Game Design Document Template

---

1. Game Overview

1.1 Game Title:

Spring Star

1.2 High Concept:

An alien (you!) crashes upon a strange planet, and must use a spring escape capsule to gather your ship parts, repair your ship, and escape.

1.3 Genre:

3D Platformer

1.4 Target Audience:

All ages, little previous game experience required but preferred. Intended to be played on a controller, PC/Console. Gaming habits good for this game include those familiar with unique movement mechanics.

1.5 Platforms:

PC/Consoles. I don't see a good way to play the game on mobile platforms comfortably, because of how the game is controlled.

1.6 Key Selling Points:

- Unique spring controls, quirky platforming, and bizarre settings
- Smooth camera controls, silly alien tales, and a pastel but pleasing art style

---

2. Gameplay

2.1 Core Gameplay Mechanics:

The player moves using the spring, doing different forms of bouncing to navigate

- WASD movement - Make small bounces in any key direction
- Charge jump - Hold space to charge up a more powerful jump, launching in the direction the camera is pointing facing.

2.2 Objectives:

Collect the scattered parts of your ship as you explore the different biomes of an alien planet. Appease your space insurance provider.

2.3 Player Experience Goals:

We want the player to connect with Mweep's silliness. Additionally, older players can relate to Mweep's plight of suffering through poor insurance coverage. Overall, we want players to be intrigued by the world and invited to explore.

2.4 Progression System:

The player, or more appropriately the spring, bounces through levels to locate ship parts buried inside. The world is structured to have a sort of central hub, where you can unlock new areas by finding gadgets in previously visited ones. The player will essentially complete the game after collecting all lost ship parts.

2.5 Controls and Input:

The game is intended to be played on a controller, but works well with keyboard/mouse as well.

On the controller (I'll use XBOX controls to describe) you use the left joystick to move, right joystick to move the camera, A to interact with NPCs, RT to charge/jump. Those are the most essential binds.

On the keyboard/mouse you use WASD to move, the mouse to move the camera, E to interact with NPCs, and SPACE to charge/jump.

Other controls include showing the compass (TAB/RMB), map (SHIFT/M), FOV slider (1 and 2), and showing the controls (I). These are more subject to change.

---

3. Story and Setting

3.1 Narrative Overview:

An alien (you!) crashes upon a strange planet, their spaceship breaking up upon entry and scattering its parts among the surface. Your escape capsule is equipped with a special mobility device for navigating difficult terrain safely: a spring. Your goal is to traverse the hostile and surprisingly platformer-oriented geography of the planet to recover all the parts of your spaceship to repair it (or, more accurately, get someone to repair it) and get back off the planet.

In your travels, you may encounter upgrades to your spring as well, which you will need to reach every part. Try to get back before dinner!

3.2 World and Lore:

We don't actually have too much of this developed, but I can speak some on what we were planning. Obviously, we have an alien in an alien world. Some parts of the world may feel familiar or more earthy, but they will retain some unusualness in their design. Many of our levels planned would mess with gravity, which seemed like a good idea for a game dealing more with physics. We did not write much history for the world itself, so I've got nothing for that here.

3.3 Characters:

Our main character is the one controlled, our hero Mweep. Mweep is mainly exploring the planet during the game, curious about its inner workings. Mweep also encounters several NPCs, including Gob, the repair shop worker that repairs Mweep's ship while it searches for parts.

3.4 Story Structure:

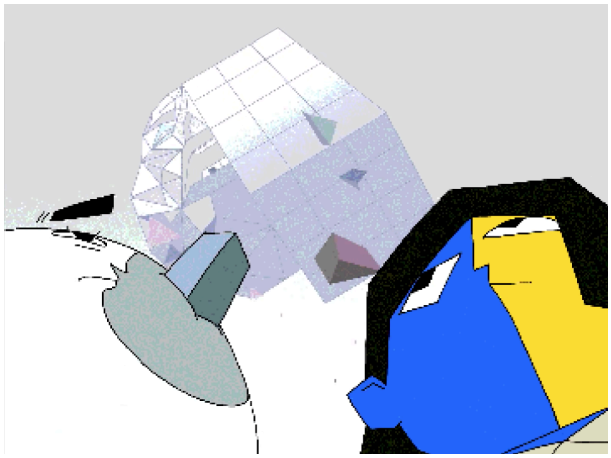
We didn't have a particularly ambitious story. It's Mweep looking for ship parts, very much like a mainline mario game plot. Mweep has to get these ship parts because its insurance for the ship does not cover the purchase of a new one. It will only cover the cost of repairs, which in reality would probably cost more than purchasing a new ship. Mweep's main motivation is to leave the planet to it can continue its research in space.

---

4. Art and Aesthetics

4.1 Visual Style:

The visual style of our game is low-poly for its geometry. I think that a neat inspiration is the youtube series ENA, including its recently released game, ENA: Dream BBQ. Pretty fun stuff. The game uses a pastel and vibrant palette, which we thought would fit the lighthearted tone of the game.



4.2 Character Design:

Mweep is designed to be colorful and round. Mweep's design was inspired by the design of little aliens from the lilalienz4ever social media account

4.3 Environment Design:

Our environment design is sort of a mixed bag, but they are meant to play into the unique movement of the spring. We start the player in a wide open desert so they can learn the movement, take them to a forest with tough sightlines but lots of options for verticality, a large tower where a wrong bounce can be fatal, a gravity lake where controlling the spring will test the mind, and more.

4.4 User Interface (UI):

The UI we planned for the game is pretty minimal. We didn't want too many things taking up the screen space, because the most important thing for the game is being able to see where you're going. We have a charge meter that appears during charging the spring, a toggleable map, and

toggleable compass that displays above the spring. Dialogue takes up some more screen space, but the spring isn't at risk of environmental hazard during dialogue.

---

5. Audio Design

5.1 Music:

The music is straight up goofy and silly. Alien beats and synthy tunes. Sorry, I don't really know how to describe it better. I can share with you the track we used for our demo, though. And another artist I took inspiration from.

Hourglass Meadow (ENA): https://www.youtube.com/watch?v=Urs2oS83Cw0

Alien Pop (Snail's House): https://www.youtube.com/watch?v=0hhLE63jHo0

5.2 Sound Effects:

The most prevalent sound effect in our game would be the bounce of the spring. This is to remind the player that they are controlling the spring. An exhaustive list of all sound effects in a game would be a monumental task, so we only implemented the one most important to the game given our time. We may have added audio cues for collecting a ship part and reaching a checkpoint, if given more time.

5.3 Voice Acting:

No voice acting. Or at least, not in any language we would know. All NPCs that speak to Mweep will have dialogue sounds similar to how Animal Crossing handles speaking for its NPCs.

---

6. Technical Specifications

6.1 Engine and Tools:

The sole game engine used to develop our game was Godot.For our project, we also used two free Godot plugins. One was Jolt Physics, which we used to improve some of the physics handling in the game, and Dialogic, an adaptive dialogue plugin. We used Jira to keep track of our tasks for the game, Github desktop for our version control, and Discord for general communication.

6.2 Performance Targets:

It's hard to determine exactly what our game would require, but I imagine the required specs are not very intensive. The game can run okay on my budget laptop, so that's a good sign. And,

because of the low-poly nature of the environment (which would probably be the most taxing thing), we aren't too concerned about accessibility for a broad range of capability in hardware. Our game has been running at 60 frames per second, which seems like a good target.

6.3 Online Features:

We did not plan for any multiplayer components. I'm sure there's something we could throw in for fun like speedrunning challenges and leaderboards, but it would not be the main focus of the game.

---

7. Level Design

7.1 Level Flow:

Initial levels will give the player lots of space to mess up. This is so the player can gain familiarity with the controls in a low-stress environment. Advancement in levels may require the player to learn less obvious properties of the spring mechanics, sort of an invisible tutorial.

7.2 Pacing:

As the game progresses, we want to make the player think deeper about the puzzles of platforming, and how they can use their knowledge of the spring's mechanics combined with accumulated gadgets to solve challenges, rather than just bouncing through a level quickly.

7.3 Key Locations:

The most important location is the central hub, where the ship parts are collected and used to repair Mweep's damaged ship. Other locations will have importance to Mweep as they will contain crash-landed ship parts.

---

8. Monetization and Distribution

8.1 Pricing Model:

I imagined that the game would be priced at 10 dollars or less. I wanted to set a price that roughly reflected the amount of time players would spend in it. It's also a game that probably wouldn't be played more than once for most people, but still good to go back to. So yeah, 10 dollars or less. Maybe more depending on how long it takes to get through the world. There would be no additional costs past this.

8.2 In-Game Monetization:

No.

If there were anything released after the full release of the game, it would be expansion of content through the world that would probably be done for fun and not charged for. That, and free bugfix patches.

8.3 Distribution Platforms:

Steam for sure. Epic Games Store, GOG. I'm not sure other than that. Plus any built-in console stores.

---

9. Production Plan

9.1 Development Timeline:

Very roughly, I see a few big phases: Movement mechanics, additional UI mechanics, Level Design, and World Development. Which is pretty much the setup that we had for our team.

9.2 Team and Roles:

I believe that our team roles would be exactly the same as outlined in our team contract, which you should have a copy of. The roles that we made worked very well for our development team.

One lead that helps with all areas, Assets, Mechanics, and Level Design/UI. Perhaps splitting Level Design and UI into separate roles would be a good idea, and it's probably what we would've done if we had a fifth person on our team.

9.3 Budget:

Considering that we got this far without spending so much as a dime, I'm pretty confident that the cost of development would be pretty low. That's not factoring in wages and hours, though. Marketing would probably be the most expensive part. Post-launch would most likely be exclusively bug fixes, which would once again boil down to wages and hours.

---

10. Appendices

- You can view our previous slides for concept art. I did make a prototype project before we started this class, which I believe I showed to you.