Mushroom Kid’s Big Grass Sword

Credits

John Haley - Amanda Rowe - Liam Armitage - Twig Sheherlis - Tyler Samay - Andrew Ptaszek - Ethan Lin - Quinne Houck - Enzo Rizzuto

Table of Contents

1. Introduction

This document outlines every design choice for the game, making it easier to review concepts and streamline development. It serves as a guide to ensure that the game's development aligns with its original vision.

Please note that this is a living document, meaning its contents can change as the game evolves. Regular reviews are recommended to keep the game's development on track with its design objectives.

1.1 Game Summary

Mushroom Kid’s Big Grass Sword is a wacky narrative puzzle platformer about a little mushroom kid named Mica on their journey to save their village with a sword that grows!

1.2 Game Pillars

- **Whimsical**
- **Characters**
- **Traversal**
- **Journey**

1.3 Scope

Mushroom Kid's Big Grass Sword should only have **6 hours or less of gameplay MAX.** It should focus on only telling **one** complete story about **one** interesting mechanic and taking it as far as we can go.

This is so production time of the game does not grow exponentially, but also gives the player what they want. **If they find a game about a funny mushroom with a giant sword we should just deliver them a funny mushroom with a giant sword.** No need to do anything else.

If the game gets popular then this can change, but until then we should only focus on the **CORE EXPERIENCE.**

1.4 Inspiration

Mushroom Kid's Big Grass Sword's sword swinging mechanic is inspired by games like **Deepest Sword** and **Getting Over It with Bennett Foddy**, having the player's main form of traversal be swinging a giant object (like Getting Over It) and having the object grow over time (like Deepest Sword).

As for the narrative structure and core progression of the game, our team wants to take inspiration from games like **Owlboy** and **Thomas Was Alone** that refine their presentation and experience to the core essentials so it's fun all the way through.

This is also inspiring our game's shift from a more rage-type game to a puzzle platformer, as we believe that our audience will be more receptive to that form of gameplay.

2. Technical Design

Anything involving more of the programming and hardware side of development, as well as production standards.

2.1 Project Overview

- **Platform:** PC & Nintendo Switch

- **Controls**: Keyboard & Mouse / Controller
- **Players**: Single-Player
- **Genre**: Puzzle Platformer
- **Engine**: Godot
- **Other Tools**
    - Dialogic 2.0
    - Wwise

2.2 Engine of Choice

2.2.1 Why are we now considering other game engines

Due to some recent executive decisions made from Unity:
https://www.axios.com/2023/09/13/unity-runtime-fee-policy-marc-whitten
https://arstechnica.com/gaming/2023/09/game-developers-unite-against-unitys-new-per-install-pricing-structure/
https://www.gamedeveloper.com/business/the-death-of-unity

It has started to become apparent that Unity as a company is starting to lose the trust they built within the game dev scene, as they continue to make decisions that retroactively hurt projects that are already in development and hurt the indie game development scene as a whole.

Although we don't expect our game to ever reach a point where these new monetization features become an issue, it still hurts our project and future career ambitions if Unity continues to hurt their image. Especially in the case of trying to pitch this game to a publisher:
https://twitter.com/devolverdigital/status/1701685282129539485?s=46&t=Qas3rpFq9mLve4J9zuOySQ

Thus, we have decided to look into the up and coming strong, open source game engine: Godot. It recently had a huge update that majorly updated its rendering engine making it an interesting alternative as a 2D Game Engine.

2.2.2 Final Decision

Based on the research done here: Game Engine Switching Analysis

Godot as an engine provides us with **plenty of alternatives to the packages we were going to use** in Unity and may even end up having a **faster production schedule due to its fast 2D art and animation pipeline and its quick code compile speed and game testing features.** We estimate that it will **only take us about a week to fully grasp the program** and feedback from **our Godot contact and lead programmer confirms this.**

We also believe that **the type of game we are creating and the features we have planned lends itself well to Godot's pipeline** and **we don't see any impending development issues facing us for the features we wish to implement.** At least, none that we wouldn't already have to deal with in unity.

It definitely **sucks that Godot's console porting features are not as strong as Unity's porting** but that is an **issue that is far in the future and doesn't affect production time.**

Therefore our **FINAL DECISION** is that if Unity does not **revert ALL OF ITS DECISIONS** by **monday morning on 9/18/2023** then we will be using **Godot 4.1.1 from then on.** The most important issue that needs to be reverted is **Unity's non-retroactive TOS github repository needs to be reinstated. Without it we can't trust Unity to not make a rash decision like this again in the future.**

This decision may even end up as a blessing in disguise considering all of the **production time that Godot's workflow may end up saving our project.**

We also will be choosing to use **Godot 4.1.1 because its usage of the Vulkan engine** will allow for the **best visuals for our game** and **most of the add-ons we are using as an alternative to Unity are fully implemented into Godot 4.1.1.**

We will also be developing most of the game in **GDScript** since it's **fast compile and testing speeds and relatively tiny performance change from C# allows for a faster workflow.**

2.3 Development Requirements

This section describes all needed tools and software that will be required for the game's development:

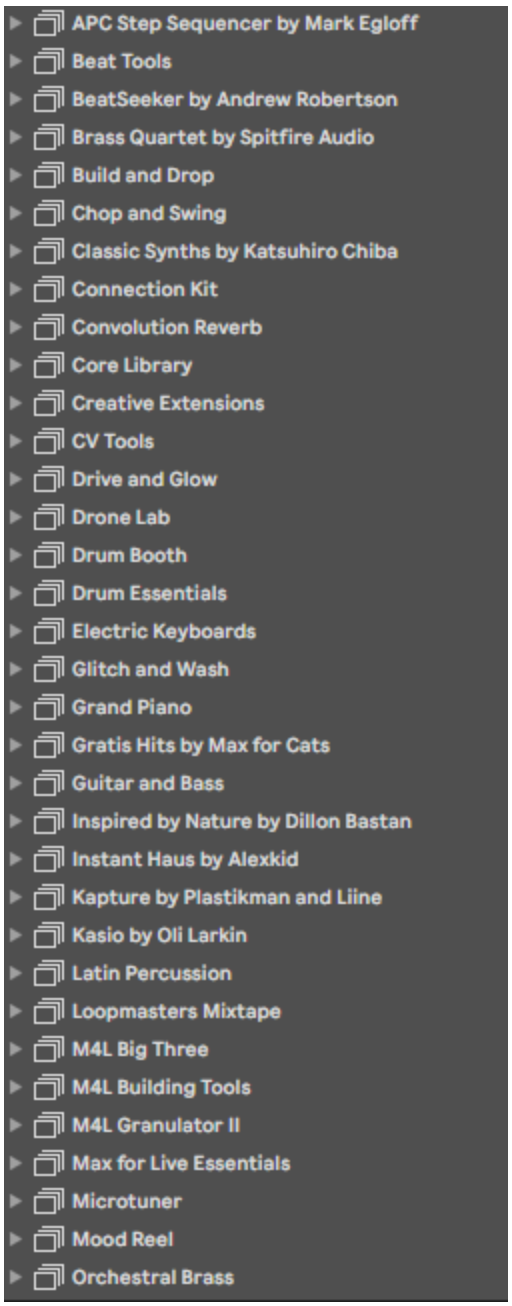
Aspect	Tool/Software
Engine	**Godot 4.1.1**
Documentation	**Google Doc**
Programming IDE	**Visual Studios 2022, Visual Studio Code**
Programming Language	**GDSCript, C#**
Project Management	**Trello, Google Spreadsheet, Discord**

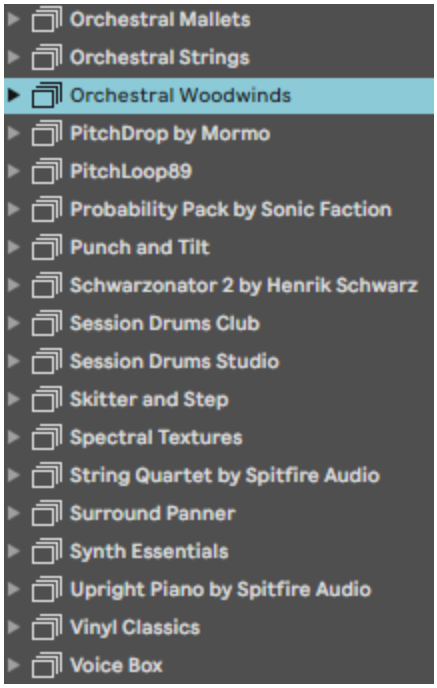
Communication	**Discord**
Source Control	**Git, GitHub Desktop**
Narrative	**Dialogic**
Art	**Aseprite, Clip Studio Paint, Pixel Studio, Procreate**
Marketing	**Adobe Photoshop, Adobe Premiere**
Distribution	**Steam**
Quality Assurance	**Google Forms**
Digital Audio Workstations	**Ableton Live 11 Suite, Logic Pro, FL Studio, Pro Tools,**
Vst2/3s & Plugins	***Bitsonic*** **Keyzone Classic** ***Mastrcode Music*** **T-Force Alpha Plus x64 bit ver** ***Izotope Plugins*** **Audiolens** **R2 (Exponential Audio)** **Excalibur(Exponential Audio)** **PhoenixVerb(Exponential Audio)** **iZotope Visual Mixer** **iZotope Relay** **iZotope Vocal Doubler** **Trash 2** **iZotope Elements Suite 6** **Iris 2** **Vinyl** **Nectar 3 Elements** **Neutron 3 Elements** **Neutro 3 Visual Mixer** **RX series** • **De-click** • **De-hum** • **De-clip** • **Voice De-noise**

	iZotope Ozone Imager 2 **iZotope Ozone 9 Elements** ***<u>Native Instruments Plugins</u>*** **Balinese Gamelan** **Community Drive** **Ethereal Earth for Soundwide** **Hybrid Keys Gift 2021** **India** **KOMPLETE START - Blocks Base** **KOMPLETE START - Expansions Selection** **KOMPLETE START - Guitar Rig 5 Factory Selection** **KOMPLETE START - Kinetic Treats** **KOMPLETE START - Komplete Kontrol** **KOMPLETE START - Kontakt Factory Selection** **KOMPLETE START - Mikro Prism** **KOMPLETE START - Play Series Selection** **KOMPLETE START - Reaktor Factory Selection** **KOMPLETE START - Supercharger** **KOMPLETE START - TRK-01 Bass Middle East** **ProjectSAM - The Free Orchestra** **Traktor DJ 2** **Twenty Five** **West Africa** **Cuba** **East Asia** **Yangqin** **Ableton Sound** ***<u>KiloHearts Essentials</u>*** **3-Band EQ** **Bitcrush** **Channel Mixer** **Chorus** **Comb Filter** **Compressor** **Delay**

	Distortion **Dual Delay** **Dynamics** **Ensemble** **Filter** **Flanger** **Formant Filter** **Frequency Shifter** **Gain** **Gate** **Haas** **Ladder Filter** **Limiter** **Nonlinear Filter** **Phase Distortion** **Phaser** **Pitch Shifter** **Resonator** **Reverb** **Reverser** **Ring Mod** **Stereo** **Tape Stop** **Trance Gate** **Transient Shaper** ***Plugin Alliance*** **Ampeg SVTVR Classic** **Black Box Analog Design HG-2** **Bx-cleansweep V2** **Bx-console focusrite SC** **Bx-masterdesk** **Bx-oberhausen** **Bx-rockrack V3 player** **Bx-solo** **Bx-subfilter** **Elysia niveau filter** **Schoeps Double MS** **Shadow Hills Mastering Compressor** **SPL free ranger** **WEDGE FORCE Matcha** ***Spitfire Audio***

	BBC Symphony Orchestra **LABS** ***Surge Synth Team*** **Surge XT** **Surge XT Effects** ***Vital Audio*** **Vital** ***Daichi Laboratory*** **Synth1**
Audio Sample & Packs	***Ableton Sound Packs*** **(Ill come back to this.)** **A picture as a placeholder**

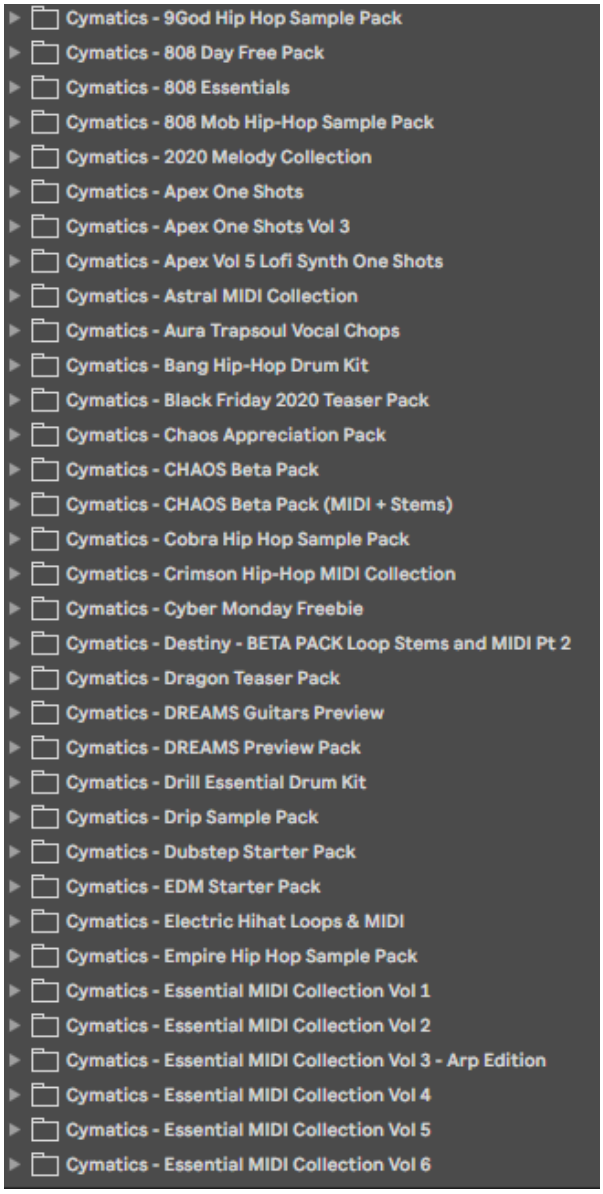
	 APC Step Sequencer by Mark Egloff Beat Tools BeatSeeker by Andrew Robertson Brass Quartet by Spitfire Audio Build and Drop Chop and Swing Classic Synths by Katsuhiro Chiba Connection Kit Convolution Reverb Core Library Creative Extensions CV Tools Drive and Glow Drone Lab Drum Booth Drum Essentials Electric Keyboards Glitch and Wash Grand Piano Gratis Hits by Max for Cats Guitar and Bass Inspired by Nature by Dillon Bastan Instant Haus by Alexkid Kapture by Plastikman and Liine Kasio by Oli Larkin Latin Percussion Loopmasters Mixtape M4L Big Three M4L Building Tools M4L Granulator II Max for Live Essentials Microtuner Mood Reel Orchestral Brass

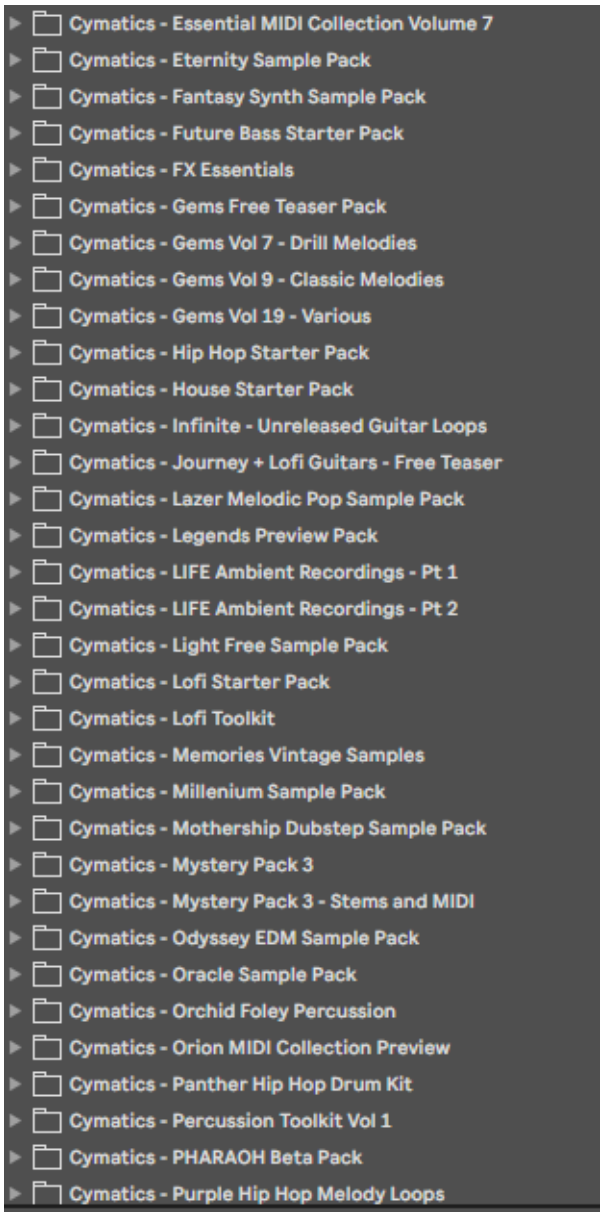


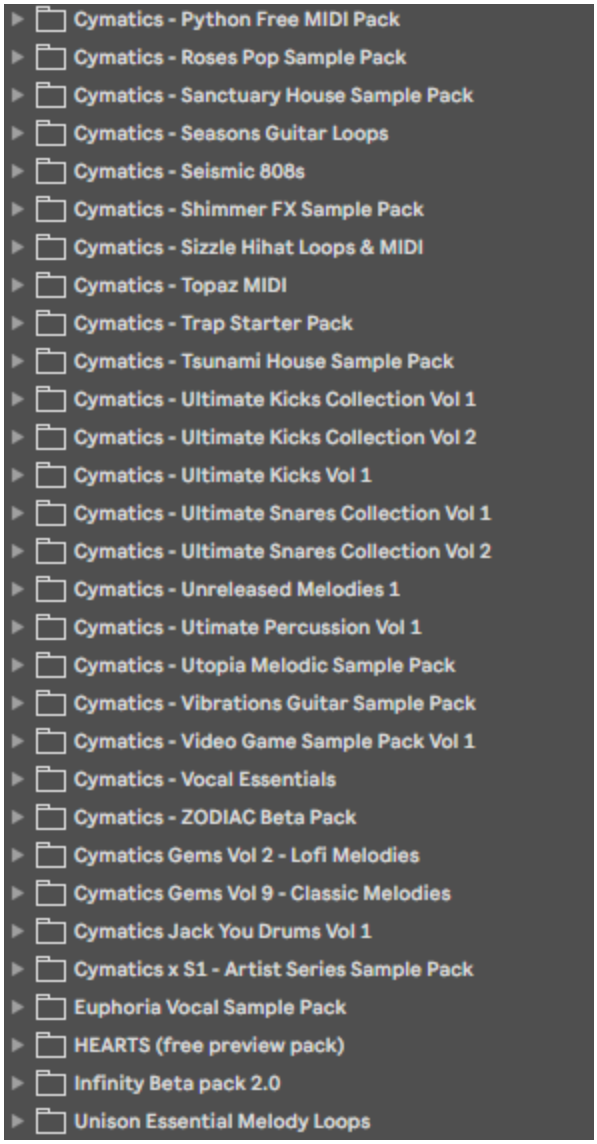
Chop and Swing
Drum Essentials
Grand Piano
Guitars and Bass
Skitter and Step
Beat Tools
Build and Drop
Drive and Glow
Drum Booth
Electric Keyboards
Glitch and Wash
Latin Percussion
Orchestral Brass
Orchestral Mallets
Orchestral Strings
Orchestral Woodwinds
Punch and Tilt
Session Drums Club
Session Drums Studio
Singularities
Synth Essentials
Mood Reel
Brass Quartet
Drone Lab
Inspired by Nature
String Quartet
Upright Piano

Voice Box

Cymatics Audio



	 Cymatics - Essential MIDI Collection Volume 7 Cymatics - Eternity Sample Pack Cymatics - Fantasy Synth Sample Pack Cymatics - Future Bass Starter Pack Cymatics - FX Essentials Cymatics - Gems Free Teaser Pack Cymatics - Gems Vol 7 - Drill Melodies Cymatics - Gems Vol 9 - Classic Melodies Cymatics - Gems Vol 19 - Various Cymatics - Hip Hop Starter Pack Cymatics - House Starter Pack Cymatics - Infinite - Unreleased Guitar Loops Cymatics - Journey + Lofi Guitars - Free Teaser Cymatics - Lazer Melodic Pop Sample Pack Cymatics - Legends Preview Pack Cymatics - LIFE Ambient Recordings - Pt 1 Cymatics - LIFE Ambient Recordings - Pt 2 Cymatics - Light Free Sample Pack Cymatics - Lofi Starter Pack Cymatics - Lofi Toolkit Cymatics - Memories Vintage Samples Cymatics - Millenium Sample Pack Cymatics - Mothership Dubstep Sample Pack Cymatics - Mystery Pack 3 Cymatics - Mystery Pack 3 - Stems and MIDI Cymatics - Odyssey EDM Sample Pack Cymatics - Oracle Sample Pack Cymatics - Orchid Foley Percussion Cymatics - Orion MIDI Collection Preview Cymatics - Panther Hip Hop Drum Kit Cymatics - Percussion Toolkit Vol 1 Cymatics - PHARAOH Beta Pack Cymatics - Purple Hip Hop Melody Loops



BlueZone corp

Future Possible libraries

Ethan's Own Recorded Samples

2.4 Project Structure

2.4.1 Folder Structure

The following tables describe the project structure and naming format for the game's folder's in-engine.

Directory	Files
Assets\Sprites	**Sprites**
Assets\Audio\Music	**Music**
Assets\Audio\SFX	**Sound Effects**
Assets\Fonts	**Fonts**
Assets\Tiles	**Tilesets**
Assets\Shaders	**Shader Files .gdshaders Files**
Scenes\Characters	**Characters**
Scenes\Levels	**Levels**
Scenes\Levels\Animations	**Animations**
Scenes\Levels\Rooms	**Rooms**
Menu	**HUD**
Menu\Pause	**Pause Menu**
Scripts	**Scripts**
Dialogue\Timelines	**Dialogic Timelines**
Dialogue\Characters	**Dialogic Characters**

2.4.2 General Naming Rules

All **scripts, scenes, and root nodes** should be written with snake case. ex. "example_name"

The following rules will always be followed whenever a new asset is created with no exceptions unless strictly specified with proper documentation and reason behind it:

Asset Type	**Prefix/Suffix**	**Example**	**Comment**
Fonts	**Font_**	Font_Roboto	
Shaders	**SH_**	SH_	
Sprites	**SP_ _[Number][Modification]**	**SP_**Mica**_01a** **SP_**Tree**_01a** **SP_**Tree**_01b** **SP_**Tree**_02a**	**Number** stands for a **whole different asset.** **Modification** is used for an asset that has been **altered.** Example: Tree_01a and Tree_01b are the same tree with b having different colors and/or sizing Due to this being a 2D game Sprites must be as **descriptive as possible** for ease of use.
Animation	**A_**	**A_**Enemy**_Shoot** **A_**Mica**_Glide**	
Placeholders / Rough Assets	**R_**	R_SP_Mica_01a	R_ goes in front of filename
Sound Effects	**SFX_(usage)_[Number][Modification]**	**SFX_Mica**_Jump**_01a**	**Number** stands for a **whole different asset.** **Modification** is used for an asset that has

			been **altered**. First word is where the SFX is used. Example: Jump_01a and Jump_02b are different that 02 is a completely different asset on its second iteration.
Music	**Music_**	Music_Village_Victory	

2.5 Feature List

- Player Features
    - Player Stats
        - Health
        - Stamina?
    - Player Movement
        - Walk
        - Jump
        - Glide
    - Sword Movement
        - Pick Up
        - Sword Swing
- Narrative Features
    - Dialogue
        - choices
        - Interactable NPCs
    - In-Game Cutscenes
- Enemy Features
    - Patrolling Enemies

    - Shooting Enemies
    - Flying Enemies
    - Aggravatable Enemies
- Level Features
    - Parallaxing Background
    - Buttons / Switches
    - Door Triggers
    - Moving Platforms
    - Collectables
- UI / Menus
    - Input Remapping
    - Volume Settings
    - Speedrun Timer
    - Save Slots
    - Auto-Saves
    - Loading Screens
- Juice
    - Jump Effect
    - Glide Effect
    - Sword Hit Effect
    - Damage Effect
    - Pick Up Effect
    - Multiple Dialogue Effects
    - UI Click Effect

2.6 Software Architecture

2.6.1 Coding Conventions

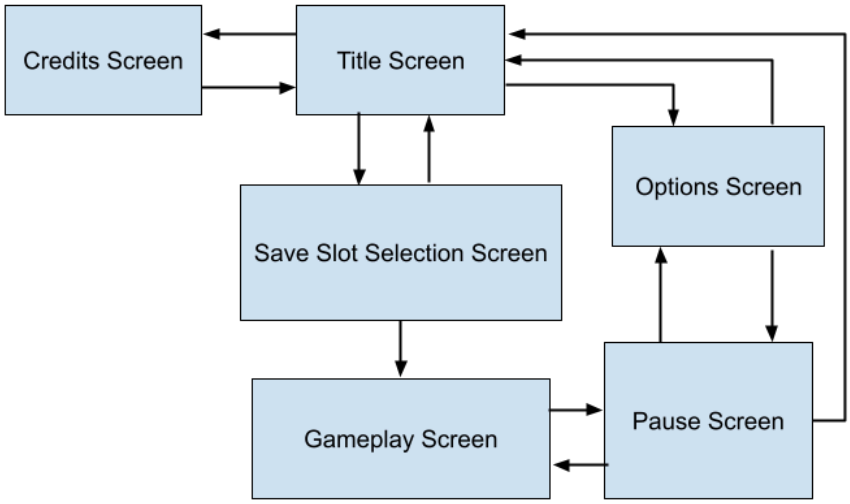
Our switch to Godot will require our scripts to be written in GDScript, meaning we will also need to adapt to Python's standard naming conventions since Godot's integrated methods use them. This means we will be writing functions and variables in **Snake Case.** ex: func function_name var variable_name

As well as the naming convention, all code is expected to be well commented. This will allow anyone on our team to more quickly understand what is going on in a script someone else had worked on prior.

2.7 UI/UX

- **Title Screen**
    - New Game / Continue
        - **Save Slot Select**
    - **Options**
    - **Credits**
    - Quit
- **Options Screen**
    - Gameplay
        - Speedrun Timer Enabled
        - Accessibility Cheats
    - Audio
        - Master Volume
        - Music Volume
        - Sound Volume
        - Dialogue Volume?
    - Video
        - Full Screen
        - Vsync
    - Controls
        - Key Mapping Menu
- **Pause Screen**
    - Resume
    - Load from last Checkpoint (Restart Puzzle)
    - **Options**
    - **Return to Title Screen**
- **Gameplay UI / Gameplay Screen**
    - Health
    - Dialogue

2.7.1 FlowCharts / State Diagram



2.8 Mechanics

2.8.1 Sword

The sword mechanic needed to undergo a massive overhaul to stop the player from clipping into walls. This meant changing the sword from a positional system to a physics based one.

2.8.1.1 Introducing Physics

The sword has been switched to a Godot RigidBody2D node attached to the player through a PinJoint2D node. This has switched the way the sword works in a couple of ways.

1. The sword was originally an extension of the player's body, and it is now its own body

2. The sword has gravity, mass, and friction for us to consider
3. We now have full control over the sword's pivot point.

Points 1 and 2 are the most notable of these changes. The first means that the sword is no longer an extension of Mica's collision shape. The second means that we can control the sword with physics impulses. These two elements together will entirely stop Mica from clipping into the wall.

2.8.1.2 Changing Mica to a RigidBody2D

Godot, unfortunately, doesn't allow for a CharacterBody2D's movement to be influenced by a RigidBody2D. This meant 1 of 2 things.

1. We need to brute force this interaction through code
2. We need to convert Mica to a RB2D

While forcing these two nodes to interact was entirely possible, it was a gateway for a slew of other bugs. Instead, the decision was made to convert Mica to a RigidBody2D. The main things we lost from this conversion were the is_on_ground() and move_and_slide() methods. Losing move_and_slide() wasn't a huge loss, as we already processed horizontal movement on our own, and RB2Ds process collision inherently. The bigger issue was losing is_on_ground(), forcing us to use RayCasts to detect when our player was on the ground.

2.8.1.3 Movement Processing

Though the old movement processing system we had still worked, directly accessing an RB2D's linear_velocity can cause unexpected issues elsewhere. To counteract this, Mica's horizontal movement processing was changed to use physics impulses on the x-axis. This also inadvertently added a natural feeling of acceleration to movement.

2.8.1.4 Two Button Control

A two-button sword control scheme was the first to be added due to ease of implementation with the new physics system. With this control scheme, torque impulses were applied in the counterclockwise or clockwise direction, depending on the player's input. This system was ultimately scrapped.

See 3.1.1 Why not use buttons for sword control? for more details

2.8.1.5 Radial Controls

Ultimately, the control scheme we wanted to implement is one that follows the player's mouse or right joystick, depending on their control scheme. This needed to be broken up into a few parts to work seamlessly.

1. Find where the player is aiming. This is done on controller by getting the Vector2 of the player's right joystick input. This step can be skipped with keyboard + mouse

2. Get the angle between the inputs and the player. This is done using the .angle() on the Vector2 from the controller input. For keyboard + mouse, using: position.angle_to_point(get_global_mouse_position()) combines step 1 and 2.

3. Find how far the sword needs to move by subtracting the angle just found from the sword's rotation value. This will give a value in radians.

4. If the distance the sword needs to move is positive or more than half the distance around the negative direction of the circle, the sword will rotate counterclockwise. If the distance is negative or more than half the distance around the positive direction of the circle, the sword will rotate clockwise. If the sword is within 4 hundredths of a radian of the desired position, its rotation will be locked and physics processes will be put to sleep.

As a footnote, if the sword is colliding, even if it’s within the desired position, it cannot be put to sleep as it will cause the sword to slowly clip into the ground.

2.8.1.6 Sword Gravity

It was decided that gravity having an impact on the sword was detrimental to the experience of using it. Gravity caused the sword to become unwieldy and bounce around while the player walked.
Due to this, the sword was given a mass of 0.01 while not colliding. This adjustment also required the force the sword moved with to be lowered (to a value of 50).

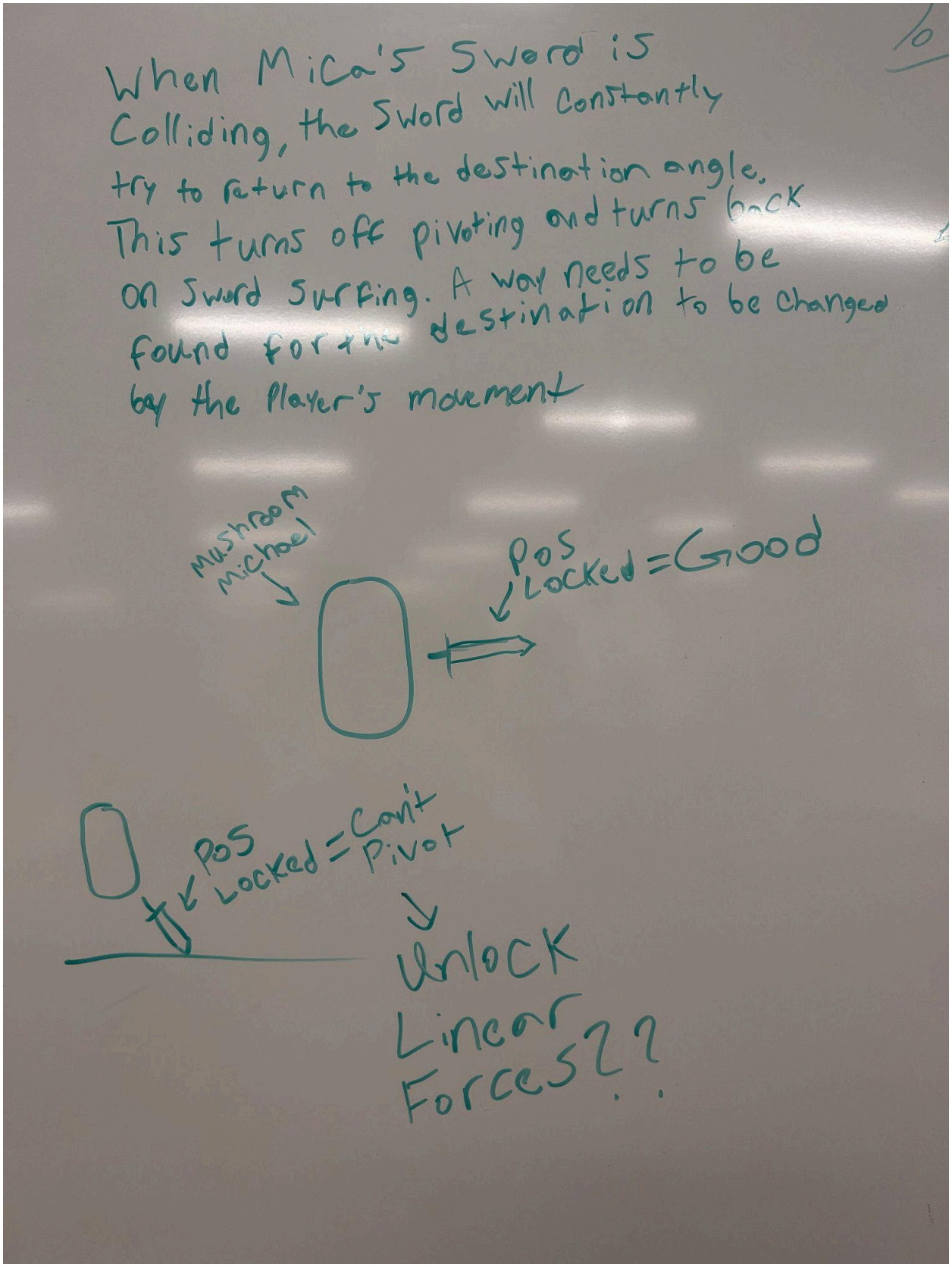
When the sword is colliding, it returns to its original mass and speed. These values are adjusted based on the length of the sword.

2.8.1.7 Pivoting

When we used a two-button control scheme, pivoting came naturally. This was not a luxury we had with radial controls, as the sword would always try to follow the player’s mouse or joystick.

To counteract this issue, a check was implemented for if the player was trying to move while balancing on the sword. If they were, the player’s inputted angle would be locked to the angle of the sword and the sword’s rotation would be unlocked. This causes the sword’s rotation to move with an opposite reaction to the player’s movement.

This was ultimately scrapped due to platforming with it not feeling tight enough.


When Mica's Sword is Colliding, the Sword will constantly try to return to the destination angle. This turns off pivoting and turns back on Sword Surfing. A way needs to be found for the destination to be changed by the Player's movement
Mushroom Michael
Pos Locked = Good
Pos Locked = Can't Pivot
Unlock Linear Forces??

2.8.1.8 Current Sword Values

- Length 15
    - Mass 15
    - Sword Speed 60000
- Length 50
    - Mass 50
    - Sword Speed 140000

2.9 Development Plan

Check Gantt Chart for full Development Plan

Sword of the Morel Gantt Chart

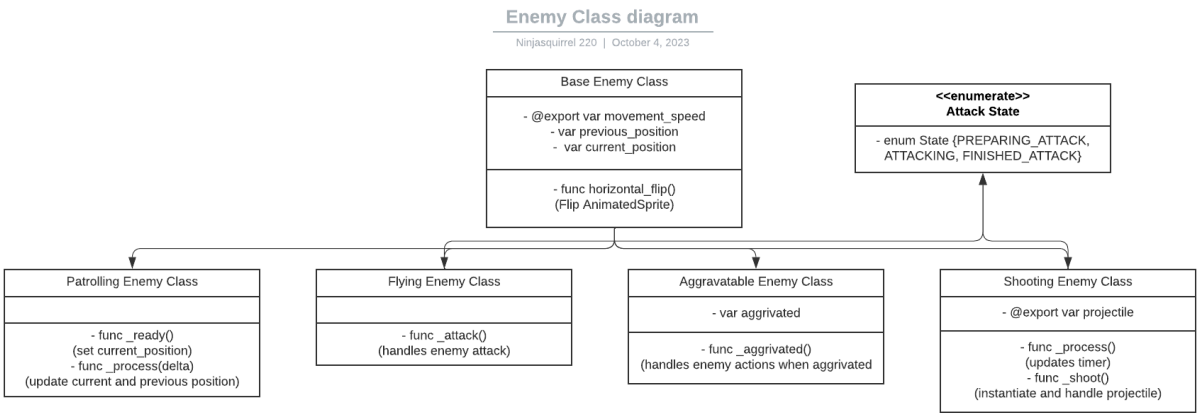
The overall plan for the development of the game is for full completion around May of 2024. The stages of development are broken up into these stages:

- Pre-Production | **9/17/2023**
    - Planning and Organization
- Game Jam Basics | **10/15/2023**
    - Recreation of everything developed in the game jam version of Mushroom Kid's Big Grass Sword
- Mushroom Village | **11/5/2023**
    - Includes the main tutorial of the game as well as the set up for the overall narrative and player motivation
- Temple | **11/19/2023**
    - Tutorial area for the sword mechanic.
- Forest | **12/3/2023**
    - Puzzles that ramp up in difficulty until the player reaches the next area.
- Showcaseable Demo & Steam Page | **12/22/2023**
    - Demo version that is polished enough to start development of trailers, clips, and other materials meant to hype up the game's full release.
    - Steam Page up and ready for wishlists.
- Release | **5/1/2024**

- Game Released Online

Of course this is all subject to change but as for right now this is the general development schedule planned out.

2.10 Enemy Class Diagram



3. Gameplay Design

3.1 Controls

The controls for Mushroom Kid’s Big Grass Sword should aim to be as easy to understand as possible. This will be somewhat challenging for a gamepad layout and less challenging for keyboard and mouse.

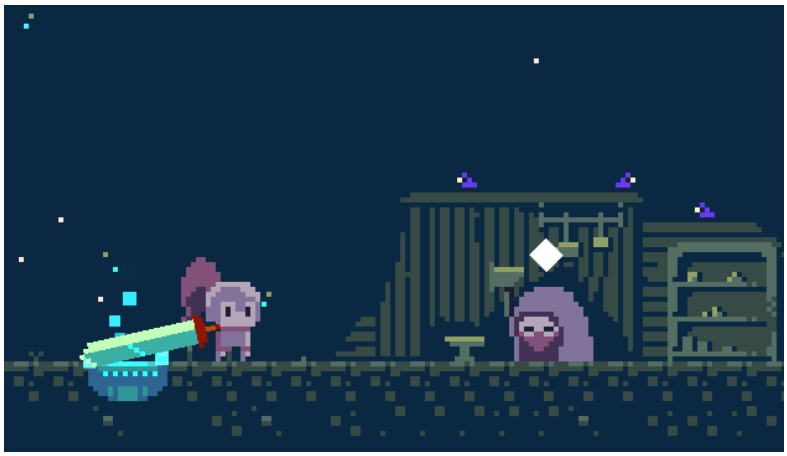
Since this is the case, all control mapping tests and most gameplay/build tests should be done on a controller, since it’s easier to translate workable gamepad controls to keyboard opposed to vice versa.

Basically test Gamepads first Keyboard & Mouse second.

3.1.1 Why not use buttons for sword control?

In our original game jam version of Mushroom Kid's Big Grass Sword we had sword controls mapped to mouse movement. This was fun for many people since the freedom of movement allowed them to pull off incredible tricks and feats in game.

Deepest Sword's Movement:



Getting Over It With Bennett Foddy's Movement:

If we were to implement sword movement more akin to the game Deepest Sword, which has the sword's turning mapped to two buttons, we would lose a lot of the interesting aspects the original mechanic had. Our sword's turn speed would have to be a fixed value, independent of the player's actions.

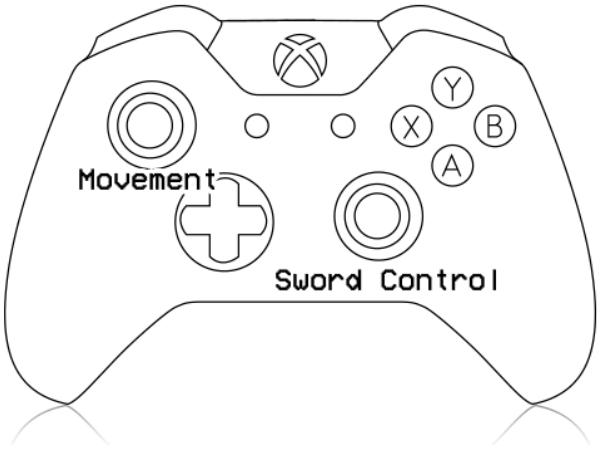
Instead, we're aiming for a sword movement system more like the hammer in Getting Over It With Bennett Foddy. We believe this approach will make the gameplay more engaging and allow for designing levels that take full advantage of this mechanic.

3.1.2 Console Controller Mapping

Mushroom Kid's Big Grass Sword's controller layout may end up a bit more complicated as we try to transfer the controls from the original keyboard and mouse mapping to controller.

Seeing as the Sword Mechanic is one of the highlights of the game, the controller input needs to be mapped in such a way that the player is allowed to move the sword freely, as well as moving the player character however they want.

This means that the two thumbsticks will always need to be in use for the player.

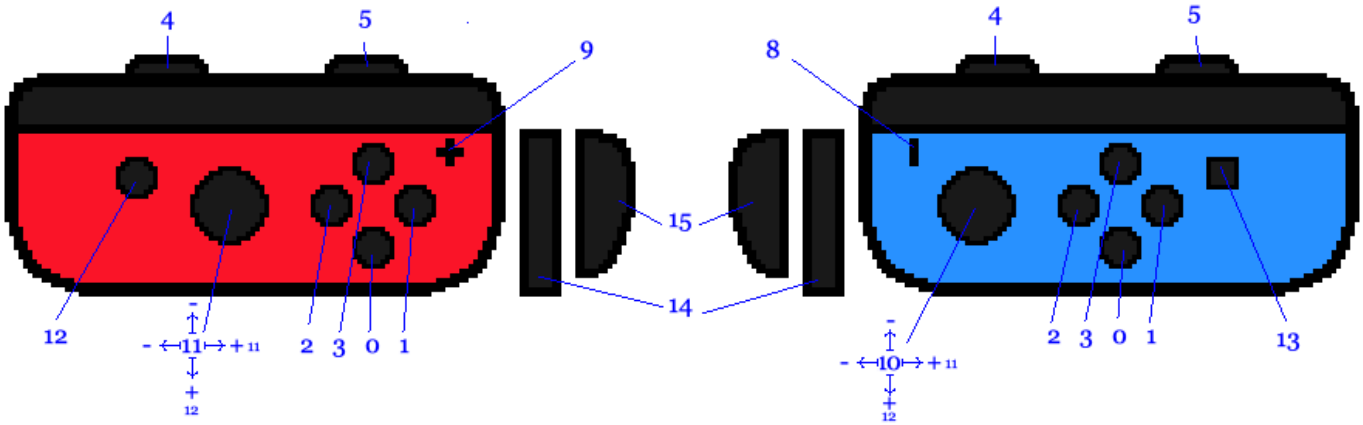


Luckily for us, the only other buttons that would need to be mapped afterwards are:

- Jump Button
- Interact Button
- Pause Button

To mediate this problem we will need to do some polling on what people prefer as their default controller mapping. Hopefully we can reach enough consensus to have that be the standard. If anyone wants to change it, we can provide input mapping in the options menu.

Also, due to the small amount of buttons left to map, it may be alright to map multiple buttons to the same action as to not overwhelm the player with button layouts.



For single joycons, the issue of less buttons is compounded. We will have even less to work with, however we still have just enough buttons to make the controls workable. Since precise sword movement is critical to our gameplay experience, we will have it mapped to the joycon's thumbstick with the rest of the movement set to buttons.

Key Mapping:

- Movement
  - Gamepad
    - Left Thumb Stick
    - D-Pad
  - Single Joycon
    - Joycon Left & Right Buttons
- Sword Swing
  - Gamepad
    - Right Thumb Stick
  - Single Joycon
    - Thumb Stick
- Jump
  - Gamepad
    - Bottom Button / A
    - Right Trigger
  - Single Joycon
    - Right Trigger
    - Bottom Button and Top Button
- Glide

        - Hold Jump Button
- Interact
    - Gamepad
        - Left Button / X
        - Left Trigger
    - Single Joycon
        - Left Trigger
- Pause
    - Gamepad
        - Start Button
    - Single Joycon
        - Minus / Plus

3.1.3 Keyboard & Mouse Mapping

Since Mushroom Kid's Big Grass Sword was originally prototyped in a game jam and showcased with Keyboard and Mouse controls, most of the key mapping from the prototype can stay the same since it did not deter play progression when testing.

Key Mapping:

- Movement
    - Left
        - A
    - Right
        - D
- Sword Swing
    - Mouse
- Jump
    - W
    - Space
- Glide
    - Hold Jump Button
- Interact
    - E
- Pause
    - P
    - ESC

3.2 Core Mechanics

3.2.1 Sword Mechanic

The Sword mechanic in Mushroom Kid's Big Grass Sword is one of the main selling points of the game. The player is given a sword that grows throughout their playthrough that they are allowed to freely swing to climb obstacles and defeat enemies.

What makes this mechanic engaging is the freedom of movement and unique platforming challenges it creates. For example, when the sword gets quite long it starts to become challenging for the player to maneuver through low ceiling tunnels, since they have to position the sword in such a way so that they do not get stuck on the ceiling or ground.

The growth of the blade also allows the player to adjust to its size over time. Now that our game is being fully produced, hopefully we can have levels that try to use the sword's current length to its benefit and introduce each length slowly.

We also believe that since movement is not fully halted by the sword this allows for less-skilled players to learn the mechanic at their own pace, rather than blocking off content if they do not understand it straight away.

3.2.1.1 Future Plans

In the future, we would like more mechanics that play off the idea that the player is swinging around a big "sword". For

example, walls that the player can stick to, but only if the tip of their sword is plunged into it.

This would need to be tested and examined. In the future, results of this should be documented in a new section under this mechanic.

3.2.2 Movement & Jumping

Movement and Jumping should be constructed in a way that is as simple for the player to understand as possible, since the player should be trying to focus more on understanding the Gliding and Sword Mechanics.

However, there are still a few details to be made clear.

Movement shouldn't feel like the player is running. Since levels will be focused more on the player trying to figure out the solution to a platforming puzzle, that means we should try to avoid the player accidentally messing up an area because the player character was too fast for them to react. However, movement should also not feel sluggish to a point that it is a chore to move through an area. This can most likely be fixed with better level design rather than movement design.

Jumping may need a little bit of experimentation to get exactly right. For now here are some notes:

- The player should be able to jump around twice the height of the player character.
- Coyote Time is required so as to not frustrate players who think they should have made a jump. Even hard games like Super Meat Boy have a small amount of Coyote Time Frames.
- In the game jam version, the jump was affected by how large the sword was. I think this may not be the right decision now and should be avoided since players would have to relearn the jump every time the sword grows. If this is not made clear to the player, this will be frustrating.
- There shouldn't be any downward acceleration applied to the jump as it may mess up the sword's physics.

Check Stat Fine-tuning document for specific numbers.

3.2.3 Gliding

By using the bottom part of the mushroom cap, the player is able to slowly descend to the ground by floating.

This mechanic gives the player a bit more time to help with their jumps and precisely maneuver their sword up ledges and cliffs.

In the game jam version of the project, gliding was very forgiving and you could do it as long as you were above the ground. For the full release of the game, I would like for us to experiment with a new system with a little less forgiving glide.

An infinite glide slows down player progression and can sometimes confuse the player if they glide over massive areas, since they can't see the ground.

3.2.3.1 Glide Decay

In place of a traditional stamina system, we've decided to use a decay system for our player's gliding. To implement this, we have our player's glide use a separate gravity scale, as opposed to a flat rate of speed while falling. The player's downward velocity is then stored to a variable, so the "effectiveness" of their glide is remembered.

On top of this new gravity scale, a penalty multiplier has been added so each repeated time the player glides, the penalty is applied to their downward velocity. This prevents the player from getting extended glide distance by mashing the button.

When the player lands, their downward glide velocity and penalty multiplier variables are reset.

3.2.4 Camera System

The camera system for the game needs to be very dynamic so that our game is always framed to give the player the amount of information they need to solve a platforming challenge.

Due to the unique methods of traversal in our game a lot of thought is needed to make sure the camera is set up right during gameplay.

3.2.4.1 Camera Spacing and Positioning

For starters, since our player will be moving a lot left and right with a giant sword that will block some parts ahead of the screen it will be important to provide some horizontal look ahead that will be dependent on if the player is moving left or right. Similar to games like Cave Story.
For Reference:
https://youtube.com/clip/UgkxFn1uvgS3Bvcoz-CefxPkXEzn7DhFhqeL?feature=shared

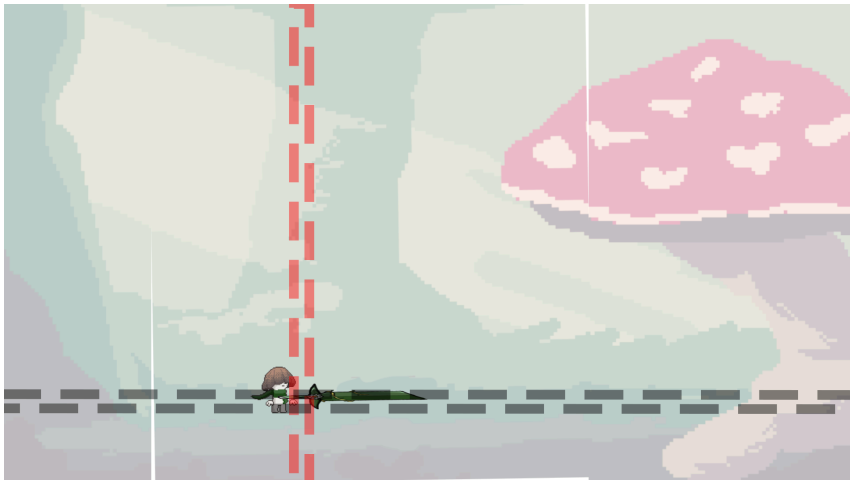
This ease between left lookahead and right lookahead should probably only occur once the player has made a considerable amount of movement towards the given direction. Otherwise the effect may be nauseating for some players.

We also have to make sure that these lookahead positions are not too far away from each other so that players who use a mouse don't start missing platforming challenges because the camera shifting messed up their mouse position. This may be fixable with code but will need to be researched first.

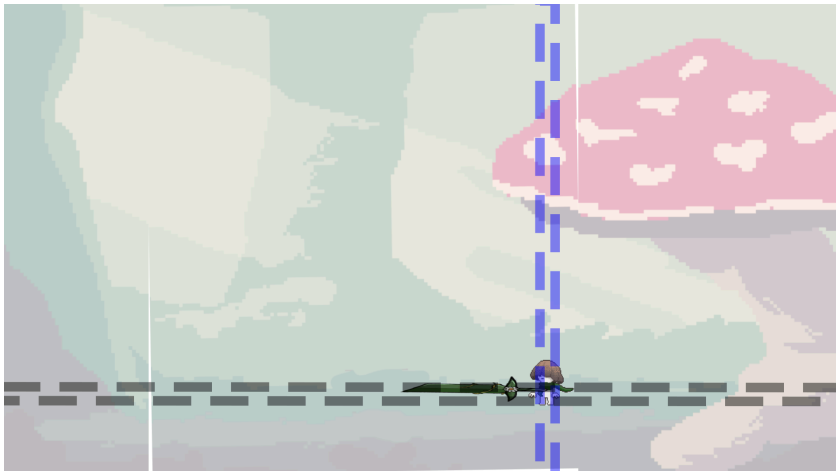
These lookahead positions should also only be modified when the player is grounded and waking. NOT when they are falling or swinging a sword towards the direction.

As for the vertical spacing, since our player will be moving it most of the time and since downard traversal is less dangerous because of the gliding mechanic, the player should be given some upward vertical lookahead in order to see what platform to climb up next with the sword.

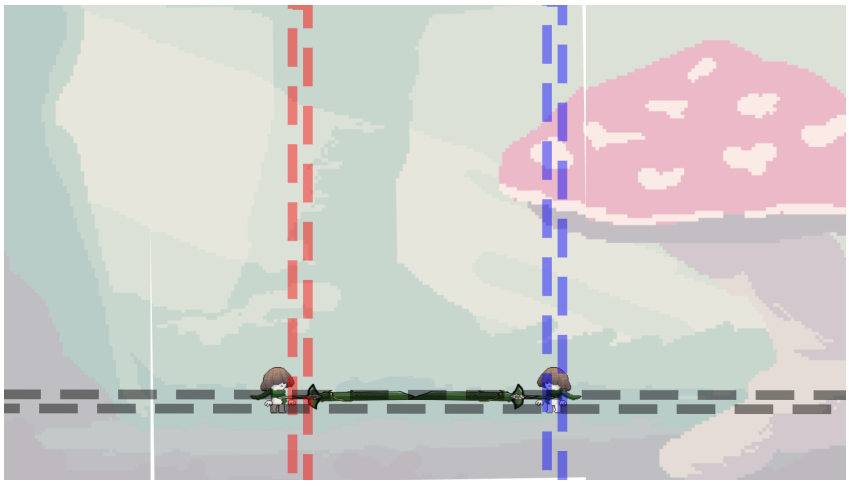
Moving Right Camera Spacing Diagram



Moving Left Camera Spacing Diagram



Side By Side Comparison



***These spacings and positions are prone to change once we get playtesting feedback, this is just an example for now**

As for jumping and other upward movement we will most likely have to experiment a bit to find what works best for our game. It might be best to just have the player always follow the player's vertical position so that it is easier to align jumps and sword swings.

Slight Camera Damping should be provided both horizontally and vertically.

3.2.4.2 Camera Zoom

We may also want to consider zooming out the camera depending on the length of the player's sword so that the zoom isn't so far out at the start of the game.

For certain areas (Specifically collectible challenge areas), It may be best to have the entire camera frame the platforming puzzle so that the player has all the information they need to clear the exceptionally challenging area. (**Similar to VVVVVV or Celeste**)

There may also be times where the camera should be zoomed out for thematic and narrative purposes like during the ending of the game where the platforming is more simplified so that the ending packs a better narrative punch and looks cooler visually.

3.2.4.3 Camera Effects

Camera Shake should be used and applied in a variety of ways to add impact and juice to the game with the two biggest uses needing to be implemented is in **Narrative Events** and **Player Events**.

During certain **Narrative Events** the camera should shake and change in order to give proper impact to the player on what just happened and keep them engaged with the story.

Certain **Player Events** should also shake the camera with the game **freezing and shaking when the player takes damage** and the **camera slightly freezing when an enemy is defeated.**

3.2.5 Dialogue System

One of the big draws of the game jam version of Mushroom Kid's Big Grass Sword was the funny dialogue throughout the game.

This was only possible through integration with Yarn Spinner. Now that the game is working towards a full release here are some design notes on dialogue system changes and implementations needed to make the dialogue and narrative more engaging and easier to work with.

- Dialogue should be progressed with the interaction button instead of a UI continue button so it is easier for players on controllers.
- There should only ever be a **max of 3 dialogue options** so the entire text box isn't filled up.
- Characters should appear on screen with the gameplay background faded so it is easier for the player to see the dialogue and for future effects to be played within the dialogue.
    - 

(Example from Turnip Boy Commits Tax Evasion)
- Text should be more animated and effects should be added to add more punch to the text on screen.
- Characters should have voices that work similar to Animal Crossing's dialogue where each character generated on screen plays a note.

3.2.6 Health & Stamina Systems

The player character's **health** should not be a system the player is constantly fighting against or managing. Instead it should mainly give time for players to situate themselves after making a mistake.

I believe that the best way to do this is to have the player character regenerate their health after not taking damage for a short amount of time. Thus, allowing for players to not worry about health management while also adding challenge to levels with collectibles and speedrunners.

Stamina should just regenerate overtime and should not be shown to the player unless the stamina is less than 100%.

3.3 Goals & Progression

Throughout the game the player's mechanical goals are straightforward.

- Explore new area.
- Solve the puzzle.
- Grow sword if possible.
- Enter new Area.

In the game jam version, we had sword growth tied to enemy kills. This no longer seems viable nor makes that much sense narratively. Instead sword growth should be tied to key locations set by the level design of the game. That way we have more control over the exact abilities of the player and we can fit in the narrative element of sunlight being the cause of a sword's growth.

As for progression, I believe it would be best if the game had checkpoints the player could save at periodically. These checkpoints would mark the start of a puzzle allowing us to also add a ***reset to last checkpoint*** feature in case the player gets stuck in a hard puzzle and wants to redo it from the start. **This will also be where the player spawns if they die.**

However, to mediate any saving issues for the player we should also have a **save slot dedicated to auto-saves.** Auto-Saves would update as soon as a player reaches a checkpoint without any interaction needed so that if the player forgot to save for a while progress isn't completely lost.

3.4 Enemy Design

Due to the different approach we are taking for the game's full release, enemies should be easy and quick to understand so as to avoid frustration in the player.

In the game jam version of the game, enemies would die after a couple of hits from the player's sword. This was honestly quite clunky. Both because of how hastily implemented the enemies were and due to the fact that there had to be a cooldown before you could hit an enemy again.

That is why I propose that enemies should just die in one hit. Allowing for players to focus on the level and less about trying to defeat the enemies in front of them. It also makes more sense for a big legendary sword to kill its enemies quickly.

Since they are killed instantly they also shouldn't have any collision with the sword since this would cause some janky physics.

Below will be explanations for **different attributes that can be applied** to enemies. These attributes should be able to be combined with the developer given the ability to enable and disable attributes they may or may not want to create a vast variety of unique enemy types that should be implemented and tested.

3.4.1 Patrolling Enemies

Patrolling Enemies are very simple. They go from one location to another and act as a moving obstacle for the player.

3.4.2 Shooting Enemies

Shooting Enemies shoot out in a straight line from a set location. These projectiles hurt the player but may not necessarily bypass the sword.

Projectiles should always be shot out in a straight line or predictable pattern so the player is able to plan their moveset.

3.4.3 Flying Enemies

Flying enemies should float up in the sky and dash periodically through the sky. Similar to enemies like the propellor rat from **Shovel Knight**.

This makes it easier for the player to dodge hits since they are given a moment to react to the enemy's current location.

They may also be combined with enemies that **patrol** a location or get **aggravated**.

3.4.4 Aggravatable Enemies

Aggravatable Enemies are just enemies that are capable of targeting the player.

3.5 Level Design

Levels should try to always be teaching the player something new about their moveset without trying to hold their hand throughout the entire process. Tutorials should be pretty much invisible with NPCs dotted around to give hints if the player needs it.

At the start of the game a lot of emphasis should be placed on having the **player use and master their sword** in every way possible to progress. This way they can build up a familiarity with the mechanic so that the addition of a new mechanic or obstacle is not overwhelming.

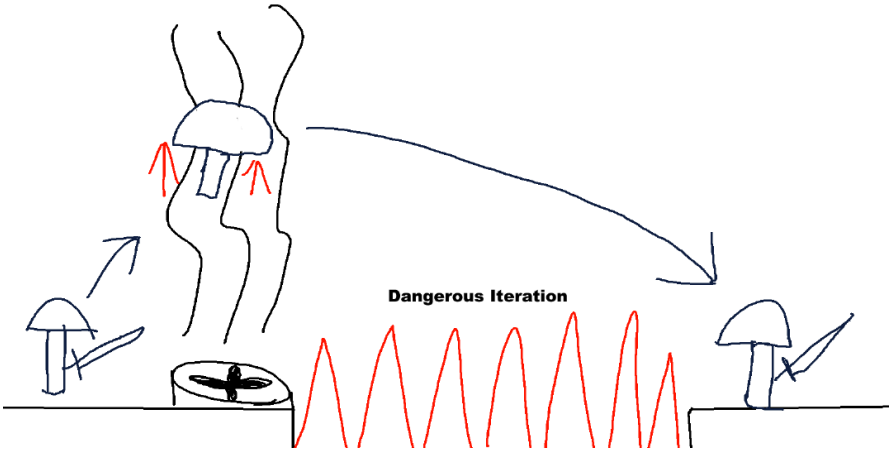
Levels should also be constructed so that a "tile" in-game is only equal to **around a fourth of the player character's height**. Thus, allowing for more complex levels to be created that can make proper use of the sword precision.

Levels introducing a new concept or mechanic should follow roughly this format:

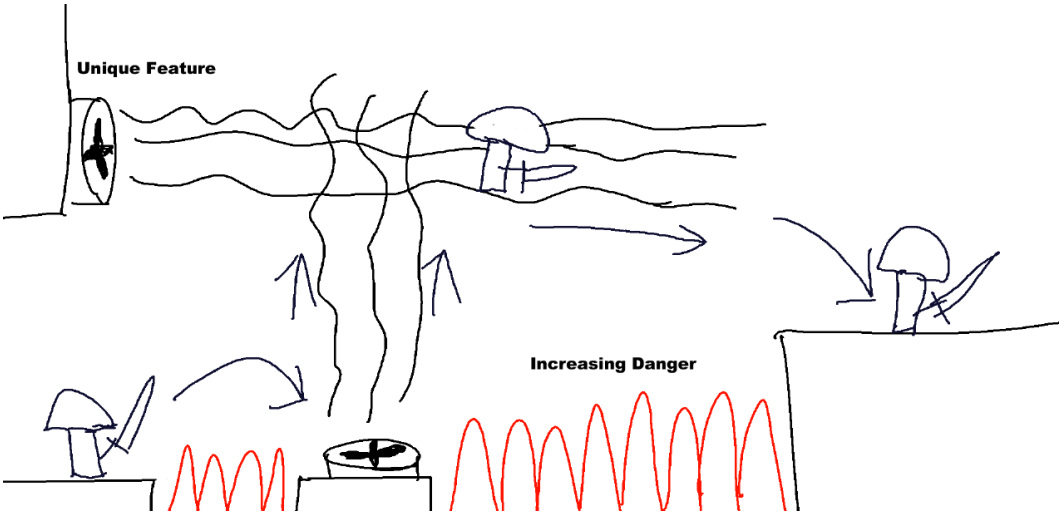
1. New mechanic is introduced in a safe environment allowing for the player to experiment with the idea for as long as they want.

a. 

2. Next, iteration on the mechanic should introduce a more dangerous element or enemy to the same mechanic mostly unchanged.

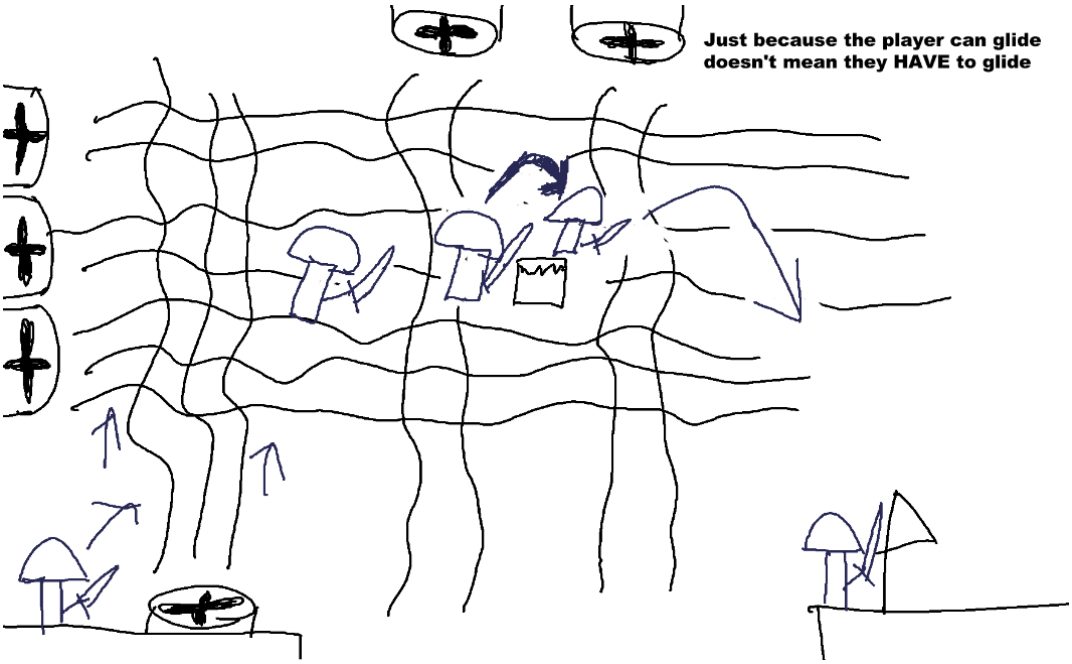
a. 

3. A unique feature of the mechanic or whole new secondary mechanic should be introduced with a bit more danger.

a. 

4. Lastly, an area should be added at the end to test if the player has truly understood the concept.

a. This can be done best by playing off an assumption a player would make about the mechanic already and challenging them to think more outside the box.

b. 

Additionally a hidden part of the level should be added for completionists to play to showcase **mastery**. With a collectible as a reward.
This could also be used to showcase a **cool level mechanic that didn't quite make it** into the main story or didn't have enough time to iterate upon.

3.5.1 Should levels have ramps or slopes?

Adding ramps and slopes to our levels could add a lot of uniqueness to our game's level design. However, there may be a lot of technical and design challenges that await us if we decide to implement them without a proper plan.

3.5.1.1 Referencing other Platformers

Games with Ramps and Slopes	Games without Ramps and Slopes
Mario Games	Celeste
Ori and the Blind Forest	Thomas Was Alone
Rayman Legends	Fez

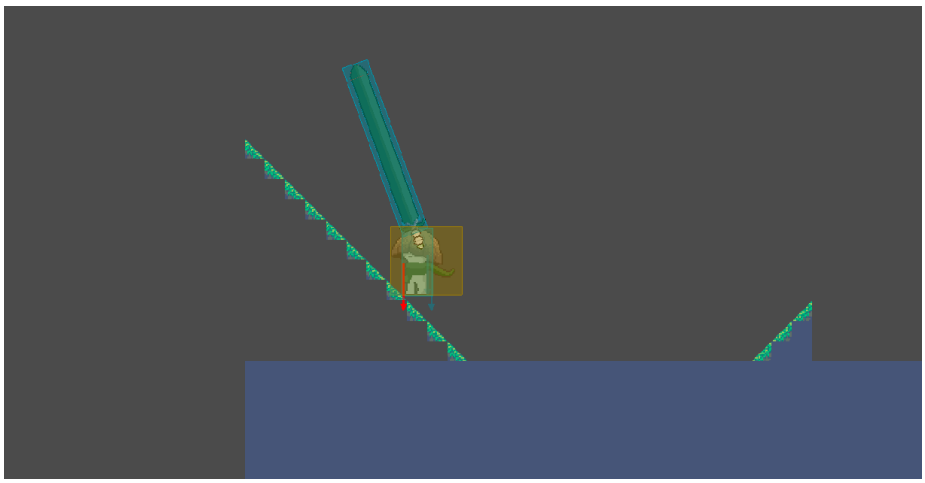
Cave Story	Shovel Knight
Sonic	Spelunky
Getting Over It	Deepest Sword
Bread and Fred	Mega Man
Braid	Hollow Knight
Super Meat Boy (**Seldom**)	
Pizza Tower	
Donkey Kong Country	
Guacamelee	

The choice of a game having or not having ramps seems to depend largely on the type of gameplay experience the game is trying to convey.
Most of the games that chose to not have ramps or slopes are very chunk based. Having gameplay confined into certain spaces.

3.5.1.2 Pros & Cons

Pros	Cons
More Interesting Scenery	More Artwork needed
May allow for more interesting level mechanics	Will cause more bugs
Can streamline movement across areas of a level without making it feel monotonous	More areas of the game will need to be tested in unique ways.

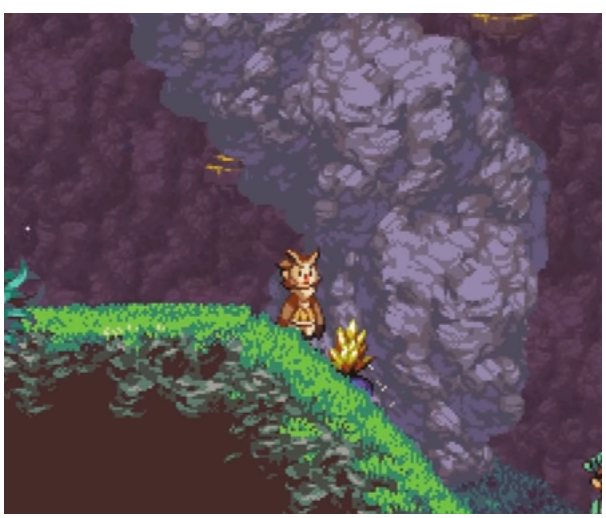
3.5.1.3 Results from Prototyping



Ramps work pretty well in the game on their own. We just need to make sure it's implementation hides the players feet so it doesn't look like it's hanging off the side.

3.5.1.4 Final Decision

Ramps and Slopes will be included in our game but they should be used for platforming challenges. Their usage should serve to enhance the visual experience like in games like Braid and Ori and the Blind Forest.

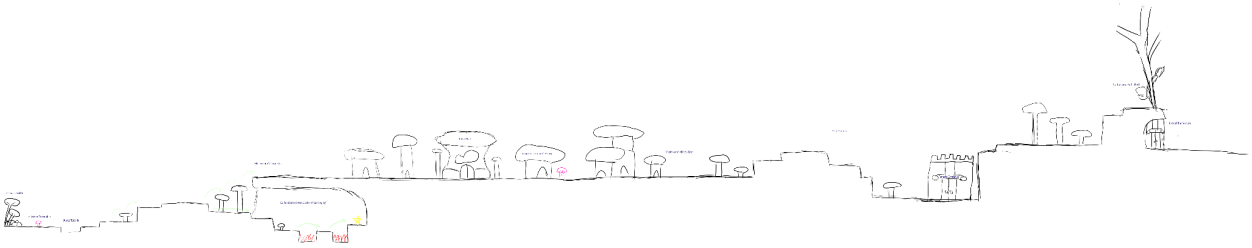


Ramps and slopes should be implemented so it doesn't look like the player is hanging off the side. This can be done by making part of the tile rendered in front of the player.

We may end up with a future area that may be able to utilize ramps in a unique way but until now that should not be considered.

3.5.2 Mushroom Village

Overview of the entire level sketched out:



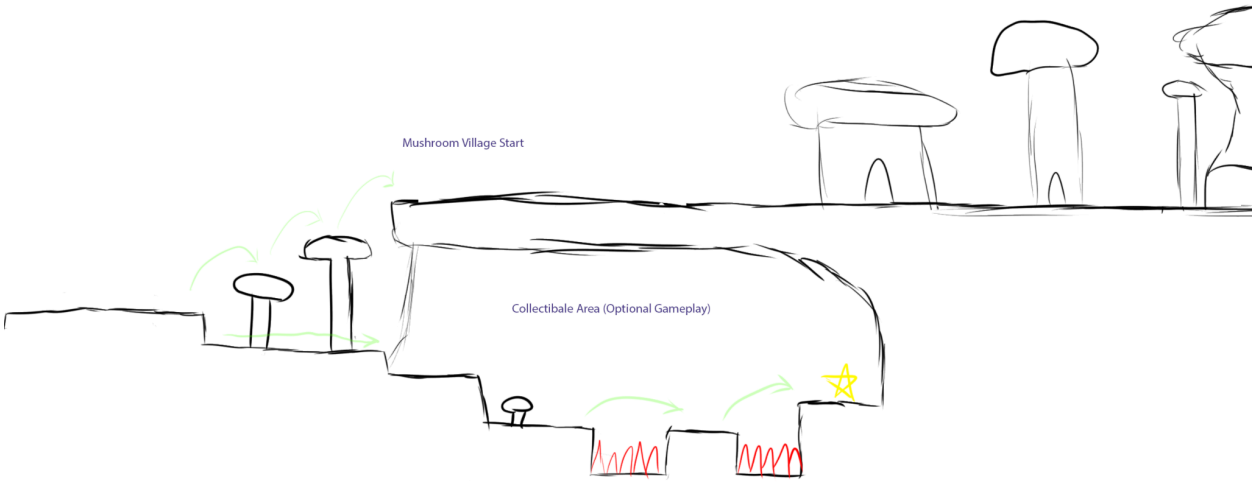
First section that includes the Movement and Jumping Tutorials:

At the start of the level the player character is shown to be playing alone in the forest. This is quickly interrupted by Grandmashroom calling their name in the distance.

This small scene will act as an invisible movement tutorial for the player. Since they will be facing right and there will be obvious environmental elements blocking their left path the player should know that they need to move to the right.

As they start moving, the player will soon find themselves in a hole that they can not get out of until they jump. This will quickly teach the player jumping as the jump from terrain element to terrain element with the next challenge needing the player to jump onto a smaller platform to make it on top of a cliff.

Second section that includes that showcases the entrance to the village and teaches players about optional collectible areas:



Once the player has passed that challenge, they will be greeted by a fork in their path with a final jumping challenge that leads to the mushroom village and a cave that leads down to a challenge area.

This area is supposed to act as a final test on jumping for the player as well as teach and reward players who truly understand the mechanic by adding a challenging, deadly area right before they reach the village. This will set a precedent for levels in the future so that they are not shocked by future challenge areas.

If the player fails the challenge they will simply respawn to the start of the level which won't be frustrating since they are not too far away from where they left off. Teaching them how death works mechanically.

Third section that showcases the Mushroom Village:

This area will mostly be set up by the environmental artists so that the area looks visually pleasing since there aren't any mechanical challenges here. However I would like there to be a save point in this area after the town hall (probably near Grandmashroom's house) that will teach players saving and help ease frustration. Especially since the player will be returning to this area after they help Grandmashroom get some cotton for her ears because the town hall is so loud.

Fourth section that teaches Gliding and foreshadows the Temple Level:

This last area acts as both a gliding tutorial and foreshadowing for the events to come narratively. As they complete the gliding challenges they will pass by an ancient temple that is currently locked. As they reach the cotton plant they need to pick for Grandmashroom's ear they plant will act as a natural barrier so the player can't keep moving right. However, in the player's view will be another ancient door that looks similar to the temple they just passed. This is further foreshadowing since this will be where the player exits once they complete the Temple Level.

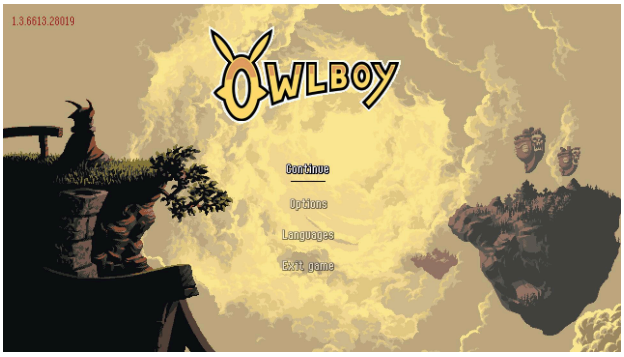
3.5.2.1 Important Notes

Each of these sections that have a tutorial should have **DYNAMIC BUTTON PROMPTS** that display when the player is unable to progress through the area after a set amount of time. Functionality of these prompts will be listed in 3.7 of the Design Document.

3.5.3 Temple Level

3.6 Menu Design

3.6.1 Title Screen



The title screen should feel very visually appealing as the title of the game should be centered on the screen due to its giant size with a sword somewhere as a visual for the big grass sword.

Buttons should also be centered with each button label following the layout of section 2.7.

It should be very clear to the player which button they have highlighted so they do not press the wrong one.

A really cool **stretch goal** would be having the sword grow and shrink depending on what button they have selected. (with quitting the game being the smallest sword length)

3.6.2 Loading Screen

The loading screen in the game should be short and sweet. Just have text that says "Loading…" with the progress bar being the blade of the sword so it's thematic and funny.

3.6.3 Save Slot Selection

Save Slots should be displayed in big bars that display other important information to the player with the buttons to continue or erase the save slot on the slot like how Owlboy does their save slots so it is easier to read.

Make sure to add a "Are you sure you want to erase this save slot" screen when the erase button is clicked or players will get frustrated.

There should be 3-4 save slots for the player to save into with an additional save slot that is ONLY reserved for auto saving with the player being unable to save or erase into this slot. Only continue.

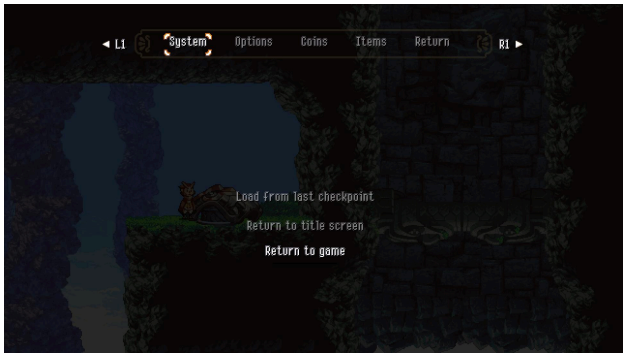
If the player wishes to override a save slot the "Are you sure you want to erase this save slot" screen should be shown for this as well.

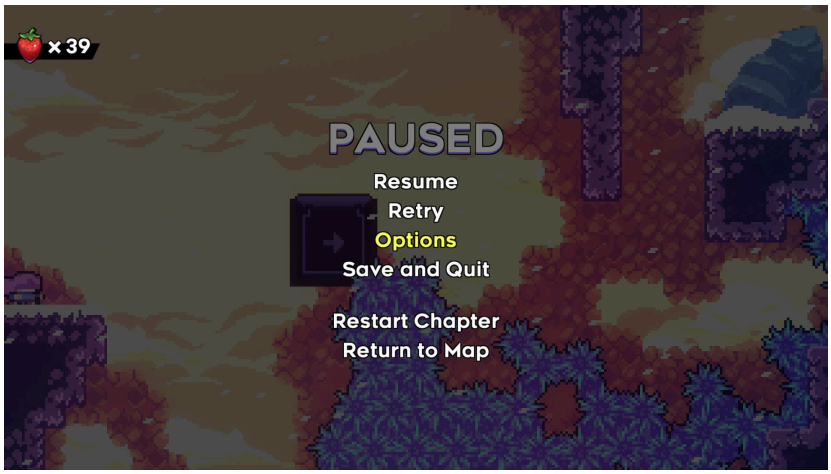
3.6.4 Options Screen

Options should be split up into 4 different sections:

- Gameplay
    - All gameplay settings like if they want:
        - Language Settings
        - How fast dialogue moves
        - Speedrun Timer Toggle
        - Accessibility Options for easier play
- Audio
    - Master Volume Slider
    - Music Volume Slider
    - Sound Effects Volume Slider
    - Dialogue Volume Slider
- Video
    - Full Screen Mode
    - Resolution Setting
    - Enable Vsync
    - Enable Bloom
    - Screen Shake Strength Slider
        - Some people get sick when screen shake is too strong
    - Brightness Slider
- Controls
    - Input Remapping Menu that should have a toggle to switch from Keyboard and Mouse remapping and Controller remapping

3.6.5 Pause Screen





The pause screen should fade out the gameplay a slight bit (probably around the opacity of Celeste's pause screen) It should also clearly communicate to the player that the game is in fact Paused with the word being displayed in the center in text.

Then the player options should be centered under the Pause title following buttons of the layout of section 2.7.

If the player chooses to return to the title screen and they have not saved in a good while then a warning message should be shown before they can fully exit to the title screen.

3.6.6 Credits Screen

3.7 Dynamic Button Prompts

Dynamic Button Prompts will act as a way to inform very new players on the in's and out of the controls without making the game feel like it's holding their hand through every decision.

By having button prompts that only show up after the player has not completed the required action we make sure that none of our players are alienated.

These prompts will probably only show up in the early areas of the game. For example, the first level will have 3 dynamic button prompts. One for **moving**, one for **jumping**, and one for **falling**. Each being displayed if the player has not completed the action in the desired area.

In the future, we may have dynamic prompts that explain level mechanics if the player spends a long time without activating them.

4. Narrative Design

4.1 Theme and Tone

The major theme we want our story to convey is **Growth** and how the events in people's lives give them the opportunity to grow and change for the better. Even if life is hard or the path forward is obscured, it's that roughness that gives us the grit to face the obstacles ahead of us.

4.2 Setting

4.2.1 Summary

The game takes place in a **vast roofed forest** that is home to a small mushroom village that has settled in one of the darker sections of the forest.
Throughout the game, the player will have to traverse **giant rock formations** by going through **tunnels and caves**. This will eventually lead them to the **tops of the trees** in the forest.

4.2.2 Forest

Home to tons of flowers and fauna. The forest should feel mystical and fun to inhabit and jump around.

4.2.3 Mushroom Village

Located in a darker section of the roofed forest. Home to several mushroom villagers.

4.2.4 Temple of the Blade of Grass

Ancient mushroom temple dedicated to the preservation of the **Blade of Grass**, a legendary sword that grows in the presence of sunlight. Murals and torches line the walls of the several connecting chambers of the temple.

4.2.5 Polluted Taiga

A cold and windy forest but other than that pretty standard. Although as the player approaches the entrance to the Heart of the Land, they will find more pollution, trash, and oil spills littering the location.

4.2.6 Heart of the Land

Takes place in the middle of the woods, a frozen car engine sitting in the middle of the Taiga.

Inside it’s parts move slowly and methodically. It’s frozen guts straining it as gears and belts are frozen over.

At the center of the Engine a war room contains a map of the land. And a vital piece of Mica’s journey.

4.2.7 Ant Hills

A Hilly forest that sits above the Ant Empire. The brown ground creates a hilly surface to the environment as blocked tunnel entrances can be found with locals of the empire working.

4.2.8 Ant Empire

Takes place in a sprawling and busy Ant Hill that used to work with the Mushrooms and holds an Embassy in their city.

4.2.9 Whispering Fogwoods

A forest covered in a light fog. Statues and monuments litter the area in imposing and eerie ways the closer the player gets to the Shrine of the Fallen. Maybe even using some for platforming material.

4.2.10 Shrine to the Fallen

The Area takes place in a Haunted Catacombs underneath a Shrine. Hosting the fallen of the Mushroom War.

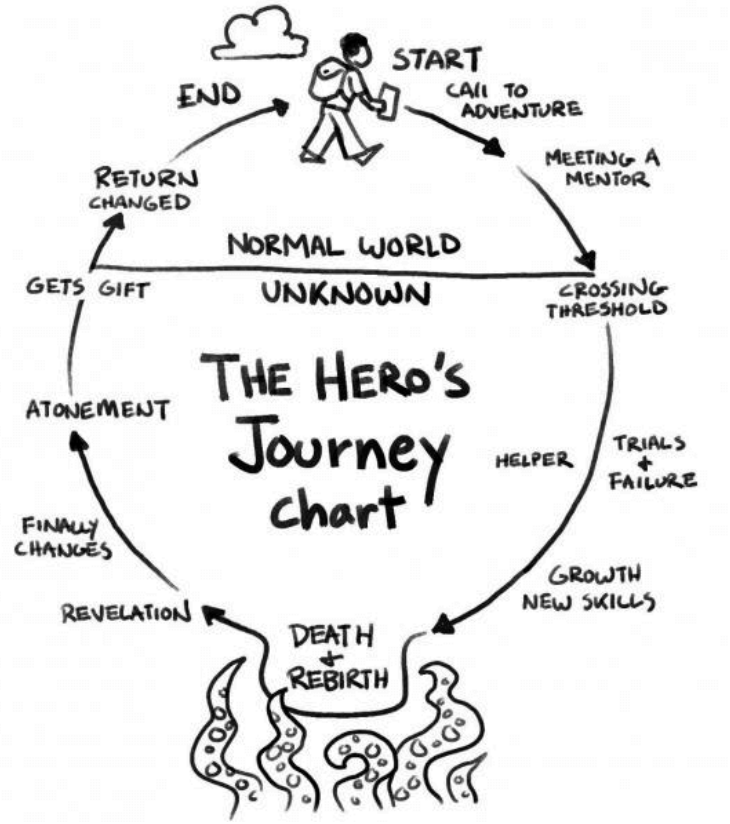
4.2.11 Rippling Forest

Takes place in a destroyed forest. Rotten and lifeless. The area is actually an illusion created by the Shrine to test Mica’s worthiness and bravery.

4.2.12 Towering Treetops

Land with tall imposing trees similar to Redwood Forest and due to the bird’s presence, bit’s and pieces of structures appear in this area destroyed or broken. Sticking out of trees and cutting into it. Seemingly from falling off the bird’s nest or from falling off when the bird was carrying it.

4.3 Plot



4.3.0 Character Growth & Narrative Flow

4.3.0.1 Mica

- Mica will learn throughout the story what it takes to grow and rise up to a challenge.
- Mica will be tested by the land in order to improve their wisdom, strength, and bravery. Using it to defeat the bird.
- Mica will learn to persevere through a hopeless situation.

4.3.0.2 Grandmashroom

- Grandmashroom will slowly learn how to get better at guiding Mica.
- Eventually when met with the Temple of Bravery, Grandmashroom will realize that she must let go of Mica in order for Mica to reach their full potential. Acting as a test of bravery for not only Mica but for herself.

4.3.0.3 Ink

- Ink will learn that a self centered solo adventure is not only a lonely one, but an inefficient one at that.
- He realizes he misses the people of the village and will grow more interested in Mica and Grandmashroom's journey.
- He will NOT end up less weird. It will just be redirected into what he has learned

4.3.0.4 Gill

- Gill will learn that she is more capable than she first believed by going on the journey to find and save Penelope.
- Although she will not end up fighting against the bird, her presence will help Mica in some way on their journey and Gill will end up more confident in the end.

4.3.0.5 Narrative Flow Flow

Each area in the game serves to build up our game's theme without causing us to add a significant amount of content outside the scope of our production timeline. This will be done by heavily reusing the setups for our two areas in the demo: the Forest and the Temple, and styling them to make it feel like a new place.

"Forest Areas" are of course more open spaces, which allow for more freedom of what can or can't be put there. Allowing for character progression, comedic moments, and time for the player to transition into the major location. There will also be a lot of traveling NPCs to set up lore and jokes. The "Temple Areas" act as major locations in the game. Closed areas that test the player's skill and build out the world of the game.

4.3.1 Call to Adventure

At the start of the game, **Mica** is playing alone in the forest until they hear **Grandmashroom** calling them to come home in the distance.

This area will be used as a basic movement tutorial and will get the player familiar with the world as they traverse back to the village.

Once they have exited the forest, they find the village bustling with mushroom people as two shrooms campaign to be the new leaders of the mushroom village. Both of the candidates speeches are filled with outlandish promises and responses that divide the mushroom people in dumb ways.

Moving past all the people in the village, **Mica** will run into **Grandmashroom** who will explain to **Mica** that today is voting day for the mushroom leader and THAT'S WHY IT'S SO DAMN LOUD. She will then give **Mica** the task of retrieving some cotton balls (or some better ear filling alternative) to dull the noise so she doesn't lose even more of her hearing.

As **Mica** ventures forth, they travel to the right of the village and traverse past a strange looking temple that the player is unable to access at that time. After climbing the temple and passing some more platforming challenges, the player will gain the item they need to help **Grandmashroom**. Conveniently, there will also seem to be a path that goes further out that the player will not be able to traverse until they have the Blade of Grass.

The blocked path forces the player to travel back the way they came. As they go back, they start to hear the chirping of a bird and may see the faint image of a bird flying past. As the player approaches the village to finish **Grandmashroom's** Quest, a cinematic is played showcasing the Town Hall being picked up and grabbed by a giant bird as the mushroom villagers scream out in horror. The town hall is then uprooted with the bird holding onto it with its talons as it flies away, dropping mushroom villagers along its path.

Once the bird is fully off the screen, the village is only left with a few villagers, **Grandmashroom**, the rest of the small houses, and a large crater where the town hall used to be.

4.3.2 Obtaining the Blade of Grass

After town hall is taken by the bird, **Grandmashroom** aggravatedly tasks **Mica** with following her outside the village. She takes **Mica** to the temple outside the village and proceeds to unlock the doors, using her cane as the key. Outside the doors torches

light up, revealing a flight of stairs that lead down. Grandmashroom forges ahead telling **Mica** not to be a scaredy cat.

As they enter the depths of the temple, the doors close behind them. Grandmashroom then presents to Mica the shrine of the "**Blade of Grass**" as it sits in the center of the shrine with a ray of light engulfing the sword.

As Mica picks up the sword, the ray of light disappears and the sword's small length is revealed. Shortly after, a door ahead opens, triggered by the blade being removed from its spot. **Grandmashroom** then explains a bit of the history of the sword and tells Mica that they will use this sword to save the village! (Even if **Mica** doesn't want to, because **Grandmashroom** is too old for this)

As **Mica** goes through the opened door they are led through a bunch of obstacles meant to teach the player the basics of the sword mechanic. Once finished, **Mica** finds themselves back at the path that was previously untraversable without the sword.

Grandmashroom is also somehow there at the end of the path.

4.3.3 Familiar Face

4.4 Player Agency

The player does not have much control over the narrative events that will occur in Mushroom Kid's Big Grass Sword. The only thing the player will affect is how certain NPCs respond to them depending on what dialogue choices they've picked. (Review Dialogue System)

4.5 Characters

4.5.1 Mica the Mushroom Kid

Mica is the main character of our story and is the character the player controls throughout the game.

Mica is a quiet kid who, before the major events in the story, could be found playing in the forest near the mushroom village by themself. They would often be practicing their jumps from branch to branch before eventually being called back to their village.

4.5.2 Grandmashroom

Grandmashroom is an old mushroom lady that is both wise and cranky. Harboring a long vivid and confusing backstory, she now spends most of her days telling inane stories and going for walks in the forest.

It isn't until the bird captures the town hall that her true wisdom and fortitude is shown, immediately choosing Mica as the heir to the legendary **Blade of Grass**. This is both because she is far too old to be saving the village and because she doesn't want to do it.

Throughout the game she acts as a mentor to help teach game mechanics and give hints on the solution to puzzles throughout the game. Her dementia also makes it so that her dialogue is always repeatable because she doesn't remember what she said previously.

4.5.3 Terry Trampoloom

A mushroom politician hoping to be the next leader of the mushroom village. He is a polite and dignified mushroom, which is why when he is eventually captured with the other mushrooms at town hall by the bird, his persona falls apart and he freaks out.

Eventually, he falls out of town hall and is greeted by **Mica** and learns about their quest to save the village from the bird. Not wanting to be outdone by some kid (and hoping to build back up his reputation) he joins Mica on their journey to save the village. If **Mica** rejects his offer then he begs them to join…

Using his bouncy head, Terry Trampoloom is able to help Mica get to areas that they couldn't reach before with the sword alone.

Since Terry is an NPC, he can be positioned in certain locations to solve puzzles.

4.5.4 Bird

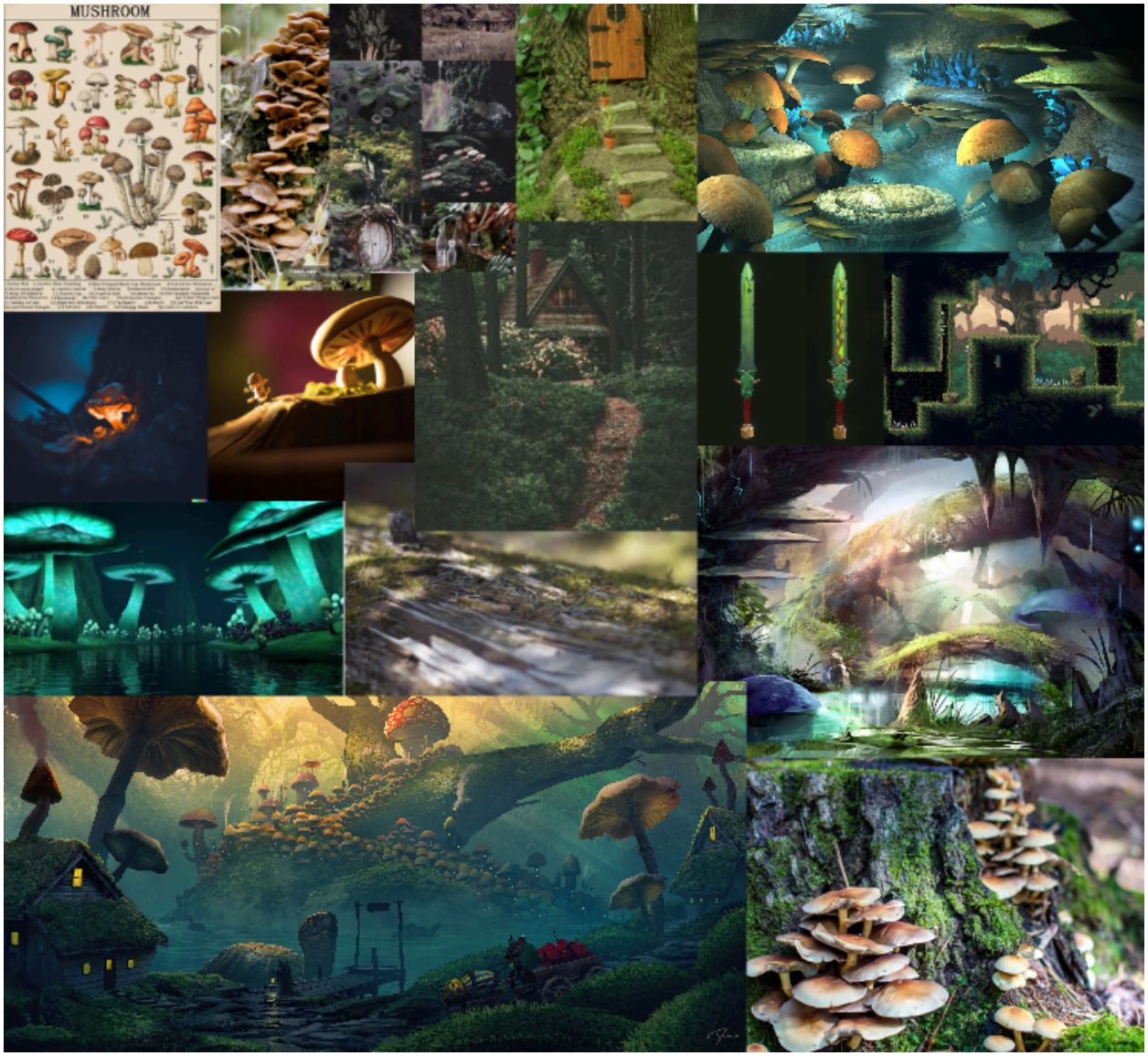
The antagonist of the game. The bird takes town hall for the simple reason of just needing supplies for its nest. There isn't much of a personality attached to the bird other than being ferocious, as the bird behaves almost the same as a real life bird.

4.5.5 Porcini

Political rival of Terry Trampoloom.

5. Visual Design

5.1 Mood Board



5.2 Art Style

The art style should aim to emulate the atmosphere and "homeliness" of Hollow Knight or Ori and the Blind Forest.

Digital Pros	Digital Cons	Pixel Pros	Pixel Cons

• Animation is more detailed and fluid • More detailed assets • Less generally restrictive • Can emulate different styles much easier • Special effects are more straightforward	• The animation process requires more steps and is generally more time consuming • More difficult to ensure a consistent art style between artists • More resource intensive	• Gives the game a charming indie feel • Faster art pipeline • Less room for error • Forces the art to be stylistic and adhere to the restrictions of pixel art • Simpler animations go further • Easier to keep character designs consistent	• Forces assets to be less finely detailed • Requires more effort to establish a unique style in comparison to other games • In-Engine standardization needs to be strict (next neighbor scaling, no compression, etc)

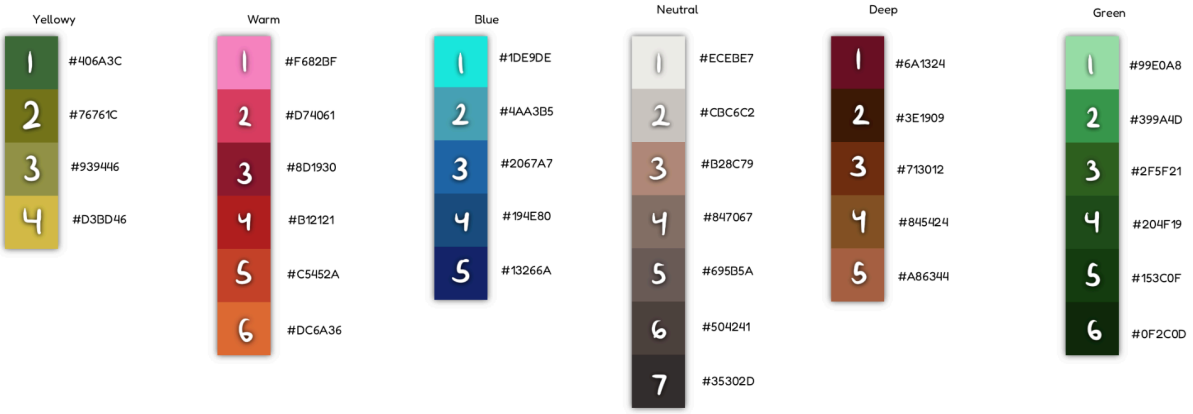
The art team has decided to go with a pixel art style for Mushroom Kid's Big Grass Sword. We feel this is the right decision for the game because as an art form, it has a much faster and more efficient art pipeline. Pixel art can also be used incredibly stylistically as we are planning to go with an internally lineless style, and we want the art in the game to be as rounded and natural as possible. This will be a fun challenge, and the result will be cohesive and stylistically unique. Also, pixel art will allow for easier cohesion between artists, a gritty natural feel to environments, and the animation pipeline in particular will take *significantly* less dev time than non-pixel art.

5.2.1 Style Guidelines

Important art style guidelines:

- All characters and objects should be internally lineless with a colored outer line
- No black!

- No right angles, all corners should be at least slightly rounded
- Color palette:



  - Use this as a guideline! Though, if another shade of an existing color is needed to make something look right, feel free to use it sparingly
- Keep things as round and soft as possible, EXCEPT for enemies or hazards. These can be sharp and stand out
- Environments have a 640x360 playfield
- PC sprite is 64x64
- Tiles are 32x32

5.3 Character Design

Character designs should be simple, yet thematic. They should also play off of the fun and silly nature of the game while keeping simple shape language and real mushroom references in mind.

5.3.1 Mica's Design

Mica is designed after a young mica cap mushroom and should have an easily recognizable silhouette. Their main palette consists of bright neutral colors that mesh well with accents of green. Inspiration was taken from other platformer player characters such as Hollow Knight and Gomez from FEZ.

Mica's cap is rounded and sectioned into 5 visible rounded pieces (10 all around) that span

in diameter about the height of Mica's visible body. These proportions should be kept relatively consistent to capture the squishiness of Mica's design. Mica's eyes will be visible during movement where the cap bounces or floats, and the general movement of the cap should be very squishy-looking. Mica's scarf adds to his silhouette and will also move with his movement. Hands stay detached for ease of animation.

5.4 Environment Design

The environment is going to be unabashedly "nature", featuring a homely, goblincore/fairycore/cottagecore village and distinct, natural scenery.

It essentially feels like a Pinterest aesthetic board, something foreign and grand, yet right outside your door.

It should be supportive of the platformer it surrounds.

5.5 UI Design & Typography

UI Design shouldn't break immersion, instead attempting to enhance the rustic, natural feel. Perhaps the writing system could appear carved into wood, or like the letters sprout up like mushrooms. Alternatively, it could emulate leaf-friendly alphabets like Balinese, Thai, and Telugu.

5.6 Shaders & Atmosphere

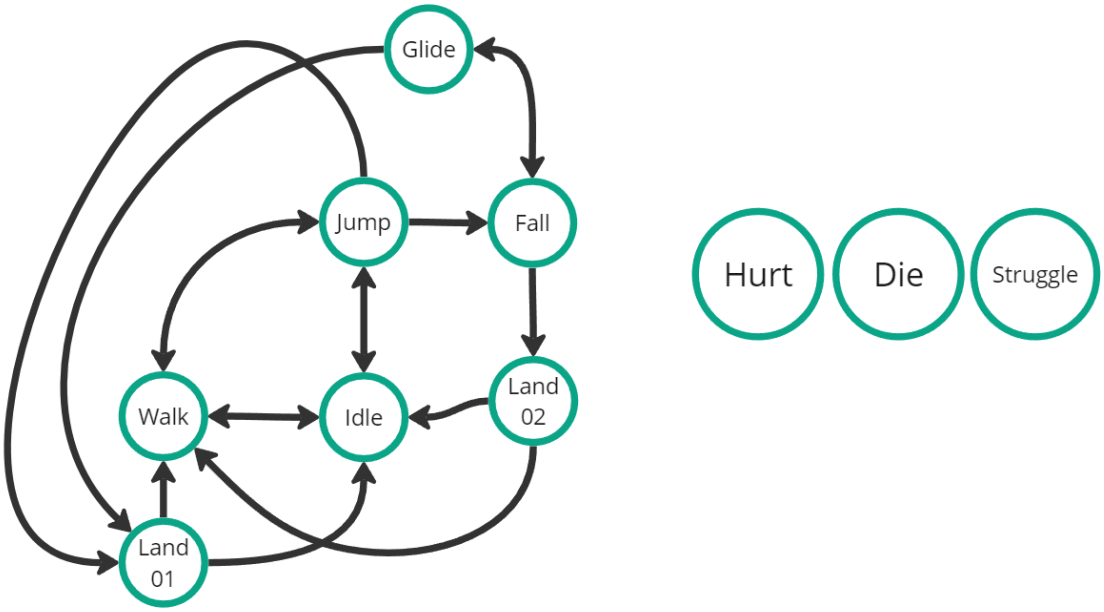
The shaders should make the game atmospheric and provide an extra dimension to the depth of the background and foreground. Things like dynamic lighting, shadows, color filters, etc.

5.7 Animation

All animation for both characters and environments should be as flowy and squishy-looking as possible. Exceptions to this include enemy animations or hazards.
Animations are played at 6 FPS and all sprite sheets should have 1 row of frames.

For example:

5.7.1 Mica's Movement



All animations tied within the graph have versions with (a) and without hands (b). Struggle does not require hands since it won't play until the player has the sword. Struggle is used when Mica is suspended above the ground because of sword-related platforming.

6. Audio Design

6.1 Music Design

The first pillar of the music design is the evolving spectrum of the "moodiness factor" through the game. In the lighter areas of the Forest and Treetops, the music should be more upbeat, acoustic, and simpler harmonically. On the other hand, the darker forest and cave areas should be more intense, as well as include more electronic elements. The second pillar of the music

design is our collaboration method between the composers. To keep coherence, we will agree on the instrumentation and any motivic ideas. To minimize stylistic differences, Andrew will focus on "setting music", with more ambient and drawn out compositions like in the Forest or Cave themes. Quinn will focus on the "action music" such as the cutscene and NPC themes.

6.1.1 Forest Music

The forest theme will start with one standard track that moves through multiple iterations as the player progresses. Instead of having separate tracks for "Rainy Forest'' and "Waterfall Forest" (for instance), different settings will just be subtle reharmonizations / reinstrumentiations of the parent track. New FX can be introduced to add atmosphere to the variations of the track. The "moodiness factor" will be set up with rich harmony and modal mixture.

6.1.2 Mushroom Village Music

The goal for the Village track is to make it upbeat, catchy, and fun. The player should feel at home, especially with contrast from the Forest theme (which the player hears first). In order to make the Village feel alive, many African and Latin percussion instruments should be used, almost seeming like the villagers are playing themselves. There will be hints of "goofiness" as well, and several ideas will be set up to be referenced in later tracks.

6.1.3 Temple of the Blade of Grass Music

The Temple music should feel ancient and nearly tribal. Sub/bass drones help to create this atmosphere. Consistent chanting drums will be used. The Temple should feel old, but not evil, so lighter minor harmonies are useful (think dorian, etc). When the player obtains the Blade, a transition into a slightly more upbeat version of the theme can be made.

6.1.4 Caves Music

The Cave Entrance should be more mysterious than dark. It should have high reverb levels and feel crystalline. Some electronic FX will be used for the atmosphere. The Cave Depths should be the darkest track of the game. It will be similar motivically to the

Cave Entrance, but be reharmonized in a darker mode. The rhythm and electronic FX should be borderline terrifying, making the player generally feel a lack of hope. Despite this, some sections can have hints of optimism with snippets of ideas from the Village track.

6.1.5 Treetops Music

Treetops should have distinct tracks for the first and second visits to the area, as well as the section climbing the branches. The tracks will share a common motif with a more uplifting/open feeling with a wide reverb space, with a hint of intensity as you approach the bird. Perhaps having music layers fade in as the player approaches the tree, and/or segments that change as it progresses. The 2 area themes will be similar, with the first being more confident and representing Mica's unpreparedness during their climb to the Bird, and the second representing Mica's growth by incorporating their theme. The climbing the branches section, after Terry helps Mica up onto the branches, will really build up Mica's theme with the treetops to feel very bold and heroic- perhaps incorporating ideas of Grandmashroom's theme too to associate the music with Grandmashroom's heroic maybe-sacrifice the first time players visited the treetops. Instrumentation can have more orchestration, esp. Strings and woodwinds.

6.1.6 Menu Music

Menu music should be simple and light, keeping the game feeling friendly and more fun to hide the 'turn' halfway through the game. Maybe a quick 'bridge' hinting at the dark parts, utilize multiple motifs to give callbacks to characters/areas/etc. If possible, having the menu music change in instrumentation/layers and build a bit as the player progresses through the game (simple w/ mica's motif at start of the game, light and adventury during the forest, very dark and ambient/basically no music during caves, bold for treetops, maybe have the credits music be the same as the finished game music, etc.). The music doesn't change too much melodically for most of it.

6.1.7 Cutscene Music

Can’t plan this out too much- recommend waiting until the cutscenes are storyboarded/prototyped/whatever before planning anything in particular

6.1.8 NPC Music

Grandmashroom: her theme should tie in with the temple/sword; hints that allow for a ‘heroic theme’ during her last stand; will need intro track, temple/tutorial track, bird battle cutscene track, maybe more

Terry Trampoloom: theme ties in with town hall; a bit anxious with feigned overconfidence, thinking like Alphys Theme; have an intro track, general track, 1st bird fight track, new general track, heroic final track at branches

6.1.9 Raw Music Recordings Library

6.1.10 B&A Master log and raw music storage

6.2 Sound Design

6.2.1 Forest Sound Design

Lots of leaf orientated sound design, specifically more focused on the higher frequencies to give off a lighter and crisp feeling of the autumn sensation. An example of such would be when you walk over a dried leaf, it makes a nice crunch noise. Inspiration should be taken from jungle ambience noises but toned down to ensure that there is space for the soundscape to paint a clear picture.

6.2.2 Mushroom Village Sound Design

A more modern approach of sound design can be taken here to give off a more civilized manner and feel to present the village. The core of mushroom village sound effects should be based off the mushroom language that will be created through the merging of existing languages. A good way to generate conversational sfxs.

6.2.3 Temple of the Blade of Grass Sound Design

Gloomy and dark, with a bit of suspense would work well here according to the narrative set for this location. Adding aspects of horror elements would be perfect to give rise to feelings of

uncertainty. Dissonant sounds provide a good way to generate the sense of uncertainty here.

6.2.4 Cave Sound Design

The most important thing for the caves section would be the applied reverb. One of the most protruding characteristics of cave sections within videogames is how the sound effects are projected. Echos and Reverb techniques should be implemented here. Low frequency sound effects would fit very well in this section to express the dangerous atmosphere.

6.2.5 Treetops Sound Design

This section contains the most sound effects that range in the higher frequencies along with wind sound effects. Mainly based on the wind that you would expect from going to a higher atmosphere. Equivalent to how the wind grows stronger as you hike up a mountain.

6.2.6 Environmental and Collision Sound Design

The sfx here should be based off of mushrooms colliding against specific types of materials. Raw recordings is the most preferred way to generate natural collision sounds with the environments within our game. Sound design through waveform shaping is also a good way to design something inorganic yet organic.

6.2.7 Raw Recordings Library

All raw recordings are stripped of any sort of modification, modulation, or effects.

6.2.8 Sound Design Log and Processes

7. Publishing and Marketing

Nowadays, we live in an age where it is not enough to just create a good game. If we want to truly compete on a professional level we are **REQUIRED** to strategize and market our game in order to raise funding and build up hype on a professional level.

7.1 Budgeting

Budgeting First Draft

7.2 Publishing Versus Crowdsourcing

7.2.1 What our team would want from a publisher.

It is important to explicitly state what we are looking for out of a publisher so that the expectations for the project are known on both ends when the pitch deck is sent.

Publisher Tasks Ranked:

1. Funding
2. Marketing
3. Localization
4. Porting
5. Trailer Production

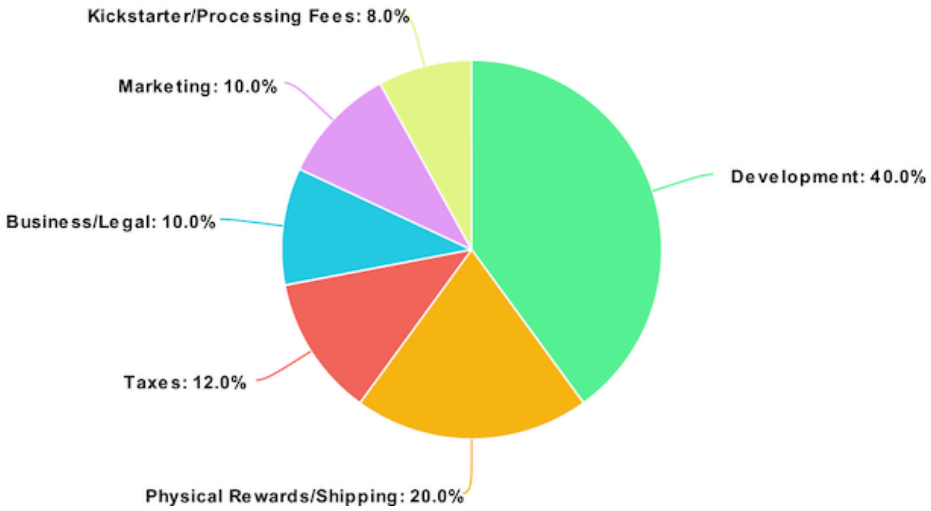
7.2.2 Publisher Pros & Cons

Pros	Cons
Financial Security: Publishers usually have a substantial amount of money to offer in upfront funding allowing our team to stay afloat to create the game.	"Indie publishing is not in a great space right now. It's harder than ever to stand out in the crowd, and publishers are looking to make fewer and safer bets, and doing a lot of due diligence" - **Ryan Alpert-Lowy** - **(Producer @ BANDAI NAMCO & Fromer Release Operations @ Armor Games Studios)**
Quality Assurance: Publishers can assist in ensuring the game meets industry standards for quality. They will be blunt about the product.	**Loss of Creative Control:** Our artistic vision will be altered while working with a publisher since they are now involved in the project journey.
Marketing: Publishers will champion a game to the very end which will help us in trying to get the word out.	**RECOUP:** Usually the team will not making any money off the game until the publisher has fully recouped the amount

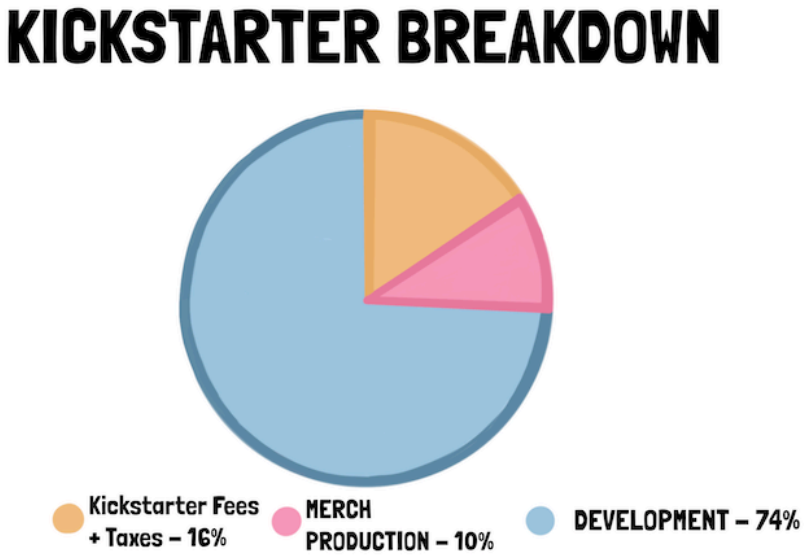
7.2.3 Crowdsourcing Pros & Cons

Pros	Cons
Creative Freedom: We have the power to do what we want with the game since the backers are not investors.	**Month of Development Loss:** Running a crowdsourcing campaign will take the team's full attention, meaning that the game will take longer to create.
Community Engagement: Running a crowdsourcing campaign may help us in our marketing endeavors.	**Reputation Risks:** Crowdfunding a game and then not delivering on it can hurt our reputation dramatically. Also if the crowdfunding campaign fails to reach its goal then we won't be able to secure a publisher in the future.
Profit Distribution: Since the backers are not investors we will receive a higher profit share when the game is released.	**Reward Tiers:** We will have to plan reward tiers that are both enticing and deliverable which may take time and money away from the development of the game.
	Kickstarter Cut & Taxes: Since this campaign would most likely be hosted on Kickstarter we would need to ask for even more money to mitigate the amount taken from Kickstarter and Taxes.

Gigasword's Kickstarter Cost Pie Chart:



[Unbeatable's Kickstarter Cost Pie Chart]:



7.3 Marketing Strategy

7.3.1 Target Audience

Mushroom Kid's Big Grass Sword generally aims to cater to almost all audiences no matter the skill level. Aiming towards an age range of around 10 and up.

7.3.1.2 Genre Interest

Players who like:

- Platformers
- Adventure Games
- Puzzle Games

- Indie RPGs
- Visual Novels

7.3.1.3 Feature Interest

Players who like:

- Unique Art Styles
- Physics Based Movement
- Speedrunning

7.3.1.4 Narrative Interest

Players who like:

- Comedy
- Character driven stories

7.3.1.5 External Interests

Players who like:

- Mushrooms
- Swords

7.3.2 Game Pitches and Messaging

Mushroom Kid's Big Grass Sword is a wacky narrative puzzle platformer about Mica - a little mushroom kid on their journey to save their village with a sword that grows.

After a bird captures the people of the mushroom village, Mica is given the Blade of Grass, a legendary sword that grows in the presence of sunlight. Throughout the game, Mica must find ways to overcome obstacles with their new blade, increasing its length with guidance from an elderly Mushroom named Grandmashroom to save the village.

7.3.3 Social Media Strategies

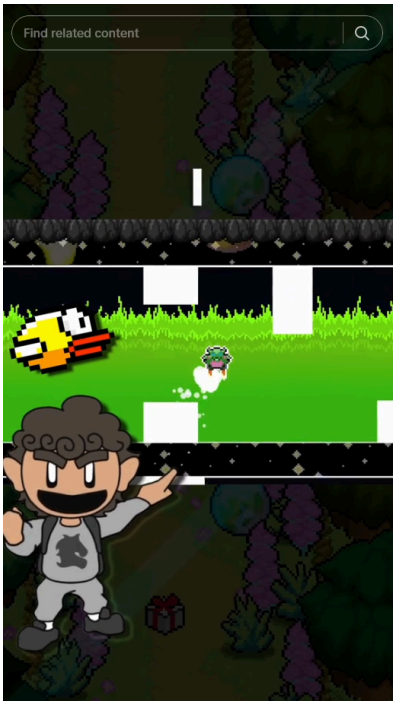
Important Notes:

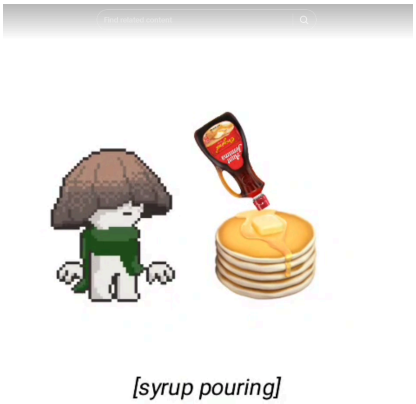
- Schedule Posts
- Engage with the community

7.3.3.1 Short Form Video Content

Short form content seems to be by far the most important area to target for our game’s marketing strategy as the conversion from

viewer to fan to wishlist is incredible and has caused many indie games in the past to achieve success.





References from: https://www.tiktok.com/@playmortalrite and https://www.tiktok.com/@athenianrhapsody

These videos should be formatted in a way that only showcases the game as a product but also present ourselves as individuals and indie developers trying our best to build the best game we can make.
Videos should be formatted in 3 separate ways:

- General Updates and Team Personality
    - These videos will show updates that are more personable about the team or the development process so more of the developer talking about it should be shown to make the video more personable.
- Gameplay Videos and Features
    - When a cool feature or significant piece of gameplay can be shown the video can be formatted to showcase the game's full screen size.
- Trend Based Video and Memes
    - These are special videos to make use of trends that are going on the app.
    - Should not be made too frequently.

7.3.3.2 Discord and Community Engagement

It is important that our game fosters a community and actively engages with it. Not only for marketing purposes but it acts as dynamic ways to make more content for the game and get feedback on features.
Areas of community engagement to look into are:

- Dev Streams
- Polls
- Competitions
    - Speedrun Competition
    - Picrew Competition
- Leveling System

7.3.3.3 Twitter & Similar Platforms

Content on Twitter and similar platforms should use specific key techniques in order to gain traction:

- Short post lengths
- Gifs and Videos
- 1-2 Hashtags
- General Engagement with the Community

7.3.3.4 Miscellaneous Platforms

Reddit is the wild west but posting should be done relatively often in subreddits that match the game since the platform has a good conversion rate.

7.3.4 Press List and Influencer Maintenance

7.6 Game Title Philosophy

7.6.1 Why not use the old title?

(Context: **Sword of the Morel** was our game's title for the game jam we participated back in **November 2022**)

Through showcasing the prototype at several different events we noticed some major issues with the title:

1. **Nobody knows what a morel is**
    a. Almost nobody who played the game knew what a morel was so several playtesters would not understand the meaning or would misunderstand the meaning of the game.
        i. "I only know morel are mushrooms from Stardew Valley"
    b. This issue is multiplied by the fact that if you say the name out loud to someone they think you are either saying "morale" or "moral" which is not what we want to convey at all!
2. **It doesn't tell you much about the game**
    a. Even though our game is about a sword that was owned by a morel mushroom, that still doesn't tell you enough about the main mechanic to really draw an audience in.

7.4.2 New Title Process

(Check: Title Name Feedback for detailed feedback)

During the ideation process for our new title, our ideas fell into 4 major categories.

- Fantasy-like Adventure Names
    - Examples:
        - Mica and the Legendary Lengthening Blade
        - Mushroom Mica's Adventure
- In-Game Lore & Description Names
    - Examples:
        - Mushroom Mica and the Blade of Grass
- Comedic Adjectives For Long Names
    - Examples:
        - Mushroom Kid and the Absurdly Long Blade

        - Mushroom Kid and the Absurdly Lengthy Blade that Doesn’t Stop Growing
- Pun Name
    - Examples:
        - Mushroom Mica's Big Grass Sword
    - *This was only discovered while receiving feedback*

As the titles were shown to other people we found that there were two major groups of people.

1. Those who prefer a more lore-based Name as the mystery is intriguing
2. Those who prefer a funny pun named with a euphemism built in.

We decided to cater towards our audience in camp 2 due to these reasons:

1. Our game is comedic
2. The pun doesn’t make the game harder to spell
3. The title is funny and interesting in spoken conversations
4. **The title makes people WANT to know more about the game**

At the end of the day, our success WILL be determined by the product we deliver, however, a title that can draw in initial interest in the game will be a deciding factor in if the game is even received at all.

Which is why we decided on the name:
Mushroom Kid’s Big Grass Sword

7.5 Marketing Socials

Site / Tool	Link
Mailing List	https://share.hsforms.com/1SXzN36uuTiWadUF7eboAPQq2pl7
Discord Server	https://discord.gg/6Vrrx6yhxh
Twitter	https://twitter.com/teambrokeki

	ds
TikTok	https://www.tiktok.com/@teambrokekids
YouTube	https://www.youtube.com/channel/UCt2CeeqmI88Z2rfj65Y_Qpg
Instagram	https://www.instagram.com/teambrokekids/
Tumblr	https://www.tumblr.com/blog/teambrokekids
Mastodon	https://mastodon.social/@teambrokekids

8. Resources

Title	Link
Google Drive	https://drive.google.com/drive/u/1/folders/1U_K3ArDwhfKDuqrrHP5dtVv5K-1-VPeN
Game Design Document	Game Design Document - Swor…
Trello	https://trello.com/b/9WZfUemC/sword-of-the-morel
Gantt Chart	Sword of the Morel Gantt Ch…
Miro Board	https://miro.com/app/board/uXjVMnNfWuE=/?share_link_id=325733709339
GitHub Repo	https://github.com/Fumpledump/Mushroom-Kids-Big-Grass-Sword