**Sparkles Magic Labyrinth**

**Labyrinth Studios**

**Table of Contents**

**User Description**………………………………………………………………..2

* Brief Description……………………………………………………….....2
* Instructions ……………………………………………………………….3

**System Requirements**………………………………………………………..…4

**Restrictions**……………………………………………………………………...4

**Technical Requirements** ……………………………………………………….5

* Characteristics …………………………………………………………….5
* Project Overview………………………………………………………......6
* Programming …………………………………………………………...…7
* GUI………………………………………………………………………...8
* Server……………………………………………………………………..10

**Feasibility Study**………………………………………………………………...11

* Operational………………………………………………………………...11
* Technical…………………………………………………………………..12
* Economic…………………………………………………………………..13
* Hours/Cost Breakdown………………………………………………….…14

**System**…………………………………………………………………………….15

* Kernel………………………………………………………………………15
* Standard……………………………………………………………………16
* Super……………………………………………………………………….16

**User Description**

* **Brief Description**

The main Objective of our project is to navigate through the 2D top down Labyrinth in search of your beloved cat Sprinkle’s. The interaction will be done using the arrow keys to move the player, right mouse button to fire arrows at enemies, left mouse button to interact with collectables, and the space bar that allows the player to blink to avoid damage. The game will be simple to navigate similar to other top down 2D games.

* **Instructions**

After opening he program the users will view a start menu where they will enter the Labyrinth by clicking the start button. Once the game starts they will use the arrow keys to navigate through the various quadrants, using the left mouse button to pick up collectables such as rubies, emeralds, sapphire, and gold that all have special properties designed to help the user complete each quadrant. The player is equipped with a bow and arrow that can be used to defeat the various bosses they will encounter. The user can press the escape (Esc) button to open the pause menu where they can view their current health, inventory, settings, save state, load a previously saved game, or exit the application. Once the user has found Sprinkles the game will end and they can view their stats, health, coins, collectables, ect..

\*a Screenshot of the menus and game play will be inserted here as an example once we finish the development of these objects

**System Requirements**

* Any Device that supports unity (generally a PC or Mac)
* Internet connection (to download the game)

**Restrictions**

The main drawback is that any person wishing to play the game must use a PC that has the unity 3D engine and C# installed.

**Technical Requirements**

* **Characteristics**

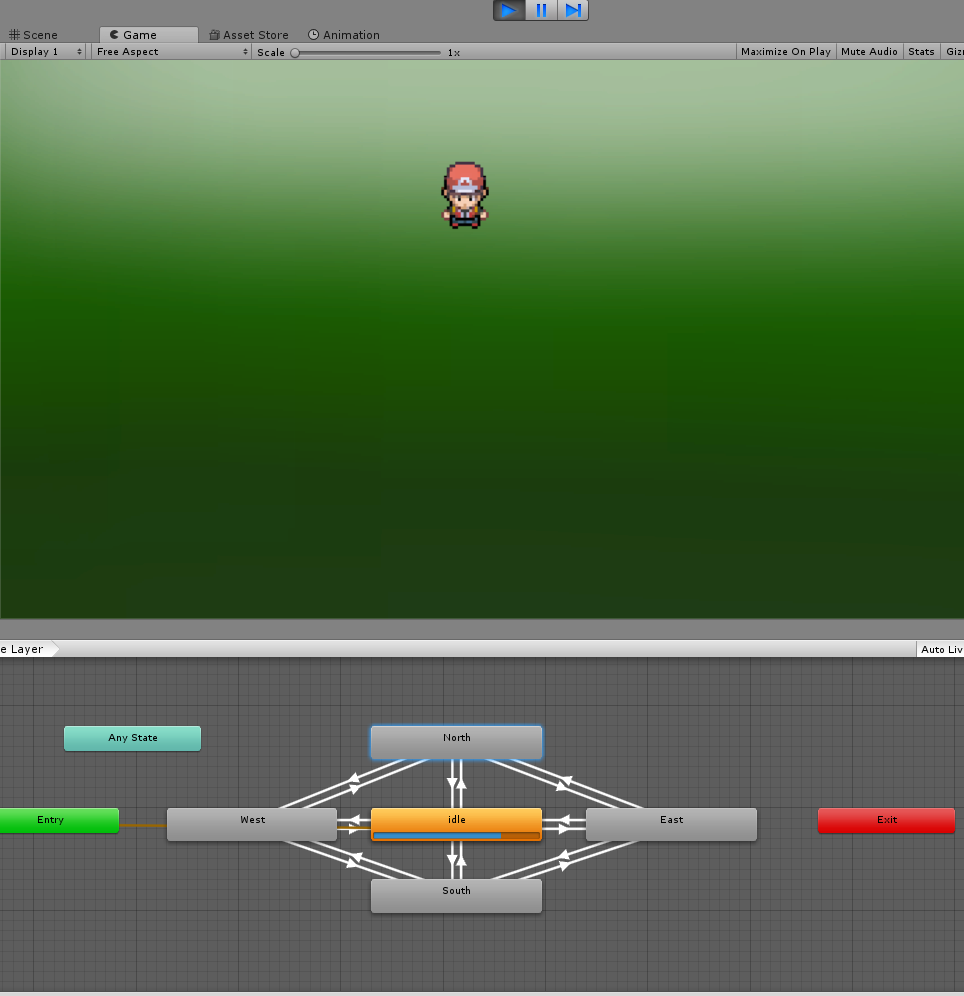
The game will be constructed so that any device that supports the unity 3D engine, and the C# language will be able to run it. At present the plan is that the user will enter a Labyrinth where they will interact with the game. Once the initial quadrants are constructed and working properly we will add more levels, and possibly more players. Overall the game is to be implemented on personal computers, but we won’t judge if you play it on your work computer.

\* insert a screen shot of the start menu or view of the game map

* **Project Overview**

The initial plans for the project include a Labyrinth with 4 quadrants that the users must complete in order, quadrant 1 through quadrant 4, to access the next quadrant and ultimately win the game. Once we have a fully functional game we may later add additional levels with increasing difficulty. The design intended to use the Unity 3D engine, which will be discussed in further detail in the programming section. Testing will be done on the various machines owned by the group to ensure cross compatibility.

\* screenshot of an early version of the game



* **Programming**

The project will be programmed using the Unity 3D engine. The Reason we chose Unity is the ability to build a beautiful yet simplistic user interface without having to build a custom game engine from scratch, that can be run on any device with C# installed. This makes the game not platform dependent and allows consumers to play on multiple platforms. The simple 2D top down design also means that the game can be run on most machines without consumers having to worry about purchasing additional hardware. The code will be compiled and developed within the unity 3D editor. The project will implement many of Unity’s built in classes and engines. These include the 2D Physics engine, and the C# scripting tool. These packages may change as we add more levels and functionality. The 2D physics engine and C# scripting will be used mainly for player interaction within the Labyrinth. This is the most dynamic portion of the project as additional features and functionality may change the implementation of these classes and engines.

* **GUI**

The main portion of the project will be developed around the GUI, the Labyrinth and menus. This will provide a simple interface for the user to change settings, view stats, and interact with game objects. The GUI is the most important part of the project as it contains most of the games functions.

\* an example of the GUI will be placed here once it is developed further

**Feasibility Study**

* **Operational Study**

We are designing the game to have at least one quadrant of the map, that is, one entire section of the game, complete and playable by the deadline. After we finish designing and coding (this includes player movement, item pick-ups, enemies, music, etc.), we will work on the other quadrants of the game.

* **Technical Feasibility**

Our team has minimal experience, but access to enough resources to complete this project. Everyone has the hardware capabilities as well. The game is simple enough for a novice team of programmers to complete, given our set of skills, but challenging enough to take the allotted time given to us. The game, once completed will only be playable by systems with access to the Unity engine.

* **Economic Feasibility**

System

      This section will discuss the projected goals for the project based on the time give. This will include the Kernal, Standard, and the Super.

Kernal (ten weeks)

Our project will be a full, short top-down 2D game created through the Unity Engine.  I/O setup, functioning menus, and much more will be involved. Our system will allow the following:

o   Reading of Keyboard/Mouse inputs

o   Real-Time recognition of input commands

o   Simplistic yet immersive story

o   Multiple screens, simulated puzzles and combat

o   Original Artwork, Concept, and Sprites

o   Custom Animations

Standard (12 weeks)

The core project will be operational. It may lack aesthetic features such as ambient lighting, however the core gameplay will be functional, screens will be complete, and the game will be playable/runnable on any Unity-Compatible system as defined in our Technical Requirements.

Super (14 weeks)

If we were to be given two more weeks we would be able to add a number of improvements such as:

o   More features and in-game effects (lighting, moving, etc)

o   Code optimization to allow better framerate/compatibility with sources

o   More screens

o   More Intelligent AI/Scaling Enemies

o   More Puzzles, Original Music

Proposal:

Group Members: Colin Crowe, Luke MacLean, Eddie Gao, Garret Spence, Anthony Gary  
  
Sparkle’s Magic Labyrinth is a top-down 2d pixel game which is based off the Unity Engine. The player will be able to explore a maze-like world, collect items, solve puzzles, and battle enemies in order to escape the maze. A lighthearted story revolving around saving your pet cat, Sparkles, will help move the player through the world. This game will be similar to other popular games such as Hyper Light Drifter and The Legend of Zelda: A Link to the Past. We have designed the game in 4 parts, having each “quadrant” of the Labyrinth being potentially standalone. This way, as we progress through the project, we can either add or remove quadrants to remain in the timeframe. The game itself will have players navigate using WASD, use various abilities to affect movement and the environment, and use problem-solving skills.

Since we are using Unity as a driving engine, any computer capable of running Unity (Windows, Mac, Ubuntu, etc.) will be able to run this game. It will have a Menu, Save States, Original Art, and more.