



Starchy Games

Sisyphus

Request for Proposal

Version 1.12

Document History

Version	When	Who	What
1.0	9/12/19	Graeme H	Template
1.1	9/15/19	Max M	Section 6
1.2	9/17/19	Damon S	Section 2
1.3	9/17/19	Damon S	Section 3
1.4	9/17/19	Max I	Section 4
1.5	9/17/19	Damon S	Section 1
1.6	9/17/19	Max I	Section 5
1.7	9/17/19	Max I	Section 8
1.8	9/17/19	Max I	Section 9
1.9	9/18/19	Max M	Section 6
1.10	9/19/19	Graeme H	Section 5
1.11	9/19/19	Graeme H	Section 10
1.12	9/19/19	Graeme H	Revisions

Table of Contents

- 1.0 Problem description
- 2.0 Project Objectives
- 3.0 Current System(s) – if any or similar systems
- 4.0 Intended users and their interaction with the system
- 5.0 Known interactions with other systems inside/outside the client organization
- 6.0 Known constraints to development
- 7.0 Project Schedule
- 8.0 How to Submit Proposals
- 9.0 Dates
- 10.0 Glossary of terms

1.0 Problem description / opportunity / expression of need

Our company needs an interactive, innovative, and new 2D platformer game that will capture the attention of gamers across the globe. Our goal is to show the industry that our company is a serious competitor capable of creating fun and engaging video games. We believe this is possible with a 2D platformer game. The question we hope to answer: “How does one make an original game that is challenging, thought-provoking, and groundbreaking in a long-stagnant genre?” This platformer must stand out from other games that have been made in the past.

The platformer is a very popular genre with historical ties to the classic video games of the 1980s. In modern times, the video game market has become oversaturated with platformer games. Modern platformers are not innovative and often simply reiterate the ideas that made them successful in the past, causing the once popular genre to fall to the wayside.

Since most companies have failed to innovate with platformers in the past, we want to show the industry that even a platformer can be an immersive, innovative, and dynamic experience for the player. To facilitate this, we propose a platformer with a huge emphasis on dynamic level/map design and gameplay. This lets the game stand out from the crowd from the start and shows the industry our capabilities and creativity.

2.0 Project Objectives

1. Gameplay Goals

- a. The main goal of the game is to travel as far as possible in the game world. The score kept for the player should be based on the distance travelled from the start point. The player will need to survive for as long as possible to get the greatest distance travelled.

2. Dynamic Gameplay

- a. Map needs to be generated randomly and procedurally.
 - i. As the player explores the map, it should be different from the last time they played.
 - ii. Sections of the map can generate in different ways. So, there may be a mountain section with steep cliffs and ridges, or a desert biome that is relatively flat with sand everywhere. Each of these map sections need to have their own textures, assets, and entities that spawn within.
- b. The map must be made so that the player cannot tunnel underneath the ground.

- c. Certain map sections will only spawn when the player reaches a set distance from the beginning of the game. These sections can be more challenging for the player so the game gets more difficult over time.
 - d. The map must be generated in such a way that the player can feasibly traverse it, meaning restrictions on the random generation algorithms.
- 3. Player
 - a. The player can move throughout the game world via input from the keyboard, or other input should the game be ported to other platforms.
 - b. If the player takes too much damage from sources in the world, such as other enemies, then he/she must die.
 - i. The game ends once the player has died, and to continue playing, the player must start over in a new world with the score reset.
 - c. The player can collect power-ups that affect the player in some way (such as jumping higher, running faster, etc.)
 - d. The player sprite must be modeled after Sisyphus from the Greek myth. He should be carrying a boulder as he journeys throughout the world.
 - e. The player should be able to attack enemies (see below) by swinging and/or throwing his boulder, which he must retrieve afterwards.
- 4. Enemies
 - a. To provide a challenge for the player, as he/she moves through the game world, sections of the map will be naturally populated with varieties of enemies.
 - b. Each of these different kinds of enemies will have unique AI behavior, weapons and attacks, appearance, sound, and overall difficulty. For example, there may be a "Titan" enemy that is melee-based and can take a great deal of damage. There also may be just a simple "Grunt" enemy with little health and damage.
 - c. All enemies must be modeled after mythical Greek creatures to go along with the game's theme.
 - d. Each of these enemies, through their respective attacks, can damage the player and ultimately cause the game to end.
 - e. When an enemy dies, there will be a chance for power-ups to be dropped.
 - f. Enemies that spawn and are placed in the map depend on the current section of the map the player is in. Harder enemies will spawn in sections of the map that are further away from the beginning of the game, and certain enemies may only spawn in specific biomes.
- 5. Interactable Objects
 - a. There must be other objects in the world the player can interact with called "power-ups".
 - b. Power-ups must provide the player with some benefit, such as restoring health, giving a speed boost, increasing damage, or the like.
- 6. Weapons
 - a. The player must carry a large boulder at all times.
 - b. The player can use the boulder as a weapon, whether by swinging, smashing, or throwing it.

7. Intuitive and Easy gameplay
 - a. The premise of the game and how to play the game must be simple. This is not going to be a complex title, but rather should be a game that anyone can pick up in a few minutes so they can jump right into the experience. This means easy movement controls with only movement for left and right and jumping. The player should be mobile enough so they can navigate through the game.
 - b. The only other control required would be an attack. The player should have three different attacks that are automatically rotated through. The attacks may or may not be functionally different, but they should all have different animations.
8. Background and setting
 - a. The game needs to feature a day/night cycle that runs as the player plays the game. This will change the appearance of the current map significantly and make it brighter or darker based on time of day. The approximate length of a full day cycle should be 5 minutes.
 - b. The section of the screen behind the map must also be occupied by a background that resembles the biome the player is in. For example, in the desert, the background may look like endless dunes.
9. Audio
 - a. The game must feature a soundtrack that plays in the background.
 - b. This soundtrack must loop fairly often to capture the endless nature of the game.
 - c. The soundtrack should be overlaid with other noises for combat, movement, or enemies.
10. Save Progress
 - a. The game must feature the ability to save progress so a player can come back to their original save when they have the time.
 - b. Once the player has died, however, that specific saved game can no longer be played, and a new game must be made.
 - c. The player may not have more than one saved game at a time.
11. Appropriate Menus
 - a. A Start menu will open on game startup that contains the following:
 - i. A button to start a new game from the start with a new map.
 - ii. A button to continue from the most recently played game save.
 - iii. A button for the game options.
 1. This will include screen resolution, texture quality, volume, etc.
 - iv. A button that brings up a written tutorial on how to play.
 1. This will include the lore behind the game, and the general premise/goal of the game.
 2. In addition, It will describe the following:
 - a. How to move
 - b. How to attack
 - c. Types of power-ups

- b. There also must be an in-game menu that the player can access by pressing the escape key.
 - i. Upon opening this menu, the game will be paused, and a new menu will open on top of the game.
 - ii. This menu will include the following:
 - 1. A button to resume the game
 - 2. A button to save and exit to the main menu
- 12. UI Elements
 - a. A minimal HUD must always be on the screen during gameplay that shows essential information about player health, power-ups, and the current score. This HUD will be in remote parts of the screen (corners) so the player can focus on what's happening in the game.
 - b. Health data of enemies will also be shown just below the enemy, so the player knows when to engage or flee. This will be in the form of a dynamically sizing bar that uses a color scheme to show how much health is remaining (e.g. green for healthy, yellow for medium, and red for low).
- 13. Platform Support
 - a. The game must be able to run on a 32- or 64-bit Windows platform.

3.0 Current system(s) – if any / similar systems

There are a variety of systems that have similar features to the project at hand, and could be seen as influence / inspiration for it. The systems listed below have similar features to what Sysiphus will have:

- 1. *Terraria*
 - a. The game Terraria is a 2D platformer game that features random map generation so that each time the player creates a new world, what they find and see in that world is unique.
 - b. The player can create and wield many kinds of weapons and tools.
 - c. The player is pitted against hordes of enemies.
 - d. The map features biomes that generate in unique ways with varying textures and structures.
 - e. The random map generation in Sysiphus will be similar to that in Terraria. Terraria features a wide variety of maps and biomes, which should be captured in our platformer.
 - f. Terraria's worlds are destructible, which we do not want for our game.
 - g. Terraria features placeable objects, which we do not want for our game.
- 2. *Super Mario Brothers*
 - a. This game is very well known for being a classic 2D platformer from the 1980s. Even though this is decades ago, there are some elements of this game that we would like to examine.

- b. The game features static levels that are meticulously designed by the game developers.
- c. The game features several different kinds of maps with unique blocks, textures, and enemies.
- d. The game is very intuitive to play, and anyone can pick up the game and begin playing in no time. Sisyphus should also be very easy to learn.
- e. Each of the maps have a different feel and different kinds of enemies.

4.0 Intended users and their basic interaction with the system

- Developers
 - Those of us developing the system to learn/experience the creation of projects with other contributors.
 - Shall interact with the system in-depth to the point of altering and creating all aspects of the system.
- Dr. BC
 - The professor who holds the developers' grade in-hand, and shall be a god when in the game to ensure enjoyment and satisfaction.
 - Interacts with the system in a speculative, yet gleeful manner.
- Gamers
 - Any one person who downloads/plays the game, for the simple purpose of enjoyment.
 - Interaction limited to normal mode/gameplay, and will not be able to affect the system in the way a developer may, unless they contact the developers and developers choose to add or change features.

5.0 Known interactions with other systems within or outside of the client organization.

- High-score server
 - An external system will keep track of high scores.
 - New high scores will be sent to the system, and old ones will be queried from the system.
- Any app/game store (i.e. Apple App Store, Google Play Store)
 - Should the game be released on mobile, interaction with the local play store for purchases and scoring will be required.

6.0 Known constraints to development

1. Development team must build for Windows. Although libraries and save files transfer well between platforms of Unity (Windows, Mac, Linux), testing and compiling will be targeted for 32- or 64-bit Windows.
2. One branch of the Github repository per developer will be used. Any changes a developer makes must be committed to own branch, and then pulled to the master branch. Attempting to develop multiple versions of the game would cause errors and confusion when all game assets are consolidated later, so frequent consolidation of code during construction is important to prevent this.
3. Team members are limited to contributing to their team roles. If a member wishes to contribute to another's branch, they must submit a pull request.
4. The game must include a Dr. BC mode which is unchallenging and trivial to complete. This mode is useful for grading purposes and may or may not be included in the post-evaluated design.
5. The game must include some form of plagiarism, which will be detailed in either the champion documents or reports. This plagiarism is only needed for evaluation and should be removed afterwards.

7.0 Project Schedule

Date	Description
9/19/19	SA Demos RFP Due
9/26/19	Weekly Status Report (WSR) 1 Class, Sequence, Gantt Diagrams Due Repository Chosen and Submitted
10/03/19	WSR 2 Everyone should have skeleton code that will "Compile and Run"
10/10/19	WSR 3 Have two compile paths (test and production) Merge all test plans into a single test overlord Test Plan Due
10/24/19	WSR 4 Should start looking like a game

	Show code running Show merged Gantt Chart Overlord Gantt Chart Due
10/31/19	WSR 5 Show running code Update Gantt Coding Standards Due
11/07/19	WSR 6 Show code running Update Gantt User Manual Due DEADLINE Proposals
12/05/19	Post Mortem Presentation
12/12/19	Final Presentation Proposal Winners Announced

8.0 How To Submit Proposals

Please submit proposals electronically through the Starchy Games Discord “proposals” text channel linked below:

<https://discord.gg/tYtwfEh>

9.0 Dates

All those interested in submitting proposals, please do so by 12:00 PM, Thursday September 26th, 2019. Winners will be announced at 5:00 PM, Tuesday October 1st, 2019.

10.0 Glossary of terms

Sisyphus - A man from a Greek myth who was cursed to push a boulder up a hill forever.

Dr. BC Mode - A mode for testing purposes where the game is either impossible to lose or trivially easy.

Biome - An area of the world characterized by similar terrain types.