Spiral Development to Post-Release

Project Flex, a macro-focused RTS

A Major Qualifying Project Report

submitted to the Faculty of the

Worcester Polytechnic Institute

in partial fulfillment of the requirements of the

Degree of Bachelor of Arts

in Interactive Media and Game Development

Submitted By:

Jason Asidi, Interactive Media and Game Development

Additional Contributors:

Benjamin Dorr, Interactive Media and Game Development

Advised By:

Professor Walt Yarbrough

Abstract

This report discusses the design, implementation, and analysis of Project Flex, a top down single player real time strategy game, or RTS, developed as part of the Spiral Development MQP. Project Flex is a casual RTS that focuses less on unit micro and base building and more on grand macro strategies, as seen in games such as Halo Wars and Dawn of War. Our goal in creating Project Flex was to explore various strategy game mechanics, including basic camera movement, unit selection, unit movement, resource gathering, unit construction, building construction, and more. In addition, Project Flex was developed using the Spiral Development framework, where we continually iterate and expand on the core game loop by designing and implementing singular mechanics, primarily to mitigate issues of scope creep.

Acknowledgements

We’d like to acknowledge and thank Professor Walt Yarbrough for his advice and guidance during the development of this project. Without his knowledge and encouragement, we may never have gotten as far as we had. We’d additionally like to extend our thanks to the many playtesters that helped us polish and refine the game throughout the development process, as well as our friends and family who supported us along the way.

Table of Contents

Introduction

Spiral Development to Post-Release is an MQP project that began in August 2023 and ends in April 2024. The major focus of the project is the use of the Spiral Development method to develop a game from start to finish, with a focus on rapid prototyping and frequent playtesting. Due to the successful completion of the project with minimal issues, we can declare that the Spiral Development model is successful in limiting the tendency for game development projects to fail due to scope creep.

Project Flex is an exciting real-time strategy game that invites players to lead cyberspace-themed armies to victory on a digital battlefield. Players must navigate strategic challenges, manage resources, and build powerful units to outmaneuver their adversaries. With its focus on strategic decision-making and battlefield control, Project Flex offers an immersive gaming experience where every choice shapes the outcome of the conflict. Whether engaging in intense skirmishes or constructing bases, players will find themselves immersed in a richly detailed world ripe for exploration and conquest.

This paper aims to describe the development process of Project Flex from start to finish, and explain several key aspects of its gameplay design. **Throughout the paper, we have highlighted some lessons for future MQP teams in bold.** In addition, we will be discussing the importance of playtesting to the development of the game, and overall the impact the use of the Spiral Development had on its development. Finally, we will review the overall success of the project and what things went well, what things could use improvement, and our final recommendations for others wanting to use the Spiral Development model in game development.

Background

Inspirations

The inspiration for Project Flex stemmed from a deliberate exploration of the strategy game genre during the preproduction phase in A term. Drawing from a diverse array of influences, the team studied various strategy games with distinct focuses to discern the optimal approach for the project. Notably, the Halo Wars and Dawn of War series emerged as major inspirations, with their emphasis on real-time strategy (RTS) gameplay tailored for a broader, more casual audience. From Dawn of War, the concept of resource generation through requisitions was adopted, encouraging dynamic map interaction by creating a contested resource shared between both players. Additionally, the Supreme Commander series influenced the decision to incorporate a diverse range of resources, compelling players to engage with multiple aspects of the game beyond mere unit management. Ultimately, the project coalesced into an RTS experience centered around macro strategy and map control, prioritizing strategic depth and player engagement over micro unit control.

Tools

Game Engine

The choice of game engine for Project Flex was a critical decision made early in the development process. Unity emerged as the preferred option primarily due to the team's familiarity with the engine's workflow and toolset. While investigating other common engines like Godot and Unreal Engine, specific considerations guided the decision-making process. Godot was excluded due to its lack of support for 3D art styles, which conflicted with the artist's specialization in 3D modeling. Unreal Engine, on the other hand, was ruled out because of concerns regarding its blueprint system and out-of-the-box support for real-time strategy (RTS) controls. Ultimately, Unity was deemed the most suitable choice given its familiarity and alignment with project constraints. **However, in hindsight, the team acknowledges that exploring more specialized engines after firmly deciding on the game concept might have yielded better results.** While Unity excels as a generalist engine, considerable time was spent developing fundamental functionalities that could have been streamlined in a more specialized engine.

Source Control

In managing source control for Project Flex, Github emerged as the chosen tool for its simplicity and familiarity among the team members. Initially employed as a means to share progress updates and back up data before implementing significant changes, Github seamlessly integrated into the team's workflow. Its user-friendly interface and widespread familiarity among team members meant it was our first choice, without any real consideration of alternative solutions. **While Github proved effective in facilitating version control, it's worth noting that the experience may not be fully representative, as only the programmer was primarily responsible for pushing changes.** Consequently, the absence of concurrent contributions minimized the risk of common issues such as merge conflicts. Despite this limitation, Github's reliability and ease of use ensured a smooth and efficient source control process throughout the development of Project Flex.

2D Art Asset Creation

In Project Flex, Gimp served as the primary tool for the creation of 2D assets, including various placeholder icons. These icons were either crafted from scratch or modified from existing free-to-use graphics to suit the project's needs. Gimp was selected primarily due to its familiarity, especially since the primary user was not an artist. While Gimp may lack some of the advanced features found in other graphics programs, its ease of use and familiarity minimized the learning curve for the team's primary use case. Overall, Gimp performed admirably in the creation of placeholder art, providing a versatile and accessible solution for teams with limited artistic resources. However, it's important to note that teams with dedicated artists may benefit from utilizing more specialized graphics programs tailored to their specific workflows. Nevertheless, in the context of modern game engines, most graphics creation programs offer similar functionality, making the choice of software ultimately dependent on their particular preferences and requirements.

Playtesting Feedback Collection

For the development of Project Flex, playtesting surveys were facilitated through the utilization of both Microsoft and Google Forms, offering comprehensive solutions for creating, managing, and collecting valuable data from playtesting sessions. Initially, the team experimented with both platforms but eventually gravitated towards Google Forms due to its seamless integration with Google Sheets and email, as well as its ease of collaboration. While both Microsoft and Google Forms offer robust features for survey creation and data collection, the decision to settle on Google Forms was driven by its compatibility with the team's existing workflow and its intuitive interface. **Ultimately, the choice between Microsoft and Google Forms should be based on the specific needs and preferences of each team**, as both platforms offer their respective ecosystems tailored to different workflows and requirements.

In addition, OBS (Open Broadcaster Software) served as the primary tool for recording playtest footage from expert group sessions, enabling thorough review and analysis at a later stage. The decision to utilize OBS was primarily driven by its accessibility as a free and open-source software solution. While testers had the freedom to choose their preferred recording programs, OBS emerged as the favored option among the majority due to its simplicity and reliability. As an industry standard for both professional and amateur recording, OBS offered a straightforward user interface and sufficient functionality for playtesting purposes. While other recording programs may offer more specialized features, **OBS proved to be a practical and effective choice for capturing playtest sessions with ease and accuracy.**

Design

Spiral Development

The core focus for our development on this MQP is the Spiral Development framework for game development, in which we begin by creating a very small but complete gameplay loop before building on this underlying structure to explore the game in stable cycles that always ends with a complete gameplay loop (King, 2021). The goal of this style of development is to avoid scope creep and keep the development process fluid until completion, thus avoiding an awkward release where entire sections of the game aren't finished on time and have to be cut back.

Scope

Throughout the development of Project Flex, managing scope was a paramount consideration, guided by the principles of Spiral Development. The project's evolution was marked by a series of deliberate steps to refine mechanics, address feedback, and prioritize features to ensure a manageable scope and a successful end product. The process commenced with the implementation of fundamental mechanics centered on isometric camera movement and object selection, gradually laying the foundation for an isometric strategy game. Over the course of A term and the initial half of B term, the development efforts were channeled into refining these mechanics while integrating essential real-time strategy (RTS) staples such as point control, resource generation, unit construction, and unit health and attacking. By the midpoint of B term, Project Flex had evolved into a rudimentary strategy game, boasting a barebones framework ready for further expansion and refinement. Playtesting during this phase predominantly focused on evaluating the smoothness and intuitiveness of controls, ensuring a seamless user experience essential for the game's success.

As development transitioned into the latter half of B term and extended into C term, the looming threat of scope creep necessitated a proactive response from the team. Recognizing the importance of maintaining focus on essential elements, efforts were concentrated on refining the feel and mechanics of combat to ensure smooth and accessible unit control. This endeavor demanded extensive playtesting to validate the goal of reducing need for individual unit control during combat, otherwise known as micro, ensuring that player interaction remained intuitive and engaging. As a result, non-essential elements to combat, such as the implementation of a minimap, a second tier of units, and skirmish building AI, were temporarily sidelined to prioritize core gameplay mechanics.

During D term, the project's focus shifted towards fine-tuning existing mechanics, reintegrating sidelined content, and gathering feedback for version 1.0 and future updates. This phase marked a pivotal moment in aligning project scope with development priorities, ensuring the game's coherence, polish, and responsiveness to player needs. Elements such as the T2 units and unit factory, previously deferred, were reintroduced into the game, enriching gameplay diversity. Additionally, significant efforts were dedicated to addressing long standing bugs that had plagued development, including issues related to fog of war visibility and rare crashes, thereby enhancing overall stability and user experience. Furthermore, the focus on polish extended to implementing "nice-to-have" features like a minimap and hotkeys, elevating the game's accessibility and user-friendliness. Throughout this phase, playtesting remained a central focus, aimed at refining gameplay mechanics and ensuring that every interaction was enjoyable and intuitive for players.

Overall, the journey of managing scope in Project Flex epitomized the iterative nature of game development. By consistently evaluating progress, prioritizing features, and responding to feedback, the team navigated the complexities of scope management, culminating in a well-rounded and engaging RTS experience.

Core Gameplay Mechanics

Real-Time Strategy (RTS) games stand as a genre renowned for their engaging gameplay mechanics, where players navigate complex scenarios, manage resources, and orchestrate large-scale battles. These fundamental elements collectively contribute to the dynamic and immersive nature of RTS games.

Isometric Camera

The isometric camera perspective is a distinctive feature that provides players with a three-dimensional view of the game world. This perspective allows for a comprehensive understanding of the terrain, promoting effective strategic planning and decision-making. Players can survey the battlefield with ease, assessing the layout of structures, resources, and potential threats. Isometric camera movement not only enhances the visual experience but also plays a pivotal role in facilitating tactical awareness and spatial navigation.

Fog of War

The inclusion of Fog of War introduces a layer of strategic complexity to RTS games. As units explore the map, the fog lifts, revealing previously hidden areas. This mechanic not only simulates the uncertainty of the battlefield but also necessitates reconnaissance and intelligence gathering. Players must carefully scout the terrain to gain insights into enemy movements and positions, adding an element of anticipation and surprise to engagements. Fog of War thus becomes a key factor in shaping player strategies and decision-making.

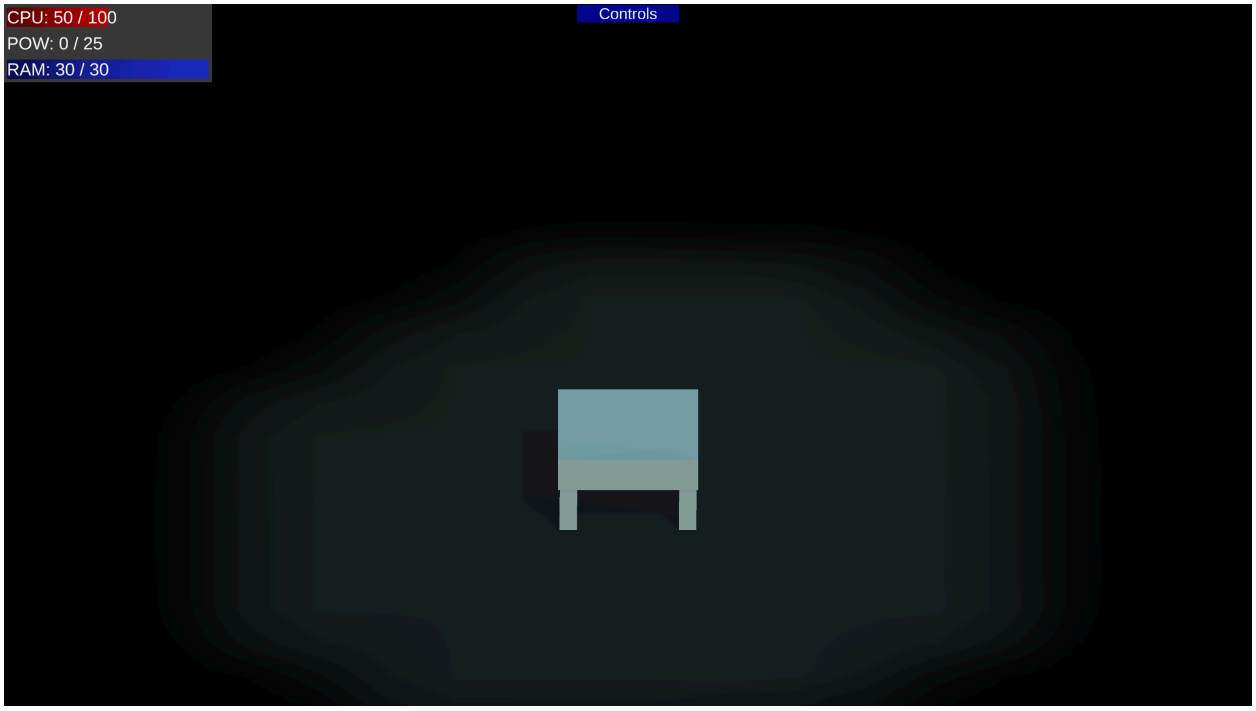

Figure 1 - An in-game screenshot depicting the fog of war.

Unit Selection and Commands

Efficient control of units is paramount in RTS games, demanding a user-friendly interface that enables players to manage diverse armies seamlessly. The ability to select individual units, form groups, and issue commands is crucial for executing complex strategies. The responsiveness of commands, coupled with the ease of unit selection, determines the player's capacity to coordinate movements, engage enemies, and adapt swiftly to changing circumstances. The skillful use of unit commands is a hallmark of expert RTS gameplay.

Resource Management

Resource gathering forms the economic backbone of RTS games, introducing a layer of strategic decision-making. Players allocate workers to collect resources such as minerals, energy, or food, with the acquired wealth used for constructing buildings, training units, and advancing technological capabilities. The delicate balance between expansion, resource extraction, and military production requires players to make strategic choices that impact both short-term and long-term gameplay. Effective resource management becomes a critical skill for success in the competitive RTS environment.

Base Building

Unit and building construction is a central component of RTS gameplay, allowing players to shape their military and economic capabilities. The diversity of available units and structures provides players with strategic options, fostering adaptability to different scenarios. Decision-making during this phase involves considerations of unit composition, base layout, and the allocation of resources to ensure a well-rounded and formidable force. Unit and building construction serve as the foundation for executing strategic plans and responding to the evolving dynamics of the game.

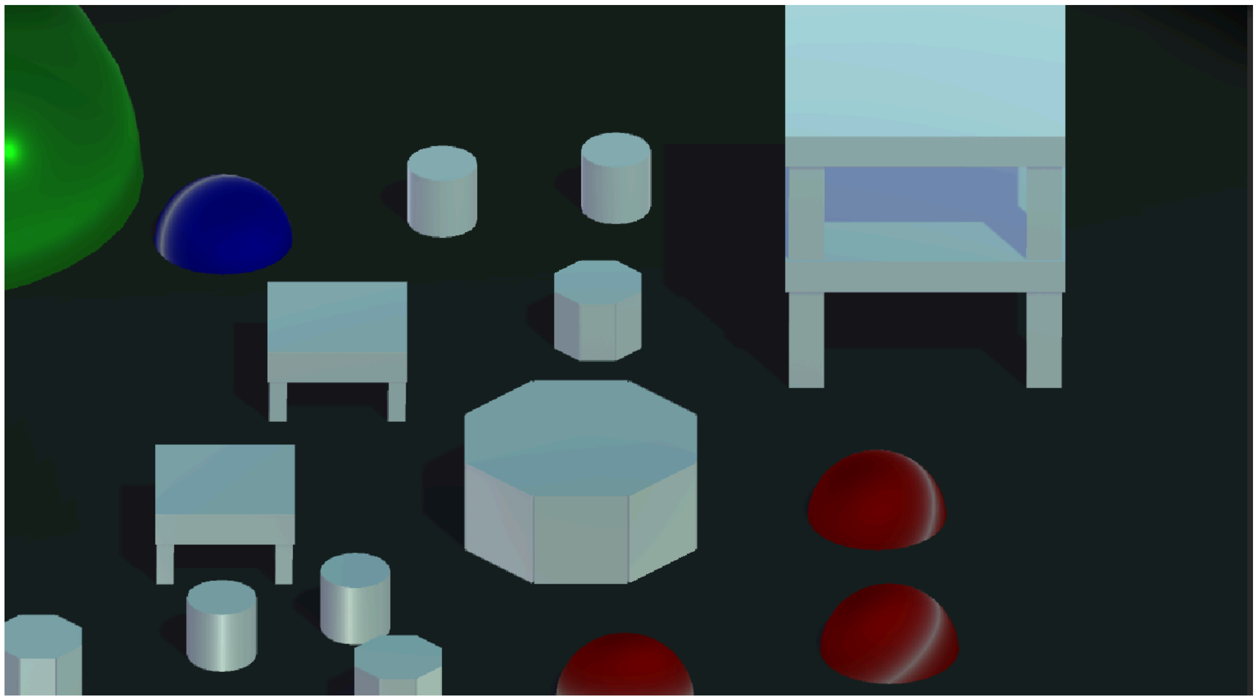

Figure 2 - A late game base.

Combat and Victory

The culmination of RTS gameplay lies in army engagement and achieving victory conditions. As players amass their armies through resource management and construction, battles unfold in real-time. Tactical decisions during engagements, such as unit positioning, use of special abilities, and coordinated attacks, directly influence the outcome. Victory conditions vary but often involve the elimination of opponents, control of key points on the map, or achieving specific objectives. The strategic depth of army engagement and the diversity of victory conditions contribute to the replayability and enduring appeal of RTS games.

Gameplay

Units

In Project Flex, the design philosophy behind unit creation revolves around strategic diversity, ensuring that each unit serves a distinct role, whether from a military or cost-effectiveness standpoint. The game features two tiers of units—infantry and vehicles—each tailored for specific purposes to foster nuanced and engaging gameplay.

Infantry

Infantry units epitomize speed and cost-effectiveness. They are the swift operatives on the battlefield, capable of capturing control points crucial for CPU generation. Their mobility allows for quick point capture, hit-and-run tactics, and dynamic responses to emerging threats. Construction of infantry units is economical, demanding only CPU and 1 RAM per unit. This cost-effectiveness aligns with their disposable and versatile nature, making them the preferred choice for early-game exploration, map control, and rapid response scenarios. Infantry based armies excel at early-game maneuvers, securing control points for sustained CPU generation. Their low-cost nature facilitates strategic map control and exploration, enabling players to establish a foundational presence on the battlefield.

Basic Infantry

The linchpin of unit design, Basic Infantry serve as the well-rounded foundation upon which other units are built. Versatile and adaptable, they form the backbone of early-game strategies and provide a benchmark for comparing other unit types.

Drone Infantry

Positioned as the cheapest unit, Drone Infantry fulfill a crucial role in exploration and point capture during the initial phases of the game. While they lack cost efficiency in combat against other units, their affordability makes them a go-to choice for early-game map control, while the fact that despite their decreased costs and stats they still take the same amount of RAM as any other unit infantry discourages a spam-oriented playstyle.

Fast Infantry

As the fastest units in the game, Fast Infantry excel in capturing contested points swiftly and executing hit-and-run attacks. Their speed allows them to evade reprisals, especially from enemy vehicles, making them ideal for dynamic, tactical maneuvers on the battlefield. Despite their utility, Fast Infantry are not as fragile as drones and still are a comparable combat unit, meaning that they can scout or eliminate key targets in the first stages of an assault before the rest of the army rolls in for support.

Heavy Infantry

Representing the pinnacle of infantry strength, Heavy Infantry serve as anchors for assaults in the pre-vehicle phase. Their robust capabilities come at a higher cost, making them a strategic investment for players aiming to control key areas and establish dominance early on. Even after T2 units start coming out, Heavy Infantry can serve as wardens for control points, tough enough to endure vehicle firepower long enough to capture a point and force an infantry response in order to recapture it.

Vehicles

Vehicles stand in stark contrast to infantry, embodying durability and strength. However, this robustness comes at the expense of speed, making them slower and more deliberate in their movements. Vehicles often possess unique characteristics or special abilities that set them apart, contributing to their role as heavy-hitters in battle. Building vehicles is a resource-intensive endeavor, requiring CPU, Power, and multiple RAM per unit. The significant investment in these units reflects their potent combat capabilities and specialized roles on the battlefield. The higher resource cost signifies their importance in mid-to-late-game strategies.Vehicles play a pivotal role in assaulting fortified positions and engaging in high-stakes battles. Their durability allows them to withstand enemy fire, making them formidable anchors in offensive strategies. The additional resource requirements underscore their importance in mid-to-late-game scenarios, where a well-timed deployment can turn the tide of battle. In addition, vehicles, being the more resource-intensive units, often boast unique features that set them apart. This specialization could manifest as increased firepower, special abilities, or tactical advantages that amplify their effectiveness on the battlefield. These distinct attributes further underscore the significance of vehicles in shaping the outcome of engagements.

Basic Tank

The workhorse of vehicle units, Basic Tanks offer straightforward firepower without additional frills. Designed for engagements against multiple infantry units, they provide a solid foundation for mid-game strategies and serve as a versatile force on the battlefield.

Missile Tank

Specialized as an anti-horde unit, Missile Tanks are unique for their splash damage capabilities. While effective against groups of infantry, they lack potency against other vehicles and structures due to their focus on dealing area-of-effect damage. They're best used for a vehicle based strategy to clear and defend control points from infantry, and are vulnerable to ambushes from other vehicles if not cared for.

Mortar Tank

Slow-firing and lumbering, Mortar Tanks represent the idea of extreme overkill. With the highest damage per shot in the game, they are dedicated anti-vehicle and anti-building units, offering strategic depth for players aiming to break through fortified positions. In addition, since they also support the longest ranged weapon in the game, but lack the vision range to utilize it by themselves, they do well as far ranged artillery to support an assault by faster units that can act as spotters for it.

Support Tank

Enhancing the relevance of infantry in the late game, Support Tanks act as force multipliers. Their aura increases the fire rate and movement speed of nearby infantry units while providing healing. While unable to attack, their supportive role makes them invaluable in sustaining infantry-focused strategies. Player feedback surprisingly landed the Support Tank as the favorite vehicle, which could be attributed to the fact that unlike the other vehicles, it doesn't require retooling of the player's production capacity to be majorly biased towards vehicles, instead supporting a combined arms approach.

Unit Roles

The strategic diversity of units in Project Flex prompts players to make nuanced decisions based on their immediate needs, the evolving battlefield, and their opponents' strategies. Each unit type has a specific role, encouraging players to consider their composition carefully, adapt to changing circumstances, and capitalize on the strengths and weaknesses of their chosen units. The two-tiered structure of infantry and vehicles, coupled with the distinctive roles of each unit type, fosters engaging gameplay and rewards thoughtful decision-making on the path to victory. The contrast between infantry and vehicle units introduces a strategic layer to unit selection, emphasizing their distinct roles, costs, and unique contributions to the battlefield. These differences are not only reflected in their attributes but also impact the overall dynamics of resource management and strategic decision-making. While infantry dominate the early game with their agility and affordability, vehicles become instrumental in the mid-to-late game for assaults and specialized roles. Players must strike a delicate balance in their unit composition, adapting to the evolving dynamics of the battlefield.

Resources

In Project Flex in particular, the three different resources–CPU, Power, and RAM, all influence player decision making in a major fashion. Each of them are named after the game’s computer theme, and each are important at different stages of the game. The relationship between the different resources and their uses in the player’s production process greatly influence the gamestate and how players begin to grind towards victory.

CPU

CPU is the most prevalent resource, with both players having a constant income from the very start of the game and greatly increasing said income stream with each capture point taken on the map. However, since it is the only resource that can't be gained from constructing buildings, it forces the players explore the map early on to find and lay claim to these points, and from there on constantly skirmish and defend these points to maintain control of their main sources of income. The significance of CPU lies in its ubiquity, as it is utilized for a wide range of functions, including unit production, base construction, and more. As such, despite being seemingly plentiful, CPU becomes the major limiter of the player's expansion of their army and pursuit of victory, and thus the main impetus for player engagement with the opponent and the map.

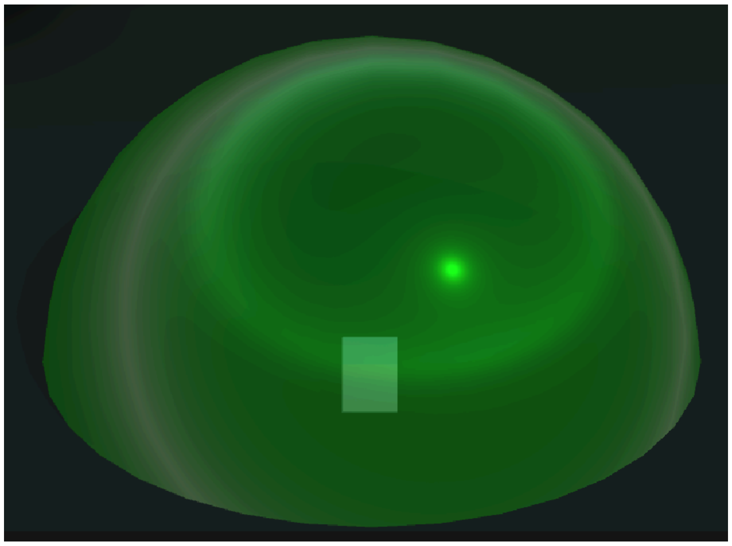

Figure 3 - An infantry unit in a captured point.

Power

Power, a more premium resource in Project Flex, is instrumental in the construction of buildings and higher-tier units. Unlike CPU, Power is not passively regenerated; it is exclusively generated by power supply buildings, and the rate of generation is relatively slow. This scarcity in the early game prompts players to prioritize the establishment of their army before heavily investing in base construction. As the battle progresses and players amass a stronger foothold, the importance of Power amplifies. In the mid-game, when armies are well-established and control points secured, players find themselves relying heavily on Power to expand their production capabilities and deploy advanced units capable of overwhelming their adversaries.

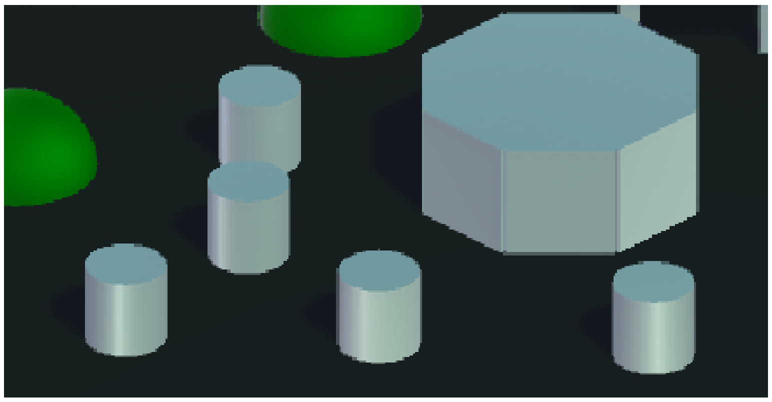

Figure 4 - A base filled with power supply buildings.

RAM

RAM introduces a unique dynamic to resource management in Project Flex. Unlike CPU and Power, RAM is never passively generated. Instead, it is immediately granted upon the construction of specific buildings and is equally deducted if those structures are destroyed. This design choice encourages players to strategically position and defend RAM-generating buildings, considering them as vital assets in their base layout. Importantly, RAM does not provide immediate value, instead increasing the maximum number of units a player can have simultaneously. This makes RAM the most late-game oriented resource, emphasizing the need for a balanced and sustainable strategy. Neglecting RAM expansion can prove to be a critical error, as it significantly impacts the size and strength of the player's army. As the game progresses, the player who can best leverage the increased army size offered by RAM is strategically poised to overpower their opponents and win the game.

In summary, Project Flex's resource system is a carefully crafted interplay of CPU, Power, and RAM, encouraging players to think critically about their approach to the map and the expansion of their base. The distinct properties and acquisition methods of each resource contribute to a dynamic and strategic gameplay experience, where effective resource management is pivotal to success in the ever-evolving battles of Project Flex.

Base Building

In Project Flex, the art of base building is a strategic dance, where players must navigate the delicate balance between expanding unit production and supporting resource generation. The tension inherent in these considerations becomes a defining aspect of gameplay, requiring thoughtful decisions and adaptability throughout the course of battle.

Military Production

The foundation of every successful strategy in Project Flex lies in the efficient production of units. Bases serve as the epicenter for unit construction, and players must decide when and where to invest in unit-producing structures. The temptation to focus solely on unit production is ever-present, especially in the early stages of a battle. A formidable army can tip the scales in engagements and provide a strategic advantage. However, this singular focus comes with a trade-off—the potential neglect of resource generation, limiting the sustainability and growth of the player's forces. A player who emphasizes a large amount of focus on their military capabilities from the start of the battle may be able to take an early lead over the opponent when it comes to map dominance, but a savvy general will be able to build up his defenses and resource production during such a rush and force the fast victory hoped for into a grinding stalemate where their more stable foundation gives them a massive advantage. As a result, while military expansion is important and should be given the focus in the early game, part of the game’s strategy comes from knowing the consequences of overinvestment.

Resource Generation

Resource generation is the lifeblood of a player's capabilities in Project Flex. The efficient collection of CPU, Power, and RAM dictates the extent to which a player can expand their forces and construct advanced units and structures. As such, the strategic placement of resource-generating buildings becomes a critical consideration during base building. Players must resist the allure of an overly militarized approach and allocate space and resources to support a robust economic engine. Neglecting resource generation can lead to a stagnation of unit production, hindering a player's ability to respond to evolving threats and opportunities on the battlefield. On the other hand however, just like an overly aggressive military buildup can spell do from the beginning of the match, trying to play passively is only a failing strategy, mainly due to the need to capture control points in order to generate CPU, the most widely used resource. Without it, even creating lots of factories, defenses, and resource generators is useless because you won’t be able to fund your economic rush. This careful balance between the need for economic stability and also maintaining control of the map is important for the strategic balance of Project Flex.

Resource Storage

Resource storage serves as the final core aspect of base building in Project Flex, playing a crucial role in managing the economy and facilitating strategic decision-making. Unlike production buildings and unit production, storage buildings do not directly increase resource generation capabilities but rather enhance maximum production capacity. This distinction underscores the importance of balancing resource generation with storage capacity, as high resource generation becomes futile without sufficient storage to accommodate it. Moreover, the amount of resource storage directly influences the number of constructions that can be initiated concurrently, highlighting the strategic significance of maintaining adequate storage levels.

Strategic resource management in Project Flex extends beyond mere accumulation to encompass the timing of resource allocation for maximum production efficiency. Knowing when to allocate resources towards expanding storage capacity, even at the expense of current resources, is paramount to fostering long-term growth in overall production capacity. This strategic decision-making process requires careful consideration of current resource needs, future production goals, and potential trade-offs between immediate gains and future benefits. By strategically balancing resource allocation between immediate needs and long-term expansion, players can optimize their production capacity and gain a competitive edge on the battlefield.

Art

User Interface

Each of the UI elements in Project Flex was carefully designed and tweaked to maximize the amount of information shown in as small an amount of space as possible. The final result balances readability with compactness in a way that appeals to most players.

Resource Bars

In Project Flex, the resource UI was designed and tested for quick comprehension, ensuring players could easily grasp their resource status. The UI elements were color-coded to align with corresponding resource storage buildings, allowing players to quickly take a look at the battlefield and identify what they're currently lacking in resource production just by looking at the colors of the buildings. With the introduction of resource storage buildings, the UI was updated to include current resource limits, allowing players to see what their current caps were at and how quickly they were filling them out. However, a minor challenge arose with the display of RAM, which behaves uniquely compared to other resources since it doesn't regenerate but rather is granted in chunks permanently. Eventually, what we settled on was to be somewhat misleading in showing both the current resource and the cap, despite the actual display being the number of units worth of RAM and the current resource. Overall, the resource UI effectively facilitated resource management, contributing to a seamless gameplay experience.

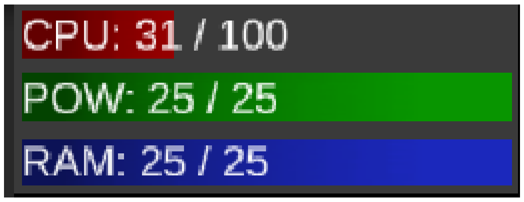

Figure 5 - The Resource Bar UI.

Unit Selection

In Project Flex, the UI for selected units was carefully crafted to offer players vital information without cluttering the screen. It succinctly displays the health and icons for up to 25 selected units, providing a quick overview of the state of the army. Initially, there were plans to allow players to select individual units from this UI, but testing revealed that this added unnecessary complexity and detracted from the game's macro-strategy focus. While players generally responded positively to the UI, there were concerns about its compatibility with final unit designs and particularly with icons that are less easy to identify. Further testing is needed to strike the right balance between detail and screen space, ensuring an optimal player experience.

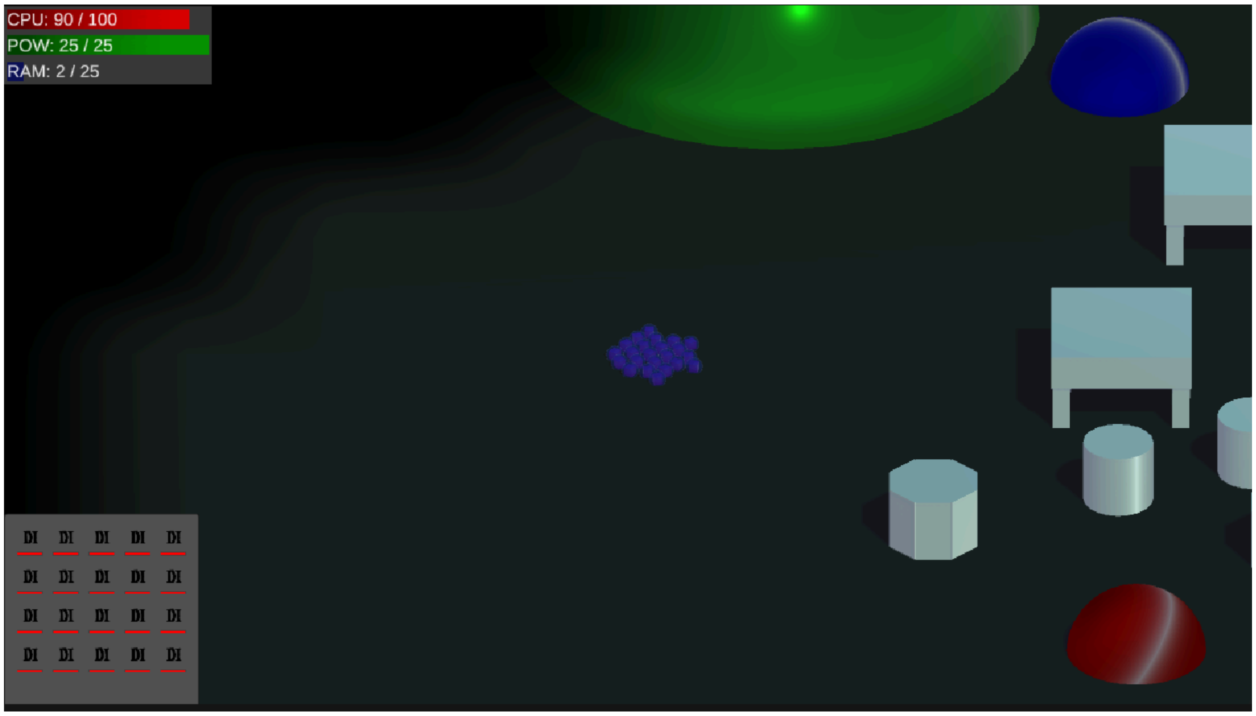

Figure 6 - The maximum number of units shown on the unit selection UI.

Unit Details

We introduced a detailed unit view UI to provide players with in-depth information when a single unit is selected. This feature offers essential details like health, unit icon, name, and relevant stats such as DPS, speed, or construction range, allowing players to better understand each unit's capabilities. The decision to include this feature stemmed from challenges in the main UI, which made it difficult to access specific unit stats. However, feedback from playtesters varied, with some players perceiving it as unnecessary for casual play, as unit differences were evident through gameplay alone. Nevertheless, we retained the detailed unit view to cater to more serious players who valued the ability to compare and contrast units for strategic decision-making.

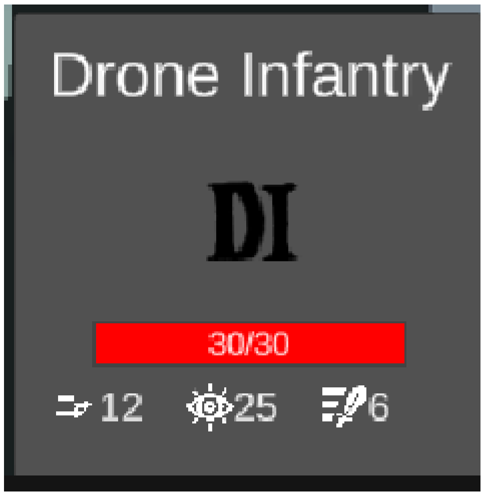

Figure 7 - The unit details for an infantry unit.

Construction

Crafting the construction menu UI was a dynamic process shaped by the diverse needs of strategy game interfaces. Needing to balance showing enough information with making sure that the menu didn't take up most of the screen, we iterated through several adjustments based on player feedback to ensure clarity and integration with the overall UI. The final design presents players with a comprehensive view of available units, including icons, queue counts, and build time remaining. To aid decision-making, the construction cost for each resource is displayed adjacent to the units. We opted to exclude unit stats from the screen, focusing instead on providing concise text descriptions of each unit's battlefield role. This approach aimed to streamline the interface and prioritize essential information. Through continuous refinement guided by player input, the construction menu UI evolved to balance accessibility and depth for all types of players.

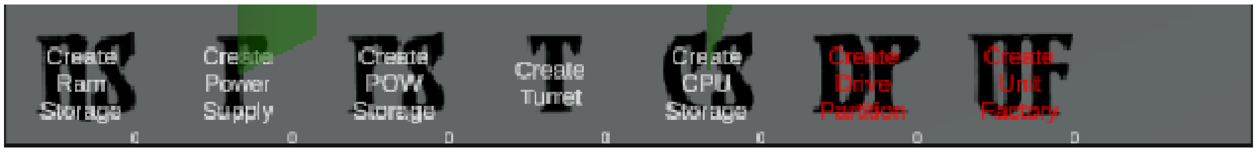

Figure 8 - The Construction UI for a building constructor.

Minimap

The minimap was the latest addition to the user interface, designed to enhance players' situational awareness and strategic planning. The minimap provides a clear overview of the battlefield, differentiating between infantry, vehicle, and building units with distinct icons, as well as between teams using colors. The ability for players to click on specific positions to shift the camera instantly allows rapid changes in focus, facilitating rapid response and precise coordination. Initial feedback from players indicates a strong appreciation for the minimap, citing its utility in planning multi-stage assaults or managing engagements across multiple areas. Its integration into the UI has notably improved gameplay dynamics, offering players a valuable tool for navigating the complexities of strategic warfare in Project Flex.

Placeholder Art

Due to the loss of the artist from the team, we unfortunately weren't able to replace the placeholder art assets for Project Flex. These temporary visuals provided a visual representation of game elements, allowing the team to focus on refining gameplay dynamics and iterating on core features without being hindered by the absence of finalized artwork.

3D Models

For Project Flex, the simplified 3D models used as placeholders were instrumental in distinguishing between various units and elements within the game. Infantry and vehicles were primarily differentiated by size, with color serving as a key identifier for both. Control points were represented by half spheres that changed color upon capture, providing clear visual feedback to players. Buildings, on the other hand, featured a range of complex shapes constructed from multiple primitives, making it easy for players to distinguish between different structures and gauge their quantities. The use of color and size as distinguishing factors received positive feedback from playtesters, with some suggesting that the theme of primitive objects as units could be further integrated into the final product with minor adjustments.

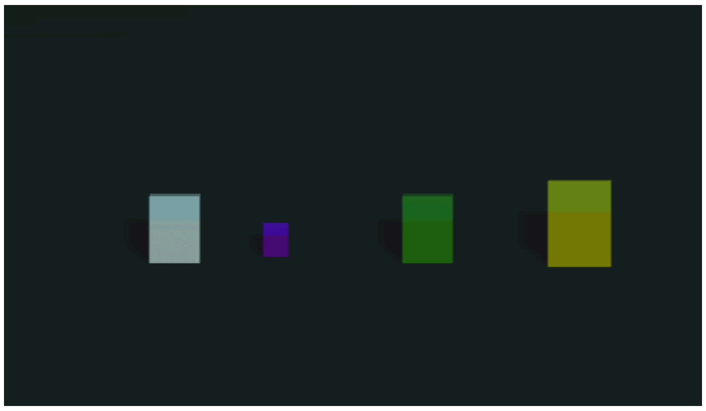
Figure 9 - The placeholder models for infantry units.

Icons

Placeholder icons were a fundamental part of the development process in Project Flex, particularly within the detailed unit UI. These icons, sourced from freely available sprites found online, acted as visual representations for the various units in the game. Despite their simplicity—comprising only the initials of each unit—maintaining these icons proved challenging due to their scattered references throughout the game. Looking back, it would have been a good idea to implement a centralized system for icon definitions to streamline asset replacement and ensure visual consistency throughout the game. During playtesting, while functional, some testers noted that the icons appeared somewhat generic or visually unappealing, and when units with similar initials were selected it could be somewhat confusing to tell at a glance what they meant. It likely would have been a better idea to pick simple icons that represented the unit's role, such as a shield for heavy infantry or a rocket for missile tanks.

Spiral Development

Benefits of Spiral Development

In the development journey of Project Flex, embracing Spiral Development methodology has brought forth numerous advantages, fostering a dynamic and adaptable approach to crafting the game. **Notably, one significant benefit has been the infrequency of having to completely cut content from the project.** Even when adjustments were needed, they were executed in a manner that preserved the game's core playability and left room for future expansion. This adaptability owes much to the modular design of the game's mechanics, which were intentionally structured to accommodate seamless integration of new resources, units, and buildings, facilitating ongoing innovation and diversification of gameplay options.

Furthermore, Spiral Development has simplified the creation of game environments. By leveraging a library of prefabricated assets, constructing maps becomes a straightforward task, requiring only additional terrain and the creative touch of a level designer to bring vibrant RTS environments to life. This modular approach not only streamlines the development process but also empowers the team to focus on creative design aspects, with less time spent on technical implementation. Additionally, the modular nature of the game's mechanics and environments encourages collaboration, enabling rapid prototyping and experimentation of ideas.

In the broader context of game development, the principles of Spiral Development offer a range of benefits. By continuously iterating on game features and content, developers can remain agile, responding effectively to changing requirements and player feedback. This iterative process not only reduces the risk of major setbacks but also fosters a culture of innovation and creativity. Furthermore, the modular design inherent in Spiral Development promotes scalability and extensibility, allowing games to evolve over time to meet the evolving needs of players and the gaming landscape. In summary, the adoption of Spiral Development methodology in game development provides a flexible, efficient, and adaptable framework for creating immersive and engaging gaming experiences.

Drawbacks of Spiral Development

While Spiral Development brought numerous benefits to the project, it has also presented its share of challenges. The relentless focus on rapid prototyping and modular development, inherent to Spiral Development, has at times proven to be detrimental to the game's progress. While the emphasis on creating reusable systems has its merits, there is a risk of overly prioritizing modularity at the expense of practicality and efficiency. **In some instances, the effort invested in making systems reusable and easy to prototype has resulted in added complexity and development overhead, making them more challenging to implement and maintain in the long run.**

In the broader context of game development, these drawbacks of Spiral Development are not unique to Project Flex but are challenges commonly encountered in iterative development processes. Effective communication, judicious selection of development tools, and striking a balance between modularity and practicality are key considerations for teams embarking on projects utilizing Spiral Development methodologies. By addressing these challenges proactively, developers can mitigate potential pitfalls and harness the benefits of Spiral Development to create engaging and successful gaming experiences.

Conclusions on Spiral Development

In conclusion, spiral development emerges as an effective approach for game development, offering a framework that prioritizes iterative refinement and rapid prototyping. The methodology emphasizes the importance of delivering playable iterations early and often, recognizing that it is preferable to have a functional prototype for playtesting rather than striving for perfection from the outset. However, it is essential to remain mindful of the inherent challenges of spiral development, particularly in small teams. With fewer oversight and viewpoints in development, small teams may struggle to sustain the rapid pace of development required by spiral methodologies, leading to potential burnout and inefficiencies. Nonetheless, by embracing the principles of spiral development while remaining cognizant of its limitations, game developers can leverage its strengths to navigate the complexities of game development and ultimately deliver compelling and successful gaming experiences.

Playtesting and Feedback

Protofest

During A-Term, we had the opportunity to participate in ProtoFest, an important event specifically designed for testing prototypes of various projects at their earliest stages. This event allowed us to present a demo build of our project, which centered around the development of a real-time strategy (RTS) game. Our demo build showcased the core gameplay mechanics of our RTS game, including camera movement, unit movement and selection, resource gathering, and unit construction. These features are fundamental to the player's experience, and we wanted to ensure they were intuitive, engaging, and enjoyable.

Camera Controls

ProtoFest attendees had the opportunity to test the camera controls, assessing their responsiveness, speed, and ease of use. This aspect is vital in an RTS game to ensure players can navigate the game world efficiently. The feedback we received was instrumental in refining the camera controls and adjusting the camera speed to a level that felt comfortable and user-friendly. Based on feedback, we fine-tuned the camera speed to ensure that players could explore the game world comfortably without it feeling too fast or sluggish, switching from the camera moving at a constant fast pace to starting slow but quickly accelerating if the camera continued moving in the same direction..

Unit Movement

Testing the unit movement and selection mechanics was crucial for us to evaluate how well players could control their units. Attendees provided insights on the responsiveness of unit commands and helped us identify any issues in unit pathfinding and selection. This feedback guided us in fine-tuning the average unit speed and ensuring that unit selection was as seamless as possible. Feedback from ProtoFest attendees allowed us to adjust the average unit speed to strike the right balance between realism and gameplay fun. From the responses we gathered, the average unit speed was just slightly too slow for what the player's expected for a unit of average size, and we increased it accordingly.

Resource Gathering

In an RTS game, resource management is a key element. We incorporated resource gathering mechanics into our demo build, and ProtoFest participants were able to evaluate the resource collection process. Their feedback allowed us to adjust resource generation rates and refine the resource gathering experience to strike the right balance in gameplay.

Unit Construction

Attendees had the opportunity to experience unit construction, an integral part of our game. Testing this feature helped us determine placeholder construction rates and evaluate the overall pace and progression of the game. The feedback received during ProtoFest played a vital role in optimizing the pacing and balance of unit construction within the game, and we used the responses from the initial feedback to inform our construction rates for future additions.

Unit Models Theme

The feedback also provided us with a better understanding of the player's preferences and expectations for unit models. This, in turn, helped us make informed decisions on how we should theme and design our unit models, ensuring they resonate with the target audience.

The feedback gathered during ProtoFest was invaluable. **It not only provided insights into the usability and playability of our game but also helped us make informed decisions regarding several aspects of our project.** ProtoFest served as a pivotal testing ground for our MQP project, enabling us to gather crucial insights, fine-tune key gameplay mechanics, and make informed decisions about the direction of our game. The feedback received during this event not only improved the overall quality of our project but also helped us align our design choices with the player's expectations, ensuring a more engaging and immersive gaming experience.

Alphafest

AlphaFest, a prominent event during B term, marked a pivotal moment in the development of our MQP. This event provided an opportunity for a broader audience to experience our game, offering a first glimpse into the culmination of our efforts in the form of an alpha build. This alpha build not only showcased refined mechanics from previous testing phases but also introduced new elements, including unit combat and a demo level that synthesized all key gameplay mechanics.

Refined Mechanics Testing

Building upon the foundation laid during ProtoFest, our alpha build featured the refinement of critical gameplay mechanics. Attendees at AlphaFest had the chance to explore and test the intricacies of camera movement, unit movement and selection, resource gathering, and unit construction. This phase allowed us to verify the effectiveness of refinements made based on ProtoFest feedback and ensure a smoother, more immersive player experience.

Combat Mechanics

A significant highlight of the alpha build was the introduction of combat mechanics. For the first time, units within the game had the capability to engage in battle with each other. This implementation brought a new layer of depth to our project, transforming it from a construction and resource management simulation into a real-time strategy (RTS) game. Attendees were now able to witness and participate in dynamic battles, adding an exciting dimension to the overall gaming experience.

Core Gameplay Loop

The focal point of our alpha build at AlphaFest was the combat test, designed to allow the player to utilize all the mechanics implemented in the game at that time. This level was not merely a collection of features but a strategic challenge that encapsulated the essence of our envisioned gameplay. It served as an embodiment of the player's journey within our RTS world, demanding resource management, strategic thinking, and tactical decision-making.

In the test level, players were entrusted with the task of constructing an army, gathering the necessary resources by assaulting control points guarded by turrets. The intricacies of our game unfolded as players navigated through the dynamic process of assembling their forces, each unit contributing to the growing strength of their army. This phase served as a test of the intuitiveness of our unit construction mechanics, ensuring that players could seamlessly transition from resource accumulation to unit deployment. As the players grew their army and controlled more of the map, they were then faced with the challenge of assaulting a fortified base defended by more automated turrets. This marked the culmination of our efforts to slowly build up mechanics into our core gameplay loop as we tested and implemented them. The demo level was not only a showcase of unit-to-unit interactions but also the first encounter with hostile structures, emphasizing the strategic importance of balancing offense and defense. The defensive turrets provided a tangible obstacle, prompting players to consider their army's numbers, positioning, and timing as they attacked the resource points and enemy HQ.

Feedback

The response from AlphaFest attendees was overwhelmingly positive. Players enjoyed the immersive experience of building their armies, strategically maneuvering units, and engaging in intense battles. The demo level successfully demonstrated the cohesion of our primary gameplay loop. While the reception was favorable, valuable feedback and minor notes for improvement were gathered. Attendees provided insights into aspects such as unit balance, enemy difficulty, and the overall pacing of the demo level. These notes, though minor, are instrumental in ensuring that our game not only captivates players but also provides a balanced and engaging experience.

In conclusion, AlphaFest served as a significant milestone, marking the unveiling of our game's alpha build to a wider audience. The success of our alpha build at AlphaFest affirmed the strength of our project's core design and mechanics. As we move forward in the development process, the feedback received will guide us in making targeted refinements. This iterative approach, consistent with the principles of Spiral Development, positions us to deliver a polished and compelling gaming experience as we advance towards the final stages of our MQP. The positive reception, coupled with valuable feedback, reinforces our commitment to delivering a high-quality and enjoyable gaming experience, and we look forward to incorporating these insights into the next phases of development.

C Term Playtesting

Starting from C term onwards, a new chapter unfolded in its development journey as the project shifted from an on-campus endeavor to a remote collaboration. This shift introduced a set of challenges, with one of the most conspicuous being the limitation on the ease of obtaining playtesting feedback from the familiar circles of roommates, friends, and on-campus events. Despite the constraints, this phase marked a turning point that demanded a proactive and strategic approach to playtesting, especially as we began entering a phase of content development which required more focus on balance and tuning various features, and thus more frequent testing. Faced with the need for a more deliberate playtesting approach, I took the initiative to reach out to colleagues with whom I had previously collaborated on a strategy game. Leveraging existing professional connections proved to be an effective solution. These colleagues, already familiar with game development dynamics and possessing a strategic understanding of gameplay mechanics, became the primary playtesters for Project Flex during this phase.

With a new set of playtesters secured, the iterative feedback process continued, albeit in a remote setting. The challenges of physical separation were mitigated by leveraging various collaboration tools. Video conferencing, shared documents, and communication through the use of Discord facilitated seamless discussions and exchange of ideas. These tools became instrumental in maintaining an effective feedback loop, enabling thorough discussions on gameplay mechanics, user experience, and overall game dynamics.

Unit Testing

As is fitting with the focus of the term on content development, the very first set of changes put into testing were the expanded unit pool, which focused mainly on infantry units.

The playtesting phase revealed that Time-to-Kill (TTK) was notably high, inadvertently encouraging players to spam the most cost-effective units damage-wise. This approach undermined the potential for strategic endurance and varied unit compositions, and hurt units that were supposed to be centered around their durability such as Heavy Infantry especially weak. In response to these findings, adjustments were implemented to lower most units' damage output, requiring at least two shots to defeat comparable units. This tweak emphasized higher health values, promoting a more nuanced approach to engagements. Furthermore, build times were equalized more across infantry and vehicle classes, aligning with player feedback to enhance satisfaction. In addition, the playtesting sessions not only identified areas for improvement but also sparked ideas for new units. These player-generated concepts were carefully considered, leading to the implementation of the Tier 2 vehicles that enriched the strategic diversity of the game. These additions were a direct result of player feedback during testing.

Base Building Rework

In Project Flex's original concept for base building, the only method of creating new factories was to find build spots gathered around the map and capture them, forcing the player to engage with the map in order to expand their production. In addition to this, there were no resource generation building options, because CPU was the only resource players had to collect. The original design with build spots was implemented with the intention of providing structure to base building. However, playtesting quickly revealed that this approach stifled players' strategic creativity and limited their ability to adapt to evolving battlefield dynamics. The predefined build spots and limited resources to utilize imposed a rigid framework, inhibiting players from exploring innovative base layouts and hindering the organic development of their military infrastructure.

Player feedback during the playtesting sessions became a compass pointing towards a more player-centric approach. The consensus emerged that the build spots method felt restrictive and was counterintuitive to the dynamic nature of real-time strategy gameplay. This prompted a reevaluation of the base-building mechanics with the overarching goal of empowering players to craft bases that aligned with their strategic vision. In response to these insights, a pivotal decision was made to transition from the constraining build spots model to a more traditional RTS approach—free building. This paradigm shift marked a significant departure from the predefined spots, granting players the freedom to place structures wherever they deemed fit. The change was transformative, unlocking strategic potential and encouraging players to experiment with diverse base layouts that suited their preferred playstyle.

The introduction of free building brought a surge of creativity to Project Flex. Players could now experiment with unconventional base designs, adapt to varying map conditions, and develop strategies that leveraged the terrain to their advantage. The newfound flexibility elevated the strategic depth of the game, fostering an environment where players were no longer bound by predetermined locations but could instead tailor their bases to suit the ever-changing dynamics of the battlefield. In addition, the addition of the two other resources, Power and RAM, allowed for a great diversity in unit build costs, allowing the addition of the T2 vehicle units that cost not only CPU but also Power, a resource previously limited to building construction alone. While the shift to free building liberated strategic creativity, it also required a delicate balance to prevent potential abuse. To address this, constraints were introduced based on construct range, ensuring that while players had the freedom to build anywhere, they still had to consider the proximity of structures to maintain a cohesive and strategically sound base.

The transition to free building, informed by player feedback and iterative design principles, had a profound impact on the overall player experience. **By embracing a continuous feedback loop, the development team responded dynamically to player insights, refining the base-building mechanics iteratively.** This player-centric approach aligned seamlessly with the Spiral Development methodology, where each cycle of feedback and refinement contributed to the game's evolutionary process. The shift from restrictive build spots to free building not only addressed player concerns but also enhanced the overall strategic depth of the game. Spiral development's emphasis on flexibility and responsiveness to user input became evident as Project Flex evolved organically in response to player experiences, fostering a more engaging and player-driven real-time strategy experience.

D Term Playtesting

Restoring Cut Content

In the final stages of Project Flex's development, significant attention was given to reintroducing features that had been previously cut, aiming to enrich the gameplay experience. One of the main highlights was the reintroduction of vehicles, which underwent rebalancing and role adjustments. Unlike before, vehicles now required power and multiple RAM, altering their strategic importance within the game. Feedback from playtesters was positive, with many appreciating the newfound balance between infantry and vehicles, leading to a more varied and engaging gameplay experience. It also helpfully mitigated the strength of vehicle rush strategies, as players could no longer stockpile resources as effectively without crippling other aspects of production.

Despite the success of reintroducing vehicles, efforts to restore AI-controlled skirmish units' ability to construct buildings proved more challenging than anticipated. While initially intended, technical complexities during implementation led to the feature being put on hold once again. However, the idea of AI construction remains a consideration for future updates, with potential solutions including implementing a system where AI opponents "cheat" by spontaneously generating buildings on set timers—a method commonly seen in popular RTS titles like StarCraft.

In summary, the playtesting phase focused on reintegrating cut features provided valuable insights into gameplay balance and player satisfaction. By carefully refining and readjusting previously removed elements, Project Flex was able to offer a more diverse and engaging gaming experience, laying the foundation for a successful final release.

New Features

Also included in this term was a significant effort to incorporate new player-accommodating features based on player feedback and requests. Among the most prominent requests from players was the desire for a minimap and hotkeys to streamline map navigation and unit management. The implementation of the minimap proved to be highly successful, significantly enhancing the ability to deploy units across multiple locations without diminishing the importance of player skill in analyzing the map and predicting strategic movements.

However, the reception to hotkeys was more mixed. While hotkeys for assigning units to command groups and panning to the closest unit or structure were implemented, they were not as well-received as initially anticipated. This lukewarm response was attributed in part to the addition of the minimap, which already streamlined map navigation to a considerable extent. Further playtesting is deemed necessary to determine whether the inclusion of hotkeys enhances or overly complicates the player experience.

Overall, the focus on making Project Flex more accessible and user-friendly for casual players was largely successful. By listening to player feedback and implementing requested features, the development team was able to enhance the game's accessibility without compromising its strategic depth or the impact of player skill. Moving forward, continued playtesting and iteration will be essential to ensure that new features enhance the overall gameplay experience in a meaningful way.

Post Mortem

Jason Asidi

What Went Right

Throughout the development journey of Project Flex, I encountered numerous hurdles and triumphs that shaped the way I approached the project. One of our most significant achievements was the creation of a fully functional game by the project's completion. This milestone stands as a testament to our perseverance and dedication to the project's success. I am particularly proud that we reached a technically "finished" build halfway through A term, showcasing the benefits of adopting a spiral development approach. This early milestone not only validated our development process but also provided us with a solid foundation from which to iterate and refine our game further.

Despite the inevitable setbacks and challenges we faced along the way, we managed to create an experience that resonated with players and ourselves alike. The journey was not without its difficulties, there were many significant obstacles both expected and unforeseen. However, our ability to overcome these challenges and adapt to unforeseen circumstances demonstrated our resilience and commitment to completing this project. Despite the setbacks, the satisfaction of seeing players enjoy our creation made every moment of struggle worthwhile.

Looking back on our project, I realize that it has given me invaluable experience for future endeavors. Working within the constraints of time and resources taught me valuable lessons in project management and resource allocation. The skills and insights gained from navigating these challenges will undoubtedly serve me well in our future endeavors as I enter the game development industry full time. While the journey was challenging, it was also immensely rewarding, leaving me with a sense of accomplishment and a wealth of knowledge to carry forward into future projects.

What Went Wrong

During the development of Project Flex, the team encountered various challenges that impacted the project's progress and outcomes. From poor communication to poor planning, several factors combined to result in delays and compromises that meant the project wasn't able to live up to its full potential.

One notable issue had been the occurrence of several issues stemming from poor communication with both the project advisor and teammates. In the iterative and fast-paced environment of Spiral Development, effective communication is paramount to ensure that everyone remains aligned on project goals and milestones. **As deadlines approached, inadequate communication between team members and the project advisor led to misunderstandings regarding project priorities and progress updates.** This lack of clear communication ultimately resulted in delays and forced us to adapt timelines and expectations accordingly. The lack of clear communication can lead to misunderstandings, delays, and ultimately, hinder the progress of the project.

Another challenge encountered during the development process was the significant amount of time I spent grappling with third-party plugins. **While these plugins offer convenient solutions for certain functionalities, in a larger team setting, coding behaviors and tools from scratch may have been a more optimal approach.** Custom-built solutions would have allowed for greater adaptability and customization tailored specifically to the game's requirements, potentially reducing the time and effort spent troubleshooting compatibility issues or limitations of third-party tools.

Furthermore, halfway through the development cycle, the project transitioned to a solo endeavor, further exacerbating the challenges caused by poor communication. With the project now solely managed by myself, the workload increased significantly, necessitating adjustments to the project scope and expectations. Even worse, this happened just as I was starting a new full time job, so my ability to both work on the project and conduct playtesting was greatly reduced. In addition, as I had to return home to begin work, I had much less availability to the student body for playtesting during the last stretch of development, and had to lean even more on my group of colleagues for most of my playtesting. As a result, there was no chance that we'd be able to replace the placeholder assets in the build, and I had to scale down the scope of both the unit and enemy rosters and enemy AI. These adaptations were essential to ensure that the project remained feasible within the constraints of the revised development timeline.

Overall, the problems we faced during development can largely be traced to both poor communication between all members involved with the project and lethargic adaptation to changing circumstances. Despite the obstacles encountered, my ability to adapt and make necessary adjustments ensured that the project remained feasible within the constraints of the revised development timeline. Moving forward, these experiences will serve as valuable insights to improve my communication practices and streamline my development workflow for future projects.

What We Learned

Reflecting on the development journey of our project, several key lessons have emerged that will undoubtedly shape how I approach future endeavors. One crucial takeaway is the importance of prioritizing functionality over perfection. While a modular system is advantageous, dedicating excessive time and resources to refining intricate features may not always yield the best results. Instead, I learned the value of allocating our efforts towards further playtesting and addressing features that resonate most with our audience. This is always something I've struggled with personally due to some of my perfectionism in wanting to have a system that can do everything I'd ever want it to do, but this project has really showcased how damaging this can be when working on tight deadlines and minimal resources.

Another vital lesson learned is the necessity of flexibility and adaptability in my development approach. **It is crucial not to become entrenched in a particular method or tool if it proves to be a hindrance to progress.** I discovered that being willing to switch approaches or tools when faced with obstacles can lead to more efficient problem-solving and ultimately accelerate development. Having faced numerous issues with the 3D pathfinding plugin used during the project made me with I had buckled down and simply made my own from scratch that would fit all my needs from the project back from the very beginning, and in the late stages I did not appreciate several of the compromises I had to make to get the unstable pathfinding system working again after there was a big update halfway through D term development.

Furthermore, effective communication emerged as a cornerstone of successful project management. I realized that maintaining constant communication, even when faced with challenges or setbacks, is essential for keeping all stakeholders informed and aligned. Professor Yarbrough notes here, his belief that all "Production is Communication." By providing regular updates and addressing concerns promptly, it’s much easier to mitigate potential tension within the team and foster a more collaborative and supportive environment.

Finally, I gained a newfound appreciation for the value of playtesting and user feedback. **Embracing the mantra of "test early and test often," we learned that involving playtesters from the early stages of development can provide invaluable insights and perspectives.** Regardless of the refinement level of a build or feature, the diverse viewpoints of playtesters can uncover unforeseen issues and inspire innovative solutions to existing problems. Moving forward, I’m going to try to integrate playtesting into my development process as a fundamental component of creating compelling and engaging experiences for the intended audience.

Works Cited

1. Casteel, B. “Wayward.” (2015, March 18). *Random thoughts on resource management in RTS*. Wayward Strategy. https://waywardstrategy.com/2014/12/18/random-thoughts-on-resource-management-in-rts/
2. Casteel, B. “Wayward.” (2021, June 19). *Some thoughts about the “early game” phase of RTS*. Wayward Strategy. https://waywardstrategy.com/2021/06/18/some-thoughts-about-the-early-game-phase-of-rts/
3. Doll, T. (2020, November 24). *Time as a resource part 1: Single Player Map Design*. Wayward Strategy. https://waywardstrategy.com/2015/05/26/time-as-a-resource-part-1-single-player-map-design/
4. Doll, T. (2020, November 24). *Time as a resource part 2: Multiplayer map design*. Wayward Strategy. https://waywardstrategy.com/2015/06/07/time-as-a-resource-part-2-multiplayer-map-design/
5. Doll, T. (2020, November 24). *Unit Design, clarity of roles and redundancy*. Wayward Strategy. https://waywardstrategy.com/2018/05/17/unit-design-clarity-of-roles-and-redundancy/
6. King, D. (2021, December 9). *Thread by @delaneykingrox on thread reader app*. Advice for Spiral Development in Game Development. https://threadreaderapp.com/thread/1468804328857038849.html