MQP MBJ-1701 & 1702 / JRW AA6L

VELUM:

A 3D Puzzle Game and Facial Analysis Study

A Major Qualifying Project Report submitted to the faculty of
Worcester Polytechnic Institute
in partial fulfillment of the Degree of Bachelor of Science

Nicholas Chaput, CS
August Beers, IMGD/CS
Connor Mattson, IMGD
Connor Thornberg, IMGD
Jie Weng, IMGD

Submitted April 27, 2017

Advisors:
Professors Brian Moriarty and Ralph Sutter, IMGD
Professor Jacob Whitehill, CS

Abstract

Velum is a first-person 3D puzzle/exploration game set in a timeless version of the Boston Public Garden. The project's narrative framework and aesthetics are based on one of the Garden's most prominent features, the Ether Monument, which commemorates the 1846 discovery of diethyl ether's effectiveness as a medical anesthetic. A sequence of nine abstract challenges is rewarded by a progressive revelation of the player's mysterious identity and purpose. The puzzle design was informed by the use of crowdsourced playtesting involving 300+ volunteers, combining standard data telemetry with AI-based facial image analysis capable of mapping player emotions to gameplay events.

Acknowledgements

Our team would like to extend a special thanks to the following individuals for their continuous support throughout our project.

Project advisors

- Brian Moriarty, Professor of Practice, IMGD
- Ralph Sutter, Instructor / Lecturer, IMGD
- Jacob Whitehill, Assistant Professor, Computer Science

PAX East IQP team

- Dean O'Donnell, faculty advisor
- Eric Cerini, Matthew Szpunar and Kelly Zhang, coordinators

Special thanks

- John Thornberg
- Mike Voorhis

Contents

Table of Figures

1. Introduction

Velum is a first person 3D puzzle game set in a timeless version of the Boston Public Garden. The game was developed using Unreal Game Engine 4 over a year by our project team. To drive the design process, the team developed *Eukaryote,* a 2D browser-based prototype of *Velum's* puzzles which was utilized to run a crowdsourced study of user's facial expression during gameplay.

Some members of the team first came together at the MQP Project Pitch for the IMGD Department. We were all very interested in one of the pitches presented: a first-person game, set in the Boston Public Garden, in which you had to capture an elusive Irish creature called a *fear dearg*. Many of us had worked together on previous projects, so when we discussed the pitches afterwards, we decided to contact Professor Brian Moriarty to form a team for the project.

At that time, our team consisted of three artists (Connor Mattson, Connor Thornberg and Jie "Stan" Weng) and one programmer (August Beers). However, August needed to complete a project to satisfy degree requirements for both IMGD and Computer Science. At this point, we set out to find an additional advisor in the Computer Science department, and a teammate to help with the additional technical challenges. August suggested Nick Chaput, a Computer Science student with an informal background in games, and we welcomed him aboard.

Professor Moriarty suggested that we invite Ralph Sutter to the team, as an advisor to the artists, and the art team agreed. After August and Nick contacted many potential CS

advisors, they eventually found Jacob Whitehill: a first year professor with a novel idea to use an Artificial Intelligence method to conduct facial analysis of users completing mathematical tasks. We realized that we could perform a similar study on players of our puzzle, with the goal of measuring emotional experience goals quantitatively. Professor Whitehill agreed to join the team, and began attending our weekly meetings during the beginning of the 2016-2017 academic year, finalizing the team until the addition of sound engineer Marco Duran during the next semester.

When approaching the design of the project, the team looked both to the games we love and to previous successful Major Qualifying Projects we identified. It was critical that the project could be the best portfolio piece possible for each of our individual goals. In general, we agreed that it would be better to make a small, highly polished game with a few key features, rather than a large game with a diverse set of features. Eventually it was decided that a puzzle game with a small, but immersive environment would provide a scoped design goal for the project.

Evolution of *Velum*'s design ended with the following dramatic sequence: The player enters the Boston Public Garden on a swan boat, and must complete puzzles to explore further areas of the park. Ultimately, it is intended that the player will discover why they are in the park in the first place. The game unfolds linearly, and gives players more places to explore and secrets to uncover as they progress.

We eventually decided that we wanted the completion of each puzzle in to influence how the player experienced the environment. After brainstorming many different ways of interacting with the park, we fell in love with one idea: design a space such that different parts

of the Garden appear as the player solves a progressive puzzle sequence. This allowed us to control what the player was allowed to interact with at different stages, while keeping them immersed in a fantastical, mysterious park environment.

August and Nick decided to pursue an analytical study to both demonstrate computer science and to help improve *Velum's* design. Professor Whitehill had previously developed software that can analyze images, extract facial data, and recognize facial engagement of subjects. August and Nick theorized that we could analyze this emotional data to provide new insights into the player experience, and help us intelligently improve our puzzle designs. We decided to create a mechanic that could function in the 2D grid based engine Perlenspiel for ease of analysis, among other reasons. We eventually realized that we could distribute a Perlenspiel based study online, through the Amazon Mechanical Turk service. We incorporated this into our study design, and developed a series of levels in Perlenspiel both as the original prototype of *Velum* and as a tool to conduct online testing.

The *Velum* team also recognized that sound design would be an integral component of our immersive environment. Several of us had limited background in audio, but we felt that it would benefit the project more to invite a specialist to do this work. We contacted another IMGD senior, Marco Duran, to do sound design for *Velum* as an independent study project. He cultivated the game's entire soundscape and worked with us to implement it over the course of a seven week academic term.

The narrative of *Velum* was not fully realized until late in the project. The team recognized the story of "Ether Day" and its connection to the Public Garden as the potential basis of a compelling narrative. With this realization in mind, the team was driven to write a

narrative script (see Appendix A). The script contains a mixture of direct quotes from the Ether Day operation and fictionalized lines. These lines correspond to three characters: the two doctors, John Collins Warren and William Morton, along with Gilbert Abbot, the patient who underwent the history-making surgical procedure. The narrative audio script is accompanied by a series of distorted photographs from the original operation (all sourced from the public domain). The dramatic story is revealed piece by piece to the player and explains the mystery of their appearance in the park.

This report will explore the various components of our development process and why we made certain decisions. It will include discussions of the various aspects of game design, including puzzle design, artistic design, game programming, and others. The report will also discuss our playtesting sessions, and the design and results of the facial analytics study.

2. Development

Development of *Velum* began in the spring of 2016, and continued for all four terms of the academic year. During this period, the design underwent a number of expansion phases, summarized in Figure 1.

Phase	Terms	Duration
Early	D16, A16	3/16 – 10/16
Middle	B16 – C17	11/16 – 2/17
Late	D17	3/17 – 4/17

Figure 1. Team schedule.

The primary task of the *Velum* team was to design a compelling 3D puzzle/exploration game. The project was divided into two major components: a game offering a complete and polished experience, and an analytical study to support the design of the game.

Our process spanned the entire development cycle, from conception through testing and polish, incorporating elements of design, writing, visual art, computer science and sound design.

2.1. Game design

2.1.1. Theme and inspirations

The setting of *Velum* was inspired by the original pitch, proposed by Professor Moriarty. In its first iteration, the game revolved around capturing a mythical Irish creature called a *fear dearg* ("green man"), a malevolent relative of the leprechaun. One of these creatures had become trapped in the Boston Public Garden, and was causing mischief there. Professor Moriarty's design intentionally left room for the team to make its own creative decisions regarding the game's direction, including the option to eliminate the *fear dearg* entirely.

We began to refine our ideas and identify key design issues, moving forward with certain elements that we felt complimented the setting, combined with our own inspirations and ideas. We compiled a list of games, movies and other media that we felt would exemplify our game's general atmosphere (Figure 2), and watched them over the summer break.

Games	**TV shows**	**Films**
Firewatch	*The Twilight Zone*	*Blue Velvet*
Kentucky Route Zero	*Night Gallery*	*Shutter Island*
The Witness	*Twin Peaks*	*The Coherence*
Myst	*Black Mirror*	*Eraserhead*

Figure 2. Inspirations for *Velum*.

We liked the idea of having a secluded game space filled with puzzles, a structure employed by both *Myst* and *The Witness*. We also wanted to convey a feeling of isolation, and have players wondering why they are there (*Shutter Island)*. Finally, we wanted our setting to feel uncanny in a distinctive but hard-to-pinpoint way, an effect achieved by all of the TV shows on our viewing list.

The first major change we made was to remove the *fear dearg's* capture as the game's primary motivator. We all agreed that the concept of organic growth would be more flexible and compelling, and made this the focus of our design. While researching this concept, we began to look into neurons. We found videos of neurons forming connections in the brain, which intrigued us greatly. A particular object of this study involved the stimulation of stem cells with the alkaloid harmine (Figure 3), a process which prompts the cells to grow into neurons instead of the variety of other cells they might otherwise become (Dakic, 2016). These topics became the inspiration for our game's basic puzzle mechanic: the formation and growth of neuron-like connections.

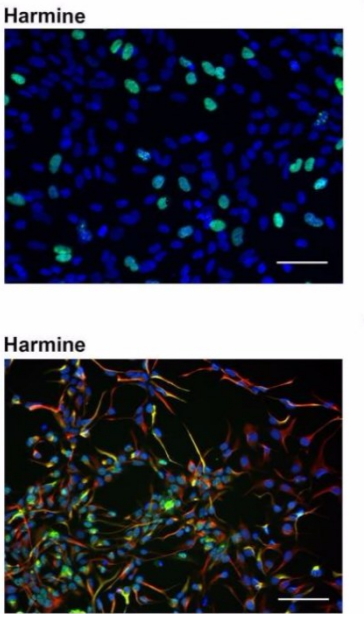


Figure 3. The response of stem cells when stimulated with harmine (Dakic, 2016).

The game's design evolved to reflect this concept on a higher level. When the player enters the game, there are no "connections" available between various sections of the Garden, so the player cannot access them. The player must form these connections themselves by solving puzzles. Structurally, the Garden models a network of neurons, a concept which proved highly appropriate to our game's psychological context (explained in more detail below).

Figure 4. August Beers exploring the Boston Public Garden. Photo by Connor Thornberg.

Despite these inspirations, we still needed to devise a reason for the player to be in the Boston Public Garden. We decided to investigate the physical landscape of the actual location for ideas. The team visited the Garden on several occasions for this purpose (Figure 4).

The Public Garden is host to a variety of monuments and memorials. One in particular impressed us. The Ether Monument looms over visitors with stoic grandeur. Snarling lions at its base guard a solemn pedestal of multi-colored marble and granite, adorned with scenes of human suffering and compassion, all crowned by a dramatic sculpture of the Good Samaritan (Figure 5). We agreed that this monument had to be included in our game, and began discussing how it might be integrated into the project. Professor Moriarty encouraged our interest, and provided us with a copy of Julie Fenster's *Ether Day,* a book which vividly describes the event leading to the monument's creation: the first recognition of diethyl ether's ability to relieve surgical pain, which took place at the nearby Massachusetts General Hospital on October 16, 1846.

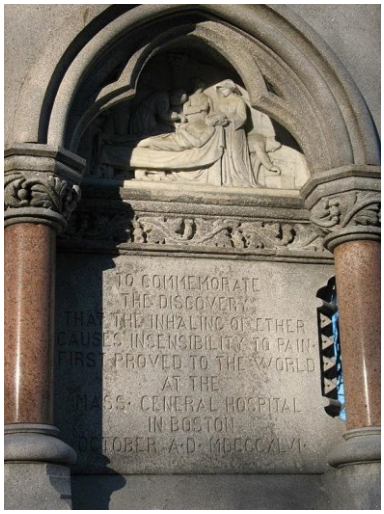


Figure 5. One side of the Ether Monument. Source URL.

Before this colossal discovery, surgery was a painful, bloody and almost unimaginably gruesome procedure. Ether anesthesia ushered in a new age of modern medicine, and an improvement in the quality of human life difficult to overestimate (Fenster).

Inspired by the Ether Monument, we reimagined our game as taking place within the mind of a surgical patient under the influence of ether. Such patients often reported vivid, dream-like experiences (Fenster), and we realized that our otherworldly version of the Boston Public Garden would be a fitting place for such a patient to find themselves in while anesthetized. We eventually developed a scenario in which players would enter the Garden on one of its iconic swan boats, figuratively explore both the setting and their own minds as they solved our puzzles, and leave on a swan boat as their anesthesia begins to wear off.

The story of the first ether operation became the framework of our game's narrative. The player assumes the role of the original patient, Gilbert Abbott, after he has fallen unconscious. Progressing through the game's puzzles, visual and audible aspects of the real-world surgery occasionally break through the veil of anesthesia. As the ether wears off, these breakthroughs become more insistent, eventually signaling the player that the game has come to an end.

2.1.2. Experience goal

Having determined the setting of our game, the team was able to identify an overarching experience goal: epiphany. We agreed that players should eventually reach a peak moment: the realization that they are playing the role of a patient under ether anesthesia. Although this goal always informed our design, we followed a methodology whereby the experience was driven by the design; not the other way around.

During the design process the mood and feel of the park was developed to support our experience goal. In order to concretize our experience goal the team had a brainstorm for some aesthetic vocabulary that describes the mood of our game. Some terms included "hazy,"

"eerie" and "ethereal," among others. This vision was instantiated in the ghostly lighting which infuses the entire game. At the suggestion of Prof. Sutter, we decided to flood the entire park in an orange ether-like ocean, further accentuating the aura of mystery and surrealism.

We also want players to feel "suspense and mystery" as they progressed through the story. By exposing the sections of the Garden once at a time, we let the player slowly discover clues about the locale, and the reason their avatar is inhabiting the gamespace. It was assumed that, upon entry, most players would not have any preconception of the story behind the game. It is up to the player to complete puzzles in order to reveal their situation.

Figure 6. Screen capture of *Velum* during early development. Image by Connor Thornberg.

2.1.3. Narrative design

The narrative design of *Velum* was iterated upon many times throughout the development process. We knew that it needed to convey to the player the reason that they had come to the Garden, since it is such a specific location. We brainstormed many different ideas to justify the player's presence, and eventually agreed upon the ether dream setting.

At first, we imagined the dream as somewhat generic, in which the player would see the shadowy figures of other anesthetized patients wandering throughout the park. This remained our plan for a large portion of early and mid-development, and impacted our first level iterations significantly (Figure 6). However, our struggle with finding a way to convey this somewhat vague context finally led to the decision to make player an avatar for Gilbert Abbott.

To align this choice with the structure of our gameplay, we decided to reveal Abbot's (and the player's) situation through nine distinct "visions." We wrote a script and recorded voice acting for three characters, augmented by period photographs to make it appear as if Abbot's historic surgery is occurring just beyond the veil of the anesthetic.

The player first meets John Collins Warren, the senior surgeon who directed the Ether Day operation. The player later hears Warren greet his partner William Morton, a rather infamous dentist who pioneered the use of ether and invented one of the first devices for administering it. At the end the player is introduced to Gilbert Abbott, whose dream they have unwittingly been controlling throughout the game.

Despite some of the more upfront narrative aspects of the design, it is possible that some players may never fully comprehend their identity or purpose in the Garden. This tradeoff was deemed acceptable in order to avoid compromising the experience goal of the design.

2.1.4. Puzzle design

Early in the design process of *Velum,* the question of including gameplay or not became a concern of the team. While a barely-interactive "walking simulator" would have sufficed as an art portfolio piece, the team felt driven to create a more conventional gameplay experience, particularly as some team members hoped to explore design in their future careers. The team was excited by Professor Moriarty's idea to employ a puzzle-based design for several reasons. Nearly every member of the team had previous experience building a puzzle game. More importantly, solving a puzzle requires a moment of realization, which aligned with our general experience goal of epiphany. Finally, the team anticipated that puzzle mechanics could easily be incorporated within the scope of our project.

Once we had settled on puzzles as our primary gameplay mechanic, we began to consider potential puzzle designs. The search soon narrowed into two domains. The first domain included modular puzzles typical of an arcade setting, in which levels are presented to the player sequentially, and can be implemented with little or no dependency on their surrounding environment or narrative. The second domain included puzzles which are deeply integrated into their environment and/or narrative.

The team eventually decided on arcade-style puzzles, because they could more easily be prototyped and tested. A modular design would enable a simplified 2D version of the game to be developed and distributed in a browser, which would facilitate large-scale testing. The team's final puzzle designs, and some steps along the journey of their creation, are described below.

The core puzzle mechanics of *Velum* were inspired by Conway's *Game of Life* and a variety of traditional puzzles. The game board is defined by a 2-dimensional grid, an example of which seen in Figure 7. Within this grid, puzzle pieces can be placed at specific coordinates in order to connect tiles that are explicitly marked as goal nodes.

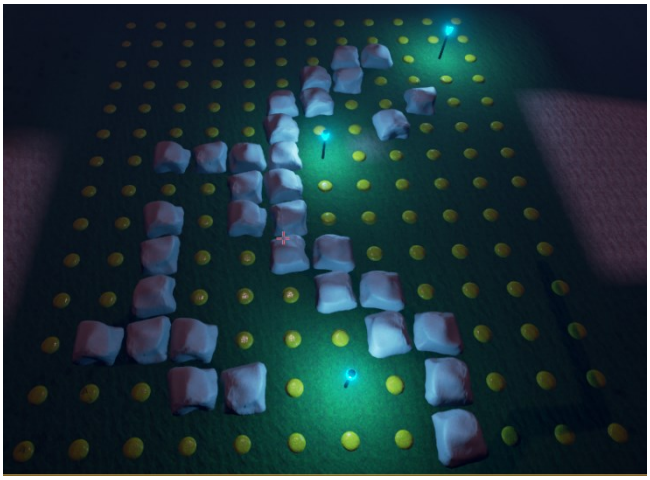


Figure 7. A *Velum* game board. Image by August Beers.

Once the pieces are placed, the player can test their solution by pressing a button which begins an automated simulation process, in which each piece attempts to occupy adjacent grid spaces by following a growth pattern unique to that puzzle. The puzzle is completed when all of the

goal nodes become connected via a network of grid pieces.

This puzzle design is strongly suggestive of our previously-identified neuron theme. The goal nodes represent neurons that need to be connected. The pieces placed on the grid act like synapses, forming a network between the neurons.

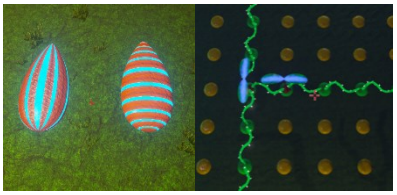


Figure 8. Horizontal and vertical seeds. Image by August Beers.

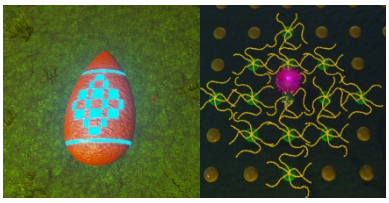


Figure 8: Expansion seed. Image by August Beers.

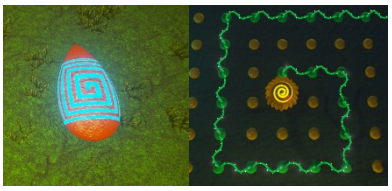


Figure 10. Spiral seed. Image by August Beers.

To make these rather abstract concepts fit into the environment of the Garden, we represented the goal neurons as glowing "lamps," the placed pieces as "seeds" which sprout into plants, and the network connecting them as "roots" growing from the plants.

There are four different seeds that can be "planted" on a grid to make connections between lamps. Each seed features a distinctive shell marking to identify its growth pattern, and a corresponding plant model to represent it when it sprouts.

The first two seeds are identical in function, but differ in orientation (horizontal and vertical). Each generates a line of roots, growing outwards in both directions from its planted sprout until their path is blocked. These seeds are identified by a pattern of lines on their shells, drawn in the direction that their roots will grow when sprouted (Figure 8).

The third seed is a bit more complicated. When sprouted, its roots expand in all directions, but only to a predefined limit of grid spaces. Its growth in any direction may be stopped by obstacles in the grid, including rocks and other barriers, as well as roots already grown by other sprouts. This seed is indicated by a circular pattern on its shell, with lines

suggesting the potential for blockage (Figure 9).

The fourth and final seed creates a spiral root pattern in the grid. Its path starts by going up, then right, down, left, up again, etc. Unless blocked by an obstacle, this seed's roots can potentially occupy more grid spaces than any other (Figure 10).

Figure 91. Zapper lantern.
Image by August Beers.

The last element of Velum's puzzle design is a special "zapper lantern" (Figure 11). If a growing root attempts to occupy a grid point adjacent to a zapper, all plants and roots on the grid are immediately removed, effectively resetting the puzzle. Unlike the seeds, zappers are embedded in their grid, and cannot be moved by the player. They serve as an obstacle which players must avoid in order to complete a given puzzle.

While many other puzzle mechanics and designs were considered, the team found that these simple components were sufficient to construct a nine-level puzzle progression rich enough to provide a satisfying experience.

To produce the nine levels, the team used a level-building tool developed in Perlenspiel, a Web-based game engine designed to allow users with no technical background to rapidly prototype abstract designs (Figure 12). This tool,

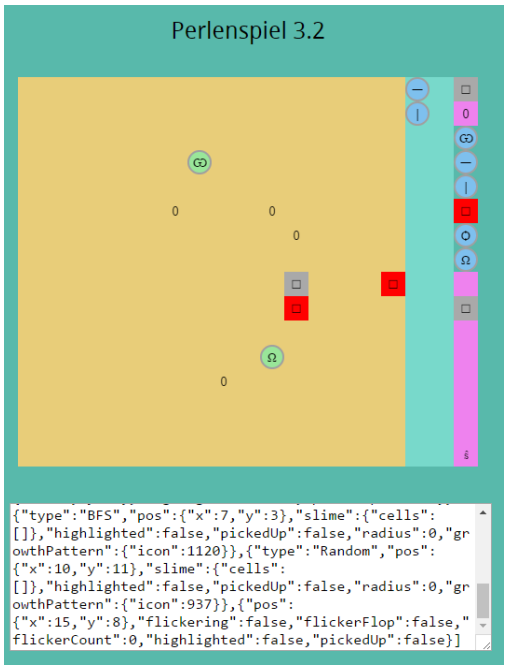


Figure 102. Level building tool.
Image by August Beers.

which we called *Eukaryote,* quickly proved its value, as it was used by every member of the team to brainstorm level ideas and finalize our puzzle progression.

2.1.5. Player interaction

One of the challenges we faced with *Velum* was teaching players how to interact with our puzzle system. In a 2-dimensional setting, the rules of interaction are simple and easily discoverable. When it came time to reproduce the puzzle in a 3-dimensional setting, a series of subtle design challenges arose. Our final design solution maintained the 2-dimensional nature of the puzzles by placing the grid on the lawn of the Garden. Players interact with the grid from overhead by climbing a special tree assigned to each puzzle.

Figure 113. *Eukaryote* gameboard. Image by August Beers.

The fundamental building blocks of *Velum*'s puzzles include a game board and a set of board pieces intended to be taken in and out of it. The circular pieces marked with glyphs can be dragged and dropped to any given grid space using the mouse. Once a given solution is configured, the player may test a solution by pressing the go button on the bottom right. After the go button is pressed, simulation begins and pieces occupy grid spaces by marking them green. A path of green square must connect the two 0s on the board for the puzzle to be completed (Figure 13).

In *Velum*, the same 2-dimensional grid is projected onto the lawn of the park. Empty grid spaces are marked by yellow dots, with barriers and walls marked by rocks. The goal nodes from *Eukaryote* appear as blue-glowing lamps protruding from the grid. Unlike *Eukaryote*, *Velum*'s puzzle pieces are not embedded in the grid. They are represented as seeds which are place on the platform in the tree overlooking each puzzle grid. When dropped into the grid by the player, seeds instantly germinate and sprout into a plant, one of which can be seen as a purple plant in the grid in Figure 9. This arrangement enables functionality similar to the drag-and-drop mechanism in *Eukaryote*, repurposed to work in a 3-dimensional environment.

2.2. Visual art

Before the art team started to make any assets for the game, we first went to the Boston Public Garden to inspect the environment and get inspiration. This visit allowed us to better understand the atmosphere, scale, and sense of space in the garden (Figure 13). It also allowed us to find intriguing landmarks that were essential to the park, and helped us to develop our ultimate art style.

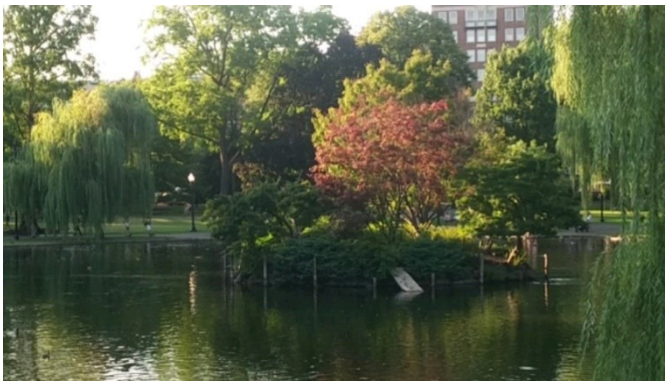


Figure 12. Island in the Boston Public Garden lagoon. Photo by Connor Thornberg.

We decided that it would be in our best interest to develop part of the garden, or a smaller version of it. This would allow us to maximize the quality of each asset, while keeping the overall scope down. However, this meant that we first had to brainstorm about which of the Garden's many features we wanted to put in the level. We identified the most iconic landmarks of the park and others that we found interesting. When comparing the locations of these landmarks, we found that the most important ones all resided on the northern side of the bridge. We decided that we could isolate this half of the park, and have the player enter from the "other side," with the

Figure 15. *Velum* concept art. Image by Connor Mattson.

double meaning of the park and the ether dream. Since the bridge bisects the park, it felt fitting for it to act as a portal of sorts.

Figure 136. Realistic style. Source URL.

The team decided to create a stylized version of the Boston Public Garden, as shown in the concept art above (Figure 15). To come up with a unique vision of the Boston Public Garden, we first compared several different artistic styles we found compelling. We compared highly realistic styles (Figure 16) with more cartoonish, abstract styles (Figure 17). We eventually decided to combine the best of both styles: low polygon models with visually intriguing, high resolution textures.

Figure 15: Cartoon style. Source URL.

To support the main theme of ether in our game, we chose to make orange one of the dominant colors in the game's art. To create the general color palette, shown in Figure 18, we selected a compelling hue of orange and other colors that we felt would complement it well. We also considered our decision of a timeless-twilight setting when choosing these colors. Ultimately, we came up with a landscape dominated by orange and purple, dotted with highlights of yellow, blue, and red, and supported by dark green.

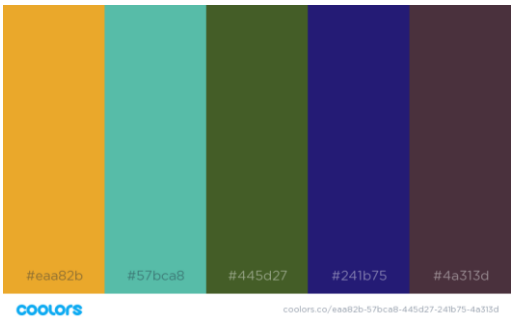


Figure 148. Color Palette for *Velum*. Sourced from coolors.com.

Next, the team cut the garden into different sections, referred to as “puddles,” based on a satellite image taken from Google Maps. We made an asset list based on what features we wanted to include in each puddle, including the different species of trees. We decided to cut up the garden so that we can set up gameplay linearly and avoid optimization problems that might happen in later development.

The team identified the most important landmarks of the park, and theorized how we could involve them in our game. We found that the most iconic components of the park included the Swan Boats, the central bridge, and the George Washington monument. We decided to use the Swan Boats to ferry the player in and out of the park, and deposit them at the dock. The bridge is revealed after the player solves the first puzzle, and the George Washington Monument right after the second puzzle. It was our hope that putting these landmarks in the game early on would help the player to learn that they are in the Boston Public Garden.

We also identified several less iconic monuments that we thought could have a compelling role in our game. The replication of the Ether Monument was obvious, due to our chosen theme. However, it is also a notable landmark to include due to the fact that it was the first public monument to be installed in America. We were also interested in the Japanese lantern by the George Washington statue, and loosely based our “zapper” pieces on the lantern’s design. We also thought that the two fountains in the north-east of the park could be interesting to include, especially when considering our ether theme. When recreating these fountains, we replaced the water with liquid ether and each statuette with an ether globe, the administration apparatus used during the original surgery.

Figure 19. Foliage in *Velum*.
Image by Connor Thornberg.

We recognized that the organic environment of the park was integral to its beauty and atmosphere. We wanted to recreate the organic assets with as much variety as possible. For this reason, we chose to create the trees modularly. Each tree had about four different branches created for it. The branches were then attached to the tree trunk in different configurations. Combining this technique with variable rotation and scale in-engine allows us to produce many believable looking trees. An example of this can be found in Figure 19, which shows our trees rendered in the game.

The workflow for these assets began with ZSpheres, and was then sculpted up to a higher resolution. That data was then projected into a normal map, and used in-engine. The wireframe and normal maps generated in this way were also used in the creation of the color and specularity maps. We used a variety of Photoshop manipulations to take advantage of the normal data, which helped us create interesting looking textures. We also applied a small emissive value to the leaves of the trees, to help them stand out against the dark sky, and make the garden feel more magical.

Figure 20. Squirrel guide in *Velum*.
Image by Stan Weng.

Outside of replicating the Boston Public Garden, we designed and created some other interesting assets to tell the story of *Velum*. A yellow squirrel waits for the player at the beginning of the game as the player's guide, shown in Figure 20 above. Along with this yellow one as the guide, there are also squirrels in blue and green wandering around the game. When we designed the squirrel, we decided that a fantasy-based squirrel would suit our game better

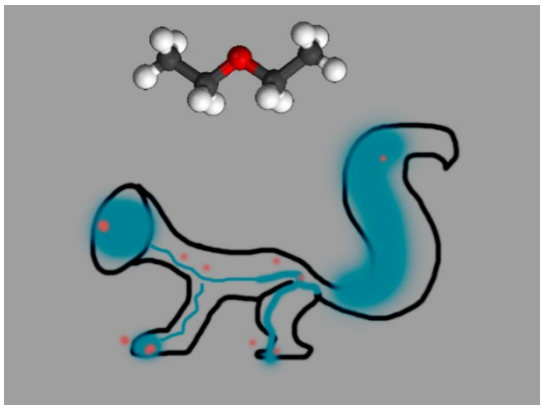


Figure 161: Squirrel representation of the ether molecule. Source URL.

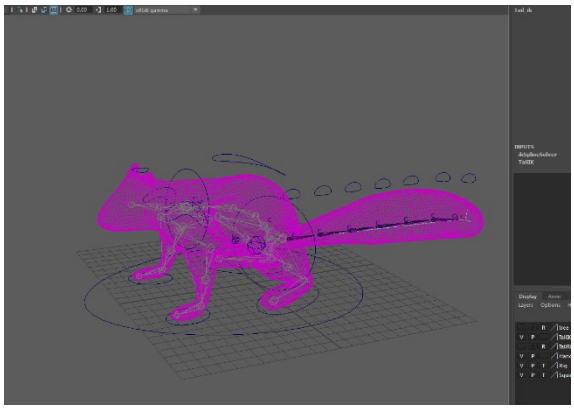


Figure 172: Squirrel rig in Maya. Image by Stan Weng.

than a realistic one. We decided to use transparent skin for the surface, with a glowing skeleton inside the body. The first version of the skeleton mesh attempted to emulate an actual squirrel's skeleton. Later, when the team discovered an image of an ether molecule online, we thought it would be fitting to make a connection between the squirrel skeleton and our ether theme. As a result, the shape of the squirrel's skeleton was changed to a stylized representation of an ether molecule (Figure 21).

Our squirrels are able to idle, run and climb trees, thanks to an AI devised by August. Originally, the team planned to have the squirrel AI capable of climbing a tree by itself. But in mid-development, we decided this AI behavior was too ambitious and out of scope. Instead, the tree-climbing animation was manually assembled using Unreal 4's Matinee tool. We

combined different animation pieces such as “climb from ground to tree,” “climb straight on tree” and “climb diagonally on tree” to make a lifelike motion.

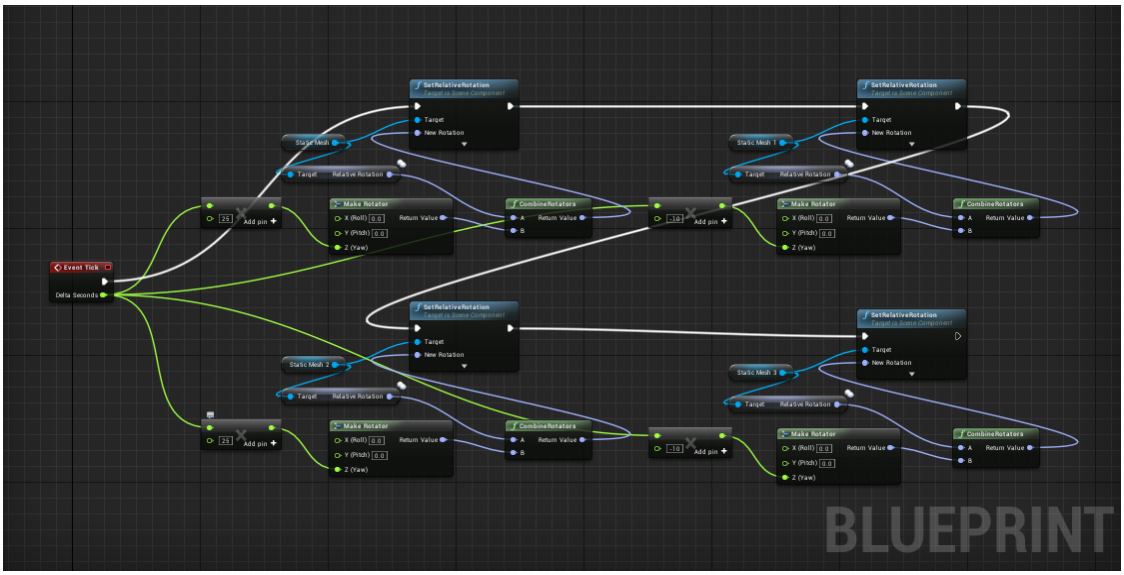


Figure 183. Animation blueprint for visions. Image by Stan Weng.

We realized that the squirrel’s tail was the most characteristic and identifiable part of its animation, because of how much it moves. Consequently, tail motion was the main focus during the creation of the squirrel. As shown in Figure 22, the tail rig was set up straight and controlled by handles that are connected with cluster deformers. We also considered applying a physical material to the tail, but we did not choose to do this because the art team was not very familiar with physical materials, and instead chose to animate the tail handles manually.

To embrace the ether theme, we changed two fountains that exist in the garden into orange ether fountains, each with an ether globe.

To help convey the story, we created visions which utilized both shimmering visuals and reverb-drenched voice clips of the narrative. We originally attempted to implement these visions via 2D animation in Unreal with sprite sheets. This method produced a result that didn’t

have sufficient movement capabilities as well as creating unwanted visual artifacts. We eventually went with a method that involved several images that were collaged in-engine and then animated to rotate via blueprint, as shown in Figure 23. This method provided a constant rate of rotation in the game world, as opposed to flickering through states of a sprite sheet.

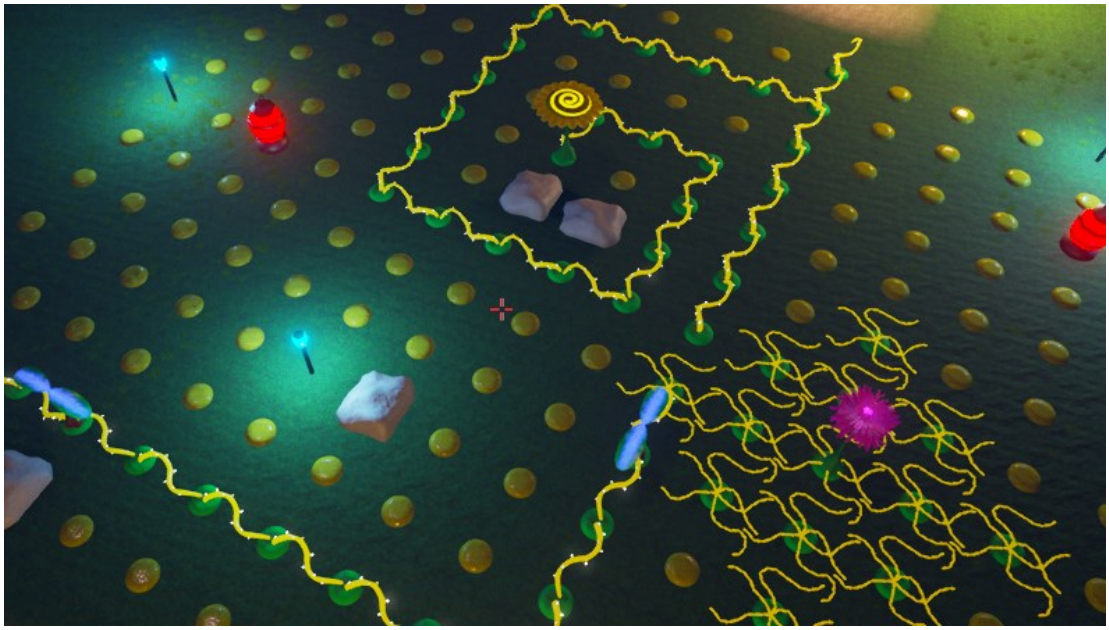


Figure 194. Puzzle growth in *Velum*. Image by Connor Mattson.

Because *Velum* is a puzzle game, we have different objects to represent the puzzle, such as roots and stones, shown as Figure 24 above. We used roots that grow from a plant and connect nodes to convey the path of the puzzle mechanics. There are small glowing orbs on the root design, which are inspired by neurons in human brain, as shown below in Figure 25. Because root animations are designed to convey the puzzle gameplay, each animation should look simple and short. To achieve this, we used a long rig with multiple

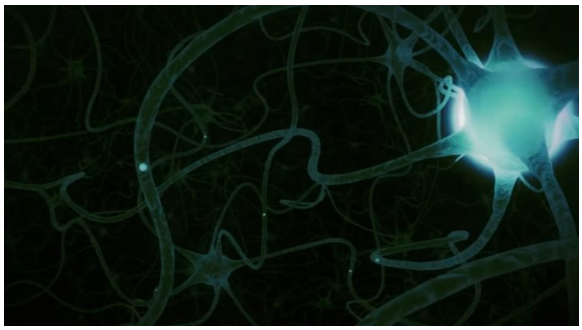


Figure 205. Visual representation of neurons. Source: Athene's *Theory of Everything*. Source URL.

joints of equal length. We found that it was very difficult to animate on the rotation value of each joint. To overcome the problem, we used cones that have the same length as each joint to set up the path first. The tips of each cone snapped to the bottom-center of the next one in a way that recreates a nice, even chain. In addition to the cone path, we gave each single joint an IK handle to control the rotation of that joint (Figure 26). We used this methodology, to make it easy for us to produce variable root animations quickly.

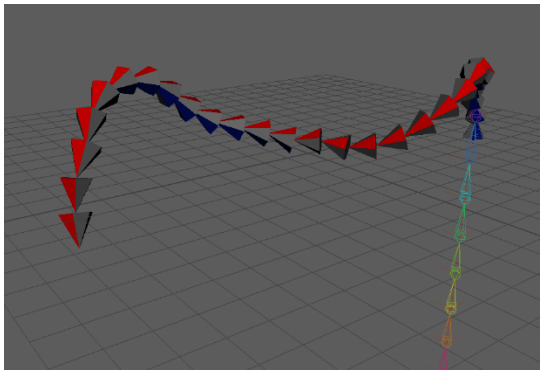


Figure 216. Root chain in Maya. Image by Stan Weng.

Each member of the art team was assigned a different type of asset, and mostly worked on them individually. We realized that time was our biggest enemy when we were making assets, so in the end, we adjusted our workflow and process to fit with our goal of "low poly models, high res textures." When most of the assets were imported into Unreal Engine, the art team came together to work on materials and textures, to make the general atmosphere more harmonious. The artistic result can be seen in Figure 27.

Figure 227. Atmosphere of *Velum*. Image by Connor Mattson.

2.3. Audio and music

During early development, the team determined that it would be important to bring in external talent to do the sound design for *Velum*. No one from the core team felt they would be experienced enough or have the time to do the sound design for the project. For these reasons, we decided to ask Macro Duran if he would be willing to take a term to do a sound ISP and work on our project. Marco agreed to work on the game and proceeded to design and create a soundscape for our game along with nine songs. Marco's contribution to the project is described in detail below.

2.3.1. Artistic focus

Marco joined the team during the mid-development period. Marco intended to design a soundtrack for the game using abstract arrangements of popular classical artists. Bach was identified as a source of compositions for several reasons. The time period that most of Bach's arrangements were created was around the same time as the surgery and is what would be heard in most concert halls in 18th century Boston. The team also felt that the mystery and intrigue of Bach's' work fit perfectly with our game's theme. After listening to many Bach arrangements while studying music theory, as well as listening and performing a concert band arrangement of the J.S. Bach Chorales, Marco had become accustomed to hearing and learning the composition of J.S. Bach. As a tribute to the artist who has sparked his interest in music, Marco felt the need to arrange his masterpieces within a modern light.

When looking for songs to place in the game, Marco spent a good amount of time looking through a database of Bach songs in MIDI format which were free to reuse in the public domain. After downloading numerous songs and listening to the MIDI recordings as well as the

original recordings, Marco narrowed down ten songs: eight from different styles throughout his life and two from the second Brandenburg Concerto. After the songs were chosen, the order of the songs had to be decided based on the story of the game.

In the story of *Velum*, there are three stages where the mood changes: before, during and after the surgery. In the first part of the game, the songs needed an eerie feel, walking tempo and no noticeable delay in the song. The second part is where the player realizes that they are under ether anesthesia. The songs played in the second part of the game have water-like filters, slowed tempos, added delay and wider reverberation. For the last part of the game, the player is waking up, and the final song is played at an increased tempo, with all of the processing effects heard during the surgery turned off.

Ambience while in an altered reality is key to setting the scene of the game. Hearing people talk about their experience while under the influence of ether was a great asset to understanding what a patient under ether could hear. From the information that Marco had gathered, sounds have a much higher pitch with a blend of unrecognizable voices and delays. While creating the ambient loops for the park, Marco added more echo and a slight delay to show that the player is no longer in the real world. When the player reaches the puzzle that is located near the Ether Monument, the ambience of the entire park changes to my interpretation of a person under the anesthetic. With the second ambient track, a filter, pitch increase, and heightened delays were added.

The squirrel is also an important cue for the soundtrack, as it is used as both a guide for the player and also to allow the player to contemplate who they actually portray in the game. When creating the chatter for the squirrels, each bit almost sounds like a small conversation in

another language.

2.3.2. Music selection

Much effort and careful decisions were made when choosing which Bach tracks to include in the game. After a song was selected, it was then modified to suit the needs of the entire arrangement. To give an example of this process the design rationales for two of the tracks created are described below.

One of the first tracks to go into the game was Prelude 1 from *The Well-Tempered Clavier, Book I*. Within this piece, the original composition is played by a single piano that revolves around a series of arpeggiations. With this arpeggiation, the main melody was doubled and had the left handed piano part become a separate organ instrument and made the right handed piano part become a synth keyboard. A phaser effect was added to the organ instrument in order to add a layer of depth and mystery that was not present in the original arrangement. This song was placed after the fifth puddle.

Another track that went into the game was Prelude 20 from *The Well-Tempered Clavier, Book I*. This song was chosen due to the original tempo and key. Originally, the song had a slower tempo with more in-between notes. The new arrangement has an faster tempo, but the added effects make the song sound slower than intended. The song compares to the original arrangement in the sense that the original arpeggiation is still present, but the instrument used effects such as a heavy grain delay and a ping pong delay effect to make the listener feel as if they are underwater. This song was placed after the completion of the fourth puddle.

After many failed attempts of picking songs that were familiar to him, Marco both expanded his knowledge of the works of Bach and also learned to give a new sound to music

that was created three centuries in the past. Following this section, the sound design challenges and solutions that were faced will be explained.

2.3.3. Sound design

Within the game, there are puzzles that are spread across the park and each completed puzzle unlocks the next one. As a way to reward the player for their efforts, each song plays over the current ambient noise in the park. Originally, the vision for the placement of Bach songs was going to occur at certain places in the park. After discussing the overall player experience with the team, it was decided that the Bach pieces would act as both a reward for the completion of each puzzle and as a symbol of progression. Another factor that justified this decision was that the songs would entice the player to explore each new section of the park. Since the player would spend most of the time in the game walking around the park, the sound of the footsteps had to be implemented with attention to detail (Figure 28). For each ground material in the park, there is a different footstep that is played to match with the corresponding ground material. There are five types of footstep materials: wood, dirt, gravel, stone, and grass. Each time the player takes a step, the game will play the left foot sound followed by the right foot sound on the next step. The reasoning behind creating a different sound for each foot is to ensure that the walk cycle does not sound repetitive. Both the left and right footsteps are panned slightly towards

Figure 238. Sound recording setup for wood footsteps. Photo by Marco Duran.

their respective direction.

In addition to recording Foley (Figure 29), Marco had also led the recording of voice acting for the narrative script. Marco had recording done for the voices of William Morton, Gilbert Abbott and John Collins Warren as described in the narrative design section. John Thornberg played the role of Doctor Warren, with August and Connor supporting him as Gilbert Abbott and Doctor Morton, respectively. While recording the voices, Marco had helped in directing the style and tempo for each voice. A major problem that occurred during the recording process was the repeated opening and closing of a nearby lab door. The microphone faithfully picked up the sound of the door, regardless of how close the actor was to the microphone, requiring many extra takes.

Figure 249. Additional supplies for Foley recording. Photo by Marco Duran.

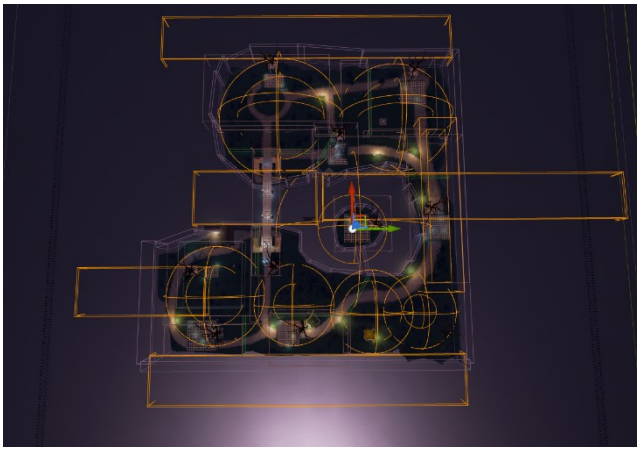


Figure 30. Top-down view of *Velum* with sound triggers. Photo by Marco Duran.

While looking at the park from a top down perspective (Figure 30), there are places in the Garden that are either surrounded by trees or overlooking the water. From this observation, Marco created two different ambient tracks: one for the interior of the Garden and one for the edges, near the surrounding water. Sounds that can be heard in the first track include various crow, squirrel, and insect samples as well as an ambient synthesizer. In the second track, there are various waterfowls, squirrels, slow-moving water and children playing.

The challenges that Marco faced were the diverse number of audio recording methods, limited artist choice and incorporating the audio in Unreal Engine. Creating an asset list and keeping to a strict schedule proved to be successful, considering the time constraint. An understanding of using multiple audio workstations had also proven to reduce the amount of time needed to create new arrangements, sound effects, and voice acting. These design challenges were crucial considerations when it came to anticipating the level of scope needed to complete his part of the project in one term.

2.4. Engineering

Technical development was the responsibility of the CS team members including August Beers and Nick Chaput. Over the course of development, two major pieces of software were created. The first project, *Eukaryote,* is a two dimensional version of our puzzle progression developed with *Perlenspiel.* The second project, *Velum,* is developed to run on a desktop PC using Unreal Game Engine. Identifying the correct game engines to use for our project was a priority in early development, and was pivotal to the project's success for several reason. At the end of development, *Eukaryote* amounted to roughly 3000 lines of code including JavaScript, HTML, CSS, and PHP. *Velum* amounted to roughly 1300 lines of Unreal C++ script supporting over twenty separate blueprint graphs. The process of implementing these projects and some details of their implementation are described below.

2.4.1. Choosing an engine

One of the team's major challenges was producing the required technical features for gameplay to occur, first within the Perlenspiel prototype, and later to produce play within *Velum.* To achieve the technical scope of the project, the tech team considered a few viable game engines currently available to independent teams. During early development, the team settled on employing two game engines. First and foremost, Unreal Engine 4 was used to produce our final, polished three-dimensional game. Second, the team employed *Perlenspiel* for use both as a rapid prototyping tool, and to fill the requirement of pushing a playtestable version of our game to the web. Along the way the team made notable technical achievements, including algorithms which drive the functionality of pieces within the puzzle, a level building tool which creates serialized save states of levels, and simple AI to employ animated squirrels in

Unreal.

In the early stages of *Velum*'s design and scoping, it was quickly established that our team would need to make use of a free to use third party game engine to support the tech required for the three-dimensional game we envisioned. In order to create our own game engine, the group would have needed a tech team more experienced in graphics programming and possibly an additional year. Once it was clear that a third party engine would be needed, the team evaluated a few free tools for consideration. These tools included Unreal Game Engine 4, Unity and GameMaker. Tool consideration did not last long, and Unreal Engine was chosen as the obvious forerunner for the team to use since it was the only large scale three-dimensional engine that the entire team had prior experience in.

Once the project was underway, the team employed Perlenspiel as a second game engine to achieve rapid prototyping and web dispersal of the game's core mechanics. By using Perlenspiel as a prototype tool, the team was able to create a testable alpha version of *Velum* early in the project lifecycle. Perlenspiel was a valuable design tool because its deliberately limited design space was much more approachable than the

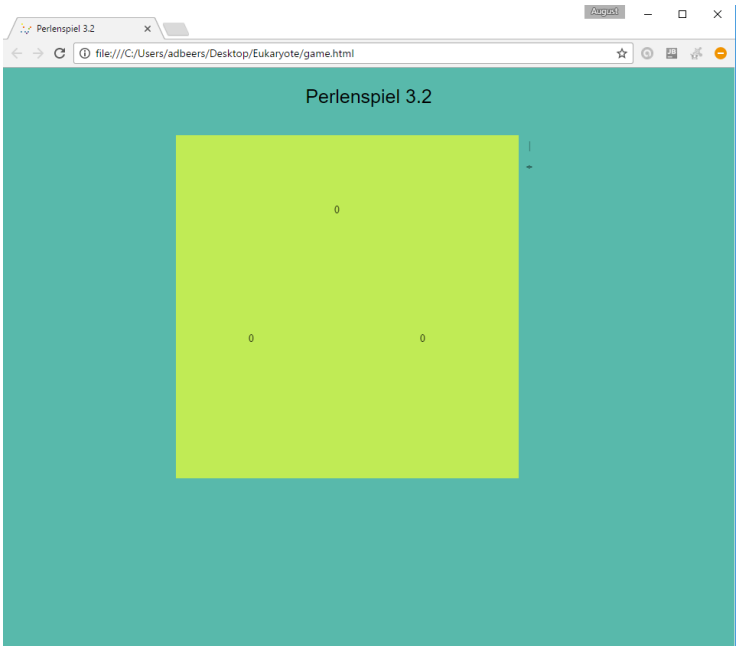


Figure 251. Early Perlenspiel prototype. Image by August Beers.

endless number of possibilities presented by Unreal. The small design space of a two-dimensional grid was also a realm in which the design team had considerable prior experience, as many of us had already created multiple games in Perlenspiel, as well as grid based board games at WPI.

The rapid prototyping provided by Perlenspiel allowed the team to nail down the design of the core puzzle mechanics early in the design process. During the first quarter of our work at WPI, we were able to turn a very loosely defined concept into a fully playable game in a matter of weeks. A screenshot of one of the earliest versions of the game is shown in Figure 31. From our 2-dimensional prototype it was easy to jump into the three-dimensional world of *Velum,* and we were able to implement a three-dimensional prototype of the puzzle before the end of the first quarter.

During our second quarter of continued development, the Perlenspiel game was given a level progression consisting of nine unique puzzles. The level editor described earlier during the puzzle design section facilitated this process. The level editor was implemented as a separate module of *Eukaryote* which relied on a shared internal game state module. This design allowed the editor to have all of the gameplay components of the actual game. The team could design, test, and then serialize their level designs into JSON all within one interface.

The playtesting potential of Perlenspiel was brought to light when the two-dimensional game, which was fully HTML5 compatible, was distributed to hundreds of playtesters across the web. This opportunity was made possible using Amazon's Mechanical Turk service. Without use of Perlenspiel, the team would not have been able to gain the confidence we have in the attributes of our puzzle design. More information about the online version of the game is

available in the study section of this paper.

During mid development, it became the team's motivation to provide the same smooth gameplay experience available within Perlenspiel redefined in Unreal's 3-dimensional environment. This design proved to be complicated, and the team eventually settled on preserving the Perlenspiel game's 2-dimensional puzzle design by having players of Velum play the puzzles on the terrain of the Unreal level. This model employs seeds as puzzle pieces, and a visible grid placed on the lawn of the Garden.

2.4.2. Coding

All of the team's software projects were developed using source control to manage the codebases. This section describes some of the technical details of the implementation of the team's two major pieces of software, *Eukaryote* and *Velum.*

Software development of *Eukaryote* started during the summer of 2016, and utilized GIT as version control throughout the project. Written mostly in JavaScript at first, the codebase consists of object-oriented code with some functional programming techniques mixed in as well. The core of the game's implementation lies in a file state.js which defines a 16x16 grid and a timer function which updates a simulation running on the grid. The grid defined in state.js is an internal representation of a Perlenspiel grid. The actual Perlenspiel grid used to play the game is treated as a visualization of *Eukaryote's* internal structure, a design driven by the Model View Controller Paradigm (Krasner, 1988).

It is upon the internal grid state of *Eukaryote* where the gameplay is defined. A puzzle piece class may be moved in, out, and around the grid in search of a win condition. Within these puzzle pieces a strategy pattern was implemented to organize the collection of search algorithms which produce different growth formations on the grid. In this design, each algorithm is encapsulated into its own object, each providing a consistent interface to its caller. We chose this design pattern because it provides an easy way to configure pieces with many behaviors while easily making those behaviors exchangeable (Gamma, 1994). Figure 32 shows an example of a growth algorithm encapsulated into an object.

```
//Breadth first search pattern grows out from node.
BFS_growth: function(myWorker){

    this.icon = E.BFS_ICON;

    //Growth algorithm
    if(E.Growing && myWorker.pos.inGrid()){
        if(myWorker.radius < 3){
            myWorker.slime.clear();
            myWorker.slime.addArray(this.growthRing(myWorker.radius, myWorker.pos));
            myWorker.radius++;
        }
    }

    //Return growth from the center of the hive.
    this.growthRing = function(radius, pos){
        var setSize = 1;
        for(var i = 0; i <= radius; i++){
            setSize +=  i * 4;
        }

        w_g = E.gameWorld.worldGraph;
        w_g.clearFlags();
        var set = E.BFS(w_g,
                    pos,
                    new E.Vector(E.GRIDSIZE + 1, 0),
                    setSize);

        return set;
    }
},
```

Figure 262. Growth algorithm object listing. Image by August Beers.

This growth algorithm utilizes a breadth first search to create a fun puzzle piece which is able to grow around and be squeezed into different shapes by the walls on the board.

By the time that the mid period of project development arrived, the tech team changed their development priority from designing and implementing new mechanics to modifying the game so that it can run within the Amazon Mechanical Turk web environment. We acquired a virtual server to run the game from, hosted by WPI. To make Eukaryote an analytical tool we

added functionality to automatically record gameplay statistics and to capture webcam footage of our players faces. PHP and JavaScript were implemented to record data on the client's machine, package it, and send it across the network back to our server at WPI. Development on this functionality was extensive and is described in greater detail in the study design section below.

Software development of *Velum* began at the beginning of the 2016-17 academic year. At first, the team used a free online GIT repository to host the project, but the Unreal project quickly grew too big. *Velum's* assets and codebase was in need of a new long term home, so the team contacted WPI to acquire a virtual server. On this server, the team utilized Perforce as a version control system because it is deeply integrated into the Unreal editor.

Development occurred side by side with *Eukaryote,* yet the 3-dimensional game was not the first priority of the tech team during early development. This decision was made because creating content within Unreal proved to be much more time consuming compared to the lightweight implementation of Perlenspiel. For this reason, major development on three-dimensional mechanics was secondary until the team developed a Perlenspiel prototype which they were confident would provide a quality gameplay experience.

At the beginning of mid development, Eukaryote had been fleshed out into an enjoyable gameplay experience, and the tech team felt confident enough to begin its three-dimensional implementation with Unreal. From the tech team, August took the lead on the 3-dimensional implementation, as Nick did not have previous experience in Unreal and had plenty of work to do in pushing *Eukaryote* into the Amazon Mechanical Turk environment.

Unreal's C++ Script is strongly optimized for object-oriented design. This meant that *Velum's* code could not be as freeform as its JavaScript counterpart, *Eukaryote*. C++ classes were created for the essential components of *Eukaryote's* puzzles. A board class composed of node classes recreated the internal grid. From there, a puzzle piece class was created which once again implements the strategy pattern to handle the family of growth patterns.

```
oid ABoardNode::plantSeed(ASeed* seed) {

  if (!seed->planted) {

      mySeed = seed;
      seed->planted = true;
      UStaticMeshComponent* rootComp = (UStaticMeshComponent*)seed->GetRootComponent();

      Mesh->SetCollisionEnabled(ECollisionEnabled::QueryOnly);

      FStringOutputDevice yo = FStringOutputDevice::FStringOutputDevice();
      CallFunctionByNameWithArguments(TEXT("Plant"), yo, NULL, true);

      //rotate to north
      if (northFlipFlop) {
          FRotator rot;
          Mesh->SetRelativeRotation(FRotator(0, 90, 0));

          if (seed->type == TEXT("Horiz")) {
              Mesh->SetRelativeRotation(FRotator(0, 180, 0));
          }
      }
      else if (seed->type == TEXT("Horiz")) {
          Mesh->SetRelativeRotation(FRotator(0, 90, 0));
      }
```

Figure 273. The plantSeed() function.

Perhaps the most impressive aspect of *Velum's* codebase is the relationship between the C++ script and the blueprint scripts which work in tandem to define the necessary parts for our game. As August grew more accustomed to Unreal development, it became obvious that some problems where better solved by C++ script and others by blueprint. To adapt to this issue, August became very proficient at weaving the execution blueprint and C++ logic together. This design was motivated by the realization that blueprint was more effective for manipulating art assets, while C++ script was more effective at creating game logic and algorithms.

Together, the two code examples in Figure 33: Function, plantSeed() and Figure 34: Blueprint Function Plant showcase this kind of design. The function plantSeed() contains logic to take a puzzle piece and insert it into a game board. On the 7th line of the function, a call to "CallFunctionByNameWithArguments" inserts the execution of code into the blueprint depicted below. The blueprint handles the orientation and visual appearance of the puzzle piece's static mesh which represents it visually in-game. This design allowed the art team to change and

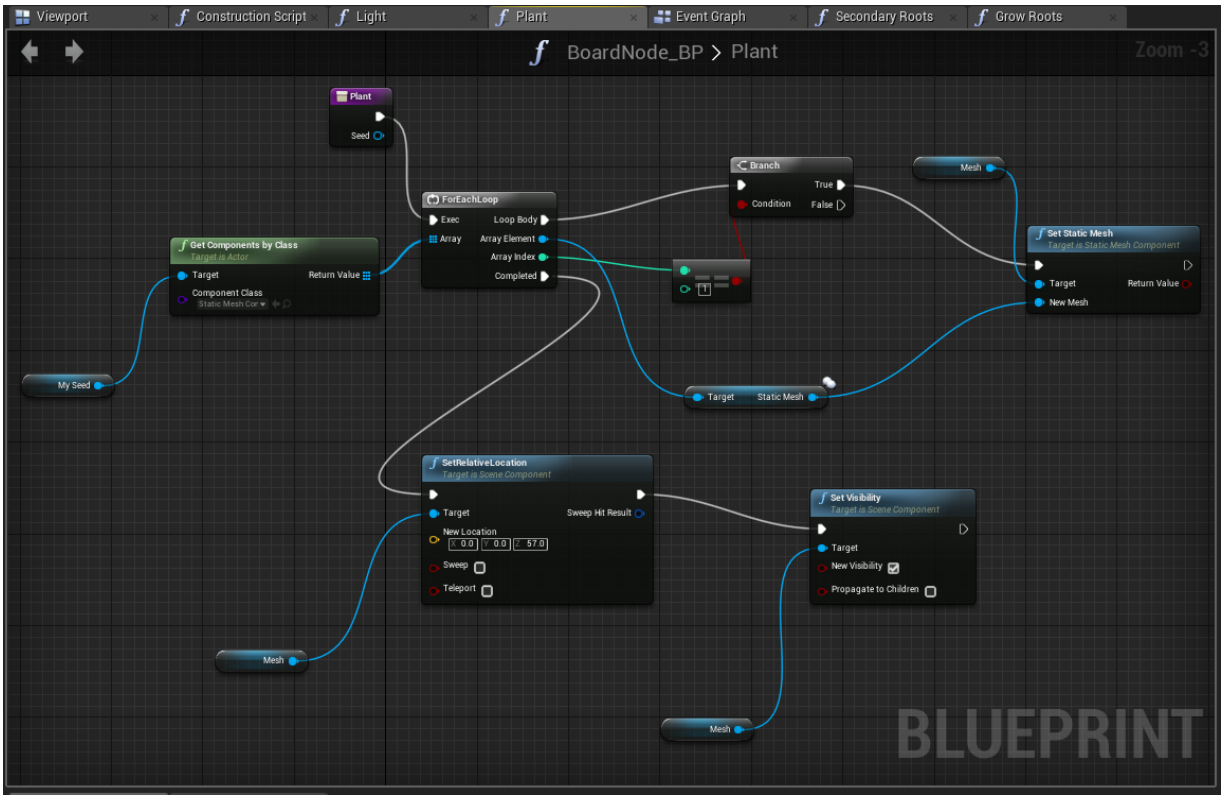


Figure 284: Blueprint Function Plant.

modify these assets on their own without needing tech team assistance.

When the blueprint finishes execution, it returns to C++ and more work is done by C++ to position the now visible asset. This kind of blueprint to C++ transition was utilized many times within *Velum's* codebase.

3. Playtesting

From the beginning, the team made playtesting and iteration a priority of the project. We knew that a puzzle game would require a large amount of playtesting to ensure consistently entertaining and quality puzzle designs. In early development, all of the team members individually designed puzzles in Perlenspiel. We relied on feedback from other team members for the first iteration of puzzle designs. This method of playtesting allowed team members to collaborate in creating a series of puzzles that feel cohesive and satisfying in their progression.

Going forward into the mid development period of *Velum*, it was time to begin playtesting the game outside of our team. We understood that because we thought of the initial ideas of each puzzle as well as the solution, we weren't approaching the game in the way that an ordinary player would. To combat this common issue of familiarity that occurs in the design process, we looked towards sources of playtesting where we could access larger groups of people who had never heard of our project before. Our very first test sessions were done with close friends of the team in a casual setting. These were mainly preliminary tests to see how ready the game was for large scale testing. After a couple of these casual tests, we moved towards more formal and organized methods of playtesting.

3.1. Professor Moriarty's class

During mid development, we had an opportunity to playtest *Velum* in Professor Moriarty's Digital Game Design class. This was the first opportunity to test our puzzles working inside of Unreal Engine. Before this playtest, the only people who had played the game had helped make it. As the developers, we were testing the game with the understanding of what

to do and how things worked; these new testers were not. We anticipated that playtesting the game with individuals new to the project would reveal overlooked design flaws. In order to asses playtesters opinions of the game we developed a survey that we thought would give us valuable information.

At the end of the playtest session, we opened open the floor to a formal critique of our game. We received a lot of feedback verbally from our playtesters and asked them some questions about their gameplay experience.

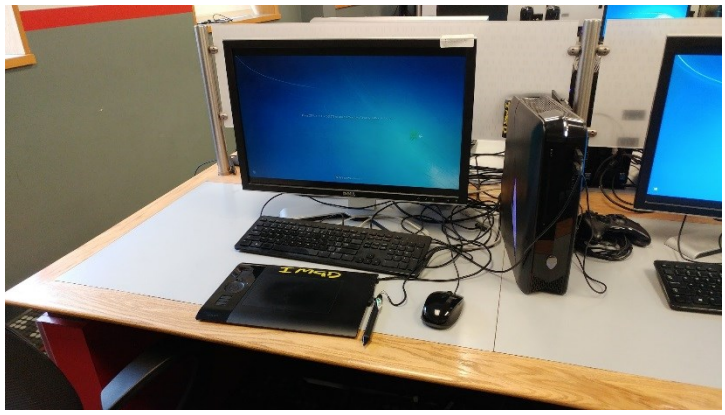


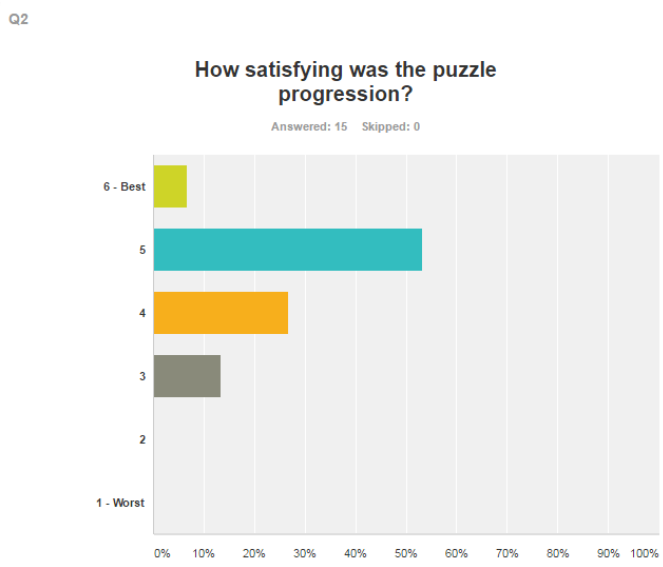
Figure 295. Lab computers in FL222.
Photo by Connor Mattson.

Conveniently, the machines (Figure 35) that were available to run the playtest with were identical to the machines that we had developed and built the game on. This allowed us to not worry about compatibility of hardware or software issues.

3.1.1. Playtest results

For the first round of playtesting we varied the format of the survey questions so that some of them were asking the player to rate a specified aspect of the game from 1 to 6, while other questions were open-ended and warranted written responses. Having different formats of questions allowed us to get a meaningful data size for more general curiosities the team had, while still allowing the player to voice concerns about a specific aspect of the game or point out bugs.

During the playtest we took notes, which are available in Appendix B. We learned that there was a game breaking bug which would cause players to be unable to complete all nine puzzles. We observed one other game breaking bug where a player was trapped out of bounds. We also learned that players were struggling to understand what their goal should be during the first few minutes of play. These bugs did not limit the efficiency of the playtest process because of a contingency plan the tech team developed beforehand; knowing that game breaking bugs could be an issue, August added in a developer hotkey which could progress the player forward to the puzzle they were on.



Answer Choices	Responses	
6 - Best	6.67%	1
5	53.33%	8
4	26.67%	4
3	13.33%	2
2	0.00%	0
1 - Worst	0.00%	0
Total		15

Figure 36. Graph of puzzle satisfaction. Image by Connor Mattson.

From the data collected by the survey, found in Appendix C, we learned our playtesters found our puzzles to be satisfying for the most part (Figure 36). Another important question we asked was if users found the squirrel to be an effective guide. Users were split almost fifty-fifty as to whether the squirrel was effective or not, which supported our observation that the beginning of *Velum's* play was not as smooth as it could be.

From a series of written responses (Figure 37), we learned that some players found the Bach soundtracks to not fit the mood of the park. This was important feedback for our sound

designer Marco, who later edited the tracks to be a little softer. We also decided, based on this feedback, not to have Bach tracks playing while users are attempting to solve puzzles.

For the formal critique by members of the class, we learned about several more interesting nuances of our design. Some users found it difficult to see the structure of the board because of the ground texture beneath the grid. Based on this feedback, we re-textured underneath all of our puzzles to represent the park's lawn. Another player was frustrated by the wait time needed to test a given solution. To address wait time, we shortened the timer between growth ticks from one half to one third of a second. Finally, we received an opinion from one playtester that the ending of the game left a little to be desired. We anticipated this critique as we had not yet implemented our narrative images and audio. We hope that through the narrative of the game, players may find more closure in the game's ending.

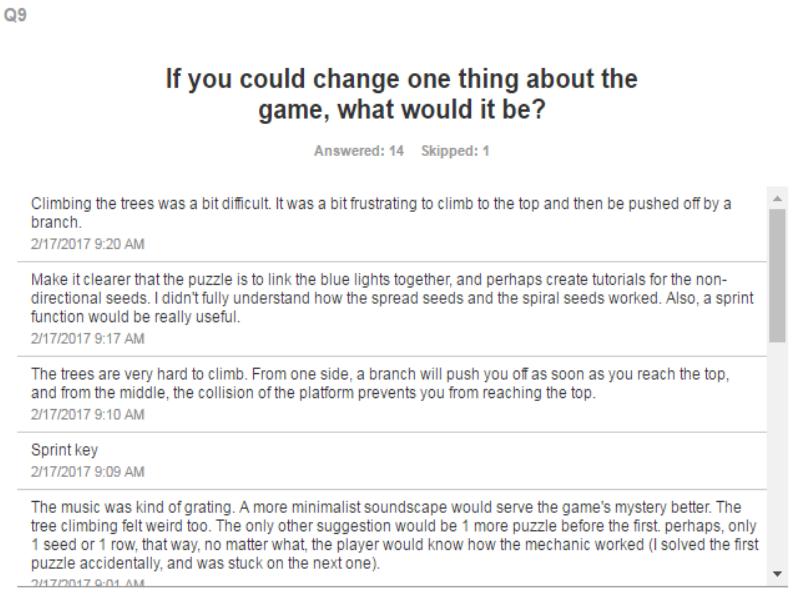


Figure 307. Series of written responses. Image by Connor Mattson.

3.2. PAX East 2017

PAX East was a valuable experience for the team as well as for the game. This was the first time the game was available to the public. We initially were not sure whether to attend PAX because the time spent preparing and going to PAX could be spent putting more work

Figure 318. Tester playing Velum at PAX East.
Photo by Connor Thornberg.

hours into the game itself. In the end, we decided that it was best for the game to get some publicity as well as give the developers who attended PAX networking experience. The three artists as well as Marco attended PAX to represent Velum at the WPI Booth.

As shown in Figure 38, our team members at PAX had the privilege of being able to watch players live as they progressed through the game. This was important because in our prior playtest within Professor Moriarty's class, there were too many people playing at once to follow how a single player progressed through the puzzles. This environment allowed us to speak with the player and answer questions they had while playing the game as opposed to simply gathering data from them afterwards.

3.3. Study design

We wanted to be confident that the puzzles in our game truly satisfied our experience goals. To this end, we worked with our advisor, Professor Whitehill, to design a study centered on playtesting *Velum*. We hoped to establish a process by which we could systematically optimize Velum to meet our emotional experience goals, an objective which traditionally relies on qualitative feedback from playtesters. Building off of Professor Whitehill's expertise, we were able to create and run a study in which Mechanical Turk workers played *Eukaryote,* the two-dimensional prototype. We captured all game events as well as images of their faces from their webcam to be used for facial analysis. We also developed a replay system that would take this data and play the game session back in real time for human observation. We used this information to steadily refine the puzzles of *Velum* throughout the development process.

3.3.1. Inspiration

From the beginning of the project, we had hoped for the CS component to be centered around improving an aspect of our game, rather than turning a noncritical component of our game into a technical accomplishment. Our original ideas were focused on developing an AI which would control people and animals in the park, or control the Fear Dearg which was central to our earliest vision of the game. In the end, we were unable to find an adviser with AI expertise who was motivated enough to work with us on this type of project. It was shortly after this that we learned of Professor Whitehill and his background with facial analysis.

As we discussed the possibilities that arose from facial analysis, we realized that we had the opportunity to design and test a game with an emotional experience goal and quantitatively measure that emotion in our players. We then reached out to Professor Whitehill and discussed

our ideas further with him. Among the possibilities we identified, one idea was to provide players with a simplified version of *Velum*, containing only its puzzles. With feedback from this prototype, we could iteratively adjust the levels of the game to meet our experience goals. We committed to this study idea once we decided that the puzzles in *Velum* would be self-contained, as this made it easy to design a version of the game that only contained the puzzles.

We began by researching how facial analysis software works. As we expected to be working with software that had roots in Professor Whitehill's work on the Computer Expression Recognition Toolbox (Littlewort et al, 2011), we looked there to learn more. The process begins by actually detecting the face. The method used is based on the Viola-Jones algorithm, which detects patterns of light in the pixel values of an image in order to identify the objects being pictured (VIola and Jones, 2004). It also employs boosting algorithms such as WaldBoost to ensure a reasonable compromise between time taken and accuracy of face detection (Sochman and Matas, 2005). The program then detects facial features using methods that are unique to each feature before extracting a cropped image of the face to use in calculating the intensity of Action Units: building blocks based on facial features that allow the program to determine the likelihood of the presence of given emotions (Littlewort et al, 2011).

We also researched previous studies that related to improving our game based on the data we collected. As we were not able to find studies specifically about using facial analysis to improve a game's design, we broke things down into two components. The first component was inspired by studies which used facial analysis to evaluate the effectiveness of teaching. These studies are relevant to our game because all games must be able to teach their players how to play. The second component was inspired by looking at how various modifications to

the difficulty of a game can affect a player's enjoyment and engagement.

To learn about facial analysis as a tool in learning, we once again turned to Professor Whitehill's work. In their paper *The Faces of Engagement: Automatic recognition of student engagement from facial expressions,* Whitehill et al describe how human observers rely heavily on watching a person's face to make judgements on their engagement to a task (Whitehill et al, 2014). They also identify a strong correlation between measures of engagement using still images and those using videos. This result indicates that the relevant information is largely contained in static features and visible in still images. Finally, they show that a program using machine learning can be trained to exhibit comparable accuracy to human observers in detecting engagement. This research served as evidence to us that we could use facial analysis to gain insights into players' engagement and ability to learn our puzzles.

It was more difficult to identify a study about how to modify a game's progression that truly seemed to fit our goals. During early development, we were primarily focused on player engagement and even considered real-time modification of the game's difficulty. Our attention was finally grasped by a paper that analyzed how players responded to both covert and overt manipulations of difficulty. We learned that players do not like to know that the game has been made easier for them, suggesting that player engagement may be related to their perception of competence (Khajah et al, 2016). As we read more, we began to feel that facial analysis and real-time modification to our puzzles may not have been feasible in the same project. We chose to prioritize facial analysis, and shift toward manually adjusting our puzzles according to the data we collected.

3.4. Experimental design

The study had two main components. The first was *Eukaryote*, our simplified version of the game containing only the puzzles that would eventually be put into *Velum.* When Mechanical Turk was introduced to the team as a way to increase our pool of playtesters, it soon became apparent that Perlenspiel was an ideal game engine for the platform because it runs in the browser. As a result, *Eukaryote* became even more ingrained in *Velum's* development as it was integrated into the second component of the study (Figure 39).

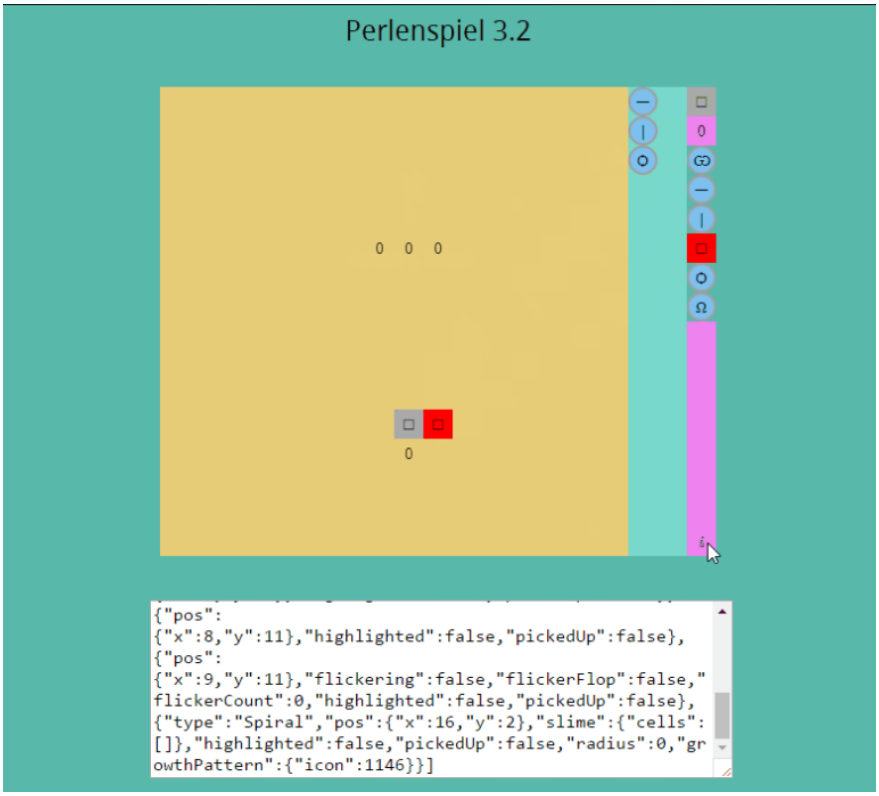


Figure 329. The *Eukaryote* level-building tool. Image by Nick Chaput.

The second component of the study was the Human Intelligence Task (HIT) which we put out through the Amazon Mechanical Turk service. This HIT needed to have workers play our game while recording data from both game events and the workers' webcams. This was primarily set up by Nick, who also created a modified build of *Eukaryote* to watch game replays in real time, and set up all of our *Eukaryote* builds to run on our web server. The design of our experiment went through frequent iteration based on feedback from Professor Whitehill and Professor Moriarty.

This ensured that we collected all the data needed for analysis without affecting the worker's experience playing the game or compromising facial data.

3.5. Methodology

To create the components of our study, we playtested each mechanic within the team. We then ranked each puzzle on difficulty and enjoyability before having a group discussion to decide which puzzles might be worth including in the final game.

Nick set up the HIT using Amazon Web Service's Command Line Interface, working off the external_hit sample. He used html5's getUserMedia() function to access the user's webcam in order to take a snapshot twice every second. He found that he was able to run this functionality within *Eukaryote* as long as he kept the required HTML elements in the page's header. Perlenspiel's engine will not load if they are placed in the body instead.

Nick then set up the team's web server, initially provided by Michael Voorhis. He worked with Voorhis to acquire and set up the certificates needed to enable an HTTPS connection to the server, which was required by the Amazon Mechanical Turk platform. He also wrote PHP scripts to log game data and save the webcam snapshots from players, then added the necessary AJAX to *Eukaryote* to actually send that data to the server. Figure 40 below shows how the logs are generated within the Perlenspiel game. Figures 41 and Figure 42 show how the previously generated logs are prepared and then sent to the server via AJAX. The previously mentioned PHP scripts can be found in Appendix D.

```
40          while (!isImgLogged) {
41              var item = G.logQueue.shift();
42              var components = item.split(" ");
43              if (components[0] == "00") {
44                  isImgLogged = true;
45              }
46              logArray.push(item);
47          }
```

Figure 40. Example of a game event being logged. Image by Nick Chaput.

```
287     var d = new Date();
288     var code = "06 " + ("00" + x).slice(-2) + " " + ("00" + y).slice(-2);
289     var logItem = code + " " + d.getTime();
290
291     console.log("bead exited at " + x + ", " + y + " " + id);
292
293     G.logQueue.push(logItem);
```

Figure 41. Code example demonstrating how a block of event logs are sectioned out for transmission to the sever. Screen capture by Nick Chaput.

```
51          $.ajax({
52              url: "imageCapture/LogEventSequence.php",
53              method: 'POST',
54              data: {
55                  img: canvas.toDataURL(),
56                  logArr: logArray,
57                  id: id},
58              success: function(result){
59                  console.log(result);
60              },
61              error: function(result) {
62                  console.log(result);
63              }
64          });
65      }
66  }
```

Figure 42. The AJAX call that sends even logs to the server. Screen capture by Nick Chaput.

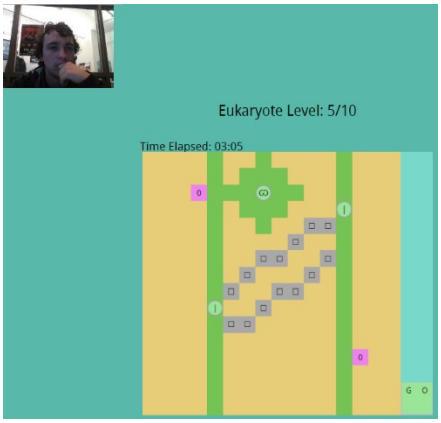


Figure 333. The replay build. Image by Nick Chaput.

Nick created the replay build of *Eukaryote* by modifying a copy of the JavaScript file that handles user interaction within Perlenspiel. As shown above in Figure 43, this build allows us to watch a player's face synchronized to their gameplay events. Nick accomplished this feature by stopping the game from responding to user input, and then reading through the log file of game events. These events were then used to create timeout events that would simulate the player performing each and every input from their play session. He managed this behavior by moving code functionality out of Perlenspiel's event handlers and into helper functions which would be called freely. See Figure 44 for an example of code bound to an event handler in the prototype, and Figure 45 for an example of that same functionality modified to work in the replay build.

```
PS.exit = function( x, y, data, options ) {
    "use strict";

    var exitPos = new E.Vector(x,y);
    if(G.dragged != null && !E.gameWorld.isCollision(exitPos)) {
        PS.glyph(x,y, 0);

        var d = new Date();
        var code = "06 " + ("00" + x).slice(-2) + " " + ("00" + y).slice(-2);
        var logItem = code + " " + d.getTime();

        console.log("bead exited at " + x + ", " + y + " " + id);

        G.logQueue.push(logItem);
    }
};
```

Figure 344. Code bound to Perlenspiel's exit event in Eukaryote. Image by Nick Chaput.

```
PS.exit = function( x, y, data, options ) {

};

function replayExit( x, y ) {
    "use strict";

    var exitPos = new E.Vector(x,y);
    if(G.dragged != null && !E.gameWorld.isCollision(exitPos)) {
        PS.glyph(x,y, 0);
};
```

Figure 45. The same functionality from Figure 44 above, refactored to work in the replay build. Image by Nick Chaput.

After each HIT, Nick compiled the image sequence from each player into a video using FFMPEG, and processed it using iMotions to analyze the players' faces and gather emotional data. As shown below in Figure 46, iMotions outputs graphs indicating the intensity of various emotions detected in the player throughout the video. An example of the raw data in TSV format can be found in Appendix E. The graphical form was extremely helpful is providing early feedback on what emotions players were experiencing while playing our game.

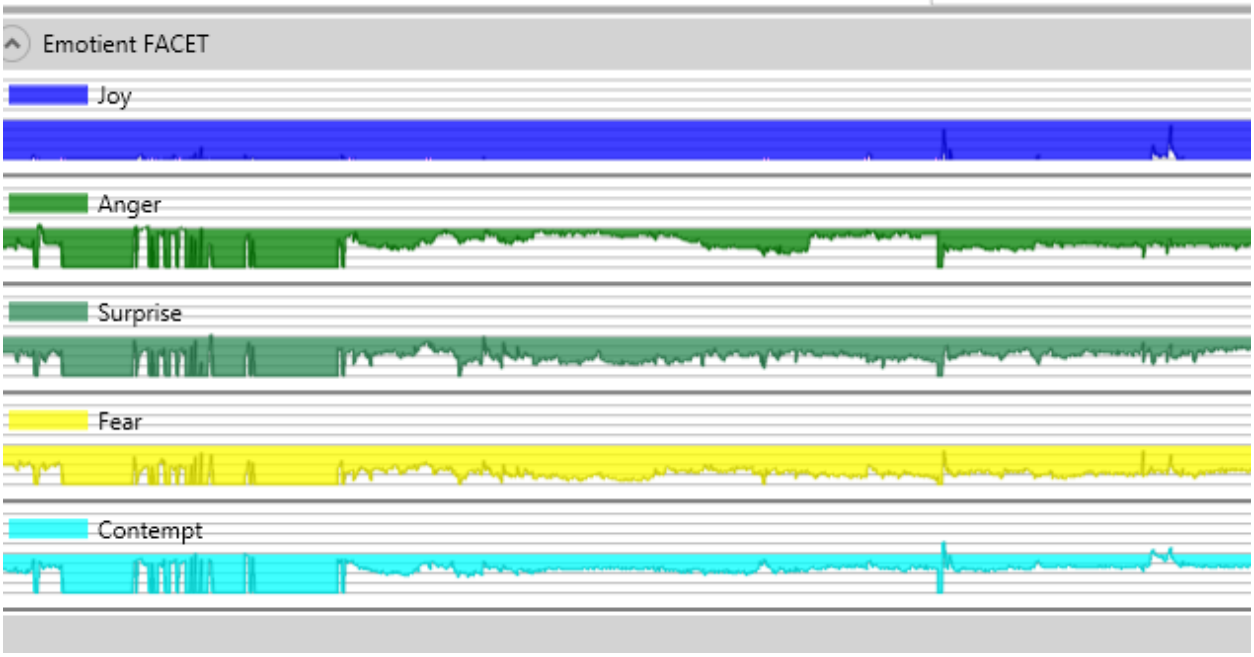


Figure 356. Emotional channel graphs produced in iMotions. Image by Nick Chaput.

We were able to run six separate HITs through Mechanical Turk. Four of these HITs were at a small scale, with 20 or fewer participants, ensuring that all aspects of our system worked as intended before attempting to run a full scale study. The remaining two HITs were at a much larger scale, with 120 participants in the first and 200 in the second. Both of these HITs were aimed at gathering data specific to a particular question.

The first HIT tested the question "Given our puzzle progression, would an extended tutorial sequence be beneficial to the player's understanding of the game, such that their performance would be better than players who did not receive it?" The goal of this HIT was to improve the learning curve of our puzzles, as we faced concerns over how difficult it was for new players to learn the game's rules.

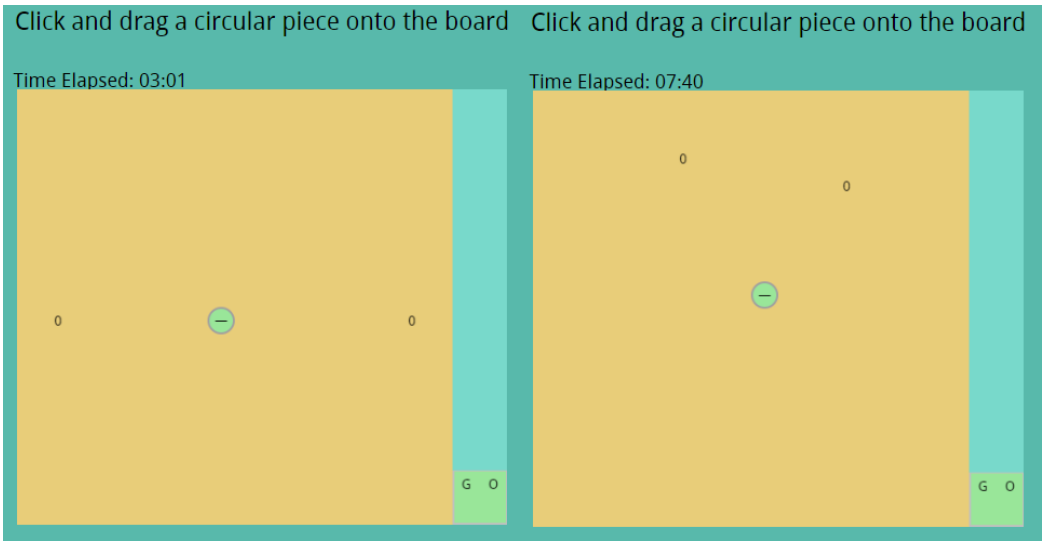


Figure 367. Additional tutorial levels tested in the first experiment. Image by Nick Chaput.

In this HIT, we randomly assigned players to either receive or not receive two simple tutorial levels, shown above in Figure 47. Regardless of whether they received it or not, they would then be given an 8-level puzzle progression. Detailed logs and facial images were recorded throughout the session. The first additional tutorial level was already solved, and the player simply needed to hit “GO” to watch the solution grow and complete the puzzle. The second tutorial level also contained a single piece that was already placed on the board, but was not already solved for the player. This required the player to shift the game piece upward before hitting “GO.” Our intuition as game designers told us these levels might help the player by teaching them some of the basic rules of the game as well as teaching them how to interact with the game board. We were interested in where this tutorial would allow them to focus more on the puzzles themselves later on.

The second HIT tested the question "Are we able to deliberately produce frustration in the player through the design of our puzzle mechanics?" The goal of this HIT was to see if it is feasible to treat emotional experience goals as measurable goals, using facial analysis.

In this experiment, we gave participants 5 puzzles. The first 4 puzzles use only the simple vertical and horizontal pieces. Participants are then randomly assigned to one of two final puzzles, both of which introduce a new growth pattern to the players. These two puzzles can be seen below in Figure X. The first of these puzzles introduces the players to a piece whose growth pattern was random. We did not tell the players this fact. The second puzzle introduces players to the breadth-first search piece, a piece whose growth pattern was entirely deterministic. We believed these piece were reasonably comparable based on our observations of how much difficulty players had with these pieces during in-person playtesting. We believed the random piece would evoke more frustration in the player as it would remove much of their control over solving the puzzle.

3.6. Hypotheses

For the most part, our hypotheses were formed with the goal of quantitatively testing questions related to game design that are normally left to the designers' intuitions. As such, each hypothesis has the form of "Based on our intuition as game designers that x is true, we expect to see some specific and measurable result in one form of the game compared to another."

In the first full scale HIT, we expected to see that a larger percentage of the Mechanical Turk workers who received the lengthened tutorial would be able to complete the game within the 15 minute limit. In order to test this, we tracked which progression players received and whether or not they finished the game. To support this hypothesis, we would need to see that the players who received the extra tutorial levels were significantly more likely to complete the game than those who did not.

Additionally, we expected to see that players who received the lengthened tutorial would typically make it farther into the game within the 15 minute time limit than players who did not receive the lengthened tutorial. In order to test this, we tracked which progression players received and how many puzzles they completed. In order to support this hypothesis, we would need to see that players who received the extra levels tended to complete more levels than players who did not.

In the second full scale HIT, we expected to see more frustration in players who received our random-growth puzzle piece on the final level than in players who received the breadth-first search piece. In order to test this, we recorded images of the players face from their webcam while they played. We then compiled those images into videos and ran them through facial analysis to acquire measurements of their frustration while playing. We also recorded detailed logs of gameplay events which we could link to the measurements in order to categorize the situations in which those measurements were taken. In order to support this hypothesis, we would need to find that players who received the random growth pattern exhibited more frustration during the final level than players who received the consistent mechanic.

In both full scale HITs, we expected to see a spike of joy in the player within 10 seconds of completing a level. In order to test this, we recorded images of the players face from their webcam while they played. We then compiled those images into videos and ran them through facial analysis to acquire measurements of their joy while playing. We once again recorded detailed logs of gameplay events which were link to the measurements in order to categorize the situations in which those measurements were taken according to what was happening in

the game. In order to support this hypothesis, we would need to find that players in general exhibited, on average, significantly more joy within 10 seconds of completing a level than they did, during all other periods of gameplay.

We also identified another hypothesis for both full scale HITS which we were unable to investigate. In both full scale HITs, we expected to see increasing frustration and confusion in players during long periods without completing a level. In order to test this, we recorded images of the players face from their webcam while they played. In order to support this hypothesis, we would need to find that players exhibited more confusion and frustration during the spans between completing a level than they did at the points in time when they completed a level. We would also need to show a correlation between the length of these spans of time and the intensity of frustration and confusion exhibited. Ideally, we would also be able to show that during these long spans of time, the players' frustration and confusion tended to follow an increasing trend from the beginning of the span to the end.

3.7. Visualization

After the studies were completed, we had two data sets with a total of over 300 data points between them. Before it was possible to make conclusions about what sort of patterns existed within the data the team had to create a few exploratory visualizations. Two Interactive tools were generated, one representing dimensions of the log data set and one representing dimensions of the emotion data set. After the first two exploratory visualization were used along with python scripts to analyze the data set, the team was able to focus in on specific calculations and visualizations which aimed to support or contradict our hypotheses about the data set.

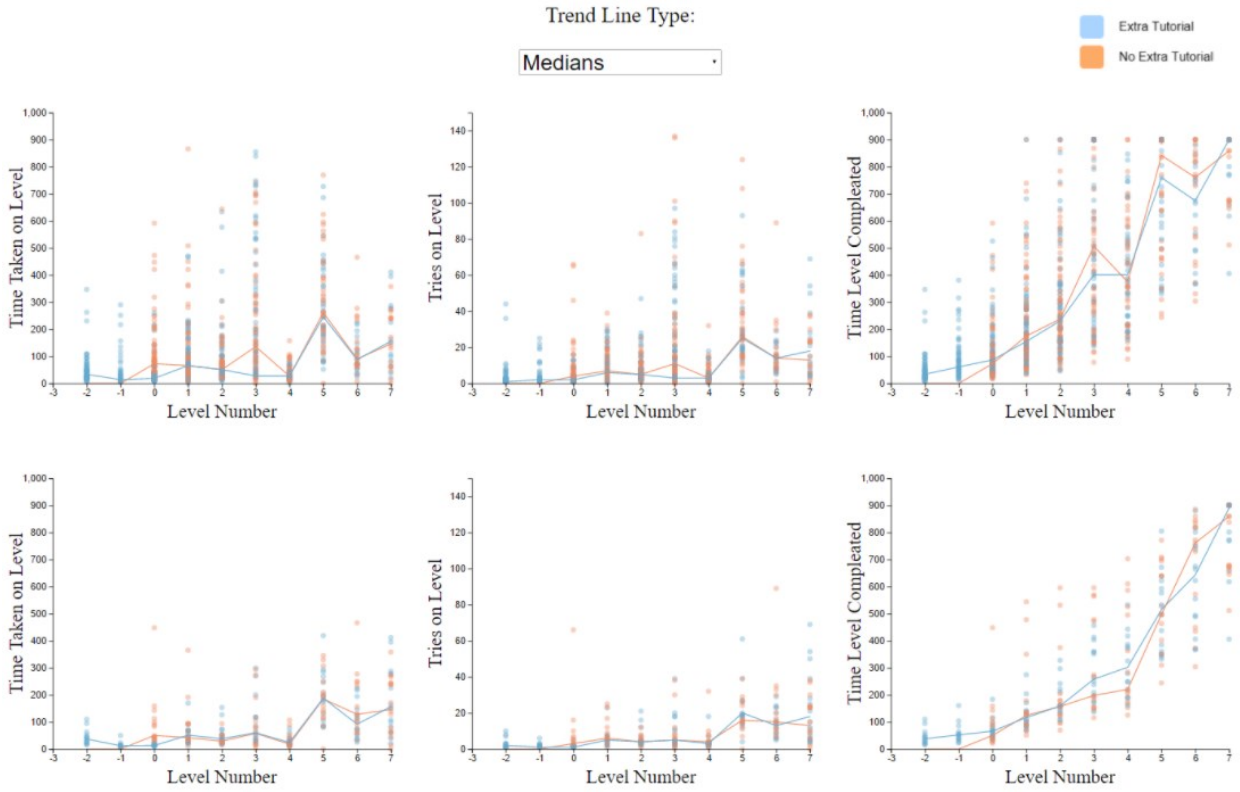


Figure 378. Log data visualizations from the first large-scale experiment. Image by Nick Chaput.

Figure 48 shows the exploratory visualizations created after the first large-scale experiment, using data from the log files. At its core this visualization is a scatter plot, we

plotted each of our directly recorded statistics against the level number, in order to look at the general trends of those statistics and compare them across the progressions. Points are categorized by color to indicate if they belong to a player given the extra tutorial or not. Alongside scattered points, average trend lines of the data is provided to summarize the denser regions on the scatterplot.

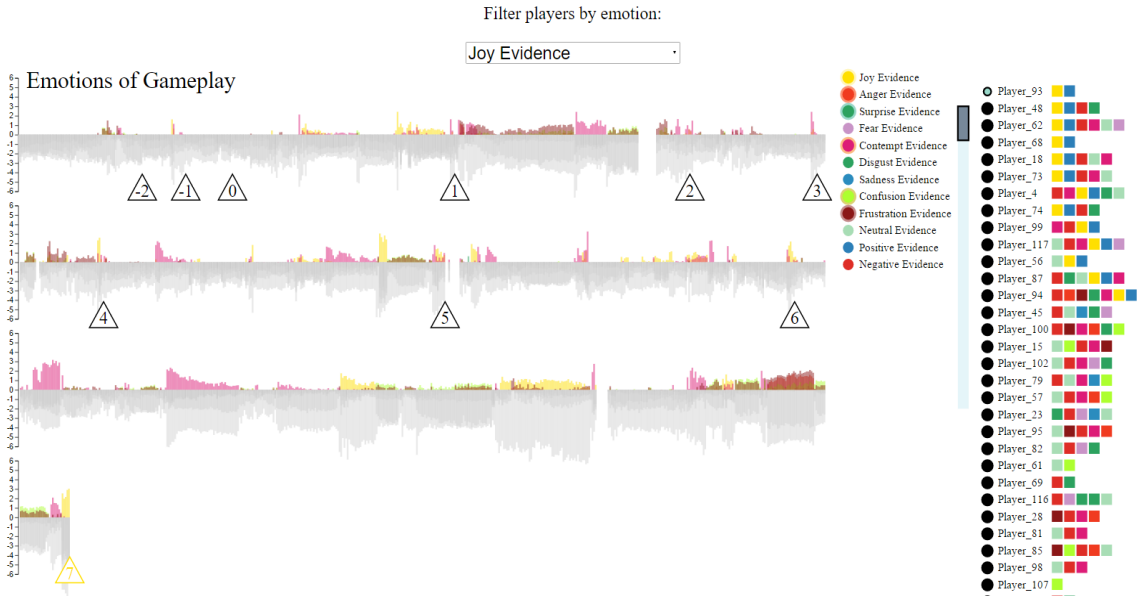


Figure 389. Emotional channel visualization from the first large-scale experiment. Image by Nick Chaput.

Figure 49 shows the second exploratory visualization we created after the first large-scale experiment, using the emotional data produced via facial analysis. On this visualization, each bar represents the intensity of a given emotion on a given frame of the player's video. There twelve emotions available which are each marked by a specific color defined in key just to the right of the graph. If a negative confidence was reported for an emotion on a given frame, a grey bar is assigned to extending below the line for that frame. To link the log events to the emotion data, the time at which the player completes a level is marked by a black triangle with the number of the level completed contained inside. If a player received the

extended tutorial, the first two levels are indicated as -1 and -2, so that level numbers represent the same levels as they were assigned to players in both progressions.

The team did not use this visualization to draw confident conclusions about the data, however it was valuable to assess the integrity of our data and to begin the exploration of our hypothesis. We were also able tentatively assess that most prevalent emotions in our dataset were confusion and frustration, followed by joy. Furthermore, we found that confusion and frustration often appeared together and in long durations, while joy occurred in much briefer periods of time. This became an early focus in our analysis as we attempted to determine if our hypothesis were correct across the dataset.

3.8. Log analysis

We analyzed the gameplay logs to gather gameplay data from each player. Specifically, we recorded how many levels each player completed, the time each level started, how long it took to complete each level, and the number of attempted solutions the player needed to complete each level. While it was infeasible to include the complete log data, we have provided a sample of formatted log data in Appendix F. We used this data to address the following hypotheses:

In the first full scale HIT, we expected to see that a larger percentage of the Mechanical Turk workers who received the lengthened tutorial would be able to complete the game within the 15 minute limit than players who did not receive the lengthened tutorial. Additionally, we expected to see that players who received the lengthened tutorial would typically make it farther into the game within the 15 minute time limit than players who did not receive the lengthened tutorial.

To test these hypotheses, we began by computing the percentage of players who completed each level given whether or not they received the extended tutorial. We searched through each participant's statistical log to determine which level progression they were assigned as well as the highest level they completed, and stored this information in arrays. We then iterated through those arrays, generating counts of how many players received each progression and how many players completed each level, which gave us the information needed to compute our final percentages. Out of 56 participants who received the extended tutorial, 14 of them, or 25%, finished the entire puzzle progression. Out of 62 participants who did not receive the extended tutorial, 15 of them, or 24.2%, finished the entire progression. This difference proved far smaller than we hoped, but it got worse for our hypothesis when we looked at each level individually. The complete results are shown below in Figures 50 and 51, but for 7 out of 8 shared levels, participants who did not receive the extended tutorial were as likely or more likely to complete a level.

```
Percentage of players who completed level 0: 1.0
Percentage of players who completed level 1: 1.0
Percentage of players who completed level 2: 1.0
Percentage of players who completed level 3: 1.0
Percentage of players who completed level 4: 0.9642857142857143
Percentage of players who completed level 5: 0.8928571428571429
Percentage of players who completed level 6: 0.625
Percentage of players who completed level 7: 0.5535714285714286
Percentage of players who completed level 8: 0.30357142857142855
Percentage of players who completed level 9: 0.25
```

Figure 50. Completion percentages for the extended tutorial puzzle sequence.

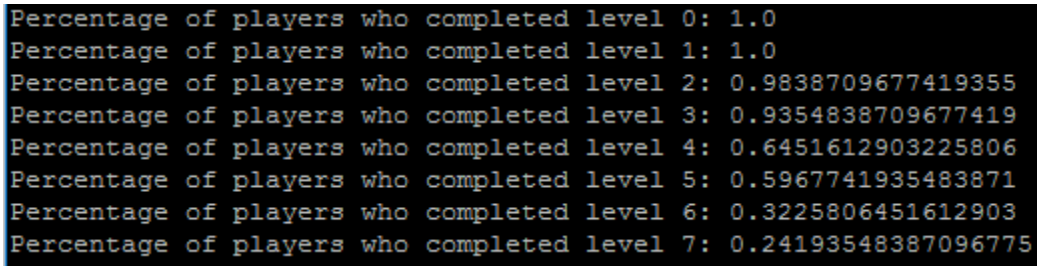


Figure 391. Completion percentages with short tutorial puzzle sequence.
Note that level 0 here corresponds to level 2 in Figure 50.

These results seemed to disprove our hypotheses, so we began to think about why this might have happened. The simplest and most negative explanation was that the extended tutorial simply offered no benefit at all to the player, or possibly even served to confuse them going forward. The most reasonable alternative to this explanation was that whatever benefits the levels may have offered were being offset by the amount of time it took for players to complete the extra levels. As a result, players may have simply lacked to time to make it as far into the game as players who did not receive the levels. To examine this possibility, we looked at the same data as before, but this time ran a survival analysis instead. While it would not be perfect, we hoped that conditioning on whether a player completed the previous level might help to alleviate discrepancies in time available, as we would only be comparing among players that were able to reach that level with the time they had. The results, shown below in Figure 52, were slightly better for the extended tutorial sequence than before, with better survival rates in 3 out of the 8 shared levels. However, this still shows that the advantage actually lied with players who did not receive the additional level, serving as strong evidence that our hypothesis was incorrect.

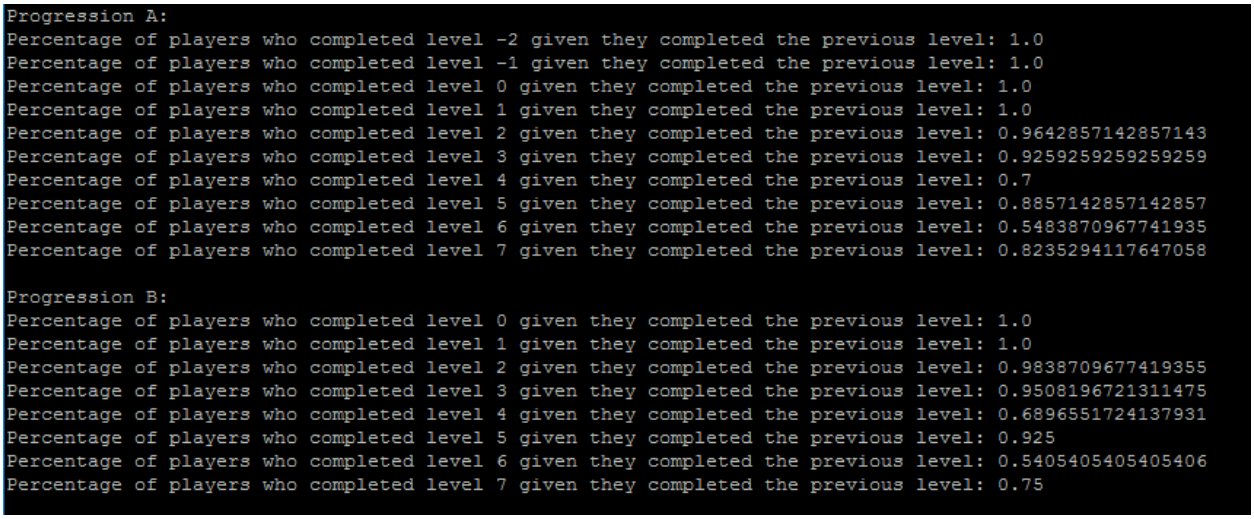


Figure 402. Survival analysis of the first large-scale study.

For Figure 52, progression “A” refers to the progression with additional tutorial levels, Progression B refers to the progression without them. Note that level 0 in Progression B corresponds to level 2 in Progression A

For each puzzle progression in the experiment, We were surprised to find that the inclusion of a drawn out tutorial did not lead to any clearly significant changes between the results for participants in these visualizations, but we did note that there were a few indications of difficulty spikes without the drawn out tutorial that did not seem to be there for participants who did receive the drawn out tutorial. We were also able to note a difficulty spike at the fifth level regardless of which tutorial the player received, and adjusted this level’s difficulty in later iterations of the game and study. This can all be seen in Figure 53, below.

3.9. Facial analysis

The facial analysis gave us a quantitative measure of players' emotions while playing, allowing us to measure and therefore test the different versions of *Eukaryote* against our emotional experience goals. These measurements, taken for 12 different emotions, were on a logarithmic scale with a value X from –4 to 4, representing that it is 10^X times more likely that the player was experiencing that emotion than it is that they were not. Notably, we used this to acquire the measurements needed to test the following hypotheses:

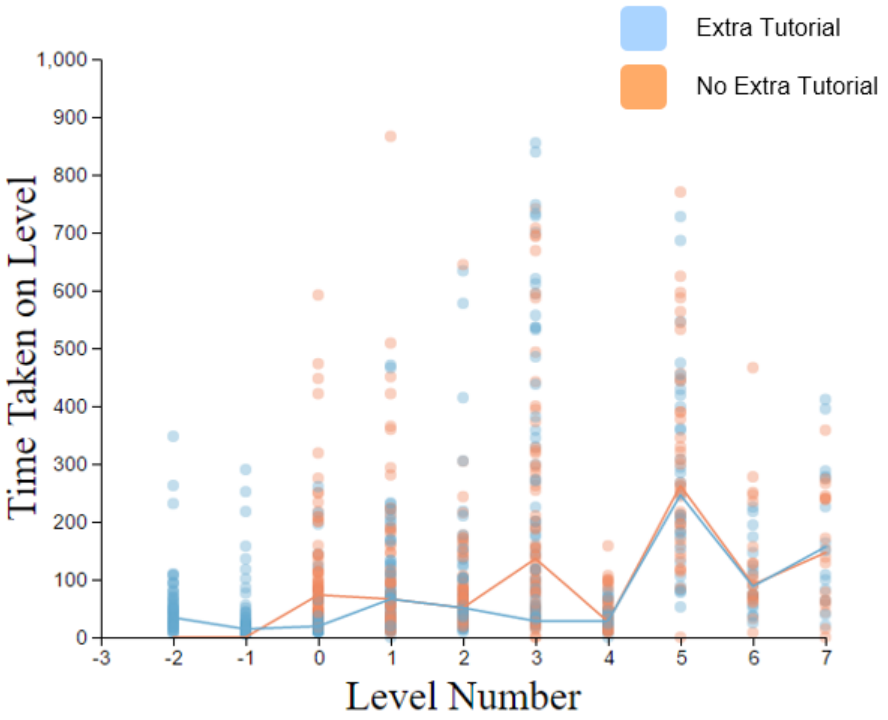


Figure 413. Level number vs. time taken graph. Image by Nick Chaput.

In the second full scale HIT, we expected to see more frustration in players who received our random-growth puzzle piece on the final level than in players who received the breadth-first search piece. While analysis of this hypothesis is not complete, the visualisations below are two examples of data which would support it. In the first graph, Figure 54, represents a player who received the random piece. The player's frustration obviously increases from their baseline when the receive the random mechanic. In contrast, Figure 55 shows an example of a player who did not receive the random puzzle piece. In this case we expect to see no change in base level anger across the levels.

Figure 424. Change in anger. Image by Nick Chaput.

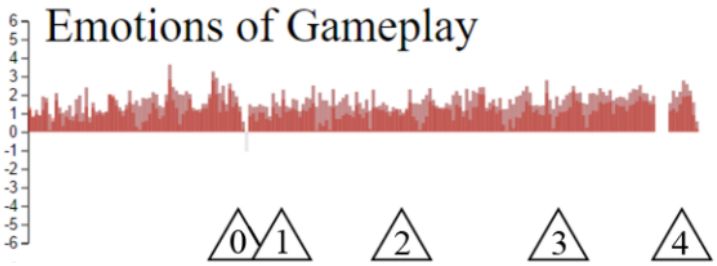


Figure 435. No change in anger. Image by Nick Chaput.

In both full scale HITs, we expected to see a spike of joy in the player within 10 seconds of completing a level. While analysis of this hypothesis is not complete, yet one example of this phenomena is observable in Figure 49.

In order to mathematically assess this hypothesis the team began the development of a collection of python scripts. These scripts serve as the basis of the pipeline required to do arbitrary analysis on the data. For the emotional data, this scripts provide Z-Scored values for all emotional channels. They then allow the user to sample from these Z-Scores for the period of time encompassing each level the player completed.

Unfortunately the team was not able to develop these scripts far enough to produce confident results which rely on emotional data before the end of this project. It proved to take too much time to write analysis scripts which were both tested for mathematical correctness and provided the necessarily statistics for analysis This scripts have been made available on a

public repository for future researchers to make use of.

3.10. Replay analysis

The replays were primarily used as evidence that the game was still working as intended for players in every iteration of *Eukaryote*. When the replay failed to play properly, it was typically an indication that we had failed to change our logging methods in a way that matched the changes to the game mechanics themselves.

The replays also served as a way to see if players were coming up with unintended solutions for our puzzles. While we typically support creative solutions that we had not come up with ourselves, we had to ensure that players were learning what they needed for future puzzles. An example of a puzzle we had to fix after watching replays was our introduction to zappers. We needed to teach players what the zappers were, which we did by attempting to create a puzzle where players were likely to hit the zappers both head on and tangentially. Unfortunately, players were able to create solutions that skirted the outside of the zappers completely, preventing them from learning everything about how the zappers worked. Once we saw this problem in the replays, it was easy to fix by simply adding an additional zapper to block the unintended solution.

We also used replays as a first pass to see what emotions players were expressing and to confirm if our hypotheses seemed like they might pan out or not. Our earliest runs were concerning, as the replays showed a handful of players with almost no discernible emotions as they played, as well as a couple players who covered their face with their hands throughout the entire session. Obscuring the face would pose a problem for the facial analysis software later on. When we ran our second small trial, however, Eukaryote had evolved further to include

sounds and smoother controls. We saw more expression on the players and those who did cover their faces did not do so for the entire session as they had in the first trial. Across all of these trials, we noted that players seemed to express primarily confusion and joy while playing, though for many players these emotions were quite subtle.

4. Post mortem

4.1. Connor Mattson

This project was a great learning experience for me. As an artist, I ended up creating all of the non-organic environmental assets such as the bridge, dock system, and swan boat. These assets were all very similar in their workflow and I ended up using a lot of the same materials and texture maps for these objects. I also produced clean, low-poly models for the other artists as starting points for the monuments in the game. I was the main artist for the ether fountains, the animated visions, the throwable seeds, and the skybox. I was also in charge of producing the trailer/teaser for our game, which was an iterative process. I did not expect to work on so many separate parts of the project and utilize so many different skills. Along with the expected contributions such as modeling and texturing, I learned how to render out clean trailer footage in Unreal, how to prepare a model for a game engine, how to utilize Unreal's blueprint system to make my artistic workflow easier and more efficient.

Prior to this project, I had not used Unreal Engine for 3D games. I worked on a 2D game in Unreal in 2016, but mostly did work outside of the engine. In hindsight, I wish I had done a small Unreal project by myself over the summer leading up into this MQP to familiarize myself with the interface and blueprint systems. Although I was learning as I went, I became comfortable with the workflow of modeling, UVing, importing to Unreal and texturing I had developed over the months. I created materials for and textured each of my models after I had already brought them into the engine. Because of our unique ambient lighting style, texturing

the models outside of the game environment proved to be not representative of what it would look like once imported into our game space.

Velum ending up having a very cohesive artistic look, yet doesn't distract players from the beautiful puzzles. We wanted to convey a certain mood and atmosphere alongside the tense music and theme rather than attempt photo-realistic objects while having a large time constraint. This was a great choice because it allowed us to subtly tweak the atmosphere and feel of the entire level quickly as the development period went on. After working alongside each other for a couple months, the art team realized a similar style and we brought everyone's work together and found a color scheme and visual style that complimented everyone's already made and future work.

4.2. August Beers

Spending a year developing a game concept with a five student team and three advisors has been a fascinating experiment in game design. Coming into this project I knew that I would have to scope out nearly two MQPs worth of work alongside a course load of three 4000 level computer science courses at WPI. While I took an average course load on paper, in reality I was performing as if I had an overloaded course load for the most of the year.

In the course of that time I completed two major works of software for the team. Firstly, I implemented *Eukaryote*, a game made in Professor Moriarty's game engine, Perlenspiel. The design of this simple grid-based linear puzzle game was used as a proof of concept for a second major piece of software I worked on this year. *Velum* is a 3-dimensional first-person puzzle game implemented in the Unreal 4 game engine. I wrote the functionally for the game alone, using a combination or Unreal C++ and Blueprint. At the end of this project, we had a game

with a 30 - 45 minute gameplay experience with a 9-level puzzle progression. I demonstrated that this game could run without crashing on two occasions for the entirety of the game's duration during formal playtesting in Professor Moriarty's class. Unfortunately, there was a bug in the game that could soft lock players into an unwinnable state during the first playtest of the game. In reflection, I now understand the tradeoffs of implementing a game with an open source game engine as opposed to utilizing my own. By choosing to use a tool implemented by others as my game engine I gave up some degree of control and understanding of my software from top to bottom. If I had taken the time to implement my own system, I feel I would have been more capable to address and solve bugs in elegant ways when they appeared. Additionally, by choosing to use Unreal Engine I also subjected my game to the problem that every time a new version of the Engine comes out I may have to change my implementation or fix a new bug that is introduced. If I was a more experienced programmer coming into this project, I would have implemented my own game engine to support our software.

Aside from developing these two games I also assisted my tech partner, Nicholas Chaput, in implementing a Mechanical Turk job that used *Eukaryote* to implement a formally designed study on our level progression. My major contribution to this aspect of the project was during the analysis phase. After we had collected data on more than 300 players, I created a data visualization using D3 and JavaScript which allowed us to do a preliminary exploration of the data set we had created. This visualization was used as a confirmation of the results of our analysis and to identify a series of bugs in the process of pipe lining the video footage of players faces through facial analysis software and into a useful data set.

I also made major contributions to a suite of Python scripts used to do mathematical

analysis of our data set. These scripts employed two libraries, Numpy and Scipy, to assist in the production and analysis of Z-scored values of our raw facial data. These Python scripts were developed with readability and clean design in mind so that other researchers may be able to use them in the future.

Unfortunately, at the end of our project we were unable to complete analysis of our large dataset on time. Had I known before how long it would take to run analysis of our data, I would have started working on analysis before the data came in. To run analysis our team has been developing a suite of Python scripts which try and find statistical measurements to support our hypothesis. It is totally possible that the team could have begun implementing these scripts on placeholder data before our actual data set came in. In this way, the tech team could have automated analysis before the dataset came in, so that once the data did come in at the end of C term, we would have been ready to disprove or approve our hypothesis in a much shorter period of time, possibly before the project is over.

My last piece of postmortem addresses how we chose to run our team. Working on a long term project at Worcester Polytechnic Institute is a serious time management problem. The difficulty lies in attempting to balance progress on distant project deadlines with the intense course load provided by WPI's short terms. The team operated for the entire year without formally establishing a manager or another leadership role on the team. I believe this lead to some of the team's problems with working effectively as a whole. For the duration of the fall, if one member of the team was failing to perform at an acceptable level, it was difficult for the group as a whole to apprehend that loss and react appropriately. Our lack of efficiency became very evident at the beginning of C term when the advisors let us know that we had not

been making substantial enough progress and were beginning to fall behind on our project. To address this feedback the team began to have official scrum meetings at the beginning of each team meeting. During scrum, each member of the team described what they had been working on prior to the meeting, what they planned to for the next meeting, and any problems they encountered. This system made it immediately obvious to the entire team if an individual team member was not putting in an equitable amount of work. Scrum meetings proved to be an effective mechanism to get the project back on track, however I think our project would have benefited greatly if we had chosen to elect a project manager to serve on a term-by-term basis. An elected project manager could have been utilized to create more structured and effective meetings while simultaneously keeping tabs on the other team members work history.

4.3. Connor Thornberg

This has been an extremely valuable project to me. I had some experience working in Unreal before, with *Revolver: Rebound*, but the creative control I had in this project allowed me to experiment and improve in many different ways. I primarily worked as the artist developing the organic assets. This included all of the trees, the grass, and parts of the monuments. I was also intimately involved in the creation and maintenance of the Unreal level. I ended up working on many different aspects of this project, and learned several skills I had not anticipated using for this project. An example of this would be the physics-simulation vertex animation and alembic caching for the willow tree.

The trees and their foliage were my largest contribution to this project. I decided to create the trees modularly, so that we could create many different configurations, to avoid repetitiveness in the garden. This process proved to be much more work than I had anticipated.

I made three to four branches and a trunk for each tree. The branches began as ZSpheres, and then were subdivided and sculpted on. I then used XNormal with both high and low poly meshes to get wireframe, normal, and occlusion maps. I then had to create color and specular maps, before importing it all into engine and creating a more complex BSP material. Having to do this for each modular component of each tree was a lot of work, and took longer than expected. Once the components were finished, I assembled them into different configurations. I also had to create color, emissive, opacity, and specular maps for the leaves adorning each tree, and then scatter these leaves on each of the different configurations. I had originally planned for seven different trees, but had to scope down to four trees in the end. I wish I had investigated this process further early in the project, so that I would have been able to plan for the project better. I still plan to create the outstanding trees (Dawn Redwood, Maple, Tea Crabapple) as we continue to develop the game. Hopefully, this will create a more immersive and compelling landscape.

I think that part of the reason that our team was successful was that we were comfortable enough with each other to honestly critique work, and to do so in a constructive way. Aside from being partners in game development, we are all close friends. I feel that this underlying friendship helped our group dynamic, as it encouraged us to hold each other to a high standard. It also helped us stay excited about the project and what we could accomplish as it progressed.

This project has taught me a tremendous amount about how to develop an idea over a long period of time. In particular, I've become much more experienced in documenting this sort of process, and it has helped our project tremendously. We've had many changes to the

themes and vision of the game throughout development. Most of these discussions were at our daily meetings, and were recorded in the notes for that day. These notes helped us to keep track of all of our ideas, regardless of how inconsequential they seemed at the time. In fact, several ideas which were originally proposed offhand have now been incorporated into the main design of the game. For example, the idea of "the ether patient's thoughts bleeding through" was mentioned during a meeting on April 1, 2016 (Appendix J). It was then revisited when Marco joined the team and we were working to create the narrative. In addition, I still feel compelled to develop some of the concepts we abandoned at the start of this project.

I regret not having developed a more significant social media presence before bringing the game to PAX East. We had many interested players, and could have used that to gain a following. However, I don't believe that the simple website we created was enough to keep the attention of players throughout the development cycle. I would encourage new MQP teams to begin a social media presence as early as possible, posting the occasional snapshot of work to keep people interested. This could be under a developer team name, so that you don't have to decide on the game title and then change it. It might even be worthwhile to create the shell of a website and be ready to launch it. By the time that we had decided on a title, we were so busy with other aspects of the project that a website wasn't something we could devote time to.

Overall, I'm very pleased with how this project turned out. The most satisfying part for me is that I feel like we actually captured the mysterious, timeless, twilight atmosphere that we had sought to create. In addition, I think that our puzzle mechanic has incredible potential, and could be used to create some really interesting, compelling puzzles if used correctly.

4.4. Jie Weng

In this project, I was mainly focusing on the squirrel with rigging and animation, as well as the roots and characters on the monuments. Working on rigs and animation is also what I personally want to focus on for the future, therefore, working on this project is a very valuable experience before I enter the game industry.

Before I worked on this project, I favored animation the most. Although I learned about rigging in lectures and online videos, rigging was still a small challenge to me. One of the hardest points was when we decided to replace the placeholder characters in the game with squirrels in early development. I never practiced on any realistic animals in either 2D or 3D, and I was not knowledgeable about animal anatomy, either. So the decision to make squirrels with animations was challenging but also stimulating for me.

When I finished the first placeholder version of squirrel animations, they were not as lifelike as we wanted, and the tail waved like a beaver's tail. But during the self-exploration on rigs and animation structures, I found a way to solve the tail issue. After I fixed the tail, the animations looked much better. Nevertheless, the way I fixed the tail was not as good as the suggestion that my advisor, Ralph Sutter, gave me later on. I gained a deeper understanding of both rigs and animals during my self-exploring. Also, because of this experience of making the squirrel, I found myself more confident when I moved on to the horse on the George Washington statue. But on the other hand, self-exploration cost me so much time that I could have had more time to make assets if I had talked to my advisors more.

For the art team, we gave each other a lot of freedom on what individuals are working on, and I didn't give much criticism on other artists' works during the early development. I was

concerned that might be resulting in some styling issues, since we all had different visions of the game in our minds. But when we approached later development, we explored materials and textures together, and had a vision that we were all satisfied with in the end.

One of the things I regret the most is during the developing period of squirrel and root, I didn't work close enough with the technical side. Consequently, some issues appeared when we compiled art assets and code together. Most of the issues were solvable without me re-doing all the work, but there would have been a better outcome if I worked with the tech team closer from the beginning.

4.5. Nicholas Chaput

This project taught me a lot about working on a long-term project. As a project that has lasted more than a year, it is by far the longest project I have worked on with a mid-sized team. I am making this distinction about team size because it means that while I was not responsible for working on every aspect of the project, I was still staying familiar with and discussing the decisions related to each part of it. There is no part of the project that I felt completely uninvolved in. Outside of group decision-making, I was primarily responsible for setting up and executing the analytical study through Mechanical Turk. I also contributed puzzle ideas during the prototyping phase.

In regards to general group work, I learned that decisions need to be treated as concrete tasks with concrete deadlines. With regular tasks, if we couldn't complete them in time, we would stick with what we did complete and come back to it later if we had time. We unfortunately did not take this approach with making decisions in the planning phase of the project, which made it hard to start working on the project and take full advantage of the extra

term we gave ourselves for the project. In particular, we were too slow in choosing our primary puzzle mechanic and deciding on a cohesive narrative. Each of those decisions made us significantly more productive going forward, most likely because we knew exactly how to tell if our work was appropriate for the game as whole.

In regards to my work on the study, I was frequently too slow to ask for help when I was not making progress. I was fine about asking for help when I truly felt stuck, but as long as I was able to find more things to try, I would simply continue struggling by myself instead of asking my teammates and advisers for help. This led to many tasks taking longer than they should have to complete, and pushed back the run of our original Mechanical Turk HIT, which left us low on time to run and analyze later HITs. It also left me without time to help August within Unreal as much as I should have, which put strain on the scope of that entire side of the project. I have since learned to answer the question “should I ask for help?” by thinking about how long it has been since I made progress, rather than whether or not I have more ideas that could work.

Another significant challenge was trying to work on both this project and my 4000 level CS courses. This project alone was a significant time commitment, and those courses were often just as much of a commitment if not more so. Especially during midterms and finals, I often had to choose between finishing my work for the classes or for this project. I don’t think there is any good solution to this issue, as it isn’t feasible to simply recommend students take their 4000 level courses prior to starting their MQP or after it finishes. Instead, I simply warn future students that this is a problem they will almost certainly encounter and they should do their best to do as much work during the non-exam seasons as they can.

This project was my first experience setting up an Apache web server with PHP support by myself. I had worked with apache by following a tutorial for one of my previous classes, and I had worked with PHP within an existing framework at my internship last year, but I had never done this on my own before. I learned a lot about issues that can occur when putting something online, especially how easy it can be to violate the Same Origin Policy while doing things that seem like they should be perfectly reasonable to do. The biggest source of this issue was attempting to use Professor Moriarty's Perlenspiel engine on our page. The engine was hosted on a different domain (users.wpi.edu rather than cs.wpi.edu), which led to a variety of the engine's resources not being loaded. In order to fix this, we had to receive a copy of the engine and host it directly on our server. We also had to get our server certified for HTTPS in order to integrate it into a Mechanical Turk HIT, which we did with Michael Voorhis' assistance.

Amazon Mechanical Turk platform was also a source of challenges. Simply put, support for MTurk is lacking. The official documentation is outdated, often contradicting itself or simply claiming that things exist when they don't. If Professor Whitehill did not have experience with the platform before this project, I am not confident we would have been able to make things work for our purposes. Looking back on things, there are countless decisions about how we implemented our game into MTurk that I would have changed to allow for greater readability and extensibility, but I simply did not understand enough about the process to do so at the time.

My largest regret with this project is that we did not leave enough time to thoroughly analyze all the data we collected. We believed we had finished collecting and organizing the data by the end of C term, leaving us several weeks of D-term to analyze it. Unfortunately, we

didn’t expect that I had failed to account for missing facial images, possibly due to dropped packets, which caused our facial data to be out of sync from our log and statistical data. Correcting this issue took time and effort we had planned to put into the analysis itself, pushing it back until shortly before we needed to be working on the paper and presentation, as well as pushing for Velum to be ready for several game festivals. I am currently looking into how feasible it is for me to go back to this after the project is officially over.

Works cited

Dakic V, Maciel RdM, Drummond H, Nascimento JM, Trindade P, Rehen SK. “Harmine stimulates proliferation of human neural progenitors.” *PeerJ* 4:e2727. (2016) Web. URL: https://doi.org/10.7717/peerj.2727

Fenster, J. M. *Ether Day: The Strange Tale of America's Greatest Medical Discovery and the Haunted Men Who Made It*. New York: Perennial, 2002. Print.

Gamma, Erich, et al. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison Wesley Professional, Boston (1994): 315-325.

Krasner, Glenn E., and Stephen T. Pope. "A description of the model-view-controller user interface paradigm in the smalltalk-80 system." *Journal of Object Oriented Programming* 1.3 (1988): pp. 26-49.

Littlewort G, Whitehill J, Wu T, Fasel I, Frank M, Movellan J, Bartlett M. “The Computer Expression Recognition Toolbox (CERT), in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. IEEE; 2011 : pp. 298–305.

Sochman, Jan, and Jiri Matas. "Waldboost-learning for Time Constrained Sequential Detection." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 2. IEEE, 2005.

Viola, Paul, and Michael J. Jones. "Robust Real-Time Face Detection." *International Journal of Computer Vision* 57.2 (2004): 137-154.

Whitehill J, Serpell Z, Lin Y-C, Foster A, Movellan J. “The Faces of Engagement: Automatic Recognition of Student Engagement from Facial Expressions.” 2014.

Whitehill, Jacob, and Margo Seltzer. "A Crowdsourcing Approach to Collecting 399 Tutorial Videos on Logarithms." *arXiv preprint arXiv:1606.09610* (2016).

Appendix A: Narrative script

Narrative beats

1. Warren: “Ah. Nice for you to finally arrive, Dr. Morton.”
2. Morton: “Take your seat, Abbott.”
3. Morton: “Gentlemen ... I have just this morning created a new apparatus to safely administer ether.”
4. Morton: “Now observe how the ether affects the patient.”
5. Warren: “Abbott? ... Abbott! ... Are you awake?”
6. Warren: “How does your neck feel?”
7. Abbott: “Well ... It feels as if my neck’s been scratched.”
8. Warren: “Gentlemen, this is no humbug.”

‘Peak’ lines

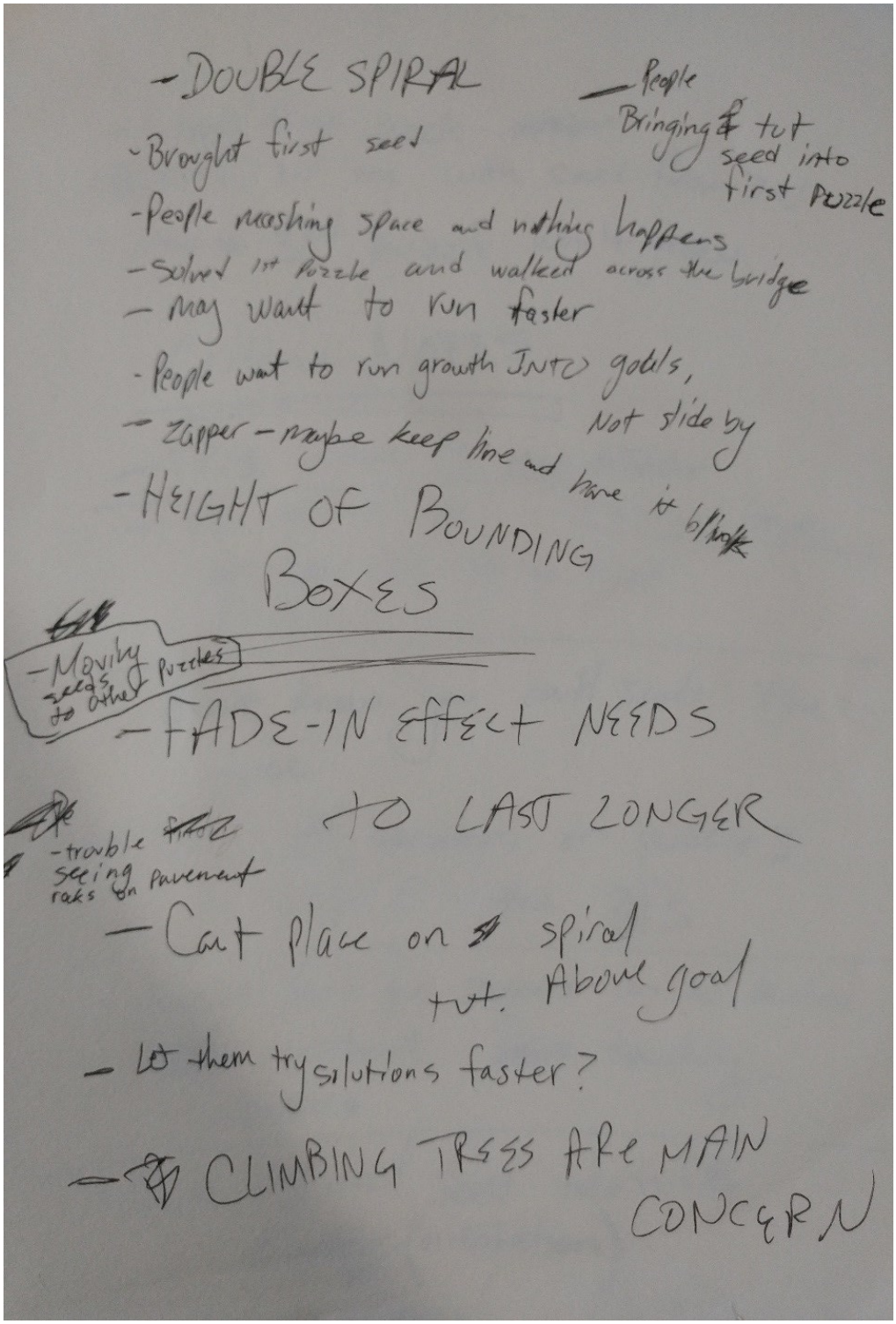
1. “Imagination is the only weapon in the war against reality.”
2. “But I don’t want to go among mad people," Alice remarked.
   "Oh, you can’t help that,” said the Cat, “we’re all mad here. I’m mad. You’re mad."
   "How do you know I’m mad?" said Alice.
   "You must be,” said the Cat, “or you wouldn’t have come here.”

Miscellaneous lines

1. “This scalpel is filthy...just the way I like it.”
2. “We will be using twelve stitches in Abbott’s neck.”
3. “Pass the scalpel.”

Appendix B: Playtest notes

February 17, 2017



- Solved first puzzle ~~with a~~
- ~~Re~~solved 2nd one with seed from 3rd
- Now ~~the~~ 3rd puzzle won't grow.

Critique NOTES

- TREE issue is Main ~~relation~~ technical issue
- Use squirrel to tell user how to do first Puzzle
- Didn't know you could scale the tree
- Cut off openings on puzzle 3 to FORCE the BFS
- tutorial level for zappers can be beaten ~~with~~ without ~~flow~~ touching zappers
- fix Spiral seed texture ~~(color)~~ (orientation)

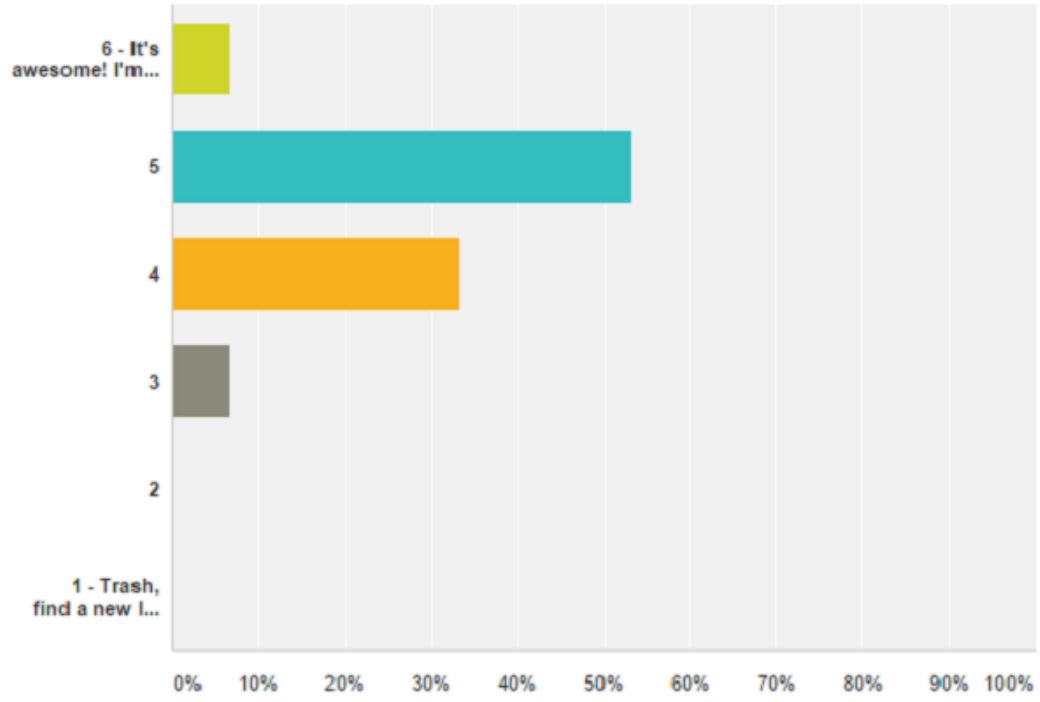
Appendix C: Playtest survey

February 17, 2017

Q1 Customize Export

How do you like the visual design of the game?

Answered: 15 Skipped: 0



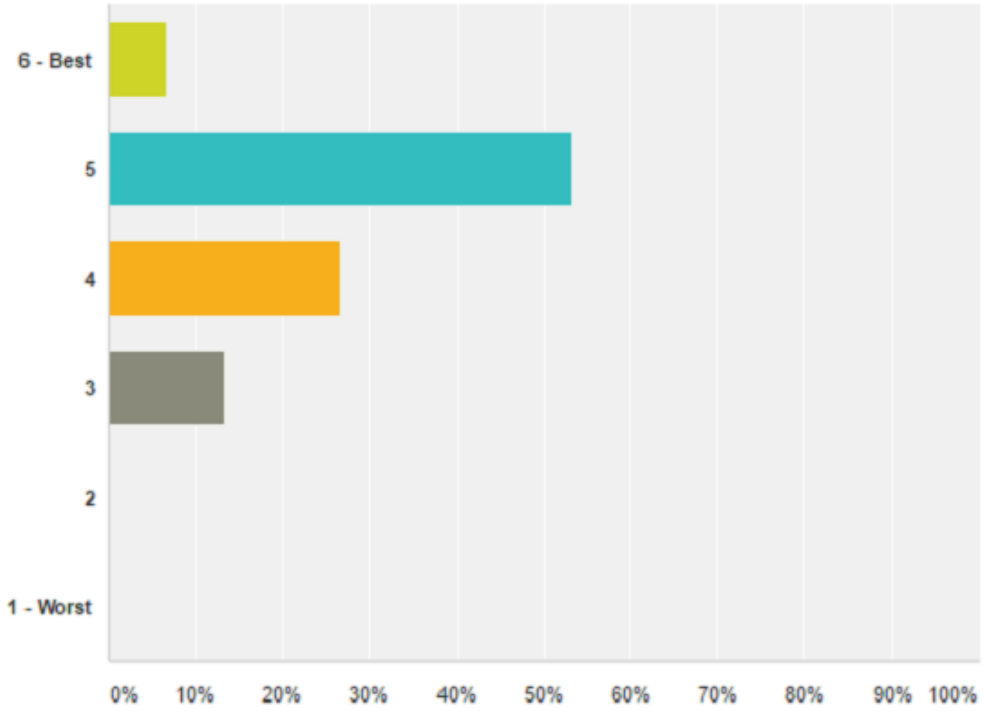
Answer Choices	Responses	
6 - It's awesome! I'm gonna frame this screenshot and hang it above my bed.	6.67%	1
5	53.33%	8
4	33.33%	5
3	6.67%	1
2	0.00%	0
1 - Trash, find a new line of work.	0.00%	0
Total		15

Q2

Customize Export

How satisfying was the puzzle progression?

Answered: 15 Skipped: 0

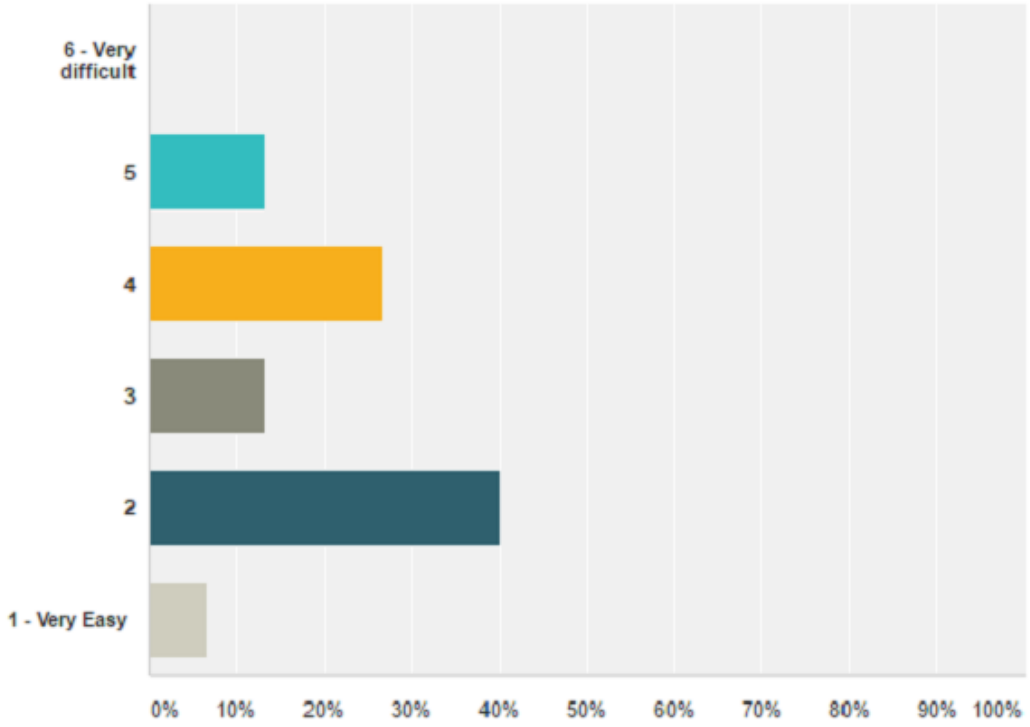


Answer Choices	Responses	
6 - Best	6.67%	1
5	53.33%	8
4	26.67%	4
3	13.33%	2
2	0.00%	0
1 - Worst	0.00%	0
Total		15

Q3 Customize Export

How difficult was it figure out where to go?

Answered: 15 Skipped: 0



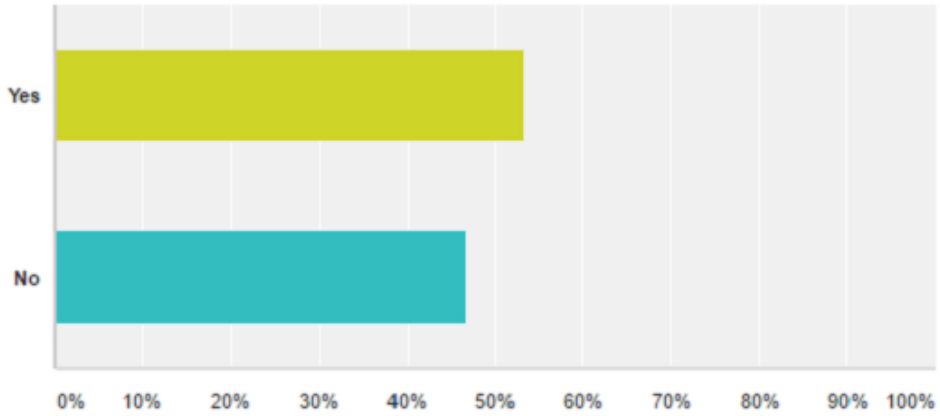
Answer Choices	Responses	
6 - Very difficult	0.00%	0
5	13.33%	2
4	26.67%	4
3	13.33%	2
2	40.00%	6
1 - Very Easy	6.67%	1
Total		15

Q4

Customize Export

Was the squirrel at the beginning an effective guide?

Answered: 15 Skipped: 0



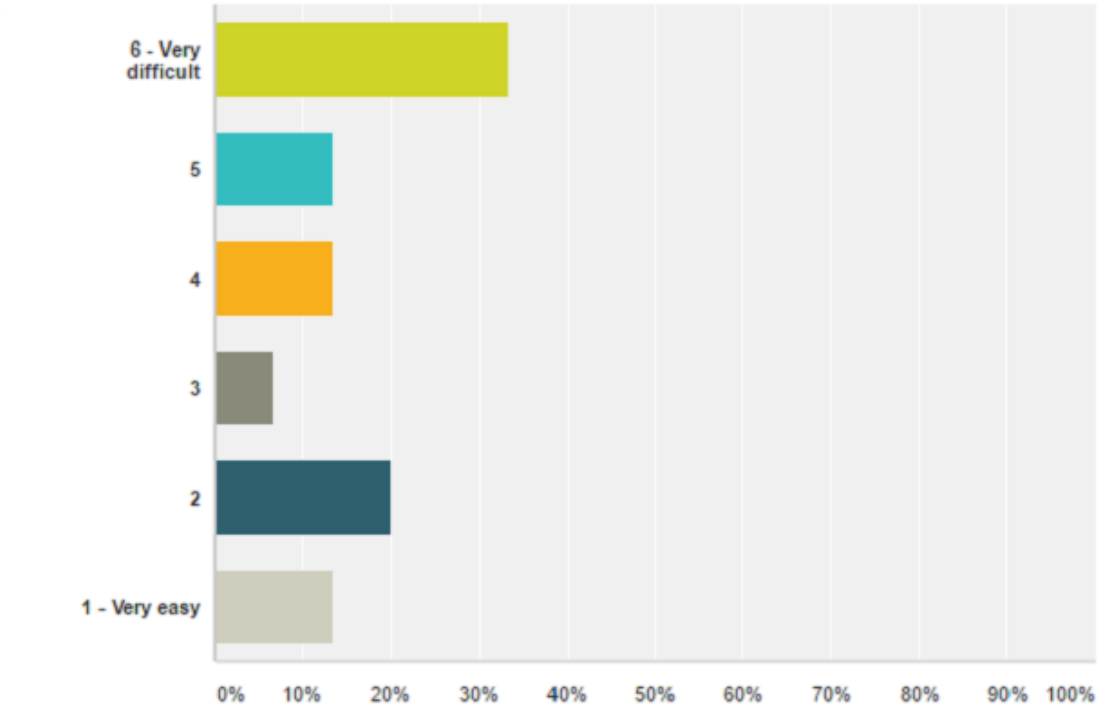
Answer Choices	Responses	
Yes	53.33%	8
No	46.67%	7
Total		15

Q5

Customize Export

Did you have difficulty figuring out how to climb a tree?

Answered: 15 Skipped: 0



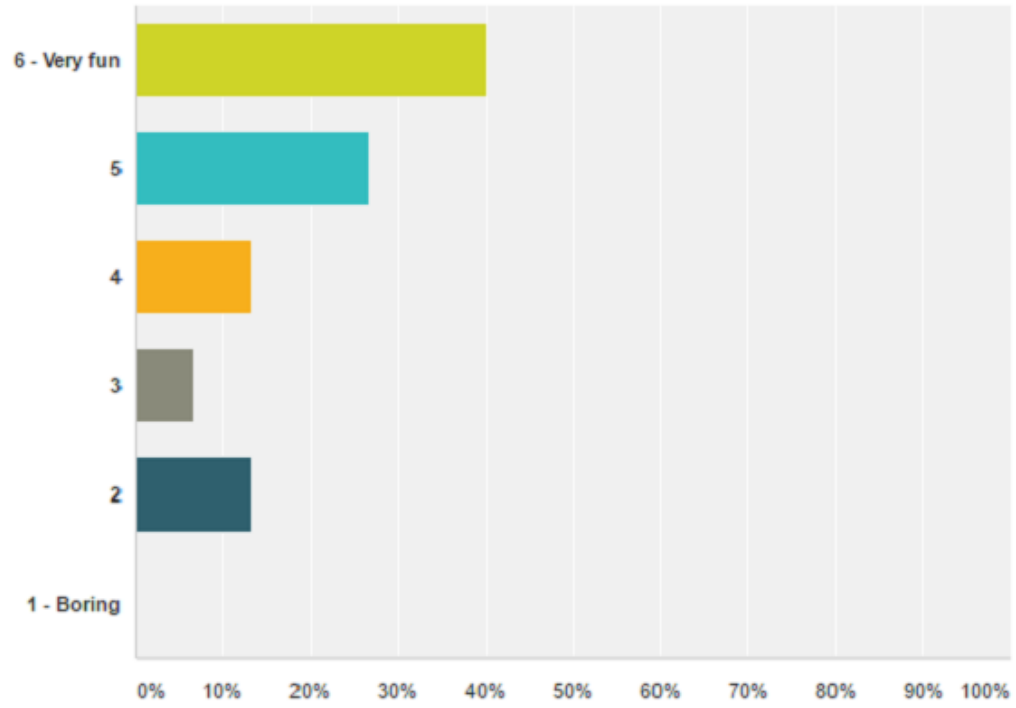
Answer Choices	Responses	
6 - Very difficult	33.33%	5
5	13.33%	2
4	13.33%	2
3	6.67%	1
2	20.00%	3
1 - Very easy	13.33%	2
Total		15

Q6

Customize Export

Did you enjoy the puzzle mechanic?

Answered: 15 Skipped: 0

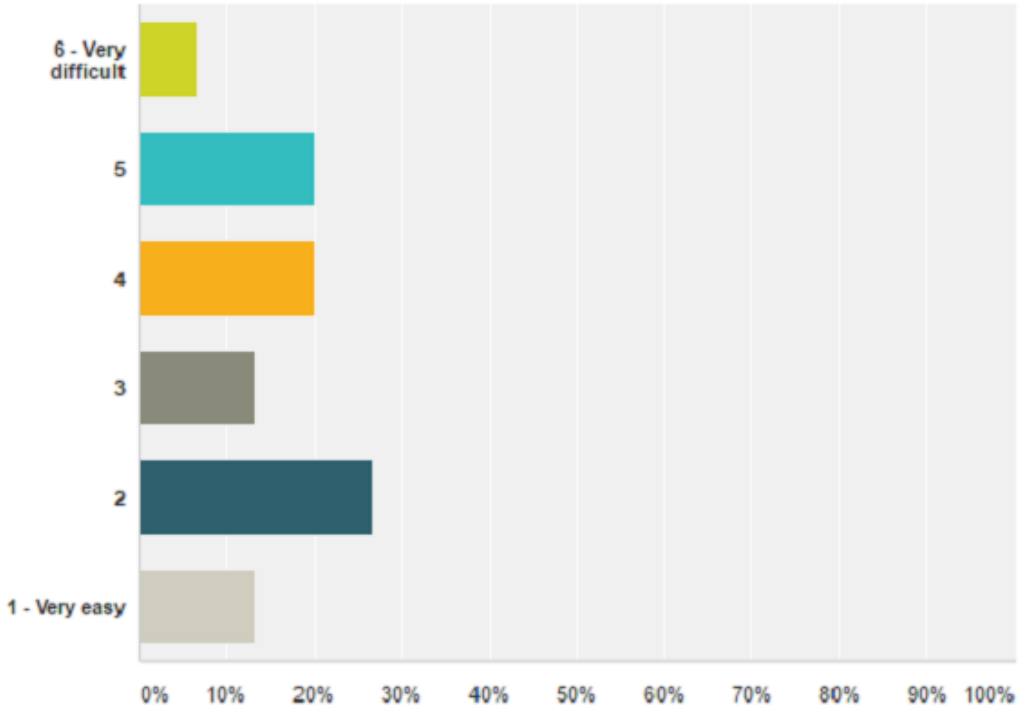


Answer Choices	Responses	
6 - Very fun	40.00%	6
5	26.67%	4
4	13.33%	2
3	6.67%	1
2	13.33%	2
1 - Boring	0.00%	0
Total		15

Q7 Customize Export

How difficult were the seeds (puzzle pieces) to place?

Answered: 15 Skipped: 0



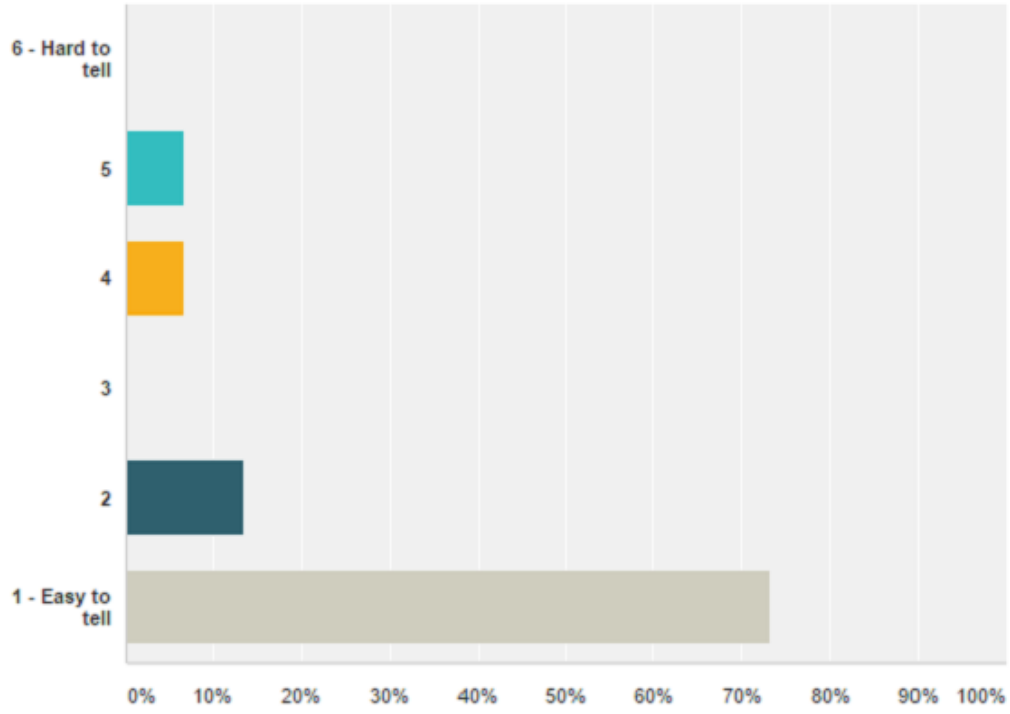
Answer Choices	Responses	
6 - Very difficult	6.67%	1
5	20.00%	3
4	20.00%	3
3	13.33%	2
2	26.67%	4
1 - Very easy	13.33%	2
Total		15

Q8

Customize Export

Were the puzzle pieces easily distinguishable from each other?

Answered: 15 Skipped: 0



Answer Choices	Responses	
6 - Hard to tell	0.00%	0
5	6.67%	1
4	6.67%	1
3	0.00%	0
2	13.33%	2
1 - Easy to tell	73.33%	11
Total		15

Q9. If you could change one thing about the game, what would it be?

- Climbing the trees was a bit difficult. It was a bit frustrating to climb to the top and then be pushed off by a branch.

- Make it clearer that the puzzle is to link the blue lights together, and perhaps create tutorials for the non-directional seeds. I didn't fully understand how the spread seeds and the spiral seeds worked. Also, a sprint function would be really useful.

- The trees are very hard to climb. From one side, a branch will push you off as soon as you reach the top, and from the middle, the collision of the platform prevents you from reaching the top.

- Sprint key.

- The music was kind of grating. A more minimalist soundscape would serve the game's mystery better. The tree climbing felt weird too. The only other suggestion would be 1 more puzzle before the first. perhaps, only 1 seed or 1 row, that way, no matter what, the player would know how the mechanic worked (I solved the first puzzle accidentally, and was stuck on the next one).

- Change the collider system to allow water walking.

- The reward music bordered on irritating due to it lasting much longer than the time it took to traverse from puzzle to puzzle, maybe consider lowering its duration?

- Fixing the issue with the tree climbing and seed placing is the number one concern I have. Tree climbing is finicky, collisions are odd on the platform, and there's not enough of an indicator that you can actually climb. The seed placing results in lost seeds. I only managed to finish my first playthrough after losing a seed on the 2nd to last puzzle (lost the checkerboard) by finding the horizontal seed I lost on the first puzzle lying on the ground.

- Make the puzzles update in real time. Add a sprint key. Make the puzzle less wait. Fix the trees.

- Change space to be a jump button. A jump button isn't necessary, but the game felt like it was lacking one.

- Make it more obvious you can climb the tree. :(

- P L E A S E L E T M E C L I M B T R E E S B E T T E R. Honestly that was the only issue I had with the game outside of the visual bugs. The game looks real sweet, my dudes!

- Make trees easier to climb.

- Make the squirrel guide easier to notice. I just thought it was a random visual effect. Also, some of the trees are just annoying to climb.

Appendix D: PHP scripts

Early consent log

```
<?php
       // requires php5
       $id = $_POST['id'];
       $worker = $_POST['worker'];
   $assignment = $_POST['assignment'];
       $consent = $_POST['consent'];
       $consent_date = $_POST['consentDate'];
       $release = $_POST['release'];
       $release_date = $_POST['releaseDate'];
       $our_use_allowed = $_POST['ourUse'];
       $future_use_allowed = $_POST['futureUse'];


   if (strcmp($assignment,'LOCALPLAYER') == 0) {
       $path = '../../../images/local/' . $id;
   } else {
       $path = '../../../images/mturk/study4/' . $worker . '/' . $assignment . '/' .
$id;
   }

       if (!is_dir($path)) {
        // dir doesn't exist, make it
        mkdir($path, 0775, true);
       }

       define('UPLOAD_DIR', $path . '/');

       $log_text = '';
       $log_text = $log_text . 'Timestamp: ' . $id . PHP_EOL;
       $log_text = $log_text . 'Worder ID: ' . $worker . PHP_EOL;
       $log_text = $log_text . 'Assignment ID: ' . $assignment . PHP_EOL;
       $log_text = $log_text . 'Consent Given (Consent Form): ' . (($consent) ? 'true'
: 'false') . PHP_EOL;
       $log_text = $log_text . 'Consent Date (Consent Form): ' . $consent_date .
PHP_EOL;
       $log_text = $log_text . 'Consent Given (Release Form): ' . (($release) ? 'true'
: 'false') . PHP_EOL;
       $log_text = $log_text . 'Consent Date (Release Form): ' . $release_date .
PHP_EOL;
       $log_text = $log_text . 'Consent for OUR USE: ' . (($our_use_allowed == 'true')
? 'true' : 'false') . PHP_EOL;
       $log_text = $log_text . 'Consent for FUTURE USE: ' . (($future_use_allowed ==
'true') ? 'true' : 'false') . PHP_EOL;
       $log_text = $log_text . PHP_EOL;

   $stat_file = UPLOAD_DIR . 'consent_log_early.txt';
   $success = file_put_contents($stat_file, $log_text, FILE_APPEND);
   print $success ? $stat_file : 'Unable to log the entry.';
?>
```

Consent log

```
<?php
       // requires php5
       $id = $_POST['id'];
       $worker = $_POST['worker'];
   $assignment = $_POST['assignment'];
       $consent = $_POST['consent'];
       $consent_date = $_POST['consentDate'];
       $release = $_POST['release'];
       $release_date = $_POST['releaseDate'];
       $our_use_allowed = $_POST['ourUse'];
       $future_use_allowed = $_POST['futureUse'];


   if (strcmp($assignment,'LOCALPLAYER') == 0) {
        $path = '../../../images/local/' . $id;
   } else {
        $path = '../../../images/mturk/study4/' . $worker . '/' . $assignment . '/' .
$id;
   }

       if (!is_dir($path)) {
        // dir doesn't exist, make it
        mkdir($path, 0775, true);
       }

       define('UPLOAD_DIR', $path . '/');

       $log_text = '';
       $log_text = $log_text . 'Timestamp: ' . $id . PHP_EOL;
       $log_text = $log_text . 'Worder ID: ' . $worker . PHP_EOL;
       $log_text = $log_text . 'Assignment ID: ' . $assignment . PHP_EOL;
       $log_text = $log_text . 'Consent Given (Consent Form): ' . (($consent) ? 'true'
: 'false') . PHP_EOL;
       $log_text = $log_text . 'Consent Date (Consent Form): ' . $consent_date .
PHP_EOL;
       $log_text = $log_text . 'Consent Given (Release Form): ' . (($release) ? 'true'
: 'false') . PHP_EOL;
       $log_text = $log_text . 'Consent Date (Release Form): ' . $release_date .
PHP_EOL;
       $log_text = $log_text . 'Consent for OUR USE: ' . (($our_use_allowed == 'true')
? 'true' : 'false') . PHP_EOL;
       $log_text = $log_text . 'Consent for FUTURE USE: ' . (($future_use_allowed ==
'true') ? 'true' : 'false') . PHP_EOL;
       $log_text = $log_text . PHP_EOL;

   $stat_file = UPLOAD_DIR . 'consent_log.txt';
   $success = file_put_contents($stat_file, $log_text, FILE_APPEND);
   print $success ? $stat_file : 'Unable to log the entry.';
?>
```

Log event sequence

```
<?php
       // requires php5
       $id = $_POST['id'];
       $log_array = $_POST['logArr'];
       $img = $_POST['img'];
       $worker = $_POST['worker'];
       $assignment = $_POST['assignment'];


       if (strcmp($assignment,'LOCALPLAYER') == 0) {
          $path = '../../../images/local/' . $id;
       } else {
          $path = '../../../images/mturk/study4/' . $worker . '/' . $assignment . '/'
. $id;
       }

       if (!is_dir($path)) {
        // dir doesn't exist, make it
        mkdir($path, 0775, true);
       }

       define('UPLOAD_DIR', $path . '/');

       foreach ($log_array as $log_entry) {
          $components = explode(' ', $log_entry);
          if ($components[0] === '00') {
           $img = str_replace('', '', $img);
           $img = str_replace(' ', '+', $img);
           $data = base64_decode($img);
           $file = UPLOAD_DIR . $components[3] . '.jpg';
           $success = file_put_contents($file, $data);
           print $success ? $file : 'Unable to save the file.';
          }
          $replay_log_text = ($components[3] - $id) . ' ' . $log_entry . PHP_EOL;
       $replay_log_file = UPLOAD_DIR . 'replay_log.txt';
       $success = file_put_contents($replay_log_file, $replay_log_text, FILE_APPEND);
       print $success ? $replay_log_file : 'Unable to log the entry.';
       }
?>
```

Log stats

```
<?php
       // requires php5
       $id = $_POST['id'];
       $level_num = $_POST['levelNum'];
       $level_start = $_POST['start'];
       $level_finish = $_POST['end'];
       $growths_started = $_POST['growths'];
       $is_level_complete = $_POST['isLevelComplete'];
       $level_progression = $_POST['levelProgression'];

       $level_duration = $level_finish - $level_start;


       $worker = $_POST['worker'];
   $assignment = $_POST['assignment'];


   if (strcmp($assignment,'LOCALPLAYER') == 0) {
       $path = '../../../images/local/' . $id;
   } else {
       $path = '../../../images/mturk/study4/' . $worker . '/' . $assignment . '/' .
$id;
   }

       if (!is_dir($path)) {
        // dir doesn't exist, make it
        mkdir($path, 0775, true);
       }

       define('UPLOAD_DIR', $path . '/');

       $log_text = '';
       $log_text = $log_text . 'Level Progression: ' . $level_progression . PHP_EOL;
       $log_text = $log_text . 'Level: ' . $level_num . PHP_EOL;
       $log_text = $log_text . 'Level Began at: ' . ($level_start - $id) . PHP_EOL;
       $log_text = $log_text . 'Level Ended at: ' . ($level_finish - $id) . PHP_EOL;
       $log_text = $log_text . 'Number of Tries: ' . $growths_started . PHP_EOL;
       $log_text = $log_text . 'Time Taken: ' . ($level_finish - $level_start) .
PHP_EOL;
       $log_text = $log_text . 'Level Completed: ' . (($is_level_complete == 'true') ?
'true' : 'false') . PHP_EOL;
       $log_text = $log_text . PHP_EOL;

   $stat_file = UPLOAD_DIR . 'stat_log.txt';
   $success = file_put_contents($stat_file, $log_text, FILE_APPEND);
   print $success ? $stat_file : 'Unable to log the entry.';
?>
```

Appendix E: Raw data in TSV format

#Study : Feb 27th 2017 Study 6 Complete
#Date : 20170227 15:02:00
#Version : 6.2.4912.4
#TestRunDate : 20170227 00:09:43.619 (-05:00)

Frame Time	NoOf Faces	Joy Evidence	Anger Evidence	Surprise Evidence	Fear Evidence	Contempt Evidence	Disgust Evidence	Sadness Evidence	Confusion Evidence	Frustration Evidence	Neutral Evidence	Positive Evidence	Negative Evidence
1000	0												
1500	1	-6.200437	-2.42191	-2.982088	-4.028389	-2.173802	-3.093202	-1.532394	-0.4736539	-1.754694	2.564289	-6.200437	-1.532394
2000	1	-4.069039	-1.237085	-2.524481	-2.330055	-1.162178	-2.800788	-1.416977	-0.02352766	-0.7032072	1.918279	-4.069039	-1.162178
2500	1	-2.574711	-1.859237	-2.806222	-2.201826	-0.00425862	-2.21017	-1.506213	-0.7036955	-0.9193746	1.774172	-2.574711	-0.004258622
3000	1	-2.893964	-2.299439	-2.583157	-2.086374	0.3318176	-2.471078	-1.677343	-1.152312	-1.413989	1.837152	-2.893964	0.3318176
3500	1	-2.646933	-1.917552	-2.692368	-2.191918	0.132055	-2.252928	-1.587982	-0.7977167	-1.013925	1.73288	-2.646933	0.132055
4000	1	-2.802607	-1.951894	-2.698189	-2.179699	0.3441617	-2.279834	-1.656804	-0.8665079	-1.116012	1.627394	-2.802607	0.3441617
4500	1	-2.915437	-2.110003	-2.530209	-2.218735	0.488801	-2.420398	-1.877516	-1.042497	-1.310197	1.650486	-2.915437	0.488801
5000	1	-3.04003	-2.300163	-2.435759	-2.26652	0.2727389	-2.593034	-1.858303	-1.158355	-1.438262	1.835443	-3.04003	0.2727389
5500	1	-2.905082	-1.887446	-2.463699	-2.073673	0.3862698	-2.427767	-1.956536	-1.070329	-1.265306	1.651824	-2.905082	0.3862698
6000	1	-2.957751	-1.579706	-2.73594	-2.157581	0.3196126	-2.309123	-1.964953	-0.8028718	-1.037455	1.429213	-2.957751	0.3196126
6500	1	-2.606804	-1.934845	-2.018486	-1.721874	0.412021	-2.481113	-2.193817	-1.410769	-1.536118	1.695037	-2.606804	0.412021
7000	1	-2.859726	-1.847011	-2.3016	-1.937164	0.4886189	-2.453744	-2.184819	-1.251633	-1.417597	1.586368	-2.859726	0.4886189
7500	1	-3.293444	-2.132425	-2.513881	-2.128304	0.5975722	-2.570828	-1.958332	-1.221126	-1.514349	1.597863	-3.293444	0.5975722
8000	1	-2.71869	-2.537163	-2.307059	-2.106048	0.4084696	-2.657748	-1.970039	-1.447496	-1.639955	1.900996	-2.71869	0.4084696
8500	1	-4.1418	0.3782974	-3.995508	-2.652004	-1.114129	-2.469605	-1.492143	1.585301	1.001278	-0.00301970	-4.1418	0.3782974
9000	1	-3.676001	-0.3664025	-3.753899	-2.710481	-0.3481314	-2.427928	-1.749763	0.7264118	0.282115	0.5507234	-3.676001	-0.3481314
9500	1	-3.500696	-1.133767	-3.328735	-2.507217	-0.06398835	-2.591191	-1.708273	0.1023487	-0.3783749	1.132594	-3.500696	-0.06398835
10000	1	-2.782222	-1.212988	-2.755558	-2.376308	0.1232301	-2.482095	-1.939473	-0.2231804	-0.4909582	1.245885	-2.782222	0.1232301
10500	1	-2.995781	-1.350747	-2.851127	-2.488862	0.1418603	-2.517549	-1.915252	-0.2652637	-0.5981027	1.259569	-2.995781	0.1418603
11000	1	-3.27688	-1.79399	-3.040719	-2.616432	0.3755226	-2.647421	-1.777318	-0.52732	-0.9101627	1.39502	-3.27688	0.3755226
11500	1	-2.82164	-1.302198	-2.651748	-2.289919	0.3522571	-2.306545	-2.069058	-0.5977979	-0.82389	1.28958	-2.82164	0.3522571
12000	1	-2.986917	-1.247306	-2.854926	-2.313956	0.5638364	-2.237046	-2.026618	-0.4960419	-0.780171	1.15049	-2.986917	0.5638364
12500	1	-2.893905	-1.920579	-2.637148	-2.877854	0.5302885	-2.21572	-1.695041	-0.5625427	-0.945536	1.535087	-2.893905	0.5302885
13000	1	-2.49563	-2.02719	-2.258783	-2.554988	0.5874377	-2.195579	-1.830206	-0.8777454	-1.194997	1.667068	-2.49563	0.5874377
13500	1	-2.509737	-1.975595	-2.412522	-2.692643	0.6319523	-2.189128	-1.865581	-0.7820247	-1.08703	1.644392	-2.509737	0.6319523
14000	1	-3.321059	-1.40296	-2.725834	-2.681337	0.5530781	-2.078839	-1.849077	-0.4776311	-0.8853412	1.325874	-3.321059	0.5530781
14500	1	-2.9605	-2.083907	-2.500493	-2.615644	0.6072422	-2.324163	-1.819589	-0.871523	-1.2536	1.744598	-2.9605	0.6072422
15000	1	-2.678619	-1.950301	-2.478105	-2.513194	0.8058938	-2.215608	-1.962418	-0.8744588	-1.166475	1.565091	-2.678619	0.8058938
15500	1	-2.535906	-1.890549	-2.65329	-2.537998	0.6311165	-2.247498	-1.830442	-0.796122	-1.022176	1.535051	-2.535906	0.6311165
16000	1	-2.961333	-1.932938	-2.712268	-2.547899	0.5866023	-2.289872	-1.778475	-0.8085147	-1.100964	1.614135	-2.961333	0.5866023
16500	1	-2.958394	-2.127751	-2.402815	-2.595323	0.4909903	-2.417002	-1.928325	-0.9539746	-1.275942	1.850337	-2.958394	0.4909903
17000	1	-3.09243	-1.968215	-2.443991	-2.407703	0.5231749	-2.281189	-1.789262	-0.8451614	-1.215226	1.726831	-3.09243	0.5231749
17500	1	-2.75329	-1.531499	-2.530478	-2.341636	0.6895909	-2.215059	-2.146332	-0.7557488	-1.010609	1.402788	-2.75329	0.6895909
18000	1	-3.402656	-2.078718	-2.539757	-2.614416	0.6817287	-2.438841	-1.986508	-0.9468783	-1.365652	1.675173	-3.402656	0.6817287
18500	1	-3.393867	-1.221864	-2.691824	-2.575292	0.4406357	-2.296245	-1.908526	-0.5035939	-0.7273784	1.355892	-3.393867	0.4406357
19000	1	-2.531379	-1.978688	-2.412589	-2.778626	0.2748352	-2.430734	-1.967192	-0.7359529	-0.9566816	1.809244	-2.531379	0.2748352
19500	1	-2.815547	-1.491474	-2.37679	-2.528641	0.3270898	-2.270595	-2.095482	-0.583079	-0.8980508	1.666024	-2.815547	0.3270898
20000	1	-2.995827	-1.905398	-2.158042	-2.371867	0.3977723	-2.542716	-2.09308	-0.907989	-1.27354	1.855923	-2.995827	0.3977723
20500	1	-3.366304	-1.695495	-2.21708	-2.431125	0.4114037	-2.395133	-2.027499	-0.8507469	-1.225425	1.695303	-3.366304	0.4114037
21000	1	-3.357767	-1.47363	-2.692751	-2.58162	0.3700776	-2.216673	-1.947586	-0.4847792	-0.9092684	1.479643	-3.357767	0.3700776
21500	1	-3.074538	-1.342142	-2.317786	-2.408046	0.2702895	-2.18738	-2.088692	-0.607366	-0.9453943	1.550602	-3.074538	0.2702895
22000	1	-3.7203	-1.548315	-2.557813	-2.491196	0.4081783	-2.320439	-1.950579	-0.6586103	-1.136769	1.475758	-3.7203	0.4081783
22500	1	-3.774639	-1.532468	-2.55206	-2.496911	0.4015947	-2.306803	-1.915027	-0.5803301	-1.083496	1.469286	-3.774639	0.4015947
23000	1	-3.682667	-1.443825	-2.591673	-2.583209	0.3096623	-2.143118	-1.911775	-0.5385985	-1.014602	1.476946	-3.682667	0.3096623
23500	1	-3.743093	-1.836641	-2.441729	-2.48149	0.3561139	-2.38691	-1.891939	-0.830319	-1.360104	1.646265	-3.743093	0.3561139
24000	1	-3.847791	-1.86405	-2.355873	-2.476109	0.3489648	-2.417916	-1.853496	-0.8993574	-1.415185	1.668424	-3.847791	0.3489648
24500	1	-4.041669	-1.891404	-2.544524	-2.656187	0.4764423	-2.437093	-1.885122	-0.8935361	-1.427823	1.546093	-4.041669	0.4764423

Appendix F: Sample of formatted log data

```
Level Progression: 0
Level: 0
Level Began at: 391
Level Ended at: 93833
Number of Tries: 5
Time Taken: 93442
Level Completed: true

Level Progression: 0
Level: 1
Level Began at: 93846
Level Ended at: 109698
Number of Tries: 1
Time Taken: 15852
Level Completed: true

Level Progression: 0
Level: 2
Level Began at: 109703
Level Ended at: 138201
Number of Tries: 3
Time Taken: 28498
Level Completed: true

Level Progression: 0
Level: 3
Level Began at: 138206
Level Ended at: 155144
Number of Tries: 3
Time Taken: 16938
Level Completed: true

Level Progression: 0
Level: 4
Level Began at: 155148
Level Ended at: 569666
Number of Tries: 47
Time Taken: 414518
Level Completed: true

Level Progression: 0
Level: 5
Level Began at: 569673
Level Ended at: 607817
Number of Tries: 5
Time Taken: 38144
Level Completed: true

Level Progression: 0
Level: 6
Level Began at: 607823
Level Ended at: 688921
Number of Tries: 14
Time Taken: 81098
Level Completed: true
```

```
Level Progression: 0
Level: 7
Level Began at: 688928
Level Ended at: 900392
Number of Tries: 21
Time Taken: 211464
Level Completed: true
```

Appendix G: *Fear dearg* brainstorming notes

March 21, 2016

Fear Dearg

- Pros:
  - Distinct goal
  - Creative restraint / guideline on project
  - Fits theme
- Cons:
  - Arbitrary ideas

Attributes of creature

- Dark, hairy skin
- Long snouts
- Skinny tails
- Rather fat
- Extremely torn and shabby clothing
- Consumes carrion as main staple
- Known for cruel and gruesome jokes
- Only played on those who irritate them
- Be very polite to avoid this
- Active in winter
- Found in polluted coastlines, swamps, & coastal ruins
- Rumored to be former human who wandered into fairy land
- Tries to prevent others from making this mistake

Two worlds

- One normal version of park
- One faerie version/past version

Entrances

- Through a painting (Charlie Bone style)
- Under the bridge on a swan boat
- Going through wilderness - tangled path, may randomly select a path to be rendered

Appendix H: PQP meeting notes

March 25, 2016

Attendees: August Beers, Connor Mattson, Connor Thornberg, Stan Weng, Nick Chaput, and Brian Moriarty

Major Points

- 2 Parks Idea (thumbs up)
- Picture Taking Mechanic (thumbs up)
- Guided Audio Tour (thumbs up)
    - Guide Could Come back later in the game for dramatic effect
    - The weirdness of the game is all part of the tours design (possibility)
- Geocaching (Possibility)
- Painters in place of pictures as a mix up (thumbs up)
- Seeing Between Realities Should not be possible unless via some sort of medium
- There could be all sorts of modular obstacles (thumbs up)
- Use the players cell phone to add to immersion (thumbs up)
- Play with the fact that the player cannot see his avatar (possibility)
    - People could treat the player differently in different worlds

Research Homework

- Kentucky Route Zero - Moriarty
- Firewatch
- Twin Peaks - Moriarty
- Black Mirror
- Blue Velvet - Moriarty
- Eraserhead - Moriarty
- Shutter Island
- The Witness - Moriarty
- Myst
- The Twilight Zone / Night Gallery - Moriarty
- The Coherence

Art Scope

- Potential for Modular art assets
- Stylized representation of characters in the park
    - Silhouette People
    - Footprint People

Tech Scope

- Potential for Custom Shader
- Watson Idea
- Analytics

- Designing Tools
- General AI

Appendix I: Two-worlds game concept

March 25, 2016

Definite:

- Two versions of the park, slightly different from one another
- Travel between them seamlessly, by navigating through the park in certain ways
    - It will not be extremely obvious to the player, but rather subtle and force them to think and constantly be looking at the environment to figure out where they are
- Really encourage gameplay through exploration and discovery

Concept:

- Solve puzzles around the park to get an environment into a certain game state
- Once puzzles are solved, the player must go a certain spot nearby and take a picture of the completed scene, from a certain angle
    - Similar to a perspective puzzle
    - Your camera will begin to glow when you are in the right position
- Way to find the puzzles:
    - There is a codex/journal depicting a hand-drawn version of certain scenes in the park
    - You must find these locales, solve the puzzle to get them into the correct game state, and then take the picture from the perspective depicted in the journal
    - Can find more pages of journal along the way, have them function as clues
    - To be clear, the pages would depict the final game state of the environment, after the puzzle has been completed

Starting off game:
You play as a photographer
First thing you do is a guided tour of the park
This will introduce the picture mechanic
Then, once you come back, you are able to move around on your own and explore
Find your way into the other world (not sure how yet)
Find codex
Real gameplay begins

Appendix J: Personal meeting notes

April 1, 2016

Mary Poppins/Blue's Clues/Charlie Bone
Painting hopping, after you solve a puzzle

Analytic - tech team considering as major component:

- Iterative testing vs dynamic
    - Iterative: perform lots of testing and use results to improve game with intelligent design
    - Dynamic:
- Trying to get into eye-tracking lab
- Trying to design puzzles that lend themselves to being measured
- Doing surveys
- If pursuing this, try to get a psychology professor on board?
- Or learning sciences professor
- Analytics will heavily affect development schedule
-

Story - need to figure out reason why you are in the park

- Why are you taking pictures?
- How does this relate to the 2nd world? (if we keep this idea)
- Can research the Boston park monuments
    - Military (civil war) heroes
    - make way for ducklings thing
    - ether monument
    - Swan boats, main bridge

Experiencing ether dream of first ether patient

- Paintings show old version of the park - no monuments
- Can travel between them
- Goal is to get back to old park
- His thoughts bleeding over as sounds clips?
- Rube Goldberg game, have to solve other things around park to solve final puzzle
    - Like Myst 3, Hitchhiker's guide to the galaxy

Should definitely see swan boats, later in april
Suspending idea of different locales/seasons
Perhaps a day/night cycle

Appendix K: Boston Public Garden visit notes

April 22, 2016

Environments that afford puzzles

- Flower beds
- Clover-shaped pools with statues
    - Make them fill up with water
    - Block off certain parts of clover or s/thing
- Drooping sphere tree (willow)
- Swan boats
- Underneath bridge
    - Perspective puzzles, passing through for something
- Electric boxes around park
- Birch tree eyes
    - Click to open/close
    - Eyes are in sets, all of a certain type will toggle when clicked
    - Eyes may be in multiple sets
- Tree Placards
    - Can use these to deliver information the players
    - (usually say species of tree)
- Spiral Hedges
    - By the Washington Statue

Beacon is very cool

- Maybe it lights up when someone enters or leaves park on a boat

General Info: http://friendsofthepublicgarden.org/our-parks/public-garden/

People involved with the ether monument:

- Thomas Lee - gifted the statue
- Henry Van Brunt - designer
- John Quincy Adams Ward - figures and bas relief sculptor
- “a famous fellow with the chisel” - other sculptor, probably William Robert Ware

Many different kinds of trees
Should decide on <10 to model for game

Appendix L: Major decision points

September 4, 2016

Identified Possible Decisions:

~An agreed upon outline of the topography of the game world (the boston garden)

~Finalize setting
    @Day night cycle or no?
    @Temporal Setting? Are we going with Timeless?

~Are we using trello?

~Narrative experience goal(s)

~Narrative outline.
    @List of essential events

~Puzzle mechanics must be revisited.
    @Select three puzzles we like for further development

~Vision for the outside of the park
    @Should it be an blurred abyss?
    @Should it be a line of buildings?

~Characters walking around?
    @People?
    @Squirrels?
    @Geese?
    @Song Birds?

~Are the players going to explicitly know there are puzzles?
    @An easy solution would be just letting the player know.

~What do we want to use or study to optimize in the game.
    @Learning curve for the puzzles?
    @Exploring an adaptive difficulty algorithm?
    @What elements of the game cause emotional reaction?

Appendix M: Puzzle mechanic "tier list"

September 4, 2106

S Tier:
- Rube Goldberg Machine
- Scavenging and building
- Picture taking/matching
- Swan boats
- Two-park travelling

A Tier:
- Sound based
- Maze
- Perspective puzzles

B Tier:
- Squirrel puzzle
  - Hide the nuts
- Make way for ducklings puzzle
- People parade

F Tier:
- See previous versions of yourself
- Walking sim
- Vast procedurally generated subterranean mining pit

Narrative ideas:
- Squirrels
- Ether dream
- Civil war monuments
- Faerie world (fear deargh)
- Industrial park
- Shadow/no face people

I. Rube Goldberg machine

A. Small-scale
   1. Incomplete but clearly identifiable machine with pieces missing
   2. Search for pieces in the environment or rearrange already present pieces
   3. Constructing the machine may or may not require effort from the player (if not, they would just attach it to the machine overall and the game would automatically place it wherever it belongs)

B. Medium scale

1. Should still be easily identified, though might require the player to step back to see the whole thing
2. Pieces rewarded to the player by solving smaller puzzles in the area, or already existing pieces manipulated by solving smaller puzzles with particular solutions

C. Large scale

1. Not easily seen, probably requires some form of artwork or dialogue indicating its presence
2. Pieces filled in by completing medium/area-sized puzzles (alternatively a "major" puzzle, if areas have more abstract puzzle sequences
3. Perhaps requires a small but difficult puzzle to trigger the device
4. Reward to player should be significant progress (ideally triggering or leading to the end of the game)

II. Perspective Puzzles

A. Object-formation

1. Standing in the correct spot allows you to see and interact with an object comprised of things visible in both foreground and background
    1. Easiest idea would be a key?
    2. How would we get the player to know they can do this?

B. Various things that would probably be considered stealing from The Witness….are game mechanics copyrighted?

C. Must find a way to keep something in view while executing some other task (original Fear Dearg idea)

III. Find the Difference

A. Take a photo, hold it up, and spot the differences

B. Two painters paint the same thing (possibly in different styles?)

1. Find the differences between them
2. Find the thing in the environment that they didn't draw/isn't correct
3. **Somehow choose which painting the environment will match and alternate to solve a puzzle**
4. Mix and match? Maybe they're layered over each other and you can somehow choose which sections of the bottom painting are uncovered
5. Take a picture to observe the environment's state before changing it, then use the picture to temporarily revert that section of the environment

Appendix N: Notes on *Ether Day* (Fenster, 2001)

September 23, 2016

Preface:

- Surgery had always been a bloody, grizzle, gruesome mess
    - Patients were conscious during operations, and sedatives/painkillers were ineffective
        - Even strong analgesics like opium were ineffective, and could prove lethal if misdosed
    - Surgery took place in an amphitheatre, with medical students lining the seats
        - Operations only happened a few times a year
    - Many people were more afraid of surgery than death by infection
    - Surgeons wore the same unwashed coat from operation to operation
        - They treated the bloody coat like a badge of honor
        - Tools were also uncleaned - but the handles were polished!

Important Figures:

- William T. G. Morton
    - The charming anti hero protagonist of our tale. The primary proponent of ether use, inventor of its mask administration method, and ambitious businessman/swindler. A dentist by trade.
    - He could not write legibly or in grammatically correct sentences.
    - He burned his way through Rochester, Worcester, Cincinatti, St. Louis, and New Orleans until he was wanted in all of them
- John Collins Warren
    - House surgeon at Mass General Hospital, son of the founder of Harvard medical, and universally respected doctor.
- Joseph M. Wightman
    - Assisted Morton with the creation of the ether delivery apparatus.
- Horace Wells
    - A dentist based out of Hartford. Morton was his apprentice, and he was the first to give Morton the idea of using anesthetics after he attended one of Colton's shows.
- Augustus Addison Gould
    - Famous, well respected physician-scientist. Hosted Morton in his home for several weeks. Rose to wealth of his own accord.
- Gilbert Abbott
    - First recognized ether patient, house painter by trade.

- Gardner Colton
    - A broke medical student who put on comedy-science shows with Nitrous Oxide

Ether and Nitrous Oxide: Pre-Anesthetic

- **Nitrous oxide** was created by a chemist named Joseph Priestly in 1772.
- Nitrous oxide was used recreationally
    - Robert Southey and Samuel Taylor Coleridge have poems about it
        - *"I felt like the sound of a harp"*
    - Samuel Colt sold it as a traveling "salesman" (drug dealer)
        - He learned of the recreational properties of the gas after working in his father's factory, where many of the workers would take hits of it
        - He also put on shows, where people would use the gas and then go on stage
            - Colt used this operation to raise money for his revolver design patent
        - "To him, like most of science in the 1830s, nitrous oxide was nothing more than a trained bear, to be taken around and shown for money. But, like the bear, it was not understood."
        - Colt was on a boat travelling the Mississippi River, when a passeger "in the throes of a cholera panic" beseeched Colt for his help (he called himself Dr. Coult). Colt gave him, and some other panicked passengers, some NOx and they miraculously recovered
            - Colt concluded that none of them had had cholera in the first place
- Thomas Beddoes was the first person to attempt to use gases in Western Medicine
    - Beddoes went on to found the first private laboratory-clinic, funded by rich humanitarians
    - While looking for associates for his clinic, he was recommended a chemist named Humphry Davey
        - Beddoes ended up making him the Superintendent of the "Pneumatic Institution," despite being only 21 and completely broke
        - In 1800, Davy theorized about use of Nitrous Oxide in surgical procedures
            - He was a regular user of the gas himself, and dedicated a great deal of the Institution's early research towards the gas
- Nitrous oxide was often used recreationally by medical students, along with **sulfuric ether**
    - "'College boys and factory girls had inhaled ether with the utmost freedom, without any ill effects upon their health,' admitted Dr. Charles Jackson, recalling the 1820s and 1830s."
    - Every teenager knew nitrous oxide and sulfuric ether produced similar effects
- Sulfuric ether is a clear liquid, which had been around since the mid-14th century.
    - It first found medical use in the 1760s, as an all-purpose remedy
    - Users can inhale the vapors or drink small amount of the substance

    - Acts as a stimulant and antispasmodic
    - Also acts as a direct sedative on the spinal system
- Gardner Colton, a broke medical student, began to give chemistry lectures about nitrous oxide, complete with the antics of Dr. Coult's shows
    - Colton became very popular, and even went on to have some shows on Broadway

Pieces Coming Together

- Horace Wells was a well known Hartford dentist, who had several n o t a b l e apprentices
    - He was a very ambitious man, endeavoring to work in many fields throughout his life
    - However, he was only ever concerned with doing good in the world
        - He was born wealthy and was rather religious
    - He took T.G. Morton on as an apprentice and later as a business partner, to expand their dentistry reach to Boston as well
    - Wells attended one of Colton's nitrous oxide theatrical shows
        - He took some of the gas, and ended up embarrassing himself
        - After the show, he spoke to another patron, who had injured himself while intoxicated, and had reported feeling no pain at all
        - This gave Wells the idea of using it surgically for "removing a tooth or amputating a limb"
    - The next day, Wells contacted on of his previous apprentices, John Riggs, and attempted to use nitrous oxide surgically himself
        - Wells administered the gas to himself and then had Riggs remove a tooth
        - It was a major success. Wells and some other colleagues began to use it in their practices immediately
    - Wells was permitted to use nitrous oxide in an operation in Boston, pulling a tooth
        - The operation was overseen by John C. Warren and a host of medical students
        - The patient groaned during the operation, and everyone wrote it off as "humbug," despite the patient's protests that he felt no pain at all
    - Wells left Boston dejected, feeling the sting of ridicule from the medical practitioners of Boston
        - Morton, however, believed Wells and committed the notion of anesthetics to his mind
- Morton visited Wells two times in the coming years, interested in his success with nitrous oxide (although not settling his debts with Wells)
- Eventually, Morton began to experiment with anesthetics of his own, taking the idea of sulfuric ether from the ether frolics popular with young Bostonians

    - He tried letting his dog inhale the vapors to study the effect. The dog became very drowsy and fell asleep in the first experiment, and refused to participate in any further experiments
- Morton visited Dr. Charles Jackson, and obtained an apparatus with which to administer ether (accounts of this encounter are conflicting)
    - He tested the apparatus on himself, and then tried to find a patient
    - A man came in requesting a tooth pull, and distraction by mesmerism (hypnotism), but Morton convinced him to try ether instead
        - Morton's account doesn't mention that this patient was a close friend of his, named Ebenfrost, and that there was a journalist named Albert Tenney present, ready to publish an article about this "mysterious chemical"
- Henry Bigelow was the key to Morton's success
    - He served as an intermediary between Morton and Mass General Hospital
    - He brought medical staff one by one to witness Morton's dental etherizations and observe its effect upon the patients.
- Morton also had a patent attorney, who looked into the feasibility of patenting ether
    - No other doctors had patented major medical discoveries
    - Morton added oil of orange to make it a compound and to disguise the unmistakeable ether smell
    - He also created a proprietary delivery apparatus, so that it would be a *marketable* ***invention***, rather than a discovery
- Morton was invited to perform an ether anesthetization at Mass General Hospital, after enough staff members has witnessed it in a dental setting
    - John C. Warren and most of the medical school were in attendance. The students were shocked to learn that anesthetization was about to be attempted
    - Morton showed up 20 minutes late, with his brand new, untested apparatus in hand.
    - Warren was about to call his whole thing off and proceed with surgery, but he let Morton use it
    - It was a huge success and the patient didn't wake up until long after the operation

Next What?

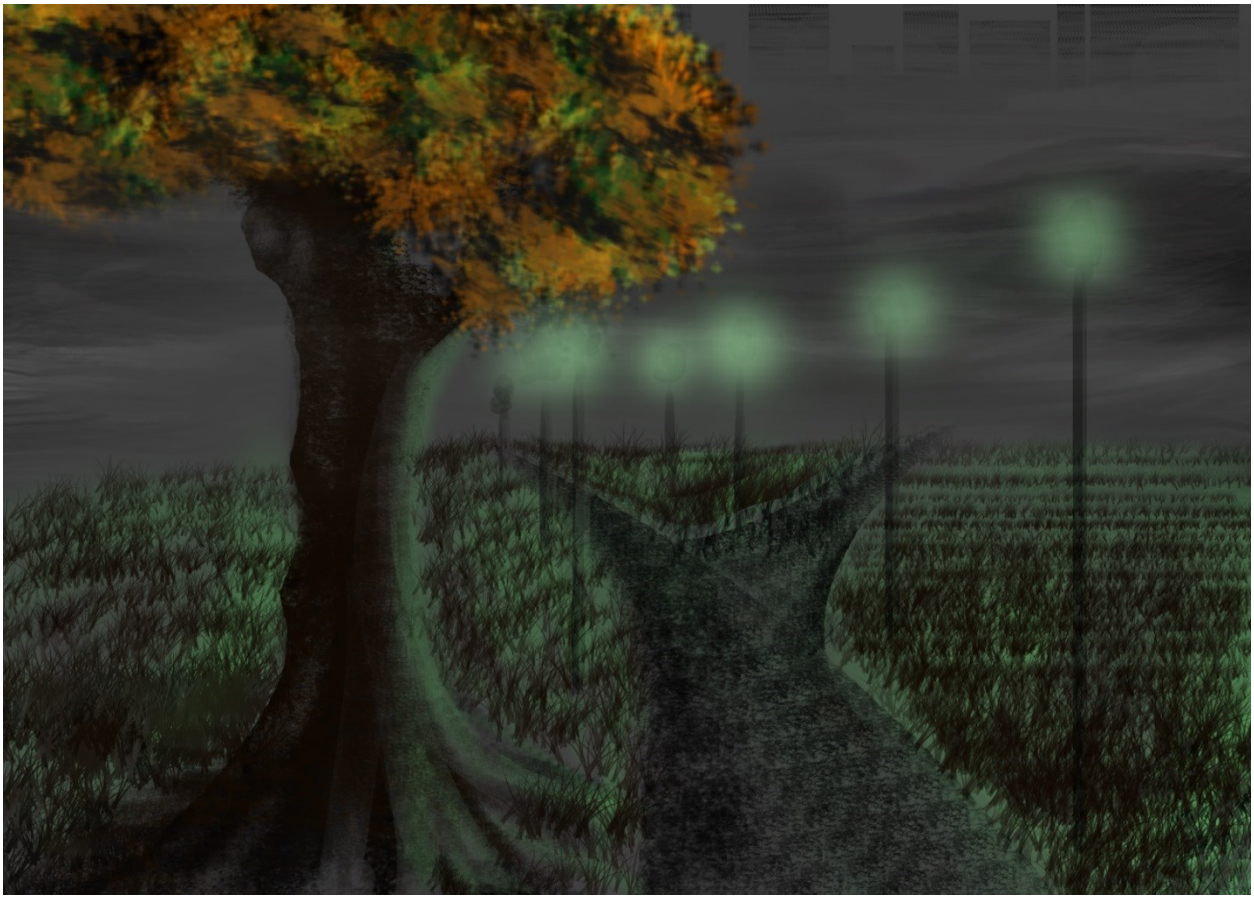
- Morton was invited back to Mass General Hospital the day after his first successful etherization
    - The etherized a woman to remove a tumor.
    - When she awoke, she wove off any questions the doctors had about the pain.
    - She only wanted to talk about the dream she'd just had "about a child she'd left at home"
        - Unfortunately, the book doesn't elaborate any further than this

Appendix O: Game design document

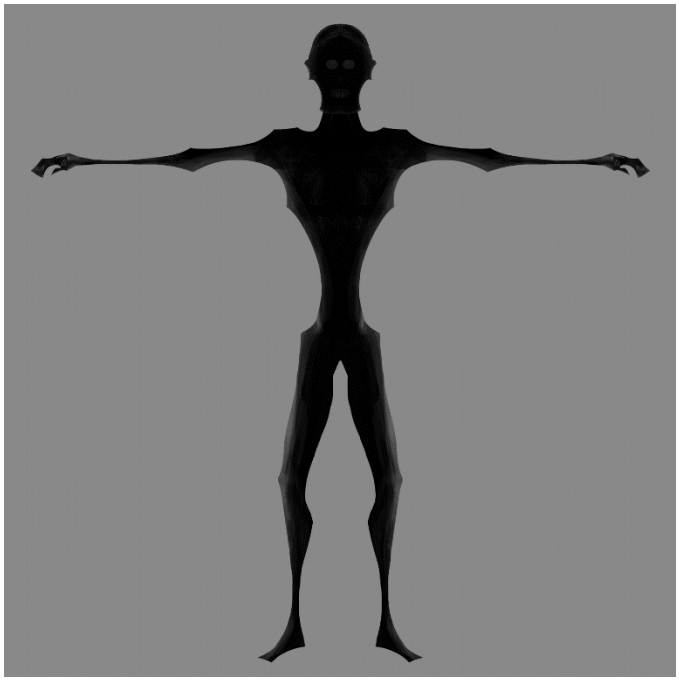
October 2, 2016
Title TBD: The Ethereal

Tagline: “Ether”

Team name: Ether-MQP
Members: August Beers, Connor Mattson, Connor Thornberg, Stan Weng, and Nick Chaput



Hero Art:



Executive Summary: Solve puzzles in the Boston Public Garden to unlock the secrets of its ethereal past.

Vision Statement

Experience goals:

Epiphany moment
Meta-Meta Ending
Learn the story of the first ether operation

Premise:

Core mechanic:

Life of a neuron inspired from neurogenesis of stem cells.
Manifestations:

- As a chess/go board
  - George's Game short
    - Game against yourself?
  - Strong grid based representation
- Neuron/Root planting
  - Flowerbeds
  - Mushroom clusters
- Canvas painting
  - Could also work well with grid
- Abstract representation
  - Geometric shapes with ghostly / spirit textures
- Water spouts
  - Something fountain related

The Backstory:

- Before the use of anesthetics in Western Medicine, surgery was a bloody, painful, gruesome mess.
- They were only performed a few times a year, and were spectated by many medical students
- In the mid 1800s, diethyl ether was introduced as a solution to pain in surgery
- On October 16, 1846 Gilbert Abbot is operated on successfully by John Collins Warren in the Ether dome at Boston general Hospital. Famous line: "Gentlemen this is no humbug."
- Its use was heavily debated at first, but it was eventually proven to be safe and useful
- To commemorate this colossal moment in medical history, there was a statue placed in the Boston Public Garden called the "Ether Monument"
- These facts are the real world basis of our game.

The setting:

- Location = The Boston Public Garden as a timeless space for ether patients to inhabit while anesthetized.
- The sun hangs low in sky and orbits like we are on the north pole
- Strange mist covers the ground like a sea in all directions.
- The player has unknowingly wandered into the garden of his own mind, on ether.

Characters:

- Patient: Gilbert Abbott
- Doctor: John Collins Warren
- Dentist: William TG Morton

The rules of interaction:

- Climb a tree to initiate the puzzle.
- Hud inventory or no?
    - Players should be able to see resources while solving the puzzle.
- Place one piece at a time
    - Throwing seeds
    - Pick up seed from environment
    - Throw seeds onto the board
- Where do the pieces come from?
- Are the pieces contained to one puzzle?
- Physical dimensions of the puzzle?

- We need a Spacebar substitute

Communication of the rules:

The first few puzzles will be designed to introduce certain core mechanics to the players. The puzzles themselves will have some sort of visual signifier to differentiate the puzzle pieces from other parts of the environment. Examples of this would be a soft glow around puzzle objects, or a post-processing volume that makes the player's view progressively more golden as they approach it.

Unknowns:

Narrative Overview:

Level design:

Cinematics:

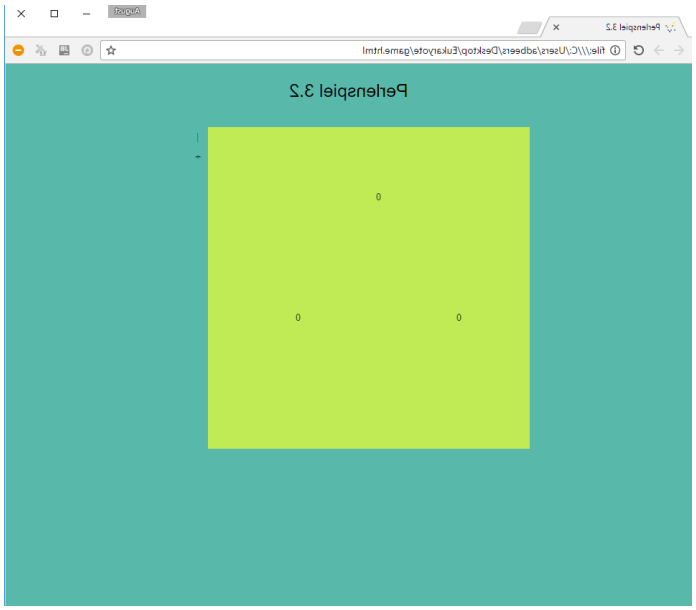
The Aesthetic:

- Mystical
- Ethereal
    - Hazy
    - Dusky
- Mysterious
- Eerie
    - Strange / creepy
    - Not scary
- Tranquil
- Living, fertile environment
    - Moving trees
    - Could encounter anything at any time
- Awe inspiring
    - Moments of environmental reveal
- Distinctly stylized

User Study and Facial Recognition

Analytics:

- To optimize the user experience of playing or game we ran an online testing using Perlenspiel



IRB description:

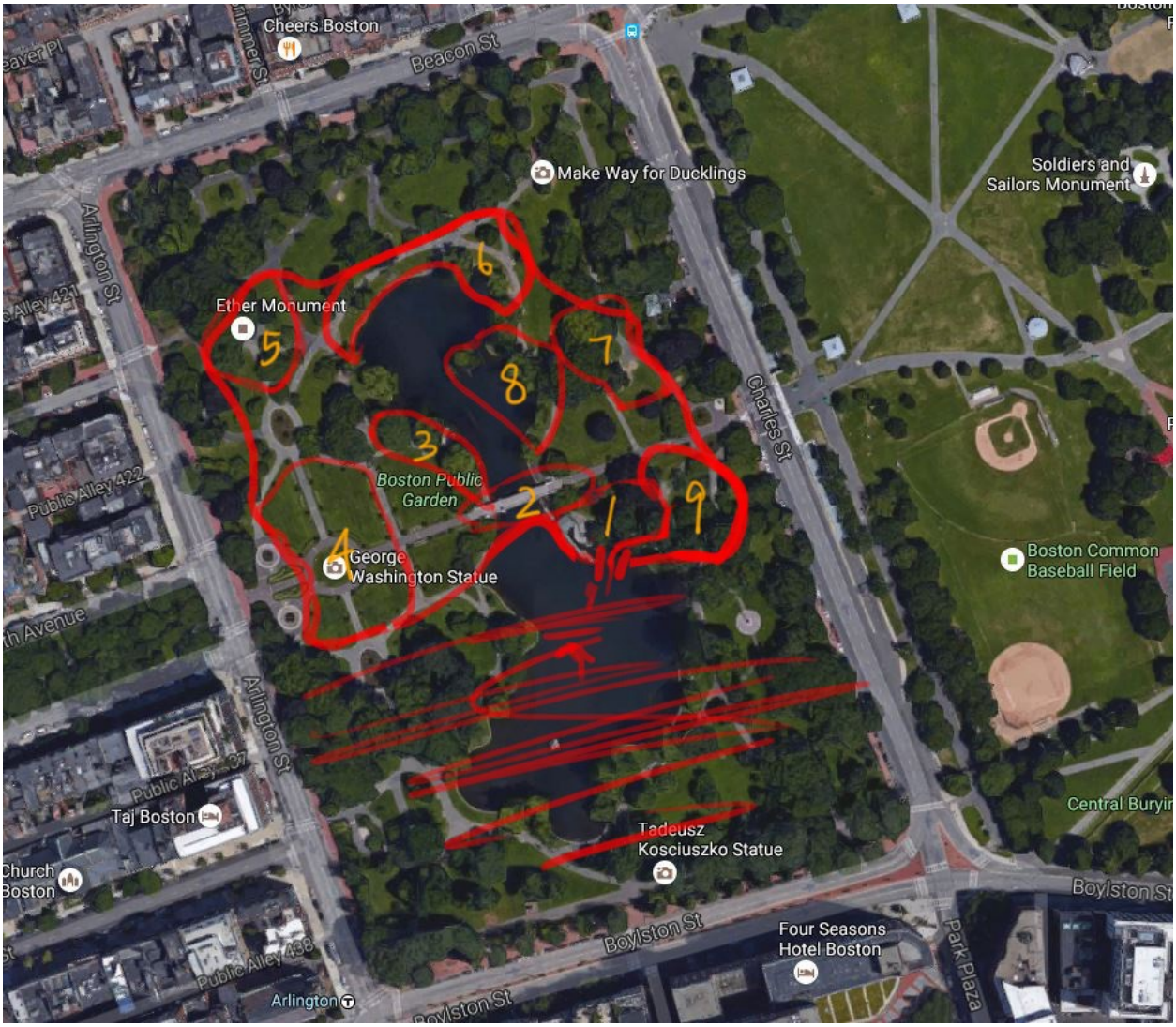
During our study test subjects will be asked to play Eukaryote, a 2D abstract puzzle game played in a web browser. Players are provided various game pieces which the can place on the game board. When the player is done placing these pieces, they can press the spacebar to cause each piece to grow into an abstract pattern over time. The player's objective is to use a combination of these patterns to connect a group of goal nodes on the board. Successfully completing this the game will take the tester to the next puzzle. Failure will require the player to reset the board, rearrange their pieces, and try again. Overall test subject will likely spend less than 20 minutes playing the game.

Everything in the following paragraph will be made known to the player before gameplay begins. Test subjects will also be asked to record their face via webcam while playing the game. Upon consent, the test subject's device's camera will be activated during gameplay. From the camera, 4 images per second will be saved to a WPI server. These images will only be accessible by our MQP team and its advisors. These images will not be shared outside of our MQP team and advisers. We will run facial analysis software to automatically make judgements about the players' experience while playing the game, which we will use to fine-tune our puzzle designs. These improved puzzle designs, and ONLY these improved puzzle designs, will be used in the making of another, larger-scale game. All other information or data from this study will NOT be included in that game.

Appendix P: Puddle definition guide

October 2, 2016

Overview of Puddles:



Puddle 1: Swan Boat Dock

Boundaries of area:

Assets Required:

- Unique Assets
    - Swan Boat
        - Boat hull
        - Bench
        - Swan itself
    - Dock
        - Pier
        - Building itself
        - Metal Railings
        - Signs
            - Swan boat sign
            - Entrance/Exit

            - Tickets
            - Keep off grass
        - Kiosks (inside building)
        - Wooden Chest
    - Stone Circle
        - Have shallow cylinder with color and normal maps of details
- Reused Assets
    - Trash Cans
    - Single Chain Fence
    - Wooden Benches
    - Stone Benches

First person views of area:


Boston Public Garden
Street View - Aug 2015
EXIT
Swan Boats
Boylston Street
Expand
Google


Boston Public Garden
Street View - Aug 2015
ENTRANCE
ENTRANCE
Swan Boats
Expand
Boylston Street
Google
Image capture: Aug 2015 © 2016 Google Terms Report a problem


Boston Public Garden
Street View - Aug 2015
ENTRANCE
Swan Boats
Expand
Boylston Street
Google
Image capture: Aug 2015 © 2016 Google Terms Report a problem

Puddle 2: Preston Bridge

Boundaries of Area:

Assets Required:

- Unique Assets
    - Bridge Ground
    - Metal Fence
    - Stone Short Wall
    - Stone Pillars
        - Supporting Bottom Section
        - Top Cable Connector

    - Bulb Lights
    - Cables
    - Staircase
    - Stone Foundation
    - Brick Slope
- Reused Assets
    - Single Chain Fence

First Person Views of Puddle:

Puddle 3: Japanese Lantern

Boundaries of Area:

Assets Required:

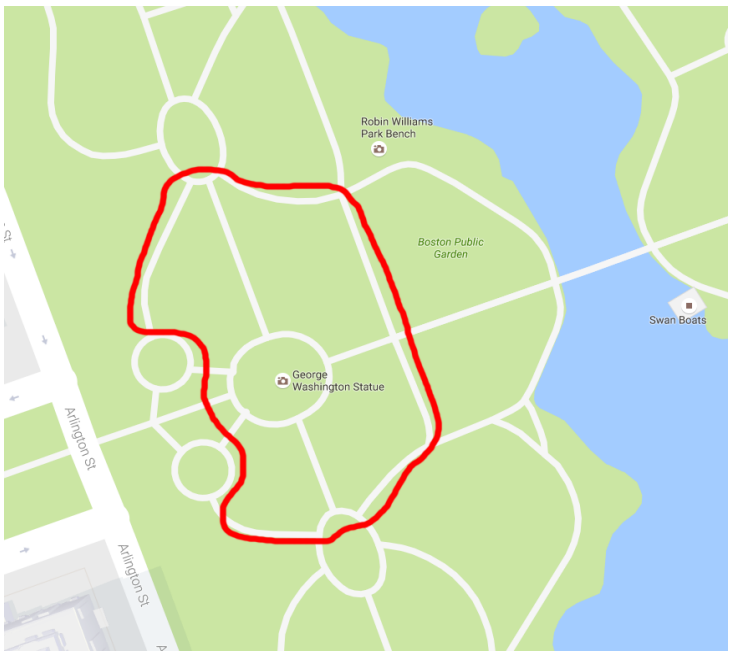
- Unique Assets
    - Stone Pedestal
    - Japanese Metal Lantern
    - Robin William Bench Memorial
- Reused Assets
    - Ground mats
        - Grass
        - Dirt

        - Pavement
    - Wooden Bench (no back)
        - Stone base
    - Overhanging Lamppost
    - Trees tbd

First Person Views:

Puddle 4: George Washington Statue

Boundaries of area:



Assets Required:

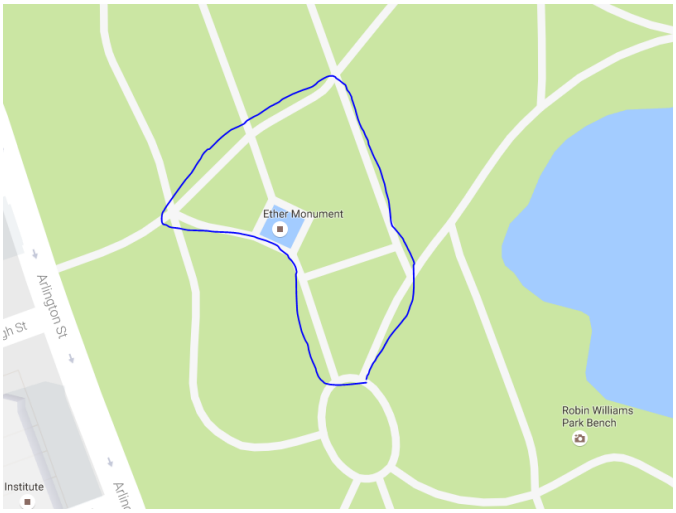
- Unique Assets
    - George Washington Statue
    - Stone Lights for Statue
    - Sculpting Trees
    - 2 Unique Flower Beds
    - Mini Palm Trees
    -
- Reused Assets

- Single Chain Fence
- Lamp Post

First person views of area:

Puddle 5: Ether Monument

Boundaries of area:



Assets Required:

- Unique Assets
    - Ether Monument
- Reused Assets
    - Trash Can
    - Lamp Post
    - Benches

First person views of area:

Puddle 6: Connector

Boundaries of area:

Assets Required:

- Unique Assets
    - Hexagonal Ground Texture
    - Unique Lamp Post
- Reused Assets
    - Benches
    - Trashcan
    - Lamp Post

First person views of area:

Puddle 7: BFB Fountain

Boundaries of area:

Assets Required:

- Unique Assets:
    - BFB Statue
    - Umbrella Tree
- Reused Assets:
    - Cross Fountain Tile
    - Bench
    - Lampost
    - Trash Cans

First person views of the area:

Puddle 8: Island

Boundaries of area:

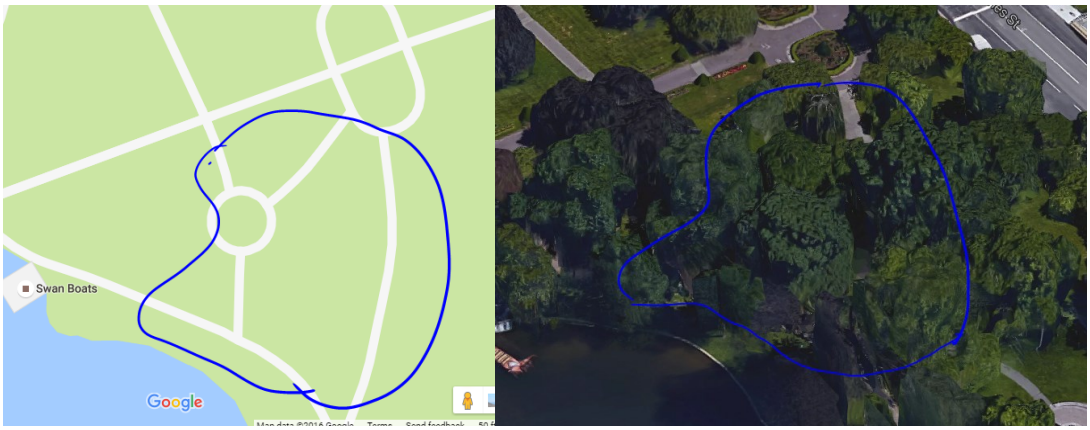
Assets Required:

- Unique Assets:
    - Ghost
    - Island
        - Bridge
    - Umbrella Willow
- Reused Assets:
    - Stone
    - Lantern
    - Trash Cans

First person views of the area:

Puddle 9: Ragheera Fountain

Boundaries of area:



Assets Required:

- Unique Assets:
    - Bagheera Fountain
        - Panther and Bird Statue
- Reused Assets:
    - Bagheera Fountain
        - Cross Fountain Tile
    - Lantern
    - Bench
    - Trash Cans

Appendix Q: IRB protocols

Consent Form

Informed Consent Agreement for Participation in a Research Study

Investigator: Jacob Whitehill

Contact Information: jrwhitehill@wpi.edu

Title of Research Study: Measuring Students' Nonverbal Responses to Puzzle Games

Introduction

You are being asked to participate in a research study. Before you agree, however, you must be fully informed about the purpose of the study, the procedures to be followed, and any benefits, risks or discomfort that you may experience as a result of your participation. This form presents information about the study so that you may make a fully informed decision regarding your participation.

Purpose of the study: The purpose of this study is to investigate the kinds of nonverbal behavior -- such as head gestures (nods, shakes), eye movements, and other facial movements -- that people make when they play computer-based puzzle games. We are interested only in people's nonverbal behavior; we are **not** interested in your identity (such as who you are, where you live, etc.).

In order to participate in this study: (1) You must be at least 18 years of age; and (2) You must have a computer or mobile device with a web camera that is able to capture your head and face while you perform the HIT. Audio will not be recorded, only video.

Procedures to be followed:

The HIT will take about 15 minutes to complete. In particular:

1. You will be asked to solve 8 different puzzles. **Instructions**: In each puzzle, you will be provided with a set of game pieces located in a light-blue column to the right of the green game board. To place the pieces, click and drag them onto the game board. Once the pieces are placed, press the spacebar to trigger a "growth pattern" starting at each placed piece. Press the spacebar again to clear the growth and try again. Each 0 on the board represents a "node" that must be connected together by the growth patterns to complete the level.
2. While you are trying to solve the puzzles, we will record video of your head and face through the web camera in your computer or mobile device. Hence, your web camera should be on and pointed at your head and face while you perform the HIT. We will also record the sequence of moves (game log) that you make within each puzzle.
3. Once you solve all 8 puzzles, you will be allowed to submit the HIT and receive payment. We anticipate that solving all the puzzles will take about 10 minutes. However, after 15 minutes of game play, you will be allowed to submit the HIT even if you were unsuccessful at solving the puzzles.

Once you have completed the HIT, the recorded video and the game logs will be saved on our server (this will result in about 50MB of transmitted data), and you will receive your payment. If, at any point during the HIT, you decide that you do not wish to participate or do not wish for your video to be recorded, then simply return the HIT -- your video will then be deleted automatically.

Please note that, if your head and face are not visible in the recorded video, then your HIT may be rejected.

Risks to study participants:

You might feel some embarrassment that you are being recorded. While we do not collect any

information about your name or location, please note that you may be identifiable from the video based on your face. You are free to leave the HIT at any time, in which case your video will be deleted.

Technical requirements for participation:
Since video will be transmitted in real time from your computer to our server, your Internet connection should support at least 100KB/s upload speed. Please note that a total of about 50MB of data may be transmitted from your computer to our server during the 10-15 minute experiment.

Possibility of software errors:
While we have tested the game software on a variety of web browsers, it is nonetheless conceivable that a technical error could arise for some users that prevents completion of the HIT. Please contact jrwhitehill@wpi.edu if you believe a software error has occurred. Rest assured that, if you make a sincere effort to complete the HIT, then you will be paid in full.

Benefits to research participants and others: You might enjoy playing the puzzle games. In addition, the data collected from this experiment will contribute to a greater understanding of how students' nonverbal behavior arises during learning tasks.

Record keeping and confidentiality: The recorded videos, as well as the game logs, will be accessible only to members of the investigator's research team. These data will be stored on an encrypted disk and will not be given to anyone else. The recorded videos may occasionally be shown at scientific meetings attended by researchers interested in nonverbal behavior and computer games research (but only if you explicitly give permission for your videos to be shown in this way on the video release consent form). Records of your participation in this study will be held confidential so far as permitted by law. However, the study investigators, the sponsor or it's designee and, under certain circumstances, the Worcester Polytechnic Institute Institutional Review Board (WPI IRB) will be able to inspect and have access to confidential data

that identify you by name. Any publication or presentation of the data will not identify you.

Compensation or treatment in the event of injury: Since the study involves only playing puzzle games and being videorecorded while doing so, no injuries are expected. There is no compensation for injuries. You do not give up any of your legal rights by signing this statement.

Payment: Upon completion of the entire task, press "Submit" to submit your work to the Amazon Mechanical Turk. As long as you tried to solve the puzzles (you do not need to have actually solved them), and as long as the video recorded through your web camera clearly contains your head and face during the entire task, then your HIT will be approved and you will be paid $2.00.

For more information about this research or about the rights of research participants, or in case of research-related injury, contact: Contact the principal investigator of the study, Dr. Jacob Whitehill, jrwhitehill@wpi.edu. If your concern has not been adequately addressed by Dr. Whitehill, you may also contact the chair of the Institutional Review Board (IRB) at Worcester Polytechnic Institute (Professor Kent Rissmiller, Tel. 508-831-5019, Email: kjr@wpi.edu), as well as the University Compliance Officer (Jon Bartelson, Tel. 508-831-5725, Email: jonb@wpi.edu).

Your participation in this research is voluntary. Your refusal to participate will not result in any penalty to you or any loss of benefits to which you may otherwise be entitled. You may decide to stop participating in the research at any time without penalty or loss of other benefits (except that you will not be paid for the HIT). The project investigators retain the right to cancel or postpone the experimental procedures at any time they see fit.

By signing below, you acknowledge that you have been informed about and consent to be a

participant in the study described above. Make sure that your questions are answered to your satisfaction before signing. You are entitled to retain a copy of this consent agreement.

To confirm your consent, enter today's date below, and then press <I consent>.

Date: _____

<I consent> <I do not consent>

Appendix R: Video release form for *Eukaryote* study

Videorecording Release Form
In this HIT, you will play a game called Eukaryote, which is a 2D abstract puzzle game. Simultaneously, video of your head and face will be recorded from the web camera in your computer or mobile device. Recording this video allows us to study people's nonverbal behavior while playing puzzle games. In order to participate in this HIT, you should be comfortable with the fact that your video (a maximum of 50MB of data will be transmitted) is being recorded. Audio will not be recorded. Note that we are interested only in nonverbal behavior; we are **not** interested in your identity (such as who you are, where you live, etc.).

Please check the boxes below to indicate which uses of your video you approve:

[ ] The video can be studied by the research team for use in the research project.

[ ] The video can be shown at meetings of scientists interested in nonverbal behavior and computer games.

If you consent to the use of your video as described above and wish to participate in our study, then enter today's date and then click "I consent". If you do not wish to proceed, then click "I do not consent", and the HIT will be returned.
Date:

<I consent> <I do not consent>

Appendix S: Example of formatted consent data

Timestamp: 1486087188744

Worder ID: AHLS6AZ1VTNH7

Assignment ID: 30X31N5D63PCBK5TVRGQVK4P7NWASJ

Consent Given (Consent Form): true

Consent Date (Consent Form): 2017-02-03

Consent Given (Release Form): true

Consent Date (Release Form): 2017-02-03

Consent for OUR USE: true

Consent for FUTURE USE: true

Appendix T: IRB approval for *Eukaryote* study

Worcester Polytechnic Institute IRB# 1

HHS IRB # 00007374

23 January 2017

File:*16-275M*

RE: Modification to IRB *16-275*, "Measuring Students Non Verbal Responses to Puzzles Games"

Dear *Prof. Whitehall*[sic],

The WPI Institutional Review Committee (IRB) approves the modification submitted to application *file 16-275* "Measuring Students Non Verbal Responses to Puzzles Games", *dated 12 January 2017* and approves the modifications in response to the adverse event dated *29 December 2016* to 1). Limit the amount of data captured/transmitted, 2). The updated instructions to subjects and 3). The procedures in the event of a technical problem.

This also resolves the adverse event for this study.

Consistent with CFR 46.116 regarding the general requirements for informed consent, we remind you to only use the **approved consent process for online survey.**

The period covered by this approval is from *23 January 2017* until *30 October 2017* unless terminated sooner (in writing) by yourself or the WPI IRB. Amendments or changes to the research that might alter this specific approval must be submitted to the WPI IRB for review and may require a full IRB application in order for the research to continue.

If the research is to continue past *30 October 2017*, a renewal application must be filed with the IRB in sufficient time for approval before *30 October.*

Please contact the undersigned if you have any questions about the terms of this approval.

Sincerely,

Kent Rissmiller WPI IRB

Chair