



WPI

Gnome Hollow

The Creation of a Large Scale Interactive Garden Maker

Submitted to the faculty of

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science in

Interactive Media and Game Development

and for the

Degree of Bachelor of Arts in

Interactive Media and Game Development

Authors:

Diego Arce
darce@wpi.edu
IMGD BA

Kirsten Roethel
knroethel@wpi.edu
IMGD BA

Alan Roush
aroush@wpi.edu
IMGD BA

Garret Sheehan
gssheehan@wpi.edu
IMGD CS

Faculty Advisors:

Prof. Ralph Sutter
rsutter@wpi.edu
IMGD

Prof. Gillian Smith
gsmith@wpi.edu
IMGD/CS

This report represents work of WPI undergraduate students submitted to the faculty as evidence of a degree requirement. WPI routinely publishes these reports on its web site without editorial or peer review. For more information about the projects program at WPI, see <http://www.wpi.edu/Academics/Projects>.

Abstract

Gnome Hollow is a 3D cooperative gardening game meant to be played on a large-scale projected screen. Two players each take control of different tools and must coordinate with each other to grow and maintain a variety of plants. Players can create their own beautiful gnome-inhabited garden without any worries of winning or losing. The goal of this MQP was to create a cute, silly, and stress-free experience for WPI students to play on campus and to meet new people. We conducted several rounds of feedback to determine what our target audience would want to see from the game, test the player experience, and help put everything together near the end of development. We wanted the game to focus mainly on the visuals and audio in order to create a pleasant sensory experience for players.

Gnome Hollow was created in the Unity engine for Windows and Mac. This report delves into the development and iterations of the project's design, art, mechanics, audio, and playtesting. Throughout the project, all team members demonstrated their skills in their respective areas of expertise and were able to create a relaxing yet engaging game.

Acknowledgements

Our team would like to thank the following for their contribution to the development and completion of this project:

Professor Ralph Sutter and Dr. Gillian Smith for advising our project, giving us great feedback, and supporting us throughout the entire MQP process.

Professor Dean O'Donnell for his contributions to WPI's IMGD program and for the original pitch for this project.

Justin Gitlin and Brian Corrigan of OhHeckYeah and Eric Newkirk of AFVX for agreeing to be interviewed, providing us with essential information about projection technology and large-scale games.

The members of the Fest IQP for organizing playtesting events and coordinating with us to accommodate our unique game requirements.

All of our playtesters who provided us with their time and feedback for helping us make our game the best it can be.

Table of Contents

Abstract	1
Acknowledgements	2
Table of Contents	3
List of Figures	5
List of Tables	9
1. Introduction	10
2.1. Project Concept	12
2.1.1. Choosing a Building	12
2.1.2. Game Iterations	13
2.1.3. Finalized Game Concept	14
2.2. Game Inspirations	15
2.3. Art Inspirations	15
2.4. Audio Inspirations	16
3. Equipment	17
3.1. Projector Research Process	17
3.2. Speakers	17
3.3. Budget and Pitch	18
3.4. Lens and Lights	18
3.5. Final Decision	18
4. Design	19
4.1. Game Perspective	19
4.2. Target Audience	20
4.3. Experience Goals	20
4.4. Mechanics	21
4.4.1. Player Controls/Tools	21
4.4.2. Plants	22
4.4.3. Gnome	23
4.5. Gameplay Loop	23
4.6. User Interface	24
4.7. Level Design	27
4.8. Game Name	29
5. Art Development	30
5.1. Aesthetic	30
5.1.1: Art Bible	30
5.1.2: Color	31
5.2. Asset Pipeline	31
5.2.1. Conceptualization	31
	3

5.2.2. Modeling	32
5.2.3. UV Unwrapping & Texturing	33
5.2.4. Rigging & Animation	34
5.2.5. Particle Effect Painting	34
5.2.6. Exporting and Importing	35
5.3. In-Engine	36
5.3.1. Lighting	36
5.3.2. Post-Processing	36
6. Technical Implementation	37
6.1. Software	37
6.1.1. Unity	37
6.1.2. GitHub	37
6.2. Player Tools and Controls	38
6.2.1. Player Control Implementation	38
6.2.2. Player Tools	40
6.3. User Interface	41
6.4. Plant Implementation	44
6.5. Gnome Implementation	47
6.6. Game Structure	48
6.6.1. Time Limit	49
6.7. Unity-Specific Technical Features	49
7. Audio Development	50
7.1. Challenges	50
7.2. Ambience	50
7.3. Tools	50
7.3.1. Sound Effect Creation	51
7.4. Sound Effects	53
7.5. Implementation	54
7.6. Soundtrack	55
7.6.1. Melody	55
7.6.2. Supporting Instruments	59
7.6.3. Extra Layers	60
8. Playtesting	61
8.1. Testing Methodology	61
8.2. Testing Results	61
8.2.1. Profest	62
8.2.2. AlphaFest	66
8.2.3. C-Term Playtesting	71

8.2.3. D-Term Playtesting	76
8.3 Major Iteration Changes	81
9. Conclusion	82
9.1: Post Mortem	82
9.1.1. Team	82
9.1.2. Art	82
9.1.3. Tech	83
9.1.4. Audio	84
9.2: Future Plans for Gnome Hollow	84
References	85
Appendices	90
Appendix A: Protofest Survey with Responses	90
Appendix B: AlphaFest Survey with Responses	98
Appendix C: C-Term Survey with Responses	110
Appendix D: D-Term Survey with Responses	124
Appendix E: Art Bible	138
Appendix F: Interview Questions	143
Appendix G: Informed Consent Form	144
Appendix H: Study Protocol	146
Appendix I: Survey Draft	147

List of Figures

Figure 1: The logo for Gnome Hollow.....	10
Figure 2: A few top contenders for our projected building from choice. From left to right: Washburn Shops, Salisbury Laboratories, Unity Hall.....	12
Figure 3: A photo of Higgins House at the angle we planned on projecting the game from.....	13
Figure 4: Comparable games for our initial concept.....	14
Figure 5: Visual and gameplay inspiration for our project.....	15
Figure 6: A mockup sketch of what areas could be used for gameplay on Higgins House.....	19
Figure 7: Team-made graphic of the arousal-valence model.....	21
Figure 8: Flow of the game between two players.....	24
Figure 9: UI concept art for a different tool selection styles.....	25
Figure 10: Frame from the intro tutorial cutscene.....	26
Figure 11: Controller button mapping.....	26
Figure 12: In-game tooltips.....	27
Figure 13: Early mockups for shelf level design.....	28
Figure 14: Level design in-game.....	28
Figure 15: Brick wall background.....	29
Figure 16: Art Bible.....	30
Figure 17: Colors used to texture each of Gnome Hollow’s plants.....	31
Figure 18: Reference Images.....	32
Figure 19: Original and Final Seed Bag Model.....	33
Figure 20: Mushroom Texturing.....	33
Figure 21: Wilt Animation.....	34
Figure 22: Particle Effect Alphas.....	35
Figure 23: Post-Processing Comparison.....	36
Figure 24: All six tools the players can use. The top row (from left to right) features the glove, seed bag, and shovel. The bottom row (from left to right) features the fertilizer bag, the watering can, and the shears.....	40
Figure 25: A prototype screenshot of the game’s cut title screen.....	42
Figure 26: Gnome Hollow’s tutorial video playing in-engine.....	43
Figure 27: A contextual prompt on the top of the screen to remind players to keep their plants watered.....	43
Figure 28: A tooltip for the fertilizer bag being prompted on the bottom half of the screen.....	44
Figure 29: All bushes in Gnome Hollow. From left to right: Bush, Berry Bush, Glacier Bush, and Fireberry Bush.....	46
Figure 30: All launching plants in Gnome Hollow. From left to right: Cactus, Mushroom.....	47
Figure 31: A prototype of the scrapped falling bricks. The top and bottom parts of the game would have been framed as stages 1 and 2 of the game respectively, with stage 3 being completely open.....	48

Figure 32: All plant audio scriptable objects. These can very easily be inserted into any plant and then be swapped out, making it trivial for the entire team to swap in sound effects.....	49
Figure 33: The scriptable object for tool sound effects.....	54
Figure 34: Part 1 of initial melody.....	55
Figure 35: Variation of part 1 for initial melody.....	55
Figure 36: Part 2 of initial melody.....	56
Figure 37: Exploring options.....	56
Figure 38: Introduction of soundtrack.....	57
Figure 39: Introduction part 2.....	57
Figure 40: Octave shifted chord progression.....	57
Figure 41: Chord progression from just before the main melody.....	57
Figure 42: Main melody.....	58
Figure 43: Main melody with variation.....	58
Figure 44: Main melody with progressive doubling.....	58
Figure 45: Main doubled melody.....	59
Figure 46: New melody.....	59
Figure 47: New melody with variations.....	59
Figure 48: ProtoFest build.....	62
Figure 49: Results from ProtoFest question “How long would you want to play this prototype?”.....	63
Figure 50: Results from ProtoFest question “How would you describe your experience? Check all that apply”.....	63
Figure 51: Results from ProtoFest interest prompt “An experience where you are scored based on your garden’s appearance”.....	64
Figure 52: Results from ProtoFest interest prompt “A fast paced round where you have a small amount of time to make your garden”.....	64
Figure 53: Results from ProtoFest interest prompt “A slow paced round where you have a large amount of time to make your garden”.....	65
Figure 54: AlphaFest build.....	66
Figure 55: Results from AlphaFest question “How long would you want to play this prototype?”.....	67
Figure 56: Results from AlphaFest question “How would you describe your experience? Check all that apply”.....	68
Figure 57: Results from AlphaFest question “Did you enjoy the multiplayer aspect of the game?”.....	68
Figure 58: Results from AlphaFest question “How would you rate the scale of the objects on the screen?”.....	69
Figure 59: Results from AlphaFest interest prompt “A game that requires cooperation to succeed”.....	70
Figure 60: C-Term build.....	71

Figure 61: Results from C-Term question “How long would you want to play this prototype?”	72
Figure 62: Results from C-Term question “How would you describe your experience? Check all that apply”	72
Figure 63: Results from C-Term question “How effectively did the in-game tutorial options inform you about how to play?”	73
Figure 64: Results from C-Term question “Was the gameplay effectively divided between both the seed planter and the tool user?”	74
Figure 65: Results from C-Term question “How do you feel about the game’s art/aesthetics?”	74
Figure 66: Results from C-Term question “Was there enough variety between the different types of plants?”	75
Figure 67: Results from C-Term question “Do the animations fit their model’s appearance?”	76
Figure 68: D-Term build	77
Figure 69: Results from D-Term question “How did you feel before playing Gnome Hollow? Check those that apply”	78
Figure 70: Results from D-Term question “How would you describe your experience playing Gnome Hollow? Check the 3 that best apply”	78
Figure 71: Results from D-Term question “How did you feel after playing Gnome Hollow? Check those that apply”	79
Figure 72: Results from D-Term question “How effectively did the in-game tutorial options inform you about how to play?”	80
Figure 73: Results from C-Term question “Was the gameplay effectively divided between both the plant and tool players?”	80

List of Tables

Table 1: The types of tools and what players they belong to.....	39
Table 2: All possible states a plant can have.....	45

1. Introduction



Figure 1: The logo for Gnome Hollow.

Gnome Hollow is a two-player cooperative gardening game designed to be projected on the wall of a building where both players cooperate to design a garden in a laid-back environment in a space inhabited by autonomous gnomes. Both players have different sets of tools, requiring them to cooperate to populate the garden with greenery and customize the scenery to their liking. It is intended to serve as a laid-back social game for WPI students to partake in, with hypothetical players playing it in between classes as a breather from their busy school lives. Many sessions typically last around 3-5 minutes, though the experience can go as short or as long as the user wishes due to it not having any strict time limits or rules aside from planting more plants.

As a Major Qualifying Project (MQP), *Gnome Hollow* showcases the skills the project team gained throughout their WPI education. The game has a heavy focus on showcasing 3D art and animations due to the main draw of the game being to build a beautiful garden and half the team being artists, though the talents of our audio developer and our programmer are also shown in wiring the entire game together and creating a rich soundscape.

Originally being proposed as a game designed around a specific building, the project eventually morphed into a more general purpose game that was meant to be projected onto the side of a wall, but not necessarily a specific building. This allowed the projected game to serve as a “window” into the world of the game, while also allowing us to have more freedom over the level design of *Gnome Hollow*.

With the game’s genre being finalized as a cooperative garden builder at the end of pre-production, we continued to garner feedback throughout the project’s early stages to determine the most effective direction for the game to go in. We used this to make sure *Gnome*

Hollow was evoking the intended feelings in the player as well as to finalize design decisions such as splitting tools asynchronously among players.

This paper is split into several sections detailing different aspects about *Gnome Hollow*. Our background details several aspects of pre-production including deciding on our building and concept alongside our inspirations. Equipment details how we went about obtaining the equipment we desired to project the game onto a building. Our design section breaks down *Gnome Hollow*'s design decisions related to mechanics, target audience, and experience goals. Art development outlines our art methodologies and pipelines, while the technical implementation section goes into the nuts and bolts behind implementing said mechanics within *Gnome Hollow*. Similarly, audio development details the production methodology and rationale behind our audio developer's decisions. Testing methodology describes how we went about playtesting for feedback and what conclusions we drew from said feedback. Finally, our conclusion section is a reflection of how well *Gnome Hollow* achieved our goals and how our process could have been improved.

2. Background

Before diving into the development of our project, we had a fairly lengthy pre-production phase to determine what we wanted to make and how we wanted to accomplish our goals.

2.1. Project Concept

Our project concept was proposed at the IMGD Project Presentation Day 2022 by Professor Dean O'Donnell. His original vision was projection-mapping an arcade game onto a campus building, using said building to create a unique experience when playing. Ideas for games initially proposed by Dean involved Brick Breaker and Rampage, where player actions appear to have an effect on the building itself. Over the first few weeks of pre-production in August-September 2022, we had to determine what kind of game we wanted to make from this initial pitch.

2.1.1. Choosing a Building

Choosing our building was accomplished in one of our first meetings, as we were fairly sure it would end up influencing the world design of our game. We toured WPI's campus, examining a variety of buildings that might be suitable for our game idea. Some general favorites among the team and advisors involved some honorable mentions:

- The side of Washburn Shops facing Unity Hall, as the building layout could offer unique geometry for the game to take place on.
- The side of Unity facing Washburn, as it was a flat surface that would be easy to project onto and the de-elevated wall meant that projecting onto it would be a fairly easy task.
- A side of Salisbury Labs with few windows, as the verticality could offer some interesting level design possibilities

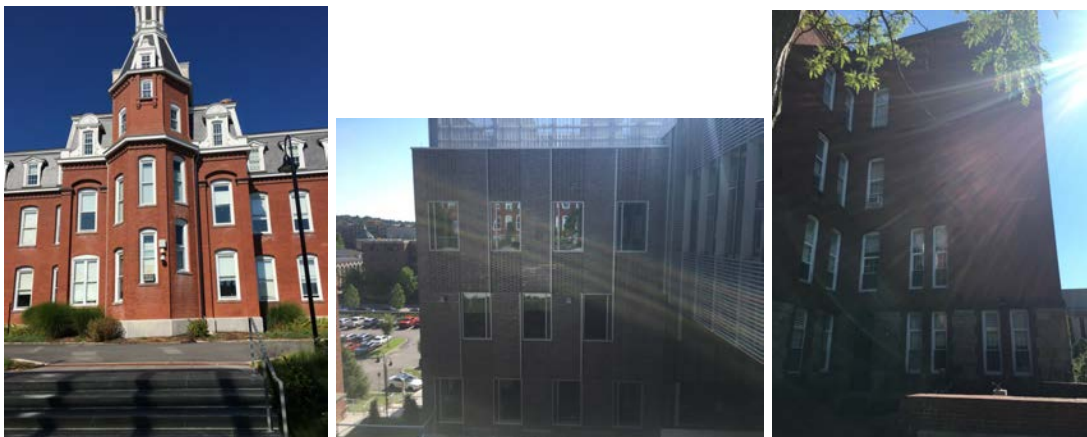


Figure 2: A few top contenders for our projected building from choice. From left to right: Washburn Shops, Salisbury Laboratories, Unity Hall

The clear favorite among the team ended up being a side of Higgins Garden House facing the Rubin Campus Center (CC) for a variety of reasons. It had a great location, as the CC was a student hotspot due to a plethora of food options and activities being conducted there in daily WPI life. The specific location also works out quite nice, as the area directly next to that side of Higgins Garden House is an outdoor lounge area, which is ideal for students to play video games in and would be a great place to organize a gathering of students potentially eager to play our game. This area was also de-elevated, making it a great place to set up a projector due to said projector not having to be tilted up in order to be usable. That area's garden aesthetic was also not unnoticed, as it would allow our game to be themed around gardening, giving a concrete art direction for our game to go in. There were some concerns raised about the glass windows and doors getting in the way of the experience, but it was roughly the same amount of glass as many of our other popular options so that issue wasn't as much of a concern going in. A picture of Higgins can be seen at Figure 3.



Figure 3: A photo of Higgins House at the angle we planned on projecting the game from

2.1.2. Game Iterations

The vision of our game changed considerably during our pre-production stage of development. Though we had decided on a multiplayer game very early on in pre-production, successive meetings with advisors influenced us to take our game in directions we hadn't previously anticipated due to unique factors surrounding us projecting a game onto a building.

We initially planned to create a basic PvP platform fighting game that would have a level of complexity similar to that of Duck Game, Stick Fight: The Game, or Towerfall: very basic mechanics that would still allow for a variety of enjoyable situations. Our most fleshed out idea

involved players having garden bases they'd have to maintain, spreading their garden coverage by moving around and aiming to destroy the bases of other players. Upon talking with our advisors, we realized that making our game a PvP-oriented experience could conflict with making an easily accessible game that anyone passing by would want to play. A player's ego would be on the line in a PvP game, which could become even more of a stressful experience in situations where they might be playing in front of a fairly sizable crowd due to the nature of the game being projected. In addition, our survival-type game concept could create a significant amount of downtime which would be a problem in long venues. Most people we talked to who created projection games of the nature we were going for stated that their game lengths were extremely short, often being under one minute.

As a result, we ended up shifting to a more bite-sized cooperative experience that would address both of these problems and better our experience goals.

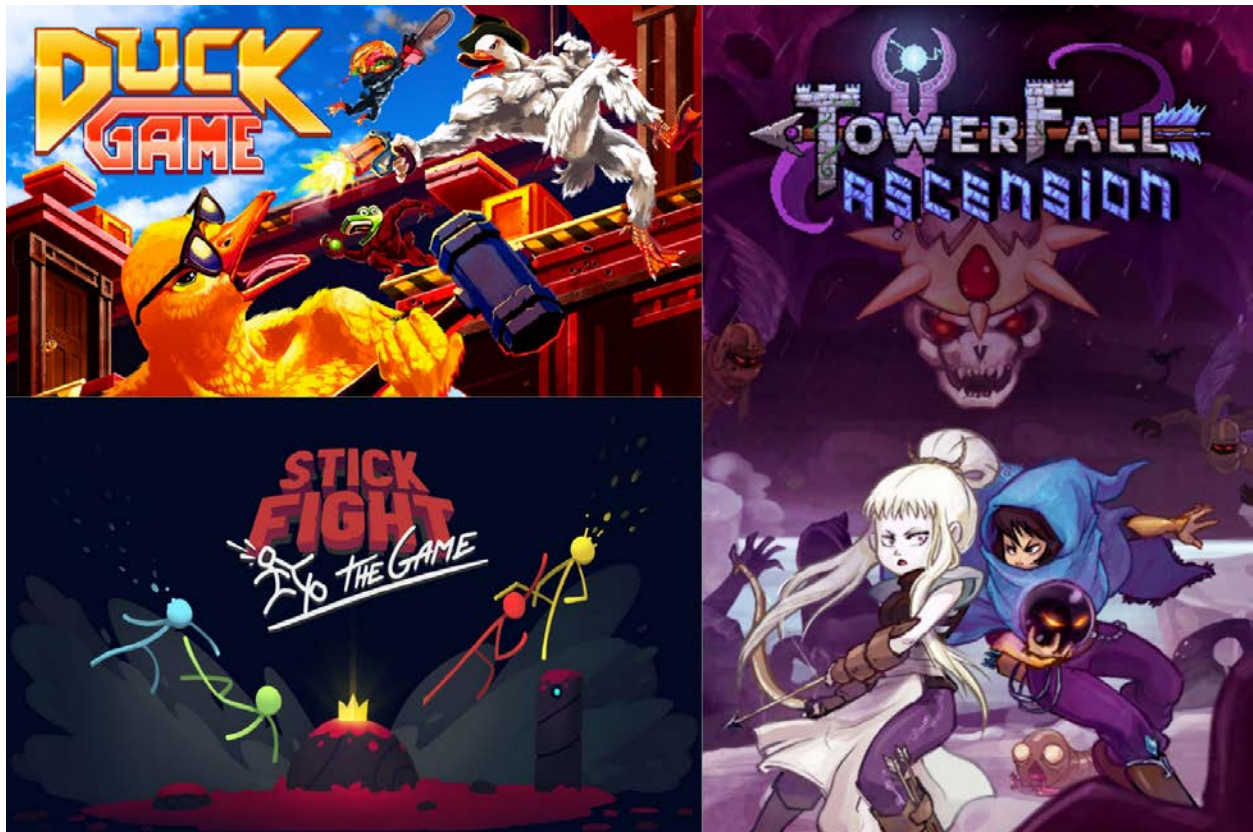


Figure 4: Comparable games for our initial concept

2.1.3. Finalized Game Concept

Our final concept ended up being a cooperative gardening game where players would work together to create and maintain a virtual garden featuring gnomes dwelling inside of it, with emphasis placed in it being a relaxing, teamwork-encouraging experience. It would feature a wide variety of plants enabling the player to freely customize their garden both aesthetically and mechanically with all the gnome interactions possible.

2.2. Game Inspirations

The team had a few inspirations from other media that evoked similar emotions and gameplay that we aimed to emulate.

One of these games is *Animal Crossing*, where players can customize their own island or town in whatever way they'd like. Some methods involve funding public projects, planting flowers and other plants to better fit your aesthetics, and interacting with the villagers on your island in various ways. These influences can be seen in our concept through our efforts to allow players to express themselves through constructing a garden aesthetically and interact with the gnomes using said plants in whatever off-the-wall methods they desire.

Overcooked! is another primary inspiration. Players collaborate with each other to effectively cook a meal with supernatural stipulations, having to work together to surpass the various obstacles in their way. We wanted to emulate this type of gameplay, as players need to work together in order to grow plants and maintain their garden.

We also took inspiration from a few projection-based exhibits in the vein of what we wanted to make, as it was repeatedly emphasized that our approach was akin to a performance art. The biggest inspiration among them was *A Musical Wall Where Little People Live*, (teamLab) as its premise surrounding manipulating a bunch of autonomous people in a projected play space was fairly similar to our idea of a garden being inhabited by a bunch of gnomes.

2.3. Art Inspirations

The art of our game was generally influenced by cartoony, stylized games. The premier games in our art bible were consistently games with cute, cartoony, and detailed artstyles such as 3D Mario games, *Plants vs. Zombies*, *Pikmin*, and *Overcooked!*. As seen in the designs of our game assets, we took after this style and made several stylized plants and gnomes front and center in *Gnome Hollow*.



Figure 5: Visual and gameplay inspiration for our project

2.4. Audio Inspirations

The soundtrack was inspired by the *Plants vs. Zombies* soundtrack. Both tracks feature a string instrument playing the melody, and they both have repetitive chords. However, over several iterations, our soundtrack became much more calming and upbeat than the eerie *Plants vs. Zombies* soundtrack.

3. Equipment

The original proposal envisioned the team using two high lumen, high definition projectors to project the game onto the face of a building, as well as using projection mapping software to distort the image to properly display the game on the building's architectural features. We also needed proper equipment for audio to ensure that the player had a well rounded experience. In this section, we will go into detail on our method of researching the equipment we needed and how we arrived at the equipment we will be using when displaying our project to the public.

3.1. Projector Research Process

As the original concept for this MQP was to project our game onto a building, we started doing research on the type of projector we would need. We started out by researching different companies that work with projection mapping. Through this, we found various organizations and reached out to them to talk about the process of working with projection mapping. We were able to get two interviews, one with Eric Newkirk from AVFX and another with Brian Corrigan and Justin Gitlin from OhHeckYeah.

We were able to get valuable information from both interviews. The questions we asked both interviewees can be found in Appendix F. When creating art assets, the surface you are projecting on should have a great influence. Not only should you incorporate features of the surface in your models, but you should also consider how the surface may unintentionally change the look. There can be occasions that cause the projection quality to degrade if there are abnormal features, like a large slope which would cause inconsistent focus. Another factor to worry about is lighting, as most projectors don't have enough power to display a visible-enough image in daytime environments. We also got advice on what to think about gameplay-wise: simplicity is key, keep the game simple and short, we were not making a game where people buy and download it, and the game is meant for people to quickly play as they pass by. The game should be self explanatory and simple to understand, unless there will be a staff member or volunteer there to give a brief explanation. When it came to equipment, at the very least we would need one high-lumen projector. Depending on the size of the wall, a second could be necessary. However, two high-powered projectors would lend itself to a high cost. As for projection mapping, we were recommended tools like Resolune, MadMapper, Spout and Siphon.

3.2. Speakers

We did not receive as much info about audio from the interview as we did about game content and projectors. When it comes to audio, we were able to use more general equipment, unlike projectors which need certain specifications. We were told to consider how important the audio is in the game. If audio is a main pillar of the game, then we should consider a high end system that covers the play area. If not, our interviewees recommended brands like JBL and Mackee, as they had self-powered products of good quality at an affordable price.

3.3. Budget and Pitch

As we were warned during the interviews, the projectors we needed were expensive. We started doing research and found out that the projectors we need are about \$15,000 or more. Because of this, we turned to the budget WPI gives to each MQP. Our advisors informed us that we would be receiving \$100 per team member, meaning we had a total budget of \$400. With our shoestring budget, we were unable to get the projector ourselves.

3.4. Lens and Lights

Due to our budget being too small to purchase a projector ourselves, we searched for alternative ways to get a projector. We turned our search to WPI's Lens and Lights club (LNL). We met with an LNL officer in B-term to learn more about the projectors that they have access to. LNL provided helpful advice on what kind of projector we would need for our game, even offering to assist us by giving us the opportunity to test one of their own projectors. They offered us a Panasonic PT-RZ120, a \$15,000 projector similar to the ones we were looking at when we did our own research. During this test, we were able to see how the game looked in different lighting conditions and at different distances. Since we were unable to afford a projector, LNL offered to let us borrow their projectors and audio equipment when they are not in use.

3.5. Final Decision

Due to the fact that we did not have a large enough budget, we decided to go with Lens and Lights' offer to let us borrow their equipment. Not only would we not need to worry about buying the projector, but we wouldn't need to store nor learn how to properly work the projector. All we needed to do was inform them of a time and place and they would handle setup and proper care of the projector.

However, due to miscommunications between LNL and the project team, they ended up acknowledging our request rather late and may not be able to allow us to use a projector for Showfest.

4. Design

Once our team had decided on our final concept of a garden-based cooperative game, we needed to work out the finer details of our game's design. When making these decisions, we made sure to always keep in mind the cooperative nature of the game, making the game easily accessible for everyone, and maintaining the whimsical and cartoony garden aesthetic. While these were important guidelines to follow, there were still many questions to be answered. For example, what would the game's perspective be? Would it be 2D or 3D? What would a standard round look like? Would players have a goal? All of these questions and more had to be answered before we could properly begin development.

4.1. Game Perspective

Our art team consisted of two 3D artists, so we knew we needed to incorporate 3D art into the design. We immediately ruled out first person, as that would break any immersion that the game was actually taking place on a wall. After taking pictures of the Higgins building and sketching some mock-up screens, we decided the best direction would be a 2D locked camera perspective from the side, but in a 3D world.



Figure 6: A mockup sketch of what areas could be used for gameplay on Higgins House

The intended effect is that players would be looking into another mini world through a hole in the wall. Having 3D assets would also allow us to play with depth, so the projection seems like it actually sits inside of the wall instead of just on it.

The building we chose and projecting our game had more implications for our game's design than just the perspective of the game. For one, the game needed to be visually striking. We wanted the game to be eye-catching to anyone passing by in order to intrigue them and get them to come over and try out the game. Additionally, projection typically leads to a loss of color and vibrance based on the projector's lumens value, projection distance, external light, and more.

We knew we would need our art design to be overly bright and colorful to satisfy all of these obstacles.

4.2. Target Audience

We had previously discussed the target audience while forming initial game concepts, but we needed to determine a more solidified answer to ensure we knew who we were developing a game for. In our initial discussions, we had the idea of our game being used as an introduction to WPI and the Interactive Media and Game Design major by combining video games and the campus itself. Another idea we considered was having the game be played during New Student's Orientation (NSO) as a way for new students to work together and meet new people, while also becoming familiar with campus. After more discussion, we wanted to expand the idea to not be limited to just a few days out of the school year. Our intended audience expanded to any of the students on campus. We wanted to create a game that would appeal to anyone: whether they were highly adept at games or had never touched a controller before. We made sure to incorporate as many different playstyles into the game as possible to allow anyone on campus to enjoy. Players can create beautiful, lush gardens with happy gnomes wandering about, or they can destroy any sign of plant life and toss the gnomes around for fun.

4.3. Experience Goals

The main idea of the game is to encourage collaboration between players in a stress-free environment. It gives students a chance to meet new people on campus and work together to achieve whatever they desire. Academic work can become stressful, so we wanted to create a game that could be experienced quickly in between classes or during social events around campus. Our design was to give students a moment of respite, so they could stop for a moment and play something that would make them smile.

The emotions we want our game to evoke can be conveyed well by the arousal-valence model, which is used to graph all human emotions on a scale of arousal and pleasure (Posner, 2005). The experience we wanted was for players to increase in pleasure as they play, until they experience a calmness or zen state of mind. Then, after completing and seeing their garden come together, arousal would also increase so they would end up feeling much more joy than before they played. This experience can be seen graphed on the arousal-valence chart in Figure 7.

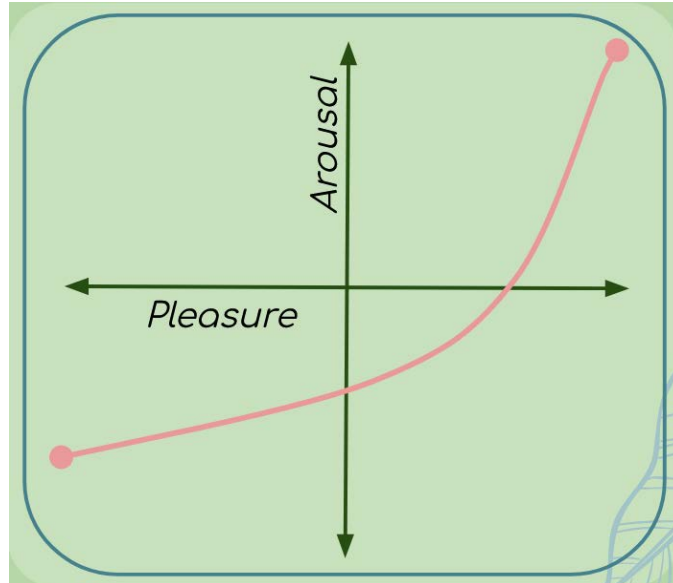


Figure 7: Team-made graphic of the arousal-valence model

The experience goals we set made an impact on the gameplay design as well. The garden would start calm, but bland and boring. The music would be simple and there would be little visual stimuli. It would be up to the players to make their garden their own: the music evolving with the plants on screen, sounds and particle effects coming from the happily animated plants, and the screen becoming more and more filled with life. The design was that players' moods would evolve with what is happening on screen, making them more lively and stress-free by the time their garden was complete.

4.4. Mechanics

Keeping simplicity and cooperation at the forefront, we decided to create a game where players work off of one another to build and maintain a garden inhabited by gnomes. Each player would have different abilities, meaning that they needed to cooperate and coordinate to ensure their garden would thrive. Each mechanic, from the controls to the non-player characters (NPCs), was designed to be endearing and simple to ensure we met our target audience and our experience goals.

4.4.1. Player Controls/Tools

Our primary goal for the game's controls was to make them as easy as possible to learn, as most players would be picking up the game on a whim without any prior knowledge of how it works. We decided that the players would be interacting with the gameworld through a cursor-based style of gameplay, with their cursors being represented using different tools with unique functions. We decided on cursor-based for simplicity, both from an audience and coding perspective. One consideration we had was for the players to be gnomes, jumping from shelf to shelf to tend to the garden. However, with only one tech student, the challenges of making

platforming feel fluid, and wanting the game to be easily accessible, we decided that having cursor-based gameplay was ideal.

Players control various tools that are used to do different tasks around the garden. Our design centered around 6 different tools that would make sense in a garden setting: a glove for interacting with gnomes, a seedbag to plant seeds, a fertilizer bag to create mounds of dirt, a watering can to water plants, a shovel to dig up and move plants, and shears to remove grown plants. With 6 tools and a cooperative style of gameplay, we then decided how to split up the tools per player. We decided to split the tools evenly, with each player having a tool with the functionality to add, move, and change. One player had the seedbag, shears, and shovel, while the other had the fertilizer, watering can, and glove.

Another design decision we needed to make was the button layout for the controller. We wanted to limit the amount of buttons used as much as possible to ensure a simple learning curve. We chose to dedicate one button per tool type, as well as having the joysticks be used for movement. To allow even easier access for players, the controls were mapped to both sides of a standard controller, meaning the game could be played left or right handed.

4.4.2. Plants

With a game centered around creating a garden, we knew we needed the plants to be well designed, as they would be the main draw of the game. We began to design a variety of plants to be in the game, from simple and common plants to more outlandish and silly ones. Our goal was to make sure that no two gardens would look the same, and an abundance of plant types was our first step in achieving this. We wanted to strike a balance of how many we had in the game: Not too many so that the players would risk becoming overwhelmed, but enough to encourage replayability and different playstyles. Additionally, the players would not be given the ability to grow every type of plant immediately in order to not confuse the players, to keep the gameplay fresh, and to add to the game's replayability. Each plant would have a unique model, but all plants would follow the same mechanics and functionality.

Growing Plants

To get the plants to actually appear in the garden, they needed to be grown first. We went through a few variations of how planting would work. At first, seeds would be dropped anywhere on a shelf, and once they were watered, they would grow into a plant. However, the plants would look unconvincing, as they would sit on the shelf and have no base or depth to them. Another iteration was having plant beds with dirt behind the shelf that the seeds would sit in until watered. The added depth worked well, but the game looked too pristine and organized, taking away from the organic and fun style we were aiming for.

Our final version of planting and growing involved combining the two previous iterations. We kept the depth by making plants sit further back on the shelf, but took away the plant beds. Instead, players would now pour dirt on the shelf wherever they wanted to plant something. Then, once a seed was dropped and watered on the dirt, a plant would grow from the

mound. Plants would also be given slight random adjustments to their size and rotation of the mesh in order to add variety to each round.

Plants

Our grown plant designs followed a consistent aesthetic of cartoony and whimsical. A common pitfall with 3D modeling is creating models that look too artificial or symmetrical. We made sure to break symmetry and used curvy lines when designing them in order to make them feel more organic and fill them with personality. Not only did this help make each plant feel believable and natural, it also helped create unique silhouettes for each plant so they would be instantly recognizable. This principle carried over to the animations as well, which we made bouncy and exaggerated. Each plant was also designed with a unique particle effect in mind. The effect would consist of a burst of multiple copies of one 2D hand-drawn sprite that would emit from the plant after it was grown or interacted with.

Maintenance

To ensure that a plant would not become obsolete after growing, we designed additional interactions for plants. Customization possibility was a major design point for us, so we wanted to include ways to move and delete plants that were grown. However, once a plant was in a spot that the player liked, there was no reason to return to it. Our solution was to include a wilting mechanic, where the plants would begin to turn brown and slump over, breaking the bright visuals of the game. To reverse the wilting, players would simply have to rewater the plant.

4.4.3. Gnome

With every iteration of our design, we've wanted to include some sort of mythical garden creature, either as a playable character or a non-playable character (NPC) to add to the player experience and make the world feel more populated and alive. Our first idea was to have fairies that worked around the garden in tandem with the players. After more consideration, we decided to have our NPCs be gnomes, as we felt they fit the garden aesthetic more than fairies or sprites. When in the game, they would walk around and interact with plants on their own, so something was always happening on screen even if the players had stopped, left, or were changing out with another player.

4.5. Gameplay Loop

Determining the structure of our game was an important step for our team. Like many aspects of our design, the game structure went through multiple versions. We wanted to keep in mind a stress free experience, while also encouraging as many players getting to play the game as possible in a short span of time. The multiplayer aspect of our design aided us in the latter goal. With two people playing at once, twice as many players would get to experience the game in the same time than if it were single player. We also originally had round timers to allow a better flow of players in case of a crowd. However, we found that timers took away from the

stress-free experience and decided to allow the game to be an ever-evolving garden, with input from a variety of people with different ideas and visions.

After deciding on this approach, we needed to consider goals, points, or win conditions for the game. In keeping with the relaxing experience goal, we designed the game to have no points and no win or loss condition. It would be a pure sandbox experience that anyone could add to, take away from, or change, similar to a campus-wide toy to be played with. Nevertheless, we intended for the game to still have some challenge to it, and it would be through the multiplayer mechanic. Instead of the game being a garden shared by two players with the same abilities, the tools would be split between the players. With this approach, players would need to communicate in-person in order to plant, water, grow, and rearrange plants, as seen in Figure 8.



Figure 8: Flow of the game between two players

Each player would have the ability to add, move, and change something on the screen, but in a different way from their partner. Working together would be the only way to make a garden, and ideally, people would meet others and enjoy the challenge together.

4.6. User Interface

Balancing user interface (UI) to make sure players were well informed of how to play but not overstimulated by information on screen was a major design challenge. We explored various UI templates, such as a toggle wheel for tools or a docked shelf to retrieve seeds from as seen in Figure 9.

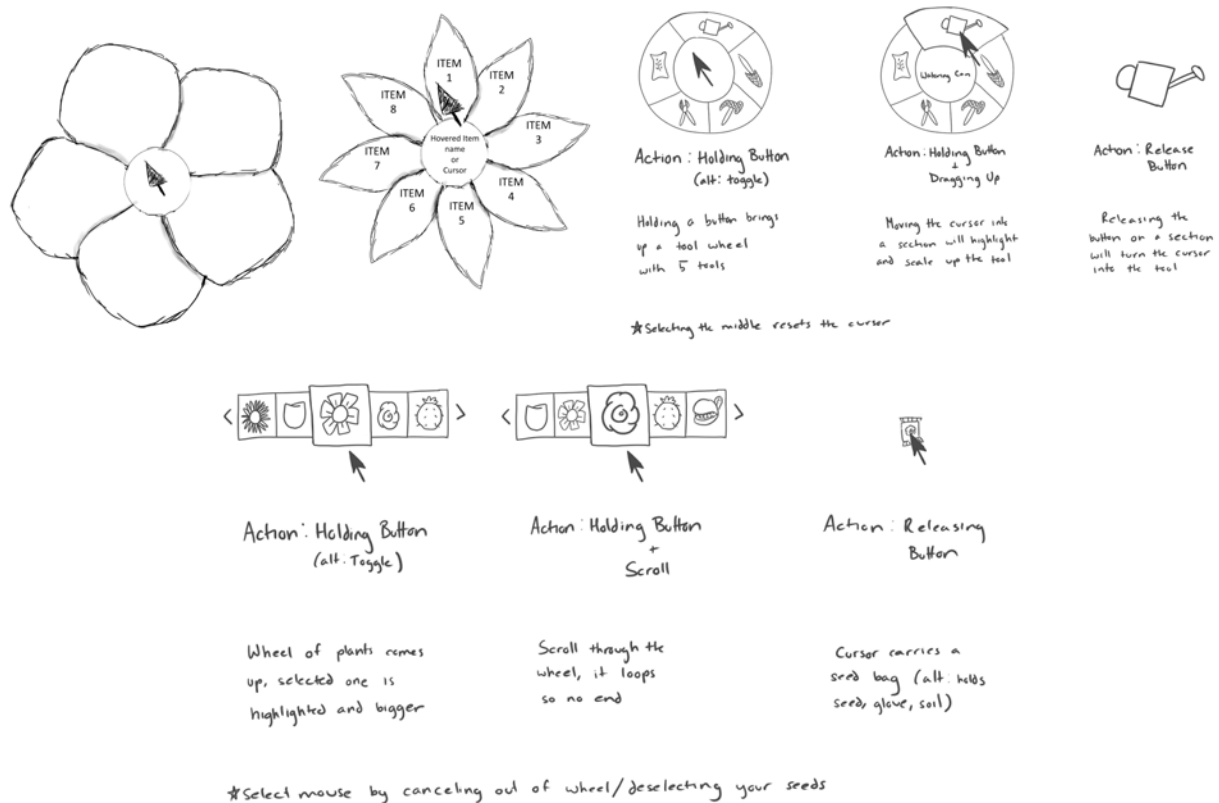


Figure 9: UI concept art for a different tool selection styles

The more UI we considered, the more alienating the game became for new players. For this reason, we determined we should be as restrained as possible when adding UI to our project. The main instances where we included UI were in creating introductory tutorials and through minimal pop-ups in game.

Tutorial

The main concepts we wanted to communicate to our players before they began playing were the mechanics of the game, the sandbox nature of the gameplay, and the controls. The first two were achieved by creating a short cutscene from gameplay footage that showed the process of piling up dirt, planting seeds, and watering them to grow plants. A frame from this trailer can be seen in Figure 10.



Figure 10: Frame from the intro tutorial cutscene

Once the short scene played, there would be an additional tutorial screen that showed a diagram of a standard controller with the button bindings explained.

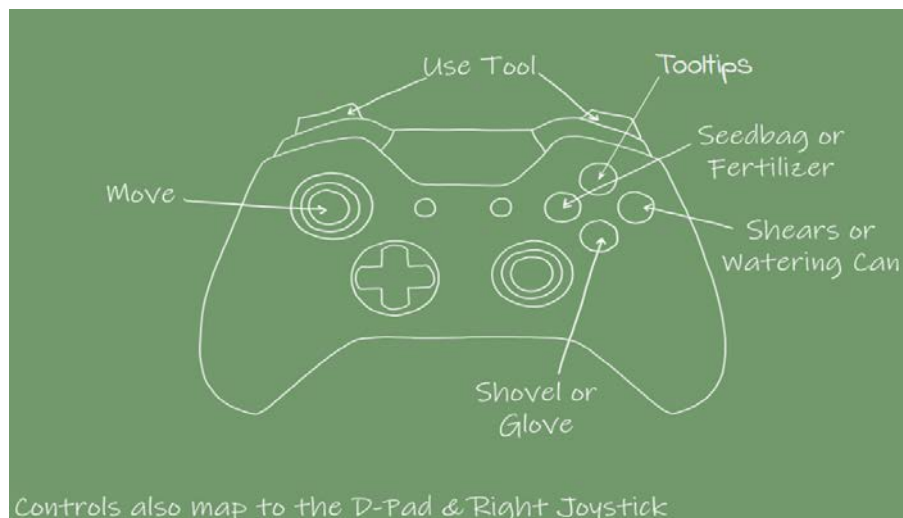


Figure 11: Controller button mapping

We made both tutorials skippable in case the player already knew the controls or wanted to get right into the gameplay.

In-Game

Simplicity was key for the in-game UI. Our game was designed to be stress free and easily accessible, and a cluttered screen would both isolate our audience and look out of place on the side of a wall. We still wanted to include some ways to guide the player in-game to supplement the introductory tutorials. To accomplish this, we designed basic tooltips to remind players what each tool is used for, such as in Figure 12.



Figure 12: In-game tooltips

We also added short, occasional pop up text that informs them of what to do next or if they have picked up a new seed.

4.7. Level Design

At first, we wanted our game to include player-based level design to add even more to the customizability of each garden. The idea was to have a simple starting layout that would then be edited and expanded on by those playing. While this was an appealing approach at first, it proved a technical challenge. We determined that having one strongly designed level was the best approach we could take as a team, and we got to designing the layout.

Designing a good level was challenging, as all we had to work with was shelves on a wall. At one point, our game would have started with the camera much closer to the shelves, then it would pull back and expose more and more shelves as the game continued. With this in mind, we wanted the central area to be as accessible for new players as possible: simple, symmetrical, and interesting. However, as we moved away from timed rounds, this concept was scrapped and we kept the camera static. Therefore, we needed a level design for the entire game screen. We wanted our design to include multiple factors: an interesting layout, broken symmetry for an organic feel, variety in shelf sizes, a balance between not too empty nor too busy, and a chance for replayability.

With all of our goals in mind, we began to sketch out ideas for how the design may look. We also had to consider verticality, as our plants mostly grew upwards. Shelves could not lay directly on top of each other, as some plants would not be able to grow under them.

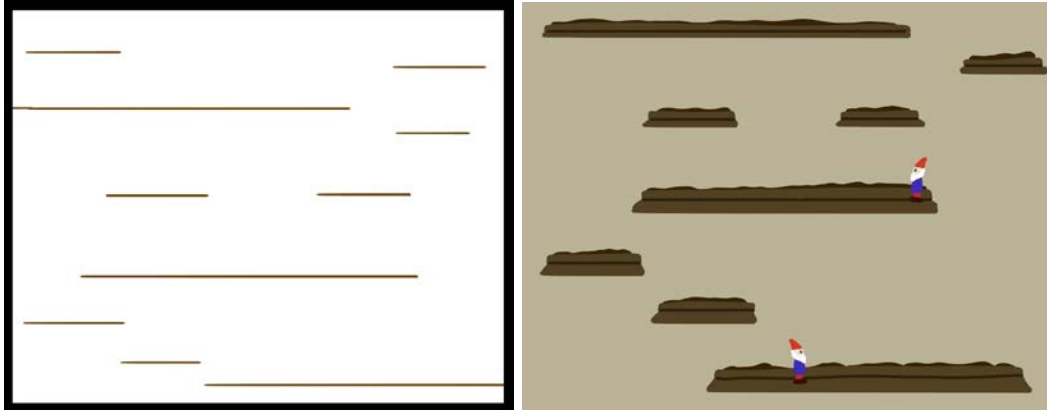


Figure 13: Early mockups for shelf level design

When designing the level in-engine, we realized we needed to experiment more with the camera. Originally, we had the camera facing the shelves head-on, but this resulted in a loss of depth, and the top shelves were viewed from the bottom. After adjusting camera height, angle, and field of view, we landed on a perspective that allowed all shelves to be viewed from above. Once that was accomplished, we tested spacing and layout of the shelves. We ended up striking a balance of long and short shelves at different heights and enough empty space that the world looked barren at first, but lively and populated once plants were grown all about.



Figure 14: Level design in-game

In our design, the shelves were in a simple spiral-like pattern that winded up the screen. We wanted players to naturally move their cursors up and down between the shelves as if they were ascending stairs on the wall. The shelves' layout allowed plenty of space for the garden to be built up and customized while also naturally guiding gameplay and looking organic.

Late into development, we went from having the game's background be a 2D image to a 3D brick wall asset. The wall added even more depth to the scene and contained spaces of grout between each brick.

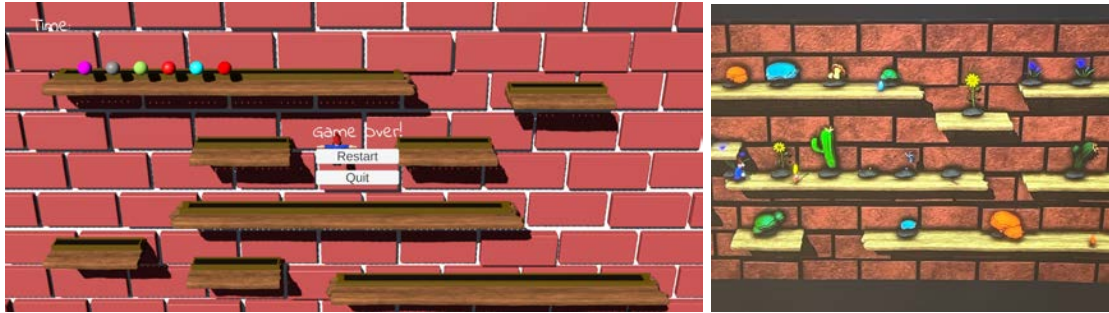


Figure 15: Brick wall background

This led to minor level design adjustments, as the shelves fit between these spaces to make them appear mounted to the wall. Most of the shelves were already in a place that only required small edits to make them fit the new background. A few assets needed to be resized, moved, or removed, but we kept the same principles as we started with to finalize the level design to fit with the new background.

4.8 Game Name

After designing mechanics, plants, gameplay, and more, we still needed to decide on the most important part of our game: its name. The name needed to be unique, recognizable, and descriptive for what sort of experience we were offering players. Our team created a document to brainstorm names with separate categories: names that reference gnomes, names that reference gardens, names that reference both, and any additional names we could come up with. We utilized references, wordplays, and alliteration to get a name that sounded fun and catchy. Some examples were *Gnomeish*, *On the Garden Wall*, *Gilded Garden*, and *Planter's Paradise*. To decide on a name, we consulted within our team, advisors, and students for what title caught their attention. Our final decision was to name the game *Gnome Hollow*, as the garden was inhabited by gnomes and took place in a hole to another world. The name was fitting, cute, and concise.

5. Art Development

Art was a core part of the game, as it was what our players perceive and interact with within our game. This included every visual aspect of the game: 3D models, textures, animations, and particle effects.

5.1 Aesthetic

Before creating any art assets, we first needed to establish an art style and aesthetic to follow. We researched various images of games and other artistic images with a common style. As the main idea of the game was to project it on the side of the building, we chose images with bright and saturated colors to compensate for any desaturation that occurred when using the projector. In this section, we will take a more detailed look into how we established our art style and color choices.

5.1.1: Art Bible

When we first started the project, one of the first steps we took was establishing our art bible. Our art bible consisted of a Google Slides presentation containing screenshots of games that had the same art style we had in mind. The screenshots were from various games, including Plants VS Zombies, Pikmin, Mario Odyssey, Scrap Mechanic, Mario Maker, Overcooked, and Animal Crossing. The art style we ended up with was a 3D cartoony game with an emphasis on vegetation and simple yet stylized characters. We also gathered images from other non-video game sources including the gnome illustrations of Rien Poortvliet, who was a great inspiration on the final look of our gnome character. The full art bible can be seen in Figure 16 and Appendix E.

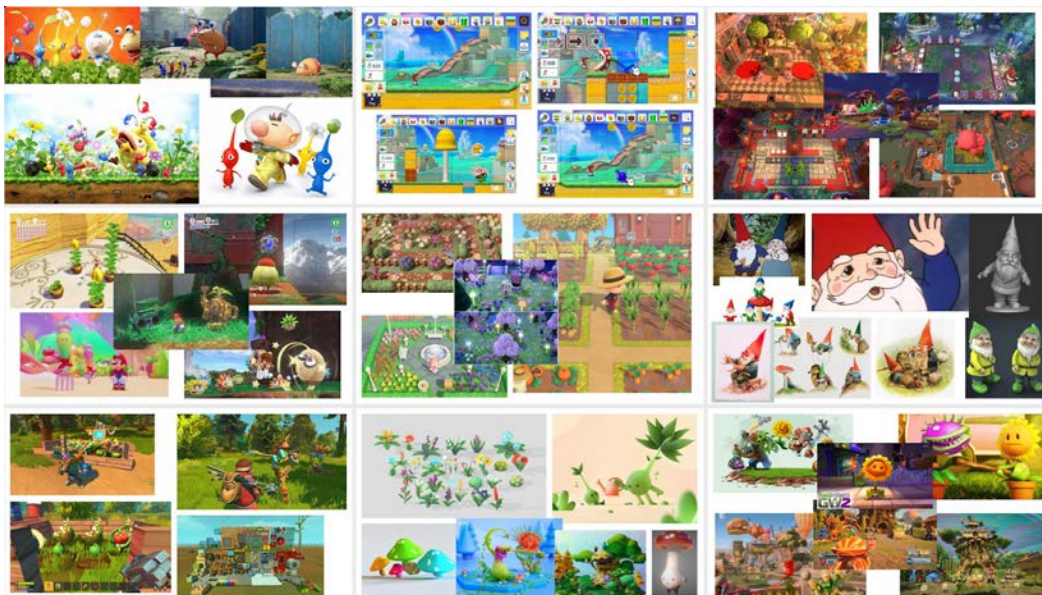


Figure 16: Art Bible

5.1.2: Color

When deciding how we would handle the overall look of the game, we also had our medium in mind. As we want to present the game on the side of a building by using projectors, we wanted to ensure that there was no loss in quality when doing so. Because of this, we decided that bright colors and high saturation would be ideal. We did this to ensure that darker and less saturated colors would not get lost because of the projector's capabilities or due to the background of whatever surface we projected on to. The colors used in the plant's texturing can be seen in Figure 17.

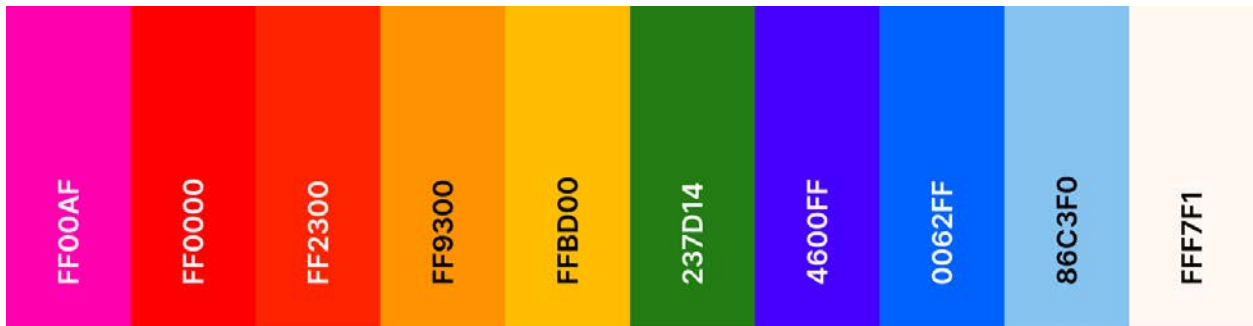


Figure 17: Colors used to texture each of Gnome Hollow's plants

5.2. Asset Pipeline

Creating a streamlined and efficient art asset pipeline was crucial to completing the game on time. Since we had two 3D Modelers, Diego Arce and Kirsten Roethel, we were able to split up the work in an efficient manner. As Diego favored hard surface modeling, he took on the creation of the tools and prop models as well as texturing for all models. Kirsten, who favored organic modeling, took on the gnome and plant models, as well as rigging, animating, and creating particle effects for all the models. In this section, we will take a detailed look at the pipeline we developed to create the game's art assets.

5.2.1. Conceptualization

We conceptualized everything by first gathering references. For the plants, we used both real life examples as well as stylized art of the plants. For the tools and props, we gathered real world references with dimensions to ensure that the tools were accurately represented. We found various gnome illustrations by Rien Poortvliet and images of David the Gnome to create the basis for our gnome design. After collecting all these images, we then referenced the art bible to conceptualize art assets resembling the bright cartoony style that we settled on. Particle effects were created from scratch based on the final look of the models and the plant they were meant to represent. Particle effects were used to give the user feedback as a way to confirm that they had successfully used a tool or planted a seed. Some of the reference images used can be seen in Figure 18.

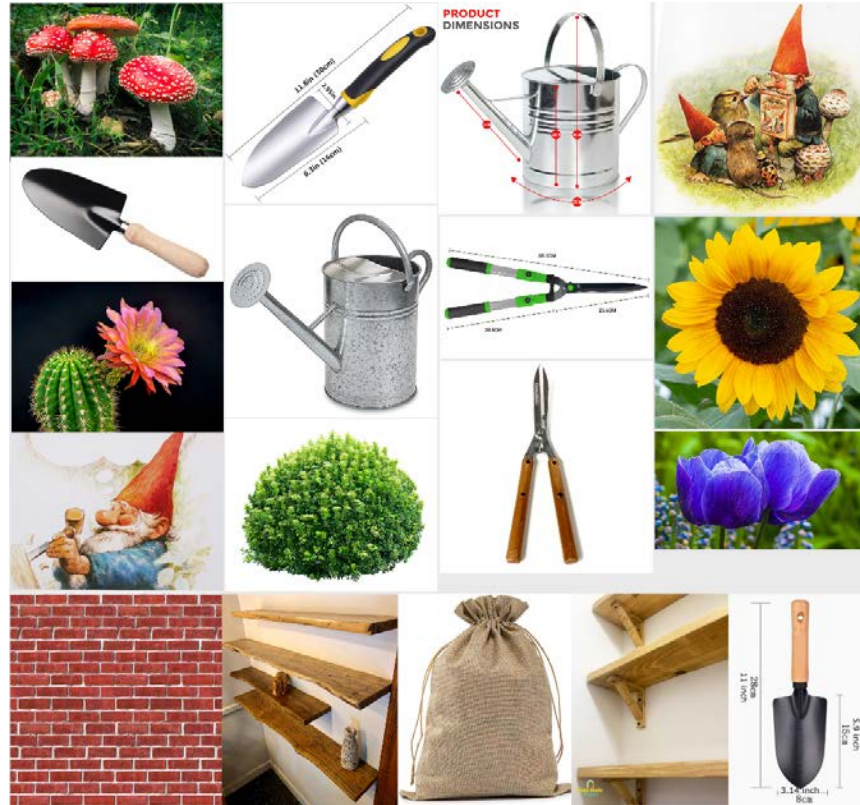


Figure 18: Reference Images

5.2.2. Modeling

As mentioned earlier, we split up modeling by having Diego do all hard surface models and Kirsten do all the organic models. Hard surface models included the shovel, shears, watering can, fertilizer bag, seed bag, seed, shelf and plant bed. They were modeled in Autodesk Maya, as the program is built mainly for hard surface modeling. Organic models included the glove, gnome, tulip, mushroom, cactus, sunflower, bush, glacier bush, berry bush, fireberry bush, and vines. These were modeled in ZBrush, as the program’s sculpture-based interface is better suited for organic modeling. We started out by creating a low polygon version of the models that had a simple shape that was still easily identifiable as to what the model resembled in game. The models created in Maya also received simple material colors to create a placeholder material to aid the texturing process. These placeholder materials were allocated to specific parts of the models that we knew would have a completely different look from other parts of the model once texturing would be completed. We used these low-poly versions in-game as placeholders during early playtests while the final version of the models were still in progress. We took the low polygon models and increased their polygon count to create high polygon versions of the models with extremely detailed features to be able to more accurately depict their real-life references. Figure 19 displays an example of low-poly placeholder models in comparison to their higher-poly final versions



Figure 19: Original and Final Seed Bag Model

5.2.3. UV Unwrapping & Texturing

The models received UV Unwrapping in the same application that they were modeled in. Models created in Autodesk Maya were mostly unwrapped using Maya's auto unwrap feature. If the program produced an unwrap that had unwanted seams, we would manually touch up the auto unwrap or unwrap the model from scratch. ZBrush models were divided into polygroups to indicate what each UV island should look like. UV Master was used to unwrap the models using the groups as a base for where the seams would be.

Texturing for all the models was done in Substance Painter. The low-poly models were brought into substance painter, where the high-poly models were baked onto them. This allowed us to have the details of the high-poly model while keeping the low-poly polygon count. We then gave the models color by using Substance Painter's material library. We found an ideal material to use as a base and then modified it to get the high contrast cartoony look we had conceptualized. We were able to add certain effects like dirt or wear to models by using Substance Painter's generator feature to make the models seem like they were used in a garden. In Figure 20, we showcase how the mushroom was textured in Substance Painter.

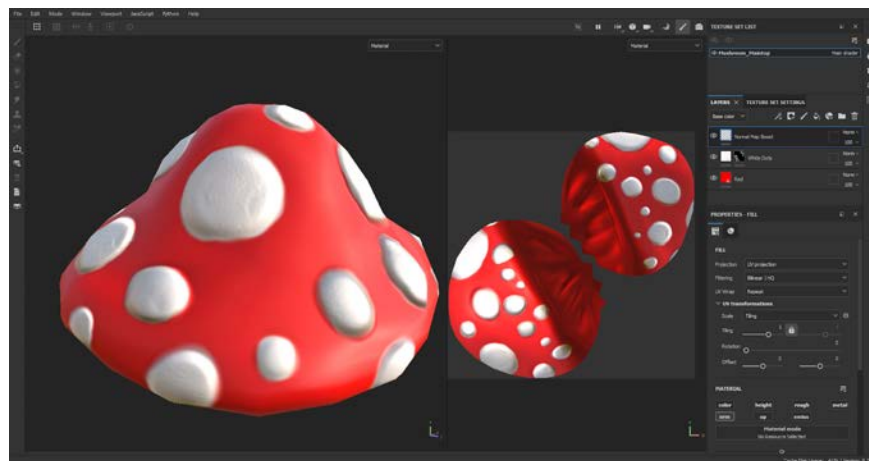


Figure 20: Mushroom Texturing

5.2.4. Rigging & Animation

Models were imported into 3DS Max for the rigging and animation process. Rigging was done using 3ds Max's Character Animation Toolkit, or CAT for short. The CAT system lets us rig our models in an extremely easy way without sacrificing rigging quality. Once we had determined the final scale of the model that we would use in-game, we began the rigging process. We created two copies of each model: One for morph targets, and one for rigging. Morph targets were used to create the ungrown, squashed, and stretched version of the plants. The copy of the model without morph-targets was given its own CAT rig which was skinned and enveloped properly. The skinned copy was then skin wrapped to the mesh with the morph targets. This resulted in a fully rigged mesh with morph targets to simplify the animation process. Growing animations were as simple as changing the value of the grow morph target while adding some squash and stretch to accentuate the bounce. An example of a wilt animation can be seen in Figure 21. The gnome was not rigged manually. Instead, we used Mixamo's auto-rigging tool to rig and animate the model using Mixamo's large animation library. Rigging was as simple as importing the gnome's FBX and pointing Mixamo to where the chin, wrists, elbows, knees and groin were on the model. Once we had the gnome's Mixamo skeleton, we still needed to clean it up before we could use it in-engine. In 3DS Max, we repainted some weights to make sure the mesh deformed properly, as well as added a root bone for the skeleton to be parented to.

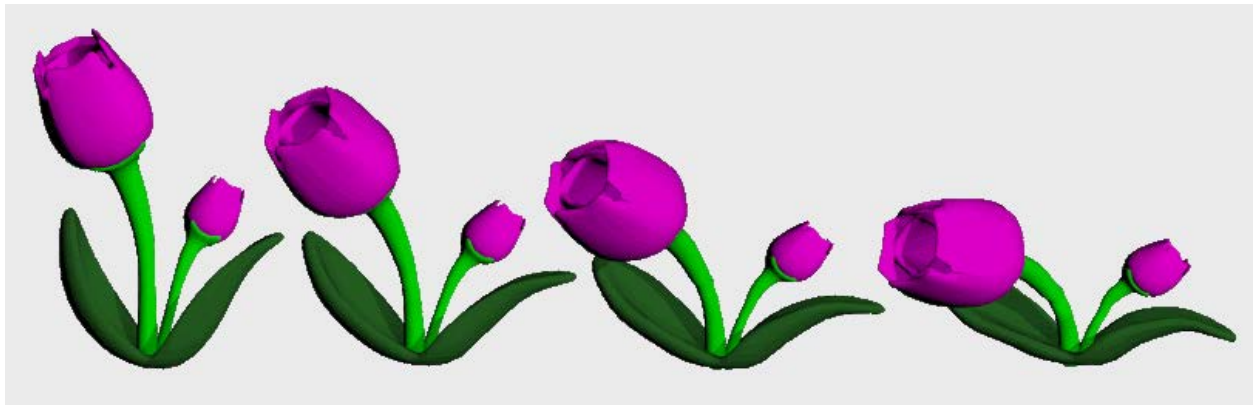


Figure 21: Wilt Animation

5.2.5. Particle Effect Painting

Particle Effects were created in Adobe Photoshop in a 512 x 512 pixel canvas. They are black and white alphas unique to nearly every plant and tool. The white areas would be completely opaque, while the black would become transparent. They were painted to add feedback to the user to ensure that they are aware that they planted a seed or used a tool. Examples of the particle effect alphas can be seen in Figure 22.

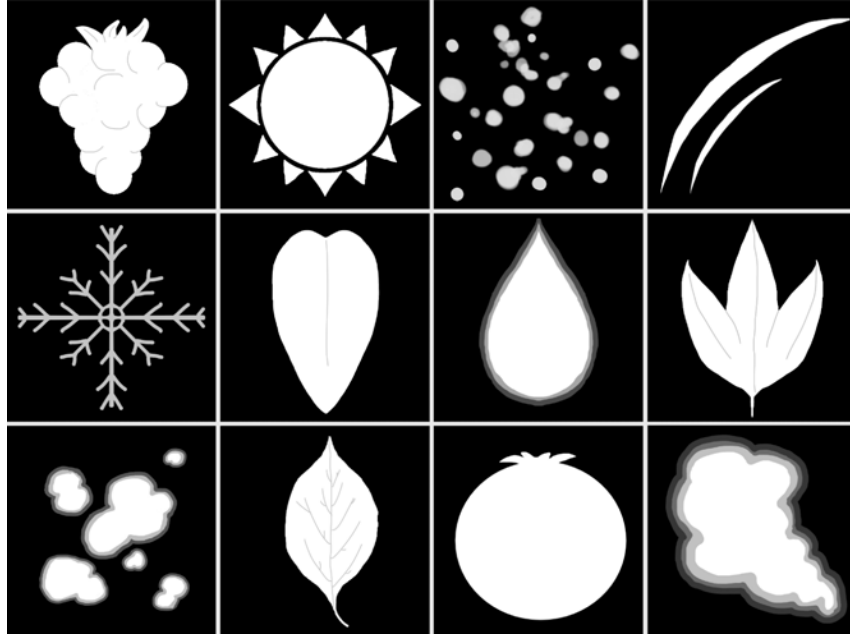


Figure 22: Particle Effect Alphas

5.2.6. Exporting and Importing

The exporting process for animation models entailed exporting a base skeleton with just its bones and no animations. Each animation was exported individually with the proper final frame count. The gnome was exported from Mixamo by simply downloading the animations without skinning information for each one. Animations were imported in their respective folders in-engine. The animation takes were duplicated and renamed appropriately for each animation. We created an animation controller with booleans for each animation state, then attached the controller to the appropriate skeleton.

Textures were exported from Substance Painter as five texture maps per mesh. The texture maps included an albedo, ambient occlusion, height, metallic and normal map. The textures were imported in-engine by simply dragging them into the appropriate folder. Unity materials were created by assigning the textures maps to their appropriate section in the material's inspector under the main maps section. We placed the albedo maps in the "Albedo" input, the metallic map in the "Metallic" input, the normal map in the "Normal Map" input, the height map in the "Height Map" input, and the ambient occlusion map in the "Occlusion" input. The materials were then put into each model's respective material input.

The particle effect alpha images were exported from Photoshop as PNGs, then dropped into their respective folder in-engine. Each particle had to have a material created using the alpha maps. We assigned a particle component to the asset that we want the particle effect to come out of, so we could modify the particle system to fit the effect we had in mind. We started by assigning the particle material we created into the component, then modified a multitude of factors including color, particle size, spread, particle count, speed, and more. By tweaking these attributes, we were able to create a small burst effect.

5.3. In-Engine

In the engine, we added simple, yet significant touches to elevate the visual appeal of the game. Using lighting and post-processing effects, we had an abundance of control over the game's appearance. For lighting, we kept it simple by only using two directional lights to mimic an outdoor setting: one a bluish hue to represent the sky, and the other a faint yellow to imitate the sky. For post-processing, we added multiple Universal Render Pipeline effects. In this section, we will showcase the specific ways that we used in-engine lighting and post-processing to elevate the overall quality of the game.

5.3.1. Lighting

In-engine we placed two directional lights. One of the lights was a white light angled at the wall. This light is meant to create shadows and fully illuminate the scene. The second light is light blue and pointed downwards. This light does not create shadows, as its intention is to brighten the shelves and everything on them to resemble an outdoor setting.

5.3.2. Post-Processing

We used Unity's Universal Render Pipeline (URP) for adding post processing effects. We added multiple effects, such as adding bloom and a vignette. The color correction effect modified post exposure, contrast and saturation. The shadow midtones highlights effect was added to change the color balance in the shadows to slightly tone down the red and green values. We took some time to experiment with the post-processing effects, going from subtle differences to extreme exaggeration. Different effects of post processing can be seen in Figure 23.

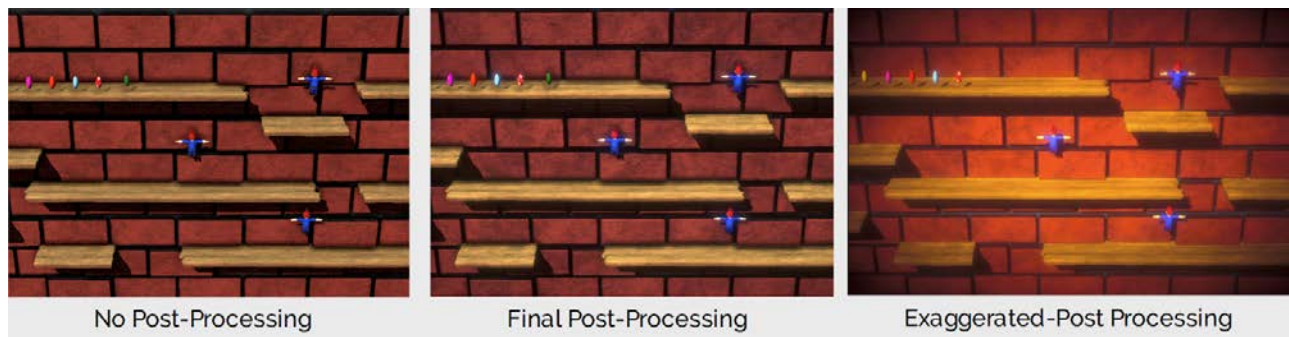


Figure 23: Post-Processing Comparison

6. Technical Implementation

To effectively showcase the team's art in an interactive environment, a game had to be engineered. In order to accomplish this, we carefully picked our software development tools and implementation strategies to both effectively facilitate the development process and keep things in scope for our Garrett, solo technical developer.

6.1. Software

In contrast to our hardware problems being fraught with expenses, the software chosen for the project was relatively cheap, with our game engine and source control software of choice being options that were easy for students to access.

6.1.1. Unity

Our choice of Unity was primarily a comfort pick for our technician, though there were also some other reasons. Garrett Sheehan, the project's technician, had had the most experience working on Unity compared to any other game engine, having a considerable number of projects and jobs under their belt developing in Unity. In addition, both of our artists would be taking a course at WPI during the period of October 2021 to December 2021 that would teach them how to go through the Unity art pipeline, making it a great long-term pick as both artists would be comfortable working in Unity at the time of importing final art assets. Our audio person also had a slight amount of experience implementing audio in Unity. Team member experience aside, Unity also had free and paid projection mapping plugin options enabling us to make difficult surfaces we projected onto easier to work with.

The team briefly considered using Unreal Engine 4 or 5, but we ultimately decided against it. One primary reason was that our technician wasn't as comfortable in Unreal Engine as they were in Unity, which could make implementing mechanics more difficult than it needed to be. Concern was also expressed over Unreal Engine 4's ability to support our team's collaborative workflow, as the engine largely rendering files as binaries and massive file sizes meant that source control collaboration would be harder to handle than in Unity. Finally, our team was unsure if our hardware options outside of the IMGD labs would be able to handle Unreal Engine 5 due to its required processing power and large storage requirements making us concerned that running it on our personal machines would be a significant challenge.

6.1.2. GitHub

Being able to effectively pass the project between team members using source control is a vital part of our workflow, resulting in our choice for source control being very important. Our primary options for source control were Perforce, GitHub, and PlasticSCM. Perforce was ruled out relatively early in discussion due to the technical barrier of setting up a server and no team members having prior experience using it. This left our primary options being GitHub and PlasticSCM. Both were very viable options for source control for different reasons, with PlasticSCM in particular being more forgiving of not-great source control methodologies that

would be catastrophic in a GitHub environment. For example, it's possible to merge a scene edited by different people together and have the changes harmonize in a way that works out, providing more safety nets in the case of a miscommunication. Ultimately, GitHub was chosen due to our technician having more extensive experience with it. Another significant reason was their experience teaching non-tech people how to effectively use GitHub Desktop, a companion application that makes GitHub operations way more accessible to artists/audio developers who may not be very familiar with traditional GitHub interfaces such as the command line. Teaching non-technical developers how to use source control would be a very important part of the project due to there only being one technical developer on *Gnome Hollow*, and GitHub Desktop was the easiest tool to accomplish that. In addition, one of our artists had prior experience with GitHub Desktop, further solidifying our choice for passing the game files around.

6.2. Player Tools and Controls

Implementing player controls and tools was more of a challenge on the design end than the implementation end, though it still took a fair amount of implementation to properly get controls up and running. They were revised often, a process which was thankfully made easier by Unity's New Input System.

6.2.1. Player Control Implementation

When implementing player controls, we chose to use Unity's New Input System to parse input from gamepads. Though it took a fair amount of effort to set up, the efforts were very worth it as it allowed player input to be easily modified. We could simply assign any number of buttons to any given input, allowing us to easily make modifications according to feedback from playtesters and advisors. It also allows us to use multiple controller types, with our gamepads of choice being the general Gamepad type and the Xbox controller type, enabling most common gamepad options to be covered.

Unity's New Input System also allowed us to easily implement multiplayer support. It boasts a unique player manager, allowing us to easily manage any number of players we could choose to spawn in. Though we primarily intended to stick with two players, having the option to go for more is always nice and more modularity with designing features is good practice when programming.

Cycling Tool Methodology

Our initial method of cycling tools had us use two buttons to cycle through 4-6 different tools, as it allowed us to reduce the complexity of the inputs from 6 buttons (one for each tool) to 3 buttons, (1 for a tool, 2 for swapping) allowing us in theory to better fulfill our experience goal of making the game easy to pick up and play. When playtesting, this proved to be not as easy to parse in practice as we hoped, leading to us workshopping a new control scheme with our advisors after playtesting.

Our new control scheme had a natural mapping system, with the six tools in the game being asynchronously split among the two players. Each player has a tool that added something new into the game, a tool that moved objects currently in the game, and a tool that changed the state of active objects. Pressing the button of a tool that was not currently active would result in the player switching to said tool, and pressing or holding the triggers would result in the tool action being executed. The only noteworthy exception to this process was the seedbag, which has two actions: swapping to the seedbag and swapping a different seed type. The former is done by clicking the appropriate face button when the seedbag is not currently selected, and the latter is done by clicking the face button again once the seedbag is on screen. This kept the complexity of the control scheme about as high as the previous control scheme while also making it more parseable to players. As a bonus, this control scheme is also friendly to left-handed players due to the only necessary controls being one control stick, three digital buttons, and one trigger. The right-handed control scheme involves the left control stick and the left, bottom, and right face buttons while the left-handed control scheme involves the right control stick and the left, bottom, and right D-pad inputs.

	Seed Player	Tool Player
Add	Seed Bag	Fertilizer Bag
Move	Shovel	Hand
Change	Shears	Watering Can

Table 1: The types of tools and what players they belong to

Detecting Player Interactions

Initially, we used raycasts from the player character into the background of the game to detect whether a player would be able to interact with an object. This worked great initially, enabling us to efficiently detect what could be picked up without much hassle. However, this did not scale with us wanting the game to detect if a player cursor was hovering over an object but not interacting with it, as we wanted to have some visual feedback as for whether a hoverable object is interactable or not. We chose to accomplish this by slightly enlarging selectable objects using DoTween if a valid selection could be made. Using raycasting for this would result in us making constant raycasts at least multiple times per second, which would tank the game's performance.

As a result, we chose to use collisions to detect if an object was being hovered instead. Entering a valid interactable object's hitbox would mark it as hovered, and pressing the action button would run the interact function on said hovered object. On the contrary, exiting it would mark it as not hovered, making the player unable to interact with it by pressing the button. This resulted in only two collision checks as opposed to a raycast every frame, making it a lot more performance-efficient, though there were some collision mismatches we would have to contend

with. The worst of these was dealing with seed pickups not being overly interactable, which was “fixed” by making their collision check hang for longer than every other check in the game, resulting in it being harder for their collision to be overwritten. Given how choosing to not pick them up is not a meaningful decision, we thought it would be fine despite the slight decrease in responsiveness. Collision boxes were enlarged over the course of development to make selections easier and allow player speed to increase without much consequence.

6.2.2. Player Tools

We had to implement a variety of tools that allowed players to interact with the garden in interesting ways. The tools are split up among both players asynchronously, with one player being intended to handle planting and moving seeds while the other player is intended to nurture said seeds. As mentioned before, each player has a tool that can add new objects to the game, a tool that can move objects around, and a tool that can modify the state of said objects.



Figure 24: All six tools the players can use. The top row (from left to right) features the glove, seed bag, and shovel. The bottom row (from left to right) features the fertilizer bag, the watering can, and the shears.

The seed bag, fertilizer bag, and watering have fairly similar underlying logic. The seed bag was given to the seed player, while the fertilizing bag and watering can were given to the tool player. Growing a plant is a three-step process involving fertilizer -> seed -> watering can, forcing cooperation between both players to effectively grow plants and fulfilling one of our experience goals. Each bag has a script attached to them that takes in a Unity prefab, and then spawns them using Unity’s update function. A delay was implemented so that said prefab would spawn as often as that delay, making it easier to control the spawning of said prefabs. The player

holds down the button to dispense, and releases it when they are finished dispensing. The primary difference between said tools is the effect their dispensable prefabs have on the environment.

The fertilizer bag dispenses clumps of dirt that are placed onto the environment. Players can then dispense seeds onto said clumps of dirt in order to “fuse” the two. In Unity terms, this makes the plant a child of the dirt mound, allowing interactions from the dirt mound to be carried to the plant. The watering can can then be used to nurture the plants.

Two of the other tools, the hand tool and the shovel tool, focus on moving objects. The hand tool primarily focuses on triggering plant interactions and moving around gnomes, with aforementioned collision-based interactions detecting what objects they are able to pick up at any given time. When an object is picked up, its position will be locked to that of the shovel/hand, forcing it to move in sync with the appropriate tool. This was done to avoid any potential desyncs when carrying around objects. The hand focuses on causing plant interactions and moving around gnomes, while the shovel focuses on moving plants around. Both the hand and the shovel tools also have a throwing force, applying said force to the rigidbodies of the plants and the gnomes. This makes the act of releasing gnomes and plants feel more satisfying, and helps to give the player more agency over where said objects end up with more ways to get them there.

The final tool is the shears. A fairly simple tool, the shears remove unwanted plants from the world, and are a relatively simple press input. This gives the player more customizability over their garden by allowing them to remove any mistakes they may have made. An animation where the plant falls away from the world was added to make the act of deleting a plant feel more grounded and fun compared to it being deleted from the world. This was given to the seed player, as they have more control over placing plants down and would be in an easier position to remove it than the tool player in many scenarios.

6.3. User Interface

We created several UI elements that can help convey the state of the game to players better. The most significant bits of UI involve our title screen, our tutorial screen, and our notification system in the main game.

Our title screen was a fairly simple scene in Unity that was intractable in a similar way to the final game. Orbs represent buttons that players can mouseover and click to perform certain actions. Our menu had a start button, a quit button, a menu button, and a spawn gnome button. This was scrapped in the final build for the sake of time and the game no longer having any sort of time pressure.

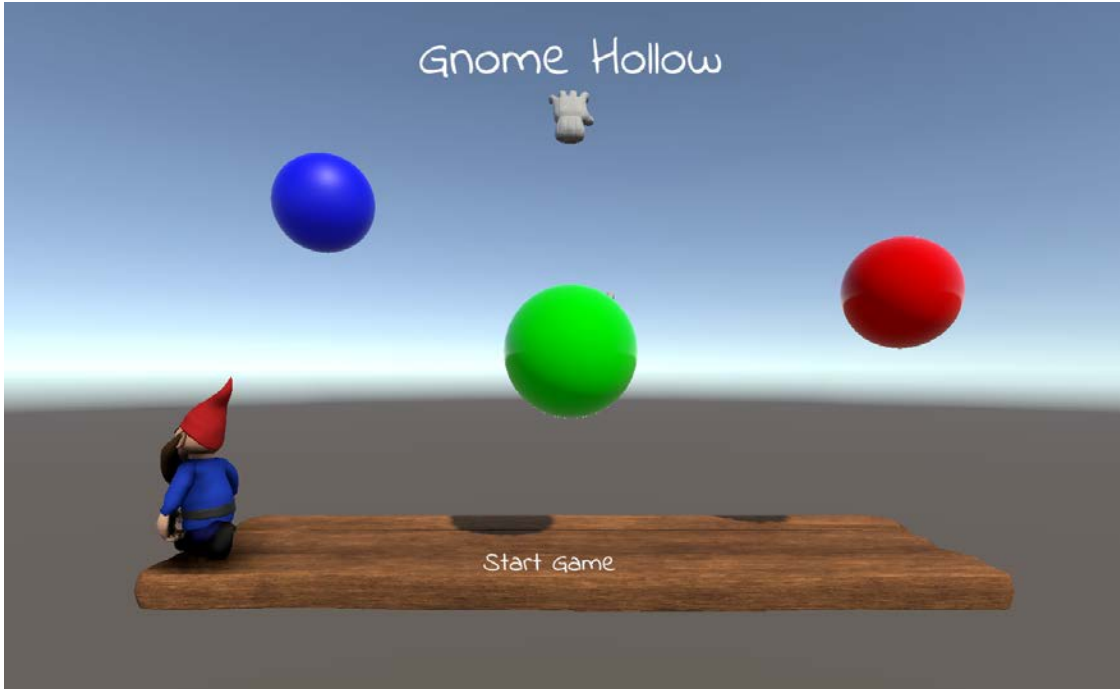


Figure 25: A prototype screenshot of the game's cut title screen

Our start and quit buttons are fairly self-explanatory. Start begins the game, while quit closes the application. The menu button is a bit more complicated, as it provides a few other options that enable users to customize their experience, with the biggest one being volume control. Finally, the spawn gnome button was included so that gnomes could be spawned in the game, allowing players to experiment with their behavior in an unstructured play environment. It could also provide something to do in the event that a player is waiting for somebody else to join. Information for what each button does is displayed at the bottom of the screen so users are aware of what each option they click on does.

Our tutorial scene serves as a tutorial video with control tips afterwards. Upon bootup, it plays a short 15-second instructional video showing players how to water plants, the most important action in the game. Afterwards, an image is shown that displays the general controls of the game, along with some tips for how to play.



Figure 26: Gnome Hollow's tutorial video playing in-engine

The main way UI is utilized in the core game is in the form of tooltips. On the top of the screen, context-sensitive notifications are given that can better inform the player of what is going on. This can include the player getting a new seed bag, an in-game event happening, or critical in-game items spawning.



Figure 27: A contextual prompt on the top of the screen to remind players to keep their plants watered

On the other hand, the bottom of the screen featured brief tips for how to use the various tools at the player's disposal. This can serve as extra instruction for the players beyond our tutorial scene before the game begins. These tooltips can be brought up by pressing the top face button.



Figure 28: A tooltip for the fertilizer bag being prompted on the bottom half of the screen

6.4. Plant Implementation

We approached implementing plants by designating a base class that all plants would follow, and then applying more specific methods and statistics depending on the plant in question for each plant subclass.

Our BasePlant class had several attributes that would carry over to all subsequent plants derived from it. The most prominent of these was a growing system that allows each plant to know its current growth status. The growth status is controlled by an enum with the following states:

Plant State	Status
Default	Debugging state, used in case assignment fails.
Seeded	The initial state of the plant. It is in its seed form. When it is watered, it will transition to the Watered state. If it interacts with the fertilizer, (scrapped) it will transition to the Fertilized state.
Fertilized	A scrapped state that could tell whether the plant was fertilized. If the plant is fertilized, it grows faster in the Watered state. This was removed from the game, as plant growth speed was not a meaningful decision in the grand scheme of things and it was more worthwhile to utilize the dirt mound idea we had in the final game.
Watered	A state that tells when the plant is watered. In this state, it grows from a seed to a full plant. It enters the Full Bloom state as soon as it finishes growing.
FullBloom	The plant is fully grown here, and will idle for as long as it is not touched. If it is not watered for long enough, it will enter the Withered state.
Withered	The plant is withered due to a lack of attention from the player. Fortunately, it is fairly easy to manage. If the player waters the plant again, it'll perk right back up and enter the Full Bloom state again.

Table 2: All possible states a plant can have

The BasePlant class also featured several parameters related to art assets that gives each plant an easy editor reference to place art assets in. This includes a sound object with unique audio for each plant, a particle system that allows each plant to own its unique set of particles, a mesh renderer that allows us to tweak the color of each plant, and a plant life timer that detects how close a plant is to wilting. In addition, abstract functions were used in order to create a set of sharable methods among all plants that can easily be called in external scripts related to certain actions taken using the appropriate plants. These include picking the plant up, dropping it, clicking on the plant, and gnome actions. We also randomized the initial size and rotation of the plant by a small amount in order to make plants feel more varied and make the garden aesthetics more interesting.

The most basic of our plants were the various flowers. These involved the sunflowers, tulips, and bushes. These only served aesthetic functions when planted, giving the players visual customization over their garden without plants necessarily needing to serve specific functions. This fits into our goal of giving players both aesthetic and mechanical customization over their garden.

Our first plants that affected the gnomes directly were the variety of berry bushes. Each of these bushes had berries that affect the gnomes' properties in different ways, with there being standard berries, fireberries, and glacier berries. The standard berry slows the gnome down,

which can be used in order to keep gnomes in one place longer more efficiently. The fireberry, on the other hand, doubles the gnome's speed, which can be used for silly interactions involving the gnome moving at ludicrously fast speed. Finally, the glacier berry enables the gnomes to water plants they pass, allowing for players to both efficiently maintain their garden and quickly grow a series of plants.

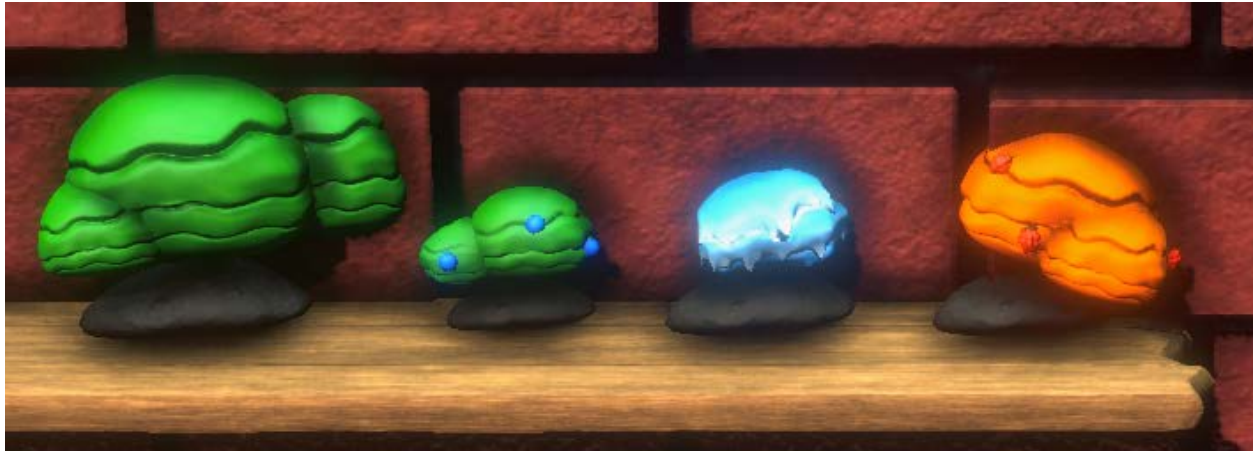


Figure 29: All bushes in Gnome Hollow. From left to right: Bush, Berry Bush, Glacier Bush, and Fireberry Bush.

A cut plant idea we had was a gnome plant. This would have been the only way a player could get more gnomes into their game, making it a fairly important pickup that players would have to pay attention to. When watered, the gnome plant would grow into a plant that promptly releases a gnome onto the playing field. It had fairly limited interactions other than adding more gnomes aside from releasing particles when clicked. More information about the gnomes can be found in Section 6.5.

Plants that affect the gnomes' state in more chaotic ways were the mushrooms and the cacti. Both launch the gnomes immediately upon contact, however their means of launching the gnomes differ slightly. The mushroom simply launches the gnome upwards, enabling it to repeatedly bounce on said mushroom. On the other hand, the cactus launches the gnome in the opposite direction from which it collided into it. For example, if the gnome collides with the cactus at a 45 degree angle, it will be bounced back at that 45 degree angle. We decided to make these physics-oriented plants because they would result in easily amusing interactions that could motivate players to experiment with other gnome plant collisions more.



Figure 30: All launching plants in Gnome Hollow. From left to right: Cactus, Mushroom

Another cut plant idea was the vine. This would have been used as a means of enabling the gnome to travel vertically, letting it get to a top platform by climbing up a vine draping off of it. This ended up being scrapped due to the complexity of moving a gnome across the vine as well as having to redefine existing systems to enable it to work out, with the vine being purely aesthetic in the Showfest build of the game.

6.5. Gnome Implementation

The gnome was one of our more significant game elements, being an entity that featured both autonomous pathfinding capabilities and unique plant interactions. Many of these interactions are described in the Plant Implementation section.

Gnomes can be picked up with the hand tool, with its position being locked to that of the hand. This will only be released when the button is pressed again. This process was described slightly more in-depth in Section 6.2.2. Gnomes also have several unique interactions with plants, which are described more in the Plant Implementation (6.4) section of the paper.

The gnome moves around the world using pathfinding AI. We decided to use Unity's default pathfinding mesh due to its simplicity with getting us fluid movement results in a 3D environment. We initially used Aronganbarg's Astar Pathfinding Project, (Granberg) but we ran into some implementation issues when converting to a 3D environment that made us gravitate towards Unity's solution instead.

Assuming no interruptions, the gnome's AI perpetually makes them go back and forth between two points when on a platform: one on the very left and one on the very right. It will not deviate from this behavior, and if you pick it up and drop it into another platform, it will inherit the points of that platform in its pathfinding. This was a fairly simple way to do gnome walking AI that was also easily iterable in the event that we had time to expand upon the AI mechanics. Aside from this back and forth behavior, in the event that a gnome ran into another gnome, both would immediately walk in the opposite direction. This was implemented to prevent gnomes

from getting stuck on each other in the case that multiple gnomes occupied the same platform, continuing their routine albeit in a slightly more limited fashion.

6.6. Game Structure

When playing *Gnome Hollow*, players initially have a very limited set of options as for what they can plant with said options growing the more they play the game.

As the players grow more plants to full bloom, seed pickups start being dispensed into the world. If the seed player picks up the seed, it is added to their collection of seeds that can be cycled through on the appropriate button tab. This gives the players more options as for what to plant as the game goes on so they are not overwhelmed by a long list of options right at the start. In addition, the plants one can obtain are randomized from a set pool, which can lead to players having slightly different experiences every time they play depending on the order of plants they roll. It is possible for players to achieve all plants in every run they make, but they'll always get certain plants before others.

One cut idea we had was the level design also expanding over time. We planned for a series of walls that drop away the more plants the player places, allowing the terrain of the level to expand alongside the players' toolkit of plants. We tried to deliberately frame wall drops so that players have simple geometry to start out but get more complex geometry over time, which can be seen in Figure 31.

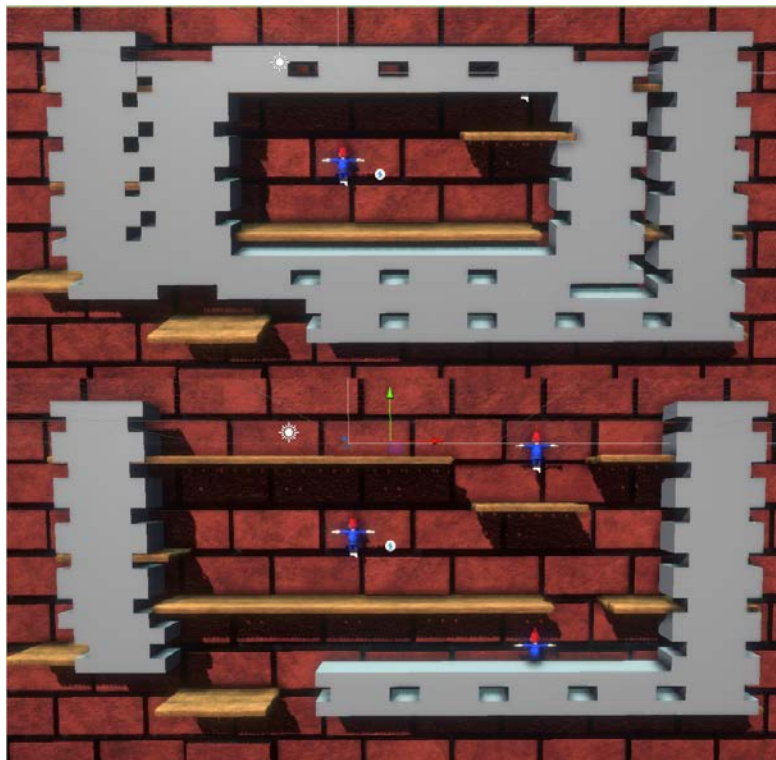


Figure 31: A prototype of the scrapped falling bricks. The top and bottom parts of the game would have been framed as stages 1 and 2 of the game respectively, with stage 3 being completely open.

6.6.1. Time Limit

Our game was initially implemented with a time limit of three minutes. This was implemented as a means to keep things moving in the case there were big venues of people playing *Gnome Hollow*, as no time limit meant that players could potentially hog the game for a while and make it hard for several people to get in on the action.

As development proceeded, both advisors and students began to question this idea, as it ran counter to our experience goal of creating a laid-back and relaxing garden. In conjunction with playtesters not seeming to interact with the game more meaningfully because of the time limit, we decided to scrap it entirely. This allowed players to freely engage with *Gnome Hollow* with their own terms without the time limit impacting their experience.

6.7. Unity-Specific Technical Features

One Unity specific feature we used is scriptable objects. These are scriptable data containers Unity provides, and can be excellent for abstracting object data such as public prefabs in order to make them more easily swappable. Scriptable objects were primarily used for audio files, as all objects using audio take their sound effects from an object containing audio files for the appropriate sound effects. This makes implementation and swapping way easier on the audio designer's end, as they don't have to wade through a potentially confusing tech stack in order to replace important sound effects. Scriptable objects for audio were created for the player, for the gnomes, and for the plants to help ease audio production.

Our plants and players used a design pattern where all player tools/needed plant assets were already loaded in the prefab, but were programmed as deactivated children. Whenever they were needed, that object would be "swapped to" in a process that involved deactivating the currently active child and activating the one that needed to be swapped in. For the players, this methodology was used to swap between tools, and for the plants, this was used to switch between seeds and grown plants upon said seeds being watered.

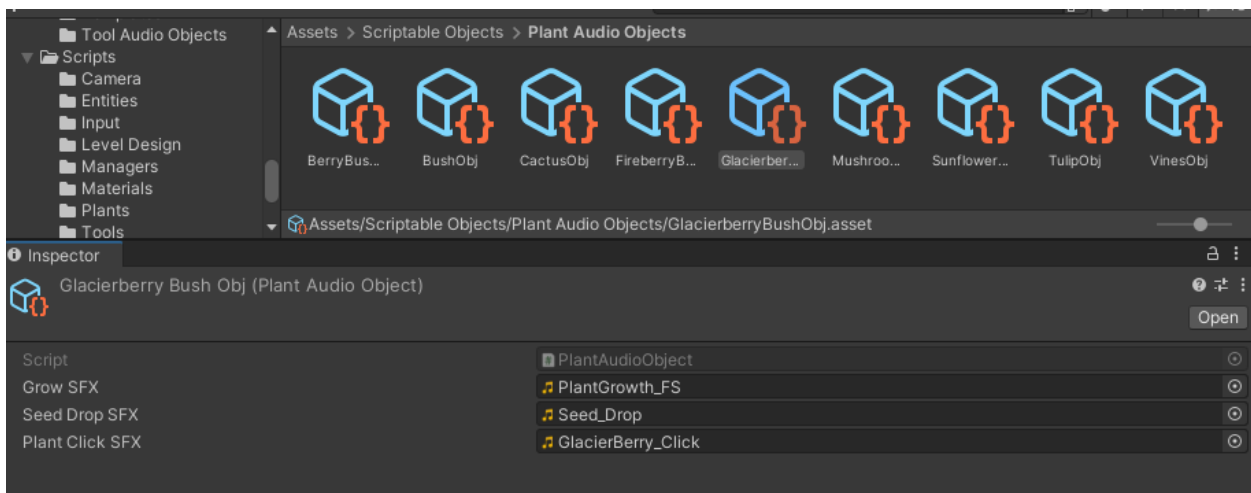


Figure 32: All plant audio scriptable objects. These can very easily be inserted into any plant and then be swapped out, making it trivial for the entire team to swap in sound effects.

7. Audio Development

Audio and music are integral to shaping a game's atmosphere and providing auditory feedback. All of the sounds were created by the audio developer, Alan Roush, except for the plant growth sound. The ambience and soundtrack were both created for *Gnome Hollow* by our audio developer as well.

7.1. Challenges

The main challenge that was brought on by projection was balancing for speakers. Typically speaking, audio design is done while wearing headphones. However, due to the scale of our game and the fact that it was being projected, players would have to hear audio via speakers. Thus, the sound had to be mixed on speakers rather than headphones. Additionally, we had to obtain speakers. We reached out to Lens and Lights, the on-campus audiovisual club, and asked about borrowing speakers to present.

7.2. Ambience

The ambience plays pockets of sound, rather than being continual. The ambience was created using recordings of a variety of birds chirping, along with occasional gnome sounds. There were three different bird chirping clips, which were scattered throughout. Some of these clips were then pitched up to provide more variation. The bird chirps were recorded on an iPhone, and some of these clips were edited using equalization, compression, and pitch shifting these chirps upward. The gnome sounds vary, with there being a giggle sound, a "whee!" sound, and a "yippee!" sound. These sounds were pitched upward to maintain the high-pitched gnome aesthetic.

7.3. Tools

All of the tool sounds were designed using foley sound design, meaning they were made from scratch, and they were edited in Reaper, a Digital Audio Workstation (DAW). We would record multiple takes and variations for each sound, and then choose the best and edit them using a Zoom H4N microphone, unless otherwise noted. An example of this foley-based sound design is the watering can: a pop sound was created with the mouth, and then was repitched and had its speed changed in order to create a cartoon-ish watering sound. In terms of choosing DAWs, Ableton Live, a DAW mainly used for music production, and Reaper were selected based upon Alan's familiarity with the programs. There were a total of 6 tools: watering can, shears, glove, seed bag, fertilizer bag, and shovel. For each tool, there were two different kinds of sound effects: selection and usage. The selection sound effects played when a player switches to a new tool, and the usage sound effects play when the player uses a tool. These sounds were frequently edited with different effects via free plugins in Reaper (such as EQ, ReaFIR, pitch shift, compression, and more). The ReaFIR (subtract) plugin was used frequently, which uses a fast fourier transformation equalization to remove the unwanted lower background frequencies. To do so, we recorded "dead air" with no sounds before recording our desired sound. Then, looped

the section with “dead air”, and used it as a sample for the ReaFIR equalization. Equalization refers to the usage of filters to remove unwanted frequencies. Compression refers to the process in which lower and higher frequencies are attenuated in order to make the overall volume of the track more consistent.

Initially, all of the tool usage sound effects were continuous clips. However, our advisors noted that this wouldn’t work if the tool was used for longer than the sound clip, and they recommended having tool sounds broken into three different clips: a beginning of the sound, the middle of the sound which would loop, and the end of the sound. All of the tool sounds were then converted into this format. Beginning, middle, and end sounds ended up being used for the fertilizer bag, watering can, and shears. This type of sound clip was utilized because the actions of these tools were continuous; for example, the watering has beginning, middle, and end clips because the action of watering is continuous. The beginning clip plays when the tool button is first pressed. The middle clip plays when the button is held down, and this clip is what loops while a tool is being continually used. The end clip plays once the button has been released.

It was realized that using this format wouldn’t be appropriate for all of the tools, which led to another format for the tool sounds: one shots. One-shot clips were implemented for tools whose actions made sense to be a single sound clip; for example, the shovel utilizes one shot clips because when using a shovel in the game, you dig once. One-shots were implemented for the shovel, the seedbag, and the glove tool usage sounds. Additionally, all of the tool select sounds are one shots.

7.3.1. Sound Effect Creation

The creation of sound effects for *Gnome Hollow* was done using Foley sound design. These recordings were imported into Reaper, and edited using plugins to create the desired sound.

Watering Can

The sounds for the watering can tool were created by pushing air out the mouth to create a popping sound. The select sound, which plays when a user swaps over to the watering can, was a oneshot “pop” noise. In order to create such a noise, the “pop” sound was equalized, using ReaFIR and EQ to remove unwanted low and high background frequencies, respectively. The “pop” sound was then pitch shifted in order to achieve the cartoony water droplet sound that was desired. As for the creation of the watering can usage noises, a similar process was utilized. ReaFIR and EQ were both used again, and the “pop” sound was pitch shifted. The “pop” was then copied and pasted back to back, with the speed of these clips gradually increasing from 0.95 speed to 1.2 speed. This playback speed change yielded a cartoony sounding watering can sound which progressed.

Fertilizer Bag

The creation of the sounds for the fertilizer bag was straightforward. In order to create this sound, we used a box of salt and a plastic container. The salt was poured into the plastic container at different rates, which were then reviewed and the best takes were selected and edited. The fertilizer bag select sound was a oneshot, while the usage sound was cut into beginning, middle, and end clips. Unlike the other sounds, no effects were applied to the original recordings.

Glove

For the glove sounds, we decided to use velcro sounds to mimic putting on gardening gloves. Both the selection and the usage sounds for the gloves are one-shots, and were created by recording velcro being pulled apart. The usage sounds were not edited with effects, but the select sound was pitch shifted downwards to create a distinct sound as well as to differentiate it from the usage one-shots.

Shears

The shears sounds were created by recording the opening and closing of a pair of scissors. Both the selection and usage sounds utilized a compressor and the ReaFIR plugin. The compressor was used to make the sounds louder, while the ReaFIR plugin was used to remove unwanted low background frequencies

Shovel

In order to create the shovel sounds, we used a spoon and salt. We dug into a plastic container full of salt with a metal spoon in order to replicate the sound of a shovel digging into dirt. Both the selection sounds as well as the usage sound were one-shots. The selection sound was pitched down, in addition to one of the usage one-shots. All three usage one-shots utilized the ReaFIR plugin to remove low frequency background noises.

Seedbag

The seedbag had three different one-shot sounds: a selection sound, a usage sound, and a sound that played when the player swaps between different seed bags. The seedbag select sound was created by making a “pop” mouth sound, similar to the water droplet, except that this time, it was pitch shifted down 10 semitones. As for the creation of the seedbag change sound, it was created by turning the dial on a washing machine. This sound utilized the ReaFIR plugin again. Finally, the usage sound for the seedbag was created by Alan clicking his tongue. This clip was then slowed down to half speed, was pitched up, and also used ReaFIR to remove the background noises.

7.4. Sound Effects

The sound effects for *Gnome Hollow* refer to both sound effects for the plants as well as the gnome sound effects. These sound effects were created using both foley as well as Ableton Live, which was used to create some of the particle sound effects as well as the game soundtrack. For the creation of the gnome grunts, we recorded grunting and edited it using Reaper. The only plugin used to edit both of the grunt sounds was ReaFIR, in order to remove background noise. The creation of the gnome squeal was made by recording a toilet paper dispenser with an iPhone, and then was equalized using a low shelf to remove any low frequency background ambience. The result was a high-pitched squeal which matches the aesthetic of gnomes and their higher-pitched voices. The sound of the gnome bouncing, which plays when the gnome steps on a mushroom, was created by stretching a rubber band and then flicking it. It was created by overlaying two of these flicks in order to create a cartoonish bouncing sound. The effects used to process the original recording were ReaFIR as well as a slight pitch shift down. As for the gnome exclamation sound, Alan recorded himself saying “whee!” and pitched it up to match the higher-pitch gnome voice. Alan recorded himself laughing at a higher pitch, and then pitched that recording up again in order to once again match the higher-pitch of a gnome voice.

There are also sound effects for the plants that play when a player clicks on a plant with the glove. These sound effects play alongside the visual particle effects, and traits of the plants were used to help guide the sound design of their respective plant click sounds. Most of these effects were made in Ableton, but two of them, the fireberry bush and the glacier bush, were made using foley sound design. The sound for the fireberry bush resembles a fire crackling. This was made by crumpling up a plastic bag and recording it. Additionally, the initial “ignition” sound at the beginning of this effect was made by blowing into the mic, which was then slightly pitch shifted up. Both of these clips were edited with compression, in order to make them louder. The resulting sound is a “woosh” to signify the fire being started, followed by the sound of a fire crackling.

The sound effect for the glacier berry resembles a winter wind blowing. The sound itself is four different clips of Alan blowing into a Zoom H4N slowed down to 0.8 times speed and overlaid together. Three of these clips are pitch shifted down and one is pitched slightly up. This resulting sound effect is a whistling winter wind.

As mentioned previously, the rest of the plant click sounds were made in Ableton Live. Timbre, or the unique sound an instrument produces, will be referenced throughout the section. The berry bush click sound is a three note clip, starting at G#3, then F#3 and ending back on G#3. The instrument playing is the “Basic Bells” instrument in Ableton, which we felt fit the berry bush. The bush sound is also a three note clip, which starts at D3, then A2, then C3. The instrument is “Choir Stream”, which is a basic choir instrument. This works well for the bush because both are relatively basic assets. As for the cactus, it felt appropriate to have a quick, “sharp” sound since cacti have sharp spines. The instrument for the cactus was “Cello Section Staccato”, which fits the earlier description of a quick, “sharp” sound. The notes played are C3, D#3, and F3. Unlike the other musical plant click sounds, the mushroom click sound is different

because it uses a synth pad called “Wet Air Pad” and only plays one note: C3. This instrument was selected for the mushroom because it gives off an earthy atmospheric sound, which felt appropriate for a mushroom. As for the sunflower sound, a “bright” sound was desired. Thus, the “Block Minor Brass” instrument was selected because it possesses a bright timbre. The notes played for the sunflower are F3, C3, and back to F3. The tulip sound uses another simple instrument to reflect how the tulip is a basic plant compared to the other plants. Thus, “E-piano Basic” was used for this sound. The notes used for the tulip click sound are: G#4, G4, F#4, G4 and back to G#4. The final plant click sound designed in Ableton was the vines. Congas and bongos seemed appropriate for the vines sound effect, seeing as both evoke jungle-esque imagery. The pattern for the vines started with “Conga Acoustified Low”, then “Conga Acoustified High”, and ending with “Bongo High Open Soft”.

One of the plant sounds that was not mentioned earlier was the plant growth sound. This is the only sound in the game that wasn’t designed by Alan. Instead, it was sourced from Freesound (014_PlantaCreciendo-004.wav) and the original creator was the user aabbccdee123.

7.5. Implementation

The implementation of audio assets was very straightforward. Garrett, our technician, made a framework which allowed tool sounds to be dragged and dropped into Unity. For this framework, the tool and plant click sounds were loaded into scriptable objects (see Figure 33), making it easy to edit. These sounds were wired into game objects which are called by the program when the associated event occurs. As for the implementation of the soundtrack and ambience, two audio sources were added near the camera. These audio sources play for the entire duration of a session.

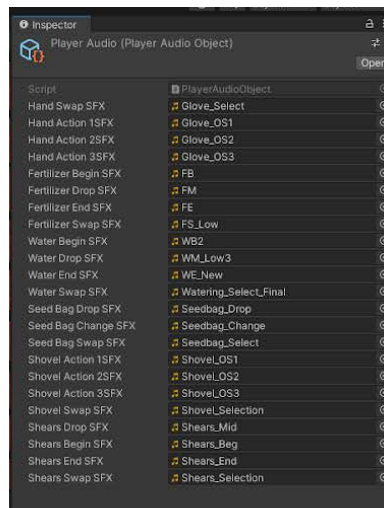


Figure 33: The scriptable object for tool sound effects

7.6. Soundtrack

A soundtrack is a core element for a video game. It provides the player with an atmosphere and influences their mood throughout gameplay. Thus, it was important for us to design and create a soundtrack which reflected the calm and upbeat atmosphere of *Gnome Hollow*.

7.6.1. Melody

The biggest challenge posed to Alan was the soundtrack. We wanted to create a soundtrack which reflected the calm, repetitive gameplay of *Gnome Hollow*. Initially, he struggled to create a melody, but then drew inspiration from the *Plants Vs. Zombies* soundtrack. This initial version was not in a specific key, used 1/16 notes, and was split into three different sections: Part 1, Part 2, and a variation of part 1 (see Figure 34). The variation of Part 1 was an octave higher, as well as adding four more notes: C#4, D#4, E4, G4. Part 1, which is shown in Figure 35, was repeated three times before going into Part 2, seen in Figure 36. Part 2 would also play three times before transitioning into the variation of Part 1. The instrument that was used for this initial melody was “Basic Flange Harp”.



Figure 34: Part 1 of initial melody

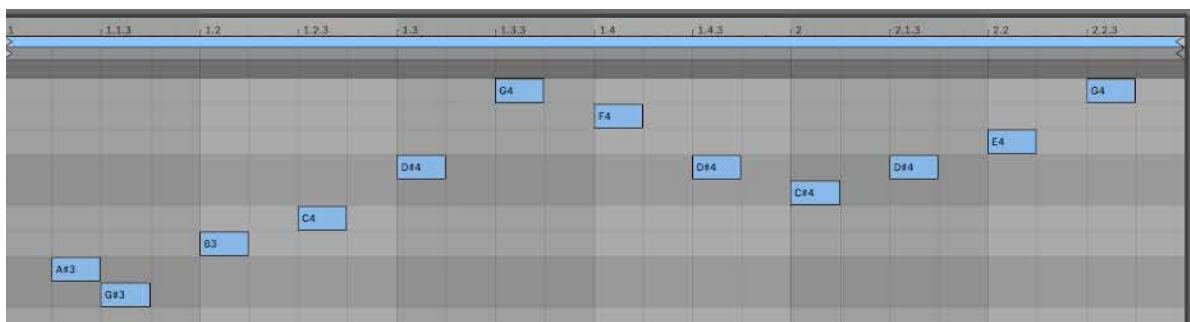


Figure 35: Variation of part 1 for initial melody

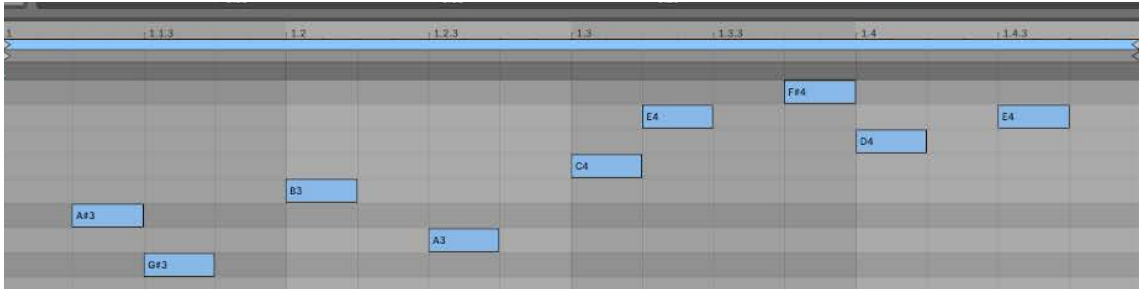


Figure 36: Part 2 of initial melody

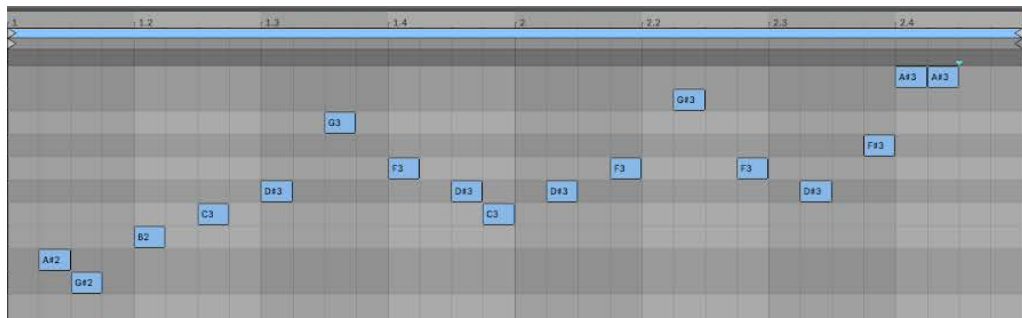


Figure 37: Exploring options

Because just these three sections were short and rather repetitive, Alan explored adding more notes (see Figure 37) to the first part of the initial melody, but this ended up sounding unpleasant and was scrapped. The initial version of the soundtrack’s melody was eerie and did not reflect the relaxing atmosphere of the game. This was due to the timbre of the “Basic Flange Harp” being too harsh and creepy, so it was switched to a different harp, “Harp Soft”, which had a softer timbre.

Despite being scrapped, the initial melody set the blueprint for the final soundtrack; Alan kept the repetitiveness aspect but added more variation. Alan definitely struggled at first in creating a calming and upbeat melody, but was assisted by his music professor at the time, Professor David Ibbett. Professor Ibbett assisted him in defining the key of the melody as well as provided general feedback and possible chord progressions. The key of the melody, E Major, was defined in the next major iteration along with adding more instruments. Additionally, chords and chord progressions were used, rather than just playing notes. The song starts with 1/16 notes playing the chord progression I-IV-V, then plays the chords in the following order: ii, vi, ii, vii°, V, vi (see Figure 38). This was repeated four times and then the chord progression I-IV-V is played another time, before being followed by the same aforementioned chords. This played only twice before a new element is added at the end: the chords IV and V (see Figure 39). The newly augmented section plays thrice before being shifted up an octave. This octave-shifted section played five times before another new section was played (See Figure 40). Within this new section, the double I-IV-V in the beginning and the IV and V chords at the end were removed (Figure 41). This new section played six times before the main melody comes in.

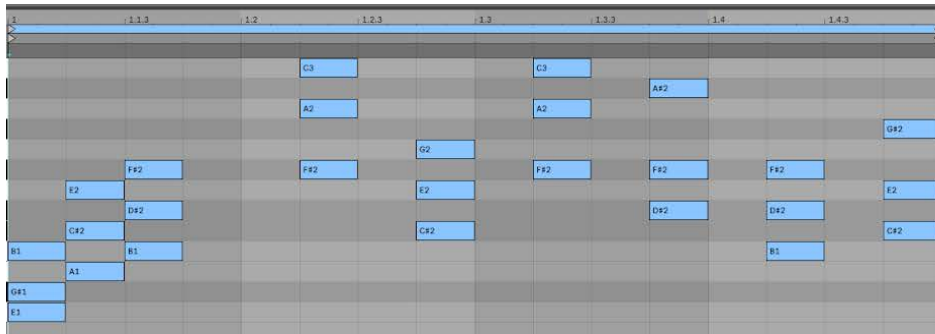


Figure 38: Introduction of soundtrack

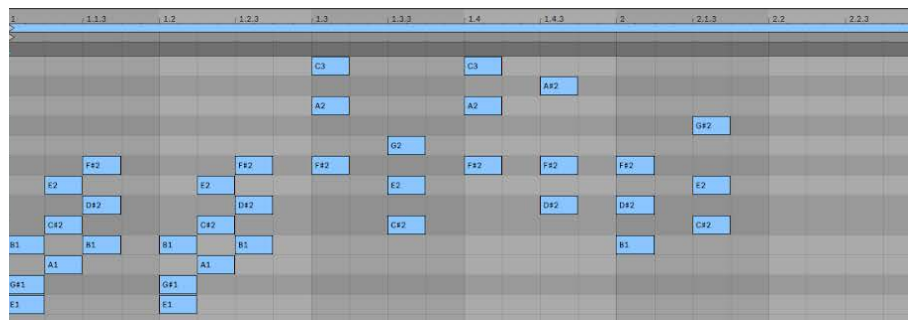


Figure 39: Introduction part 2

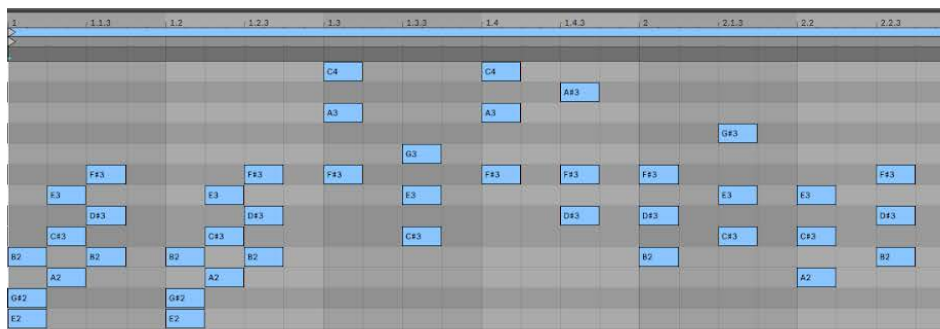


Figure 40: Octave shifted chord progression

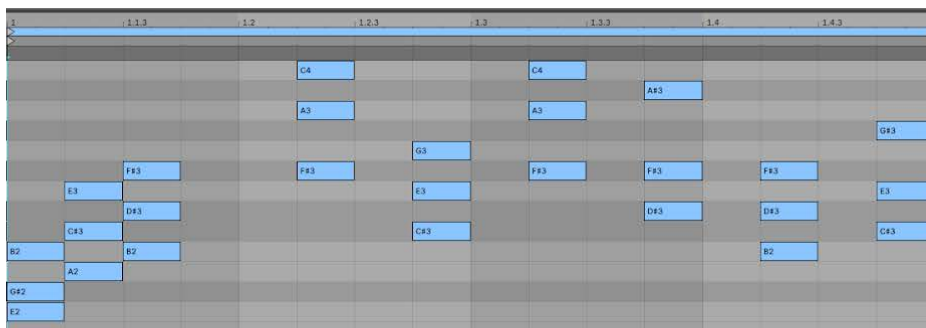


Figure 41: Chord progression from just before the main melody

Similar to the player's garden, the soundtrack becomes more complex over time. In the beginning, there was a lot of repetition. However, as the main melody comes into earshot, it was still somewhat repetitive, which reflected the gameplay, but still had variation. The main melody used $\frac{1}{4}$ notes and began with the chord progression vi-IV-V-I, repeating twice with one beat in between each progression, and with one beat at the end (see Figure 42). This transitioned into the same chord progressions being played again twice, except there was no empty beat between them; instead, the IV chord was played (see Figure 43). Following this was a section using the same chord progression, but with the beginning chord doubled up (vi-vi-IV-V-I) (see Figure 44). Each time this chord progression played, another copy of a chord was added until there were two of each chord (vi-vi-IV-IV-V-V-I-I). This is referred to as the main doubled melody (see Figure 45). This transitioned into the next section which played the doubled up chord progression six times without any empty beats in between, before moving into a new chord progression.

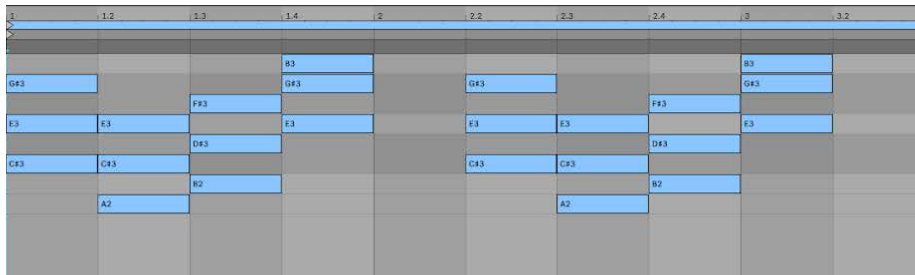


Figure 42: Main melody

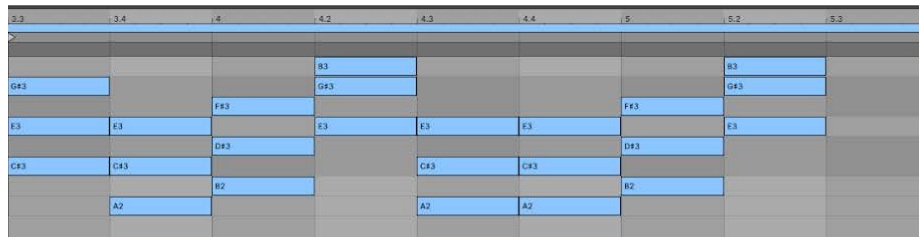


Figure 43: Main melody with variation

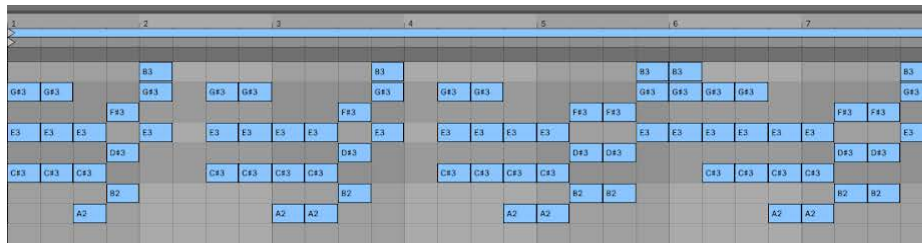


Figure 44: Main melody with progressive doubling

had a light, relaxing timbre that synergized well with the harp. The bassline was not present throughout the entire soundtrack; it began when the main melody began and was not present when the variation of the new melody played as well as when the new melody played for the last time. The bassline highlighted the roots, or in other words, the beginning key of the chord, of the harp chords. The volume of the bassline was automated and started off at negative infinity, meaning it cannot be heard. As the main melody was playing, the volume slowly automated to a level at which the bass can be heard. Additionally, the “Choir Reverb” instrument is used to highlight the root chords of the harp. The timbre of this instrument was light and airy and played twice for the main melody, stopped, played until the main doubled melody played for the last time, stopped, and then played again when the variation of the new melody plays. The chimes had an ethereal timbre and played whenever an E note was played, which provided an upbeat yet relaxing sound.

7.6.3. Extra Layers

A feature which was proposed by the advisors was to have the soundtrack evolve as the players planted more plants. Every 5 plants unlocked a new layer, for a total of 20 plants unlocking 4 extra layers of instrumentation. Similar to the bassline, these layers also highlighted the root of chords from the harp. The instruments used are: “Long Hollow - Brass Quartet”, “Island Bells”, “Attack Key” and “Flute Mellow”. These 4 layers were all muted at the beginning of a session, and would gain enough volume to become in earshot once unlocked via planting.

8. Playtesting

As we developed our game, we wanted to ensure that we were testing and receiving player feedback on the project as often as we could. Since our game was meant to be playable by any student on campus, we were eager to hear from various perspectives about what the game should include, what it should look like, and how it should play.

8.1. Testing Methodology

Our team recognized the importance of conducting playtesting sessions and receiving feedback from players when it came to improving our game, so we made sure we were testing as often as possible. When we had the opportunity for students to playtest our game, we made sure to follow some simple methods that would give us the best possible feedback. Our game appears small on a typical laptop screen, so we always showcased our game on a large, projected screen. This presented us with additional useful information, as we were able to gauge people's reaction to seeing a large-scale game and ask them about what caught their attention. We made sure to push the attention-grabbing aspects further, as we always wanted the game to look appealing and interesting to passersby. We spoke with players as they played to understand what they were thinking and allowing them to voice their questions, concerns, and real-time feedback.

After players had tried the game, we made sure our feedback forms would be effective in giving our team the best idea of what direction to take. Our forms asked about the art, animations, gameplay, experience, audio, and more. We needed to ask about the scaling and colors of the game as well, as having the game projected meant a loss of color and a different size of the objects on screen. For the questions themselves, we mostly asked testers to respond on a linear scale of 1-5. We felt that this would give us the most clear feedback, as we would style the questions for the values to correlate with how strongly the player felt about the topic. For instance, when we asked the question "How did the speed of the cursor controls feel?", we would have a response of 1 tied to "Too slow" and 5 meaning "Too fast". This allowed players to give more nuanced answers, such as a response of 2 implying "Slightly too slow". We could use the distribution of answers to get an understanding of what the average player felt for each question. In addition to these style of questions, we included open ended short answer questions after each section. Players could give their thoughts on the audio, the art, the gameplay, and other aspects each in their own dedicated text box. These guided short answer questions gave us more detailed feedback than one question at the end asking for any additional input without specifying the topic. We also wrote our forms to have one final text box where players could add any input they wanted, from gameplay suggestions to plant ideas and more. With these methods, we could collect information from many unique student perspectives and tailor our game to have the widest appeal possible.

8.2. Testing Results

Our testing mostly took place during B, C, and D-Term, as we were still designing our game concept and doing projection research for the majority of A-Term. Our testing sessions

were a combination of WPI organized events, such as ProtoFest and AlphaFest, and our own personal sessions that we set up and scheduled independently.

8.2.1. Protofest

The first time we were able to showcase our game was ProtoFest, a first-time event at WPI. The ProtoFest took place in early B-Term on October 26th, 2022 and was for MQPs to display very early concepts and prototypes to receive feedback from peers.

Game State

At this early stage in development, our game was incredibly simplistic. The only models in the game were the gnome and some of the tools. Everything else was a basic cube. Our level design consisted of two shelves at different heights, with the camera perspective being head on, so the game looked 2D without any depth. Player roles were not split up, so each player could use any of the tools independent of each other. The basic gameplay included planting a seed anywhere on a shelf, and watering it to “grow” it, which changed the cube’s color and physics. Players could spawn gnomes and move anything on screen around to wherever they desired. The game was basic, but it was a serviceable prototype to give players an idea of what was to come.



Figure 48: ProtoFest build

What we Tested

With such a basic prototype, we were mostly testing the concept of our game. The main design aspects we were certain of was a cooperative game centered around making a garden. We wanted to hear from students after showing them our concept. Was the idea appealing? Should we add a goal? Do we pursue multiplayer? What should the UI look like? We kept the feedback open to allow any input from players about what they would like to see in game, from plant types to mechanics.

Feedback

Our first round of playtesting gave us a surprisingly abundant amount of feedback for how simple our game state was. One of the first things we wanted to hear from players was how

long the game would keep their attention. Even though our rounds were approximately 3 minutes long for ProtoFest, we wanted players to be invested enough that they would want to come back for another round or want to play again.

How long would you want to play this prototype?

14 responses

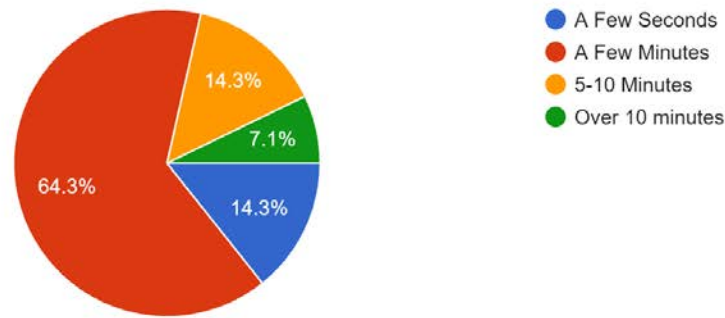


Figure 49: Results from ProtoFest question “How long would you want to play this prototype?”

Overall, players were happy to play for a few minutes, but there was still room to improve our gameplay loop as some would only want to play for seconds at a time. We made sure to ask questions about our participants’ experience while playing as well. To do so, we asked players to check off descriptions of their experience. There were a mix of positive, negative, and neutral emotions. We wanted to see that players were having a fun and relaxing time, while hopefully seeing little to no stress or frustration.

How would you describe your experience? Check all that apply

14 responses

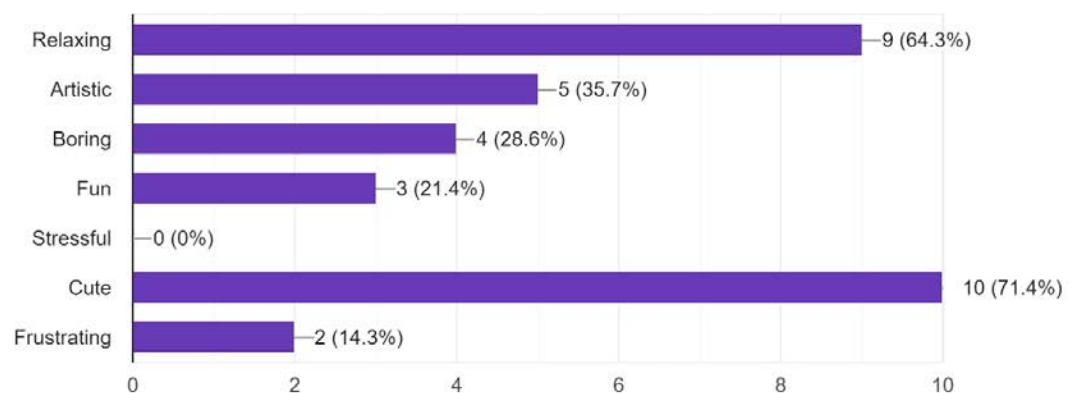


Figure 50: Results from ProtoFest question “How would you describe your experience? Check all that apply”

Even with the game in its simplest state, players were enjoying the concept and hitting our desired experience goals. There was some frustration, mostly from the latency experienced from connecting our computer to a projector wirelessly. We knew we wanted to maintain our time for holding player attention as well as our experience goals, so these questions reappeared in all playtesting forms.

Since the game was still in an early form with potential for some minor design changes, we wanted to hear from as many perspectives as possible about what players wanted from our game. We asked multiple sentence-based questions, where players would rank their interest in the proposed mechanic on a scale of 1 to 5, 1 being not interested and 5 being very interested.

An experience where you are scored based on your garden's appearance
14 responses

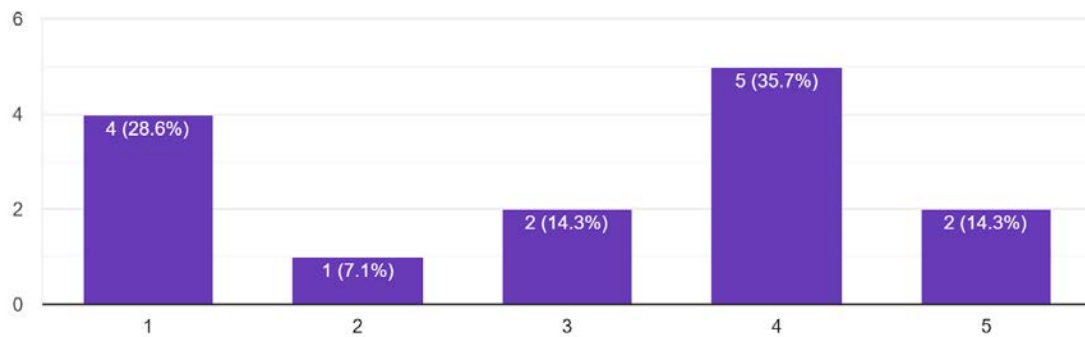


Figure 51: Results from ProtoFest interest prompt “An experience where you are scored based on your garden’s appearance”

A fast paced round where you have a small amount of time to make your garden
14 responses

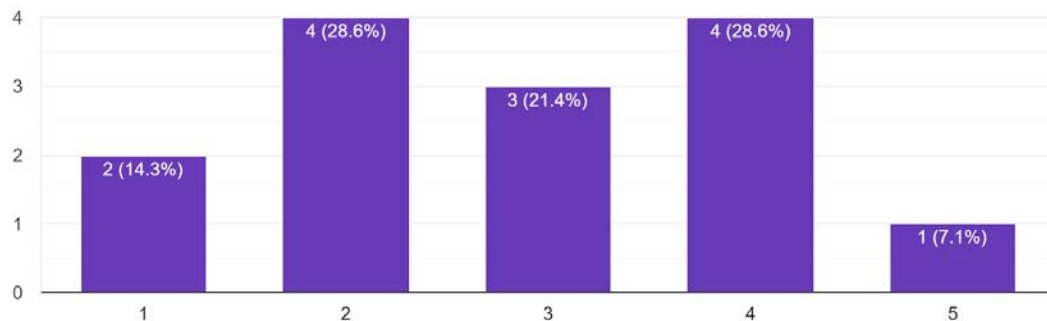


Figure 52: Results from ProtoFest interest prompt “A fast paced round where you have a small amount of time to make your garden”

The main ideas we wanted to gauge interest for were the pace of the game and the goal of the game. As seen in Figure 51, asking players if they wanted to be scored on their garden resulted in very mixed responses. While more people felt generally positive about the proposal, the second most common response was a 1, meaning absolutely not interested. We did not want to implement a mechanic that would be so divisive, as we wanted as broad of an audience as possible. The same principle goes for the next question, as seen in Figure 52. The answers are very evenly distributed, so we looked to other options instead.

A slow paced round where you have a large amount of time to make your garden

14 responses

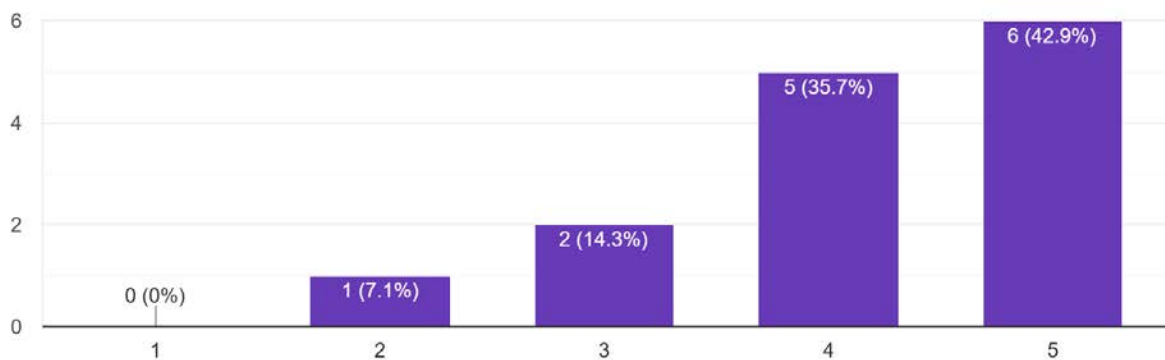


Figure 53: Results from ProtoFest interest prompt “A slow paced round where you have a large amount of time to make your garden”

When it came to asking about slower rounds, the results were much more decisive. Only one response dropped lower than a 3, and the majority ranked an interest of 5. With questions like these, ProtoFest testing helped us solidify that our game would take a slower sandbox approach with no competition or win conditions.

Changes Made

Once ProtoFest concluded, we wanted to make as many adjustments as we could before AlphaFest. There were not many major gameplay loop changes, but we kept in mind the results from this survey for the future. The first major change was altering the camera perspective. Instead of our 3D assets looking flat from a head-on view, we raised and angled our camera downwards so the tops of the shelves would become visible and there would be a sense of depth. Additionally, we wanted objects to feel like they had more mass to them, so the cursors had followthrough added. That way, players would feel the weight of the tool they were using or the object they were carrying. As for art, we began working on implementing placeholder particle

effects to be played when plants were interacted with. We also created more placeholder models and grow animations so that players would get a better sense of the game's art direction.

We received an abundance of feedback regarding what players were interested in seeing this game become. While our time before the next playtesting event was short, we did incorporate a way for tools to be split between players so that we could begin testing each role and how cooperation would work in-game. There was a plethora of information we gathered from ProtoFest, but we did not incorporate much before AlphaFest. Instead, we kept the information we had gathered from testers in mind as we continued receiving feedback.

8.2.2. AlphaFest

Just over 2 weeks later, we would show our game once again at a WPI event. AlphaFest took place on November 11th, 2022. The event was a showcase for any student projects to be displayed in their alpha builds, so there would be a much higher student attendance. In the short amount of time we had between ProtoFest and AlphaFest, we incorporated as many new changes as we could in order to get the best feedback from players.

Game State

We were able to make decent progress from prototype to alpha in the short turnaround time. All of the assets now had placeholders: We had a gnome, fertilizer bag, shovel, hammer, shears, watering can, seed bag, mushroom, and daisy. The only animations in place were a basic grow animation for the two plants. We amplified our assets by creating particle effects to add to the whimsical and visual design of the game. We also began to play with depth, as we angled the camera down to show the tops of the shelves. While gnomes had been implemented for ProtoFest, they were more dynamic for our alpha. Players were still able to spawn them, and they would move back and forth on the shelves. When they interacted with a daisy, they would activate particle effects, and when they touched a mushroom, they would bounce. We also included minimal audio to give players a better idea of what was happening on screen. The biggest change technically was trying out a split between the player's roles: Giving one person the role of using tools, while the other player controls all of the seeds and the gnome spawning. This iteration was also the introduction of a time limit in order to allow a better flow of players to test our game. We kept the time at around 3 minutes, which would prompt the player to restart once it ran out. Our game had almost all mechanics in place for our alpha, so we were ready to test just about everything.



Figure 54: AlphaFest build

What we Tested

We were able to incorporate some feedback in the weeks between ProtoFest and AlphaFest, as well as making general progress on our build. Since our concept was solidified at this point, we were able to ask more specific questions about our design decisions. We wanted player feedback on the placeholder art assets in order to get an idea of what aesthetic we could lean into. Additionally, we were testing the scale of these assets when they were presented on a projected screen. Since AlphaFest was a heavily populated event, we asked participants what drew their attention to the game. Knowing what attracted players to our game would allow us to focus on making those aspects even more prevalent in order to draw in more crowds. With AlphaFest being still early in development, we were still able to ask questions about general game design, UI, or any open ideas that players may have thought of while playing.

Feedback

The opening question for AlphaFest's form was "What (if anything) grabbed your attention about our game that made you want to play?" We wanted to know what drew players to the game, so we would be able to capitalize on the tester's answers. Our game was meant to appeal to as many people as possible, so knowing this information was vital. Overwhelmingly, the response was the silly gameplay and gnomes. Out of 27 text-based responses, 15 mentioned the gnomes specifically, and most of the additional responses centered around the cute and silly gameplay and aesthetics.

The questions about player retention time and player experience returned for AlphaFest, as we hoped to see some improvement with the changes we had made. For playtime, we saw a major improvement. No players wished to spend only a few seconds with the game, and there was a near even split between wanting to play for a few minutes and wanting to play for 5 or more minutes.

How long would you want to play this game?

27 responses

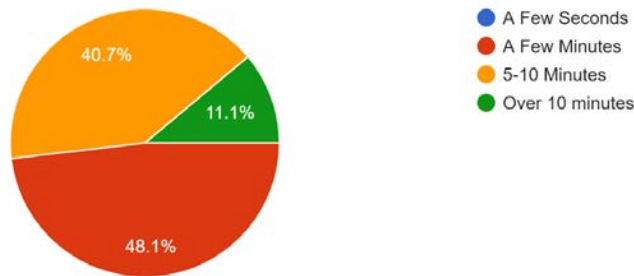


Figure 55: Results from AlphaFest question “How long would you want to play this prototype?”

Our gameplay loop had improved, as players were looking to play longer than before. As for player experience, we saw a somewhat similar distribution to our results from ProtoFest.

How would you describe your experience? Check all that apply

27 responses

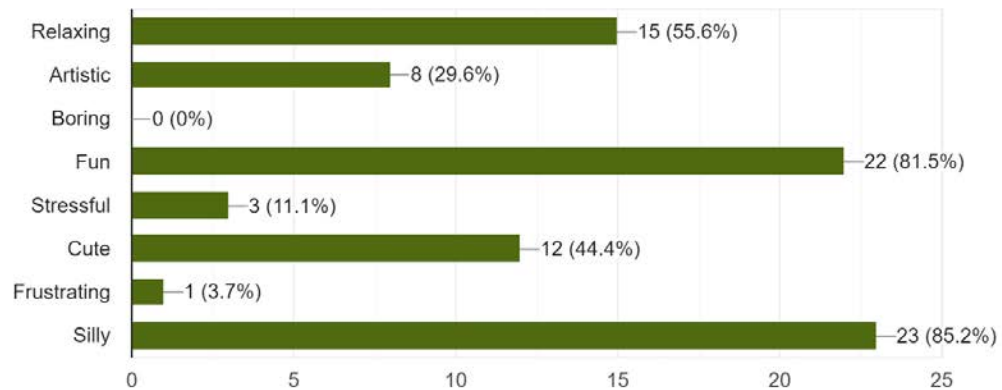


Figure 56: Results from AlphaFest question “How would you describe your experience? Check all that apply”

We saw a favorable distribution of players describing their time playing *Gnome Hollow* as silly, fun, and relaxing. There were a few responses indicating that some parts of gameplay caused stress and frustration, so we continued to work towards eliminating those feelings in players.

While we had always intended to incorporate co-op into our game, we wanted to make sure that players were enjoying the way we implemented the two player gameplay. There was still time to change direction if needed, so we made sure to receive feedback on our multiplayer gameplay.

Did you enjoy the multiplayer aspect of the game?

27 responses

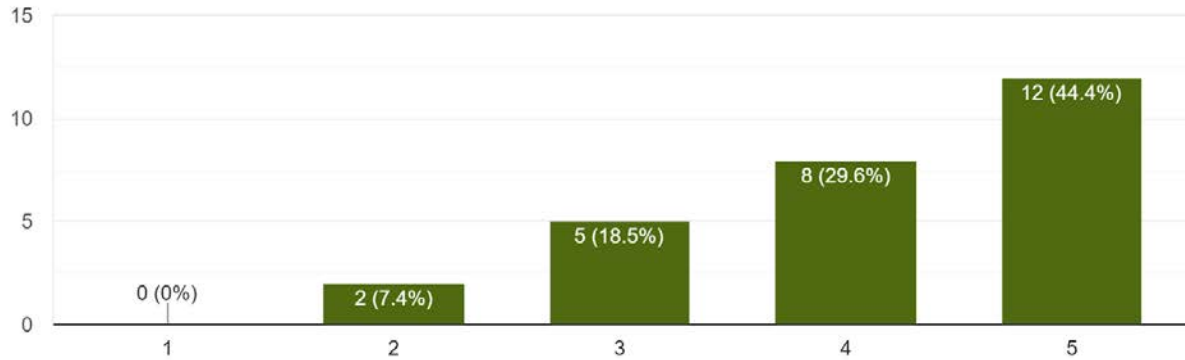


Figure 57: Results from AlphaFest question “Did you enjoy the multiplayer aspect of the game?”

Players had a mostly positive response to the multiplayer in *Gnome Hollow*, so we continued moving forward with the cooperative style of gameplay in mind.

With AlphaFest came a new set of questions based around the look of our game on a projector. Since most of our experience with the game was on smaller computer screens, it was hard to judge how scale would translate to large projected images. We wanted to determine what felt good for both the colors that we used for the materials and the scale of the models on screen.

How would you rate the scale of the objects on the screen?

27 responses

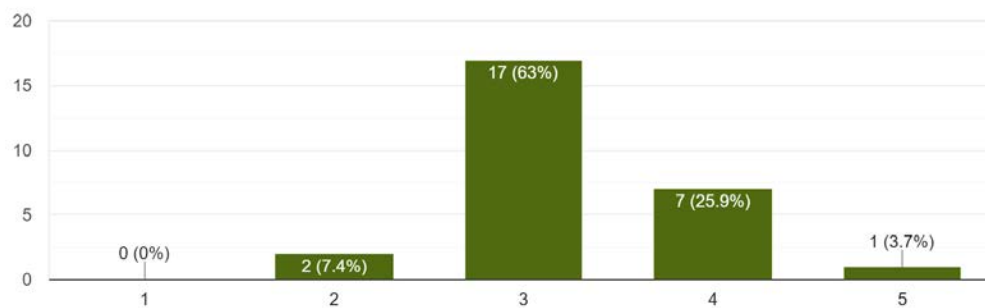


Figure 58: Results from AlphaFest question “How would you rate the scale of the objects on the screen?”

Luckily for our first test, we landed on the right scale. A majority of players answered 3, which is exactly between the extremes of too small and too large. Knowing what a good scale for

playability was early on allowed us to use the objects we had in the scene as reference for later assets.

With multiplayer a core aspect of our game that we solidified during this survey, we wanted to know what type of multiplayer would interest the most players for our gameplay loop. The game could have a mechanic where both players are capable of performing the same actions with the same set of tools, or it could be structured so each player has a unique set of tools that are used for different purposes. We wanted to know which would be more appealing to players, so we asked if cooperation should be necessary to play the game.

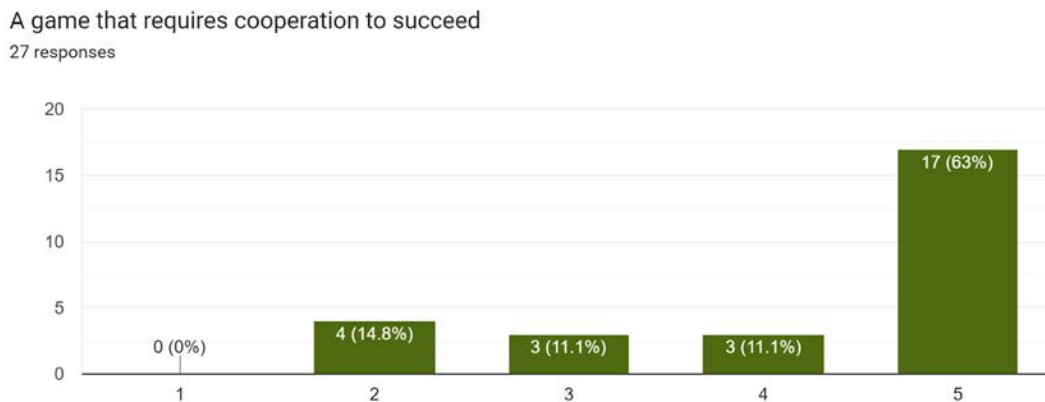


Figure 59: Results from AlphaFest interest prompt “A game that requires cooperation to succeed”

Once again, we saw a landslide in our results. It was clear that players wanted to see a game where they needed to work with their partner to succeed, so we made sure to keep this fact in mind when continuing to design our gameplay loop. AlphaFest helped us get a clear understanding of how players viewed our game and what they wished to see from the gameplay loop.

Changes Made

AlphaFest was pivotal for our development pipeline. After hearing from all of our playtesters, we had a clear sense of what our players wanted their experience to look like. We continued to tweak the way player tools were split, as well as the way players would need to interact with each other to accomplish their goal. With the general mechanics of two player cooperation established, we moved into art production. From here on out, the biggest changes came from art. Our team began creating the final assets for plants and tools, as well as their animations and particle effects. We also began working on creating a more dynamic and fun level design.

8.2.3. C-Term Playtesting

After getting substantial feedback from the previous term, we went into C Term focusing heavily on production. Our work became split between art, audio, and tech. Once we were able to merge our work back together for the first steps of implementation, we wanted to test the current state of our game with the placeholder assets replaced with their final models. We scheduled a large lecture hall with a projector and sent out an email to all students asking for their participation in our playtesting session. We had the room for 4 hours, with sessions being 15 minutes long and two people signed up for each time slot. We wanted to ensure that the game was played with two students instead of one student and one of us, as it gave us a better idea of how the game would play given we were not there to guide them.

Game State

Replacing placeholders with final assets was the major change for this iteration of our game. All of the plant, tool, and gnome models were implemented with basic animations. We also expanded the playing area by zooming the camera out and adding more perspective with the camera angle. Additionally, instead of two shelves, we had a much more elaborate level design that spiraled around the screen. We also included a main menu screen and a tutorial image for the controls before the game began. Once again, our player roles were split between a tool player and a seed player.

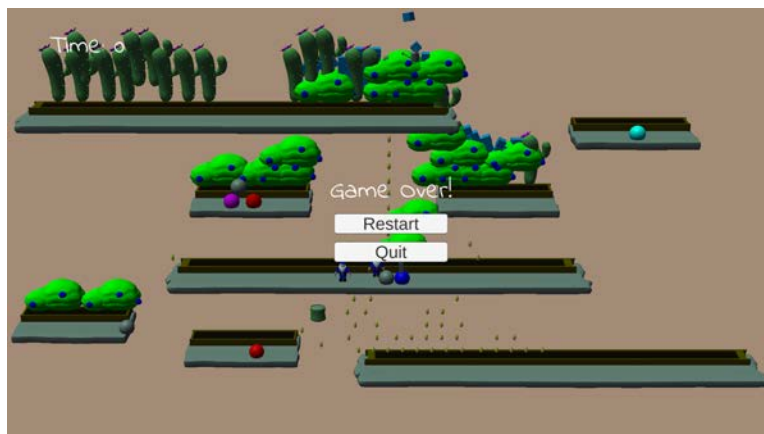


Figure 60: C-Term build

What we Tested

At this point in development, our design was set in stone, and art was finishing up. We wanted to get player feedback on how the final assets looked: their style, animations, colors, size, and more. The two player tool distribution had also changed, so we wanted to see if the balance was suitable for both players. We had also begun creating in-game tutorials and text pop-ups to inform and guide the player about the game, so we needed to receive feedback on how effective they were in helping the player.

Feedback

It had been some time since we had playtested *Gnome Hollow*, so we wanted to make sure that with all of the changes made, we were keeping our player retention and experience where we wanted. Once again, we saw a major improvement in the amount of time players wanted to spend playing our game.

How long would you want to play this game?

17 responses

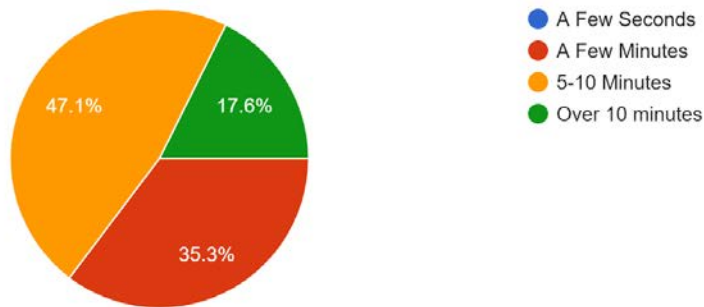


Figure 61: Results from C-Term question “How long would you want to play this prototype?”

Both the amount of people who wanted to play 5-10 minutes and over 10 minutes increased, meaning that implementing functional art assets was helping make our gameplay loop feel more complete.

When asking about the player experience, we found similar results to the last playtesting sessions. Players were describing their experience as we had intended, but there was still some improvement to be made.

How would you describe your experience playing Gnome Hollow? Check the 3 that best apply

17 responses

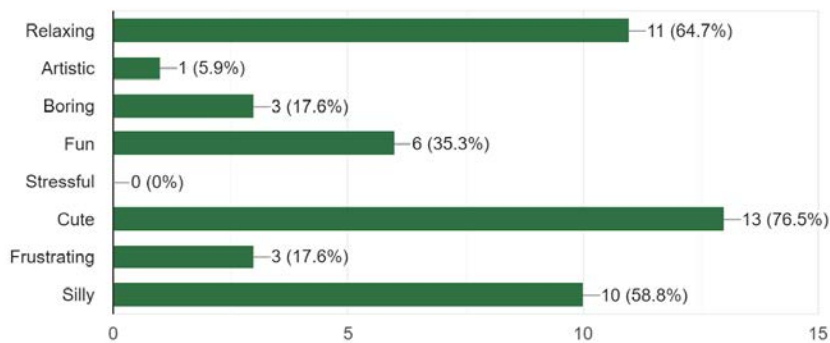


Figure 62: Results from C-Term question “How would you describe your experience? Check all that apply”

We had reduced the amount of stress players were feeling since our last survey, but with that came an increase in frustration. We found from both the text-based responses and from speaking with playtesters that there was frustration coming from the thought that they were losing when the timer ran out. The timer was beginning to conflict with our experience goals, so we needed to revisit its inclusion.

With the addition of some of the first tutorials and tooltips in-game, we wanted to see how effective they were for players.

How effectively did the in-game tutorial options inform you about how to play?

17 responses

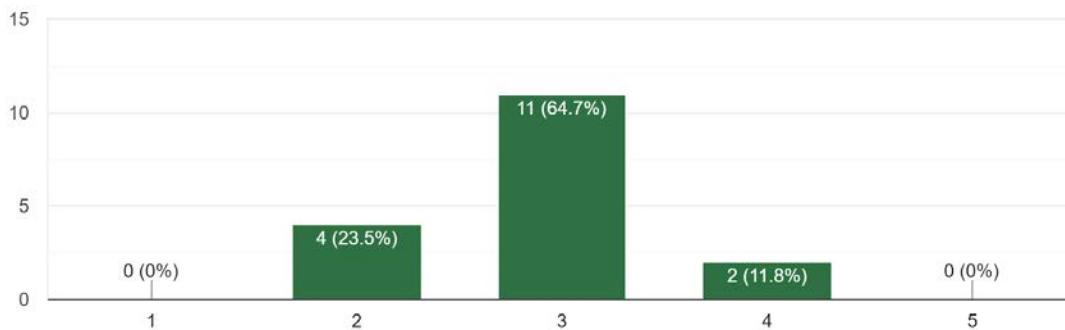


Figure 63: Results from C-Term question “How effectively did the in-game tutorial options inform you about how to play?”

The results were not where we wanted them, as many players felt the game was somewhat or not really informing them about how to play. There needed to be more guidance on how *Gnome Hollow* worked and what the player is meant to do.

Another question with a text response we asked was “Were there any aspects of gameplay that were unclear? Please list them below.” There were three main answers: the goal of the game, the function of the tools, and the fact that players had different roles. We knew these points specifically must be addressed in any new tutorials or in-game guidance that we created.

While previous builds had always involved two players, we really wanted to start making sure the game felt balanced now with our art being implemented. We gave players an opportunity to play both roles, then asked them how well the gameplay was divided.

Was the gameplay effectively divided between both the seed planter and the tool user?

17 responses

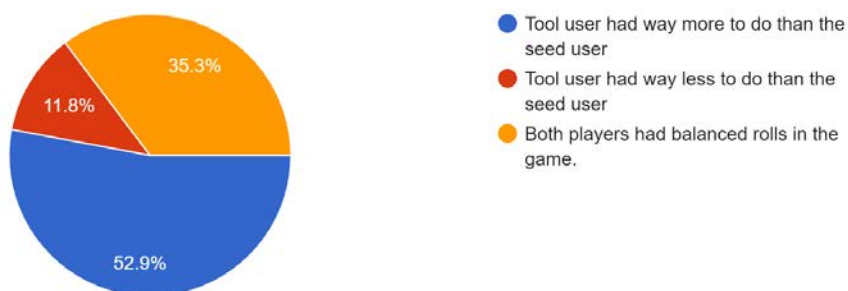


Figure 64: Results from C-Term question “Was the gameplay effectively divided between both the seed planter and the tool user?”

While some players did feel the roles were balanced, over half of them felt that the player who had the tools had much more to do than the player with the seeds. Rebalancing needed to be a priority.

The most notable change for this playtesting session were the visuals, as it was the first time we had any final assets in-game. The models were complete, but they were untextured, so they looked very low-poly. We wanted to see how players felt about the aesthetics, as they were the main draw of our game and we were still hoping to reach a large audience. We also kept in mind that this was not completely representative of how the final assets would appear.

How do you feel about the game's art/aesthetics?

17 responses

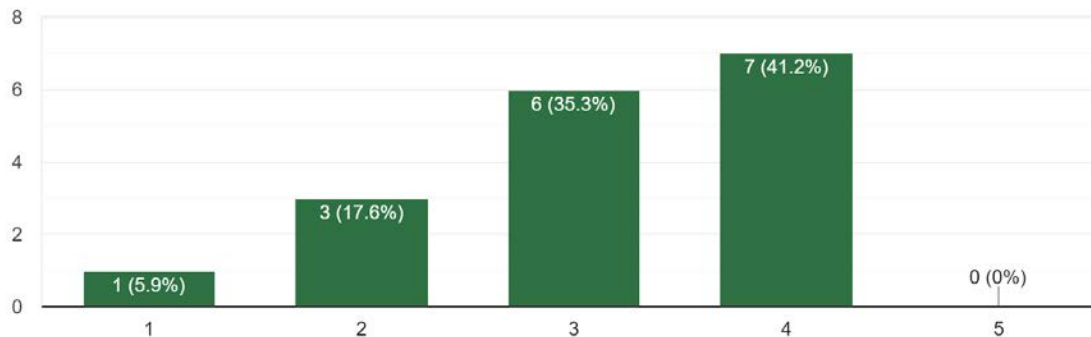


Figure 65: Results from C-Term question “How do you feel about the game's art/aesthetics?”

While the results climbed in a positive direction, there were no responses that loved the art style. As mentioned before, this iteration was not representative of the final visuals, but we still made sure to give as much polish as we could to our textures and other remaining art.

C-Term was heavy in art development, with multiple plants being made per week. We had completed 9 out of an intended 12-15 plants, so we implemented what we had and asked how players felt about the variety of plants.

Was there enough variety between the different types of plants?

17 responses

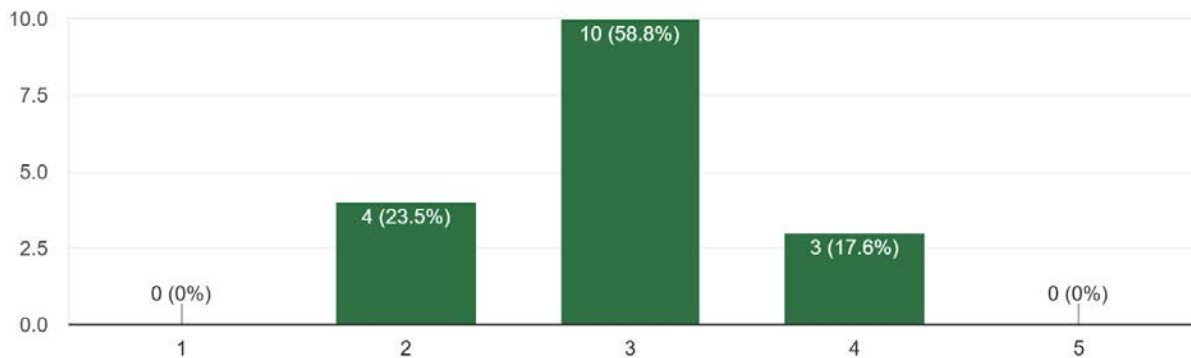


Figure 66: Results from C-Term question “Was there enough variety between the different types of plants?”

On a scale of “Too little variety” to “Too much variety”, we were seeing results landing right in the middle. Our conclusion was that players felt there was an appropriate amount of plants for our game, so we decided to make sure the plants we had were as high quality in all aspects as they could be instead of crunching to model more.

The next step in art development was rigging and animating the plants. For this build, we added a simple grow animation for all plants and use animations for the tools. We hoped that with these animations, players would be able to get a feel for how the other animations may look. Since our aesthetic was very stylized, we wanted to make sure we were not alienating our audience either by going too overboard or not adding enough flair.

Do the animations fit their model's appearance?

17 responses

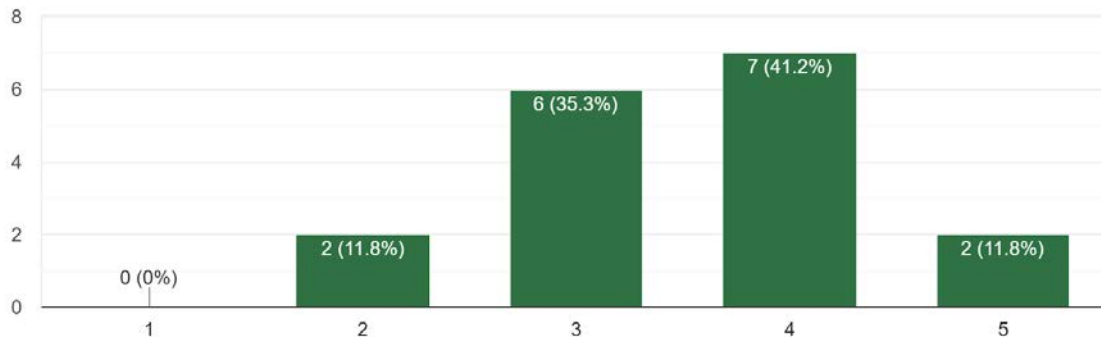


Figure 67: Results from C-Term question “Do the animations fit their model’s appearance?”

Overall, players felt that our animating approach did fit the style of our models. With this in mind, the plants were rigged and animated to be bouncy and cute as we originally intended.

Changes Made

Since our game had very recently begun implementation, there were a lot of adjustments to be made to make sure everything fit together well in the end. Art development continued on as it was going, as all that was left was to finalize and implement all assets. As for the gameplay, we made some major changes. The first was repurposing the two player roles. Players felt the balance was unequal, so we reconsidered how tools would be split. We also adjusted the process of planting by introducing dirt piles. Instead of plants appearing out of the plant beds, we removed the beds and instead focused on just shelves. Players would need to pour dirt on the shelves before being able to plant and grow seeds. The next was remapping the button layout, as players had a hard time learning which buttons were used for which action. We simplified the controls, with one button mapped to each tool, and the triggers to use the selected tool. Players also needed more guidance on what the game’s purpose was, so we started to work on more informative tutorials to make it more clear that *Gnome Hollow* is a sandbox game with no win condition.

There were some other small alterations we made to the game. For one, we increased the size of the tools on screen, as it was easy to lose track of them in a populated garden. We also removed the time limit, as we felt it was conflicting with our experience goals of a calming game. Players felt they had lost once the time ran out, which was not our intention.

8.2.3. D-Term Playtesting

As we approached the final term of the year, our focus was implementation. All of our art assets were complete, and we were able to see everything come together. Throughout D-Term,

there were a few designated MQP playtesting sessions, which we attended in order to get some last-minute feedback. Our goal was not to make any major changes, just to see if we were meeting our experience goals and to work out any last bug fixes.

Game State

Our final term centered around final implementation of all final art assets, specifically skeletons, animations, materials, and audio. Everything was put in game and adjusted to fit together aesthetically. We also added a 3D model of a brick wall for the background to mimic the wall we had chosen on Higgins House. Gameplay had changed in some ways, the main differences were redistributing the tools to each player in order to balance gameplay and introducing dirt piles. One player would be given the glove, fertilizer, and watering can, while the other would have the seedbag, the shears, and the shovel. To grow a plant, the first player must use the fertilizer bag to drop dirt on the shelf. The second player chooses a seed to plant in the dirt, then the former must water the dirt mound for the plant to grow. *Gnome Hollow* was at its closest to completion, as every mechanic and asset was on display.

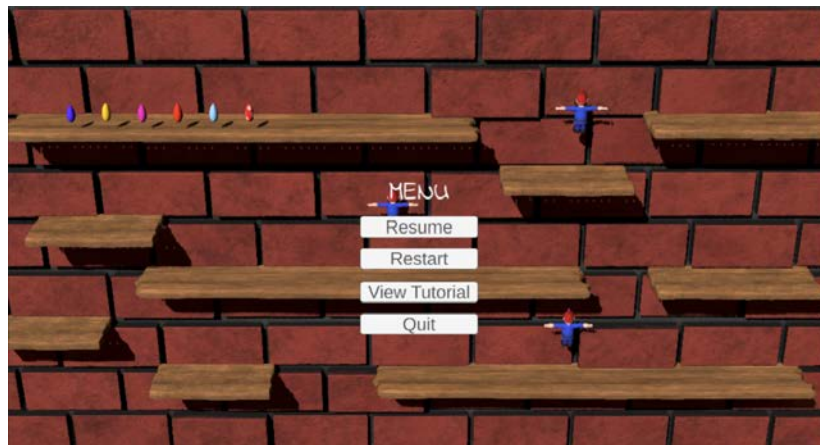


Figure 68: D-Term build

What we Tested

With everything implemented and all gameplay mechanics in place, these testing sessions were mainly for testing the game as a whole: how cohesive all the pieces were together, how the game felt to play, and any bugs that would need to be worked out. The general idea was to get feedback on parts of the game that could be adjusted easily: wilting times, player movement speed, hitbox sizes, and so on. We still kept some feedback open for any ideas players may have, but would only consider ideas that were easily attainable with the short time we had remaining.

Feedback

For the final rounds of feedback, we tested with a very small pool of players. The results for the final survey are only from 3 total responses, but we were able to discuss in detail with the players about their experience playing the nearly finished version of our game.

As with all of our other surveys, we wanted to once again confirm that our experience goals were being met. Instead of just asking how players felt while they played the game, we also wanted to see how they felt before and after. With these three questions, we could gauge if we were meeting our goal of helping players destress.

How did you feel before playing Gnome Hollow? Check those that apply.
3 responses

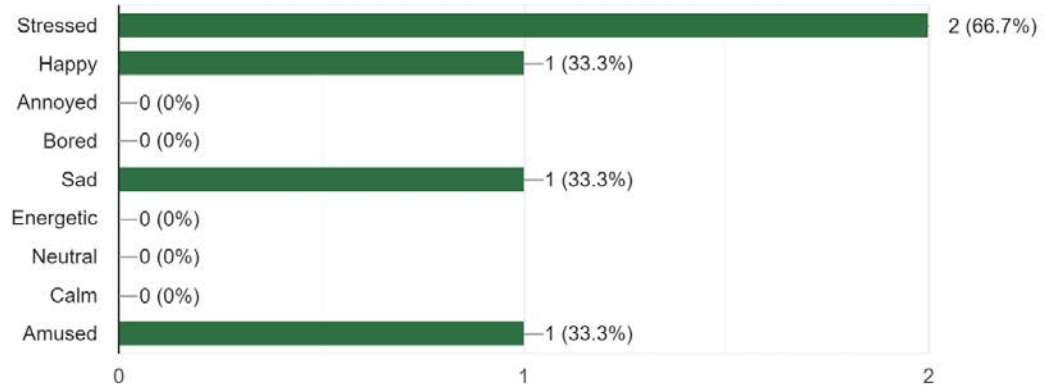


Figure 69: Results from D-Term question “How did you feel before playing Gnome Hollow? Check those that apply.”

Before playtesting our game, players felt more negative emotions on average, with the most common response being “stressed”.

How would you describe your experience playing Gnome Hollow? Check the 3 that best apply
3 responses

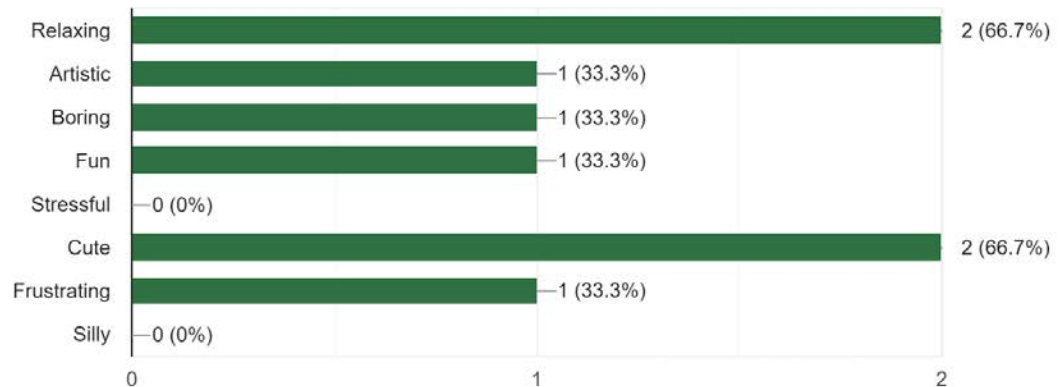


Figure 70: Results from D-Term question “How would you describe your experience playing Gnome Hollow? Check the 3 that best apply”

As the playtesting took place, we saw a decent distribution of what players were experiencing. Unfortunately, there were still some responses indicating boredom and frustration with the game that we found mostly came from players not knowing there is no goal to *Gnome Hollow*. We still had some minor work to do to make sure we were getting this information across to players.

How did you feel after playing Gnome Hollow? Check those that apply.
3 responses

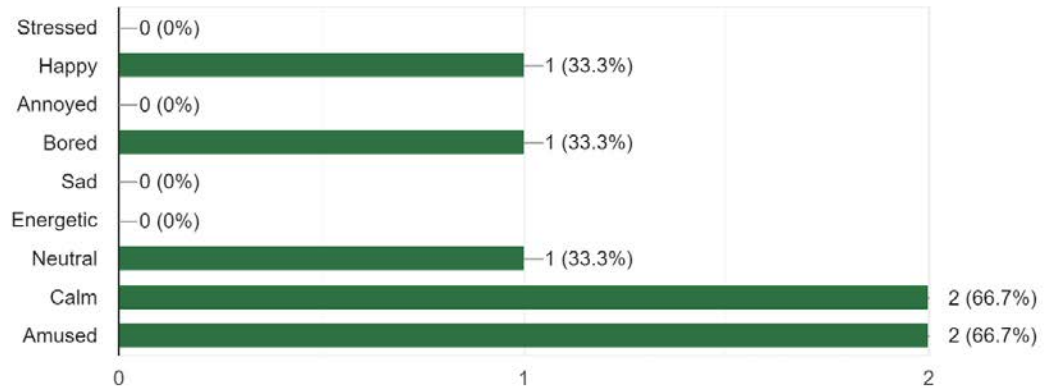


Figure 71: Results from D-Term question “How did you feel after playing Gnome Hollow? Check those that apply.”

After playing *Gnome Hollow*, there is a noticeable positive shift in players’ moods. While one response indicated there was some boredom, players mostly felt calm and amused. The results of these three polls show that we were hitting our intended experience goals.

One of the biggest struggles we faced was finding a way to indicate to players that our game was a sandbox with no goal nor win condition. Our plan was to incorporate a tutorial cutscene that explained this detail, but at the time of testing, it had not been implemented yet. Therefore, our only tutorials were the controller start screen and the in-game tooltips.

How effectively did the in-game tutorial options inform you about how to play?

3 responses

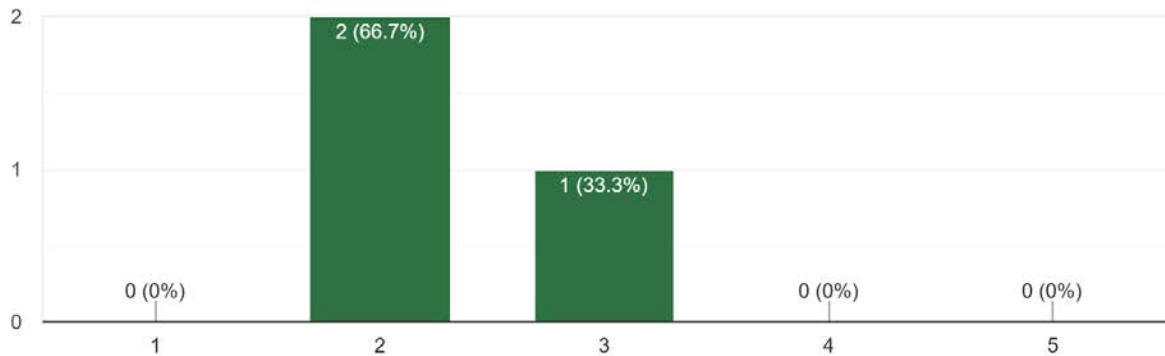


Figure 72: Results from D-Term question “How effectively did the in-game tutorial options inform you about how to play?”

The results showed that this cutscene was absolutely necessary, as players were not being informed well enough by what we currently had implemented.

Another important aspect of the game we wanted to test for was the tool split between the two players. Each time we tested, we had a different division of tools, and we were so far unable to strike the necessary balance of gameplay. This iteration gave each player a move, create, and change tool. The hope was that having each of these functions for different parts of gameplay would allow players to bounce back and forth off of each other to grow and nurture their plants.

Was the gameplay effectively divided between both the plant and tool players??

3 responses



Figure 73: Results from C-Term question “Was the gameplay effectively divided between both the plant and tool players?”

After many iterations, we had found the balance that worked best for players. The cooperative play was at its best and most enjoyable, as neither player felt like they had nothing to do while playing.

Our final open ended questions were intended to help us find and remove as many bugs as possible from the game and for players to give some final thoughts. When it came to bugs, there were a few notable ones. For example, swapping between the different types of seeds would cause the previous seedbag mesh to clip into the new one, resulting in multiple meshes with differently timed animations playing on top of each other. There were also various minor bugs relating to the gnome, as sometimes they would not be able to be picked up. Other times, they would vanish when clicked. The final rounds of playtesting were extremely useful in snuffing out any last bugs that we may have missed with internal testing.

Changes Made

With the end of the project approaching, no major changes took place after testing in D-Term. Our most important feedback was once again needing more clarification from the game's tutorials about how to play. From here, we created a short tutorial video that highlights each part of gameplay. The newest implementation was a pause menu that could be used to revisit the beginning tutorials, reset the garden, or close the application. Besides these, the testing was mostly for finding and addressing bugs. Some of these bugs included dirt and plants being able to stack on top of each other, gnomes disappearing, clipping issues with some of the player cursors, and physics bugs that would cause meshes to bounce randomly. We were able to sort out all the bugs that were found to affect typical gameplay.

For the last bit of polish, we began to incorporate post-processing effects to make our game look even better.

8.3 Major Iteration Changes

Gnome Hollow saw many alterations, both large and small, over the course of the year. Features were added, cut, changed, and evolved. The most impactful changes that we included over the course of our playtesting sessions were as follows:

- Splitting the gameplay between players so that they did not share the same tool sets.
- Moving from player-based level creation to a designated unchanging level design.
- Adding wilting as a mechanic so players would have a reason to keep revisiting grown plants.
- Changing the fertilizer bag's functionality from increasing plant growth speed to creating a pile of dirt for seeds to be planted in.
- Creating a brick wall background for the game environment to increase the immersive feeling of playing on a wall.

All of these changes came from player feedback and led to the final iteration of *Gnome Hollow*. Each adjustment helped to make our game more user friendly, improved the gameplay loop, and helped us reach our experience goals for our players

9. Conclusion

As the development of *Gnome Hollow* comes to a close, there are many things we have learned and many things we want to do. This section of the paper will feature a post-mortem reflecting on our work this year, alongside our limited future plans for *Gnome Hollow*.

9.1: Post Mortem

Reflecting on *Gnome Hollow*'s development cycle, there are several things we could have done better throughout development that could have considerably improved the quality of *Gnome Hollow* had we followed through on them. However, we were still able to accomplish a lot that we were proud of over the course of our final year here. Throughout this chapter, we discuss what we could have improved on, what to avoid in the future when working on these types of projects, and what we executed well during our time on *Gnome Hollow*.

9.1.1. Team

What Went Wrong

The biggest flaw in our development cycle was how segmented and separated our process was. We would largely discuss our overarching plans and then set out on our own weeks-long journeys to get there. Our art team and audio developer designed finalized assets on their own, with our programmer working on implementing and polishing all the required features. This workflow resulted in all of the final art assets getting into the game approximately 2-3 weeks before our ultimate deadline of April 16, 2023, which caused the end of development to be more taken up by art than we would've preferred.

Had we gotten together more frequently for non-advisor meetings, we probably could have done more things like internal playtesting and determining ideal in-game effects that would have allowed us to develop and iterate on the game's feel far more effectively than we ended up doing throughout the development process.

What Went Well

When we got together, we were able to communicate what we needed very effectively to each other. In addition, we collaborated fairly well to get the animations and art assets in quickly. Our work was also able to come together fairly well in the end, which speaks to a shared understanding of the experience goals of *Gnome Hollow*.

9.1.2. Art

What Went Wrong

Scope is an obstacle for nearly every project, so we attempted to balance our work as much as possible when designing our game. With an art-heavy team, we were ambitious with how many plants we wanted in game. Since each plant works the same technically, we could theoretically have as many plants as we wanted without it taking a serious toll on our tech student. While we had originally designed about 15 unique plants, we ended up finalizing 9.

There was also some art functionality that needed to be cut for time, such as randomized plant material colors. In the end, we wanted to make sure we had quality over quantity. There were some minor issues when it came to passing over assets as well, as occasionally there were miscommunications as to which assets were final or works in progress. We also could have had assets prepared earlier to avoid crunch in D-Term.

What Went Well

There were many things that went well in our art pipeline. For one, the assets that we made all fit our original style inspiration. The meshes were cartoony and cute, and did not fall into the artificiality that can come from 3D modeling. Additionally, we had next to no troubles texturing, rigging, or animating the plants and tools. Each part of production was very similar, if not identical, for these assets, and the pipeline was successful. The way we created our animations also lent itself well to using Unity's animator controllers, so implementing our final skeletons and animations was incredibly simple and effective. Overall, *Gnome Hollow*'s art was a success throughout, and our artists were very satisfied with seeing their work come to life in-game.

9.1.3. Tech

What Went Wrong

The struggle of the art rush was accentuated by the fact that there was only one programmer on our project, which meant that every asset the team implemented would have to go through them, halting development on all other technical features. Though our game was scoped well for there being only one programmer, systems could have been better set up that allowed artists/audio people not familiar with the game's tech stack to incorporate their assets without requiring technical assistance. As a result, there were more bugs in the Showfest build of the game than we would've preferred.

The team also did not playtest the game internally as much as we should have, with many not touching the game beyond ProtoFest and AlphaFest until late February or early March of 2023. This resulted in a lot of bugs that would have been easier for anyone aside from the programmer to catch not being addressed for longer than they should have.

What Went Well

Our technical developer was able to get a working prototype with reusable systems up fairly quickly, with that serving as a great base to develop the rest of the game. Importing newer plants into the game was a fairly simple task, as systems set up earlier enabled lots of behaviors to be easily reused. In addition, they were able to abstract the audio implementation system to the point where our audio developer could easily replace sounds.

9.1.4. Audio

What Went Wrong

The biggest challenge facing audio was time and organization. The sound effects, as well as the soundtrack were not finalized in a timely fashion. Additionally, there was a lack of communication between the audio developer and the programmer. This resulted in the sound effects being implemented extremely late, and left not a lot of time for rectifying any technical bugs for our developer, as well as not having a lot of time for mixing. Furthermore, the soundtrack was finalized later than we would have preferred. Scope also affected audio, with sound effects such as the hammer being made, but were later scrapped.

What Went Well

The soundtrack created a sense of relaxation and moved to become upbeat to match the atmosphere of the game. It fit well with the overall aesthetic of the game and matched the repetitiveness of the gameplay loop. Additionally, the gnome and tool sounds fit the gameplay well and were also well received.

9.2: Future Plans for *Gnome Hollow*

Due to our technical developer not accomplishing every technical task in *Gnome Hollow*, they plan on fixing all the bugs in the weeks following the conclusion of the academic year to create a more polished build. There are currently no plans to add new art, animations, or audio to the game. The polished build is expected to be finished at the end of May 2023.

We would also like to convert the game to use Unity's HDRP at some point, as we realized it was a viable option too late and would both improve the game's look considerably and enable our artists to use many more post-processing effects.

In addition, we are considering having another MQP team pick this up in subsequent years. One goal that we could not accomplish for *Gnome Hollow* was making it a fixture on WPI's campus to achieve our goal of making it a relaxing game people can play during downtime between classes. A subsequent MQP team could sort out logistics for this while expanding on the game's plant count, gnome behaviors, and other avenues such as tool capabilities. The codebase would also preferably be polished up by our technician post-MQP to enable future teams to be able to work with more effectively, as it is a bit too duct-taped by the technician's standards.

References

Animal Crossing: New Horizons (Switch version) [Video game]. (2020). Nintendo.

Blue tulips. (n.d.). All About Gardening. Retrieved from

<https://www.allaboutgardening.com/blue-tulips/>.

Brick Wall Backdrop. (n.d.). Amazon. Retrieved from

<https://www.amazon.com/Brick-Backdrop-Party-Accessory-count/dp/B000R4OHJO?th=1>.

Burlap Bags Drawstring. (n.d.). Amazon. Retrieved from

<https://www.amazon.com/10-14-Burlap-Bags-Drawstring/dp/B00JAVUZFK>.

Carnivorous plants. Dribbble. (n.d.). Retrieved from

<https://dribbble.com/shots/15586239-Carnivorous-Plants>

Cartoon Forest: 3D model. Pinterest. (2017, August 31). Retrieved from

<https://www.pinterest.com/pin/811914639043879911/>

Cesun Metal Watering Can. (n.d.). Amazon. Retrieved from

<https://www.amazon.com/Cesun-Galvanized-Removable-Gardening-Gallon-Vintage/dp/B07V9MFQQZ>.

CreativeTrio. (n.d.). *Stylized low poly flowers pack 01 - by Creative Trio*. Sketchfab. Retrieved from

<https://sketchfab.com/3d-models/stylized-low-poly-flowers-pack-01-c75f371ee7f94f05a6198a35863363df>

Cute plants: Character design. Pinterest. (2021, January 7). Retrieved from

<https://www.pinterest.com/pin/91268329939909810/>

David the Gnome close-up. (n.d.). Trouble City. Retrieved from

<https://trouble.city/articles/2020/5/25/22p986n8jzjkoesosroiagwsycoqlw>.

David the Gnome. (n.d.). Facebook. Retrieved from

<https://www.facebook.com/TheGnomeDavid/photos/pb.100063595356306.-2207520000./2259726214062494/?type=3>.

Fairbanks, S. (n.d.). *Argentine Giant Cactus.* Birds and Blooms. Retrieved from

<https://www.birdsandblooms.com/gardening/drought-tolerant-gardening/cactus-flowers/>.

Garden Shears Prop. (n.d.). NewRuleFX. Retrieved from

<https://www.newrulefx.com/products/newrulefx-garden-shears-prop-with-functional-moving-parts>.

Garden Shovel. (n.d.). iStockPhoto. Retrieved from

<https://www.istockphoto.com/photo/garden-shovel-isolated-on-white-background-gm666079998-121374293>.

Garden Trowel. (n.d.). Walmart. Retrieved from

<https://www.walmart.com/ip/Garden-Trowel-Hand-Shovel-1-Piece-Cast-Aluminum-Gardening-Tool-Garden-Hand-Tools-for-Men-Women-and-Kids/858785602>.

Giardini, D., Yukhanov, D., & Penner, R. (2014). A unity tween engine. DOTween

(HOTween v2). Retrieved April 26, 2023, from <http://dotween.demigiant.com/>

Gnomes and mushroom in cartoon style on white background. Freepik. (2020, June 24).

Retrieved from

https://www.freepik.com/free-vector/gnomes-mushroom-cartoon-style-white-background_8917374.htm

Granberg, A. (2022, November 8). A* pathfinding project. Retrieved April 26, 2023, from

<https://arongranberg.com/astar/>

Hedge Shears. (n.d.). Ecogardener. Retrieved from

<https://ecogardener.com/products/extendable-hedge-shears>.

HHayes. (n.d.). *Mushroom character - 3D Model* . Sketchfab. Retrieved from

<https://sketchfab.com/3d-models/mushroom-character-de47d2e84d984b54bece5b2b47904ea1>

IMDb.com. (2014, March 11). *Towerfall: Ascension*. IMDb. Retrieved from

<https://www.imdb.com/title/tt9214738/>

Metal Watering Can. (n.d.). Dennyshop. Retrieved from

<https://www.dennyshop.co.uk/products/10l-galvanized-metal-watering-can>.

Nintendo. (n.d.). *Duck Game*. Nintendo Official Site. Retrieved from

<https://www.nintendo.com/store/products/duck-game-switch/>

Nintendo. (n.d.). *Stick Fight: The Game* . Nintendo Official Site. Retrieved from

<https://www.nintendo.com/store/products/stick-fight-the-game-switch/>

Overcooked (Switch version) [Video game]. (2016). Nintendo.

Pikmin 4 (Switch version) [Video game]. (2023). Nintendo.

Plants vs. Zombies: Garden Warfare 2 (PC version) [Video Game]. (2016). Electronic Arts.

Poortvliet, R. (n.d.). *Gnome carpenter*. Etsy. Retrieved from

<https://www.etsy.com/listing/666069840/vintage-art-print-80s-illustration-of>.

Poortvliet, R. (n.d.). *Gnome Carving Wood*. Etsy. Retrieved from

<https://www.etsy.com/au/listing/576668642/vintage-art-print-80s-illustration-of-a>.

Poortvliet, R. (n.d.). *Gnome sitting on a mushroom*. Pinterest. Retrieved from

<https://www.pinterest.com/pin/321725967141506758/>.

Poortvliet, R. (n.d.). *Rien Poortvliet Gnomes*. Pinterest. Retrieved from

<https://www.pinterest.com/pin/405464772682116015/>.

Posner, J., Russell, J., & Peterson, B. (2005). The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology.

Development and Psychopathology, 17(3), 715-734. doi:10.1017/S0954579405050340

Round ornamental bush. (n.d.). Shutterstock. Retrieved from

<https://www.shutterstock.com/image-photo/round-ornamental-bush-isolated-on-white-132902558>.

Rustic Wood shelf. (n.d.). Etsy. Retrieved from

<https://www.etsy.com/listing/820957273/rustic-live-edge-solid-teak-wood>.

Sanchiesp. (n.d.). *Garden gnome 4 - 3D model*. RenderHub. Retrieved from

<https://www.renderhub.com/sanchiesp/garden-gnome-4>

Scrap Mechanic (PC version) [Video game]. (2016). Axolot Games.

Shovel Dimensions. (n.d.). Amazon. Retrieved from

<https://www.amazon.com/Newooh-Universal-transplanting-Corrosion-Resistant-Ergonomic/dp/B088NGSTKM>.

Sunflower. (n.d.). Wormenkwekerij Wasse. Retrieved from

<https://wormenkwekerijwasse.nl/product/zonnebloem-grootbloemig/>.

Super Mario Odyssey (Switch version) [Video game]. (2017). Nintendo.

Super Mario Maker 2 (Switch version) [Video game]. (2019). Nintendo.

teamLab. (2017). A musical wall where little people live. teamLab Learn and Play! Future Park.

Retrieved April 26, 2023, from

<https://futurepark.teamlab.art/en/playinstallations/kobitowall/>

Vinnikova, I. (n.d.). *Red and white mushrooms*. Discover Magazine. Retrieved from

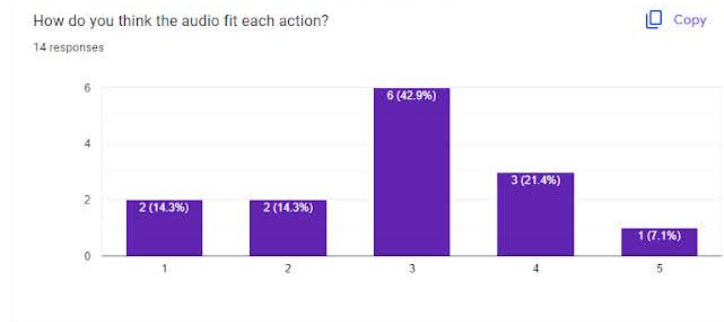
<https://www.discovermagazine.com/health/the-other-magic-mushroom-why-a-canadian-startup-is-suddenly-interested-in>.

Wood Shelf. (n.d.). Pinterest. Retrieved from

<https://www.pinterest.com/pin/815010863829581281/>.

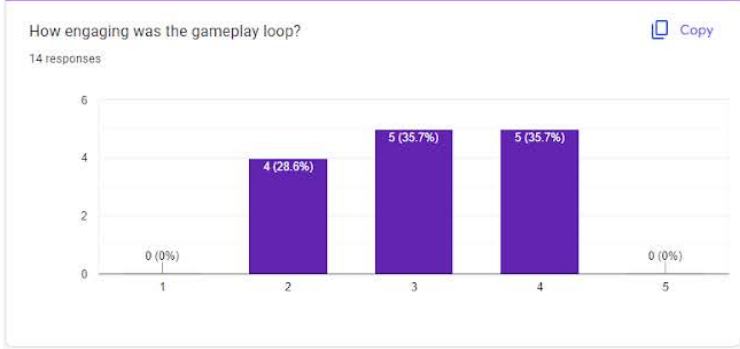
Appendices

Appendix A: Protofest Survey with Responses



- If you answered "Did Not Fit" for the previous question, what about the audio was off? 8 responses
- I could not hear the audio
 - I couldn't hear it lol
 - no audio
 - It's too long
 - sound effects lasted too long and when you spam a button they overlap
 - There was no audio
 - Didn't play with audio. lol
 - couldn't hear

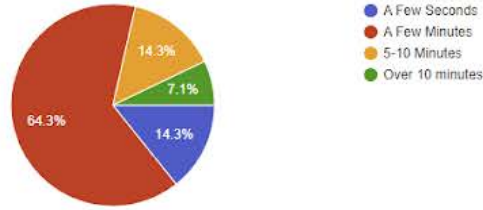
Playtesting Questions



How long would you want to play this prototype?

[Copy](#)

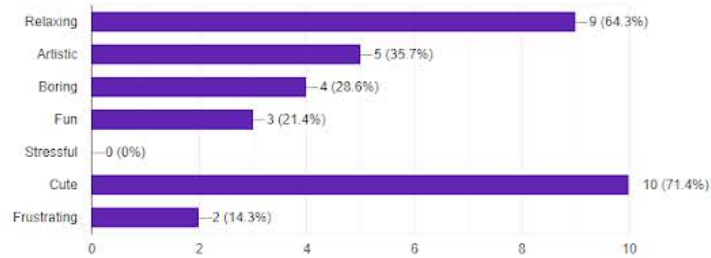
14 responses



How would you describe your experience? Check all that apply

[Copy](#)

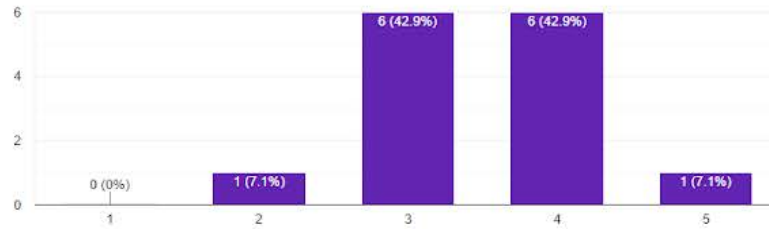
14 responses



How intuitive was the game's control scheme?

[Copy](#)

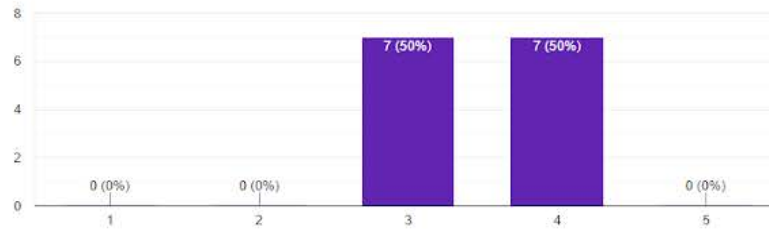
14 responses



How would you rate the scale of the objects on the screen?

[Copy](#)

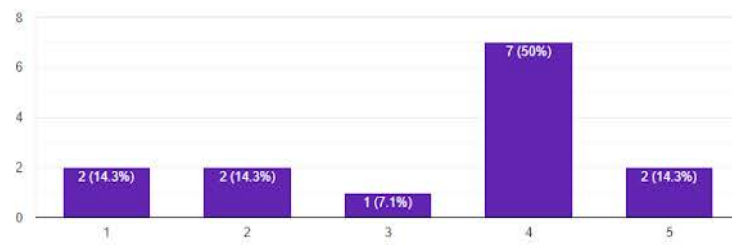
14 responses



How clear is a tools function based on its appearance?

 Copy

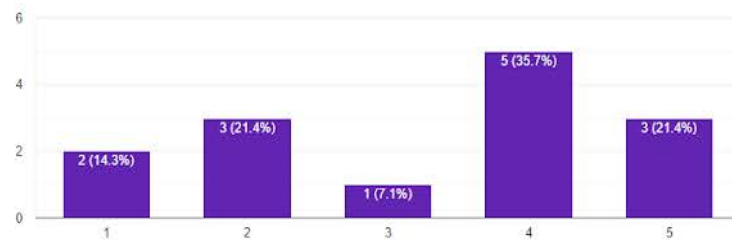
14 responses



How would you rate the consistency of aesthetic between assets?

 Copy

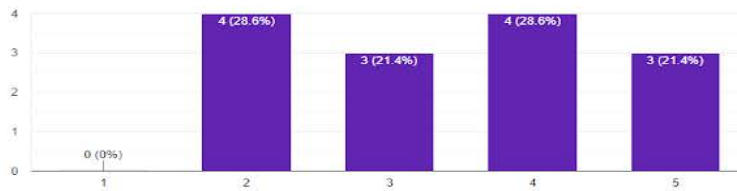
14 responses



How do you feel about the game's art/aesthetics?

 Copy

14 responses



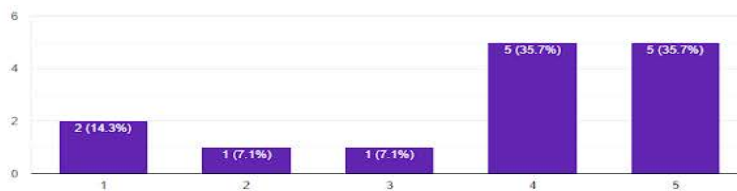
Player Interest

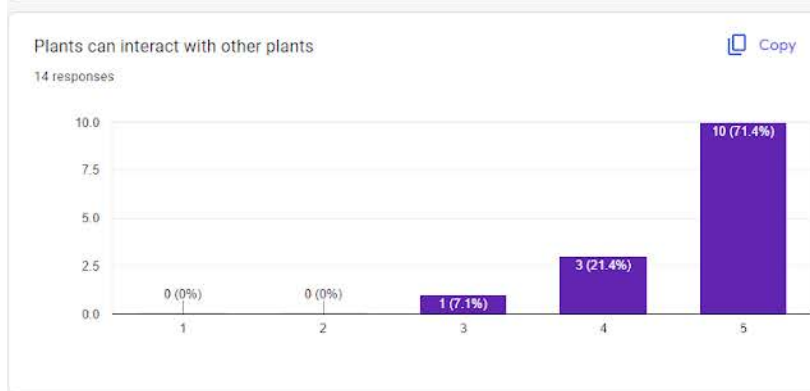
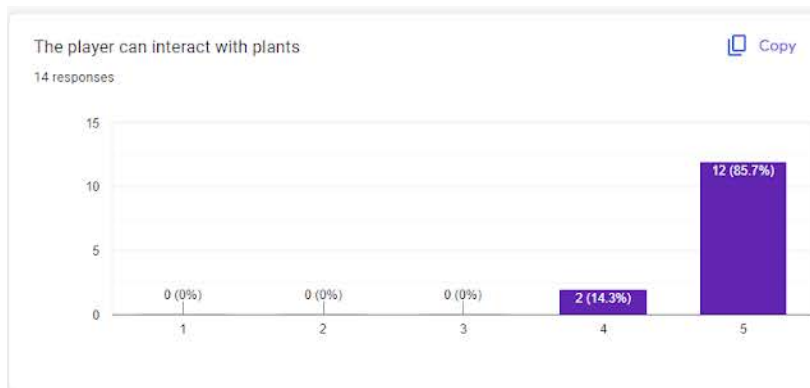
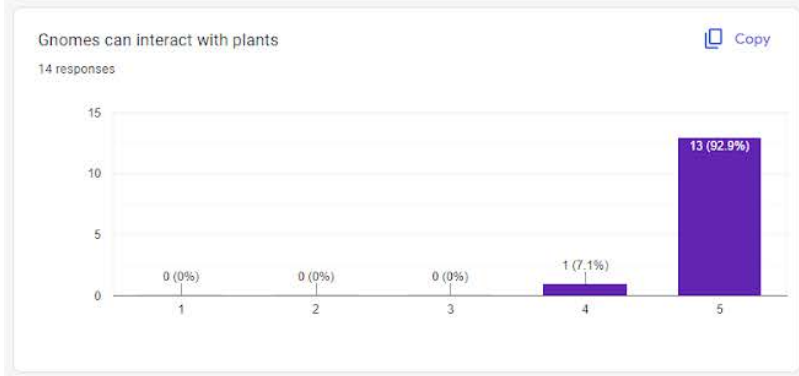
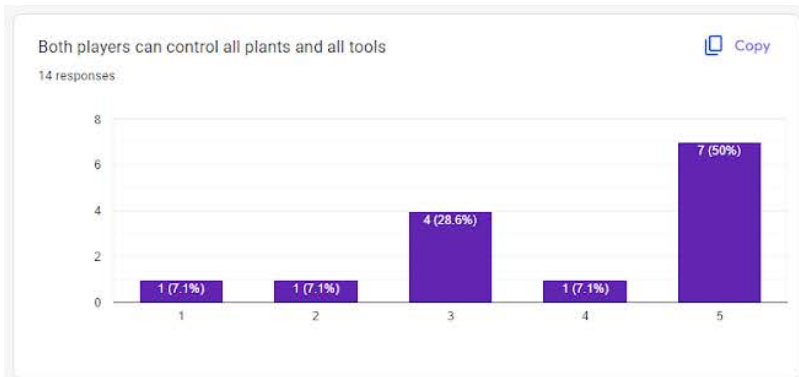
Game Mechanics Questions

One player controls all tools (watering, fertilizer, etc.) and the other controls all plants.

 Copy

14 responses

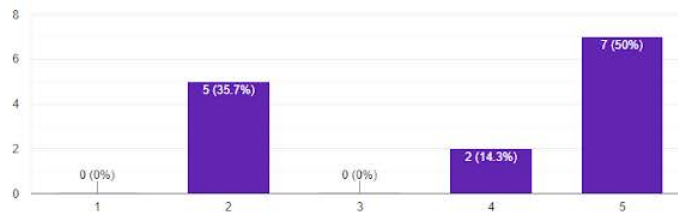




Players can obtain currency to buy and unlock more plants

Copy

14 responses



Is there any additional feedback you would like to give in regards to gameplay?

9 responses

The controller didn't take analog input, making movement feel choppy and heavy. Even if movement speed was constant, I'd rather have movement with the slightest push on the stick, not maximum push.

Gnomes are cool

I love the idea of a cooperative game where one player controls one thing and the other player other things. It helps to make the game more unique!

Right now there's no clear goal or reason for it to be multiplayer

Going well

The cursor has momentum-it should be nice and predictable instead.

More diverse plants, tools, and other objects would add to the variety.

I would be interested if this game had a plants versus zombies type of art style. The description given gives me the idea that the game should have multiple levels with each object and plant life. Each level would be different and there would be a time-limit and objective for each level. I also feel like this game should have an interesting background that displays the theme or objective of each level. Possibly using Higgens garden as inspiration.

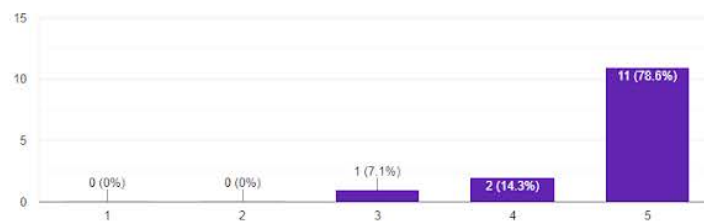
there are no real goals or purpose atm. And you will also have to consider how people can just pick up and put the the game. with such a large screen and the purpose of being there for people walking by to use, you need to make it be quickly enjoyable and put down able.

User Interface Questions

A "weapon wheel" for tools.

Copy

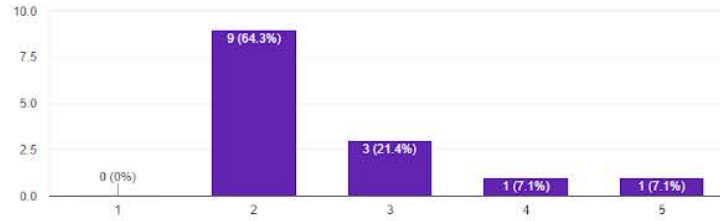
14 responses



A "toggle" for tools. Players click until they land on the tool they desire.

[Copy](#)

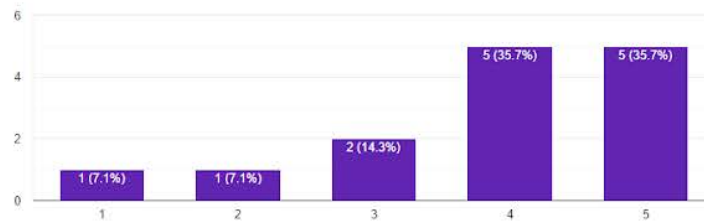
14 responses



A "shelf" for plant seeds that is permanently on the screen.

[Copy](#)

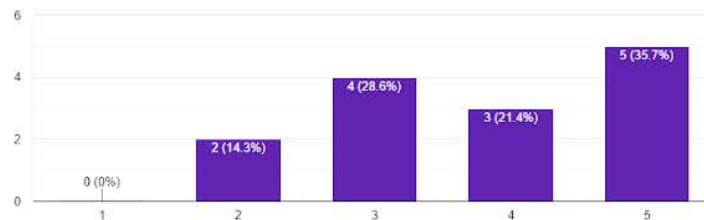
14 responses



A "column" for plant seeds that can be cycled through to select.

[Copy](#)

14 responses



Is there any additional feedback you would like to give in regards to user interface?

7 responses

- I'd love to see a timer and some kind of completion / accomplishment stat or bar
- Definitely tooltips to remind you what tools do and what plants are what
- I think that having all the seed options on the screen at once would be better than having to scroll through them
- Try to minimize ui if you're broadcasting to a building. I think the hard part is making sure the player knows how to play in the first place
- It should also have text describing what each object does.
- I believe having a column to toggle through the items/ tools is the best way for this type of game from playing it. There was a space between the platforms and I think that can be where the toggle UI can be places. Sort of like Minecraft.

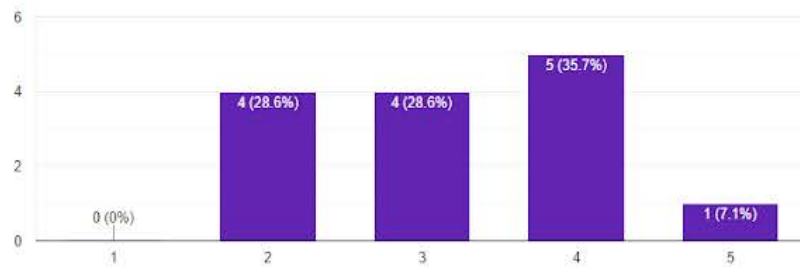
you should be able to quickly access any tool to any other tool. having a 1 way cycle is annoying if i mispress and go past what i want

Player Experience Questions

An experience that is more about visual and auditory effects than gameplay

[Copy](#)

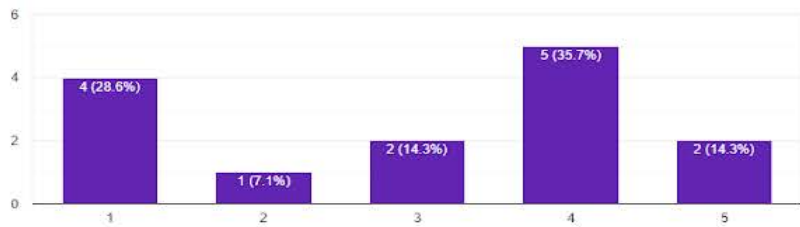
14 responses



An experience where you are scored based on your garden's appearance

[Copy](#)

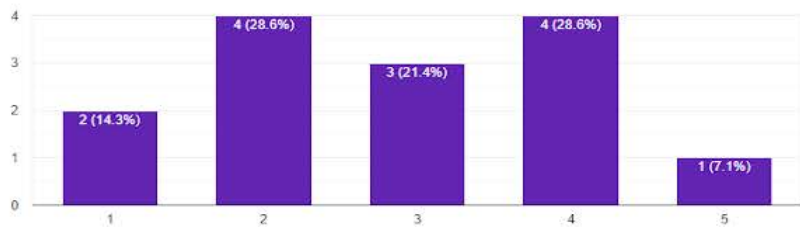
14 responses



A fast paced round where you have a small amount of time to make your garden

[Copy](#)

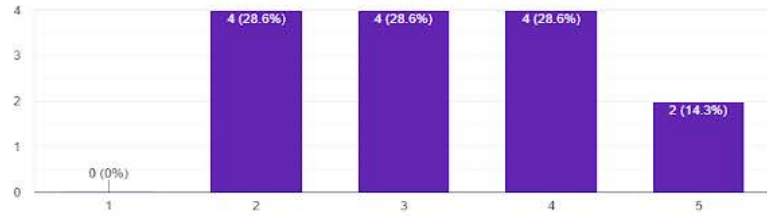
14 responses



A game that does not require cooperation to succeed

Copy

14 responses



Is there any additional feedback you would like to give in regards to player experience?

6 responses

Seems important to see last garden or highlights of last playthrough between games.

I think there is a lot of potential for something chaotic here and I encourage leaning into the chaos

make the gnomes troll the player

A game like this is best suiting for casual play, not fast-paced rounds — although making a competitive garden building simulator could be interesting if executed well.

Make the game like a level builder can be interesting. So that each gameplay can be different and allows for player replay-ability.

learning curve needs to be very low

Appendix B: AlphaFest Survey with Responses

What (if anything) grabbed your attention about our game that made you want to play?

27 responses

I liked the art style and poster

Gnomes are pretty rad

the Gnome did

The graphics grabbed my attention the most.

It looked cute.

The game being on the projector.

growing the plants

Gnomes!!

ironically, the grabbing feature. the game is hugely fun when it comes to the various interactions

making lots of gnomes!

the chaotic fun nature

sandbox game, allowed to do anything you'd like

Gnome Spawning

meme theme

Spawning things is fun

gnomes

The gnomes looked silly

The focus on gnomes, and the absurd amount of them

I thought the gnomes looked funny bouncing around.

The funny gnome stack animations

The gnome

gnomes are funny looking

2 player, chaotic interactive elements

Gnomes!

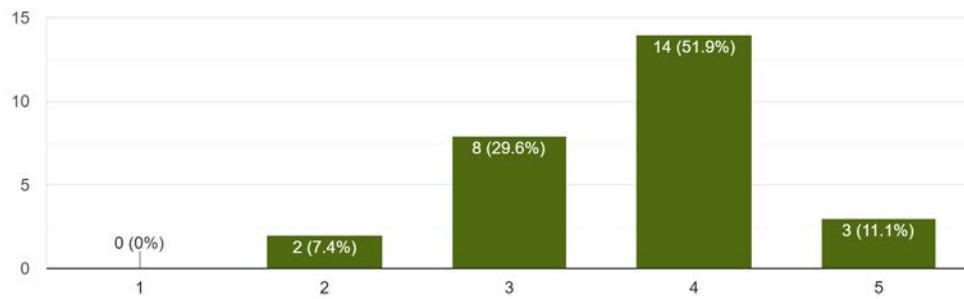
The camera angle of the game and the number of objects made the game stick out

gnomes and physics

The gnomes. I love how goofy they are.

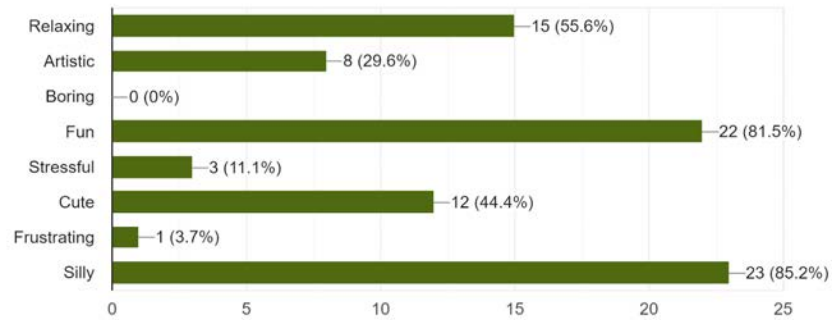
How engaging was the gameplay loop?

27 responses



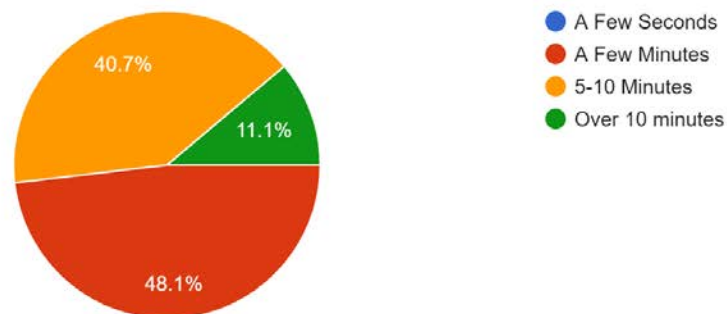
How would you describe your experience? Check all that apply

27 responses



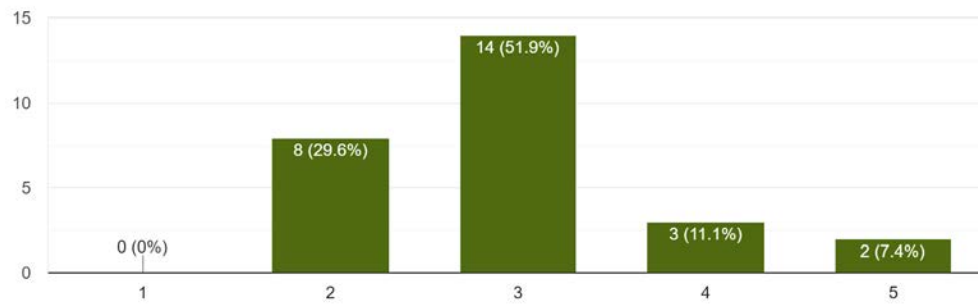
How long would you want to play this game?

27 responses



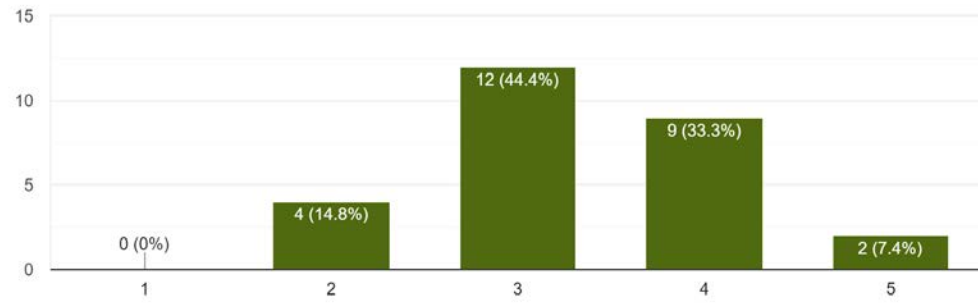
How would you rate the speed of the player's cursor?

27 responses



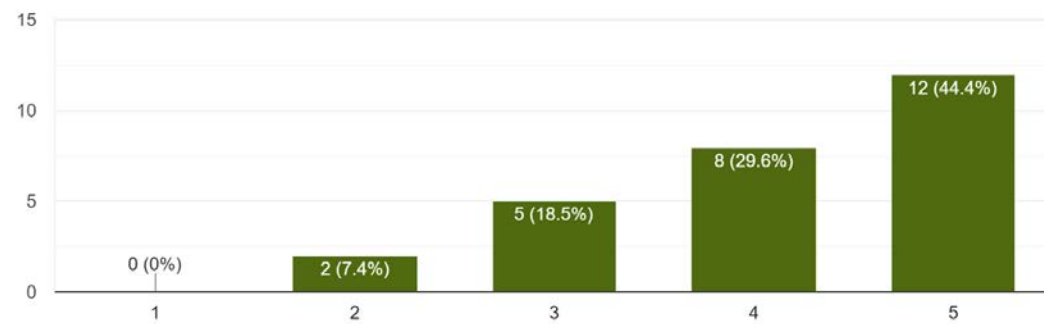
How intuitive was the game's control scheme?

27 responses



Did you enjoy the multiplayer aspect of the game?

27 responses



Is there any additional feedback you would like to give about the multiplayer gameplay?

18 responses

I liked the fact tools were split up so you had to work together

it kind of seemed like a non-factor at this stage.

change the cursors for players

Map design could be given more thought

Picking up a gnome seemed to drop it in front of the platforms

it was fun. can you play with more than 2 players?

it would be cool if there were things that you add to do in sync

Players don't really have much interaction with each other- they should have more tools that differ from each other

I ended up playing the game semi-competitively. I feel like it could be a deliberate design decision to make teamwork difficult with players trying to screw each other over.

it honestly makes no difference for me if i play solo or with another player

It would be interesting if there was more interaction between the two players, like for example if one player had to plant the seed and the other had to grow it. Some sort of collaboration would be cool. I did like the sandbox feel the game had though.

perhaps a competitive feature

I think it would be more fun if there was benefit to working together, such as mushrooms growing faster when the other player waters them

should be able to make gnomes fight when each player holds them :D

Allow the players to grab and move the other's cursor

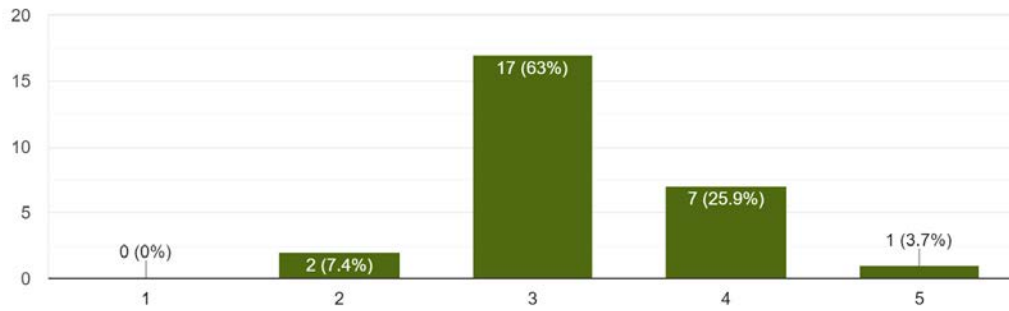
could have a garden background image

Things were chaotic, they kept on flying all over the place

didn't do multiplayer

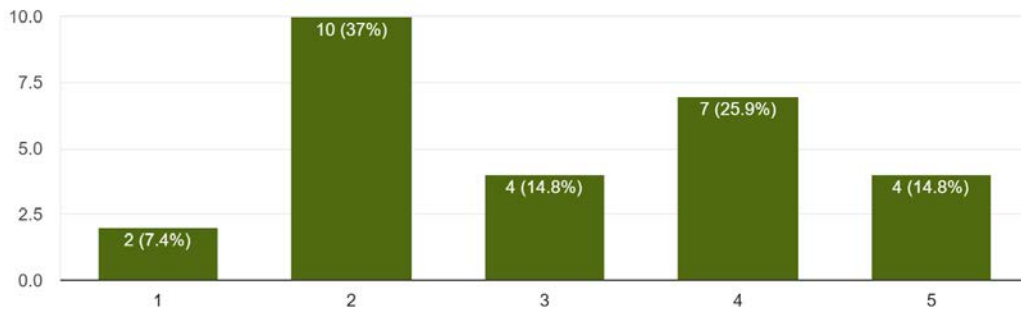
How would you rate the scale of the objects on the screen?

27 responses



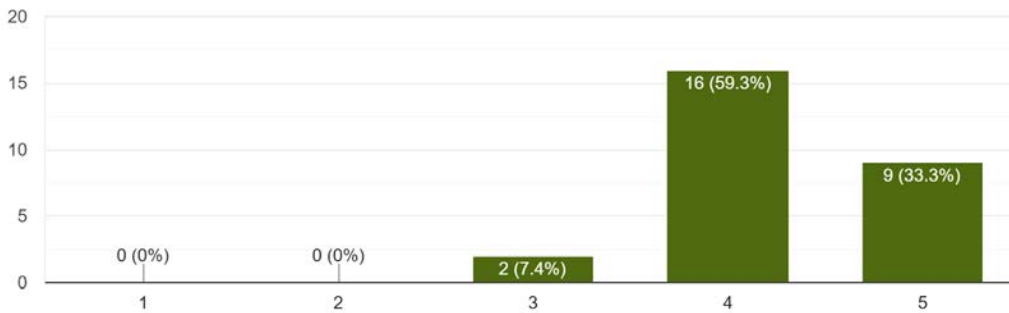
How clear is a tools function based on its appearance?

27 responses



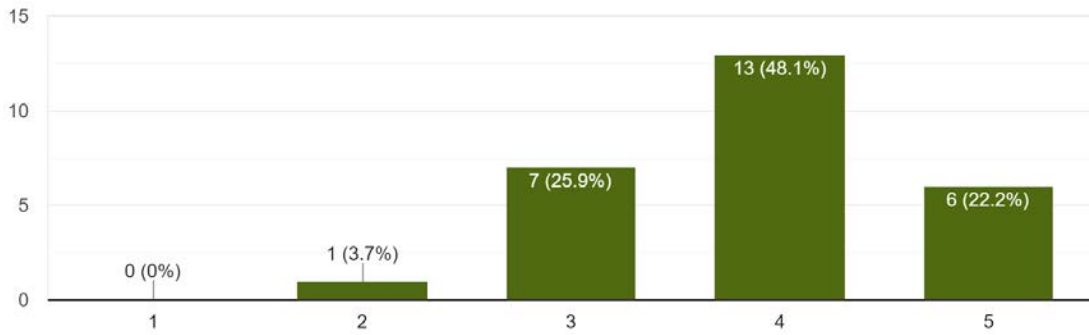
How would you rate the consistency of aesthetic between assets?

27 responses



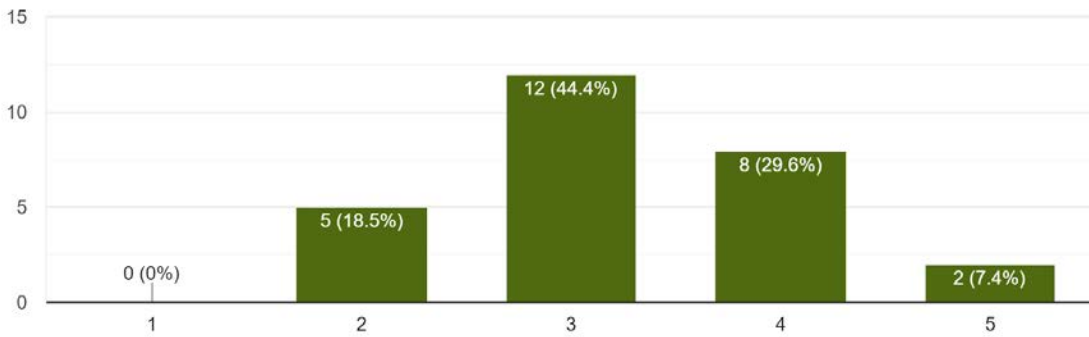
How do you feel about the game's art/aesthetics?

27 responses



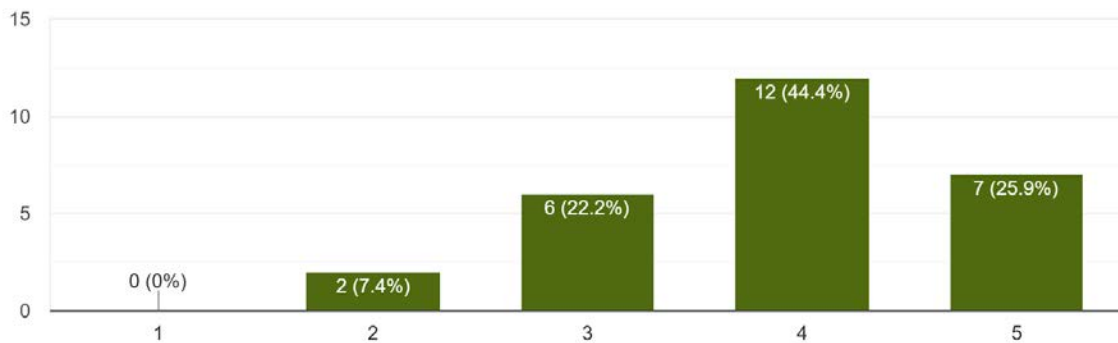
How were the game's colors when projected?

27 responses



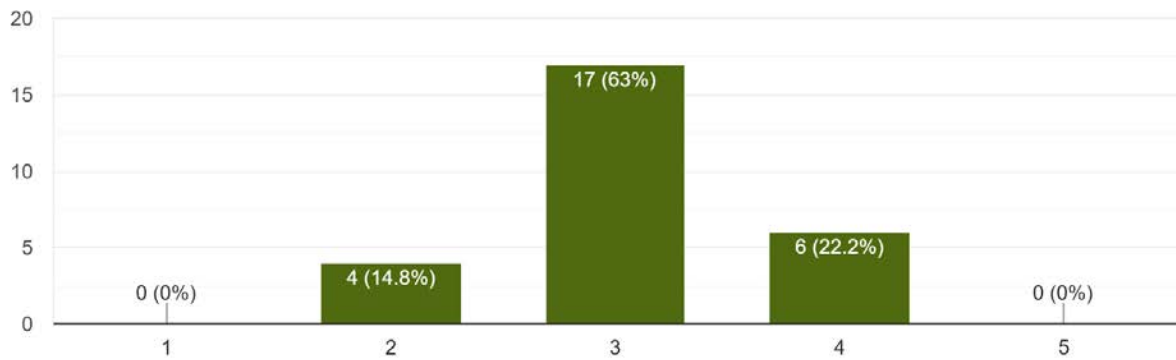
Do the animations fit their model's appearance?

27 responses



How would you rate the animation speeds?

27 responses



Any additional comments about the art and animations?

11 responses

perfect

i like the simplicity of the art

The gnomes were cute but a bit pixelated, if it were on a building wall I don't think it would be as evident

it's cute!

very cute!

The water bucket animation indicated it was shooting the water horizontally. It took me a while to realize the blue blocks were the actual water.

It was funny to see a stack of gnomes bouncing in mushrooms. It's also funny to summon tons of gnomes.

Lovely gnomes

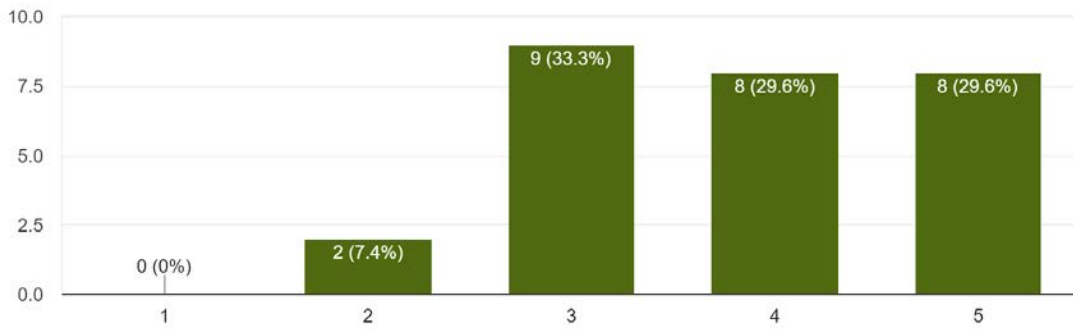
the seed felt too big. also didn't know there was more than one seed until i was told. I think you should

The water blocks coming out from the bottom of the can instead of from the nozzle made the object feel odd to use

The gnomes should probably be shmovin

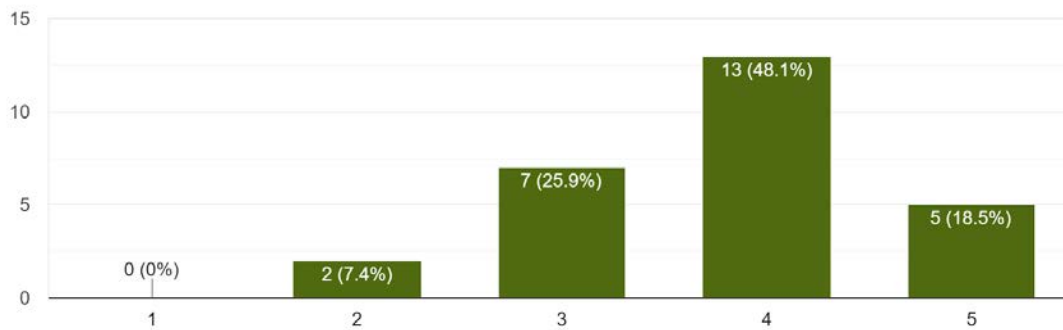
How do you think the particle effects affected gameplay?

27 responses



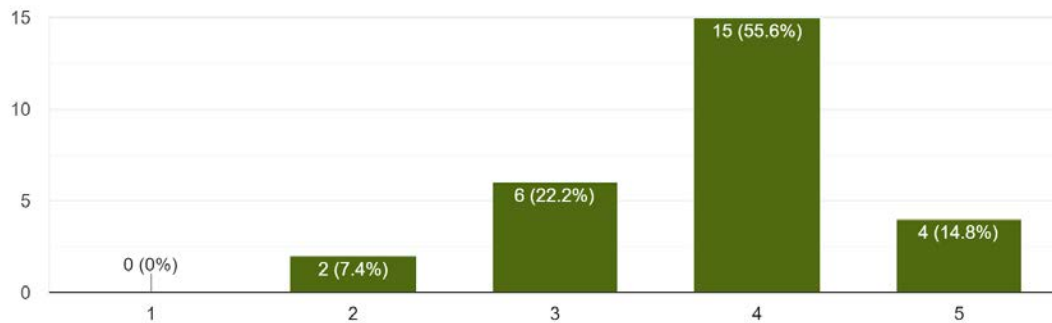
How do you think the particle effects fit each action?

27 responses



How would you rate the size of the particle effects?

27 responses



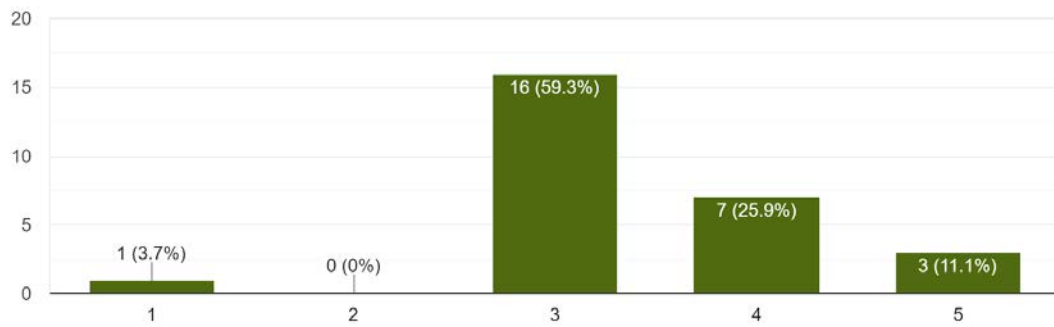
Is there any additional feedback you would like to give about the particle effects?

6 responses

- none
- the water was my favorite
- The blue particle effects perfectly fit the water from the water jug.
- Make them smaller and more apparent
- could have a machinegun mode
- I didn't know what some of them did

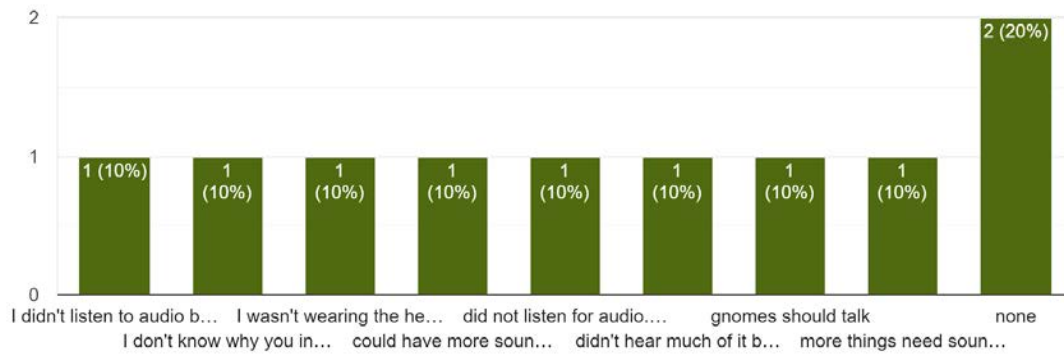
How do you think the audio fit each action?

27 responses



Is there any additional feedback you would like to give about the audio?

10 responses

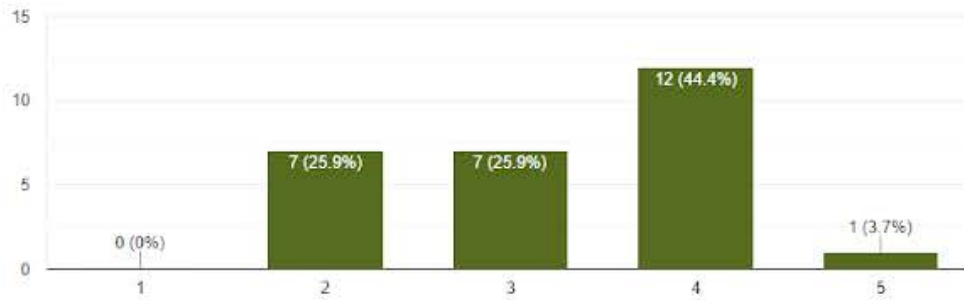


Player Interest

An experience that is more about visual and auditory effects than gameplay

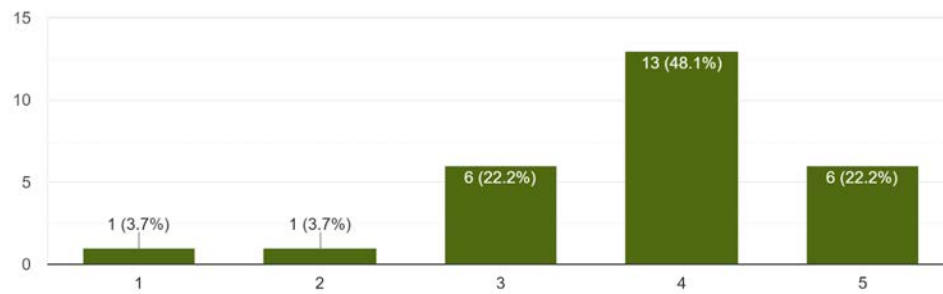
 Copy

27 responses



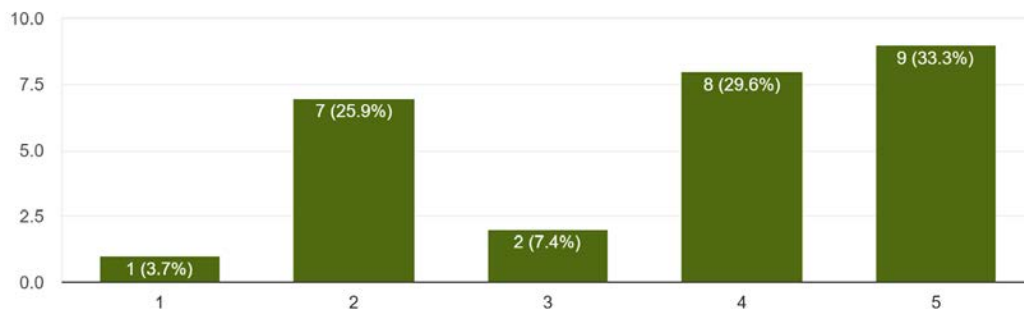
An experience where you are scored based on your garden's appearance

27 responses



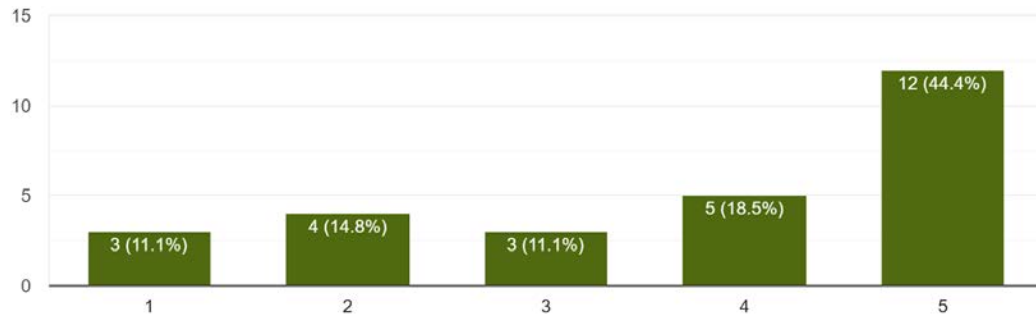
A fast paced round where you have a small amount of time to make your garden

27 responses



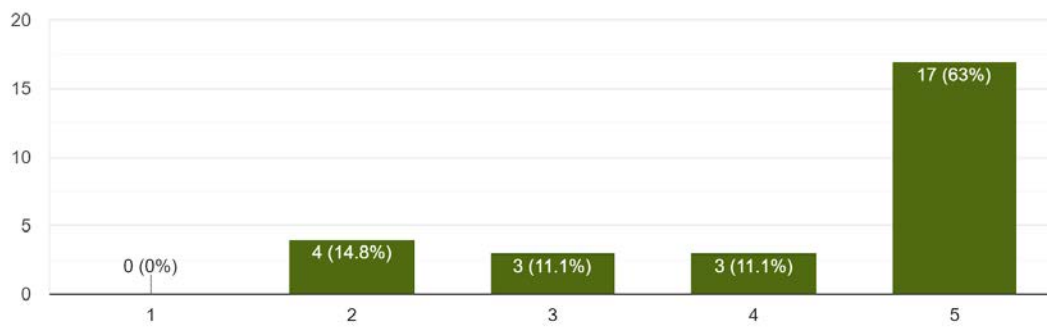
A slow paced round where you have a large amount of time to make your garden

27 responses



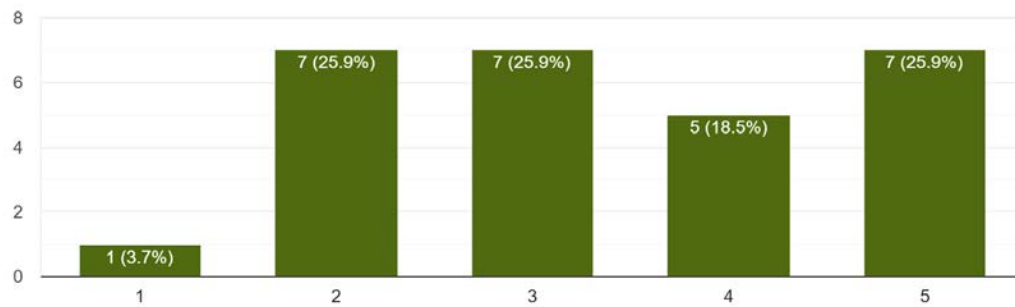
A game that requires cooperation to succeed

27 responses



A game that does not require cooperation to succeed

27 responses



Is there any additional feedback you would like to give in regards to player experience?

10 responses

the ps5 controller x button didnt work as a use button but it worked when changing the seeds/ pressing square button

A co-op goal would be very cool to implement.

multiplayer is fun

amazing game that has a lot of potential- I look forward to its future!

I feel like you could emphasize some competitive aspects of the game while keeping it a cooperative experience. Overcooked is a good example of a game that makes cooperation necessary but difficult.

Had a lot of fun! Interesting proof of concept

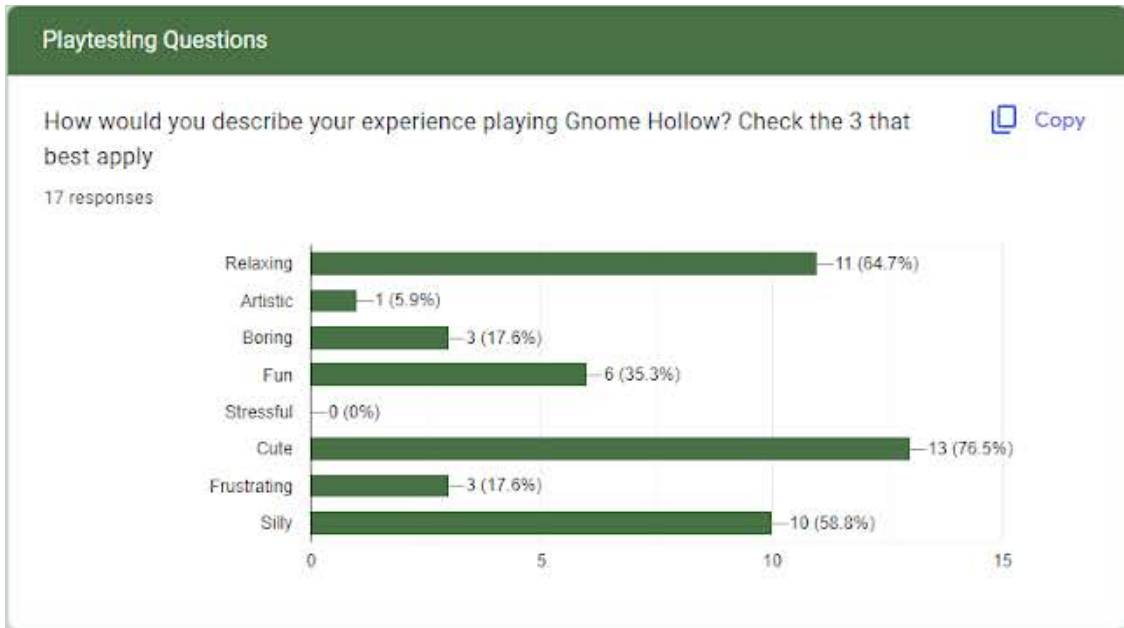
It would be cool if there were different modes to cooperate or interfere with each other

it's a lot of fun

make a competition to make the better garden in a short amount of time, while adding the ability to use gnomes to stop the other player and mess up their garden

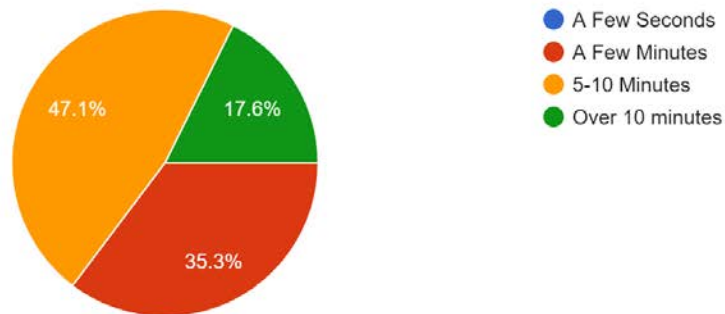
if the ground was dirt textured and there was UI related to gardening I think it'd communicate the goal of the game more

Appendix C: C-Term Survey with Responses



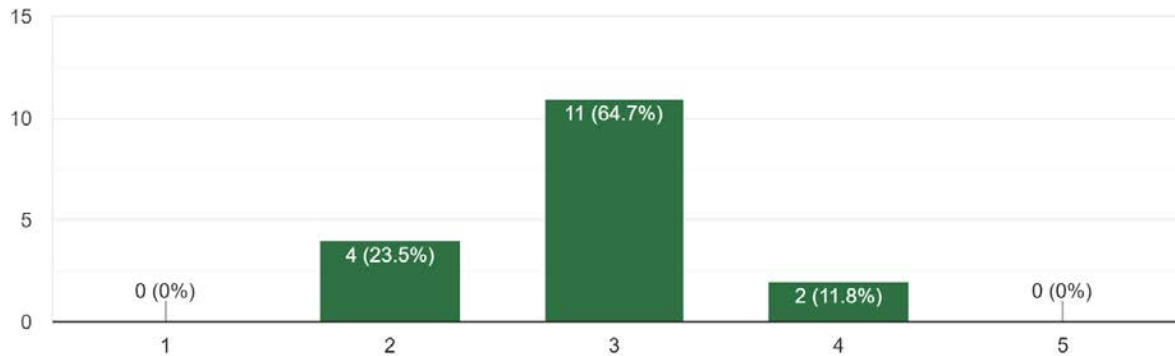
How long would you want to play this game?

17 responses



How effectively did the in-game tutorial options inform you about how to play?

17 responses



Were there any aspects of gameplay that were unclear? Please list them below.

17 responses

It wasn't immediately clear how certain tools worked or what had to be done exactly. The game became somewhat overwhelming with all the different prompts appearing on screen both on the top and bottom.

A few of the tools didn't have clear descriptions as to what they did but after trying it out with the gameplay they were pretty clear. The only object I wasn't sure about was the box so maybe a little description of each object would be helpful. I also wasn't sure how the seeds and switching them worked so maybe displaying the instructions clearer would be more helpful.

I wasn't sure what some of the tools did and the text at the bottom didn't always appear to explain. Also not entirely clear what the overall goal is or if there are any interactions between the gnomes and the plants.

It took me a bit to realize there was a difference between the two players in terms of seed planter vs waterer, the tools were clear though and so were the mechanics on the intro screen showing the controller. I was also unclear about what the role of the gnomes were in the game at its current stage.

The controller hints and little text messages on the screen helped me understand how to control the game, however it was a little unclear what each person's specific job was. I also remember it saying to interact with the gnomes, however it was unclear how exactly to interact with them. I also really like the concept behind the gnomes and everything, but it wasn't immediately obvious how they come into play in the game; are they supposed to harvest the plants we plant?

I didn't understand that there was a difference between the player's tools, what the tools did, what the goal was, what the purpose of each of the tools was (I could only find a use for the shears and the watering can), or what the gnomes were for or how to interact with them.

The goal of this game. Items tracking (don't know where did I land the seeds if I did not focus).

Wasn't sure what the purpose of all my tools was.

I didn't realize that only 1 player could plant seeds.

Player responsibilities (ie. one person has the tools and the other has the bag of seeds).

The difference between Player 1 and Player 2 was unclear, although that might just be because the person running the game went through the tutorial screens too fast and I didn't catch it.

I'm not sure what the goal of this game is?

The two blue guys just wandering in the map, I not quite sure the functions of them in this game.

The role of each of the players was unclear to me. I didn't know what I was supposed to be doing and that we were even assigned different roles

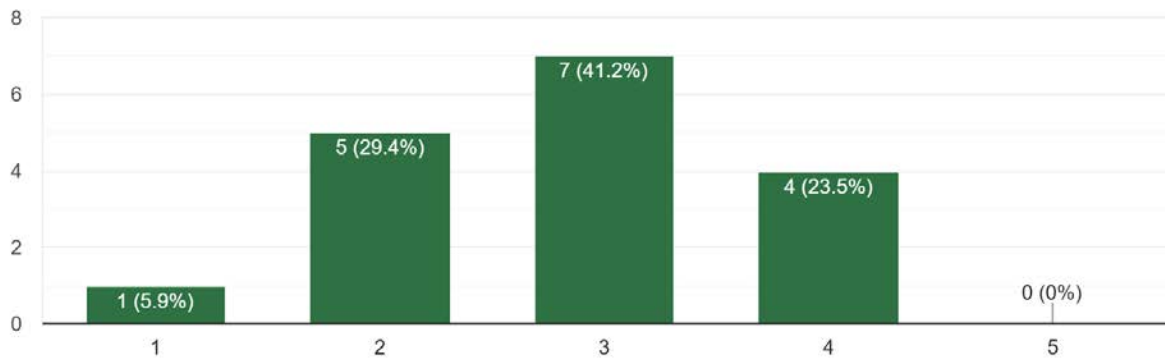
Objectives, possible tools

I thought that it was initially unclear that only one person would have seeds and the other would only have a watering can.

A lot of what the tools did was hard to keep track of after it was shown, was very easy to forget what everything did and I spent a good amount of time seeing what did what.

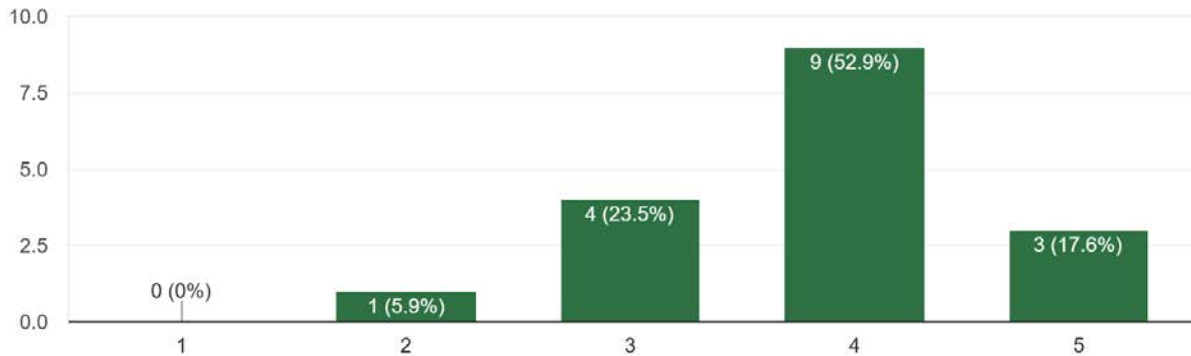
Were the functions of each tool clearly conveyed?

17 responses



How intuitive were the controls?

17 responses



If you wanted to alter the game's controls, how would you do so?

11 responses

Maybe making the swap seeds button use the triggers so you can go left and right through them.

Having the seeds and watering can only drop water/seeds when you hit the button instead of just toggling on and off, but that may be your plan already

I'm not sure how I would alter the controls, but I only used the bumpers and the left face button. I don't remember what the other buttons were for

Keyboard and mouse might easier for playing.

Maybe using the triggers would give a more tactile feel to the game

I wouldn't. The reason they were slightly unintuitive to me is that I don't play a lot of console games. Once I'd played for about 30 seconds, I'd gotten it.

I'm confused about the character(the blue guy). Do players need to control him finish something?

I think the controls are good, maybe no need to alter.

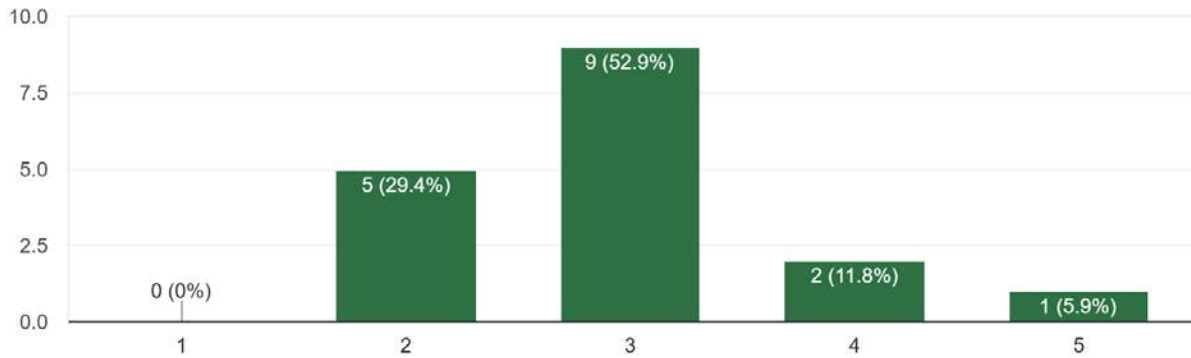
The controls were fine for me

More a button, less x button

I would have planting the seeds be the right trigger personally but that might just be because i am comfortable with fps on controllers

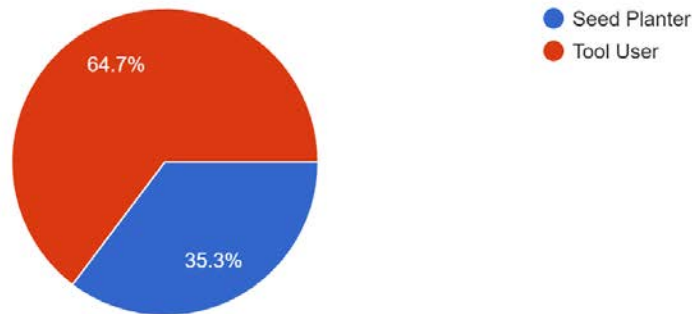
How did the player's cursor control feel?

17 responses



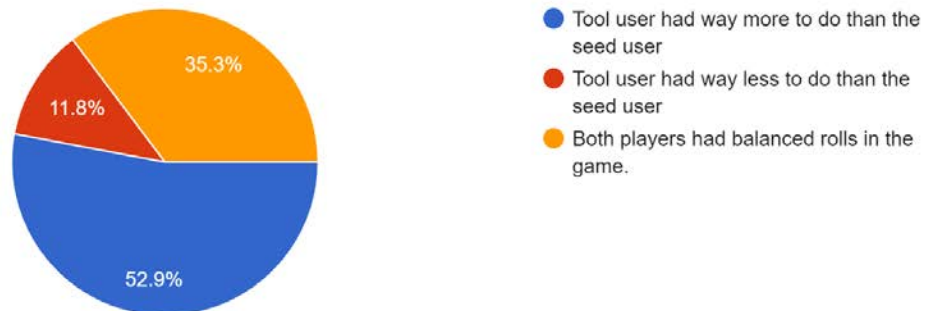
Did you play as the tool user or seed planter?

17 responses



Was the gameplay effectively divided between both the seed planter and the tool user?

17 responses



Any additional feedback regarding the multiplayer gameplay?

11 responses

Giving the seed planter extra things to do, or giving some tools of the tool user to the seed user like the ability to pick up plants, could help balance it.

Maybe clearer descriptions of the roles and what they mean would be helpful for coordination.

I played both rolls and found they were pretty equal. In the future, will cut plants have effects on the gnomes?

N/A

The cursors should be able to clip through the garden levels, if possible.

It needs some time for me to figure out which is my character.

Maybe the Gnome need to interact with plants

I was unsure of what my job was at the start and didn't know what I or my partner needed to do

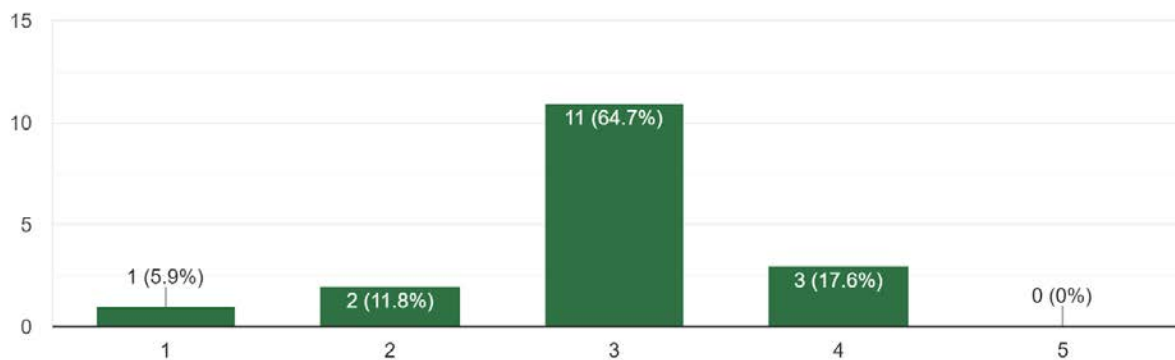
Give both players both tool sets, maybe? That way it isn't by chance which role you can play

I thought the idea of it being multiplayer was fun because it allows for the tasks to be split up and each person to have a unique role

In comparison, the tool user has far too many options compared to the seed planter, I feel like this game would benefit if both players had equal responsibility

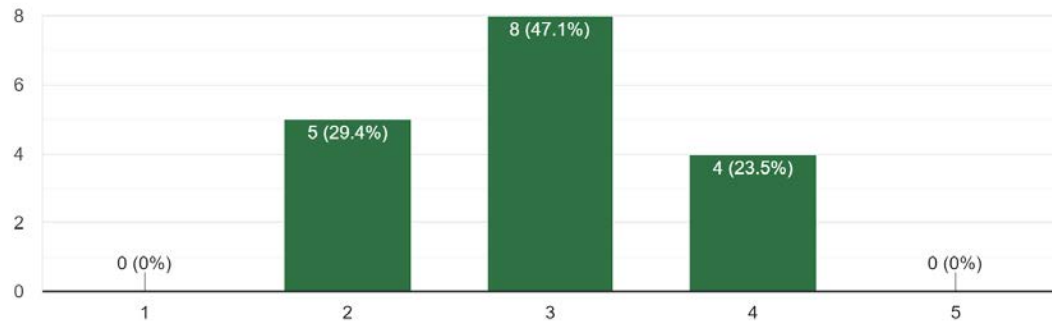
Was the amount of time given for one in-game round sufficient?

17 responses



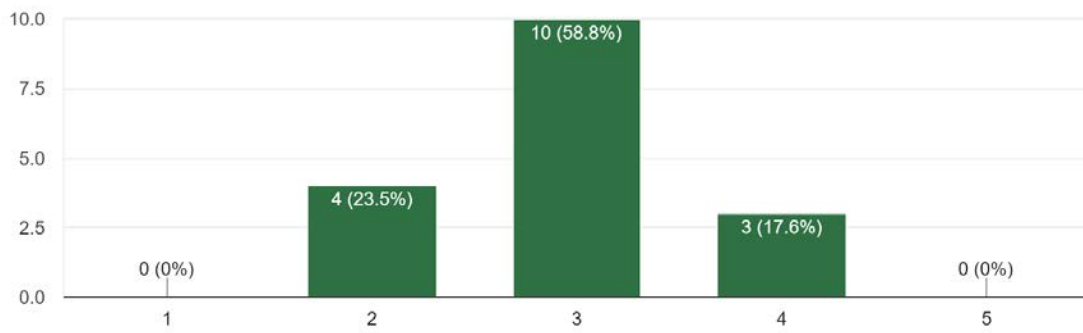
How would you rate the number of different plants?

17 responses



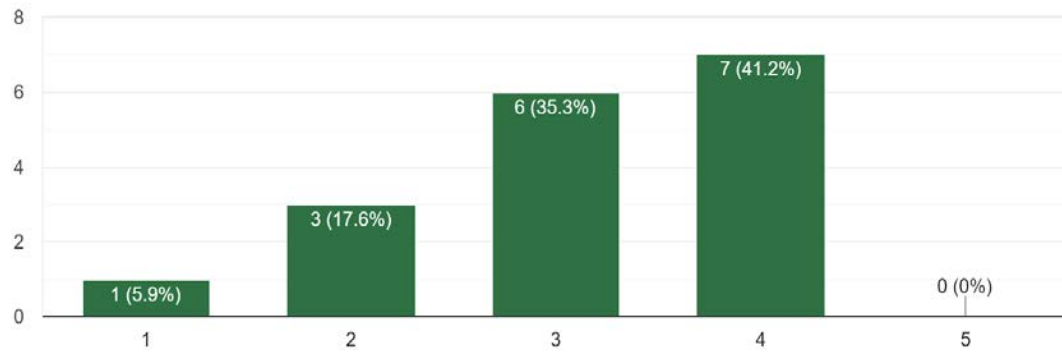
Was there enough variety between the different types of plants?

17 responses



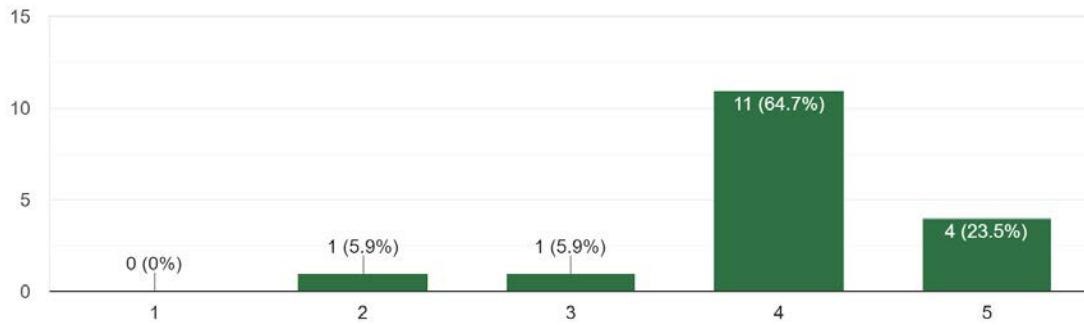
How do you feel about the game's art/aesthetics?

17 responses



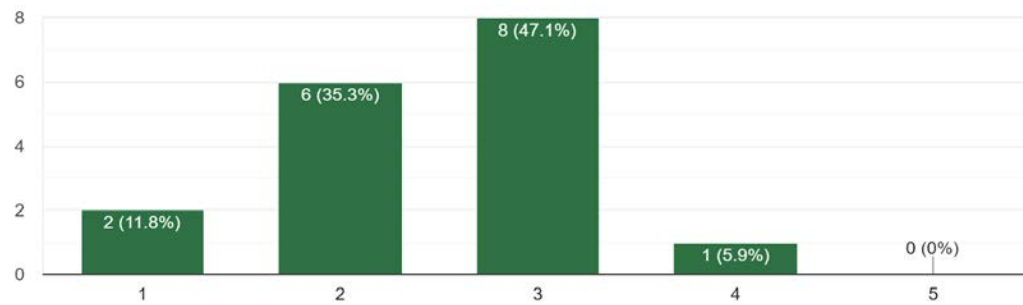
How would you rate the consistency of aesthetics between assets?

17 responses



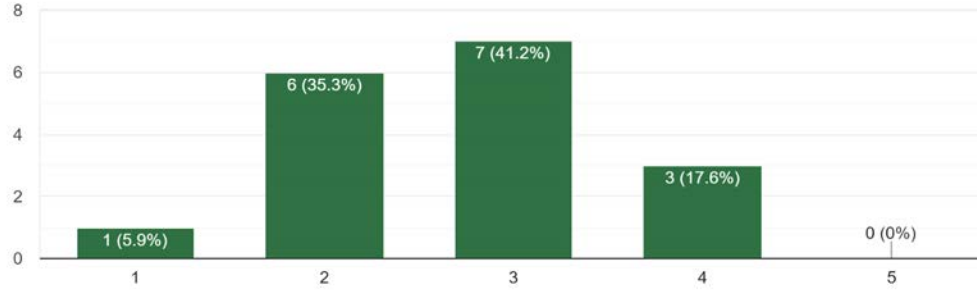
How would you rate the scale of the objects on the screen?

17 responses



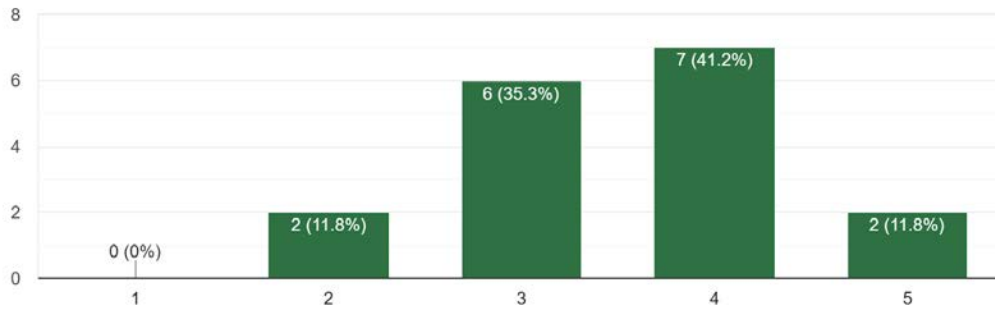
How were the game's colors when projected?

17 responses



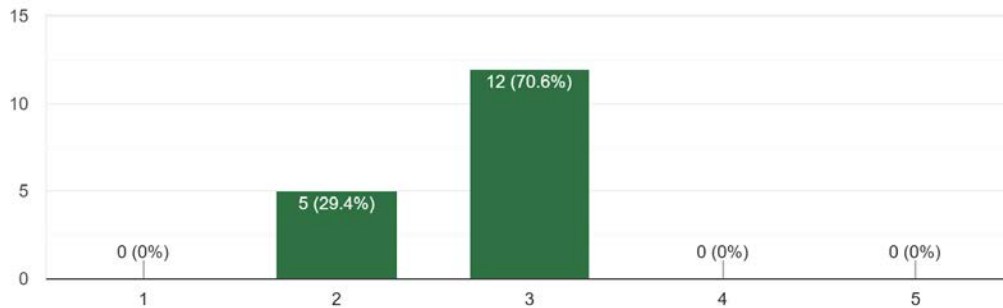
Do the animations fit their model's appearance?

17 responses



How would you rate the animation speeds?

17 responses



Any additional comments about the art and animations?

10 responses

More animations for the tools and their actions would help immensely. It is difficult to tell when a tool is properly working.

Some of the plants were overlapping each other which made it a little confusing and the gnomes were pretty small compared to the rest of the environment.

If the gnomes had a walk cycle, their movements would feel much more fluid than the sliding they are currently doing.

Good :)

No additional comments

The gnomes look slightly out of place because they're more detailed than everything else.

The water can be more water. Cubes make me confused.

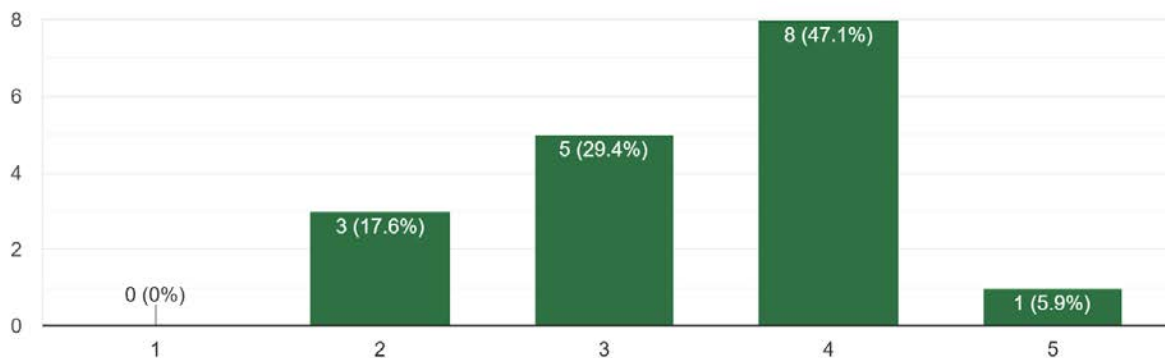
I think use low-poly style can make the whole game feel more cute and relaxing.

Wildly inconsistent quality, gnomes just sliding is strange

I liked the plant growing animations! Although they were a bit long in my opinion i thought that they were well done.

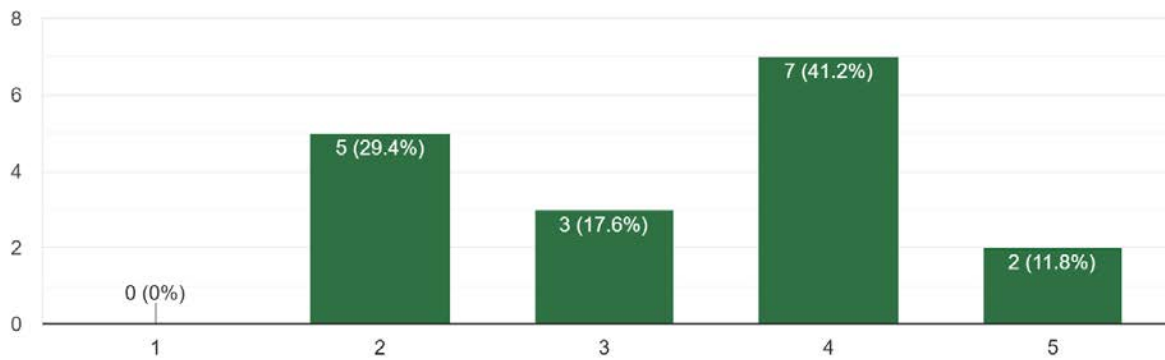
How well did the audio fit the player actions?

17 responses



Did the gnome and plant sounds fit their onscreen behavior?

17 responses



Any additional feedback about the audio?

9 responses

Some music could help with the soundscape of the game. Currently it is only sound effects that could get tiring for some players

it felt a little unresponsive with the animations of the objects but the sounds fit when they played.

The gnome sounds were funny but got a bit annoying after a while.

N/A

Couldn't really hear the audio too well if at all

Could barely hear anything except for the gnome grunts.

Need background music.

The hmms are very annoying and distracting.

fun noises and it added flavor to the game

What did you enjoy most about the game?

17 responses

The multiplayer aspect was interesting, and made for a silly yet relaxing experience

moving the gnomes

the sounds were pretty funny, I liked those

I enjoyed building the garden and think it was fun working together to make the plants grow.

I liked the sandbox feel a lot as well as the multiplayer aspect of it.

The gnomes were funny and I did find the planting satisfying

The idea is good. It would be great if this game aiming to kids.

It got chaotic in a fun way, like me and the other person filled the screen with plants

I enjoyed the teamwork with the planting and watering

I liked the idea of projecting the results and displaying them.

The cooperative play. I like having a role in a game and knowing what it is.

Finding different plants I can have.

Get different seeds and plants.

I liked the cooperative aspect of the game

plant growth

I liked the gnomes making sounds when you clicked on them and also the variety of plants was cool. I liked how they had unique looks and animations.

Experimenting seeing what did what was very fun, after the game ended I wanted to make sure I found out what everything did.

If you ran into any bugs, please share them below

13 responses

some trees spawned on top of each other

hard to select the start button

I'm not sure if it's a bug or not, but I noticed the water and fertilizer would get stuck on other plants and wouldn't make it all the way down to the troughs.

Sometimes plants would stack higher than the soil and would grow in the air.

Several.

collision issues with the platforms, some tools react weird to certain platforms

Picking up new seed orbs didn't work sometimes

Some issues grabbing new seeds/ some lag in player movement (probably just latency and not actually the game's fault)

Cursor still inside game after game over. Can't exit the game on my own. As the tool player, couldn't pick up gnomes at one point.

Swapping between the different seed types sometimes led to one of the bags being displayed much lower than the cursor and other plant types.

Super unresponsive controls initially, with movement not matching inputs

sometimes the new seed bubbles would get stuck in the platforms which made them hard to grab. Sometimes seeds would get stuck mid air above others

Some tools have collision that allow you to push the gnomes (unsure if this is a bug or a feature)

Open feedback: Let us know of anything you would add, whether it be new plants, new mechanics, you name it!

11 responses

more plant colors would be cool :)

maybe some way to combine plants together and form weird hybrids.

I think it would be cool if the more plants you plant, the more gnomes get attracted to your garden. Then, these new gnomes that come to your garden could also bring different types of plants that help expand your garden even further, and attract other gnomes that then also bring other plants. So it would kind of be a cascading type effect. All in all, I really like the game and the concept!

Clearer directions about what the goal is and how to achieve it, background music?

I think it would be great if this game separate into 2 screen for 2 players to compete. In stead of working together, player that plant more will be the winner.

Might be good to have more of a goal for the players to work for, like certain gnomes want certain plants within a time frame. Or have the goal expressed more clearly, I wasn't sure what I was working towards

Whimsical background music

Lotus.

Maybe you can add a score system, each of the plants can add some scores, player can use scores to change the weather or the color of Gnome.

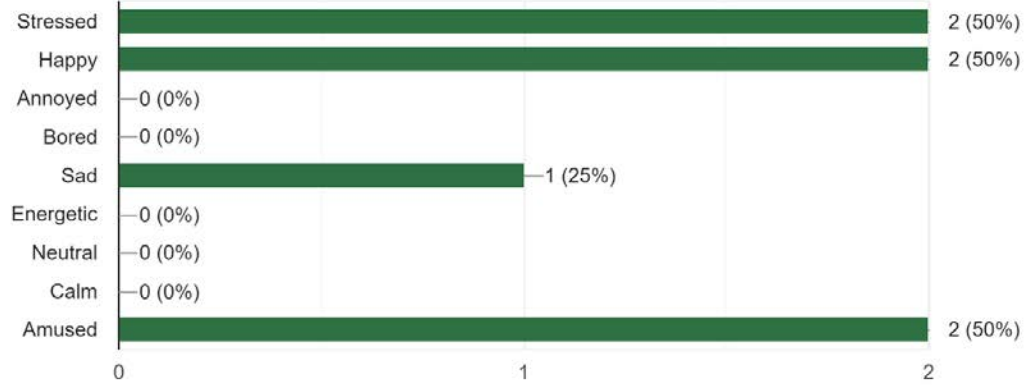
Why aren't the players gnomes, and if they are, how are you conveying that? Also planter beds look like roads with current coloration/contrast on screen

I think the game is fun one major thing i would add is more engaging background music to engage the plays while they play. I would add my flavor to the game in my opinion.

Appendix D: D-Term Survey with Responses

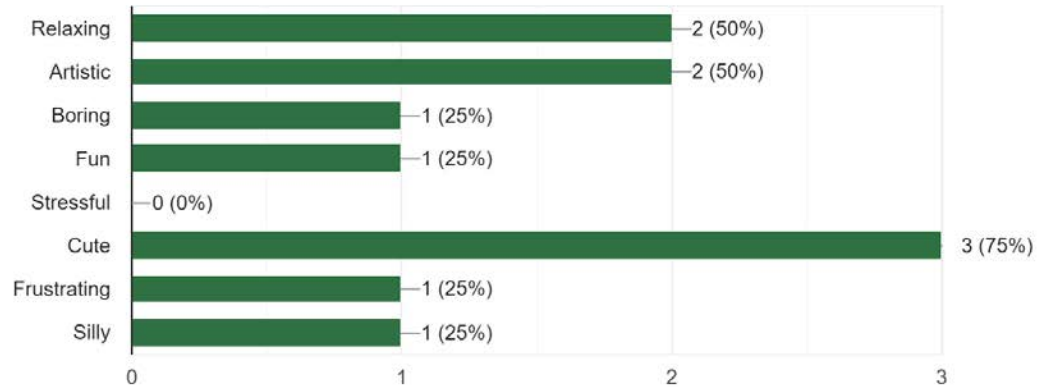
How did you feel before playing Gnome Hollow? Check those that apply.

4 responses



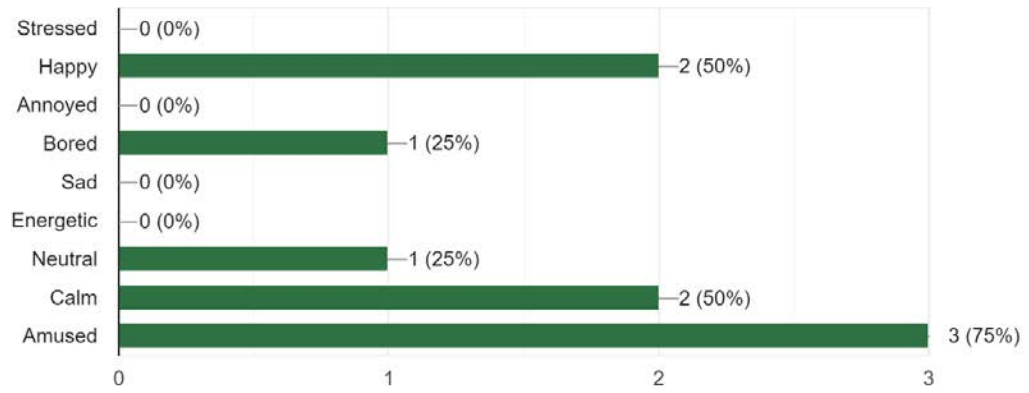
How would you describe your experience playing Gnome Hollow? Check the 3 that best apply

4 responses



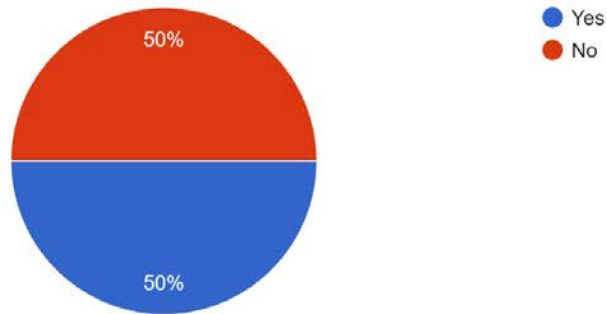
How did you feel after playing Gnome Hollow? Check those that apply.

4 responses



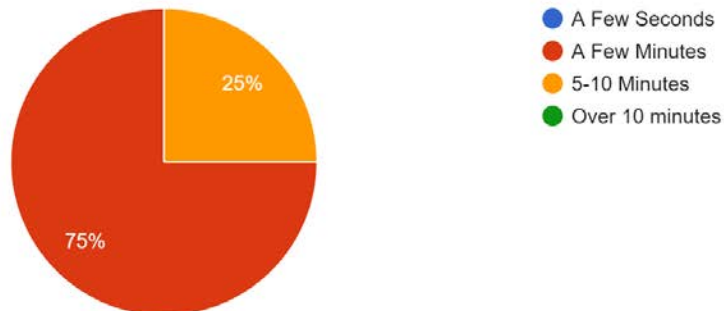
Gnome Hollow is a game without a goal or win condition. Was this clearly conveyed while playing?

4 responses



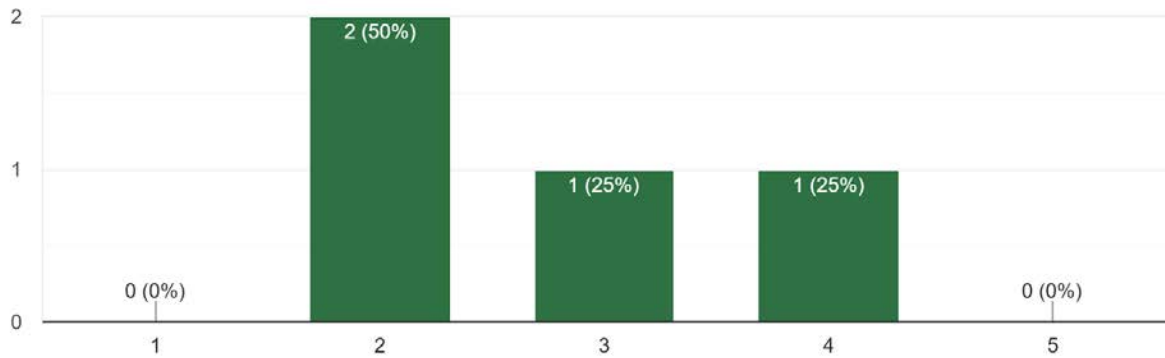
How long would you want to play this game?

4 responses



How effectively did the in-game tutorial options inform you about how to play?

4 responses



Were there any aspects of gameplay that were unclear? Please list them below.

4 responses

No

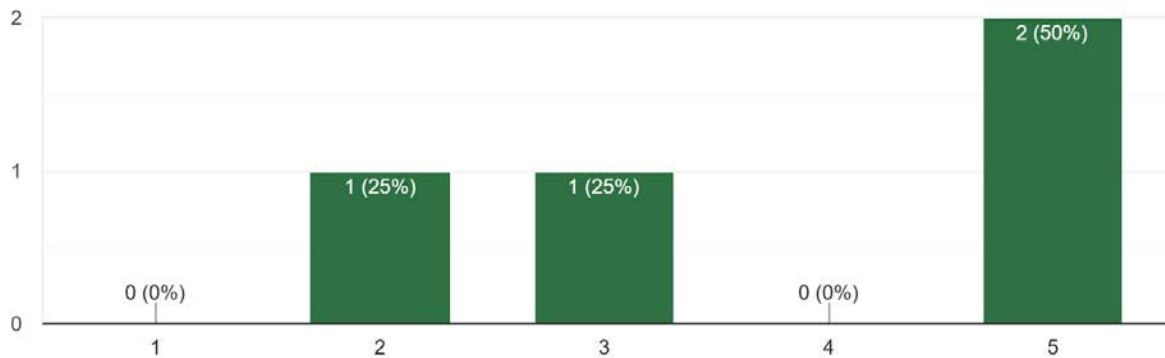
Mostly conveyance stuff that I talked to Garrett about: it's hard to tell why some things aren't placed/selected

I don't know if it was due to my ineptitude with a Mac or if something else was going on, but I think this was supposed to require 2 controllers? Probably? Wasn't entirely clear, but I grew mushrooms nonetheless.

This is me from the future after reading further in this survey, multiplayer is definitely a thing, neat.

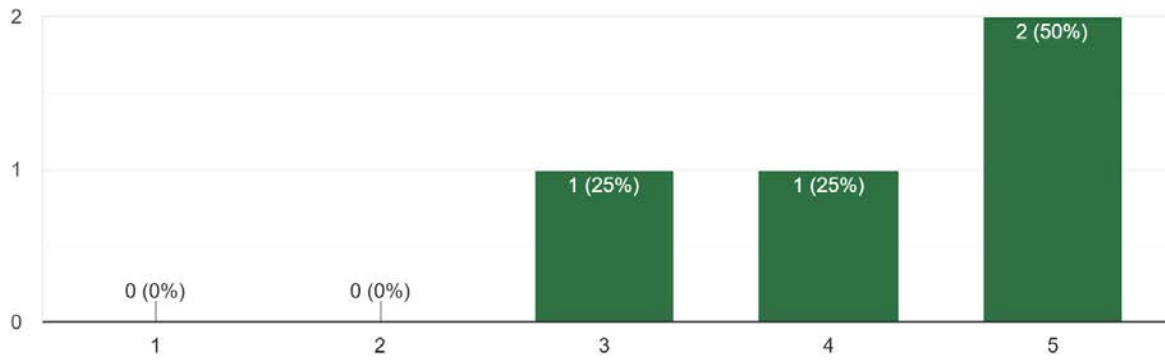
Were the functions of each tool clearly conveyed?

4 responses



How intuitive were the controls?

4 responses



If you wanted to alter the game's controls, how would you do so?

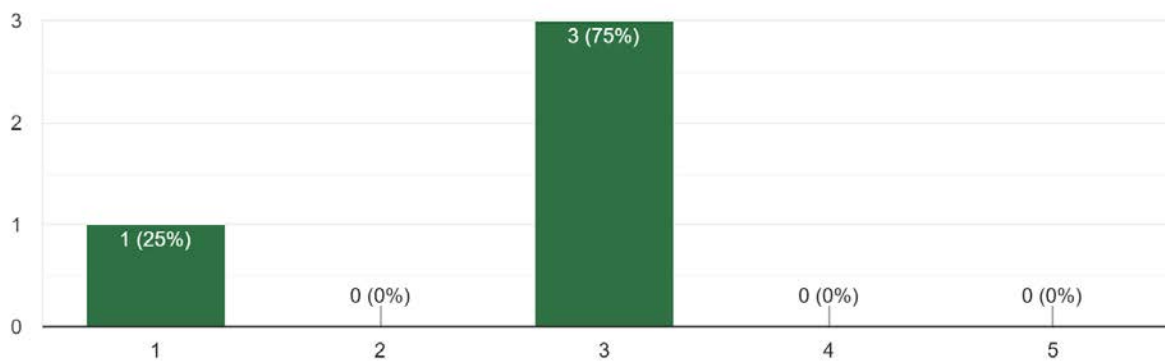
2 responses

I'd have one btn to toggle through all the tools

Using the numpad to swap between tools without a controller would be pretty cool.

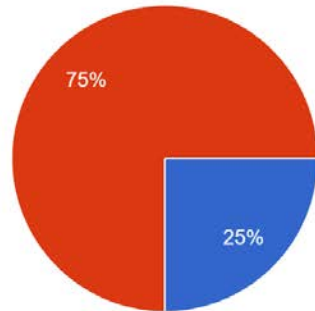
How did the player's cursor control feel?

4 responses



Did you play as the tool or plant player?

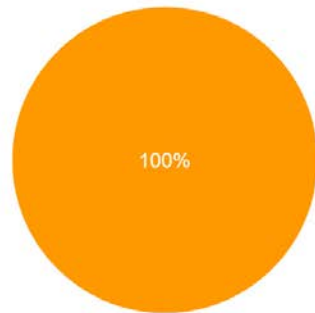
4 responses



- Plant Player (Seed Bag, Shears, Shovel)
- Tool Player (Glove, Watering Can, Fertilizer)

Was the gameplay effectively divided between both the plant and tool players??

4 responses



- Tool player had way more to do than the plant player
- Tool player had way less to do than the plant player
- Both players had balanced roles in the game.

Any additional feedback regarding the multiplayer gameplay?

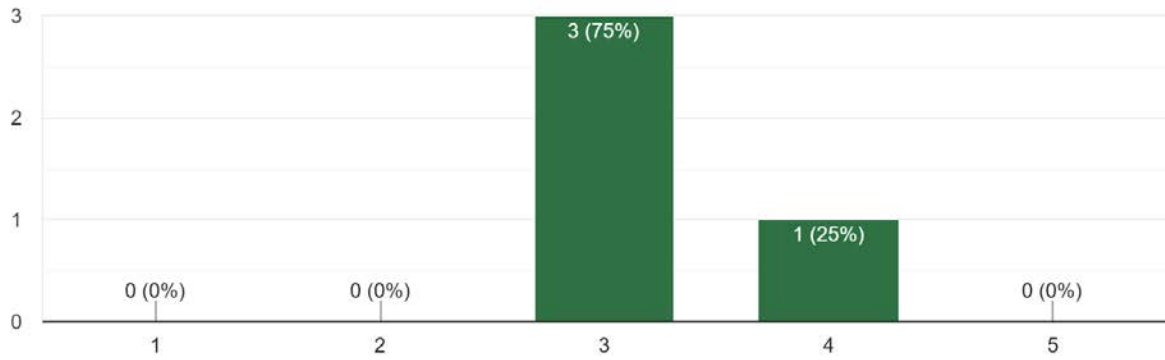
2 responses

I would consider making all options available to both players; theres a lot of creative exercise one player loses out on

I liked yeeting the gnomes with the glove

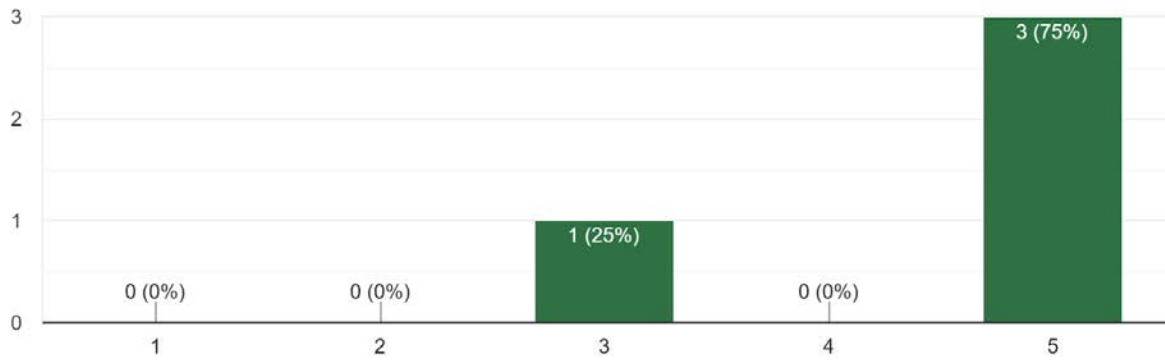
Was there enough variety between the different types of plants?

4 responses



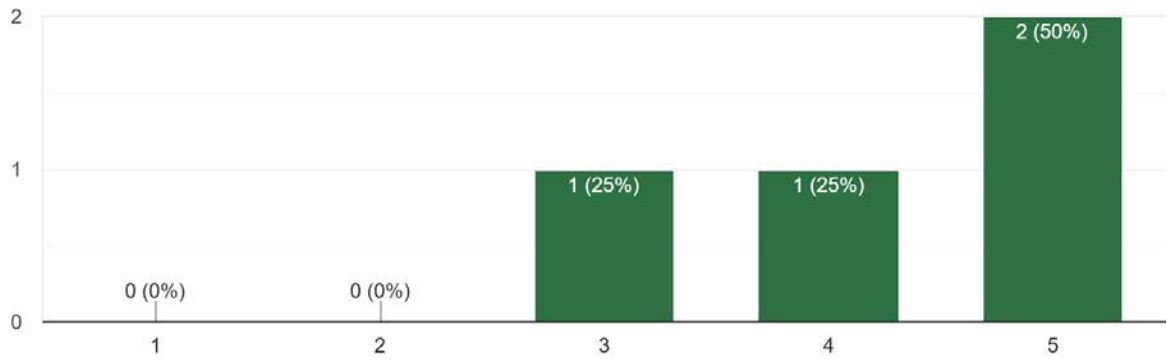
How do you feel about the game's art/aesthetics?

4 responses



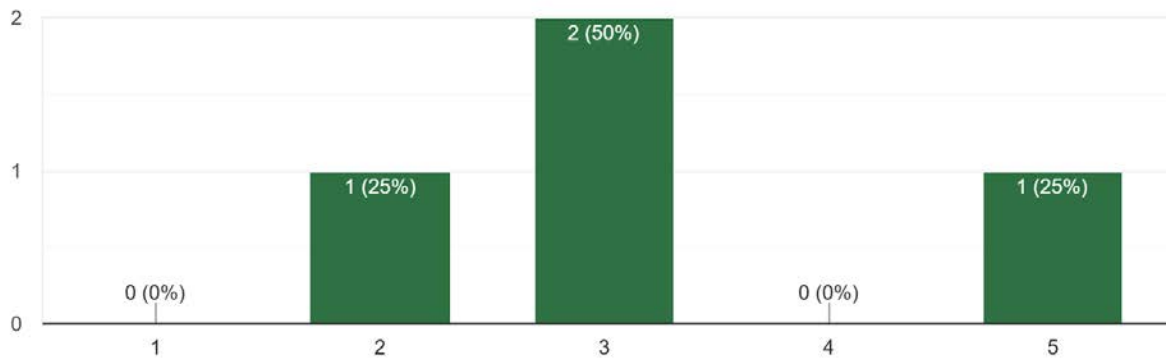
How would you rate the consistency of aesthetics between assets?

4 responses



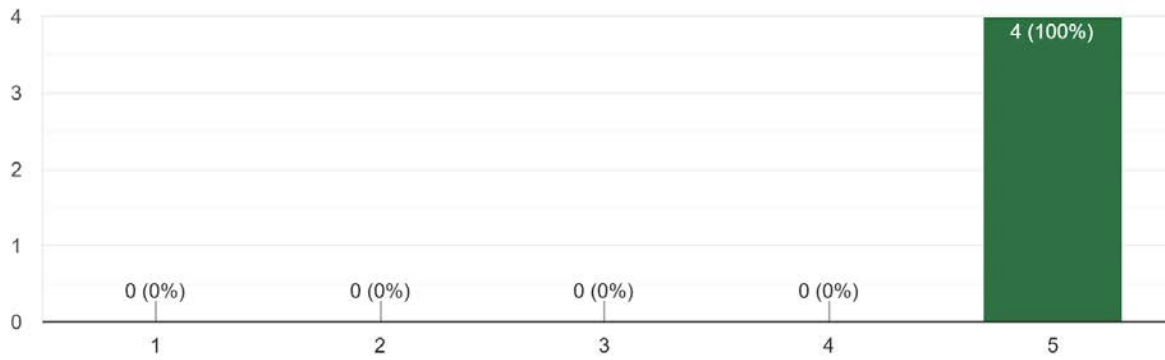
How were the game's colors when projected?

4 responses



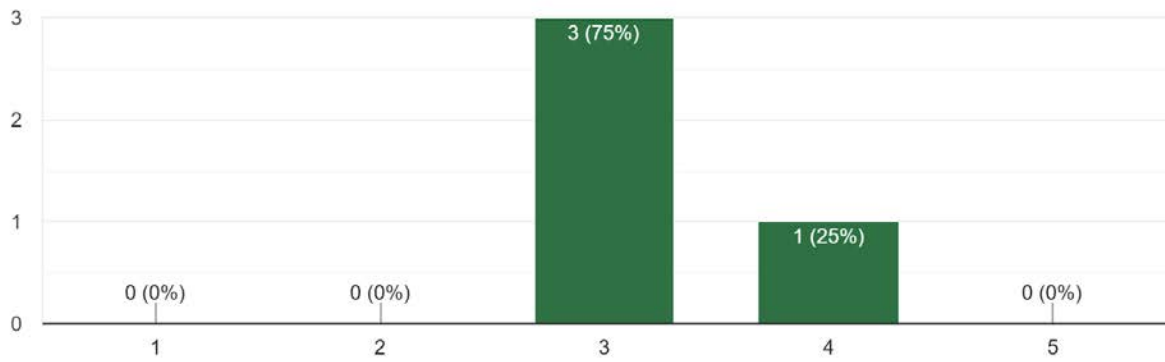
Do the animations fit their model's appearance?

4 responses



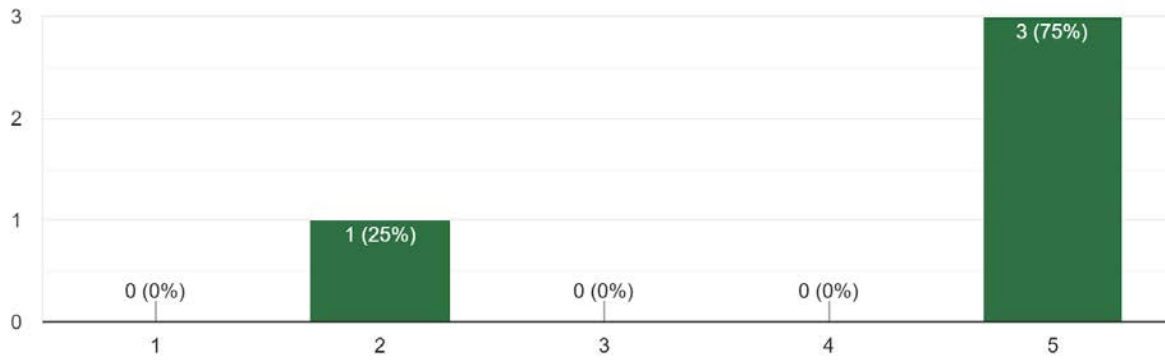
How would you rate the animation speeds?

4 responses



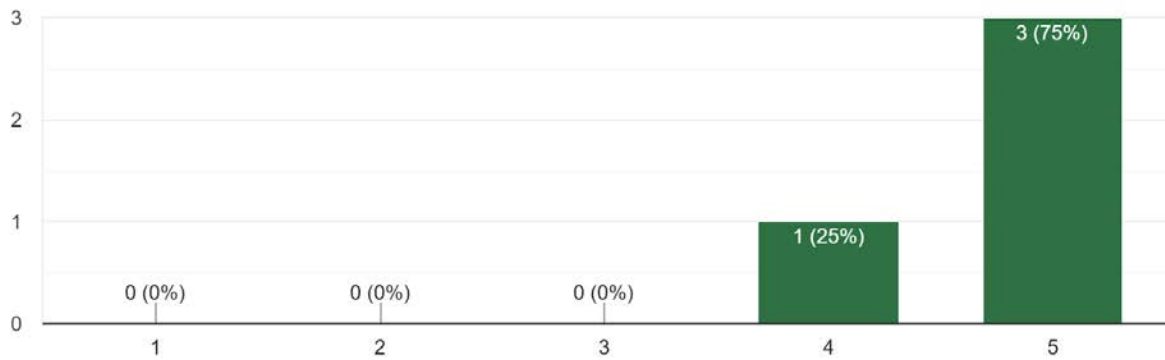
How clearly did the animations convey what was happening in game?

4 responses



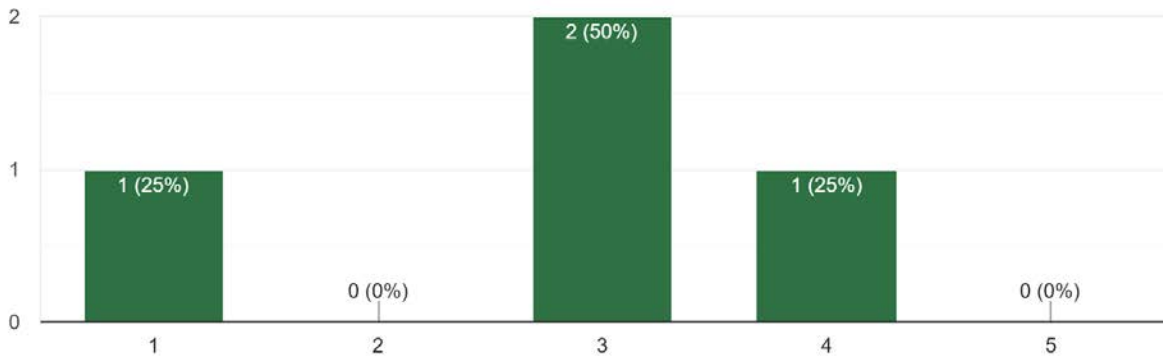
How well did the effects fit their plants?

4 responses



How was the size of the plant particle effects?

4 responses



Any additional comments about the art and animations?

3 responses

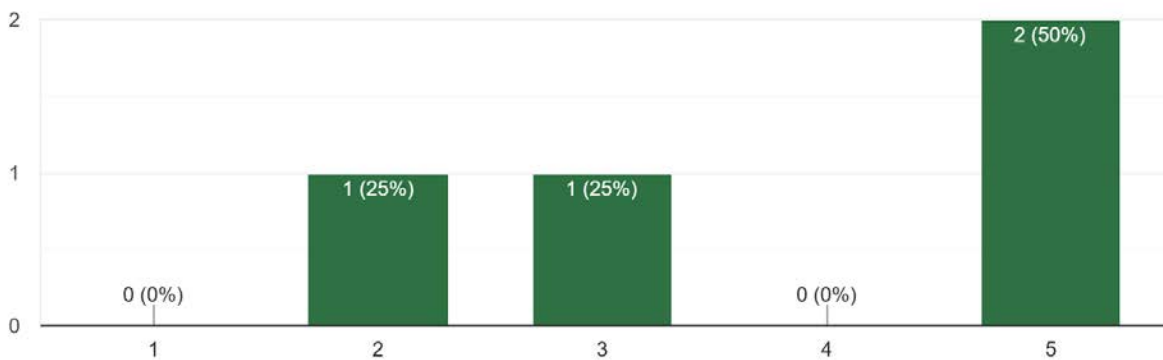
Seeds were hard to see, maybe make them brighter?

Consider some more natural lighting, idle animations, a grab animation, etc

The gnomes might've just been shy, but I don't think there was an animation for them turning around, so it looked a bit janky at times. That was part of the charm though. Gnomes are goofy, as is tradition.

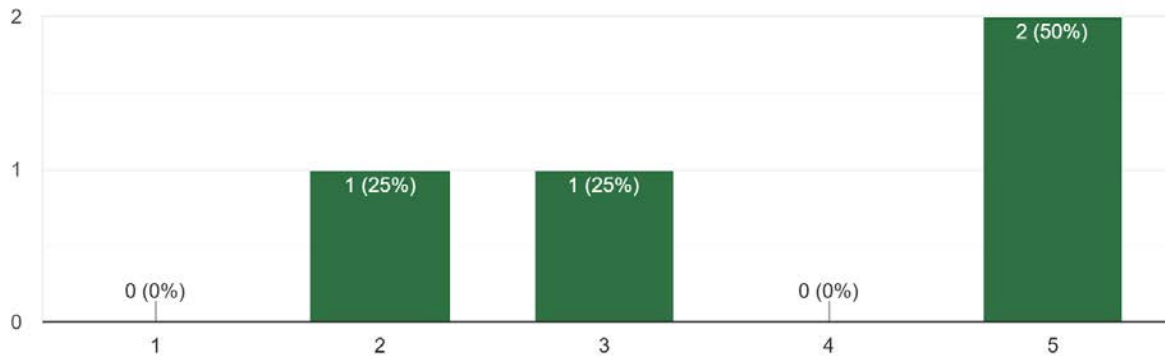
How well did the audio fit the player actions?

4 responses



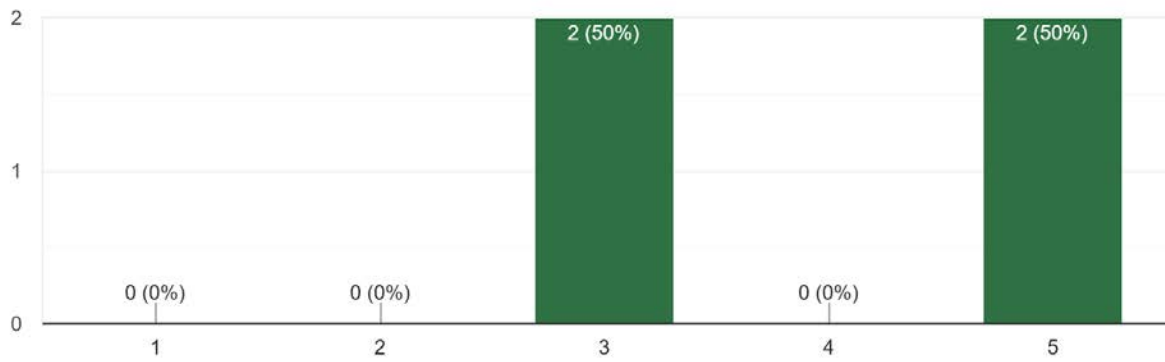
Did the gnome and plant sounds fit their onscreen behavior?

4 responses



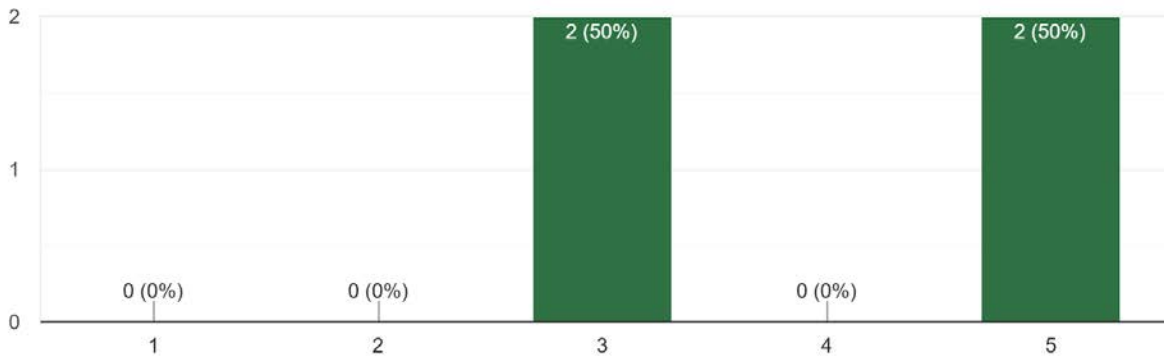
How well did the soundtrack fit the game?

4 responses



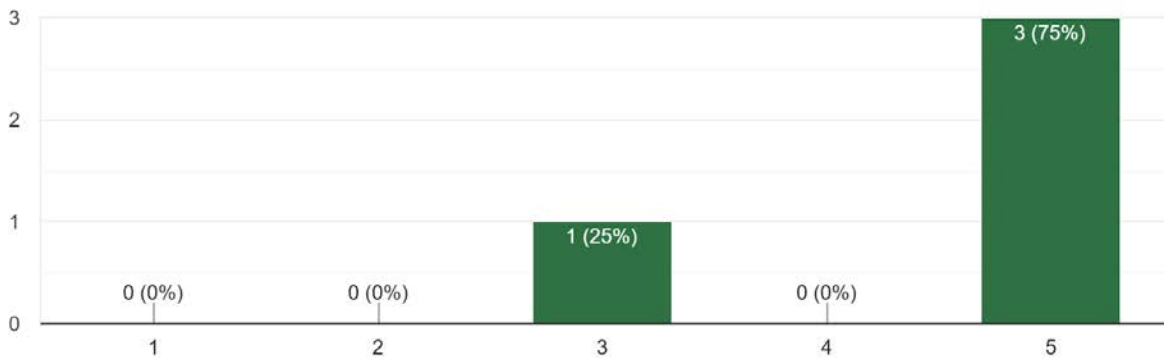
How well did the ambience fit the game?

4 responses



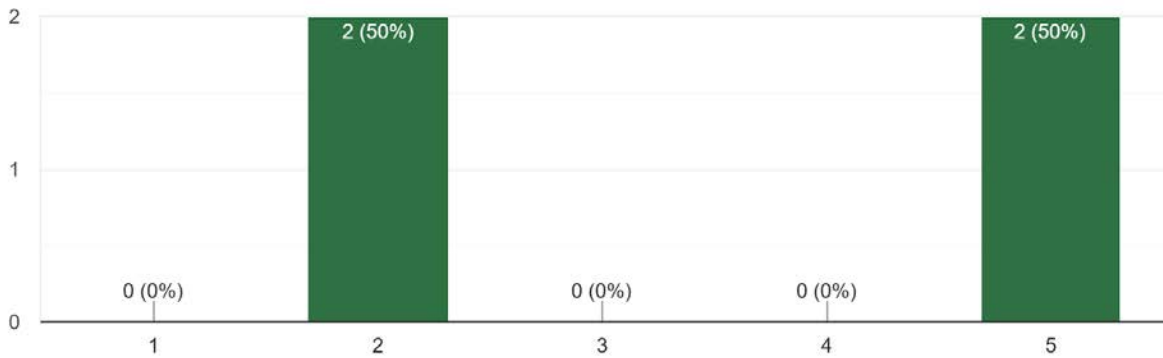
How well did the particle effect sounds effects fit?

4 responses



How was the audio mixing? (Volumes of each sound effect relative to other game sounds)

4 responses



If you answered below a 3 on the last question, what sounds were poorly mixed (too loud/quiet)?

2 responses

Music and some sounds too quiet

I couldn't really hear the audio

Any additional feedback about the audio?

0 responses

No responses yet for this question.

What did you enjoy most about the game?

4 responses

The animations

I like the gnomes running around the garden I've made. very toyetic

I like the art assets and animations

The satisfaction of figuring out how to get this game running on a Mac while also hooking up controllers to it to get to play the full game myself. Very satisfying. Gnomes are cool too.

If you ran into any bugs, please share them below

3 responses

Bags glitching together

Have trouble clicking on things....objects seem to get stuck on other objects...controls feel somewhat unresponsive

Could not pick up gnome, watering can and soil bag did not always work, seed bags did not see my to go away

Open feedback: Let us know your thoughts on anything!

2 responses

Conveyance conveyance conveyance!

You guys are pretty cool

Appendix E: Art Bible











Appendix F: Interview Questions

Intro Questions

- What kinds of projection projects have you worked on before? What has been your favorite project to work on?
- What is your design philosophy when it comes to projecting games/experiences?

General Questions

- How do you begin this type of project?
- What kind of equipment do you tend to utilize for projection-based games?
 - Are there any other tools aside from a projector that would be necessary to set up our scene?
- How do you choose a projector for each project?
- How did you incorporate body movement into gameplay? Was it a challenge to implement?
- How do you test your games before they are projected on a building?

Tech Questions

- What engine would be best for creating projection mapping games?
 - Do you know of any free and/or paid tools to aid the projection mapping process?
 - Are there any additional projection mapping development resources that you would recommend?
- What are the greatest obstacles to developing games utilizing projection mapping?

Art Questions

- How does projection affect art assets and their pipeline?
- How do you account for perspective distortion when projecting?
- Have you ever used 3D assets in your projection projects?
 - If so, do you have any advice for implementing them in a 2D projected space?

Audio Questions

- Do your projects include interactive audio?
- What are typical problems you run into when integrating audio?
 - What is your preferred middleware?
 - What type of speakers do you recommend?

Appendix G: Informed Consent Form

Investigator: Ralph Sutter

Contact Information: rsutter@wpi.edu

Title of Research Study: Gnome Hollow

Sponsor: WPI

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 obtain feedback on the project in order to facilitate design improvements and find/address operational bugs.

Procedures to be followed: You will be asked to play a brief game lasting less than ten minutes. After completing the game, you will be asked to complete a brief, anonymous survey describing aspects of your subjective experience.

Risks to study participants: There are no foreseeable risks associated with this research study.

Benefits to research participants and others: You will have an opportunity to enjoy and comment on a new game under active development. Your feedback will help improve the game experience for future players. Additionally, we will provide playtesting credit for any IMGD class that requires it.

Record keeping and confidentiality: Records of your participation in this study will be held confidential so far as permitted by law. However, the study investigators 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: There is no foreseeable risk of injury associated with this research study. Nevertheless, you do not give up any of your legal rights by signing this statement.

For more information about this research or about the rights of research participants, or in case of research-related injury, contact the Investigator listed at the top of this form.

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. 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.

_____ Date: _____
Study Participant Signature

Study Participant Name (Please print)

_____ Date: _____
Signature of Person who explained this study

Appendix H: Study Protocol

Title of Research Study: Gnome Hollow

Purpose of study

To obtain user feedback in order to determine if experience goals are being achieved, locate operational bugs, and identify opportunities for design improvement.

Study protocol

Participants are directed to a Web URL where they view the Opening Briefing (below), complete the Informed Consent Agreement and download the game software.

After completing the game, participants are directed to another URL and asked to fill out a short survey to characterize aspects of their subjective experience and solicit suggestions for improving the experience.

Opening briefing for testers (provided online)

“Hello, and thank you for volunteering to test our project. Before we begin, please read and sign this Informed Consent form. Once you have completed the form, you will be asked to play our game. When your session is complete, we will ask you to complete a brief survey about your play experience. At no point during your test session, or in the survey after, will any sort of personal and/or identifying information about you be recorded. You may back out of playing at any time. Please begin playing when you feel ready.”

Appendix I: Survey Draft

1. How engaging was the gameplay loop?

[Lickert scale 1-5] 1 = Not Engaging, 5 = Very Engaging

(Optional) Comments _____

2. How would you rate the speed of the player's cursor?

[Lickert scale, 1-5] 1 = Too Slow, 5 = Too Fast

(Optional) Comments _____

3. How do you think the audio fit each action?

[Lickert scale, 1-5] 1 = Did Not Fit, 5 = Perfect Fit

(Optional) Comments _____

4. If you answered "Did Not Fit" for the previous question, what about the audio was off?

Response: _____

5. How long would you want to play this prototype?

1. A Few Seconds

2. A Few Minutes

3. 5-10 Minutes

4. Over 10 Minutes

(Optional) Comments _____

6. How would you describe your experience? Check all that apply.

• Relaxing

• Artistic

• Boring

• Fun

• Stressful

• Cute

• Frustrating

(Optional) Comments _____

7. How intuitive was the game's control scheme?

[Lickert scale, 1-5] 1 = Not Intuitive, 5 = Very Intuitive

(Optional) Comments _____

8. How would you rate the scale of the objects on the screen?

[Lickert scale, 1-5] 1 = Not Intuitive, 5 = Very Intuitive

(Optional) Comments _____

9. How clear is a tool's function based on its appearance?

[Lickert scale, 1-5] 1 = Not Clear, 5 = Very Clear

(Optional) Comments _____

10. How would you rate the consistency of aesthetics between assets?

[Lickert scale, 1-5] 1 = Not Consistent, 5 = Very Consistent

(Optional) Comments _____

6

11. How do you feel about the game's art/aesthetic?

[Lickert scale, 1-5] 1 = I Hate It, 5 = I Love It

(Optional) Comments _____

(Optional) Please add any general comments and suggestions here.
