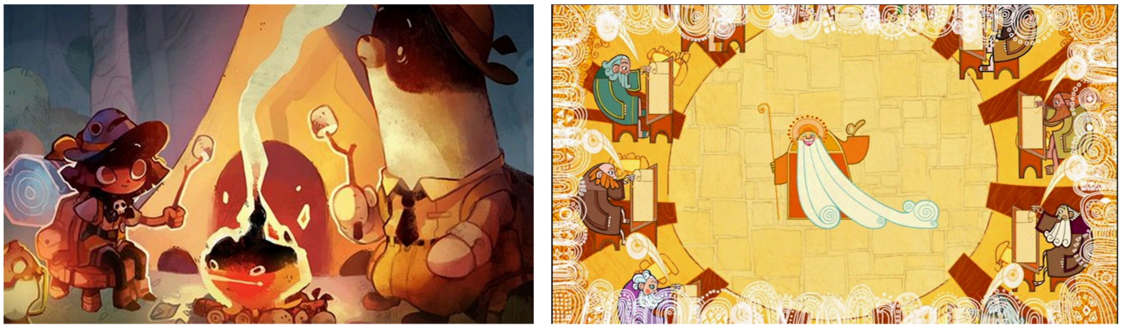


Figure 15: *Cozy Grove,* developed by Spry Fox, PC/Mac, 2021, screenshot (left) and "St. Colmcille scene," directed by Tomm Moore at Cartoon Saloon, *The Secret of Kells*, 2009 (right).

For each style, tests were done to compare designs for the major aspects of the game: one main character, one background character, one store, one tree, and one dish.

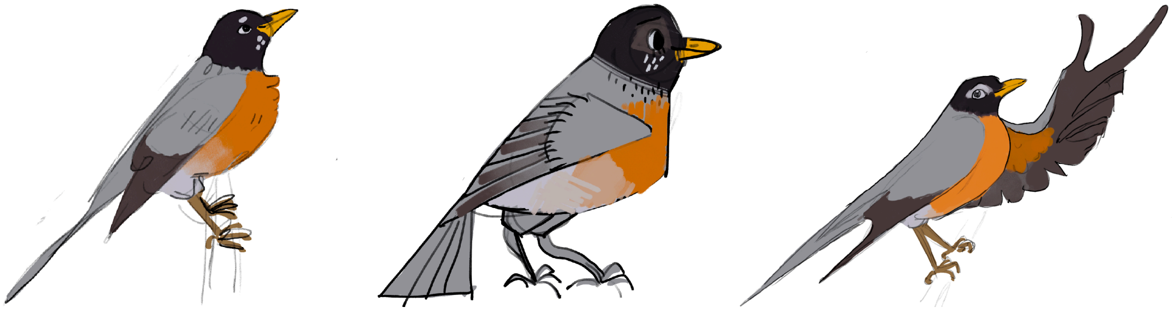


Figure 16: Main character birds in the three styles: impressionist, scribbly, and storybook.

Through feedback sessions with one another and the project's advisors, we analyzed the styles, giving special consideration to the following self-imposed requirements:

1. **Readability as Game Art**, such that it does not conflict or obstruct gameplay
2. **Narrative Fit**, such that the style is compatible with the chosen game narrative
3. **Uniqueness**, such that the style does not remind players of any existing game or IP
4. **Ease of Replicability**, such that all members of the Art subteam could work in the style within the allotted time frame for production.

When narrowing down the style, we also felt it important that the choices made were mindful of the artists whose work we were taking inspiration from. For this reason, we were drawn more toward artists whose work was meaningful and told messages that we supported and which carried the spirit of the story we were trying to tell. For example, the works of Jean-Michel Basquiat, an African American man who grew up in poverty and made art with the intention to express himself and his experiences,[39] felt meaningful to our story in the way that Childe Hassam's educational trips to Europe to adopt the popular style of Impressionist Painting did not.

After significant iterations, and taking all these aspects into consideration, we ultimately chose the scribbly style, whilst incorporating elements from the storybook style — such as the shape language of the characters and the creative use of perspective. From then on, we began perfecting our hybrid-scribbly style, along with defining standards, specifications, and significant documentation, including an Art Bible and supplemental written guides detailing how various aspects of the art were created.

We used many tools to create, store, and share the art during this ideation process, most of which continued to be used throughout the game's development, as this pre-production period provided us the chance to test our initial workflow and processes. Google Drive was used for file organization and storage, Trello was used for pipeline management, and the artistic tools chosen

[39] Jean-Michel Basquiat, interview by Geoff Dunlop and Sandy Nairne, *State of the Art*, Illuminations, UK, January 11 1987.

for creating final game art were Clip Studio Paint, Adobe After Effects (AE), and Adobe Media Encoder.

With the art style and tools chosen, the art team then readied to create our internal tools: our style guides and art bible. Style guides are a vital part of working on a team to create a cohesive art project with a distinct and uniform style across, particularly on a large one. With multiple artists working on the same types of assets across the game, the process of creating each asset had to be approached in the same way. Easily understandable, visually communicative guides had to be created to enable this form of collaboration. During pre-production, the artists identified key places where guides would be needed, and made tasks to create these guides as aspects of the art style were figured out down the road.

As the Art subteam moved from pre-production into production, we had made most of the major decisions regarding style (minus UI designs) and had streamlined our workflow and feedback process; finally, we were ready to begin creating game-ready assets.

3.2.5 - Programming

A video game is a collection of interwoven systems that encourage the player to interact with them to produce an experience. While the messaging and story of that experience are determined by a game's writers, the systems and gameplay loops decided upon by its designers, and the visual style dictated by its artists, the physical act of implementing and scripting the mechanics used to deliver these features and produce a viable game falls to its programmers.

For this technical role, software and game development alike both benefit from healthy pre-production and planning. Jumping straight into development of either without proper preparations is asking for trouble, but outlining every aspect of a game's development before production begins fails to account for the highly volatile and changing nature of projects of this sort. As such, the Programming subteam's goals of pre-production can be best summarized as identifying the tools required to accomplish this project and formalizing their usage for developing the project from a technical perspective.

Perhaps the largest decision to be made in a technical team's approach to game development is their choice of game engine for the project. Game engines are software frameworks designed for the development of video games, and while not every engine fits every project, the right one can propel development forward, saving months or even years of work that would otherwise be spent setting up core systems such as rendering pipelines, resource managers, and physics engines. In the case of *Operation Breadcrumbs*, the Unity game engine was chosen. Despite this engine undergoing some recent controversy regarding company royalties and pricing model changes, the team determined Unity to be the best fit for us, due to both the programmers' existing experience with the engine and Unity's affinity for simple 2D game development.

Following this decision, the Programming subteam identified and selected various software packages that would further excel development by extending the engine's functionality and providing a larger toolbox for us throughout development. Examples of this include DoTween,[40] a tweening library that provides code-based animation, and Yarn Spinner, the dialogue framework described in Section 3.2.2. Formalizing how these packages and software would be used required programmers to learn with hands-on experience; short technical demos and internal documentation were created after experimenting and researching the software to ensure we could make the most of our features as part of including them in the project. Other tools were also determined at this stage, such as our choice of source control software, Git and GitHub, and integrated development environments, such as JetBrains Rider.[41]

After codifying the tools we would utilize in the project, the programmers' next task for pre-production involved planning and mapping the game's general systems. By creating a diagram to visualize how different mechanics fed into and interacted with one another, the Programming subteam was able to foresee how large systems would function throughout development, allowing us to plan ahead as part of the project's development.

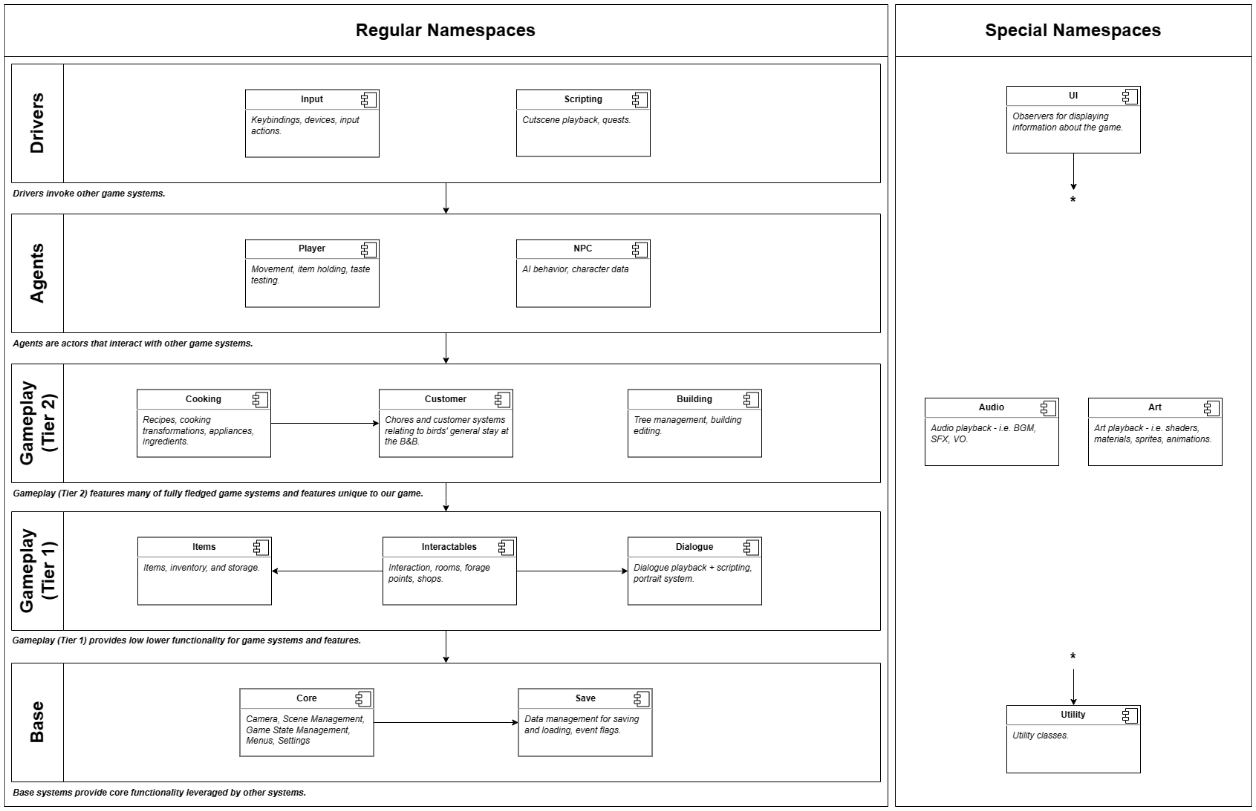

Regular Namespaces
Drivers
Input
Keybindings, devices, input actions.
Scripting
Cutscene playback, quests.
Drivers invoke other game systems.
Agents
Player
Movement, item holding, taste testing.
NPC
AI behavior, character data
Agents are actors that interact with other game systems.
Gameplay (Tier 2)
Cooking
Recipes, cooking transformations, appliances, ingredients.
Customer
Chores and customer systems relating to birds' general stay at the B&B.
Building
Tree management, building editing.
Gameplay (Tier 2) features many of fully fledged game systems and features unique to our game.
Gameplay (Tier 1)
Items
Items, inventory, and storage.
Interactables
Interaction, rooms, forage points, shops.
Dialogue
Dialogue playback + scripting, portrait system.
Gameplay (Tier 1) provides low lower functionality for game systems and features.
Base
Core
Camera, Scene Management, Game State Management, Menus, Settings
Save
Data management for saving and loading, event flags.
Base systems provide core functionality leveraged by other systems.
Special Namespaces
UI
Observers for displaying information about the game.
*
Audio
Audio playback - i.e. BGM, SFX, VO.
Art
Art playback - i.e. shaders, materials, sprites, animations.
*
Utility
Utility classes.

Figure 17: Programming subteam's systems planning document.

[40] DoTween, (Demigiant, 2014), PC.

[41] Rider, (JetBrains, 2017), PC.

This mapping is depicted by the systems diagram shown above. Although this is another instance of a living document expected to undergo significant changes and alteration throughout development, the initial diagram served as a big picture guide for the subteam, indicating how lower-level systems are influenced and leveraged by higher ones. As requirements for the game and its systems were fed to the Programming subteam through the Writing, Design, and Art subteams, this document evolved to broadly map out the mechanics and systems required by each subteam. By mapping systems in this manner, the Programming subteam could identify how major game features and mechanics interact. Working towards achieving this tight interaction of systems is the crucial step meant to ensure the successful development of video games as a software.

By the end of pre-production, programmers had identified, learned, and tested the tools they planned to use. It also created a visualization of major systems and their place in the software. Now poised to begin development, the subteam launched early into the production phase, beginning the development of the core systems identified as the underpinnings of the video game.

3.2.6 - Marketing

Having a marketing team for an MQP is quite atypical, but such is not the case for professional game development teams. Marketing is crucial to ensure a game reaches a wider audience; thus, early into pre-production, the team decided that dedicating resources towards a marketing team would be a worthwhile investment in the end. However, this decision was made with the stipulation that the Marketing subteam be as efficient as possible to prevent too much time and energy being taken from the development of the actual game itself. As a result, this team was the smallest of the bunch, tied with Production.

With the significance of efficiency in mind, the Marketing subteam used the start of their pre-production phase to dive headfirst into research. We analyzed the social media accounts of popular indie studios and games — Snoozy Kazoo's *Turnip Boy Commits Tax Evasion,*[42] Aggro Crab Game's *Another Crab's Treasure,*[43] and Voracious Games' *Potionomics*[44] — while paying close attention to how their marketing teams grew their audience while encouraging engagement. This research had the bonus of providing plenty of inspiration and source material, as Marketing began tracking post ideas that our team could easily put our own spin on.

Through our research, we decided that the best platforms to target were X (formerly Twitter), TikTok, and Instagram (specifically Instagram Reels), finding that these platforms had the most potential for organic growth. In addition, the team picked Steam, one of the most prominent digital video game distribution platforms, to host the game, with all marketing

[42] *Turnip Boy Commits Tax Evasion*, (Graffiti Games, 2021), PC, Mac, Linux, Switch, Xbox One or later, iOS, Android, PlayStation 4 or later.
[43] *Another Crab's Treasure*, (Aggro Crab, 2024), PC, Switch, PlayStation 5, Xbox Series X/S.
[44] *Potionomics*, (Xseed Games, 2022), PC.

pointing towards it as the release date approached. After creating the *Operation Breadcrumbs* accounts, we recorded and organized the information in our Marketing Bible. This document is meant to hold all account information, visual information — such fonts, colors, and dimensions — and posting guidelines — such as hashtags and captions. As yet another instance of a living document for the team to update and maintain, it was important to note that most of the major details essential to establishing the game's branding could not be decided until the game's art style, narrative, and core gameplay loop had been solidified.

With research done, socials created, and as much of a Marketing Bible that the subteam could create during this phase established, we began a draft Posting Schedule document. This consisted of a calendar spanning all of production with planned posts made on a weekly basis across platforms; we came up with ideas for various series of posts, or posts with similar themes that would be posted on the same day each week. This document, in addition to the Marketing Bible, was constantly updated and changed as the project's vision became clearer.

Once this foundation was set, the first instances of posting content began. The subteam had an array of tools at hand, such as TikTok, CapCut,[45] and Adobe After Effects for video editing, and Procreate and Clip Studio Paint for graphics. In our attempt to begin posting as soon as possible despite our lack of content, the subteam decided to create bird-themed posts, such as posting bird facts or creating bird-sized breakfasts. Though we did see engagement with these posts, we soon became aware of a quickly approaching roadbump: we had begun growing an audience for bird enthusiasts rather than an audience for our game.

While researching Voracious Games' *Potionomics*, we found that despite the team's great success in gathering an audience through marketing, they struggled to keep that audience. On their Steam page, players complained that they were blindsided by a very fast-paced and stressful management simulator rather than the calming potion-brewing dating sim that they were advertised. "I wanted a relaxing shopkeep game. Instead I get a stress simulator," reads one review.[46] Seeing this series of events beginning to unfold in our own marketing campaign, the team decided to take a step back and revamp the content being posted. Bird-related videos quickly changed to more relevant posts such as character introductions, narrative-related jokes, and quirky digs at the game development process.

By the time our pre-production had finished, we were confident in the new, more relevant content we planned for. Stepping into production, the goals that the Marketing subteam had begun this process remained unchanged: it was time to grow our audience.

[45] CapCut, (ByteDance, 2019), iOS.

[46] Dandere-Hivemind, Steam review of *Potionomics* on December 24, 2023, Voracious Games (2022), accessed April 5, 2024, https://steamcommunity.com/profiles/76561198045738147/recommended/1874490/.

3.3 - Protofest

This pre-production period is best characterized by each discipline's completion of independent work critical to future game development. Production assessed deadlines, composed roadmaps, and prepared product management tools; Writing developed a main scenario and characters; designers weighed different potential gameplay styles and formats; Art developed a cohesive art style and guides; Programming standardized their software tools and planned the structure of their codebase; Marketing created a coordinated social media campaign and scheduled posts coinciding with the future terms. This approach empowered teams to determine their own work and requirements necessary to set themselves up for future success, yet at the inevitable cost of team cohesion and upholding a shared vision. This shortcoming became readily apparent to the team about two-thirds of the way into A-Term and pre-production, coming to a head with an event called Protofest.

Protofest is an event hosted by IMGD designed to showcase early plans for MQPs and student projects. It is also the first in a line of showcase events organized by the department, its existence an important milestone for ensuring that pre-production and early project development is on track. Naturally, our team was compelled to participate, though our showing at the event encapsulated many of the strengths and weaknesses of our pre-production style.

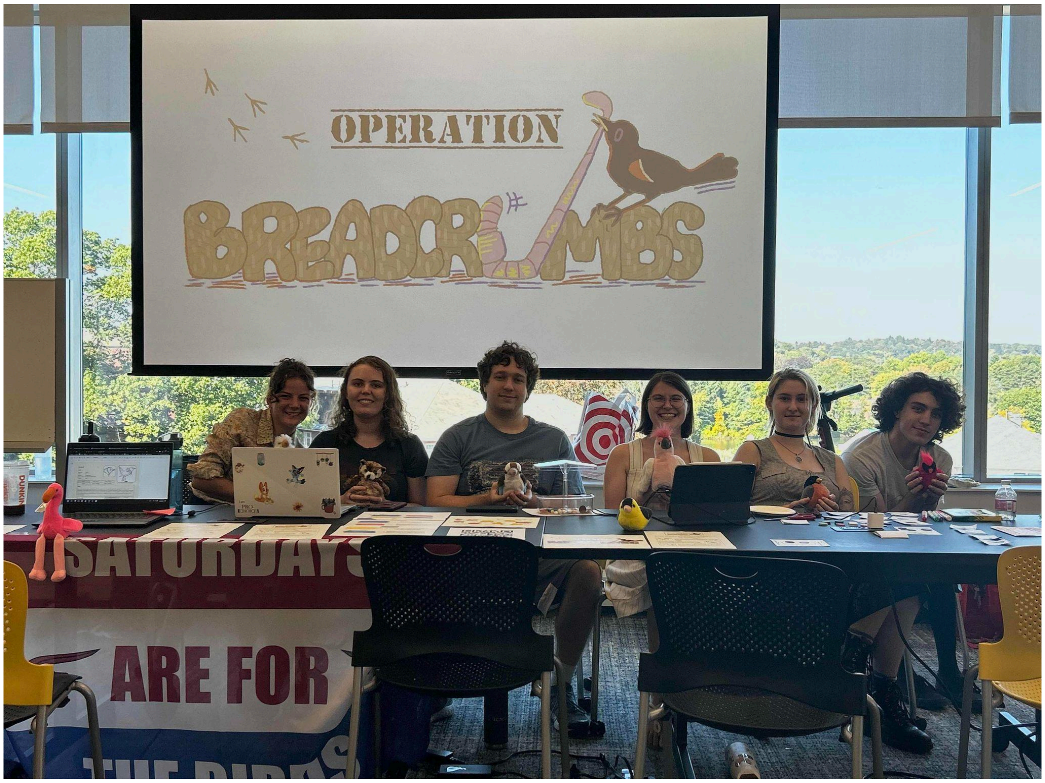


Figure 17: *Operation Breadcrumbs*' team and booth for the Protofest Event.

For Protofest, many of our subteams were responsible for producing showcase materials to convey the work they'd accomplished thus far throughout pre-production and demonstrate their future plans for the project. Character sheets, a prototyped card game, and slideshow on the game's chosen art style were just some of these materials present at our team's booth, yet all were developed almost entirely independently within their respective subteams. In doing so, the Protofest booth showcased several distinct visions for the game, many of which were even news to ourselves. Writing had developed the game's world which introduced elements that were unrecognizable to the original pitch; the historical messaging and themes — rooted at the heart of the planned art style and its inspirations — hadn't reached those outside of the Art subteam; members not on Design were completely unfamiliar with the prototyped card game and its rules.

The strength of this discipline-centric approach to pre-production was made readily apparent in the abundance of content the team was able to showcase at the booth, yet so too was its weakness. In self-contained units, the team struggled with a lack of interdisciplinary communication regarding decisions and drifted from their original shared vision.

The final weeks of pre-production and A-Term instead focused on a serious restructuring and pivoting of plans to foster better working relationships between the subteams. Decisions made by independent subteams were walked back, such as eliminating some of the elements from Writing's world-building and mechanics from Design's card game. Regardless of the painful nature of cutting this work that had already been done, the team knew such action was essential to refocus on the original shared visions for the game.

In addition to realigning everyone's vision of our final product, the team also found it necessary to acknowledge where we came from: we were a team of six white Americans coming from a privileged worldview. If we were going to make an inherently political game, we had to ensure that the themes that we were discussing were presented in the appropriate manner — including proper, deep research into the socio-economics at play. A more thorough round of research regarding these corrupt systems, how they bled into other topics such as race, and hearing from those affected was conducted.

From this new perspective, the team then reevaluated the key plot and worldbuilding elements. The game would still follow the player character, Robin, as they revitalized and regrew a community through interactions centralized at the B&B. Yet the scenario Robin faces, and the state of the B&B, would more closely reflect that of the wider town and community, pushing them onto the decline ever since their lands had been inherited by the corrupt landlord issuing severe rent hikes. To combat this, Robin and the other birds in town would need to participate in community interactions and mutual aid efforts — forming a "rebellion" the likes of which may be attainable for any player.

Despite this all, we knew these specific changes to the narrative and the reversals of several other subteams' efforts alone would not have been enough to safeguard us against future derailments. Straying from the game's original shared vision had arisen from a lack of consistent communication between the subteams, a fault that was bound to cause problems again were development plans to resume as they had. Correcting the issues that manifested from Protofest meant not just dealing with its symptoms, but addressing the root cause beneath — especially the self-contained nature of the subteams.

In the original structure of the subteams, there was very little commingling between the IMGD Bachelor of Arts (BA) and Bachelor of Science (BS) students on the team. While it was expected that IMGD BA students work in the Art subteam and BS students in Programming, overlaps in the other four subteams were scarce. Only one BA student worked in Design and Writing, and no BS students were involved in art-centric subteams such as Marketing. Restructuring was necessary to foster better communication, but it was also something that team members found themselves wanting to do. The nature of the project, and the opportunity to work in new subteams outside of the standard Art and Programming which BA and BS students were familiar with, is unique to an MQP of this type and something the team wanted to make better usage of.

By the end of the team's internal restructuring, five of the six team members joined new subteams or changed which subteams they were already a part of, allowing for significantly more collaboration between BS and BA students. As part of this, the team also recommitted themselves to being cognizant of interdisciplinary communication by adjusting how we conducted team meetings, incentivizing all members to comment and report on the updates of each discipline. As the team left pre-production, a new philosophy was adopted, existing as a broad consideration which would color much of the future discussions and development for the project, as well as this paper: How can we approach problems and find solutions with interdisciplinary collaboration in mind?

4 - Alpha

As part of this chapter, we explore how the subteams transitioned out of pre-production and into the first stage of development. This section pays special attention to the work carried out across teams to create and implement the game's core loop and world, concluding with how these changes defined our experience with Alphafest.

4.1 - Introduction

As the team exited the pre-production phase, we were quite eager to start full scale development. Having recommitted to the game vision with a new focus on interdisciplinary collaboration, it was finally time to begin creating the game and its many features.

When beginning development on a project with an extended production timeline such as a video game, it can be easy to immediately begin making as many features as possible, as it's always exciting for the team to see the bits and pieces of their shared vision come to life through gameplay, visuals, and storytelling. However, the team didn't want rushed development to lead to the finished product being made up of mediocre and half-baked ideas, where pieces of an original vision don't correctly align to form the larger picture. In this sense, it was important to the team, especially with our limited timeline, that the portion of the game we would be able to create not only reflected the shared efforts and vision of the team, but also the best version of its interwoven mechanics possible.

In order to achieve this, the team approached the beginnings of development — the Alpha phase — as an opportunity to build and test the central mechanics for the game. Pulling from Design's original plans for the game, as well as all the thought and planning that the disciplines had put into refocusing their efforts at the end of pre-production, the conclusion was clear. After considering the identity of the game as a management and narrative title, the team identified two core features to correspond to these genres.

The focus of the Alpha development period was therefore twofold. The team and its subteams approached the new problems of first building a compelling core loop that would keep players engaged, and then populating the game with the characters that would bring its story to life.

4.2 - Establishing the Core Loop

The first of these focuses, the establishment of a compelling core loop, was especially important to developing a true video game. A video game's core loop is a chain of actions the player will repeatedly undergo throughout regular gameplay. Defining, fine-tuning, and perfecting this core loop is imperative to the success and healthy development of any game; it was also at the top of Design's responsibilities entering the Alpha stage.

4.2.1 - The Core Loop

The Design team's pre-production phase (Section 3.2.3) had already pinned down the broad structure of the game: identifying a desired combination of management tasks, character interaction, and exploration. While this balance would continue to evolve throughout development, one system in particular became the focus of the game after its Protofest debut: cooking.

The original card game prototype had its fair share of lessons, the most reassuring of them was that players were intrigued by its flavor system, seeing the mechanism as something unique. Although it had too many flavors and special cooking rules to manage in the context of a card game, the idea was interesting, and the team decided to use this design element to flesh out our core loop. Combining ingredients to produce an end result that balanced positive and negative flavors from a limited number of options paved the way to making cooking breakfast for birds not just a daily task that needed to be completed, but a puzzle that required deliberate thought and experimentation.

This core mechanic of cooking meals for the birds staying at the B&B became the primary focus of designers throughout the Alpha. The interaction may have worked in the card game, but making it into the mechanic at the heart of the game's core loop meant tweaking, cutting, and adapting parts of the system into one that would work in a long-form video game. Changing the number of flavors from the card game's nine down to four — and then back up to six — was a prime example of how work went into adapting the original format for a video game. Additionally, the customers and their preferences underwent their own batch of adjustments as the team analyzed their relationship with the cooking system.

In order to maintain the puzzle-like gameplay of the cooking system, the player needed to be challenged with cooking meals under shifting parameters while following a consistent ruleset. The consistency involved in the interaction became obvious: players would put ingredients into an appliance, such as a blender, stove, or oven, to combine their flavors into a final dish. The shifting parameter was still a question.

The team elected to keep customers' liked and disliked flavors as consistent information about the character rather than randomly generated daily preferences: this tradeoff meant that each bird became more of a character than a game mechanic, but the puzzle stayed the same so long as the same meal was being made for the same character. The solution — as well as the answer to our need for a shifting parameter — became clear. If the customer's flavor preferences aren't being changed, then change the customer.

A unique nature of a B&B is the constant circulation of guests and travelers passing through it. Any given bird could stay a day or two, giving the player the opportunity to familiarize themselves with their flavor palette and good combinations of ingredients for them. As part of the normal management style, that bird can then leave, opening a slot for a new bird, and with them a new flavor palette to work with. The criteria of the cooking puzzles being solved are subject to the birds they're being solved for, and gameplay begins to encourage a relationship to form between the player and the characters being served.

The core gameplay loop therefore positions players to tailor what they make each bird for breakfast during their limited time at the establishment. Accommodating the unique preferences of guests means posing the player with the puzzle of making the best food possible for them. That puzzle became the core mechanic of the game; beyond Alpha, the team would continue to expand the system with recipes and unique meal requests, working to make a guest's individual stay more varied. Yet at its heart, the gameplay loop became about catering to different guests, and solving the puzzles of preferences associated with them.

4.2.2 - Implementing the Game's First Features

Once Design finalized the core loop, Programming began implementing the game's functionality using plans established during pre-production to create a foundation for systems and features that could continually be expanded. Given that this was the first of the game's many features to be implemented, it also became the trial run for the protocols the Programming team would follow throughout development.

With three programmers, it was imperative that we establish protocols that aided communication and allowed us to maintain a quality codebase; to facilitate this, Programming scheduled weekly meetings. These meetings consisted of taking plans created by content-driven teams such as Design or Writing and converting them into manageable, bite-sized tasks. Known as ticket grooming, this standard practice is employed in software development industries and allowed Programming to plan out a task's implementation while integrating both inside and outside of their respective system. From there, Programming would assign each task to a programmer.

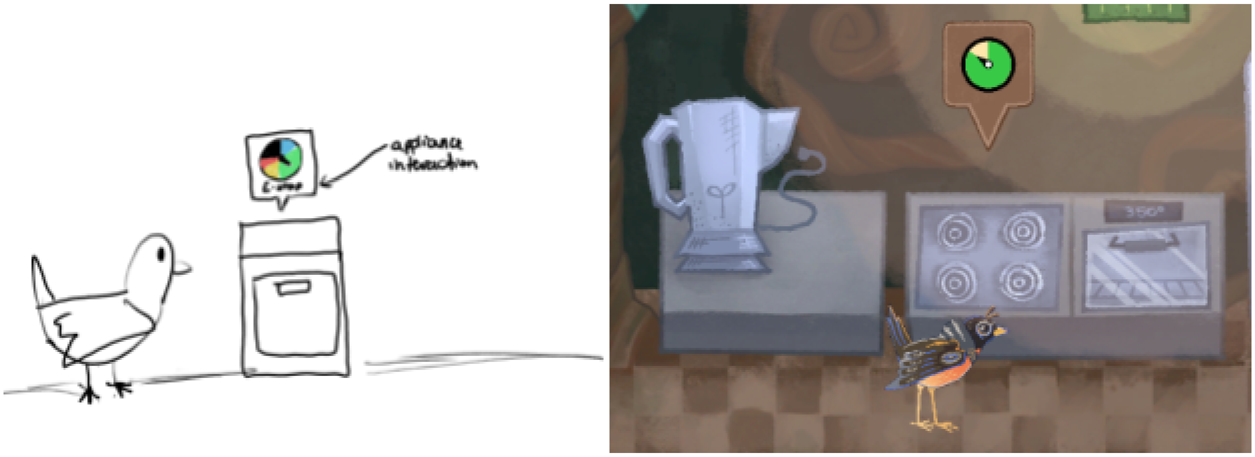


Figure 19: Design mockup for appliance cooking progress (left) and in-game screenshot of the timed cooking feature (right).

As part of this, Programming often designated individuals to specialize in specific systems, such as for cutscenes, the player character, and non-playable characters (NPCs). This person would be responsible for undertaking the majority of that system's tasks, and could be consulted if another programmer was making a change that would affect said system. This helped our efficiency and communications, but – if done too strictly – could lead to programmers being pigeonholed into working on a single system.

To counteract the idea that only one programmer could efficiently undertake and complete a task of a certain category, the ticket grooming process sought to ensure that all programmers would at least have a fundamental understanding of how certain features worked. This was incredibly helpful, as any programmer could be knowledgeable enough to work through issues when another programmer became stuck on a task. Additionally, this shared knowledge of tasks ensured each programmer could provide code reviews of one another's work, an essential practice employed by software developers to maintain a clean and ideally bug-free codebase.

Programming employed these code reviews as a standard part of their working protocol: once a programmer was finished with a task, they would then create a pull request (PR) on GitHub, and another programmer would be assigned to review that task. This was done to verify that the code worked well, was sufficiently documented, had no bugs, and met the task's requirements. As people reviewed others' tasks, they simultaneously broadened their understanding of other systems and contributed their own knowledge to tasks.

Code reviews were critical with limited production time; it couldn't be wasted writing incompatible code that wouldn't scale. We conducted our code reviews in this manner as Programming would already have a good understanding of how the system functioned, thanks in part to ticket grooming, and thus group code reviews would be an inefficient use of time. Yet, one of the drawbacks of our code review workflow was that a programmer in the middle of coding a feature may be pulled away, as code reviews were a higher priority. However, consistent and honest communication of everyone's task statuses lessened these negative effects, ensuring the team was able to employ industry-standard practices while implementing the foundational features on which the first stages of the game relied.

As part of building towards the core gameplay loop of cooking and serving food, the first features implemented by the team were an effective trial run for this workflow, as they required the implementation of player movement and interaction systems. Although simple, these features were a critical foundation for future systems and work. Because games are interactive experiences, this feature would significantly define how the player interacted with the game's world.

Based on Design's original plans for how the player character would be controlled, Programming began designing a character to fulfill these needs, using modularization to break out separate parts of the system such as player input, locomotion, physics collision, and interactors. Following the established ticket grooming and review process helped to further generalize the system, ensuring that its code was readable and well-designed. In this capacity, our workflow helped to ensure that the feature was easily scalable, allowing programmers to later use and extend this feature later on as part of the dialogue, cooking, and chore systems.

4.2.3 - Content Strategy Shifts

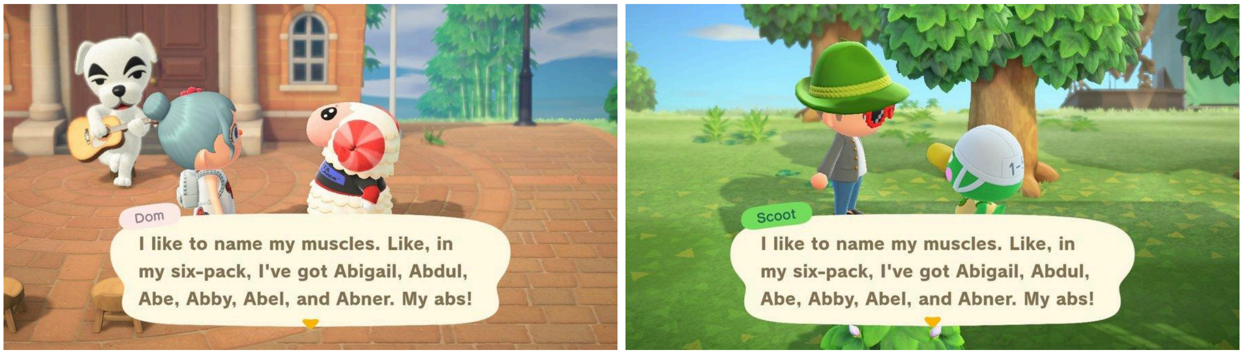
After the team's realignments and establishing our core loop at the beginning of Alpha, it became clear that a revisit to our pillars and experience goals was necessary to maintain a collective vision. *Bed and BEAKfast* would not be the first game about providing a service for customers. However, instead of player success being measured through monetary gain, progression in *Bed and BEAKfast* would be driven by how well the player caters to each individual customer; rather than min-maxing profits, the player should be motivated to cultivate deeper relationships by preparing food for their customers. It was at this time that the team decided on what came to be our most essential pillar: "Management simulator that prioritizes community over business" (Section 1.3).

With this greater emphasis on the community interactions through cooking meals, content-driven disciplines like Writing and Art expanded their workflows to focus on two categories: Birds and Food. We needed a wider cast of characters, enough to provide variation for the customers at the B&B, both in terms of visuals and dialogue. Similarly, to support the cooking mechanic central to the gameplay, there was a need for a large number of icons and descriptions for food items that were distinct and easy to produce. Based on these requirements as outlined in Design's core game loop, other disciplines began to make strategic decisions regarding how to best format the work that they were creating.

In terms of characters, the problem the Writing subteam came to face early on was the need to write large amounts of dialogue for different categorizations of characters. Design planned for a significant cast of rotating Travelers, birds staying at the player's B&B for a short period of time before leaving permanently, as well as a more regular set of Townies, the local characters who were to be a more permanent fixture in the town. In an attempt to mitigate the high demand for dialogue that a plan like this demanded, the writers looked to one of our game comparables in Nintendo's *Animal Crossing,*[47] as that game already employed a helpful approach to handling writing for numerous villager characters.

Animal Crossing's solution was to create archetypal personalities such as Jock, Snooty, and Peppy; in the game, each of these personalities have a custom set of dialogue responses which could be applied to many characters. Instead of creating enough dialogue to provide variety for each character, writers only needed to make dialogue to provide a similar amount of variety among each personality.

[47] *Animal Crossing: New Horizons*, (Nintendo, 2020), Switch.



Dom

I like to name my muscles. Like, in my six-pack, I've got Abigail, Abdul, Abe, Abby, Abel, and Abner. My abs!

Scoot

I like to name my muscles. Like, in my six-pack, I've got Abigail, Abdul, Abe, Abby, Abel, and Abner. My abs!

Figure 20: Two examples (left[48] and right[49]) of dialogue from the Jock personality in *Animal Crossing: New Horizons*, developed by Nintendo, Switch, 2020, screenshot.

Writing expanded on this original idea by allowing each traveler to have multiple personalities (such as grouchy, sleepy, goofy, ambitious, snooty, and friendly), and then enabling support for optional custom dialogue on top of this. This combination of personalities and unique dialogue was intended to obfuscate the simple rules used to characterize birds' dialogue, making it more difficult for players to pick out the repetition of content when it occurred. With this strategy, writers could easily add new dialogue for unique characters without significantly expanding their workload.

Conversely, Art was facing a growing pile of unique assets that needed to be created with each new bird. Writing may have developed a manageable solution to increasing scope while adding characters, but that only exacerbated the problem for the artists. With each new character to be added, not only were character designs and sprites required, but those sprites needed idle, walk, and flight animations, along with portraits for use in close-up dialogue boxes. The assets required for a single character were not insignificant, and the team recognized this. To counteract the bottleneck in the character creation pipeline, a few key decisions were made early on.

First and foremost, the decision to puppet animate all of the characters drastically reduced the required animation time. This animation style, involving the rigging and movement of individual components, saved a drastic amount of time over traditional hand-drawn animation (see Section 4.3.3 for more information).

[48] Fye (@FyeACNH), "Just Dom being Dom. Naming his abs. Lol. #AnimalCrossing #ACNH #NintendoSwitch" June 20, 2020, 4:24 AM, https://twitter.com/FyeACNH/status/1274256848883822592.

[49] Kit Ellis (@kitosan), "*Chef's kiss*" X, April 24, 2020, 10:45 AM, https://twitter.com/kitosan/status/1254058948019314695.

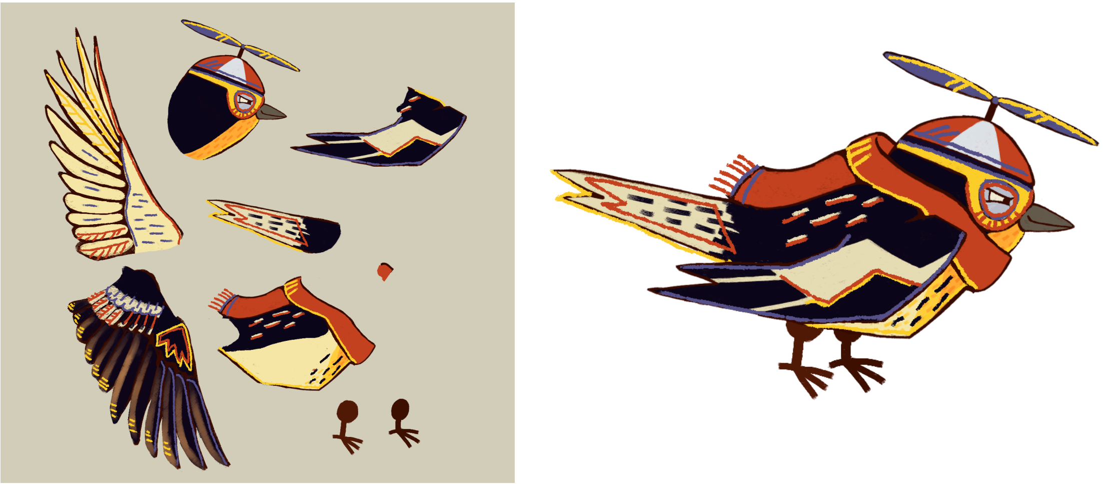


Figure 21: An example of the pieces of a puppet (left) and the composited result (right).

The fact that the sprites were created at a very large file size also meant that when enlarged, they could be repurposed as portraits for dialogue boxes without a significant difference in quality or style from the custom portraits created for main characters. While the portraits were not particularly hard to create, the team wanted to prioritize making emotive portraits for the main characters as needed for emotional cutscenes. The ability to use sprites as portraits meant that the artists could focus on supporting the game's story without sacrificing the amount of content needed to uphold the core loop.

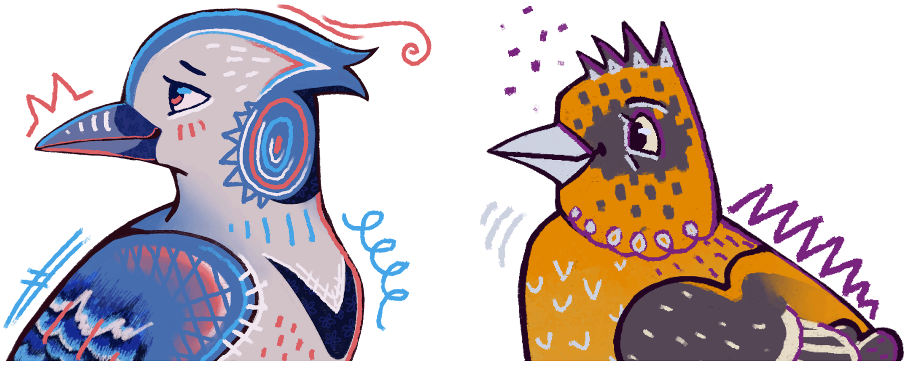


Figure 22: A custom portrait (left) compared to a zoomed-in sprite (right)

These strategies enabled the creation of character art and sprites, allowing for the Art subteam to keep up with Writing in this department. Yet ironically, the opposite story rang true when it came back to the vast library of foods needed for the game's cooking system.

Rather than struggle with the asset demand placed on them by Writing and Design, artists quickly realized that the overall art style of the game lent itself well to quick sketches that looked intentional and cohesive alongside the rest of the game's art. Icons created for the ingredients and meals in the game were therefore easier to create in bulk, meaning the artists' ability to create these images didn't bottleneck any plans when developing and pushing out this form of content for the game.

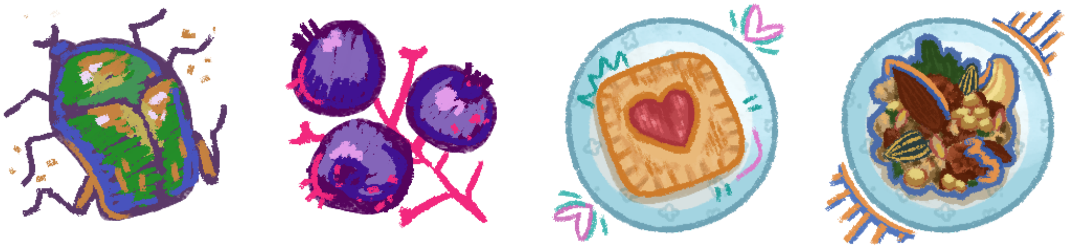


Figure 23: Example foods of beetles, pokeweed berries, pastries and granola (left to right).

However, Writing struggled with the nature of these types of asset requests, as the need to write and adjust meal descriptions based on the rapidly changing nature of foods and their flavors according to Design's repeated balancing posed a consistent issue that threatened to pull Writing's focus from other tasks. In fact, there were several places where holes in the development of content began to manifest — holes that needed to be filled with a combination of outsourcing and scope cutting.

Writing's challenges were twofold: first, the nature of the team composition itself. With the team being made up solely of artists and programmers who had other disciplines to worry about before their writing tasks, the team worried about being able to rise to the challenge without causing other parts of the game to suffer. For this reason, an Independent Study Project (ISP) student was brought onto the project as a dedicated writer for the remainder of the game's development. Not only did this provide an extra hand, but an outside perspective and an additional understanding of the writing process was added to the team, improving morale and productivity.

The second challenge, tying back into the bothersome problem that meal descriptions provided the writers, involved a separate ISP student to take on a very different role at the end of the project; this separate writer, who would be scheduled to join the project during the final, polish phase of development, would be the main writer responsible for finalized item and food descriptions. As such, the demand for this content was mitigated on the writers during the early phases of development, and the flavor text used to flesh out foods and their ingredients became irrelevant to the team while developing the core content and mechanics during the beginning of development.

For Art, issues arose in the animation process; although they were able to make sprites and portraits on pace with the team's demands, creating the many necessary animation states and transitions necessary for a bird just wasn't feasible. Although an ISP animator was scheduled to join the team for the term following Alpha, the large number of flight animations and their required transitions movements (landing, taking off, gliding) was still going to be unmanageable with an additional artist. As such, the team ultimately came to the conclusion that only allowing a few select characters to fly (the player and important NPCs) would be beneficial: the scope cut would decrease the animation work by more than half, cutting out the time spent making up and down wing sprites, and reducing the load on Programming to create pathfinding systems for bird movement around the tree. While the whole team loved the liveliness that a tree full of flying birds would bring, we knew that it was better to have a smaller, more polished game than a larger game that had been stretched too thin. In this regard, the plan shifted such that NPCs would remain for the most part stationary, idling in their given locations and pacing around to add the illusion of full movement.

The combination of onboarding ISPs to outsource components of content development and strategically placed scope cuts that benefitted multiple subteams became important strategies for planning and optimizing game content as part of Alpha development and beyond. Without these plans and the flexibility to make content creation as simple as could be, the team would no doubt have been hard pressed to develop and iterate upon their plans for future content.

4.2.4 - Planning Adjustments

Designing, implementing, and adapting our content around the game's core loop was an important correction, especially following the missteps made in the pre-production phase. More importantly, this new organization worked well, allowing all subteams to recenter themselves with a more explicitly defined goal. Naturally arising from this development however, was the question of what was next: how could the team best capture this momentum as plans began to expand beyond the core loop?

The original plan, as outlined by the producers during the pre-production phases of development, set aside specific milestones for project development to turn over between Alpha, Beta, and Gold periods, but like most of the work done by the subteams in that period, it was created in a vacuum without significant input from the other disciplines or any practical idea of how the team would operate during development. The new approach to development required a redefinition of this plan, one that both defined how the rest of the game would immediately expand out from the core loop, while also paving the road for long-term development: this is where the team began to use layers as a tool to define incremental development within the milestones.

Layer	Deliverable	Features
1	Basic Cooking	Player Character, Cooking, Flavors & Preferences
2	Expanded Cooking	Appliances, Dialogue, NPCs, Simple Foraging
3	Management Gameplay	Traversable B&B, Flight, Cleaning Chores, Timer Functionality, Daily Review, Notification System
4	Expanded Management Gameplay	Town Characters, Guest Book, Rudimentary Building, Cutscene Implementation, Save System

5	Simple Town	Farmer’s Market Town, Interactive Foraging, Recipe Book, Settings Menu
6	Full Town	Shops, Explorable Town, Quests

Table 1: A chart of the initial plan for layers and what features each included.

Primarily, these layers provided a hierarchical organization of features the team planned to add to the game. The first layer, accordingly named Layer 1, started from this simplest iteration of the core game loop, and introduced essential gameplay mechanics such as the player character and a menu to cook food. Layer 2 would go on to add customers, dialogue, and appliances, and so on. In this manner, subsequent layers continued to introduce more intricate and expansive elements, with the intention of incrementally building upon the variation, depth, and complexity befitting a full game. In doing so, layering provided content-oriented subteams such as Art and Writing with clear indicators as to where related content would be required from them, while also aiding Design and Programming in planning the immediate priorities and features to be fleshed out and completed.

In adopting the layer-based approach, the team shifted away from the previously defined roadmap production had created; instead, the hard milestones of Alpha, Beta, and Gold periods coalesced behind the new idea that the team could produce a cohesive and playable game as part of each layer. With this mentality, while Layer 3 might not have as many of the planned game systems as the team’s more optimistically planned Layers 9 and 10, the game would still serve as a viable product at the end of any Layer.

This approach better suited the team, fitting the needs of evolving design considerations and ensuring that each iteration contributed meaningfully to the game’s overarching vision and player experience. Though the traditional game development model’s clarity between Feature Complete, Content-Complete, and Polish phases was lost, the team felt that the continuing development model fit the needs of this project the best. The flexibility of being able to build and test the game in phases, and then respond to that feedback and reshape plans for the next layer, served as instrumental in the development of the game beyond this core loop.

4.3 - Bringing the Characters to Life

The second of our Alpha development goals, character creation to bring its story to life, was essential to the team’s early aspirations, especially considering the game’s story was intended to be a character-driven narrative. As such, establishing characters that felt both lively and believable in-game was an important task in the early development stages. Throughout the development of the Alpha, every subteam in the group would work in collaboration to make this happen, from the initial creation of interesting and story-ready characters, to the development of the colorful bird sprites and their animations in engine.

4.3.1 - Writing Characters

For each major character to be added to the game, the natural starting place of this long journey was with the Writing subteam. As part of realizing an overall narrative structure, it was important to create characters that felt meaningful and drove the story forward. For this reason, the three main characters were created alongside each other, whilst carefully considering and balancing their relationships.

First, there was the player character, Robin, and their childhood best friend, Casse. These two characters were designed to be foils, meant to draw attention to the player character's role and maturity. Despite their history of being so close, the two have opposite characterizations at the story's opening. Robin's arc starts with them being quite self-absorbed; through their interactions with others and as part of managing the B&B, their growth results in their learning how to care for their community. Inversely, Casse starts out as an entirely selfless character, putting far too much pressure on herself. Over the course of the story, she learns to rely on others in order to be able to find her own happiness.

The last of these main characters, Sterling the landlord, exists as a separate but equally important foil to Robin, demonstrating two contrasting sets of values and messaging: while Robin inherits their late grandfather's family business and legacy, Sterling inherits his father's money and capital, including the tree on which Robin's B&B resides. The different choices these two characters make with what they have been given contrast one another, with Robin ultimately using their position to contribute to the growth and betterment of their community, while Sterling's greed continues to harm the wider population.

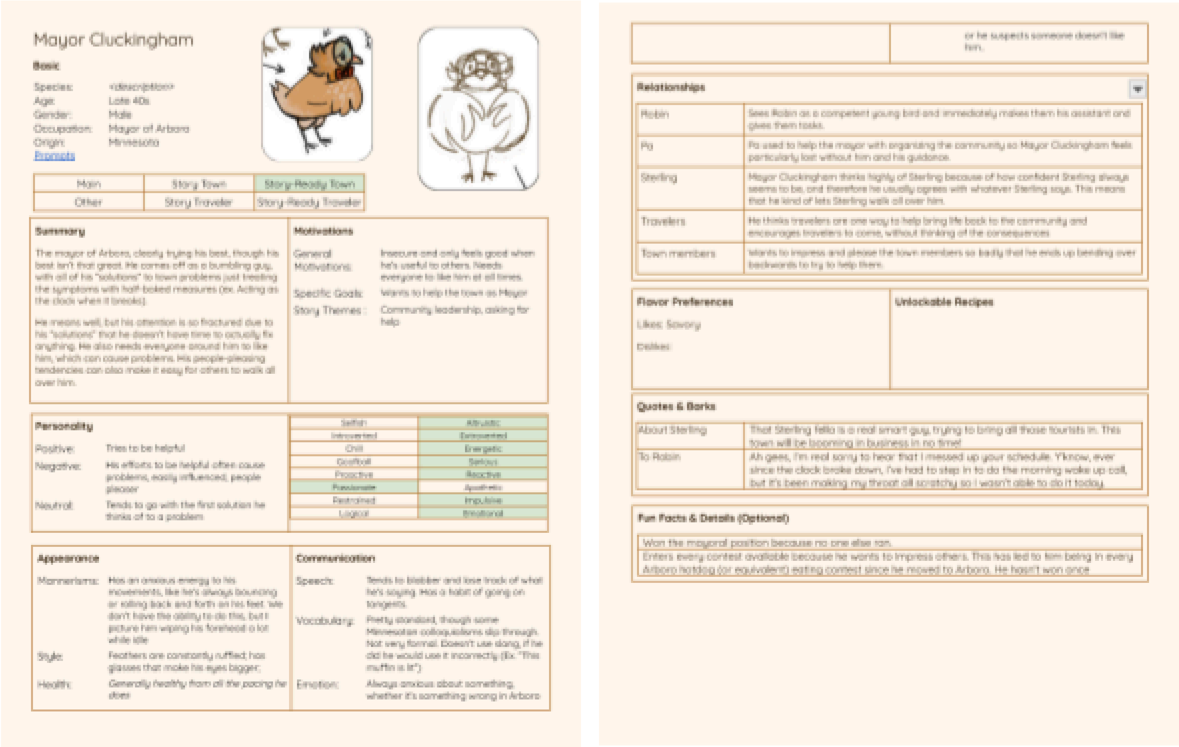
Once the main characters had been created, the Writing subteam moved on to populating the rest of the world. From Writing's initial plans during pre-production, the game's narrative was intended to be driven by many other smaller stories the player could experience by interacting with individual characters; the rough estimation the team had made involved five separate story-arcs from characters, each consisting of around three cutscenes (Section 3.2.2). Rapidly, it became evident that such a plan wouldn't be feasible, especially considering the replanning phase required of Writing in the wake of its pre-production.

Due to the time constraints and the extremely limited scope of a demo as compared to a full game, the writers realized there was little room for separate character arcs outside of the introduction, which focused primarily on the main trio. Regardless, Writing still felt it essential that a lively-feeling world was established; even if this demo was a vertical slice of gameplay, it was critical to create memorable characters who felt they may have complete story arcs and growth in a full game.

To facilitate this, the process of populating the world with story-ready characters began. Story-ready was a term used to indicate characters to whom a narrative arc might apply. As such, they were written based on some sort of instability or paradox that could believably lead to a deeper, more complex arc exploring that aspect of their character. Once a few of these NPCs were created, the team was able to pick which characters and future arcs would be the most effective to include as part of this vertical slice.

An important distinction to make when discussing the creation of these characters is the aforementioned split between Townies and Travelers. As previously stated, Travelers refer to the birds passing by on their migrations, serving as customers who stay at the B&B. They visit once per season, stay for a duration between a few days and a week, and return the following year; in the demo, this would only manifest with these characters appearing for a short period of time before being permanently absent. Given the short form nature of their appearance, Writing elected for most of these characters to rely primarily on the personality pools and shared dialogue that was discussed in Section 4.2.3.

Townies, on the other hand, needed to rely on more personalized dialogue, due in part to their role as year-round residents of the community, living in the forest surrounding the B&B. Given that the player would be interacting with these Townies characters more consistently across the game, it was important that they felt more distinct to the player, with many of them being characterized by their relationship to the player.



Mayor Cluckingham

Basic

Species:	<description>
Age:	Late 40s
Gender:	Male
Occupation:	Mayor of Arboro
Origin:	Minnesota

Prompts

Main	Story Town	[✓ Story-Ready Town]
Other	Story Traveler	Story-Ready Traveler

Summary

The mayor of Arboro, clearly trying his best, though his best isn't that great. He comes off as a bumbling guy, with all of his "solutions" to town problems just treating the symptoms with half-baked measures (ex. Acting as the clock when it breaks).

He means well, but his attention is so fractured due to his "solutions" that he doesn't have time to actually fix anything. He also needs everyone around him to like him, which can cause problems. His people-pleasing tendencies can also make it easy for others to walk all over him.

Motivations

General Motivations:	Insecure and only feels good when he's useful to others. Needs everyone to like him at all times.
Specific Goals:	Wants to help the town as Mayor
Story Themes :	Community leadership, asking for help

Personality

Positive:	Tries to be helpful
Negative:	His efforts to be helpful often cause problems, easily influenced, people pleaser
Neutral:	Tends to go with the first solution he thinks of to a problem

Selfish	[✓ Altruistic]
Introverted	[✓ Extroverted]
Chill	[✓ Energetic]
Goofball	[✓ Serious]
Proactive	[✓ Reactive]
[✓ Passionate]	Apathetic
Restrained	[✓ Impulsive]
Logical	[✓ Emotional]

Appearance

Mannerisms:	Has an anxious energy to his movements, like he's always bouncing or rolling back and forth on his feet. We don't have the ability to do this, but I picture him wiping his forehead a lot while idle
Style:	Feathers are constantly ruffled; has glasses that make his eyes bigger.
Health:	*Generally healthy from all the pacing he does*

Communication

Speech:	Tends to blabber and lose track of what he's saying. Has a habit of going on tangents.
Vocabulary:	Pretty standard, though some Minnesotan colloquialisms slip through. Not very formal. Doesn't use slang, if he did he would use it incorrectly (Ex. "This muffin is lit")
Emotion:	Always anxious about something, whether it's something wrong in Arboro or he suspects someone doesn't like him.

Relationships

Robin	Sees Robin as a competent young bird and immediately makes them his assistant and gives them tasks.
Pa	Pa used to help the mayor with organizing the community so Mayor Cluckingham feels particularly lost without him and his guidance.
Sterling	Mayor Cluckingham thinks highly of Sterling because of how confident Sterling always seems to be, and therefore he usually agrees with whatever Sterling says. This means that he kind of lets Sterling walk all over him.
Travelers	He thinks travelers are one way to help bring life back to the community and encourages travelers to come, without thinking of the consequences
Town members	Wants to impress and please the town members so badly that he ends up bending over backwards to try to help them.

Flavor Preferences

Likes: Savory

Dislikes:

Unlockable Recipes

Quotes & Barks

About Sterling	That Sterling fella is a real smart guy, trying to bring all those tourists in. This town will be booming in business in no time!
To Robin	Ah gees, I'm real sorry to hear that I messed up your schedule. Y'know, ever since the clock broke down, I've had to step in to do the morning wake up call, but it's been making my throat all scratchy so I wasn't able to do it today.

Fun Facts & Details (Optional)

Won the mayoral position because no one else ran.
Enters every contest available because he wants to impress others. This has led to him being in every Arboro hotdog (or equivalent) eating contest since he moved to Arboro. He hasn't won once

Figure 21: Mayor Cluckingham's character sheet. See Appendix E for enlarged version.

A mix of story ready Travelers and Townies was intended to give the players the impression of a large possibility space for stories and future narratives that could be explored beyond the demo. The usage of Townies, both story-ready and not, provided a continuous source of familiar faces and dialogue, filling out the world and ensuring that it never felt empty; conversely, the circulating nature of Travelers would provide a continuous rotation of characters wholly new to the player, ensuring that fresh faces — and stories — could always feel just around the corner.

4.3.2 - Designing Effective Visuals

After characters were created and any potential story-arcs accounted for, their information was passed to the Art subteam to turn them into sprites and portraits. Iteration was an important part of this aspect of development, especially for the main characters; if the player was going to see a character over and over again, or constantly throughout gameplay, we needed to ensure that the character was memorable, distinct, and of course, pleasant to the eye. As part of this, the team established a system for feedback which enabled many iterations of a character within sprints.

Within these iterations, the choice of different bird species were played with a lot: while some of the characters passed from Writing had a species already chosen for narrative or thematic reasons, others required Art to make decisions that incorporated the research done during the pre-production phase of the project (Section 2.1). Naturally, different birds had different characteristics, behaviors, and shape language, all of which contribute to a character's presentation to the audience or player. Casse, for example, was tested as both a Blue Jay and a Dark-eyed Junco, as the Art subteam was unsure of which parts of her character to emphasize: appearing put together, independent, and seemingly aloof as the former, or meek, sedentary, and nurturing as the latter.

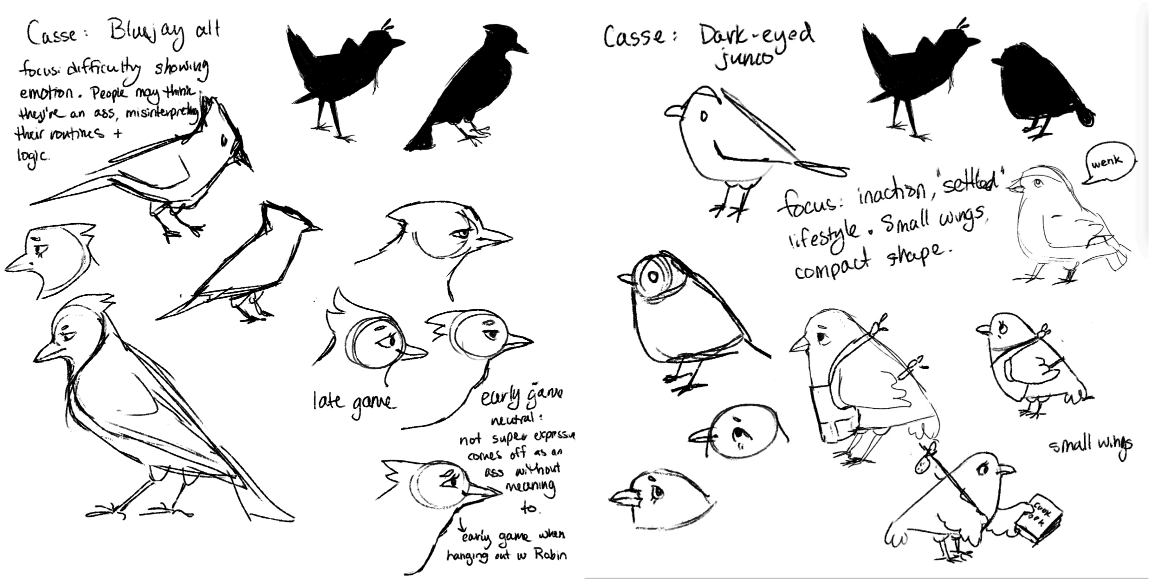

Casse: Bluejay alt
focus: difficulty showing emotion. People may think they're an ass, misinterpreting their routines + logic.
late game
early game
neutral: not super expressive comes off as an ass without meaning to.
early game when hanging out w Robin
Casse: Dark-eyed junco
focus: inaction, "settled" lifestyle. Small wings, compact shape.
wenk
small wings
Cook book

Figure 25: Initial designs of Casse as either a Blue Jay (left) or a Dark-eyed Junco (right).

Characters fitting in with their companions and foils was also important; as such, main characters were designed in tandem to make sure their color and shape language complemented one another. An example of this includes how Casse and Robin shared a few colors across their designs to emphasize their closeness and allude to how they grew up like siblings despite being from different families. Similarly, the team ensured that Casse's silhouette is made of long, smooth contour lines, whereas Robin is made of shorter, more forward-moving and energetic lines. Their contrasting personalities needed to shine through for these two as foils, while still paying homage to their similarities and origin.

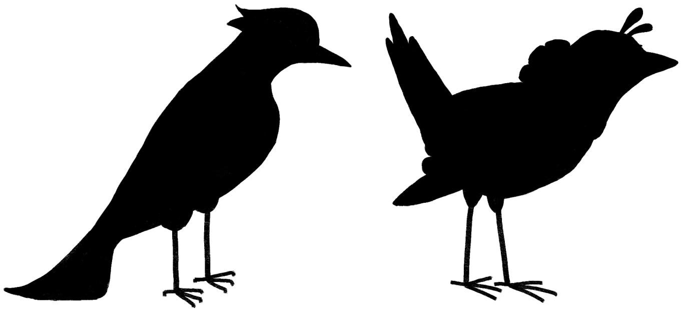


Figure 26: Silhouettes of Casse (left) and Robin (right).

When working on other characters as directed by the Writing subteam, the artists continued to employ these techniques, and this became the most common form of the pipeline for story-ready and Townie characters. Yet for Travelers, many of whom were open-ended as to which personalities they were to depict, this direction reversed; here, artists were able to take the creative reins and pull from common birds migrating through New England in our research. Choosing from these birds allowed the artists to experiment with different shapes and colors to create characters naturally fitting to one or two of the writer's defined personality traits.

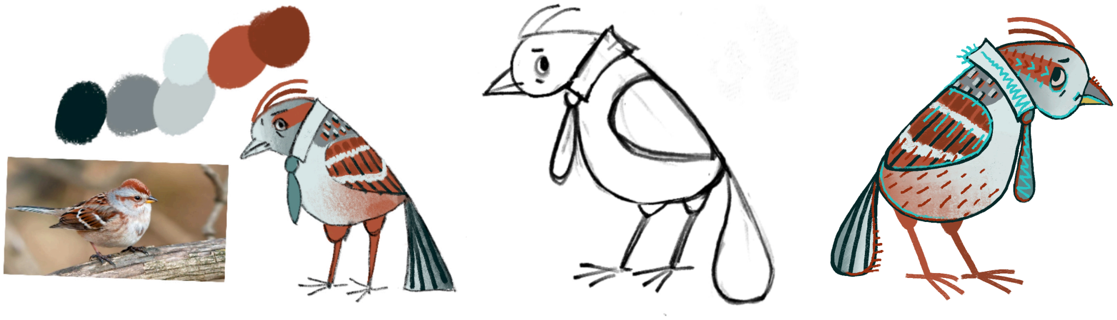
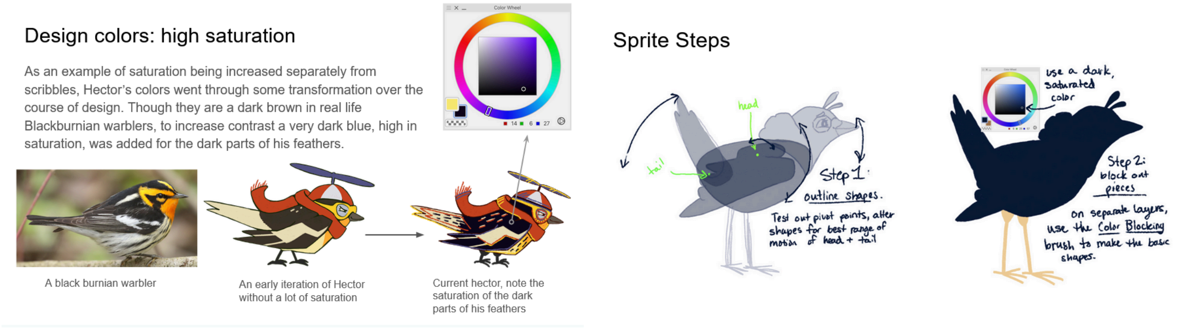


Figure 27: The progression of Reginald the American Tree Sparrow (left to right) was made using many tear-drop shapes that resulted in a weighed down, droopy appearance.

To effectively communicate to one another and potential future project contributors the process of making these characters, a guide was created detailing each of the steps, from design to puppet creation to finishing scribbly touches. Steps were listed visually and concisely, and run by one another to ensure their effectiveness. Later on in the process, all of the artists were able to pitch into creating characters when the need called for more hands.



Design colors: high saturation

As an example of saturation being increased separately from scribbles, Hector's colors went through some transformation over the course of design. Though they are a dark brown in real life Blackburnian warblers, to increase contrast a very dark blue, high in saturation, was added for the dark parts of his feathers.

A black burnian warbler
An early iteration of Hector without a lot of saturation
Current hector, note the saturation of the dark parts of his feathers
Sprite Steps
head
tail
Step 1: outline shapes.
Test out pivot points, alter shapes for best range of motion of head + tail
use a dark, saturated color
Step 2: block out pieces
on separate layers, use the Color Blocking brush to make the basic shapes.

Figure 28: Excerpts from the character creation guide.

These pipelines for creating effective designs that communicated personality, contrast, and the narrative elements of characters were an essential part of bringing characters to life and making the world feel more lived in and full. Without these protocols, the collaboration between Writing and Art regarding one of the most important aspects of a narrative game — its characters — couldn't have occurred, which is what made it all the more important for the team to establish these protocols and more early on as part of the Alpha development period.

4.3.3 - Animating the Characters

Once the designs were created, the next step of bringing characters to life was animating them by employing the puppet-style animations chosen by Art as part of simplifying their development overhead (Section 4.2.3). The task of balancing the many animations for a character, as well as their implementations in engine as part of the player and character controllers, was an important collaboration between the Art and Programming subteams.

The puppet method of animation was chosen after significant research, testing, and deliberation; while the Art subteam was most familiar with hand-drawn animation, the sheer number of characters — let alone their intricate designs — made this option impossible with the limited time, resources, and artists available. As such, we decided on a paper doll-esque, puppet approach to character animations done in Adobe After Effects (AE). This would require significantly less time to produce, allowing us to ensure that every character could be animated; it would also match the stylized aesthetic of the game. We wanted to maintain a bit of hand-drawn animation, however, and decided the best place for this would be on the character's legs – initial rigging tests proved too time consuming for a team wholly unfamiliar with this technique and did not produce the desired results. As such, legs were frame-animated in Clip Studio Paint (CSP), behind the frames of the previously exported AE animation.

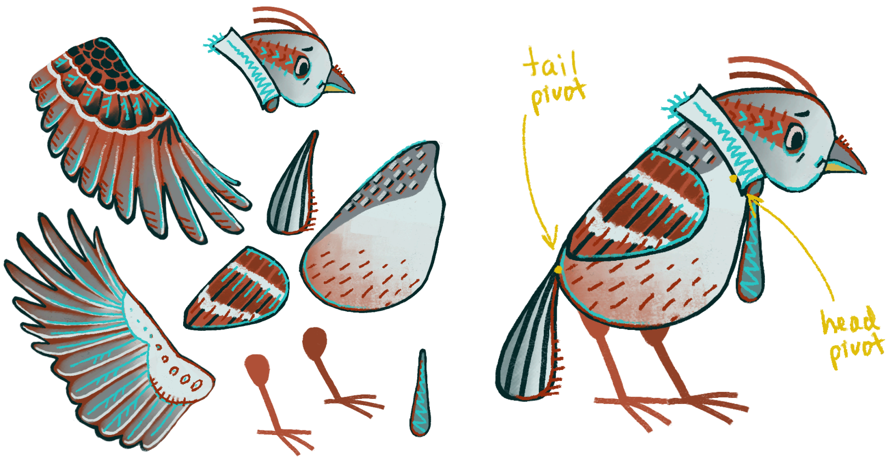
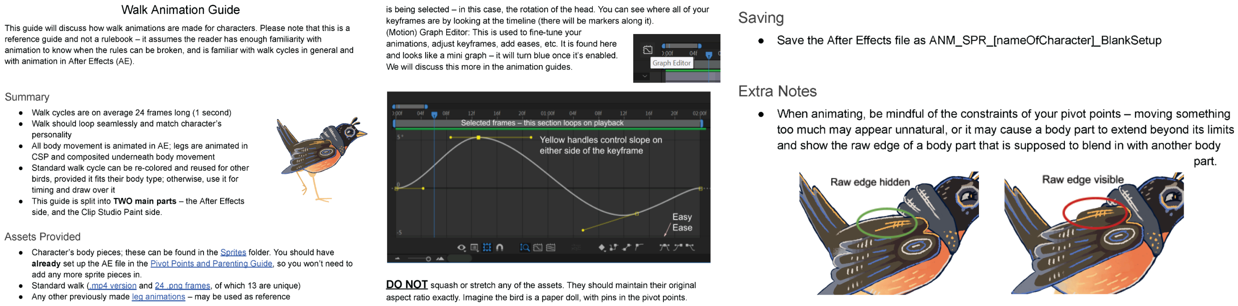


Figure 30: Sprite pieces of Reginald (left) with pre-defined pivot points marked (right).

All animations started with creating a base file for the character in AE; this involved setting up the sprite pieces with the correct pivot points and layer order, as instructed by the sprite's creator. The animator would then begin parenting pieces together to create a hierarchy, with the body piece acting as the parent to all other parts. This ensured that when the body was moved, the extremities would remain attached to the body in the same relative position. Pivots were then tested out to ensure maximum range of natural motion without compromising the sprite's appearance. This base file acted as a template that all the character's animations could be created from.

While we cut down the workload by choosing puppet animation over hand-drawn animation for the body, animating the legs traditionally was still quite time consuming. In order to better standardize and document this process, as well as to allow for possible future members to continue work on the game, multiple guides were created detailing how animations should be created for NPCs.



Walk Animation Guide

This guide will discuss how walk animations are made for characters. Please note that this is a reference guide and not a rulebook – it assumes the reader has enough familiarity with animation to know when the rules can be broken, and is familiar with walk cycles in general and with animation in After Effects (AE).

Summary

- Walk cycles are on average 24 frames long (1 second)
- Walk should loop seamlessly and match character's personality
- All body movement is animated in AE; legs are animated in CSP and composited underneath body movement
- Standard walk cycle can be re-colored and reused for other birds, provided it fits their body type; otherwise, use it for timing and draw over it
- This guide is split into **TWO main parts** – the After Effects side, and the Clip Studio Paint side.

Assets Provided

- Character's body pieces; these can be found in the Sprites folder. You should have **already** set up the AE file in the Pivot Points and Parenting Guide, so you won't need to add any more sprite pieces in.
- Standard walk (.mp4 version and 24 .png frames, of which 13 are unique)
- Any other previously made leg animations – may be used as reference

is being selected – in this case, the rotation of the head. You can see where all of your keyframes are by looking at the timeline (there will be markers along it).
(Motion) Graph Editor: This is used to fine-tune your animations, adjust keyframes, add eases, etc. It is found here and looks like a mini graph – it will turn blue once it's enabled. We will discuss this more in the animation guides.

Graph Editor
0:00f
04f
08f
12f
16f
20f
01:00f
04f
08f
12f
16f
20f
02:00f
Selected frames – this section loops on playback
5°
0
-5
Yellow handles control slope on either side of the keyframe
Easy Ease

DO NOT squash or stretch any of the assets. They should maintain their original aspect ratio exactly. Imagine the bird is a paper doll, with pins in the pivot points.

Saving

- Save the After Effects file as ANM_SPR_[nameOfCharacter]_BlankSetup

Extra Notes

- When animating, be mindful of the constraints of your pivot points – moving something too much may appear unnatural, or it may cause a body part to extend beyond its limits and show the raw edge of a body part that is supposed to blend in with another body part.

Figure 29: Excerpts from select guides – Walk Animation (left) and Pivot Points and Parenting (center and right). See Appendix G for enlarged version.

For animations like walk cycles, we set up a system of standardizing character walks based on our own template: all of Robin's animations were perfected as much as possible and then converted into standard cycle templates to be used as the baseline for all other birds. Robin's leg frames were separated and saved, and more guides were created instructing artists on how to create new walk cycles based on these. At its core, the process involved animating a bird to the timing of Robin's legs in AE, bringing that animation into CSP, and modifying said legs to match the bird by recoloring, resizing, and/or drawing on top of them.

As Robin is the player character, they have significantly more animations than any other character in the game. Animators worked closely with programmers to make all motions look as seamless as possible, while also allowing for quick response time to player input. This work attempted to maintain the player's immersion in the created world — too short and a transition looks choppy, but too long and it delays player movement. If the character's motion lags behind the player's input, even the most beautiful of animated transitions can frustrate the player and will not be very helpful as game art. Notable was the collaboration between artists and programmers to create flight, ensuring animations were properly made and imported to balance beauty and usability.

As each of those animations was created using individual frames, the in-engine animations were created by simply using the Animation tool in Unity to play all of the frames at 24 frames per second and setting the animation to loop when appropriate. From there, each animation was either input into an animation controller or an animation override controller. Robin, due to their complexity, uses their own controller, while most NPCs share a generic controller and use overrides to change which animation is played.

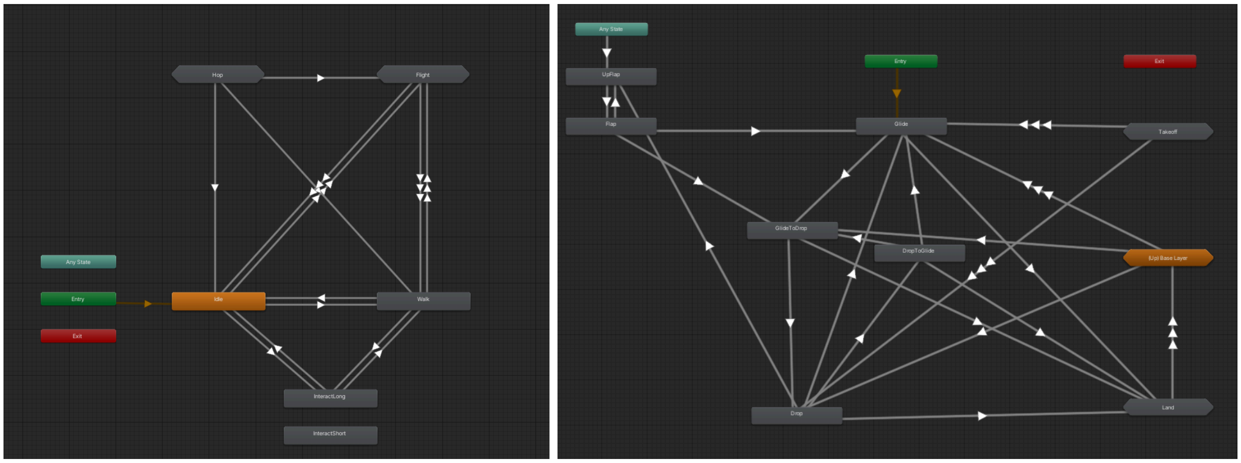


Figure 31: The overall animation state controller (left) and the flight subsection of the controller (right).

For Robin, the animation controller is relatively complicated, taking care of many different gameplay state possibilities. Their animations are sorted into five categories, which transition into each other and contain their own states and transitions within. Those categories are: idle, walk, hop, flight, and interact. Idle, walk, and interact are simply the corresponding animations, which can interrupt each other as the player walks and stands on the ground. It can then transition into hopping, which contains a sequence of three states that compose a flightless jump that lands back on the ground, initiated by pressing the spacebar. This hop can be interrupted by the player initiating a wing flap, which transitions into flight. Flight contains many sub-states, starting with a transition into the wing-out state. From there, the player can swap between flapping (by pressing the spacebar again), dropping (by pressing the down arrow key or 'S' key), or gliding (by pressing neither). Each of these three states has a small transitional animation to the other two, making the feeling of flying much more fluid and natural. This tri-state can be exited by landing, which plays a different landing animation depending on whether the player was gliding or dropping. This complex system comes together to give Robin a diverse set of movement animations that truly bring them to life.

Non-playable characters (NPCs) have far simpler animation systems. They largely only have idle and walking animations, which are used while they wander around the B&B. These are swapped out by animation override controllers that contain animations specific to each bird. In addition to this system, a small number of important NPCs (Casse and Ruth) have either flying or alternative animations, which are used during cutscenes and particular story beats.

4.3.4 - Introducing the Characters to a Wider Audience

Amongst all the subteams, Marketing was provided a unique opportunity from this focus on character design and creation during the early phases of development. Whereas they had previously struggled with the game's branding — first awaiting further instruction from the rest of the team, before starting with bird-centric social media content, and then struggling to move away from that niche — this aspect of Alpha development provided ample material for the team.

Naturally, many of the posts during this period still heavily revolved around birds due to the game's subject matter, yet the crucial difference was that the birds being discussed were either current or prospective characters. In this way, even posts not directly showcasing gameplay — which was still being developed and often not in a state to be displayed — could still tie back into building excitement about the game.

This shift was helped immensely with the guidance of a WPI alumnus and friend of the team who had worked much more extensively with game marketing than the team itself. This alumnus taught us about several essential components of making social media work for a game, such as how to create a hook, the fact that retention rate was based significantly on the first frame and five seconds of a video, and how to reuse gameplay footage to get consistent views. From their advice, we created a few new series of posts: Character Introductions, Day in the Life videos about the team behind *Operation Breadcrumbs*, and general game development videos about the humorous aspects of our team dynamics and work processes.

Character Introductions were posted each sprint and comprised a mini character sheet posted to X (formerly Twitter) and a short video about the development and personality of the character to TikTok and Instagram Reels. By allowing followers to meet the characters and take a peek behind the curtain at the work which went into the game's creation, we aimed to give them a chance to become familiar with and be intrigued by the cast, thereby increasing their emotional investment in tracking the game's progress.

Day in the Life videos were longer, more involved videos assigned to a different member of the team every few weeks; they offered viewers a look into the typical (or not so typical) day in the life of one of the game's developers. This allowed the audience to peek behind the curtain into our lives as students and people — not just developers — in an attempt to humanize the makers behind the game.

The last category of posts included many one-off videos as the team continually adjusted our approach to video-making to keep up with the latest trends, with most of these types of videos being directed towards lighthearted and silly jokes about game development that were less structured and thus easier to frame in the context of current internet jokes. This type of content was hard to predict and account for, but oftentimes became the most inspired and exciting posts for the team; algorithms such as TikTok's seemed wildly inconsistent and tough to manage, but the team overall found viewers more consistently were drawn to these funny happenings from developers rather than learning about gameplay.

By combining these marketing strategies, the team was able to more effectively grow its audience. Before, videos on TikTok averaged around 100-400 views and 10-30 likes; after pivoting towards more game development centered content, we were able to reach upwards of 3700 views and 400 likes. In less than two weeks after the launch of our Steam page, we were able to get over 70 wishlists — at the time of writing, we now stand at 151!

4.4 - Alphafest

In the wake of Pre-Production and its subsequent course correction, this phase of development was an encouraging time for the team; as we ironed out the core mechanics of gameplay and determined how to make the most out of our characters, we found ourselves making progress that quickly outpaced those doubts lingering from the initial stages. Those sentiments would be put to the test with the first instance of the game being demoed in a live setting, necessitating the incorporation of these core features into a playable build, which fittingly came to a head as part of WPI IMGD's annual playtesting event, Alphafest.

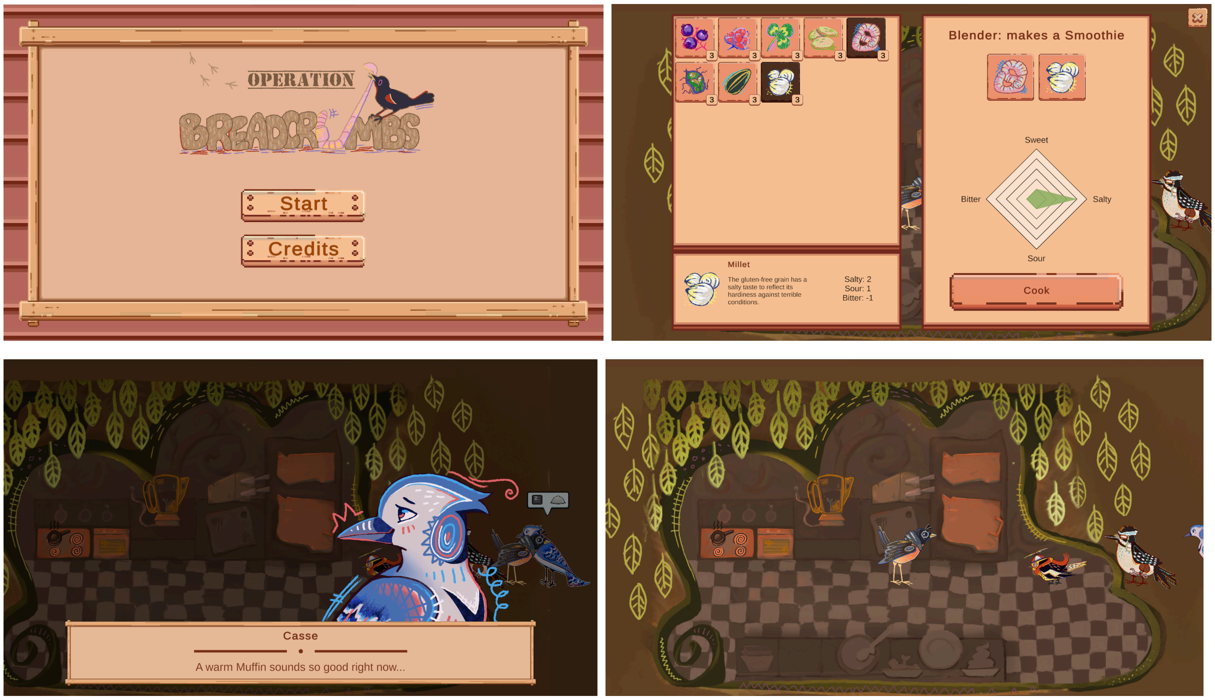
Figure 32: Alphafest playtesters at the team's booth (left and right) and the team minus one member (center).

Alphafest is a community gathering of IMGD students and faculty that showcases interactive media being developed. Taking place in the latter weeks of B-Term, it serves as the first significant milestone for MQP teams, who are required to demo their projects at the event. As such, it is traditionally recognized as the first time project groups show their work off to a larger audience, gathering feedback, information, and data from a large crowd of playtesters.

Traditionally, the term Alpha in the games industry denotes the stage in which all key functionality and mechanics of the game are implemented: a feature-complete version of the game ready to test all aspects and how they interact.[50] Our original production optimistically placed this deadline shortly after Alphafest, meaning that programmers and designers would be completely focusing on planning and implementing all core features of the game within one term; needless to say, this did not occur.

Instead, as part of the pivoting from these traditional labels to the layer-based design approach, our alleged Alpha build was instead a more evenly distributed showcase of features and contents for the first two layers of the plan (Section 4.2.4). This version of the game included a cooking system that made use of menus to cook different foods based on the appliance the player interacted with. The ingredients the player chose to cook with would affect the flavors in the resulting dish. The player could then use another menu to serve the customer their food.

[50] Jennifer Javornik, "Alpha, Beta, Gold: A Commitment to High-Quality Game Development," web log, *Filament Games* (blog), May 10, 2017, https://www.filamentgames.com/blog/alpha-beta-gold-commitment-high-quality-game-development/.



Millet

The gluten-free grain has a salty taste to reflect its hardiness against terrible conditions.

Salty: 2
Sour: 1
Bitter: -1

Casse

A warm Muffin sounds so good right now...

Figure 33: Screenshots from the Alphafest Build of the game. Includes the placeholder title screen (top left), the cooking menu (top right), basic NPC dialogue (bottom left), and the first instance of the game world (bottom right). Placeholder UI art from the *Complete UI Essential Pack.*[51]

Alphafest, and with it the culmination of our work into a playable game experience, enabled the team to observe significant progress from their missteps as part of Protofest. Overall, the team had a much more unified presence and goals: while this can partially be attributed to the single product now being worked on by all subteams, the value of our new communication protocols, team distribution, and directed focus of work cannot be understated.

The team used Alphafest to gauge the readability of gameplay systems and mechanics from this early stage: Did players understand what they were doing? Was the flavor system, the crux of cooking, more intuitive and simpler than in our Protofest card game? Were the core mechanics of the game engaging? Although the three-hour event provided a relatively small sample size, the feedback we saw to these questions was reassuring and served to validate the restructuring the team had made in past weeks.

[51] Crusenho, "Complete UI Essential Pack," asset pack, *Itch.Io* (Itch), accessed April 2, 2024, https://crusenho.itch.io/complete-ui-essential-pack.

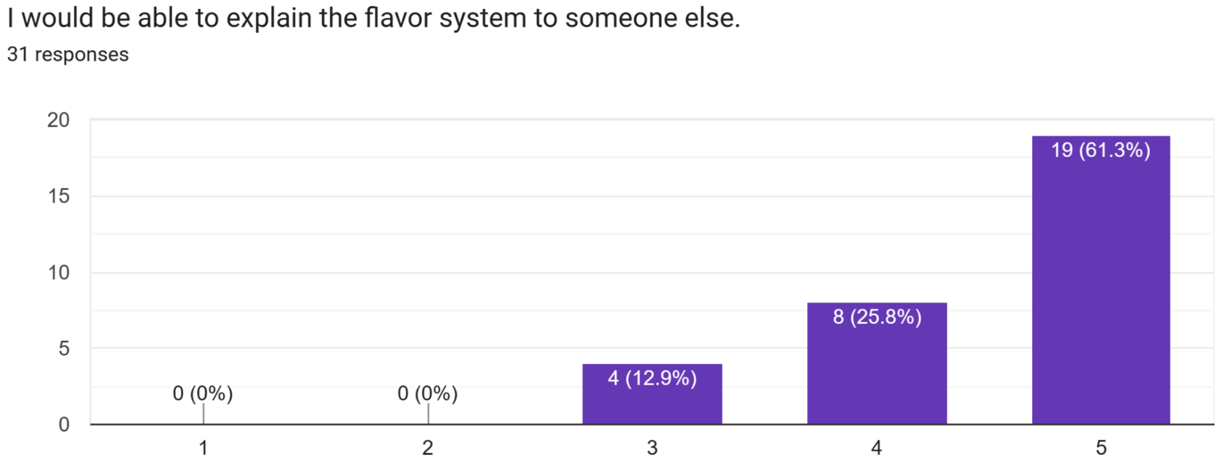


Figure 34: One of the quantitative responses gathered from the Alphafest Survey. See Appendix A: Alphafest for more survey questions and responses.

To gauge an answer to these questions, a survey was employed to gather quantitative data that investigated players' ability to understand the game mechanics they had interacted with. By pairing this with direct observation of the playtesters and their gameplay, the team noted a consistent comprehension of the central cooking mechanics of the game. Even in short gameplay sessions, players readily understood flavors and their visualization via the radar chart. The ease with which players grasped this system — one which was more difficult to understand in its original card format — gave us the confidence to move forward with some of our future plans to expand the cooking system: in addition to increasing the number of available flavors from four to six, cooking would eventually evolve to include additional nuance involving unique dishes and food groups made from cooking specific combinations of ingredients.

The players' reaction to the characters themselves, although they were limited in number and with barely any dialogue, was also overwhelmingly positive. Many players expressed their desire for more, a response that reinforced many of the decisions we had made as part of streamlining the character creation process.

These responses were a relief, serving as important reassurances that the team's primary focuses in the earliest phases of development hadn't been led astray. Yet there was one part of this early build which players responded to in a manner the team hadn't anticipated: the player experience.

Many players saw this early build as less of a management game and more as a puzzle game. This wasn't wholly surprising, given that we had focused on the cooking system and its flavor combinations as the main game mechanics. Given that it was the only system to interact with, it was natural for engaged players to spend significant time on the menus, pouring over the possible options to find the perfect permutation of flavors. However, the Alphafest build's heavy reliance on these menus was not something the team had planned to incorporate in the long term — rather, they were born out of necessity, as a simpler alternative to more complicated plans for interaction beyond these first two layers.

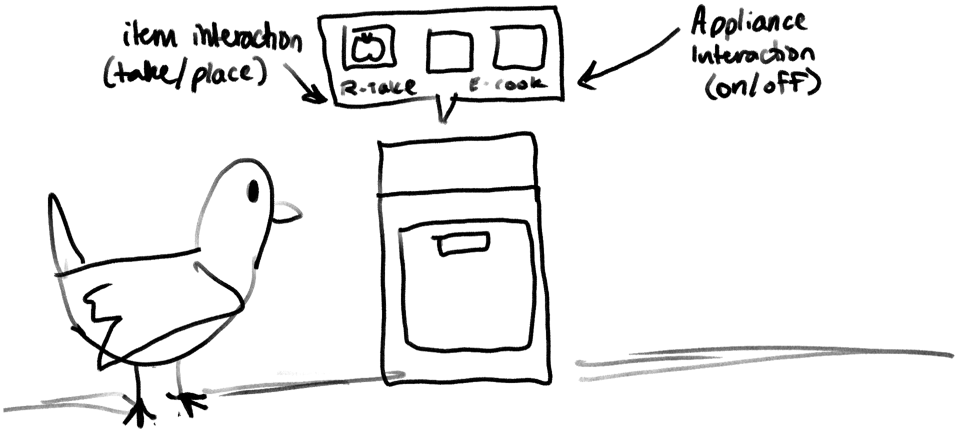


Figure 35: A mockup of the held-item interaction mechanism envisioned for the game.

Eventually, the team wanted to shift from using menus for cooking and serving foods to a *PlateUp!*[52] inspired control scheme of physically retrieving, moving, and placing ingredients into appliances. The intention was that this form of interaction would lend itself better to a management game that required the player to balance cooking multiple dishes alongside other tasks, such as cleaning rooms and tending to customers. The usage of menus with placeholder art as part of this build was meant to supplement that mechanic for the earliest stages of development, allowing playtesters of the Alphafest demo to focus on the cooking system itself and how ingredients could manipulate flavors.

Instead, we found these players doing mental math and calculations in the cooking menu, hand-picking with excruciating detail all the ingredients going into a dish, and treating the act of combining ingredients like a puzzle to be solved, in a level of detail and scrutiny we hadn't expected. In the wake of Alphafest, we had a gameplay system and an interaction mechanism that was working. Therefore, the designers had a series of decisions they had to face: after some deliberation and analysis, as well as a reexamination of the game's pillars and what was important to the team's members, the choice to pivot away from our initial plans and focus on cooking and serving through menus was obvious.

In this manner, the Alphafest demo showcased the strengths at the heart of the layer-based approach. We had originally viewed this as a weakness of the design methodology:

[52] *PlateUp!*, (Yogscast Games, 2022), PC, Switch, PlayStation 4 or later, Xbox One or later.

as part of creating fully playable games for each layer, it required resources and time to be spent creating temporary solutions to fill gaps left in the incremental design, such as these menus and providing players with mechanisms for interacting with cooking. However, by wading into these partial solutions and testing them, the process opens itself up to new ideas and approaches our team might have never considered otherwise.

Following Alphafest, we pivoted several of the upcoming layers, refocusing our priorities as we began looking beyond the core loop into a more featured game. Some of these decisions were big, like changing entire interaction mechanisms and the nature of the player experience. But most of them were small, minor tweaks intended to push us forward and empower the layer-based development approach as we made our way into full-fledged development.

5 - Beta

This next chapter concentrates on the process of iterating upon the game throughout playtesting and feedback. Through this phase, the team worked to expand the game's world and gameplay, including customer variants, the completed tree, and the full cooking system.

5.1 - Introduction

With the game's central mechanics well established — as well as proven and tested by their positive reception at Alphafest — development continued onward into a Beta period, where greater attention was spent on expanding the game's core features and creating content. This step would enable the team to host playtest sessions aimed at collecting feedback during this intermediate stage — feedback which would be used to iterate upon the game's systems and content, ensuring that future players of the final release might have the best experience possible. As such, it was imperative that this stage be closely representative of the final release, meaning that the Beta period needed to allow players to interact with the broader world and finalized gameplay systems.

Just as in the Alpha period, accomplishing these large tasks required the team's full commitment to cross-disciplinary collaboration, as well as hard looks at our original plans and vision for the game. At the midpoint of development, the team needed to come together to answer questions about the game that weighed our initial aspirations against their feasibility.

For establishing a broader world, this meant determining what the game looked like outside of the kitchen. Questions abounded about the world's size, how many characters were in it, and how long the demo was going to be since the project's inception, but they had previously been impossible to answer without having gauged the team's development speed; as such, Beta was the first real opportunity the team had to answer them and shift plans accordingly.

Similarly, expanding the core loop into something fitting for a full game was essential. The cooking mechanics which took centerstage during the first pass of gameplay in the Alpha period wouldn't be enough; players needed a reason to keep engaging with the experience after understanding these basic building blocks, and the simple flavor combination system alone wasn't enough. However, after observing players' experiences with the early build at Alphafest, it became clear that pivoting to the interaction system the Design team planned would be a misstep, preventing players from interacting deeply with the puzzle elements of gameplay they were enjoying. In this vein, additional questions needed to be answered regarding exactly which directions the core loop ought to expand in.

5.2 - Growing the Game World

The characters were, at this stage, bright and vivid and full of life – but they needed a world to live in. This world needed to both match their style, and support them without overpowering them. It needed to be game-ready, interesting, and evoke the proper emotions that the game required: cozy, homey, and natural.

5.2.1 - The Tree

One of the challenges we faced on this project was figuring out how the game's world will be laid out. This was critical to get right as it would help the player understand how they can interact with the world, and had a large impact on the feel and tone of gameplay. When designing the tree, the Artists and Designers had to keep in mind multiple factors: the feel of navigating different spaces, the cohesion with the rest of the game, and the game design that all of the art supports.

In the study and design of games, it is helpful to define different sizes of spaces in which the player can move, for the purpose of understanding the experience of navigation in a game. Three important types are narrow spaces, intimate spaces, and prospect spaces. Narrow spaces are very small and confining, with little room to move and barely room to interact. Intimate spaces are slightly bigger: not too big to take in, and allowing for comfortable player movement to reach all parts of the space. Finally, prospect spaces are large, allowing for the player to move between intimate spaces and view locations from a different, further away perspective.[53] In traditional games with enemies and death states, narrow and prospect spaces elicit feelings of fear: the inability to properly move in narrow spaces creates feelings of claustrophobia, and the open land with hidden areas in prospect spaces can make the player feel exposed to attack. However, in our game, the player is never in danger of attack, and so the size of the space has a different impact on gameplay experience.

The narrow spaces in our game were the guest rooms. We created a semi-monochromatic guest room design with furnishings that were separated from the background; this way, we could create multiple layouts in different colors to give visual variety without a significant increase in workload. Having different colored rooms would also allow players to conceptually differentiate between the nests, which all used the same exterior asset. To be easily accessible while walking along a branch, the entrance was placed in the center of the nest – this also aesthetically matched the warped perspective used throughout the game. Due to being modeled after birds' nests, the rooms are very small. There is barely room to fly, and all the player can do is walk back and forth and interact with characters. This was designed to prevent the player from worrying about

[53] Christopher Totten, *An Architectural Approach to Level Design: Second Edition*, (New York: A K Peters/CRC Press, 2019) 161.

navigation or gameplay, in favor of focusing on the nest's inhabitant and serving their food or speaking with them about their stay.

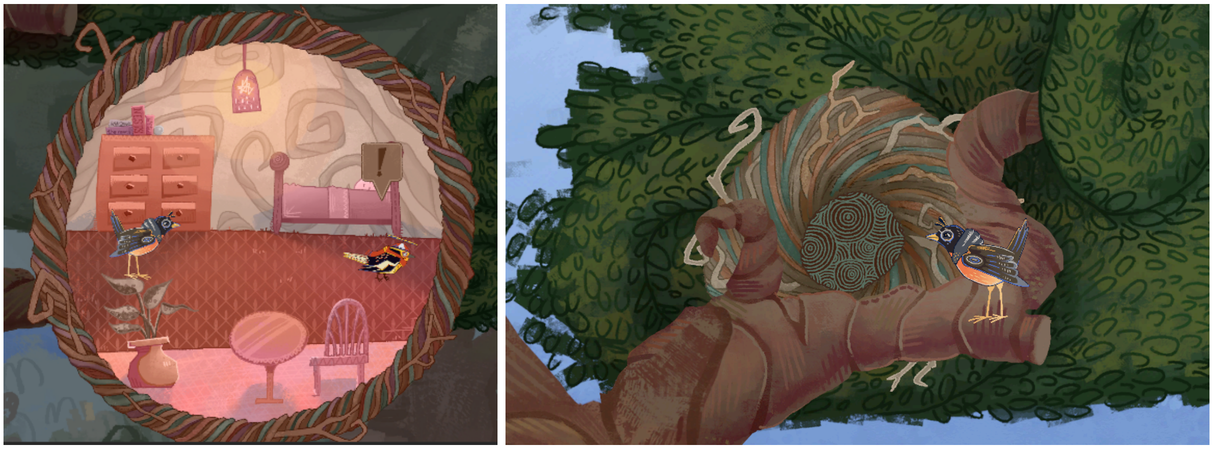


Figure 36: Interior of guest room (left) without door and exterior of guest room (right).

The main intimate space in the game is the kitchen, where the main gameplay mechanic of cooking occurs. Due to its function as the most highly used gameplay space, this room was designed first, and its stylistic choices directly influenced the rest of the backgrounds. Significant thought was given to how the player would appear against the background due to the stylized perspective of the entire game. Traditional side-scrolling perspective was considered, along with a side-scrolling version that allowed for a more top-down perspective on the kitchen, creating more room and easier access for the large amount of appliances that were being considered at this phase of development. Ultimately, the team decided on a combination of the two.

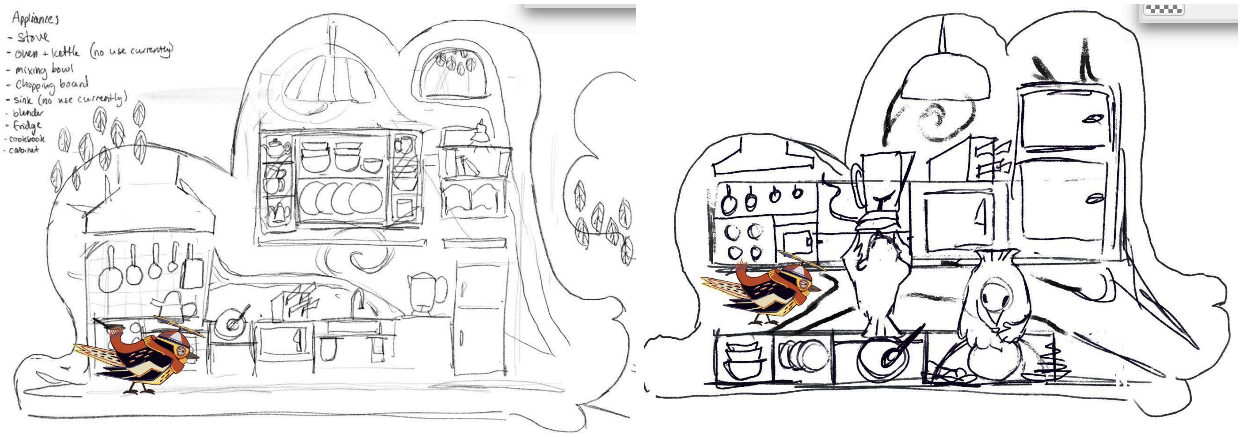


Figure 37: Early options for the kitchen perspectives: sideview (left) and warped top-down (right).

Once the perspective was decided, actual asset creation could begin. It was important to create a space where the player character would stand out in terms of shape language, color, and general sizing, yet still seem to fit in the space. After many iterations based on internal and external feedback and playtesting, a final version was created, with interactable appliances separated from the background to allow for easier in-engine highlighting and possible future manipulation of their positioning.

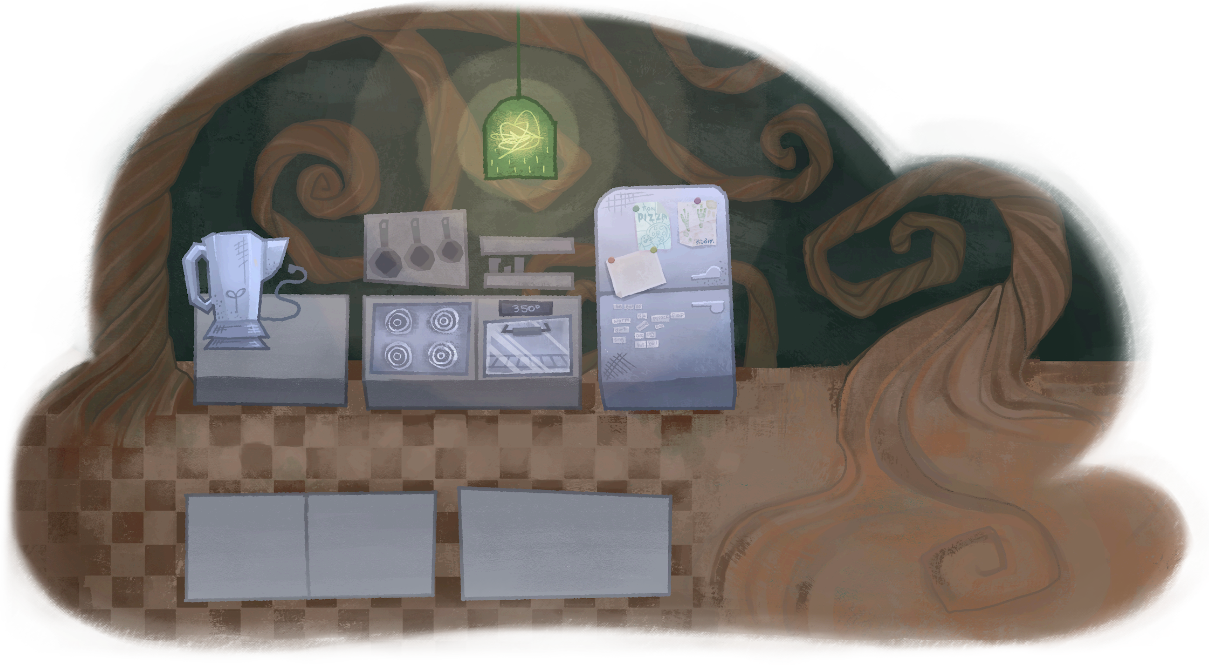


Figure 38: Final kitchen design, before the pass for adding clutter.

With the focus on visual growth as a means of showing player progression, as defined in our pillars (see section 1.3), the design of the tree focused on modularity and growth. The tree was meant to start small, and allow for the player to grow branches out of it to increase its size. This fit well with our layered approach, allowing the artists to focus first on the kitchen, nests, and base, before expanding to include more branches as the scope allowed.

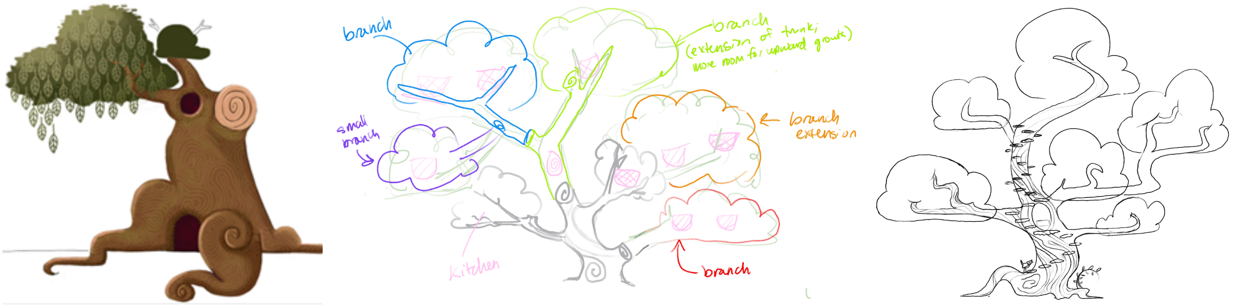


Figure 39: Early design of the B&B tree (left), mid-iteration (center), and final tree design (right).

Due to the large amount of pieces that went into the tree, this collection of assets serve as a good example of the effective use of art guides. A guide was created alongside the trunk, allowing for the task of creating branches to be passed around among the artists, depending on what needed to be done in each sprint.

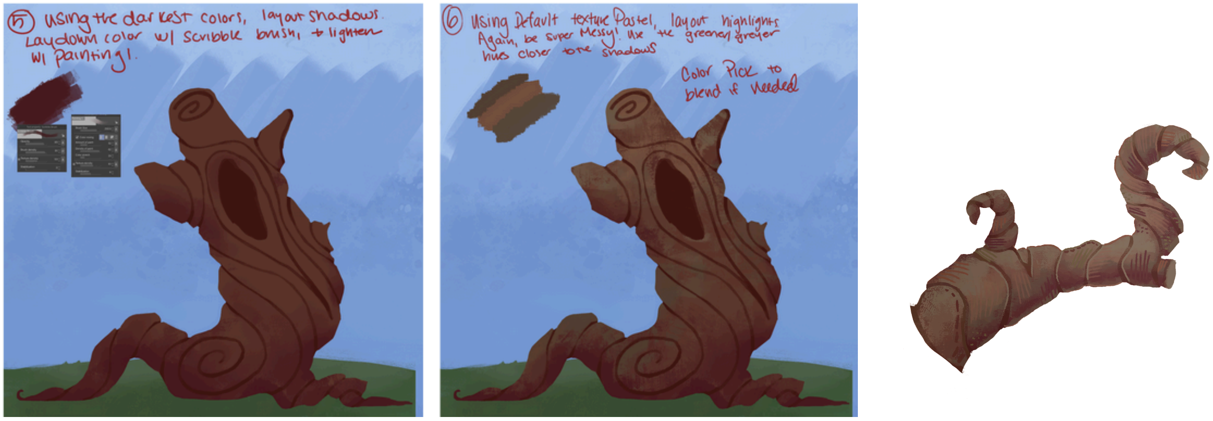

5 using the darkest colors, layout shadows. Laydown color w/ Scribble brush, + lighten w/ painting!.
6 Using Default texture Pastel, layout highlights Again, be super Messy! Use the greener greyer hues closer to the shadows
Color Pick to blend if Needed

Figure 40: An excerpt from the tree painting guide (left and center) and one of the branches created using it (right).

5.2.2 - Customers

An important aspect of expanding the game world meant populating it with a varied collection of birds and characters — chiefly among them, the customers that visit the B&B. Whereas the process of making a character had chiefly been focused on designing, animating, and writing them to be memorable and interesting, making these characters into customers required careful balancing and consideration as to how each individual, their preferences, and the challenge they posed would fit into the overall gameplay structure.

The current customer design and their balancing came about from the feedback of several playtesting sessions as well as the realization that we would need to more heavily script the gameplay experience of the player. Because of this, all customers have specific and predetermined preferences and spawning opportunities meant to fit the tutorialized nature of the demo.

One area that was initially more complex and random was the circulation of customers throughout a player's experience with the game in a long-term context. In that version of the game's design, customers would potentially randomly arrive at the B&B if there were empty rooms, becoming more frequent as the player became more successful. Currently, the first few days have specific characters appearing to teach the player basic game mechanics. In the rest of the game, a number of customers appear every day, with the particular number increasing as the demo progresses.

These customers have different spawning behavior depending on whether they are wandering Travelers or citizens of the town. Travelers stay multiple days in a row, and are randomly chosen from the list of Travelers at the start of their stay. They will often spawn inside of their designated rooms in the B&B. This allows the player to quickly learn about their preferences and character, but there is a chance that they will not be seen for the rest of the demo once they leave. On the other hand, town characters (Townies) are randomly chosen every day and often placed at random spots throughout the tree or on the ground. This allows the player to learn their preferences over the course of the demo, as they will likely reappear multiple times.

Said preferences are particular to their characters, and are designed to elicit different gameplay experiences. The first couple birds enjoy flavors that are able to be created with the small number of ingredients in the tutorial. Other birds have large numbers of likes, or large numbers of dislikes, to indicate that they are either an adventurous or picky eater. This is combined in the scoring formula with an internal point threshold for each bird that determines how good a dish must be for them to enjoy it. Additionally, most characters' flavor preferences correlate with their favored dishes: either positively or negatively. For example, Hector is the first bird who visits the B&B, and most of his orders are for fruit smoothies. Fruits, in game, are commonly sweet, making his order make sense to the player, guiding them toward the right choices and rewarding them for attaining both the simple dish and the simple flavor request. In contrast, the final guest of the game, Fernando, requests apple crisp, which requires an ingredient that adds flavors that he dislikes. At this stage of the game, the player is challenged to balance the flavors back to something he would enjoy, despite his order being at odds with his preferences.

5.2.3 - Supporting Gameplay

When the game was passed to playtesters, focus needed to be kept on testing the design decisions made about cooking, characters, and movement. The artwork needed to support and augment this focus, while being careful not to distract from it. Creating art for a video game is a constant battle between maintaining aesthetic integrity while also balancing the players ability to clearly and easily interact with said art. Because the tree is the central point of gameplay, the artists gave special attention to each stage of the design process. The process of rendering the tree consisted of multiple iterations between hues, shades, and additional details in an attempt to marry a more painterly style with bold, saturated scribbles imitating bark. Eventually, the subteam was able to settle on a design that encapsulated both of these styles while also remaining simple enough to not overshadow the detail in the birds themselves.

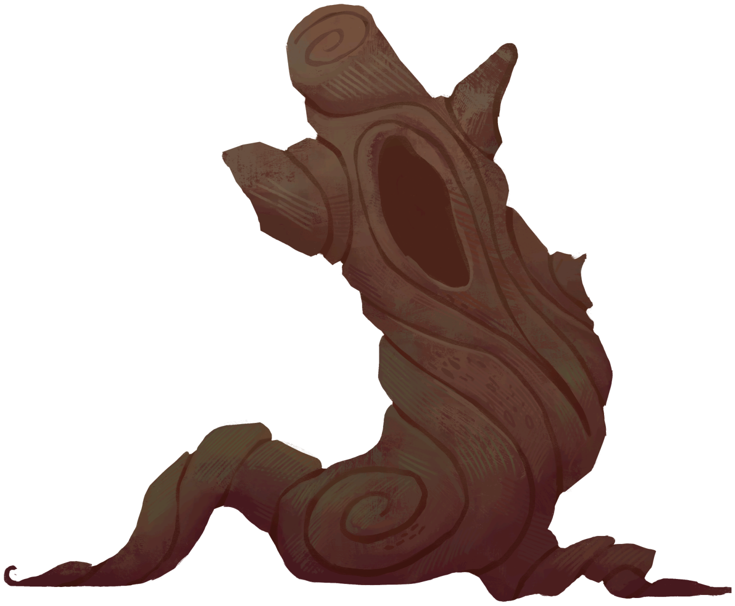
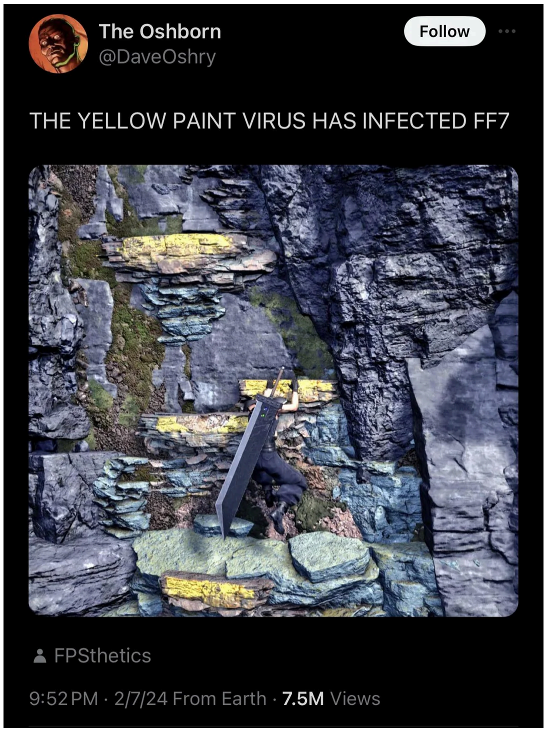


Figure 41: Tree Base before navigation edits.

Though the team was satisfied with the carefully planned out final design, it was quickly discovered upon playtesting that further iterations were needed. The tree, while traversable, was not as navigable as the art team had hoped. Most players had extreme difficulty finding the kitchen; in fact, the majority of players only found the kitchen by accidentally stumbling or falling into it. Since the kitchen is essential to gameplay, players were both confused and frustrated as to what their next steps would be.

When trying to make environments navigable, it's common for some developers to use bits of brightly colored paint to highlight interactable objects and paths that further the player's journey. While this is a fine solution in some situations, this method has recently come under increased scrutiny in both gamer and game development communities following a post by Dave Oshry,[54] founder of New Blood Interactive.

[54] Dave Oshry (@DaveOshry), "THE YELLOW PAINT VIRUS HAS INFECTED FF7," X, February 7, 2024, 9:52 pm, https://x.com/DaveOshry/status/1755469586999505137?s=20.



The Oshborn
@DaveOshry

THE YELLOW PAINT VIRUS HAS INFECTED FF7

FPSthetics

9:52 PM · 2/7/24 From Earth · 7.5M Views

Figure 57: Aforementioned post from Dave Oshry regarding yellow paint signifiers.

Players and developers expressed mixed opinions, but generally came to the conclusion that this method is appropriate in some cases, but can be a result of lazy or undefined artistic direction in its intersection with design. Many developers jokingly added their own yellow paint to their games to navigate absurdly obvious levels, something which Marketing was able to capitalize on in our own game.

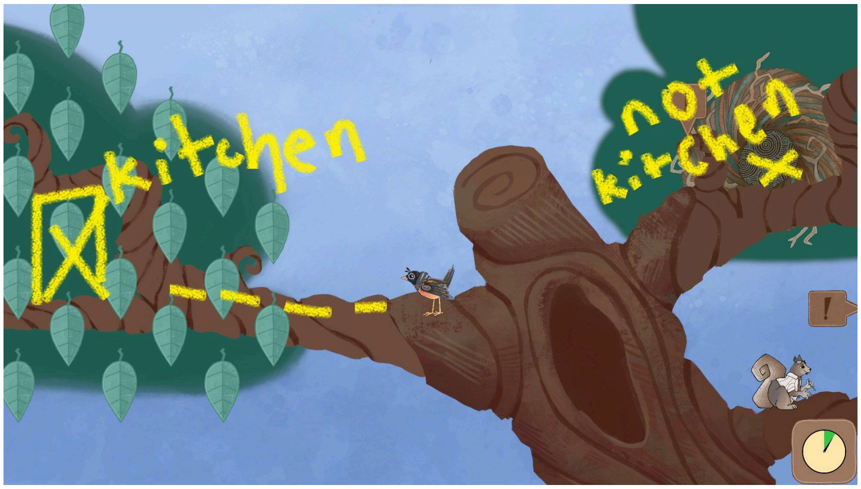


Figure 42: *Operation Breadcrumbs*' parody of Dave Oshry's original post.

As an Art subteam, we decided that simply brightly marking the kitchen was not the solution we wanted to go with; we could, however, take inspiration from the concept to make something more appropriate for our style. The team came up with four solutions as shown in Figure 43. Sketch A plays with the idea of rustic signages marking the kitchen, with the addition of a town sign inspired by signs for towns in Massachusetts. Sketch B concepts animated swirls of classic cartoon scent lines which would lead the player to the inside of the kitchen. Sketch C is perhaps most closely related to the yellow paint method, by having the bright lights of the kitchen peek through the leaves of the kitchen, encouraging the player to take notice and enter. Finally, Sketch D takes the idea from Sketch A of using signs, and integrates them into the tree in a more natural way through carvings.

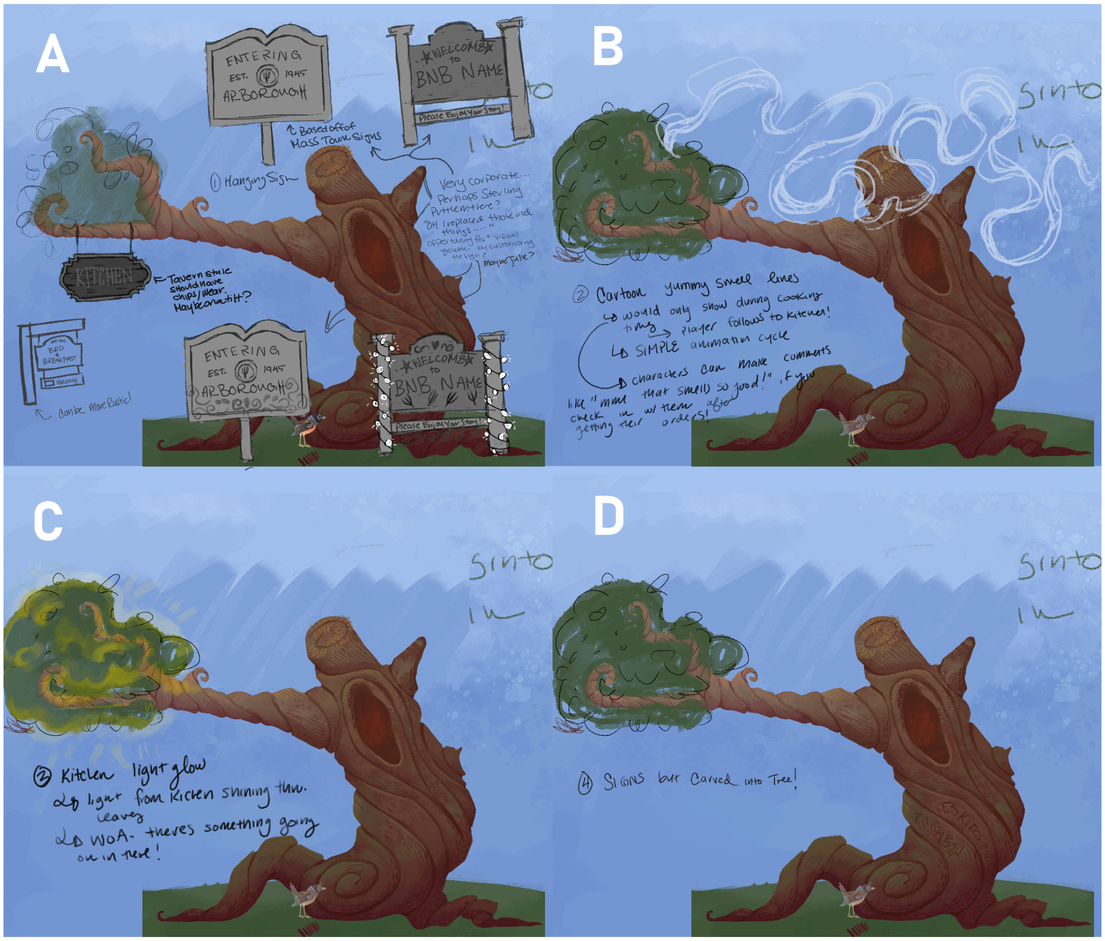

A
ENTERING
EST. 1945
ARBOROUGH
★WELCOME★
to
BNB NAME
Please Enjoy Your Stay!
↑ Based off of Mass Town Signs
① Hanging Sign
Very corporate... Perhaps Sterling Puttemthere? "OH I replaced those old things...." opportunity for visual growth by customizing the sign? Maybe Julie?
KITCHEN
← Tavern style should have chips/wear. Maybe on a tilt?
EST. 1941
BED & BREAKFAST
Vacancy
↖ Can be More Rustic!
ENTERING
EST. 1945
ARBOROUGH
★WELCOME★
to
BNB NAME
Please Enjoy Your Stay!
B
② Cartoon yummy smell lines
↳ would only show during cooking time → player follows to kitchen!
↳ SIMPLE animation cycle
↳ characters can make comments like "mmm that smells so good!" if you check in w/ them after getting their orders!
C
③ Kitchen light glow
↳ light from kitchen shining thru. leaves
↳ WOA- theres something going on in there!
D
④ SIGNS but Carved into Tree!
← KITCHEN
ROOMS
Sinto
in

Figure 43: Options for kitchen signage; Sketches A (top left), B (top right), C (bottom left), and D (bottom right).

After discussion, the subteam came to the conclusion that Sketches C and D both aesthetically made the most sense and would be the easiest to execute without dramatically increasing our points in the sprint. With a few additional tweaks, the final tree was complete!

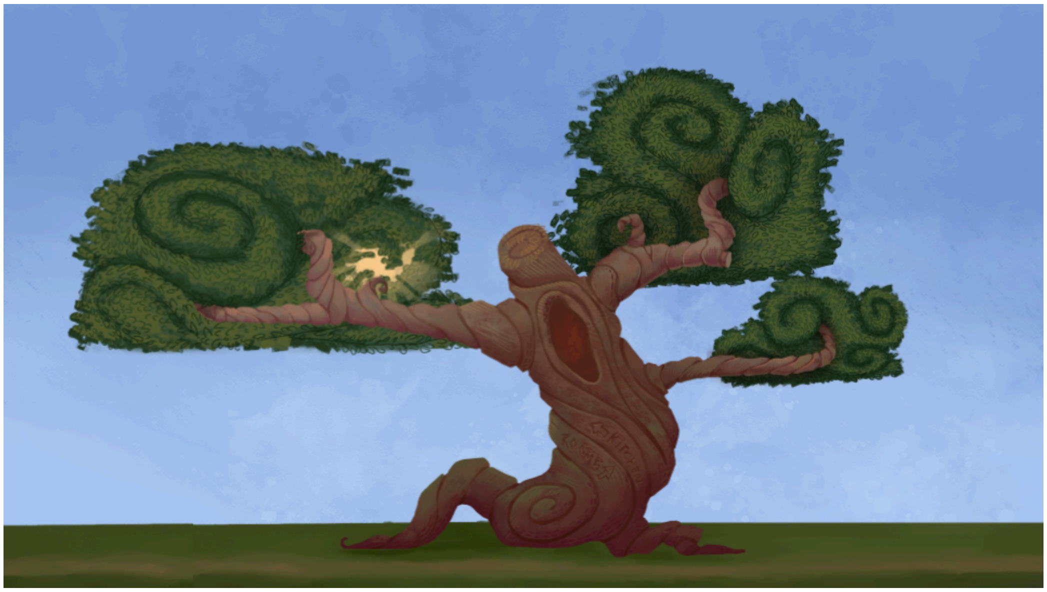


Figure 44: The final tree base with the new kitchen signifiers.

At our next playtesting session, the edits that the subteam had made to the tree were overwhelmingly positive. Game observers noted that players had much less difficulty navigating through the tree, leading to an overall more satisfying and enjoyable gameplay experience. The collaboration between Art and Design proved to be a success.

5.2.4 - Providing Context Through Cutscenes

With the world being built up around the player character, an important bit of context was still missing: why was the player here? It was unclear who Robin was, what their relationship with the B&B was, and even who Sterling was. As part of writing the story of the game, the Writers were in charge of showing the player through circumstances the history and situation of the game at the start. This was accomplished using cutscenes to tell important story beats in addition to the daily dialogue and quests.

Cutscenes were planned strategically. There were multiple arcs that needed to be covered; the most crucial was the general introduction arc which spanned the demo, with additional side character storylines that happened as part of this arc or separately.

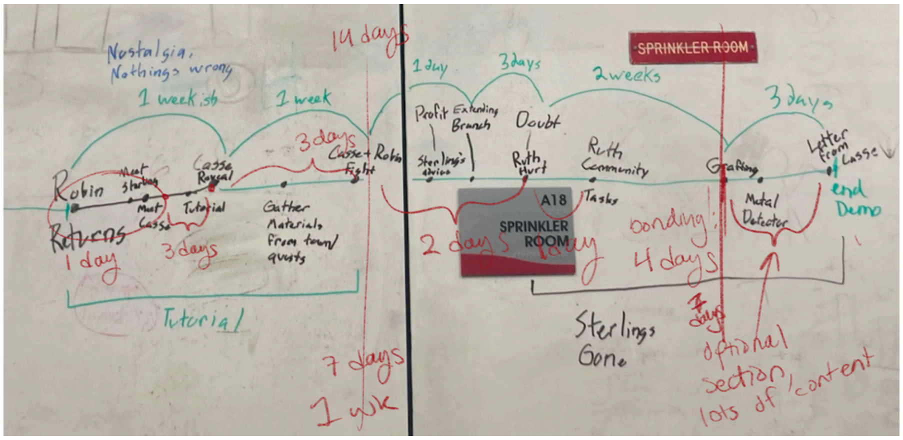
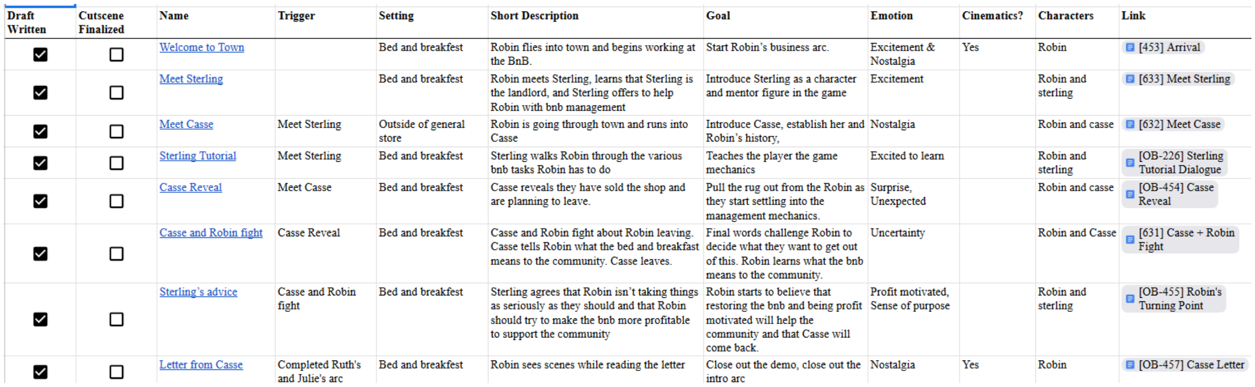


Figure 45: The introduction arc with a revised timeline (see Appendix F for cleaner image of current narrative structure).

The first step was to make sure that the introduction arc fit into the demo. This timeline had been drawn up early in production, and it was now adjusted to fit a fifteen day demo schedule. As part of this, the character arc of Ruth the carpenter and her friendship with Robin was essential to include. Her injury from the tree and subsequent instruction to Robin on how to grow the tree themselves served as an important introduction to the town that would set the game up for a story about Robin getting closer to their neighbors. An additional storyline that was drawn up but discarded was that of Julie, a crow with a habit of collecting junk and a problem of not knowing what to do with it. In it, Julie crafts something for the town from her collected items before leaving; her departure exemplified a way to incorporate the traveling birds in the story as short-term but still impactful members of Arboro's community. Due to time constraints however, Julie's storyline was moved past the intro arc and was not written for the demo.

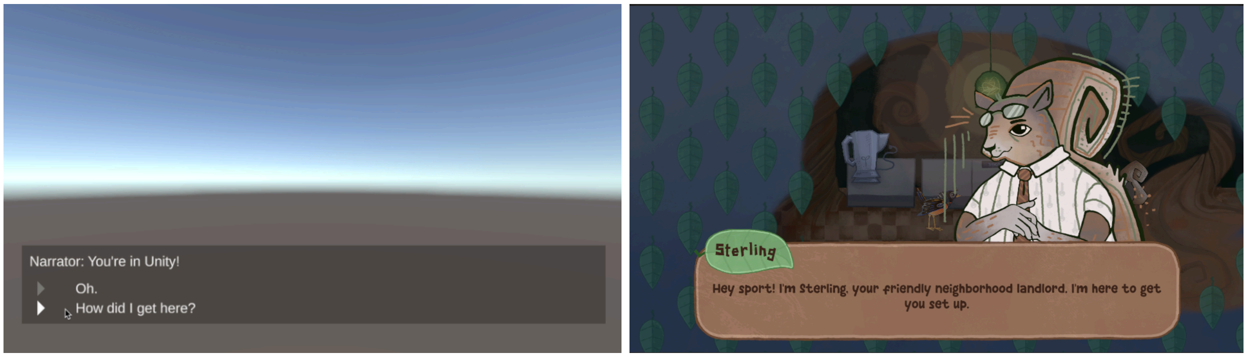


Draft Written	Cutscene Finalized	Name	Trigger	Setting	Short Description	Goal	Emotion	Cinematics?	Characters	Link
☑	☐	Welcome to Town		Bed and breakfest	Robin flies into town and begins working at the BnB.	Start Robin's business arc.	Excitement & Nostalgia	Yes	Robin	[453] Arrival
☑	☐	Meet Sterling		Bed and breakfest	Robin meets Sterling, learns that Sterling is the landlord, and Sterling offers to help Robin with bnb management	Introduce Sterling as a character and mentor figure in the game	Excitement		Robin and sterling	[633] Meet Sterling
☑	☐	Meet Casse	Meet Sterling	Outside of general store	Robin is going through town and runs into Casse	Introduce Casse, establish her and Robin's history,	Nostalgia		Robin and casse	[632] Meet Casse
☑	☐	Sterling Tutorial	Meet Sterling	Bed and breakfest	Sterling walks Robin through the various bnb tasks Robin has to do	Teaches the player the game mechanics	Excited to learn		Robin and sterling	[OB-226] Sterling Tutorial Dialogue
☑	☐	Casse Reveal	Meet Casse	Bed and breakfest	Casse reveals they have sold the shop and are planning to leave.	Pull the rug out from the Robin as they start settling into the management mechanics.	Surprise, Unexpected		Robin and casse	[OB-454] Casse Reveal
☑	☐	Casse and Robin fight	Casse Reveal	Bed and breakfest	Casse and Robin fight about Robin leaving. Casse tells Robin what the bed and breakfast means to the community. Casse leaves.	Final words challenge Robin to decide what they want to get out of this. Robin learns what the bnb means to the community.	Uncertainty		Robin and Casse	[631] Casse + Robin Fight
☑	☐	Sterling's advice	Casse and Robin fight	Bed and breakfest	Sterling agrees that Robin isn't taking things as seriously as they should and that Robin should try to make the bnb more profitable to support the community	Robin starts to believe that restoring the bnb and being profit motivated will help the community and that Casse will come back.	Profit motivated, Sense of purpose		Robin and sterling	[OB-455] Robin's Turning Point
☑	☐	Letter from Casse	Completed Ruth's and Julie's arc	Bed and breakfest	Robin sees scenes while reading the letter	Close out the demo, close out the intro arc	Nostalgia	Yes	Robin	[OB-457] Casse Letter

Figure 46: The list of cutscenes from the introduction arc.

Arguably the two most important parts of the introduction arc were what would likely be the most memorable to the player: the beginning and end. The team chose to make both of these cutscenes letters between Robin and Casse, to frame the narrative in a symmetrical manner that highlighted the voices of two important characters. The artists chose to give these letters more attention than the other cutscenes, giving them custom backgrounds rather than having them take place on top of the gameplay camera, as most other cutscenes did.

Yarn Spinner[55] served as an essential base of the cutscene system, allowing writers to easily script and implement cutscenes in engine; however, we knew that we would need to expand upon it greatly in order to tell the story. Yarn Spinner provided a bare-bones, visual novel-style display and an easy-to-use script format for the writers that allowed them to script lines of dialogue and player choices. It also allowed writers to trigger custom functions, a feature used to make cutscenes affect the game's world.



Narrator: You're in Unity!

Oh.

How did I get here?

Sterling

Hey sport! I'm Sterling, your friendly neighborhood landlord. I'm here to get you set up.

Figure 47: The default Yarn Spinner display (left) next to an in-game screenshot (right).

Given that the game's story is largely character-driven, we thought it was important that the cutscene system be able to display character portraits and allow for characters to switch portraits at the writer's discretion. This was especially important for the more emotional beats of the game like Robin and Casse's fight in the introduction arc. While not all characters in the game would have multiple emotive portraits, all characters had a neutral portrait to further showcase their personalities, humanize the characters, make the game's world feel richer, and achieve a cozy, community experience.

In order to enhance the dialogue in the management portion of the game, we expanded the dialogue system to allow traveler characters to choose dialogue from a specified set of personalities. This enabled characters to choose from a pool of dialogue on each player interaction, to keep the dialogue feeling fresh (as explored in Section 4.2.3). We also employed the use of text animations, a feature that Yarn Spinner does not natively support, as a way to bring the dialogue to life.

[55] Yarn Spinner, (Yarn Spinner, 2015), PC.

5.3 - A Recipe for Success

With the core loop of the game completed, it was time to expand upon it and test it to its fullest. This meant adding additional systems on top of the existing one, improving readability and comprehension of the first few days of gameplay, and enabling the user to easily navigate and understand the many moving parts of the game systems.

5.3.1 - Expanding the Core Loop

As development continued, we iterated and refined what would become the final systems of the core game loop. Alphafest had already helped the team identify which parts players found fun in the original builds; expanding on that fun and our plans for the main gameplay loop would be the constant challenge that future iterations would try to answer.

Initial plans for the gameplay loop, established in Design's pre-production, had considered both cooking and management tasks to be completed around the bed and breakfast as equal components to gameplay (Section 3.2.3). Tasks such as cleaning rooms, checking birds in and out of the establishment, and balancing the cook time of appliances during the morning section of gameplay were meant to make gameplay timed-based and challenging, differentiating it from an afternoon section of gameplay where the player could wander the town, forage for ingredients, and interact with NPCs to progress the story. At this point in the development process, most of these design intentions remained non-specific plans. Only the cooking aspect existed, in which players could combine ingredients and balance flavors to get a generic dish assigned to each appliance. Given that this was the only element so far in the morning section of gameplay, it was essential that this segment was fleshed out to create an engaging core loop, even before focusing on other sections.

With early playtesters drawn to the puzzle aspects of the cooking system and the team deciding to lean into this appeal, Design found themselves flatfooted: plans for timer-based gameplay and forcing the players into making these complex choices with a limited amount of time no longer fell in line with our pillar for deliberate experimentation (Section 1.3). Furthermore, a lot of the other tasks associated with managing a bed and breakfast weren't engaging without this timed element; chores such as cleaning rooms were interfering with enjoying the breakfast creation, with multiple playtesters finding the task of making beds confusing or even pointless.

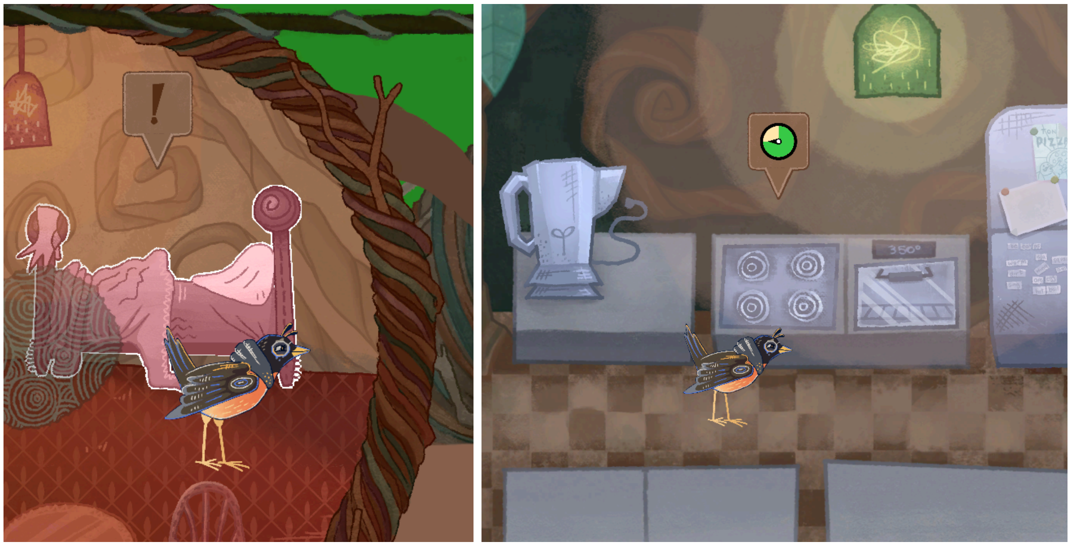


Figure 48: The discarded bed-making interaction (left) and timer graphical interface (right).

Eventually, the choice to cut timed gameplay mechanics in favor of expanding the aspects of the game that were already working, such as the cooking system, became the clear option for the team. This of course meant discarding many of the mechanics and content created related to chores like cleaning rooms, but it ultimately ensured the expanded gameplay loop was in line with our pillars.

To expand the cooking system beyond the simple mechanic of combining ingredients to get the correct combination of flavors, Design took a new approach involving the types of foods being created. In earlier layers, each appliance had correlated to a single meal, and the player needed to use that appliance to create the right meal as ordered by each guest. The only complexity involved in this mechanic was that different appliances had a different number of ingredients they could accept.

After resolving to give cooking more depth of gameplay, this mechanic was replaced with a more intricate series of combinations and discovery; different foods were created based on the type of ingredients and appliance used. Whereas any type of ingredient put in the blender had made a smoothie in the previous iteration, the new system provided for unique recipes and dishes: fruits in the blender might make a fruit smoothie, vegetables could make a vegetable juice, and meat could become minced meat — an in-betweener ingredient that could then be used to make a variety of dishes such as sausage.

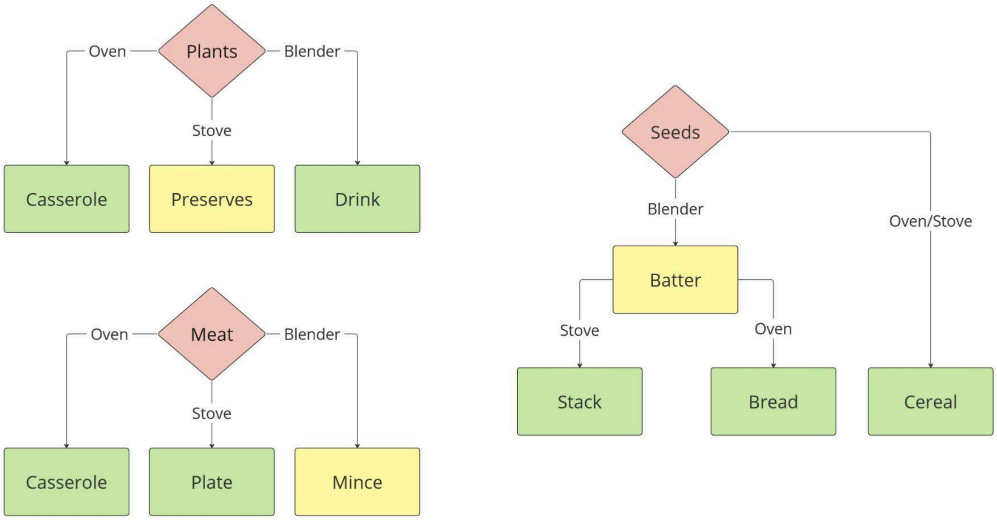


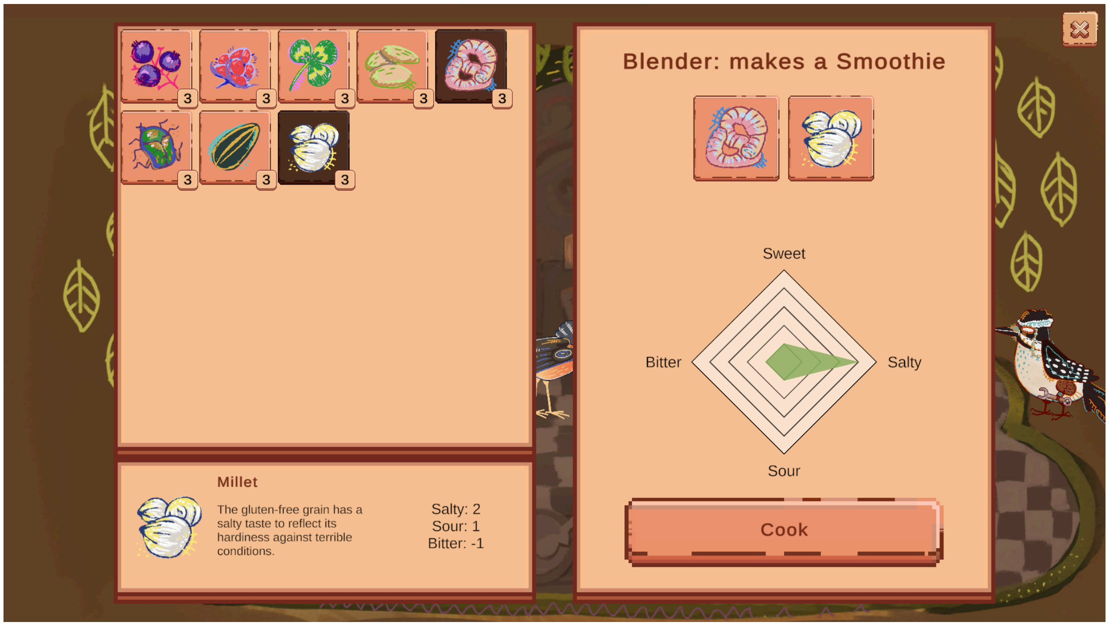
Figure 49: An early iteration of the rules for combinations used to make different food groups.

Broader classifications of these foods called food groups would evolve to encompass all the types of breakfast foods players might come to expect, with the Design team final settling on six different types of dishes: Drinks, such as the aforementioned smoothies and juices; Plates, like stir fries and omelets; Stacks, the ever-familiar pancakes and waffles; Breads, a general classification for baked goods; Cereal, most unbaked, grain-based breakfast items; and Casseroles, for non-bread items baked in the oven. Outside of these six, Slop remained, existing as a catch-all for any combinations deemed too gross or eccentric to fit into a recipe.

Within this system, customers would request a specific meal when they give the player their order. Players would still need to be cognizant of a customer's flavor preferences, but they would also be scored based on serving the correct type of meal — or failing that, receive partial points for a dish in the correct food group. Now, the player would need to pay more attention to the combinations of ingredients and appliances used to create a dish. This undoubtedly made the core gameplay mechanic more complicated, but it also opened the path for the player to discover more through experimentation; an entire onslaught of dishes could be uncovered through the different combinations a player might make.

5.3.2 - An Interface for the User

The decision to focus primarily on cooking meant that the associated UI would need substantial alterations, especially since the changes meant that all of cooking would now take place from within these menus. We had not planned on making a menu-focused game; as such, not as much attention was given to them in the beginning, and we had to adjust our focus based on the changes.



Millet

The gluten-free grain has a salty taste to reflect its hardiness against terrible conditions.

Salty: 2
Sour: 1
Bitter: -1

Figure 50: Initial design of the cooking interface.

The initial cooking interface required players to know their customers' orders and flavor preferences with no visual cues; naturally, players were frustrated that they had to memorize so much information in order to prepare a single meal. If they forgot an order, they would need to go back to the customer, interact with them again, and then return to the kitchen — a tedious back and forth.

To solve this, we developed the meal ticket system. Every time a player interacted with a new customer, they would receive a meal ticket with the customer's name, food request, and flavor preferences. This ticket would stay at the top of the screen until the food was served, preventing the player from making unnecessary trips to the same character.

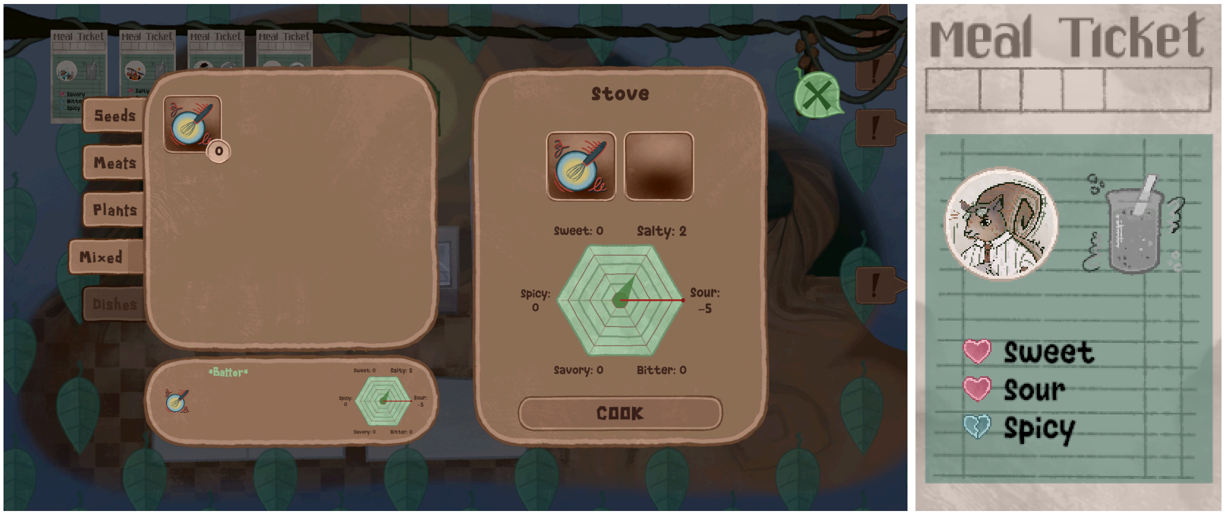
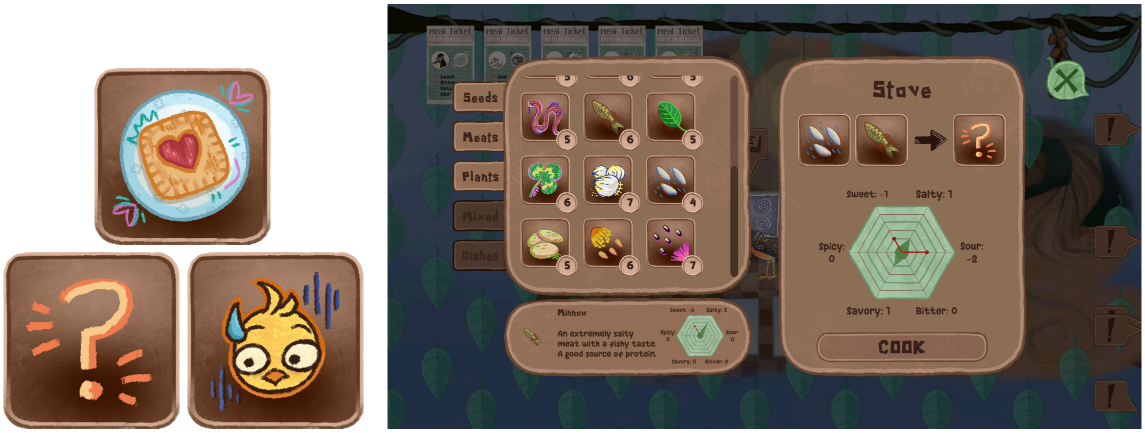


Figure 51: Second design of the cooking interface (left) with a meal ticket (right).

Once the cooking system expanded to include unique meal types and food groups, new complications arose (Section 5.3.1). As part of encouraging deliberate experimentation from the player, no feedback had been given to them as to what types of meals they were making in the menu. Instead, players had to keep a running mental catalog of all the food combinations they had tried, as well as the resulting meals and various methods for achieving them. While we intentionally did not add the ability to auto-fill new recipes to compel players to experiment with making different meals, this system quickly spiraled into a memorization game. Playtesters often commented that they felt as though they needed a notepad to keep track of what they had done and learned.

Based on the feedback we received, the third version of the cooking interface added the ability to see what would be made by combining the two selected ingredients, substantially reducing the need for guessing and checking, provided the player had already discovered how to make the meal. This preview was indicated by an arrow and a third box that would show either a question mark (if the player had not discovered the resulting meal), a concerned bird face (if that combination would result in slop), or the icon of the resultant meal (if that meal had been discovered already).



An extremely salty meat with a fishy taste. A good source of protein.

Figure 52: Icons of a pastry dish, an undiscovered combination, and a concerning combination (left) and the third design of the cooking interface (right).

As the amount of information visible on this screen continued to grow, a tipping point was finally reached involving the aforementioned meal tickets. While players in early playtests had appreciated the system, one of its main flaws was the inability to actually see the customers' meal tickets from within these menus; when they were open, their size and placement entirely obscured the tickets. While players were no longer forced to return to the same customer multiple times if they hadn't memorized their orders and flavor preferences, they now needed to switch between the cooking and navigation views — the same core issue, just with less travel involved.

To accommodate this and adjust for all the new information present in these menus, the final iteration of the cooking UI involved a near-total overhaul of the layout in order to include the meal tickets, food combination previews, and the original information from the menu. The flavor chart was made larger to increase legibility, and the meal ticket was redesigned to include each bird's flavor preferences as simple symbols rather than words. This would decrease the number of words on screen and allow for a cleaner look in general, even as the amount of information being conveyed to the player on a single screen increased. These symbols were created to be specifically distinct from one another and easily recognizable from their silhouette; they replaced the words on the flavor chart for the same reasoning.

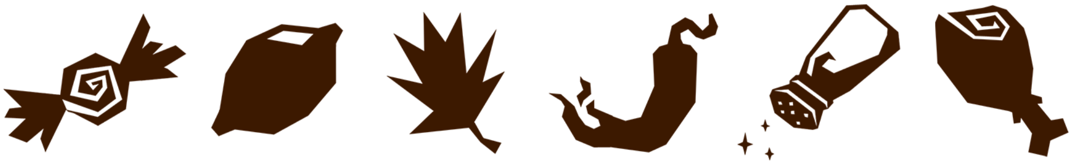


Figure 53: The six flavor icons: sweet, sour, bitter, spicy, salty, savory (left to right).

Additionally, meal tickets were split into two versions: the simple version and the extended version. The simple version had just the customer's portrait (seen in the upper left corner below) and is what the player would see when scanning over all the meal tickets. The extended version is what would be seen when an individual meal ticket was clicked on; since only one meal could be made at a time, there was no need to overwhelm the player by showing all the flavor preferences for every bird's request.

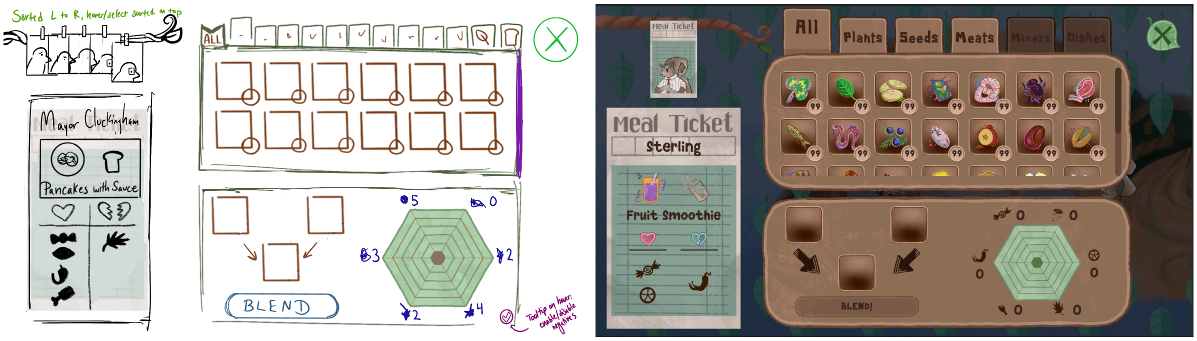
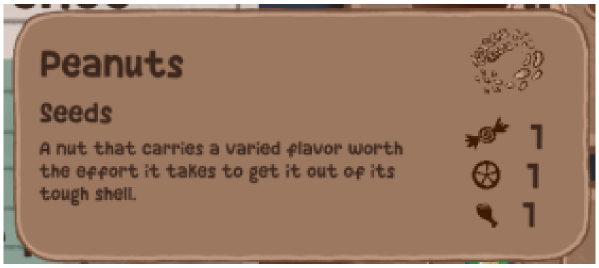


Figure 54: Fourth design of the cooking menu interface (left) and its in-game implementation (right).

Although beneficial, components like item descriptions were not strictly needed for cooking. Players seldom paid them much attention, and the small radar chart for each individual item was so small as to render it useless. Instead, this information was moved to a tooltip which appears when hovering over an item, allowing this valuable spatial real estate to be given to the meal tickets.



Peanuts

Seeds

A nut that carries a varied flavor worth the effort it takes to get it out of its tough shell.

Figure 55: A tooltip showing an ingredient's description and flavors.

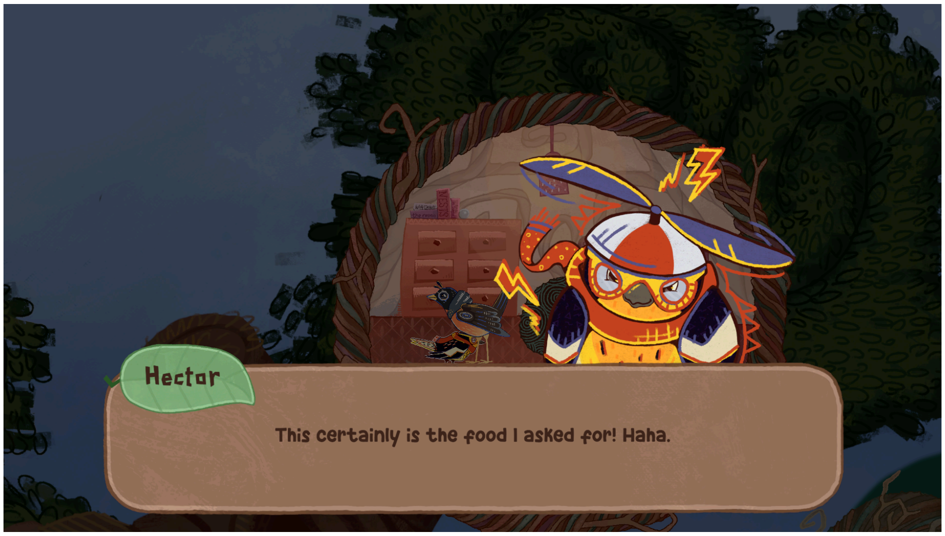
As the UI entered its final form, players were able to view all meal tickets from inside the cooking menus, easily interpret the flavor chart system, and not be bombarded by copious amounts of small text. Recipes for more meals and variations of existing ones could now be introduced without adding confusion for the player. Despite not initially planning to create a menu-focused game, the team adapted well to the changing status of gameplay to create a complex — but not confusing — layout for the system.

5.3.3 - Characters Reception of Meals

As the cooking system underwent changes with recipes beginning to be implemented, it was important that our first pillar — prioritizing community over money — continued to be a point of emphasis. It would be all too simple for this expanded system to compel players to prioritize high scores for the sake of doing well in the context of gameplay over their interactions with characters. To make this element entertaining for players, the team determined it was crucial to craft believable character responses to their meals.

With support from Design, Writing had first tackled this task by dividing customer responses into three categories: Bad, Neutral, and Good. Characters would typically have two responses per each of these categories, allowing for slight variation while keeping scope low. Players would receive a Bad response if they served characters a dish with flavors that they disliked, a Good response for dishes that contained mostly flavors that they liked, and a Neutral response for anything in-between**.** In keeping with the previous implementations of characters and their dialogue, Townies would have their own, specific responses for each of these categories, while Travelers would respond based off of the personality pool that they were in.

All this intended to provide another mechanism for customers to meaningfully interact with the player, bolstering their connection to Townies and Travelers within the wider community. However, as Design began to refine the scoring and cooking systems to accommodate recipes, the straightforward nature of these responses and their link to flavors was lost. Without adjusting the content itself, the different categories of response were instead assigned to buckets on a scale of 0-5 points, where a different number of points were given for correct flavors and correct food types. In this new implementation, some responses began making less sense, especially when they now had to consider food types along with flavors.



This certainly is the food I asked for! Haha.

Figure 56: Neutral response given by a customer based on an average, regardless of whether the correct food type was actually given to them.

Through playtesting, disconnects such as this were readily exposed to the player. Furthermore, many responses weren't helpful indicators of how a particular customer responded to a meal; players often didn't understand whether the response that they were receiving was either Neutral or Good, and their only indication of success ending up being the numerical response they received at the end of the day — a reliance on numbers the team had wanted to avoided. Players needed to receive immediate feedback so that they could quickly adjust their strategy and better tend to the other customers at the B&B.

To fix this, a second pass on dialogue was done to differentiate the types of responses. As the score range of 0-5 was determined to be too narrow feedback for the player, Design expanded it to 0-10; six of those points were attributed exclusively to flavors and how well they matched the character's preferences, and the remaining four went to getting the right type of food. This allowed Writing to widen the gap between response types while adding a fourth category of response for the best possible scores, fittingly named Very Good: this type of response is overwhelmingly positive, emphasizing when the player was doing a great job during gameplay.

After implementing these new meal responses and testing how players responded, Writing and Design deemed our efforts a success. The improved feedback allowed players to immediately understand how well they were preparing meals according to character's likes and dislikes. As a result, gameplay became both less ambiguous and less confusing, with character interaction becoming more satisfying and rewarding as players were able to catch onto a successful strategy more quickly than before.

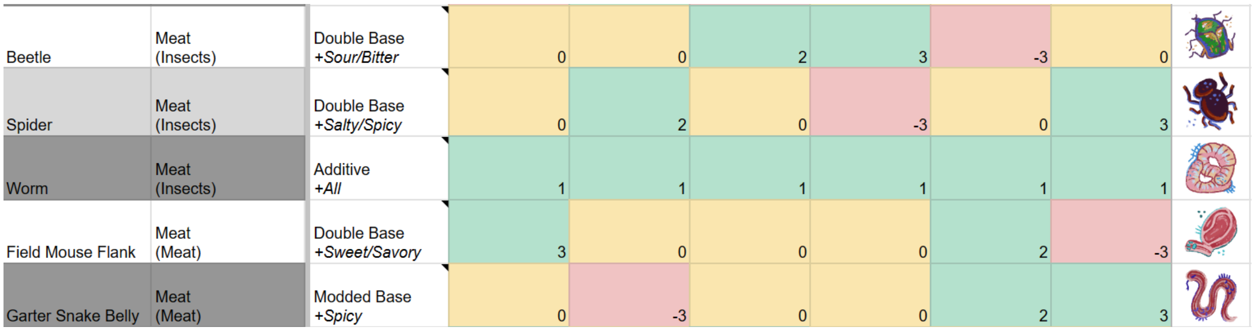
5.3.4 - Balancing & Tutorialization

The core mechanic behind cooking had been well received from the start: the card game we showed off at Protofest worked well, and the early build that was playtested at Alphafest was received positively. However, once the complex structure of recipes were added to the game, the use of flavors became a little muddled. Players were struggling with hitting their target flavor profiles within the restriction of a meal; a smoothie is simple to make, but since most of the plants were initially centered around sweet and tart flavors, it became hard to include salty, savory, or bitter flavors without accidentally making slop. On the other hand, early playtesting showed that some players gave up completely on achieving the right food group in favor of getting the correct flavors, indicating that the imbalance of flavors was distracting from cooking's other main mechanic (Section 5.4).

In order to solve this lack of cohesion within the cooking system, the Design subteam took on two new goals: balancing flavors in ingredients such that they complemented each other and allowed for emergent gameplay, and drip-feeding the complexities of the system alongside recipes so that players better understand how the two systems interact. Together, these two goals were vital to making sure that the progression of gameplay made sense to the player, and that emergent gameplay continued to be entertaining over the course of the game.

As part of balancing the flavors present in ingredients, it was important to establish a foundation for values that could easily be changed to accommodate design choices. To facilitate this, general categorizations were established to designate for the designers what purpose individual ingredients served. For example, some ingredients served as a simple base, providing two points to only one flavor. Others were neutralizers, focusing more on removing flavor than adding.

In combination with these categorizations, spreadsheets were used to balance the number of positive and negative modifiers for each ingredient. This empowered designers to make sure that no flavor was too strongly present at the expense of another.



Beetle	Meat (Insects)	Double Base *+Sour/Bitter*	0	0	2	3	-3	0	
Spider	Meat (Insects)	Double Base *+Salty/Spicy*	0	2	0	-3	0	3	
Worm	Meat (Insects)	Additive *+All*	1	1	1	1	1	1	
Field Mouse Flank	Meat (Meat)	Double Base *+Sweet/Savory*	3	0	0	0	2	-3	
Garter Snake Belly	Meat (Meat)	Modded Base *+Spicy*	0	-3	0	0	2	3	

Figure 57: A sample of ingredients on the spreadsheet used for balancing.

When balancing these ingredients and categories, another important mechanic needed to be considered: recooking. Recooking is an action that allows players to put meals back in the appliance in which they were created with certain ingredients to add or subtract more flavors, such as reblending a smoothie with more fruits or reheating a stir fry with more vegetables. Due to the limited slots in appliances, this ability is key to controlling flavor profiles in a meal. It also means, however, that some meal requests become very easy: if a character likes sweet food, requests a smoothie, and you have ten raspberries in your fridge, each providing +2 sweet, you can throw a handful into the blender to make the perfect meal without needing to think about it. Meals that were more challenging necessitated complex ingredients that added unwanted flavors, meaning the player would have to work with what they had to remove them. This meant that when balancing, designers had to strategically place negative flavors, carefully plan recipe placements, and consider difficulty when assigning preferred flavors. These factors heavily contributed to how content would be introduced to the player at the start of gameplay.

The second goal, the controlled unlocking of ingredients, was necessary to gate flavors and recipes according to what the game balance needed, ensuring that concepts and challenges were introduced gradually.

Drinks were the first food group presented to the player, meaning only a small selection of plants needed to be available to the player: as such, the first two ingredients were planned to be simple berries that only provided two flavors. This decision was made to compel the player into interacting and learning about the core mechanics of flavor combination and cooking without additional complexities — complexities such as negative flavors, which would be introduced the next day, alongside a challenge whose success necessitated using them. As gameplay progressed, this trend would continue, as the need to incorporate or avoid more flavors was introduced with more and more flavors becoming readily available to the player through the ingredients they unlocked. With each day and each new wave of ingredients the player unlocked, new flavors and meal types became attainable, expanding the possibility space the player engaged with.

When the player gets to day 10, the final appliance is unlocked: the oven. One of their first meal requests is for a bun, which is part of the food group Bread, which has been unlocked with the oven. The bun is the simplest form of cooking bread: all it takes is putting batter in the oven. The player has learned how to make pancakes by putting batter on the stove, so this new challenge directly pulls from rules that they already know. It also sets them up for new types of breads: once they understand that putting an ingredient in the oven makes Bread, they can make the connection that putting other ingredients in the oven will make new forms.

Overall, tutorialization and balancing of the mechanics and resources in the game were essential to making sure that all gameplay systems flowed smoothly, and none of them overshadowed the others. Even after making these changes, adjustments continued up until the end of development, as we constantly tested and improved the effectiveness of these systems on real players.

5.4 - Playtesting

As the intention of the Beta phase of development was to create a version of the game representative of final gameplay systems and world, the bulk of testing occurred concurrently to the development throughout this period. Playtesting provided the team with an opportunity to observe how players responded to the game and its systems, empowering us to iterate and tweak the game's design throughout the Beta and Gold periods into launch — a fact that has been made readily apparent, given how many of the previous sections detailed the problems identified as part of this playtesting and the subsequent solutions carried out. As such, it is quite evident that these sessions, hosted through late C- and early D-Terms, were critical in gauging players' responses and reactions to gameplay. Being that this is the case, so too is it relevant to clarify how the team utilized this tool.

In order to provide a legitimate environment for these sessions, the team established a consistent procedure to test and evaluate players and their experiences; each of these playtest sessions employed a combination of observation, surveys, and interviews to gather meaningful data to answer a series of questions posed by the team. Live observations of players' gameplay — in which investigators were paired with players to record and note their experience with the game — was essential to gain an understanding of how new players interacted with the game; surveys enabled a easy collection of quantitative information, useful to characterize a player's time with the game; informal exit interviews, where playtesters were asked to answer open-ended questions, allowed us to collect essential context and information about their overall experience.

The earliest of these sessions focused on players' engagement and understanding of the expanded core loop — an essential aspect of any game's development. In this early build, the team and its designers had yet to nail down exactly which of the directions the core gameplay mechanics would expand into; as such, simplified iterations of many systems would be implemented and tested. The build for this playtesting session included simple chores such as bed cleaning, as well as timed interactables and the first iteration of unique recipes; the results from the session ended up being both informative and disheartening for the team.

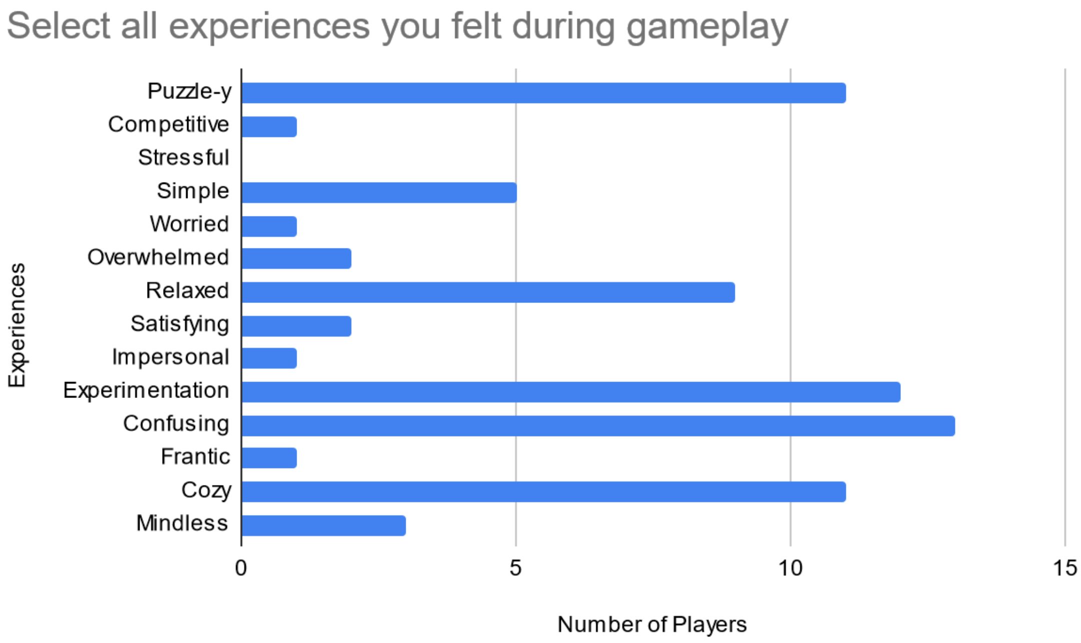


Figure 58: Responses to a survey question that asked players to select all player experiences that applied to them during the first playtest session.

This playtesting session made several things apparent to the team. Primarily, the lack of cohesion between gameplay systems, in combination with limited feedback and aid provided by the original UI, resulted in a game that the overwhelming majority of players classified as "Confusing" as seen in the graph above. Exit interviews clarified this with a vast collection of worrying developments: players struggled to identify what combinations made which foods (Section 5.3.4), couldn't recognize what was a positive or negative response from the birds they served (Section 5.3.3), and had difficulty navigating the scene overall (Section 5.2.3). Even more concerning was the fact that many of the players failed to recognize the context for the game, with the player's role as the manager of a bed and breakfast completely lost through gameplay (Section 5.2.4).

As a result of this feedback, the priorities of the team and its subteams shifted. Plans for future developments pivoted from new features, systems, and content to providing clearer feedback and context to the player through the gameplay mechanics, menus, and cutscenes. Programming and Design cut the implementation and plan for much of the expanded gameplay loop, eliminating timed systems and bed-cleaning chores for a clearer focus and balance of cooking with recipes (Section 5.3.1). Art and Design underwent several iterations to finalize the information conveyed through the UI and menus (Section 5.3.2) as well as the readability of the tree for gameplay (Section 5.2.3), just as Writing updated and expanded the different types of responses customers had for the meals they were served (Section 5.3.3).

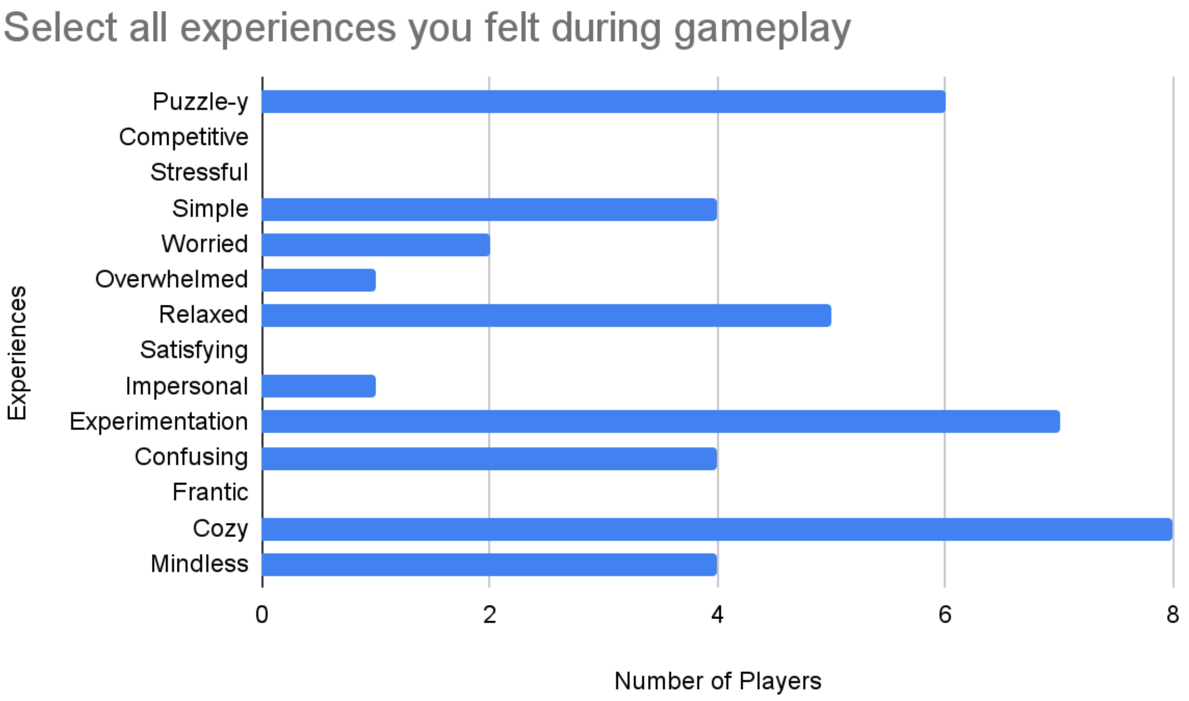


Figure 59: Responses to the same survey question as in Figure 58 during the third playtest session.

By the third playtesting session — which continued our focus on improved feedback, clearer interactions, and a refocused gameplay loop — the response would be significantly more positive, indicating a trend towards more polished gameplay systems and player understanding that the team would continue to pursue through launch.

Yet all this shifting and the iterations needed to make these improvements would come at a cost. Playtesting revealed just how much polish the game's core systems and world would need, and the energy the team committed to these parts of development meant future plans wouldn't be realized. As the team approached the end of Beta, it became clear that the final, complete version of the game would be a modified version of Layer 5: a layer that included cutscenes, gameplay, and scripted content for fifteen in-game days, but none of the team's more optimistic plans for the town, exploration phase of gameplay, quests, and foraging.

Cutting these future plans was undeniably a severe disappointment to the team — from the earliest vision of the game, exploring the wider town outside of the B&B and interacting with it through meaningful gameplay and dialogues had been a part of our plans (Section 1.2.1). Being without a town meant removing a large portion of gameplay; the entirety of the split between morning and afternoon sections wouldn't be possible (Section 3.2.3). However, it was a completely feasible and realistic cut to make to our game — and one that the layered design plan accounted for by stratifying town content and gameplay into Layer 6.

The important lesson the team took away from playtesting, plus the subsequent iterations and scope cuts it required, cannot be understated. Just as building up the core features and systems during Alpha had demanded an interdisciplinary approach, expanding and improving the game through Beta had required the same collaborative commitment across subteams. We also collectively decided what layering and scope cuts to make.

With the removal of the town and the accompanying afternoon section of gameplay, each of the subteams would see a reduction of work meant to lessen their workload as they took on additional tasks related to iterating on and expanding existing systems. Fewer locations and scenes would free up Art from additional environmental art, allowing for multiple UI overhauls; removing interactive foraging and shops meant Design wouldn't need to balance several other sources of ingredients, freeing them up to better balance and tutorialize the cooking system; significantly less dialogue would be needed from Writing to flesh out town characters, so they could rewrite meal responses and focus on cutscenes; and Programming would overall have less new content and features to implement, meaning they could work on adding and updating features based on the newest plans and iterations.

In this manner, playtesting served an important role. Were we to delay beyond Beta to playtest, the team might've been able to squeeze in several of the features and plans from their initial vision, but it would've come at the cost of being too late to fix up and iterate on the core systems and gameplay at the heart of the experience — something that needed to be done before the final, polish period of development was entered in the final weeks of production.

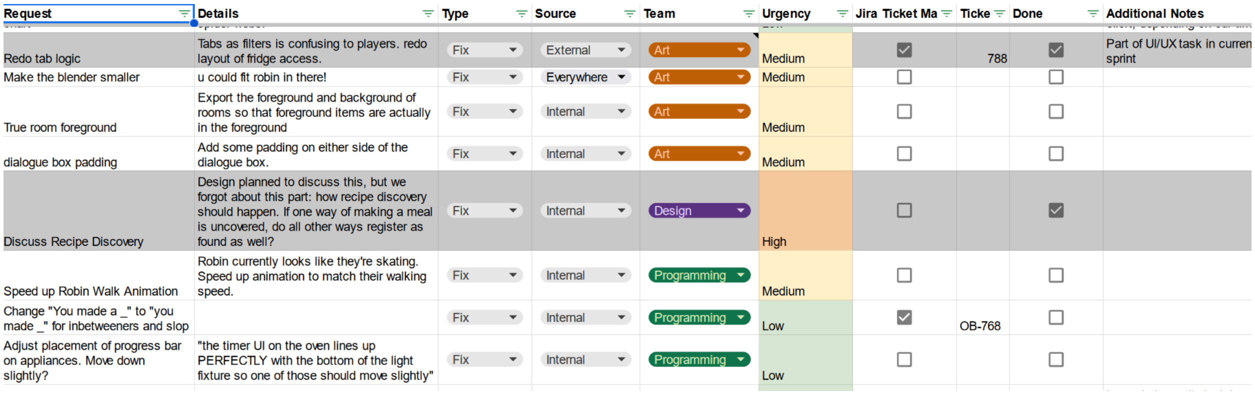
6 - Gold

For this chapter, the focus is turned towards the final phases of game development, and the team's experience polishing and refining the project for release. Miscellaneous fixes, content implementation, quality of life and improvements, and the final marketing materials necessary for launching a product became the primary focus as the team pivoted towards showcasing and demoing the game.

6.1 - Introduction

From the early excitement of ideation, to the solidification of ideas in pre-production and the subsequent development and iteration across the Alpha and Beta phases of the game, the team focused hard on executing the ideas and processes to reach completion. When C-term ended, the team was able to take a step back and review our creation. One thing jumped out: the game was *big;* at least, much bigger than it was when we initially created our team processes, documentation, and goals. It was also, as all games are during production, full of bugs, rough patches, and quality of life issues. Our means of operating and communicating tasks that needed to be completed no longer sufficed; things were inevitably going to start slipping through the cracks as we moved into the Polish phase of development. The backlog of Jira was filling up very quickly, and was missing vital organization criteria that we needed such as prioritization. A new tool was needed to fill this gap.

Subsequently, a spreadsheet was created to track all of the necessary – or even just desired – changes that came up when reviewing and playtesting the game. This spreadsheet could be added to by anybody on the team, and held a request's details, associated team, priority, and tracking information. All requests were reviewed by the production team to ensure team agreement on new features and correct prioritization.



Request	Details	Type	Source	Team	Urgency	Jira Ticket Ma	Ticke	Done	Additional Notes
Redo tab logic	Tabs as filters is confusing to players. redo layout of fridge access.	Fix	External	Art	Medium	☑	788	☑	Part of UI/UX task in curren sprint
Make the blender smaller	u could fit robin in there!	Fix	Everywhere	Art	Medium	□		□	
True room foreground	Export the foreground and background of rooms so that foreground items are actually in the foreground	Fix	Internal	Art	Medium	□		□	
dialogue box padding	Add some padding on either side of the dialogue box.	Fix	Internal	Art	Medium	□		□	
Discuss Recipe Discovery	Design planned to discuss this, but we forgot about this part: how recipe discovery should happen. If one way of making a meal is uncovered, do all other ways register as found as well?	Fix	Internal	Design	High	□		☑	
Speed up Robin Walk Animation	Robin currently looks like they're skating. Speed up animation to match their walking speed.	Fix	Internal	Programming	Medium	□		□	
Change "You made a _" to "you made _" for inbetweeners and slop		Fix	Internal	Programming	Low	☑	OB-768	□	
Adjust placement of progress bar on appliances. Move down slightly?	"the timer UI on the oven lines up PERFECTLY with the bottom of the light fixture so one of those should move slightly"	Fix	Internal	Programming	Low	□		□	

Figure 60: A portion of the Change Request Log.

This sheet was used to track a variety of different types of changes, with each discipline using it for different means. Writing's changes reflected the story, ensuring that enough information is presented to players in cutscenes, dialogue, and cinematics for them to understand the context and meaning of the game. Programming's attention was called to bugs and other issues, as well as quality-of-life changes to enable a better player experience when navigating the game world and menus. The art team, in addition to responding to story requests and UI updates, focused on making sure the game world looked consistent and all art pieces looked complete. Meanwhile, the marketing team worked around the issues present to create content showcasing the game as release rapidly approached.

6.2 - Adding the Garnish

The polish phase touched every aspect of the game: from the story to final balancing of ingredients to fixing the functionality of the game. Despite no longer focusing on features, the team still had to have all hands on deck to make sure every aspect of the game was polished to a shine.

6.2.1 - Final Story & Content

With drafts of the cutscenes already made, it was time to begin polishing them and preparing them for release. The Writing subteam came together and decided to polish our cutscenes in "Batch Revisions", with each writer revising the cutscenes they wrote based on feedback they received throughout the development process. As part of their Batch Revisions, writers also had to implement character actions and the visual changes that happened in their cutscenes. For this, the artists had to go through the cutscenes and determine what art assets could feasibly be made. This involved determining what the most important character portraits and cinematic backgrounds were for the game, and making notes for when some asset wasn't within scope.

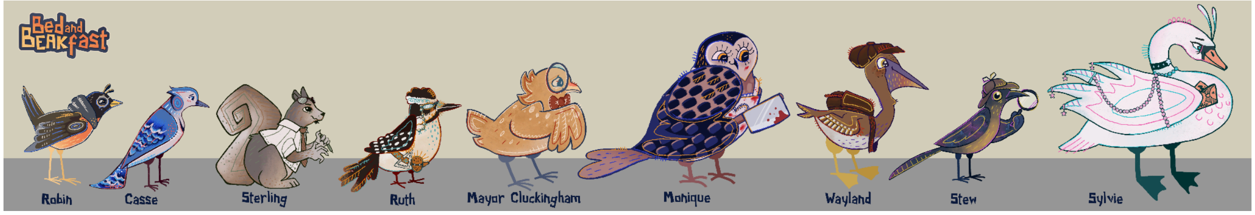


Figure 61: Photo of the game's Townie characters and Robin.

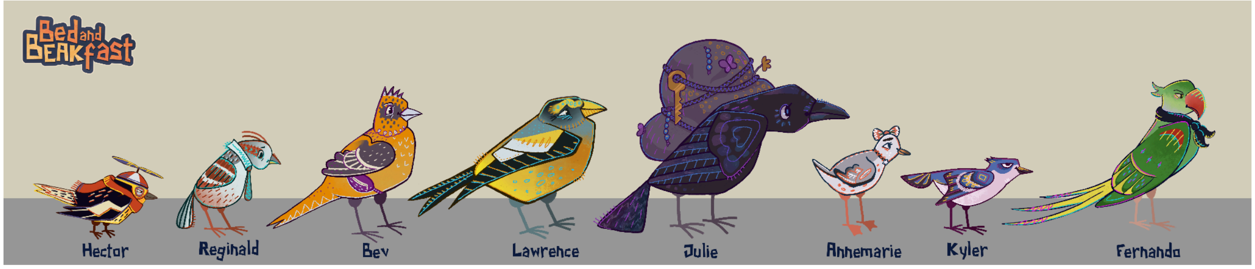


Figure 62: Photo of the game’s Traveler characters.

The artists and writers started talking about what characters would get portraits and which characters would have special emotion portraits. This was a critical discussion to have early on as character portraits would help convey both the tone of the game and the emotions that the characters were going through in the cutscenes. The game’s visual novel style dialogue also meant that each of the 17 characters (8 Travelers, 8 Townies, and Robin) that appeared and spoke in the game would need at least one portrait. The writers and artists determined that each character would have one portrait and that the main characters (Robin, Sterling, Casse, and Ruth) would have unique portraits displaying different emotions. The writers gave artists early drafts of the cutscenes so that the artists could determine what emotion portraits for the main characters were needed and if suggested emotion portraits could be combined to reduce scope.

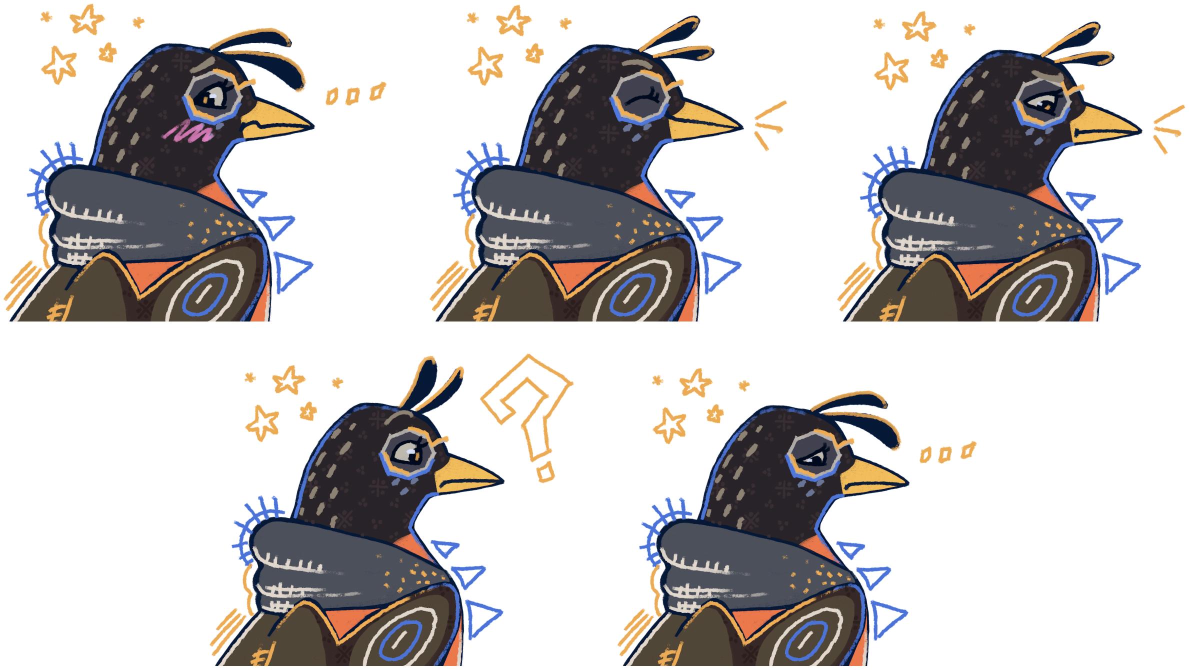


Figure 63: Emotion portraits of Robin (top left to bottom right: embarrassed, happy, determined, surprised, and sad).

Throughout this process, writers worked with the artists to readjust some cutscenes to reduce the assets needed. For instance, in the cutscene where Robin checks in on Ruth after her injury was originally meant to take place in Ruth's shop, however, it became evident that it wouldn't be possible for the artist to create the interior and exterior of the shop with the remaining time. Thus, the scene prior to that, which had Ruth quit and stomp away after being injured was changed to her grudgingly staying as she needed to complete her contracting job. Instead of treating this as a limitation, the writers treated these situations as ways to tighten up and enhance the existing story, with the updated cutscene helping to emphasize Ruth's inability to ask for help or admit when she can't do something.

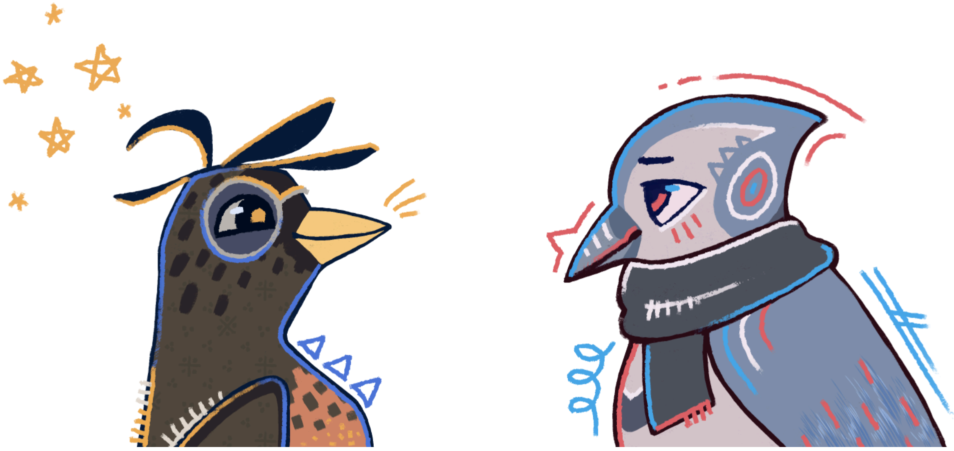


Figure 64: Portrait of a young Robin (left) and Casse (right).

Cinematics also faced cuts due to art assets. Initial plans for Casse's flashback cinematic would be displayed in more of a comic book format, with younger versions of Casse and Robin appearing on custom backgrounds and using speech bubbles to display dialogue. This was changed back to the visual novel format to reduce scope on both the artist and programmer ends, and to make the cinematic consistent with the cutscenes. Instead, the cinematic would have custom backgrounds, but would use portraits that displayed younger versions of Casse and Robin, which could be reused more easily in the cutscene and potential future flashback scenes.

Once the art asset scope was agreed upon, the writers implemented the various portraits and backgrounds into the cutscenes and did a final round of reviews. Final reviews were done to ensure the tone and character actions were consistent and that the story made sense with the updated scope.

6.2.2 - Quality of Life

Development was not exclusively focused on implementing features and crafting content (Section 6.1); fixing issues and improving the quality of life during gameplay was also a major priority, especially during the polishing phase. Due to the scope of our game, there were many issues that ranged from small but noticeable details to entirely frustrating interfaces. Given our small window of opportunity for polishing, this necessitated prioritization of the most effective and least time-consuming quality of life improvements. This judgment was performed in the Change Request Log by giving each idea a type (fix, feature, or juice), and then giving each one an urgency rating. The most urgent and critical fixes would be prioritized, especially if they prevented a game-breaking bug (Section 6.2.3). From there, feature improvements and juice additions would be considered if they would not cost too much development time.

One of our first feature improvement tasks was to make the cooking menu far more readable, yet also compact to facilitate displaying the cooking ticket alongside it. The process of creating that menu, as well as the new flavor icons that it used, was documented in its own dedicated section (Section 5.3.2). The focus on improving our menus went beyond just rearranging them, however. We also made moving between menus easier with the help of improved keyboard utility; while menus previously required the player to click the dedicated on-screen exit button to exit them, they could now be exited by pressing the escape key or whatever key was pressed to open them. In addition, the serving menu now only opens from the second time onwards that the player interacts with a customer, to reduce the friction of traversing around the tree and collecting meal orders.

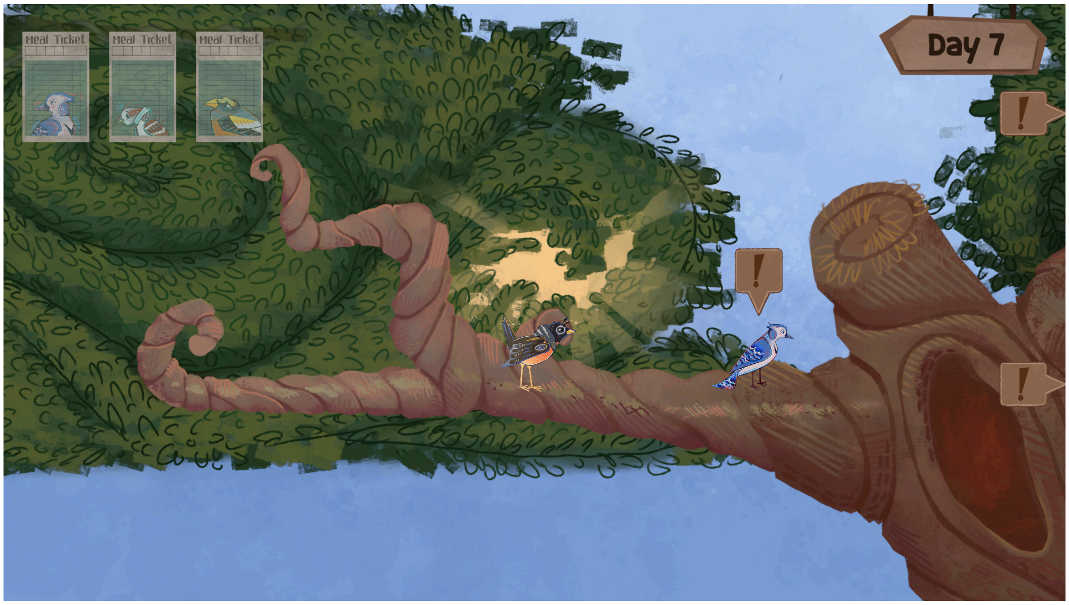


Figure 65: A mock-up of a trail on the kitchen branch of the tree. Also included is the day counter that was added during the polish phase of development.

Outside of the menus, there were and continue to be additional improvements to the movement gameplay. The tree proved to be challenging to traverse, so the places where the player could stand were changed so that the branches could be easily jumped between without having to fly. The walkable areas are also planned to be further highlighted through art changes that add a trail of wear marks. Another helpful change was a user interface element that tells the player which day they are on, to help the player feel a sense of progression and facilitate discussion of particular days. All of these changes were proposed to improve the game by adding on top of what was already relatively feature complete, but bug fixes are even more important to ensure that our game is at a satisfactory level of quality.

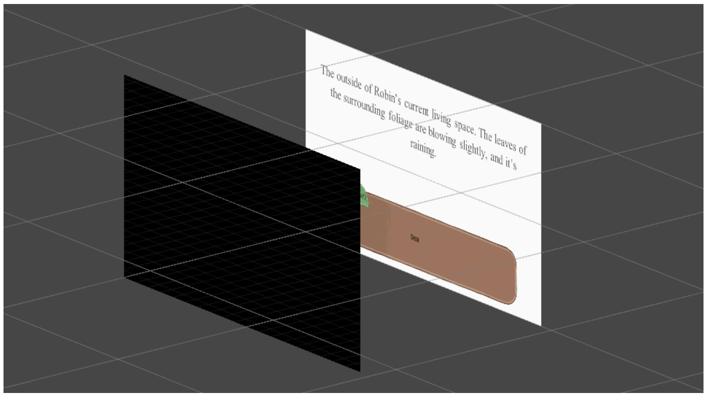
6.2.3 - Bugs & Fixes

Previously, we referenced the Programming subteam's use of code reviews and pull requests as "essential practice(s) employed by software developers to maintain a clean and (ideally) bug-free codebase" (Section 4.2.2). The reason for that "ideally" is quite simple, and one which has been quite elegantly surmised by *The Android Game Developer's Handbook*:

> "A bug free product is a myth." [56]

These bugs — unanticipated errors in software, such as games — are an inevitable byproduct of development; the more complicated the software, the more likely these bugs are. The nature of game development, working on a constantly evolving and expanding product, meant that by our Gold period, there were no shortage of bugs reported to the Change Request Log: bugs that would each take time to hunt down, recreate, and fix. This magnitude of bugs would by no means be an excuse to ignore the issues these software defects would cause our players; rather, few things interfere with a players' experience so much as the game behaving unexpectedly, or even crashing. In order to balance the demands of bug-fixing with the limited window before release, it was essential that the development team identify, isolate, and immolate those bugs that would most significantly impede the players' experience, ranging from game-breaking, to gameplay interfering, to purely cosmetic.

As the name would suggest, "game-breaking" bugs are the most detrimental bugs to the player experience; as such, they were prioritized first, as a matter of urgent attention. These bugs would often lead to crashes, where the game unexpectedly quit, or softlocks, which are problems with the game where the player had no choice but to restart it themselves. Although the programming team's use of code reviews had been helpful in minimizing the presence of these bugs, some unavoidably slipped through the cracks.



The outside of Robin's current living space. The leaves of the surrounding foliage are blowing slightly, and it's raining.

[56] Avisekhar Roy, *Android Game Developer's Handbook: Discover an All in One Handbook to Developing Immersive and Cross-Platform Android Games* (Birmingham, Mumbai: Packt Publishing, 2016).

Figure 66: The scene transition bug, visualized in 3D space. The black loading screen is still loaded, blocking the game view, even though regular gameplay content is loaded behind it.

One such instance of a game-breaking bug that required the team's attention during this phase involved transitions between scenes that were halted due to a timing issue involving the animations, randomly resulting in the player being unable to start a day of gameplay. Even though the next day of gameplay would be completely loaded and playable in the background, the black loading screen used to mask the transition between scenes would sometimes fail to fade back into transparent. Although a fix was easily implementable by the programming team, hunting down the bug meant testing and confirming that it wasn't coming from any number of different sources, a process called debugging; it took a bit of time to identify that the bug was caused by the animations with the loading screen and not the loading itself — after all, from a user's perspective, the game never even appeared to finish loading the next day.

The next set of bugs, of second rate urgency to be fixed, were those that interfered with gameplay in some manner or another. Although these issues wouldn't crash the game or directly force the player to restart, they would serve as a source of consternation and frustration, impeding players' enjoyment of the game.

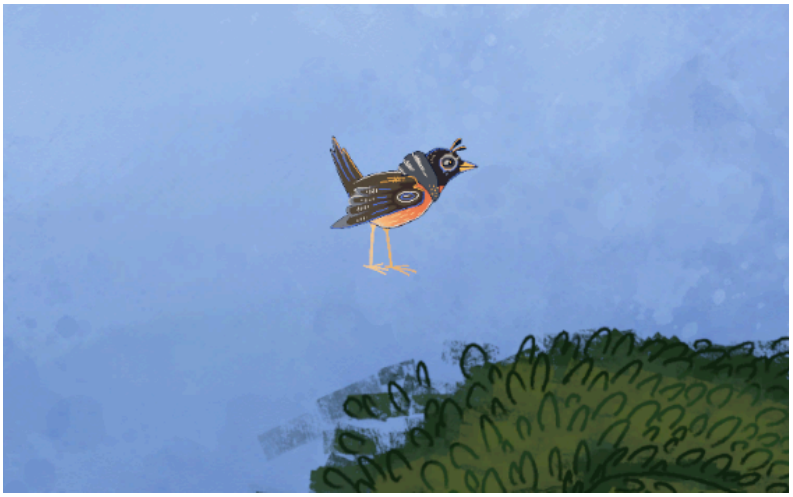
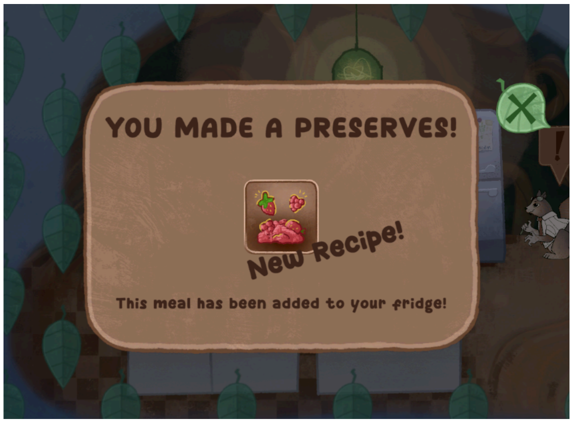


Figure 67: A bug related to player and world geometry which resulted in flight being interrupted and stopping in midair.

A good example of this bug type involved the collision and animations systems employed for player flight. From the first pass of flight animation (Section 4.3.3), issues had emerged involving how it handled specific collisions, such as steep inclines and areas where room and branch collision merged. In order to correct these issues, the programmers used the Gold phase to take another pass over the tree's collision geometry, eliminating instances of excessively deep or complicated geometry wherever possible, while also cleaning up the player character's collision detection systems.

Finally, the lowest priority of bugs the team focused on could be characterized as purely cosmetic, meaning that they had little to no impact on gameplay, but would still be likely to taint the user experience if left untreated. Sprite sorting issues, typos in dialogue, and stray artifacts in artwork all served as good examples for these types of bugs, so all subteams had their fair share of tasks to tackle regarding these issues.



YOU MADE A PRESERVES!

New Recipe!

This meal has been added to your fridge!

Figure 68: An example of the cosmetic bug created by the game's original dialogue implementation, which resulted in improper grammar rules when referring to foods.

One such bug that exemplified this categorization involved the flagrant disregard of grammar rules demonstrated by many of the foods. Although the original implementation of cooking and dialogue systems provided for any type of food to be created or ordered using the menus, this freedom was not reflected in the sentence structure used to identify these foods. Logically, it was perfectly normal for a customer to order "a granola bar" or "a fruit smoothie," but it was grammatically incorrect and oftentimes jarring that customers would order "a pancakes" or "a omelette," rather than the anticipated order of "pancakes" or "an omelette." While this didn't have any immediate issues interfering with gameplay, making it less of a priority, the common rate at which the problem was noted by early playtesters indicated that the issue was enough to take them out of the game, interrupting their experience. After dealing with our game breaking and gameplay interfering bugs, identifying and weighing the priority of these types of bugs would become a characteristic part of polish.

Given the limited timeline of development and the even shorter period which our Gold phase spanned, it was no surprise that the team wasn't able to identify or eradicate all of the bugs before the game launch; yet by recording the bugs our internal and external playtests revealed throughout development and then prioritizing their fixing during this final sprint of production, the team was able to ensure the majority of the polished game was free from those errors prone to most harshly impact the player experience.

6.2.4 - Release Material

To prepare for the impending release, the Marketing subteam created many release materials to present to our audience. For example, even though we did not yet have a fully polished game, it was important that we got a Steam page up as soon as possible. This was particularly important for showcase events, such as PAX East; for these events, we created marketing materials such as a postcard from the town of *Arboro* that contains a QR code to the Steam page, allowing people to wishlist the game.

Graphic	**Purpose**	**Dimensions**
Library Capsule	The game's main cover image on Steam.	600 x 900
Vertical Capsule	The game's sales page image on Steam.	374 x 448
Main Capsule	The game's home page image on Steam.	616 x 353
Library Hero	Shown at the top of a user's library details page for the game.	3840 x 1240
Header Capsule	Shown in the recommendations section on Steam.	460 x 215
Small Capsule	Shown in search results on Steam.	231 x 87
Main Capsules	Shown in a carousel on the main store page on Steam.	616 x 353

Table 2: The various required graphics & assets required for a Steam page.

The marketing team first got the Steam page up and running during C-Term. Many different graphics were needed in order to publish the page itself (see Table 2 for a full list); the team of two resorted to making temporary graphics using already finished art assets. For example, the vertical capsule featured sprites and the temporary title screen; these assets were also used for graphics such as the capsule header and library hero. The page also required screenshots of gameplay; these were easily continually updated until release as the team added new and more polished features.

During the D-Term polish phase, the Marketing subteam finalized the Steam page graphics. The library capsule and vertical capsule are two of the pages main displays; these function as the cover of our game. We wanted this screen to give potential players a quick look into the player experience of the game; we thus created a splash screen featuring Robin, the title, The Early Bird, and some ingredients and dishes that the player can cook. The main capsule and library hero are bigger, wider versions of the library and vertical capsule (Figure 69). To create these, the team just added slightly more details and featured more characters. The header capsule, small capsule, and main capsules are relatively small, but important graphics. Because of this, the team needed to make sure that the visual language was extremely clear and direct. These graphics contain the title as well as a few key game features. When put together, the graphics made for the Steam page communicate to the player both the aesthetic of the game and the type of game we have created.

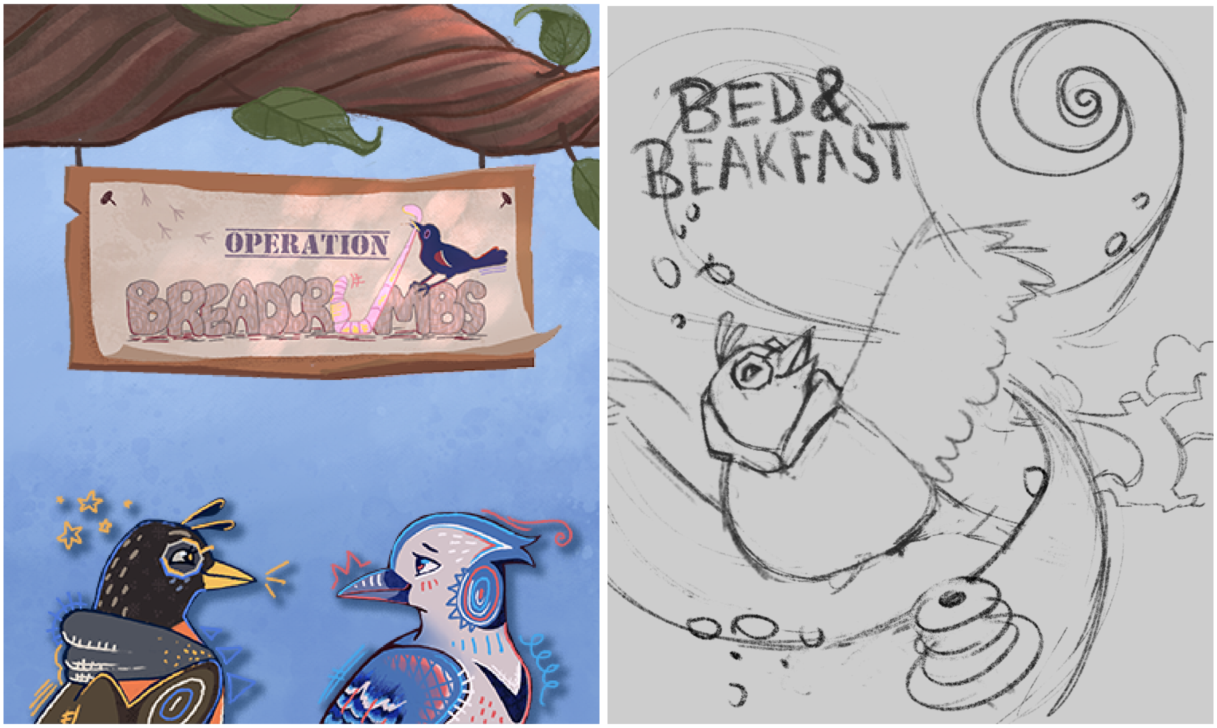


Figure 69: Old layout (left) and new layout (right) for the vertical capsule from the game's Steam page.

6.3 - PAX

As the team readied the game for release, it was important to keep in mind the perspective of its intended audience. We had been looking at our project for a very long time, and it was necessary to get it and its Steam page into the attention of players. Playtesting was helpful for identifying the necessary fixes and improvements required to get the game into its final state, but sitting players down to try the game and see if it was enjoyable was another story entirely. PAX East — a hugely popular convention celebrating recent and upcoming game releases attended by upwards of 60,000 people[57] — presented this opportunity.

In the final term of the year, the team was invited to show the game at the WPI IMGD booth at PAX East 2024 alongside various other student projects. In such a booth, event goers would be relatively consistently sitting down to play the game, seeking out marketing materials, and expecting a smooth gameplay experience on par with the large amount of professional games being presented at the event. It was an extremely important opportunity for the team, and presented a few challenges: deciding on the best way to present a part of a demo to players, and creating professional and compelling marketing materials to leave a lasting and physical impression on players.

[57] "Pax East," PAX East, accessed April 15, 2024, https://east.paxsite.com/.

Figure 70: The team at PAX East standing by the build of the game, minus one member.

When planning for the event, the team had to consider the best way to present the gameplay in a relatively short format, that was both easy to learn while still presenting enough challenge to be fun. The build was modeled after the most recent playtesting session: the first three days, each one short and tutorialized, were played first. The fourth day was presented as a jump forward, and the player was given many new ingredients and more complicated meal orders to simulate the challenge down the road.

Once the build was put together, the team had to plan our presence at the booth itself. The layout of the booth was up to the organizers, but we had the opportunity to visually represent the game in a poster and in handouts. The poster was designed carefully to show artwork from the game while still representing the theme of the start of the game: Robin's approach to the large tree. The postcards mentioned in the previous section were also printed for the event, with the understanding that similar postcards are very common to be handed out about professional indie titles. They marketed the aesthetic of the game well while also providing a link to more information about the game in a QR code on the back.

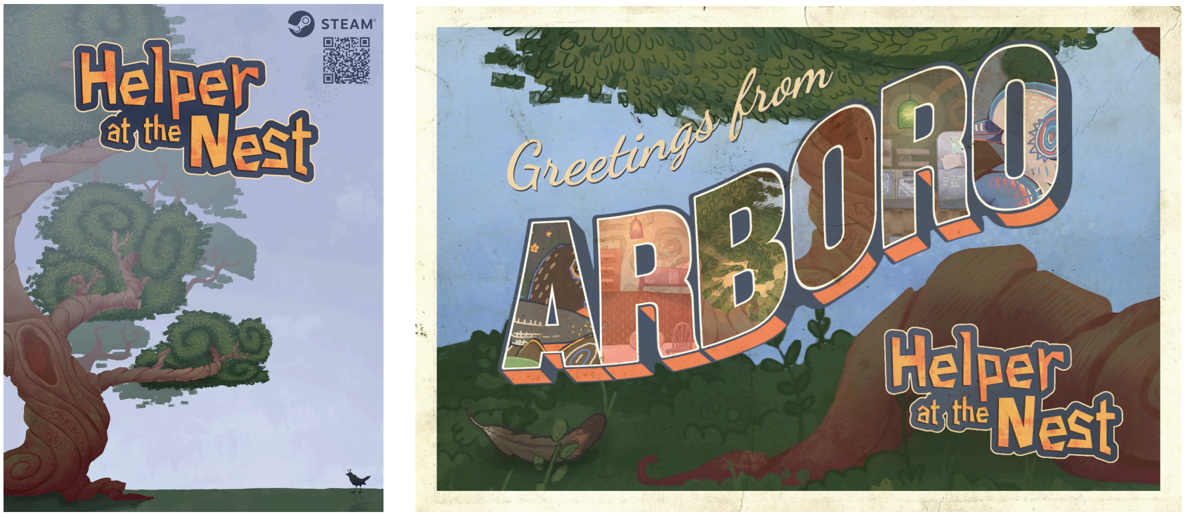


Figure 71: The poster (left) and the postcard handed out to players (right). Both feature an old title before the final one was decided.

This event was a great success for many different reasons. The game was well received, and so many postcards were taken that we needed to print another batch, which was then completely gone by the end of the event. Most importantly, the team was able to have the exciting experience of presenting the project that so many hours had gone into, and seeing it alongside professionally published games. It was a huge morale booster, giving the team extra motivation to push for completion of the game by the time of release.

7 - Retrospection

This final chapter provides the closing thoughts of the project team regarding the game, its development process, and wider context. Along with that, this section includes an analysis of the team's initial expectations, the lessons learned throughout development, and the final anecdotes relating to its release.

7.1 - Context

Bed and BEAKfast was initially pitched in 2023, with development occurring during 2023 and 2024. While the team was entrenched in the process however, the video game industry at large was suffering. From January 2023 to January 2024, 16,000 workers in video games had been laid off, with 6,000 being laid off in January alone as the trend continued well through March.[58] During the Covid-19 pandemic, there was a huge rise in video game demand.[59] As a result, video game companies expanded staff, took out loans, and made big investments and acquisitions.[60] In 2023, a slowing down in this growth contributed to many companies seeing mass layoffs despite record profits. Ultimately, the CEOs and higher ups in these companies continued to profit off of their own poor financial decisions and instead made those who put the physical work into the product to bear the consequences.[61]

As students who were soon going to be entering the industry themselves, this was terrifying. Much like Robin, we were already in the middle of that childhood-to-adulthood transition that occurs between the shift from college to corporate life. On top of that, we were watching our older peers, those who we looked up to with more experience, agonize through the continual layoffs and job drought. As these events unfolded, a heavy fear shrouded over the team and the projects' development; could what we had been working towards for four years be unobtainable? This demoralizing realization could have been the nail in the coffin for us — and this project. But, we didn't let it.

Ultimately, *Operation Breadcrumbs* blossomed from a group of six friends who loved to make games, and wanted to make something together. It was born of a passion for unbounded creativity, the love of creating fun experiences, the desire to share new perspectives with others, and of course, birds. Knowing that this was perhaps our last chance to make something entirely of our own creative control and with the institutional resources at our disposal, as well as the state of the industry that awaited us after, the team went all in. Instead of letting outside events

[58] Ted Litchfield and Wes Fenlon, "The Impact of 16,000 Games Industry Layoffs, in One Chart," PC Gamer, February 7, 2024, https://www.pcgamer.com/the-impact-of-16000-games-industry-layoffs-in-one-chart/.
[59] Nicole Carpenter, "What's Going on with All These Video Game Industry Layoffs?" Polygon, February 19, 2024, https://www.polygon.com/gaming/24074767/video-game-industry-layoffs-explainer.
[60] Tomas Franzese, "The Video Game Industry Flew Too Close to the Sun in 2023," Digital Trends, December 21, 2023, https://www.digitaltrends.com/gaming/game-industry-acquisitions-issues-2023/.
[61] Tomas Franzese, "What the Video Game Industry's Layoff Wave Means for You," Digital Trends, February 2, 2024, https://www.digitaltrends.com/gaming/game-developer-layoffs-january-2024/.

demotivate us, we poured all of our fear, anger, and sadness that we felt into making *Bed and BEAKfast* as great as it can be, turning it into the colorful celebration of community that it is now.

7.2 - Expectations

When comparing our original vision for *Bed and BEAKfast* to its final state, we occasionally felt disappointed or frustrated at the disparities between them: large swaths of the story were cut, entire systems got scrapped, and beloved characters had to be left behind. When phrased like this — with the knowledge of the lofty goals we had aspired to meet — it was easy to feel disappointed at what it could have been. And yet, the final product does not feel incomplete; rather, it is a small vertical slice of an enjoyable, polished, heartfelt game which achieves all its pillars.

Despite the pitfalls and speed bumps encountered along the way, the team did a remarkable job of creating something we can all be proud of. It is easy to look back and judge our own mistakes; our team's end-of-development meeting was filled with laughter and long sighs as we reviewed our naiveté from this time last year. While we'd like to think we should have known better, in reality no amount of warning from advisors or predecessors could have prevented us from making them. Those hurdles are simply part of the process for a student project, and the lessons cannot be learned except through doing. However, the remarkable part was our ability to shoot for the moon, while still creating plans for landing among the stars — and further plans still for gracefully falling to Earth. By creating the game in layers, we were able to iterate upon the game while maintaining a complete experience and avoid crashing in an unfinished state. Our capacity to plan, self-reflect, and pivot when needed was invaluable to this project as a whole, allowing us to adjust as needed. While these scope cuts were effective and practical, it was easy to become fixated on the list of unfinished tasks we were presented with after each layer, rather than celebrating the work and progress that we had made.

7.3 - Lessons

While each team member had created games before, this year-long project was a unique experience that taught us valuable lessons about the game development process. We learned the true value of pre-production, the importance of communication across disciplines, and the strength of the iterative development process. We hope that by acknowledging these lessons, they can be passed on to future teams, so that they may avoid these same mistakes and missteps.

1. **Pre-production is about establishing a shared vision.**

We knew coming into the project that pre-production was important; however, we didn't fully grasp just how much it would impact our project's development. Pre-production is the foundation upon which a project is built, establishing the precedent and protocol for development. It is also essential that each team member exits pre-production with a shared vision for the game they are making. In our initial plans for pre-production, this is something we overlooked.

When a team comes together, they come in with all of their past experiences and ideas for what the game is, many of which will differ. It took our team a while to realize that we weren't always on the same page, from our understanding of mutual aid to our ideas for the game's design. Part of the reason we weren't all on the same page is that we were well into pre-production before we finalized our game's core pillars. Without properly defined pillars and a shared understanding of why those pillars were important to the team, the original objectives we had established in pre-production were ineffective. It is essential to use pre-production to define pillars which guide the whole team and emphasize their goals throughout development. Doing so would help ensure that the team and its members are able to share a singular, cohesive vision across all parts of the project.

2. **Communicating across disciplines is critical.**

Issues with our team's interdisciplinary structure were a key factor in several of our early speed bumps. A lack of collaboration and communication across disciplines served as a point of contention and confusion during the earliest stages of development, when many disciplines were unaware of the work and decisions being made by the other subteams. Initially, our team overlooked this issue, assuming that everyone was in the loop as we communicated asynchronously and shared online documentation and updates. However, this meant that throughout pre-production the team was only truly updated on the status of subteams once a week during advisor meetings, when attention was called to these decisions in an official capacity.

This brings us to the next lesson we learned: you can't over communicate when working across disciplines. With a large team like ours, communication is vital and is something that the team, being made up of friends, took for granted. Subteams need to be able to communicate changes early and often to each other, especially those that are relevant to other groups' plans. To facilitate this, our team integrated weekly team meetings and update sessions into our regular schedule. These allowed for a consistent place for subteams to provide feedback, give updates, and touch base with the rest of the team's plans.

3. **Use the iterative development process to avoid overscoping.**

As six highly ambitious perfectionists, it was inevitable that we overscoped. It's incredibly easy to overestimate the amount of things you can achieve at the beginning of a game's development and lose sight of what you can reasonably do, especially as a student. The layers system compensated for our ambition by chunking up our goals into achievable blocks. Given the ambitions of the original game vision and concept, we would have run into issues without this solution.

Therefore, the final lesson learned by our team is quite clear: an iterative development process is essential. Using layers allowed us to be as ambitious as possible, while grounding our goals and providing a framework we could use when choosing our next tasks. Even from very early stages of development, we had a minimal viable product that qualified as a game; this fact alone alleviated much of the stresses of game development, ensuring the team could turn their attention to making that game good. With this, we were able to avoid the worst effects of overscoping, ensuring that we never sacrificed the core of our game.

7.4 - Release

At the end of this process — with the game finished as far as we could hope it to be, and the focus of the team turning towards polish and showcasing — the inevitable was soon upon us: the end of the project.

Even from its earliest conception, the goal of this project had been to release a game demo via Steam, a highly popular virtual marketplace for video games. In doing so, the team not only would have created a game worthy to be the final product of our MQP, but one that would serve as a legitimate, published piece of media that we could reference in the future as an example of our abilities. With the publishing fee paid and the store page filled with marketing materials, all that remains is the submission of our game — under a finalized title.

It is not wholly unconventional to lack a proper name late into development; after all, the Operation Breadcrumbs moniker was chosen as a codename for the project during its earliest stages of development, when we were still uncertain of what exactly the game would be. However, the various possible titles the team workshopped and tested before settling on the final name is, in retrospect, a humorous anecdote, perhaps indicative of the iterative development process itself.

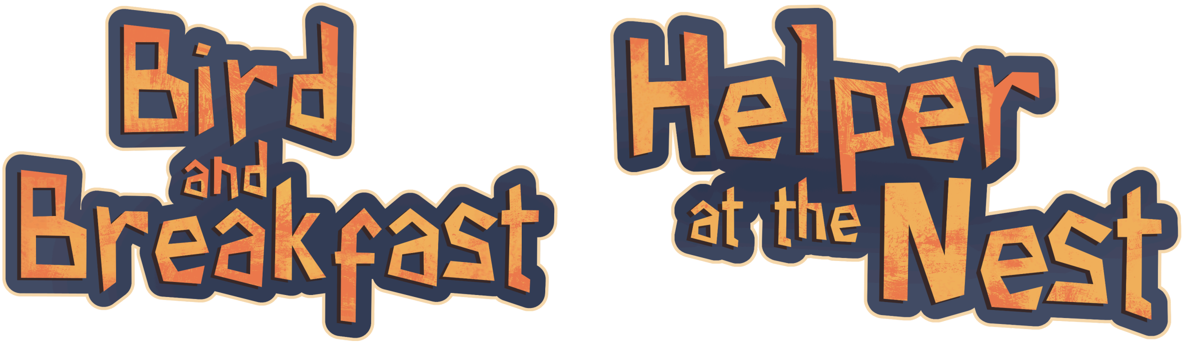


Figure 74: Different versions of the game's title and logos, while the name was being debated.

Bird and Breakfast was the original pun used to pitch the game, and one that the team stuck strongly to throughout development; concerns that it shared too much of a similarity to the published game *Bear and Breakfast* pushed the team away from the idea. Another title, *Helper at the Nest,* was pitched as a play on bird behavioral ecology, as the term "helpers at the nest" is used for birds who remain with their parents to help raise subsequent litters.[62] However, this title was decided to be too unfamiliar to a potential audience, given the lack of widespread adoption for the phrase.

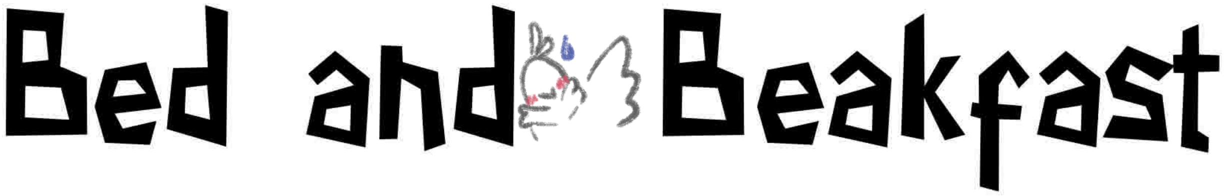


Figure 73: The internal announcement image used to reveal the final name of the game.

In the last weeks of development, this major hurdle would finally be overcome through the combined efforts of the whole team. Recognizing the weaknesses of their previously proposed titles, the team set out to create a title representative of the game's concept and themes while still being unique. The title *Bed and BEAKfast* was proposed and chosen by the team, serving as an alternative pun that still indicates this game is about a bed and breakfast for birds. Simultaneously, this title held narrative significance, in keeping with the story's themes; the imagery of the 'r' falling off a Bed and Breakfast sign being representative of the state of disrepair that Robin finds the B&B in at the start of the game.

In the wake of this decision, and the final touches of polish and release preparation that went alongside it, the project had reached its conclusion. Making the game had demanded a considerable amount of effort, along with significant collaboration between every discipline. It is with incredible pride that we wind down its development and move on to the future, now with the knowledge, skills, and memories we've gained during our time with *Operation Breadcrumbs*.

After Showfest and the final review of our game by the Steam marketplace, *Bed and BEAKfast* will be released in early May, 2024.

[62] Alexander F. Skutch, "Helpers at the Nest," *The Auk* 52, no. 3 (July 1935): 257–73, https://doi.org/10.2307/4077738.

References

Adam Robinson-Yu. *A Short Hike*. Adam Robinson-Yu. PC, Mac, Linux, Switch, PlayStation 4 or later, Xbox One or later. 2019.

Aesop, and Milo Winter. *The Aesop for Children*. Chicago: Rand McNally & Co., 1919.

Aggro Crab. *Another Crab's Treasure*. Aggro Crab. PC, Switch, PlayStation 5, Xbox Series X/S. 2024.

All About Birds. Accessed April 2, 2024. https://www.allaboutbirds.org/guide/.

Alsop, Fred. *Birds of North America*. New York, NY: Covent Garden Books, 2006.

Atlassian. Jira. Atlassian. PC. 2002.

Audubon. Accessed April 2, 2024. https://www.audubon.org/.

Basquiat, Jean-Michel. *State of the Art*. By Geoff Dunlop and Sandy Nairne. Illuminations, UK, January 11 1987.

Black Salt Games. *Dredge*. Team17. PC, Switch, PlayStation 4 or later, Xbox One or later. 2023.

Bond, Sarah E., and Joel Christensen. "The Man behind the Myth: Should We Question the Hero's Journey?" *Los Angeles Review of Books*, August 12, 2021. https://lareviewofbooks.org/article/the-man-behind-the-myth-should-we-question-the-heros-journey/.

Burke, Carolyn L., and Joby G. Copenhaver. "Animals as People in Children's Literature." *Language Arts* 81, no. 3 (January 2004): 205–13. https://doi.org/10.58680/la20042896.

ByteDance. CapCut. ByteDance. iOS. 2019.

Carpenter, Nicole. "What's Going on with All These Video Game Industry Layoffs?" Polygon, February 19, 2024, https://www.polygon.com/gaming/24074767/video-game-industry-layoffs-explainer.

ConcernedApe. *Stardew Valley*. V. 1.5. ConcernedApe. PC, Mac, Switch, PlayStation 4 or later, Xbox One or later, iOS, and Android. 2016.

Cotts, Josh. "The Wave of Stardew-Likes Is Finally Here (& They'll Probably Keep Coming)." CBR, October 20, 2023. https://www.cbr.com/stardew-valley-likes-are-finally-here/.

Crusenho. “Complete UI Essential Pack.” Asset Pack. *Itch.Io*. Itch. Accessed April 2, 2024. https://crusenho.itch.io/complete-ui-essential-pack.

Dandere-Hivemind, Review of *Potionomics*, by Voracious Games (2022). Steam (December 24, 2023). Accessed April 14, 2024. https://steamcommunity.com/profiles/76561198045738147/recommended/1874490/.

Daniele Giardini. DoTween. Demigiant. PC. 2014.

Dean, Jodi. “Capitalism Is the End of the World.” Mediations 33, no. 1–2 (2020). https://mediationsjournal.org/articles/end-of-world.

“Dialogue.” *Interface In Games.* Accessed April 2, 2024. https://interfaceingame.com/screenshots/animal-crossing-new-horizons-dialogue/.

Ellis, Kit (@kitosan). “*Chef’s kiss*” X, April 24, 2020, 10:45 AM. https://twitter.com/kitosan/status/1254058948019314695.

Fisher, Mark. *Capitalist Realism: Is There No Alternative?* Winchester, UK: Zer0 Books, 2022.

Franzese, Tomas. “The Video Game Industry Flew Too Close to the Sun in 2023.” Digital Trends, December 21, 2023. https://www.digitaltrends.com/gaming/game-industry-acquisitions-issues-2023/.

Franzese, Tomas. “What the Video Game Industry’s Layoff Wave Means for You.” Digital Trends, February 2, 2024. https://www.digitaltrends.com/gaming/game-developer-layoffs-january-2024/.

Fye (@FyeACNH). “Just Dom being Dom. Naming his abs. Lol. #AnimalCrossing #ACNH #NintendoSwitch.” X, June 20, 2020, 4:24 AM, https://twitter.com/FyeACNH/status/1274256848883822592.

Gummy Cat. *Bear and Breakfast*. Armor Games Studios. PC, Switch, PlayStation 4 or later. 2022.

Hibbett, Maia. “What Happens When Gentrification Comes to a Postindustrial City?” *The Nation*, January 30, 2019. https://www.thenation.com/article/archive/gentrification-worcester-urban-development-homelessness/.

Infinite Fall. *Night in the Woods*. Finji. PC, Mac, Linux, Switch, PlayStation 4 or later, Xbox One or later, iOS. 2017.

It's Happening. *PlateUp!*. Yogscast Games. PC, Switch, PlayStation 4 or later, Xbox One or later. 2022.

Javornik, Jennifer. "Alpha, Beta, Gold: A Commitment to High-Quality Game Development." Web log. *Filament Games* (blog), May 10, 2017. https://www.filamentgames.com/blog/alpha-beta-gold-commitment-high-quality-game-development/.

Jeffres, Leo W., Cheryl C. Bracken, Guowei Jian, and Mary F. Casey. "The Impact of Third Places on Community Quality of Life." *Applied Research in Quality of Life* 4*,* no. 4 (October 13, 2009): 333–45. https://doi.org/10.1007/s11482-009-9084-8.

JetBrains. Rider. JetBrains. PC. 2017.

Lim, Nick. "The Importance of Knowing Your Game's Target Audience." Sonamine, April 4, 2023. https://www.sonamine.com/blog/the-importance-of-knowing-your-games-target-audience.

Litchfield, Ted, and Wes Fenlon. "The Impact of 16,000 Games Industry Layoffs, in One Chart." PC Gamer, February 7, 2024. https://www.pcgamer.com/the-impact-of-16000-games-industry-layoffs-in-one-chart/.

Miller, R. Eric, and Murray E. Fowler. *Fowler's Zoo and Wild Animal Medicine: Volume 8*. St. Louis, MO: Elsevier, 2015.

Nintendo EPD. *Animal Crossing: New Horizons*. Nintendo. Switch. 2020.

Oshry, Dave (@DaveOshry). "THE YELLOW PAINT VIRUS HAS INFECTED FF7." X, February 7, 2024, 9:52 PM. https://x.com/DaveOshry/status/1755469586999505137?s=20.

Orwell, George. *Animal Farm*. London, England: Secker and Warburg, 1945.

Parker, Billie Jo. "The Player Menu: Everything You Need in One Spot." Web log. *Stardew Valley Blog* (blog), March 4, 2021. https://blogs.uww.edu/stardewvalleyblog/2021/03/04/the-player-menu-everything-you-need-in-one-spot/.

"Pax East." PAX East. Accessed April 15, 2024. https://east.paxsite.com/.

Pears, Max. "Design Pillars – the Core of Your Game." Web log. *Game Developer* (blog). Informa PLC, October 12, 2017. https://www.gamedeveloper.com/design/design-pillars-the-core-of-your-game.

Rappole, John Hilton. *Bird Migration: A New Understanding*. Baltimore: Johns Hopkins University Press, 2022.

Rawitsch, Don, Bill Heinemann, and Paul Dillenberger. *The Oregon Trail*. Minnesota Educational Computing Consortium. HP 2100 Minicomputer, CDC Cyber 70/73-26 Mainframe Computer. 1975.

Roy, Avisekhar. *Android Game Developer's Handbook: Discover an All in One Handbook to Developing Immersive and Cross-platform Android Games*. Birmingham, Mumbai: Packt Publishing, 2016.

Shaw, Kate. "Gentrification: What It Is, Why It Is, and What Can Be Done about It." *Geography Compass* 2, no. 5 (September 2008): 1697–1728. https://doi.org/10.1111/j.1749-8198.2008.00156.x.

Skutch, Alexander F. "Helpers at the Nest." *The Auk* 52, no. 3 (July 1935): 257–73. https://doi.org/10.2307/4077738.

Snoozy Kazoo. *Turnip Boy Commits Tax Evasion*. Graffiti Games. PC, Mac, Linux, Switch, Xbox One or later, iOS, Android, PlayStation 4 or later. 2021.

Spade, Dean. "Solidarity Not Charity: Mutual Aid for Mobilization and Survival." *Social Text* 38, no. 1 (March 1, 2020): 136. https://doi.org/10.1215/01642472-7971139.

Thunder Lotus Games. *Spiritfarer.* Thunder Lotus Games. PC, Mac, Switch, PlayStation 4 or later, and Xbox One or later. 2020.

Totten, Christopher. *An Architectural Approach to Level Design: Second Edition*. New York: A K Peters/CRC Press, 2019.

Voracious Games. *Potionomics*, Xseed Games. PC. 2022.

"What Is Mutual Aid?" Mutual Aid. Accessed March 16, 2024. https://www.mutualaid.coop/what-is-mutual-aid/.

"What Is Scrum?" Scrum.org. Accessed April 2, 2024. https://www.scrum.org/resources/what-scrum-module.

Yarn Spinner. Yarn Spinner. Yarn Spinner. PC. 2015.

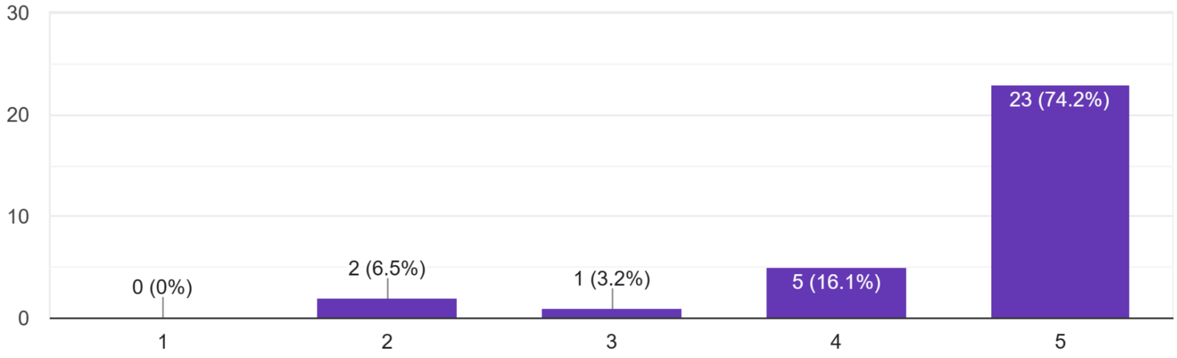
"Yellow Warbler." All About Birds. Accessed March 24, 2024. https://www.allaboutbirds.org/guide/Yellow_Warbler/maps-range.

Appendix

Appendix A: Alphafest Survey

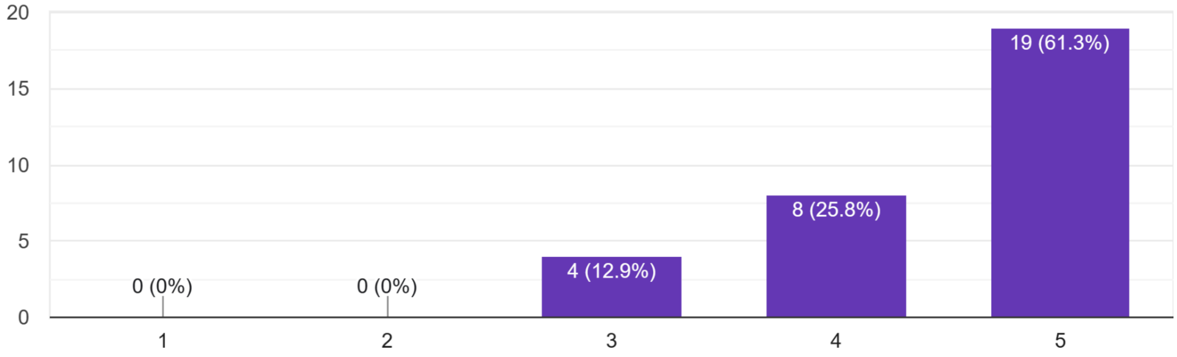
During the playthrough, I understood that I was playing as a bird making food for other birds.

31 responses



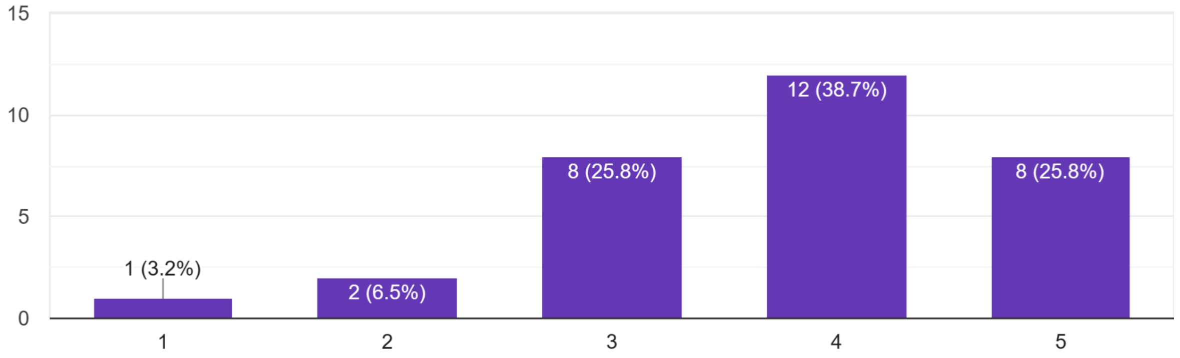
I would be able to explain the flavor system to someone else.

31 responses



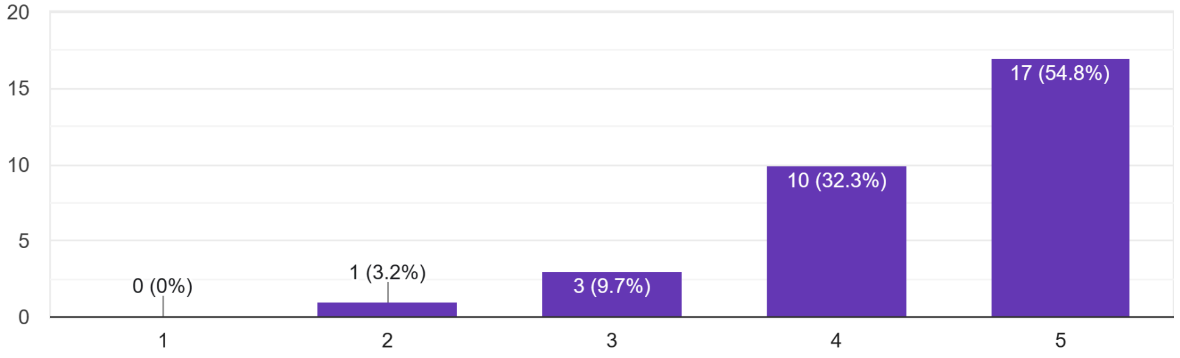
I would be able to explain how the flavor system affects a player’s score to someone else.

31 responses



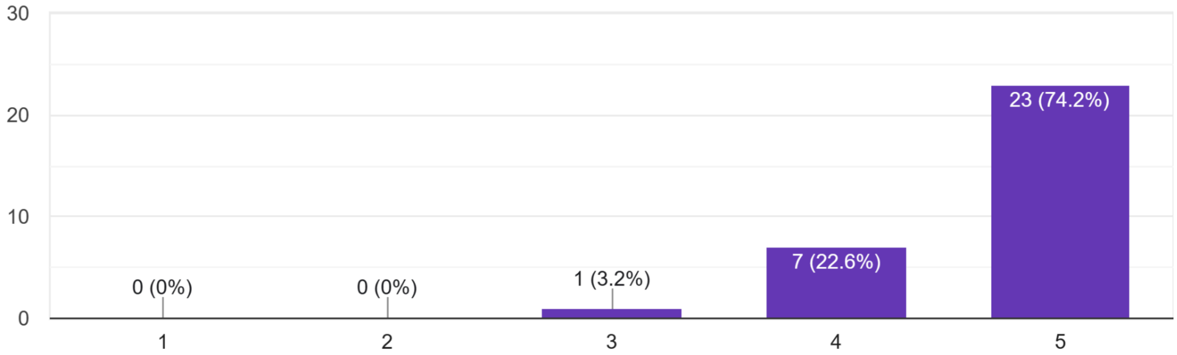
I would be able to explain the flavor graph (the radar graph/chart) to someone else.

31 responses



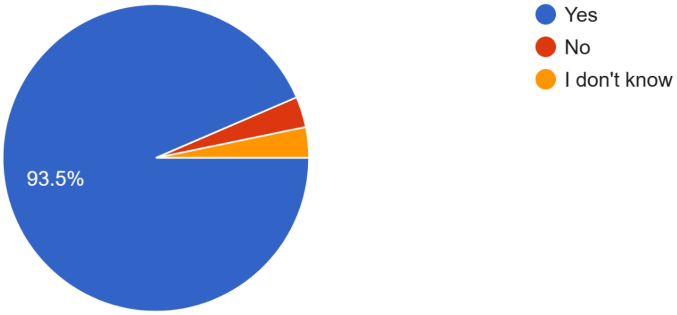
It was clear which objects were interactable.

31 responses



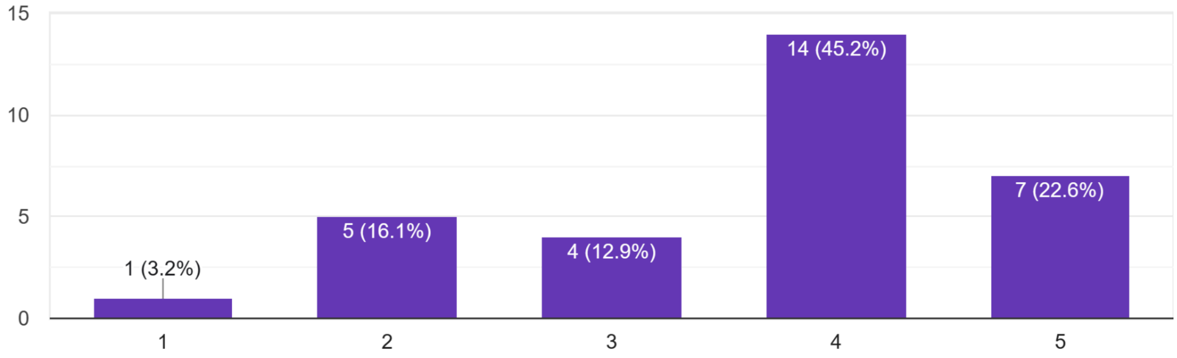
During your playthrough, did you complete the whole day?

31 responses



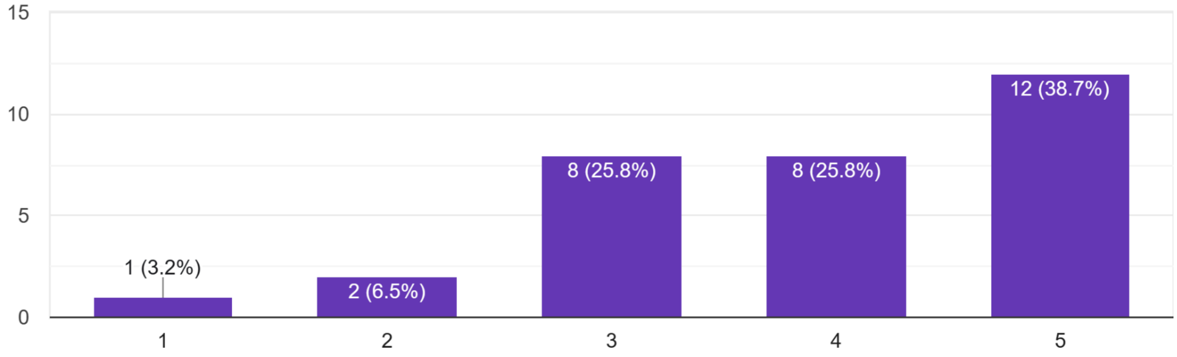
During my playthrough, I was focused on my score.

31 responses



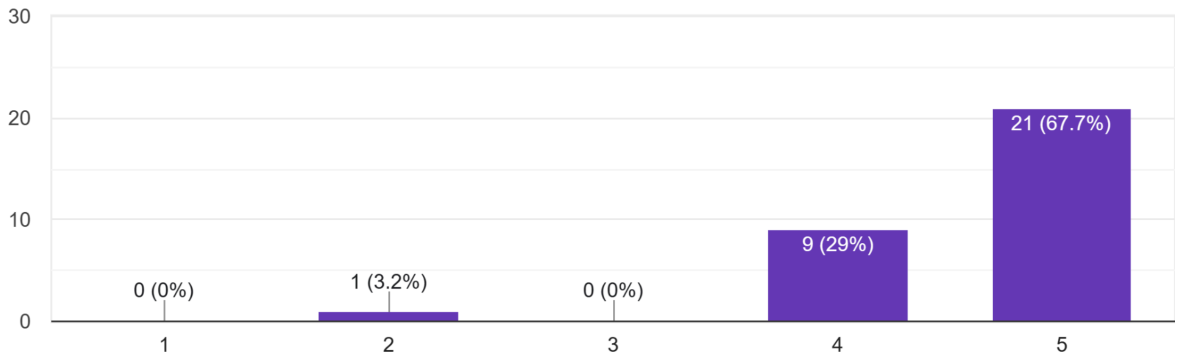
During my playthrough, I was focused on the quality of my meals.

31 responses



I would want to play a more complete version of this game.

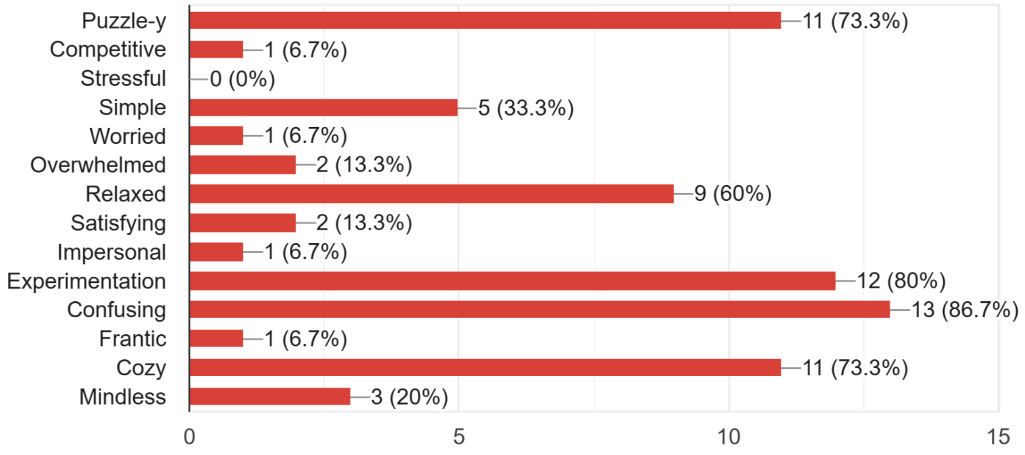
31 responses



Appendix B: Playtesting Survey 2/14

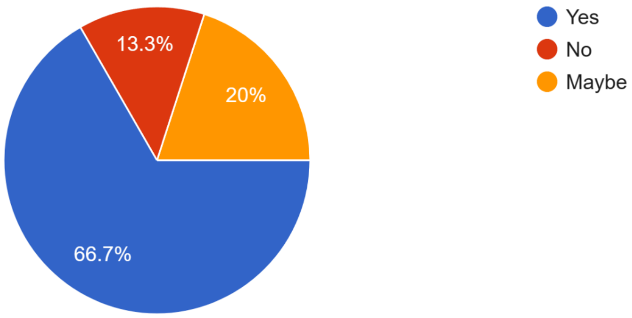
Select all experiences you felt during gameplay

15 responses

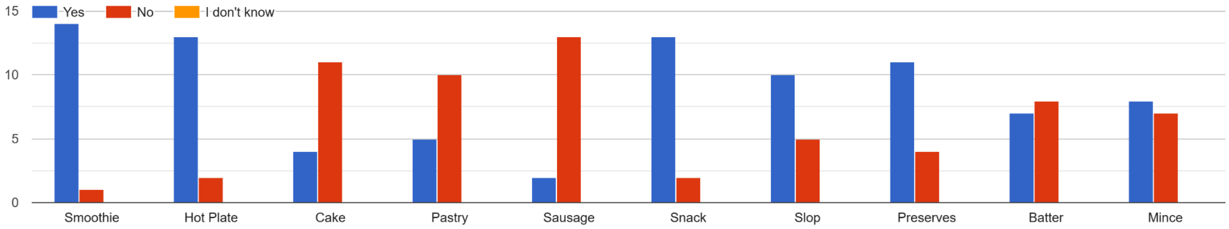


Would you want to play more days if given the chance?

15 responses

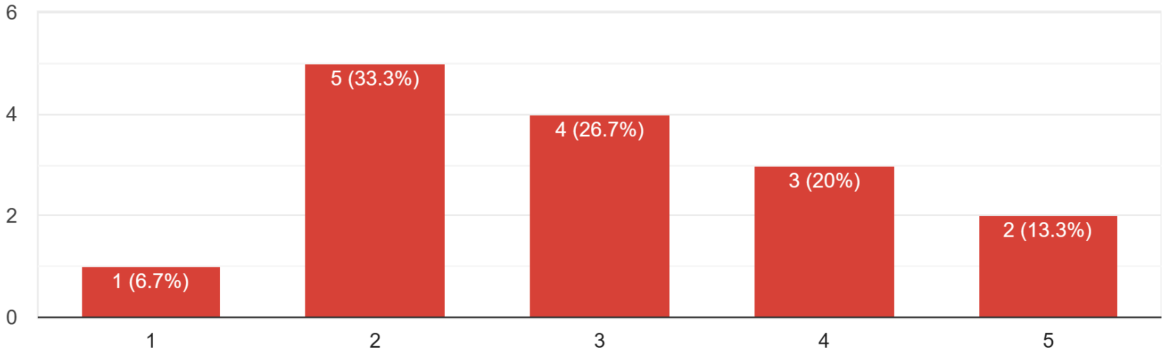


Did you make any of the following types of food?



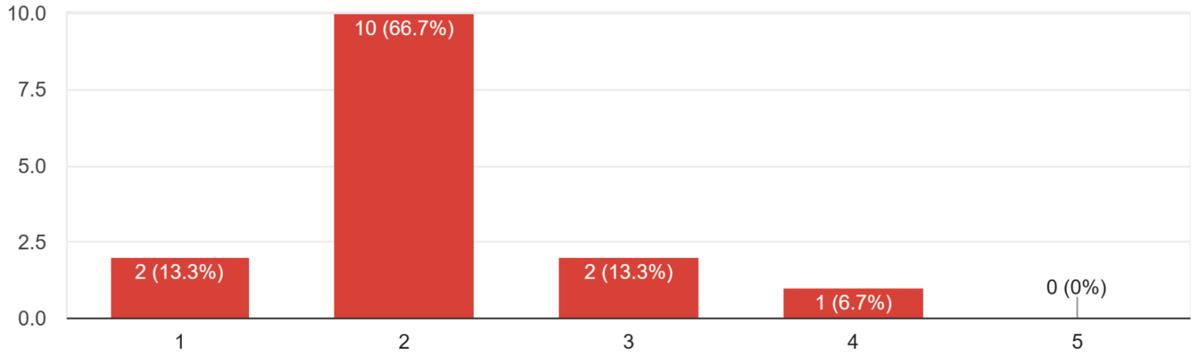
How well were you able to manipulate flavors in your cooking?

15 responses



How well were you able to control the types of foods you were cooking?

15 responses



Appendix C: Project Presentation Day Poster

Bed and Beakfast

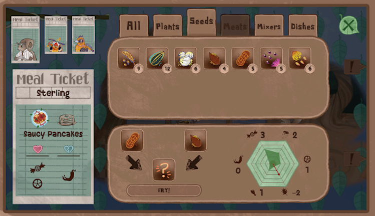
STEAM

The Operation Breadcrumbs team has worked diligently to create a video game as its MQP. Our game, Bed and Beakfast, is a management, cooking, and narrative game that features anthropomorphic birds at a bed and breakfast.

Players control Robin, a young bird who has returned to their hometown to find their late grandfather's B&B has fallen into disrepair. With the guidance of a helpful but suspicious squirrel named Sterling, Robin works hard to master the art of cooking and improve the B&B; follow their journey as they meet other birds, learn from their past mistakes, and grow beyond the nostalgia of childhood.

Gameplay revolves around cooking and discovering how to make meals for the various feathery customers that visit Robin's B&B. Fly around the tree which houses the B&B to talk with these customers. Interactions, cutscenes, and cinematics that play across the 15 days of this game's demo begin to tell Robin's story using a robust dialogue system.

A playable demo for Bed and Beakfast will be released on Steam in the coming weeks.



Cook Foods!

Cooking involves two interwoven mechanics: flavors and recipes. When the player cooks a meal, the ingredients they add determine the flavor qualities of the dish. Every customer in the game has a unique set of flavor preferences, and a threshold that a meal must meet in order to satisfy the customer's palette and request.

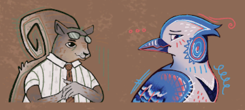
Different types of foods are created based on their recipes, which are determined by the types of ingredients added and the appliance used when cooking. The selected recipe is chosen using a search algorithm that prioritizes recipes that use more unique ingredients; in total, the game has over 60 unique foods!

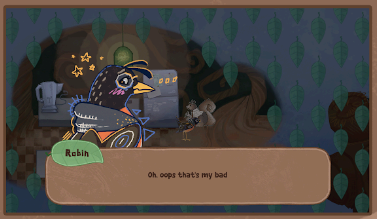
The Dialogue System is made to handle various types of scenes, including interactions, cutscenes, and cinematics.

Normal interactions involve selecting random dialogue lines for a customer to say based on the current game state and the player's actions. Customers pull lines from assigned pools of dialogue, where each pool corresponds to a specific personality or character.

Cutscenes and cinematics are scripted dialogue interactions in the game. These dialogue types involve event scripting, information loading, character portraits, animations, unique backgrounds, and more.

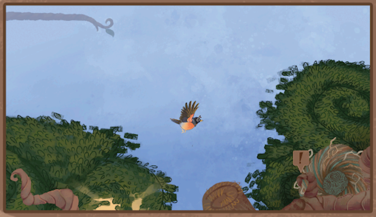
Meet Birds!





Robin

Oh, oops that's my bad



Take Flight!

The player has a diverse set of animations that they use while traversing the B&B. From walking to jumping to flapping their wings, a state machine controls the logic behind these animations and their transitions, working in pair with the Unity animator and the player controller.

The player can seamlessly transition between walking, jumping, flapping, gliding, falling, and landing, all with transitionary animations between those states. Combined with the player's movement physics, Robin is an expressive and enjoyable character to play as.

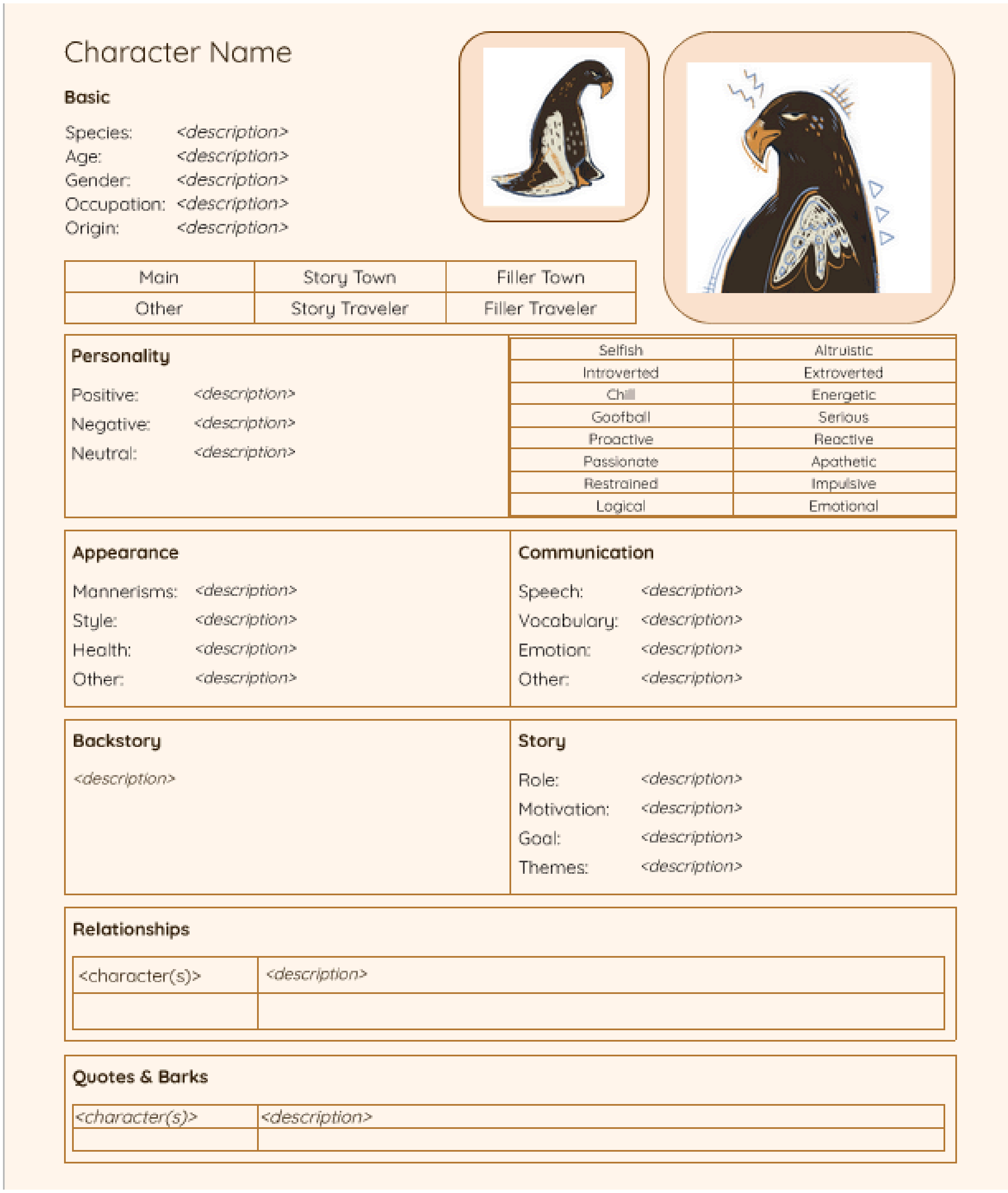
CS / IMGD (BS)
Tate Donnelly
Nicholas Frangie
Jade McEvoy

IMGD (BA)
Skye Pritchard
Abigail Rauch
Kerri Thornton

Advisors
Adryen Gonzalez
Ben Schneider
Gillian Smith

ISPs
Ethan Chau
Alistair Gilmour
Teddy Ker
Jess Liano
Camille Prats

Appendix D: Writing Subteam’s Character Sheet Template



Character Name

Basic

Species: <description>
Age: <description>
Gender: <description>
Occupation: <description>
Origin: <description>

Main	Story Town	Filler Town
Other	Story Traveler	Filler Traveler

Personality

Positive: <description>
Negative: <description>
Neutral: <description>

Selfish	Altruistic
Introverted	Extroverted
Chill	Energetic
Goofball	Serious
Proactive	Reactive
Passionate	Apathetic
Restrained	Impulsive
Logical	Emotional

Appearance

Mannerisms: <description>
Style: <description>
Health: <description>
Other: <description>

Communication

Speech: <description>
Vocabulary: <description>
Emotion: <description>
Other: <description>

Backstory

<description>

Story

Role: <description>
Motivation: <description>
Goal: <description>
Themes: <description>

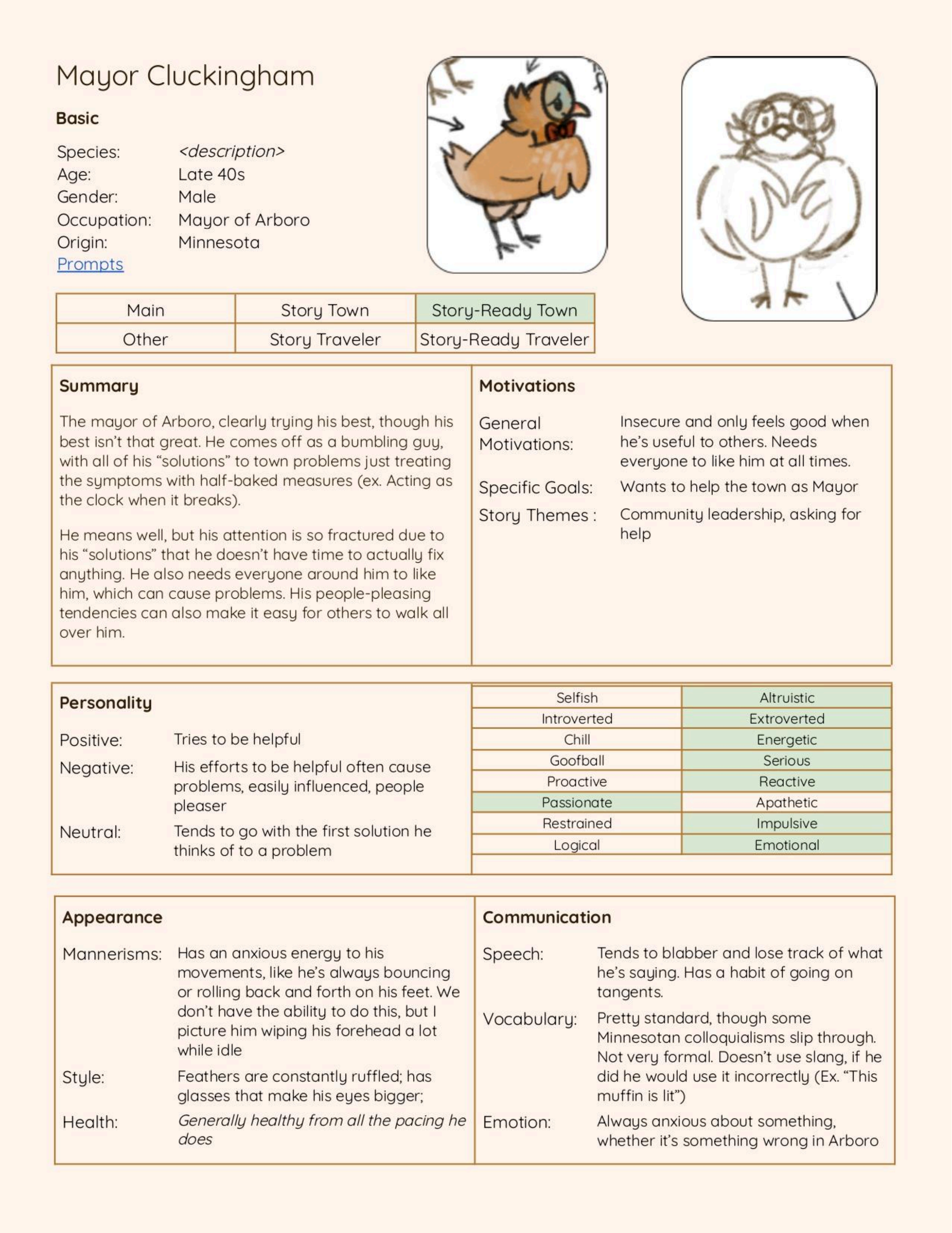
Relationships

<character(s)>	<description>

Quotes & Barks

<character(s)>	<description>

Appendix E: Mayor Cluckingham Character Sheet



Mayor Cluckingham

Basic

Species:	*<description>*
Age:	Late 40s
Gender:	Male
Occupation:	Mayor of Arboro
Origin:	Minnesota

Prompts

Main	Story Town	[✓ Story-Ready Town]
Other	Story Traveler	Story-Ready Traveler

Summary

The mayor of Arboro, clearly trying his best, though his best isn't that great. He comes off as a bumbling guy, with all of his "solutions" to town problems just treating the symptoms with half-baked measures (ex. Acting as the clock when it breaks).

He means well, but his attention is so fractured due to his "solutions" that he doesn't have time to actually fix anything. He also needs everyone around him to like him, which can cause problems. His people-pleasing tendencies can also make it easy for others to walk all over him.

Motivations

General Motivations: Insecure and only feels good when he's useful to others. Needs everyone to like him at all times.

Specific Goals: Wants to help the town as Mayor

Story Themes : Community leadership, asking for help

Personality

Positive: Tries to be helpful

Negative: His efforts to be helpful often cause problems, easily influenced, people pleaser

Neutral: Tends to go with the first solution he thinks of to a problem

Selfish	[✓ Altruistic]
Introverted	[✓ Extroverted]
Chill	[✓ Energetic]
Goofball	[✓ Serious]
Proactive	[✓ Reactive]
[✓ Passionate]	Apathetic
Restrained	[✓ Impulsive]
Logical	[✓ Emotional]

Appearance

Mannerisms: Has an anxious energy to his movements, like he's always bouncing or rolling back and forth on his feet. We don't have the ability to do this, but I picture him wiping his forehead a lot while idle

Style: Feathers are constantly ruffled; has glasses that make his eyes bigger;

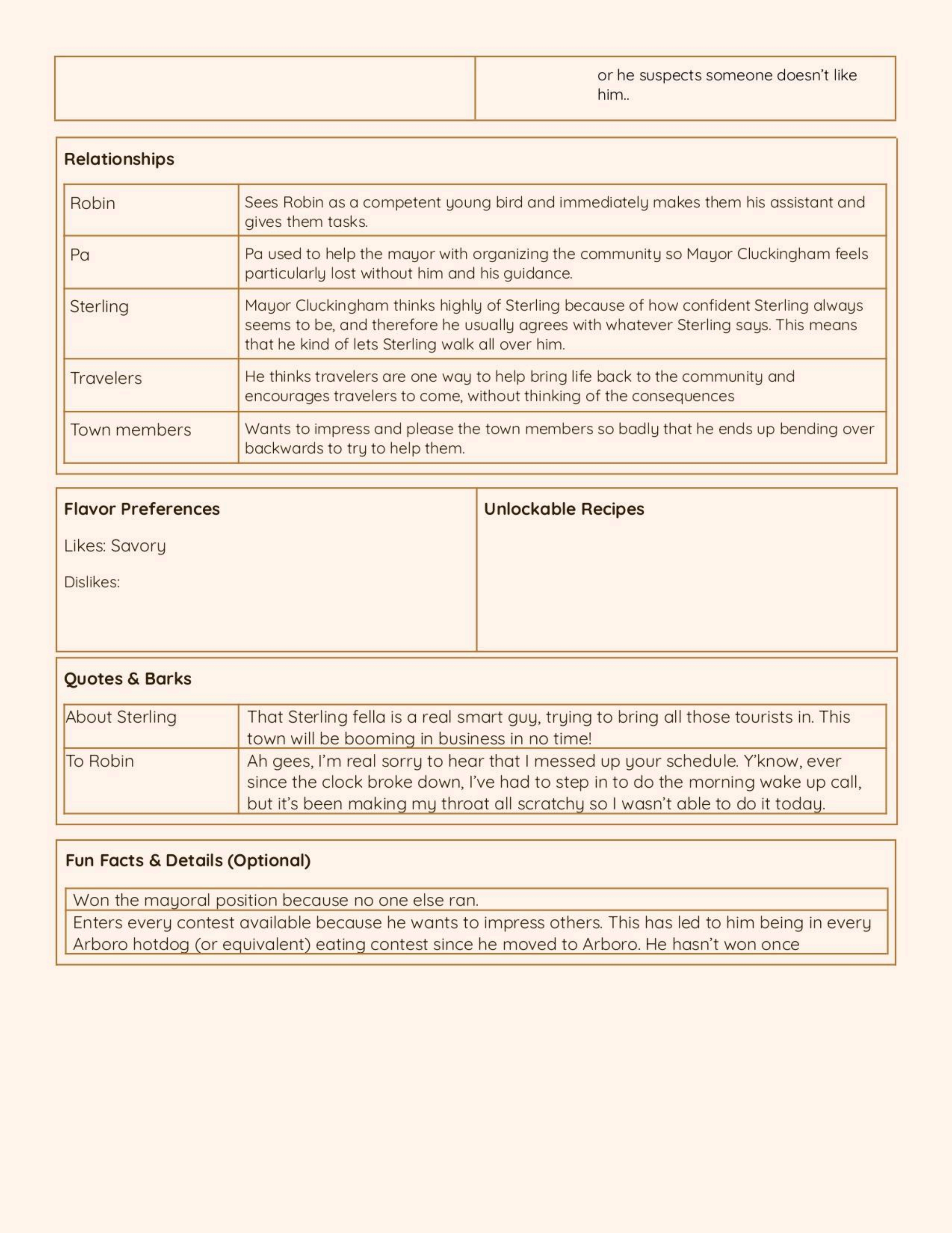
Health: *Generally healthy from all the pacing he does*

Communication

Speech: Tends to blabber and lose track of what he's saying. Has a habit of going on tangents.

Vocabulary: Pretty standard, though some Minnesotan colloquialisms slip through. Not very formal. Doesn't use slang, if he did he would use it incorrectly (Ex. "This muffin is lit")

Emotion: Always anxious about something, whether it's something wrong in Arboro



	or he suspects someone doesn't like him..

Relationships

Robin	Sees Robin as a competent young bird and immediately makes them his assistant and gives them tasks.
Pa	Pa used to help the mayor with organizing the community so Mayor Cluckingham feels particularly lost without him and his guidance.
Sterling	Mayor Cluckingham thinks highly of Sterling because of how confident Sterling always seems to be, and therefore he usually agrees with whatever Sterling says. This means that he kind of lets Sterling walk all over him.
Travelers	He thinks travelers are one way to help bring life back to the community and encourages travelers to come, without thinking of the consequences
Town members	Wants to impress and please the town members so badly that he ends up bending over backwards to try to help them.

Flavor Preferences	**Unlockable Recipes**
Likes: Savory Dislikes:	

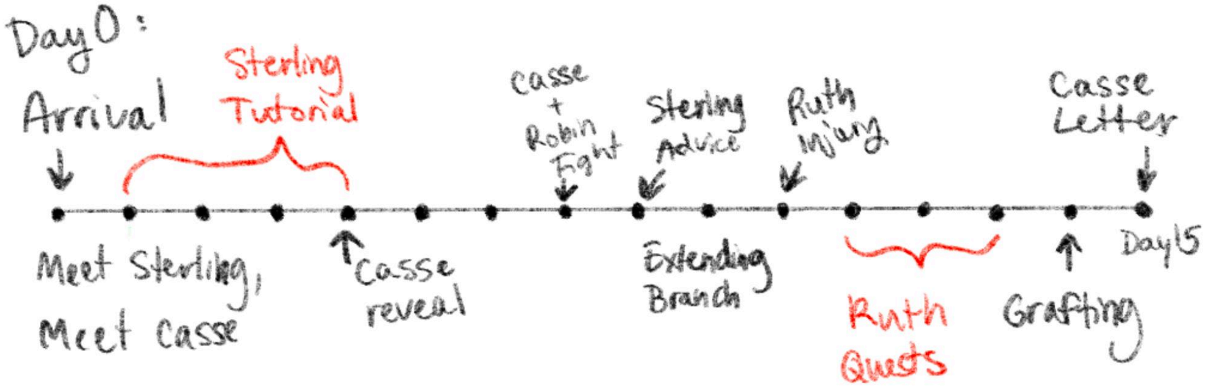
Quotes & Barks

About Sterling	That Sterling fella is a real smart guy, trying to bring all those tourists in. This town will be booming in business in no time!
To Robin	Ah gees, I'm real sorry to hear that I messed up your schedule. Y'know, ever since the clock broke down, I've had to step in to do the morning wake up call, but it's been making my throat all scratchy so I wasn't able to do it today.

Fun Facts & Details (Optional)

Won the mayoral position because no one else ran.
Enters every contest available because he wants to impress others. This has led to him being in every Arboro hotdog (or equivalent) eating contest since he moved to Arboro. He hasn't won once

Appendix F: Narrative Diagram



Appendix G: Excerpts from Animation Guides

Walk Animation Guide

This guide will discuss how walk animations are made for characters. Please note that this is a reference guide and not a rulebook – it assumes the reader has enough familiarity with animation to know when the rules can be broken, and is familiar with walk cycles in general and with animation in After Effects (AE).

Summary

- Walk cycles are on average 24 frames long (1 second)
- Walk should loop seamlessly and match character's personality
- All body movement is animated in AE; legs are animated in CSP and composited underneath body movement
- Standard walk cycle can be re-colored and reused for other birds, provided it fits their body type; otherwise, use it for timing and draw over it
- This guide is split into **TWO main parts** – the After Effects side, and the Clip Studio Paint side.

Assets Provided

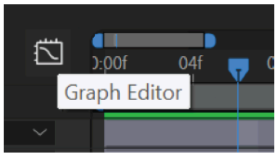
- Character's body pieces; these can be found in the Sprites folder. You should have **already** set up the AE file in the Pivot Points and Parenting Guide, so you won't need to add any more sprite pieces in.
- Standard walk (.mp4 version and 24 .png frames, of which 13 are unique)
- Any other previously made leg animations – may be used as reference

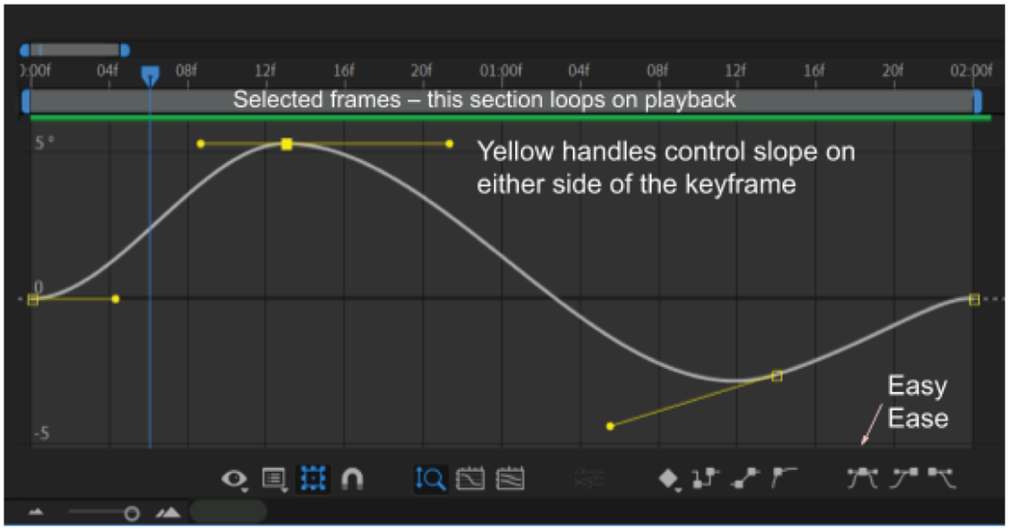
Before Beginning Animation

- ***IMPORTANT***! You **NEED** to read through the Frame-Animated Technical Legs Guide before starting this guide. This guide will be referencing that one; you should have both guides open for this process. You will be looking at each of them at various times.
- **Walk Cycle Process Timeline:**
  - I.e., *"Help! I'm confused and don't know what order to do things in!"*
    1) Read through the entire Frame-Animated Technical Legs Guide
    2) Create your After Effects file by making a duplicate of the file made in the Pivot Points and Parenting Guide and naming it ANM_SPR_[characterName]_Walk_NoLegs_v1
    3) Import the .mp4 version of the standard walk to the previously mentioned AE file and place it underneath the body. Make sure it is playing back at **24 fps – not 30fps**.
    4) Animate the body in AE to match the timing of the legs.

is being selected – in this case, the rotation of the head. You can see where all of your keyframes are by looking at the timeline (there will be markers along it).

- (Motion) Graph Editor: This is used to fine-tune your animations, adjust keyframes, add eases, etc. It is found here and looks like a mini graph – it will turn blue once it's enabled.
- We will discuss this more in the animation guides.





Selected frames – this section loops on playback

Yellow handles control slope on either side of the keyframe

Easy Ease

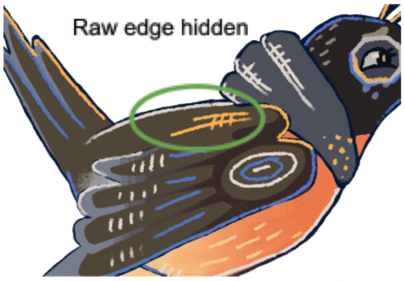
- **<u>DO NOT</u>** squash or stretch any of the assets. They should maintain their original aspect ratio exactly. Imagine the bird is a paper doll, with pins in the pivot points.

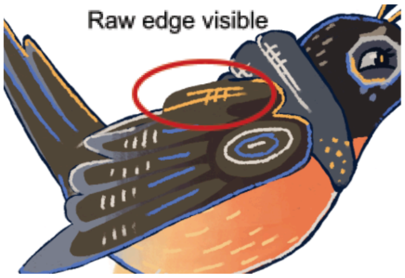
Saving

- Save the After Effects file as ANM_SPR_[nameOfCharacter]_BlankSetup

Extra Notes

- When animating, be mindful of the constraints of your pivot points – moving something too much may appear unnatural, or it may cause a body part to extend beyond its limits and show the raw edge of a body part that is supposed to blend in with another body part.


Raw edge hidden


Raw edge visible