Project Description Document

for

CS4397 Final Project: Mario 2D

Group Name: Super Sonic Fox Beam
4/27/17

Group Members: Rachel Cramblitt
Richard O'Neal
Christine Hueber
John G. Toland
Jessica Hao

Table of Contents

Introduction

This document is intended to inform the reader of the process of designing and implementing the Mario 2D game. The section *Project Description* will provide a detailed description of the game play environment, tools used, code management configuration, and the core features. Paper prototyping and artwork related to the design of the project will be described in the *Paper Prototype* section preceding the *Project Description*. Sprite animation and functionality related to each animation will be discussed in the *Sprite Animation* section. Last but not least the resources for asset acquisition will be provided in the *Resources* section at the end of the document. These resources include the following: sprites, sound effects, music and foundational feature implementation.

Project Description

Overview

The Mario 2D application is a two dimensional game made in Unity. The game is a side scrolling clone of the Super Mario Brothers game for Super Nintendo, directed and designed by Shigeru Miyamoto. The objective of the game is to travel through the world and avoid the enemy Gumbas that are out to hurt Mario. The game is won by the player when Mario has completed all the levels.

Mario 2D is mainly written in UnityScript, better known as JavaScript, which made the project a little more challenging to code. Unity has deprecated many JavaScript functionalities attempting to weed out the use of JavaScript in the Unity environment. Furthermore, the default editor, MonoDevelop does not offer any code completion for JavaScript. This makes syntax mistakes harder to debug and identify. There is also a little of C# used in the project; these scripts control the Graphical User Interface (GUI).

Camera Functionality

CameraSmoothFollow is a script attached to the camera allowing it to follow Mario throughout the game. This action achieves the feeling of a scrolling background as the player moves through each level. The camera will stop when the game is paused, Mario dies, and when the game is won.

Mario Functionality

The Mario character is controlled by the player using the Directional Pad (D-Pad) to move on the horizontal and vertical axes in two dimensions. The character jumps in the air when the player presses the spacebar. Holding the control button while walking will allow the character to run. These controls are for the standalone builds.

The Android build has buttons that display on the handset screen that give control of the character to the player. For movement in the horizontal and vertical directions, there is a virtual joystick that mimics the functionality of the D-Pad. The “A” button will allow the character to run and the “B” button allows for the character to jump.

Mario is injured or killed depending on the health of the character when he encounters an enemy gumba. Mario is instantly killed when he falls off the ground into the abyss.

Enemy Gumba Functionality

Enemy Gumbas are controlled with a navigation script that mimics artificial intelligence (AI). The AI is further advanced with path nodes that exist in the level and are transparent to the user. When a gumba enters a path node the navigation script will change according to the respective command related to the path node entered. This helps to fool the user into thinking the gumba is making decisions on the fly.

The gumbas have a search range, there area of Unity units where the gumba will search for the Mario character. Gumbas also have an attack range, an area where if Mario enters the gumba will increase speed and advance towards Mario. If Mario and the gumba collide, then Mario’s health is reduced. If Mario is small when he collides with a Gumba, then Mario will die.

Mario can destroy the enemy gumbas if and only if the characters feet collide with the enemy gumba’s head. The Boss Gumba is larger in size, relative to Mario and the other enemy gumbas, and requires three collisions with Mario’s feet and the Boss Gumbas head.

Pickup Functionality

There are four different types of collectables (PickUps) in the game environment. When Mario collides with any of the four pickups, then the pickup is applied to the *PlayerProperties* script and is essentially reflected on the GUI or in the physical appearance of Mario.

Available Pickups

1. Coins: Mario collects coins and the GUI updates the total coin count
2. Lives: Mario collects a green mushroom, giving the player an extra life (increment by 1)
3. Mario Big: Mario collects a red mushroom, increasing Mario’s health and size
4. Mario Fire: Mario collects a fire flower, changing Mario’s appearance and allowing the player to shoot fireballs at enemies

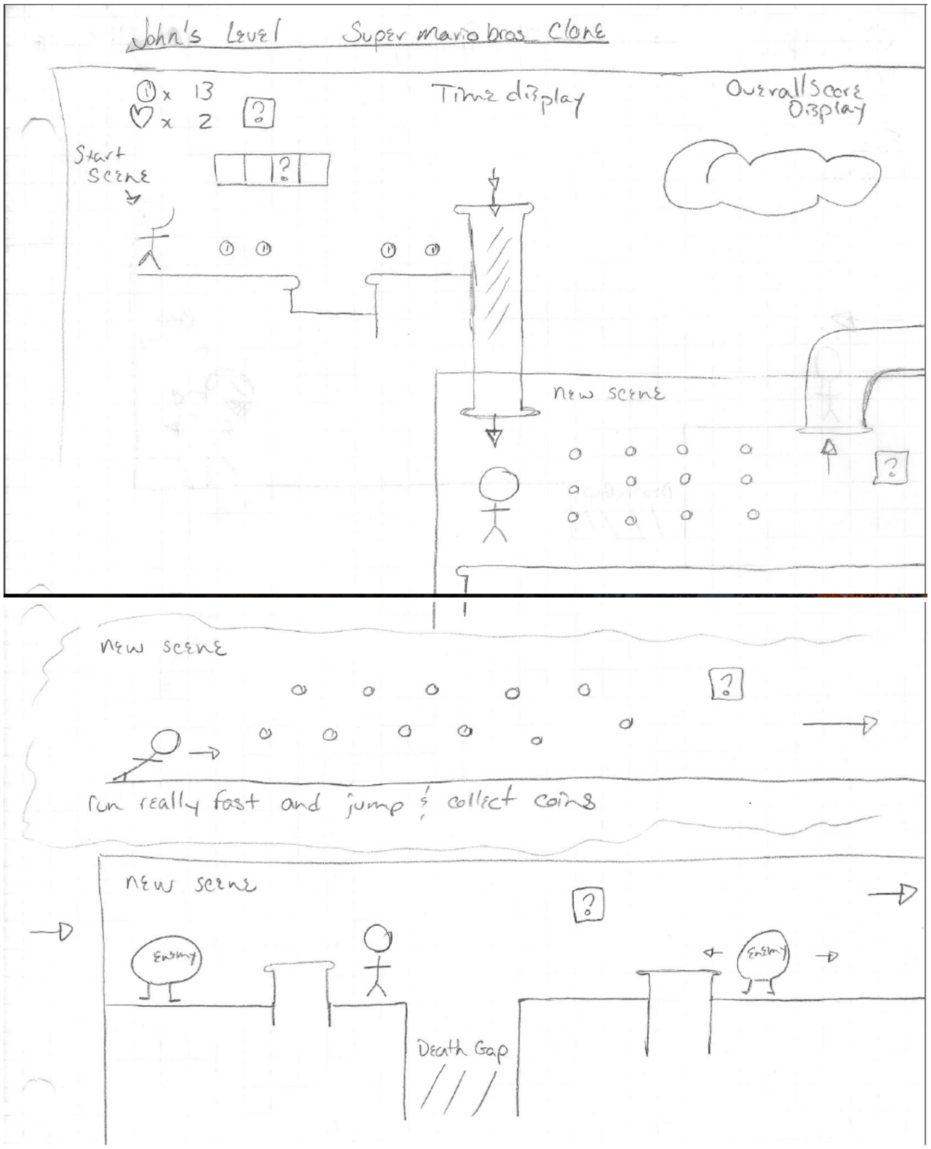
Graphical User Interface

The Graphical User Interface (GUI) consists of two parts. First is the Heads Up Display (HUD) which displays the players lives, coin count, time left on the clock, and the pause button. The second part is the actual GUI that allows the displays when the pause button is pressed. This interface allows the player to exit the game, adjust the volume, and switch to a different level (If able to implement before due date of project).

Paper Prototype

The following figures represent the storyline and artwork created during the design phase of the project.

Figure 1: John's World



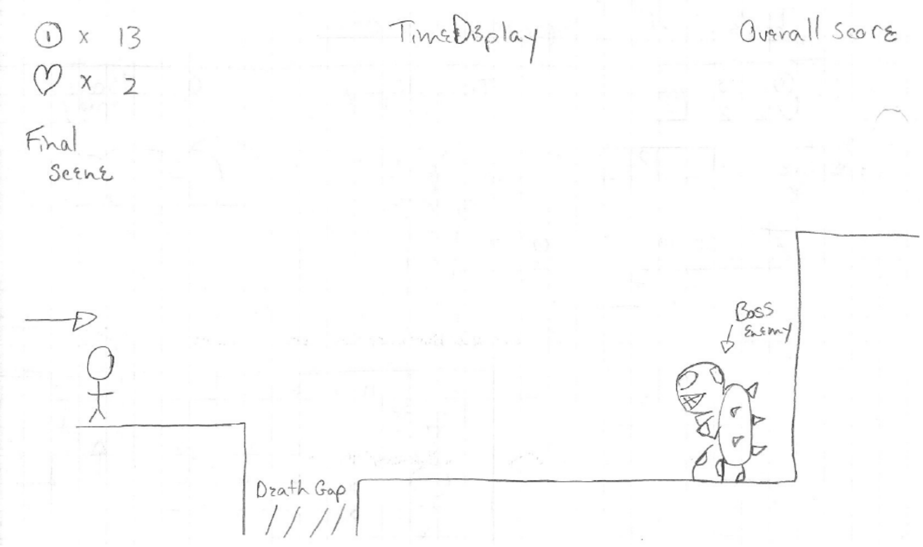


Figure 2: Rachel's World

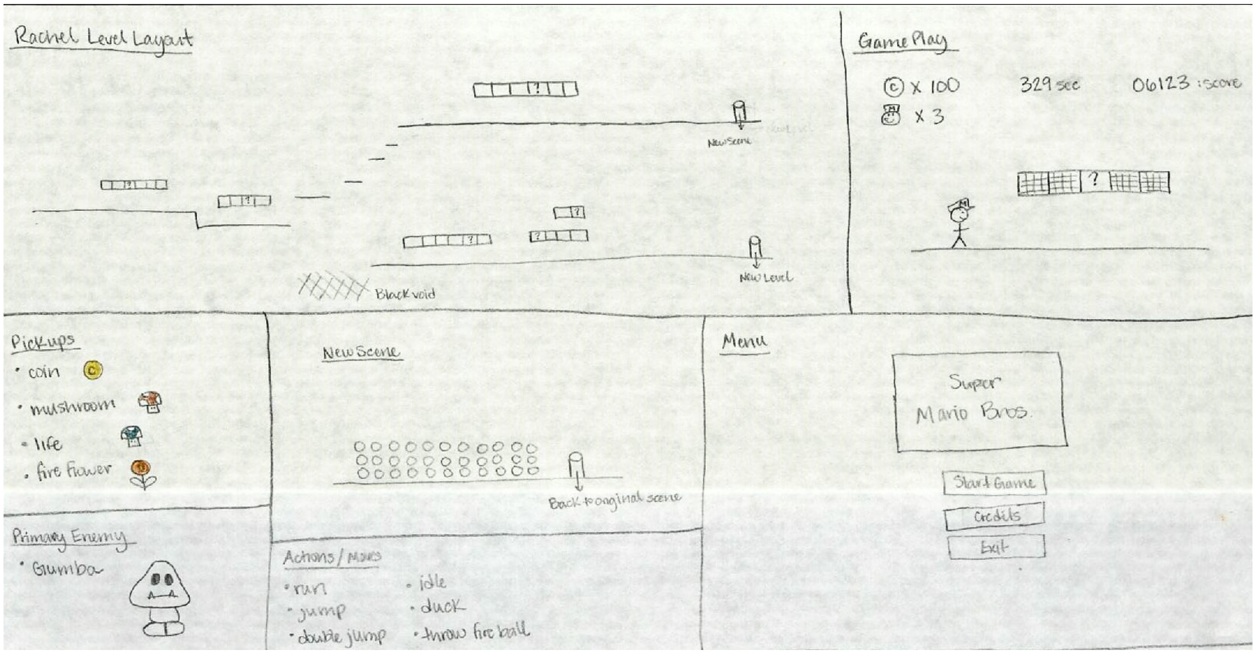
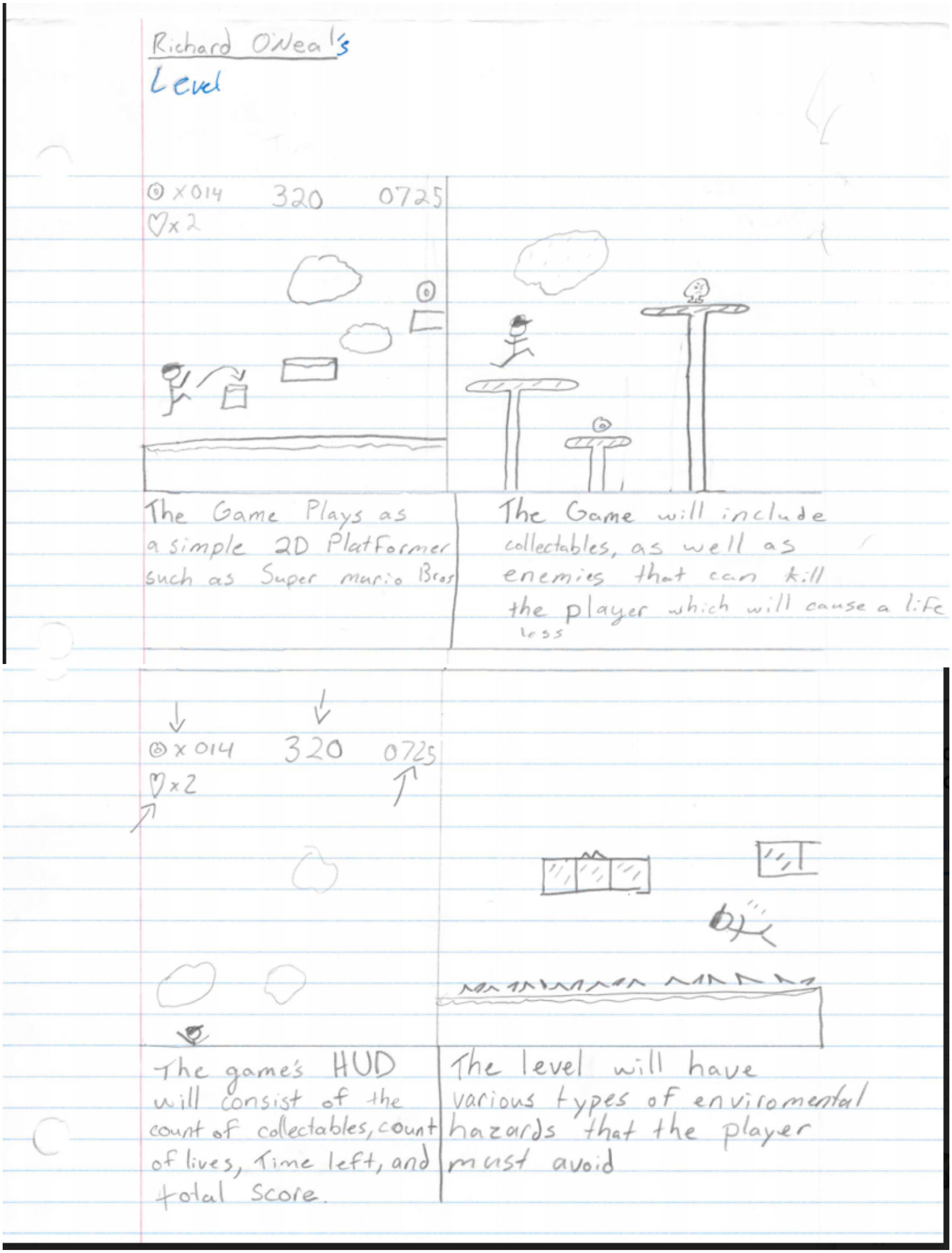


Figure 3: Richard’s World



Richard O'Neal's Level

The Game Plays as a simple 2D Platformer such as Super mario Bros

The Game will include collectables, as well as enemies that can kill the player which will cause a life less

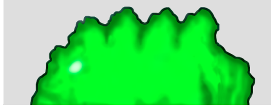
The game's HUD will consist of the count of collectables, count of lives, Time left, and total Score.

The level will have various types of enviromental hazards that the player must avoid

Sprite Animation

The following is a subset of the sprites used listed with respective functionality:

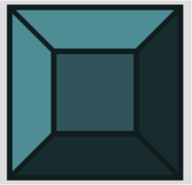
1. Bush: Environment



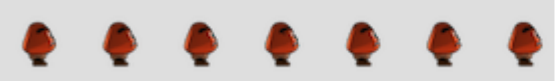
2. Underworld Background: Environment

3. Underworld Ground: Environment

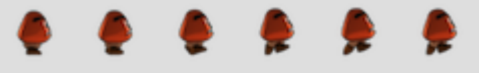
4. Underworld Brick: Environment



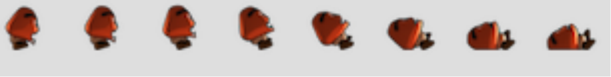
5. Gumba Walk Right: Enemy AI



6. Gumba Run Right: Enemy AI

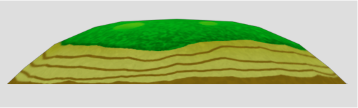


7. Gumba Die Right: Enemy AI



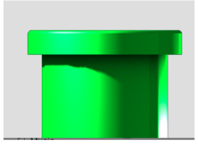
8. Castle: Environment

9. Mound: Environment



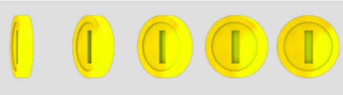
10. Mountain: Environment

11. Pipe: Teleportation



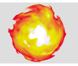
12. Red Mushroom and Green Mushroom: Player Properties, Increase Health, Increase Lives

13. Coin: Player Properties, Coin Count Increase

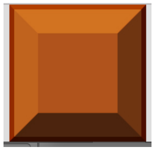


14. Fire Flower: Player Properties, Change Mario's look and give him fireballs

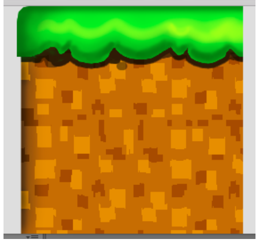
15. Fireball: Mario Attack Weapon



16. OverWorld Brick: Environment



17. OverWorld Ground: Environment



18. Question Block: Mario interaction, spawning of pickups, changing self state

19. Breakable Geometry: Allowing the blocks that Mario hits to break apart

20. Mario Run Right: Player Controls, movement

21. Mario Jump Right: Player Controls, movement

22. Mario Crouch Jump Right: Player Controls, movement

23. Mario Crouch Right: Player Controls, movement

Resources

Foundational Features

The project was based off a tutorial found on Walker Boys Studios website. The tutorial was intended to provide the student with advanced 2D design objectives through the creation of a 2D Mario remake using Unity. The completion of the tutorial provides the individual with the essential steps in creating 2D games. All of the sprites, sound effects, and music were provided by Walker Boys Studios.

Features Not Included in Tutorial (Team Additions to Features)

The following features were designed and implemented by team members without the help of the tutorial other than residual knowledge acquired during similar features.

1. Graphical User Interface
2. Victory Stage: The completion of each level
3. Screen Wipe: Covering the screen when player dies, game over, and game won
4. Pipe Teleportation Upwards
5. Underworld Scene: Mario is underground

Walker Boys Mario Tutorial link:
http://www.walkerboystudio.com/wbstudio/project-2d-mario/

Sounds and Audio

Sounds were provided by the Walker Boys Studio tutorial package. The sounds used by Walker Boys were cited in the project's assets. Here are the links that were used for collecting the audio and sounds.

Sound and Audio resource links:
http://www.themushroomkingdom.net/media/nsmb/wav
http://www.mariomayhem.com/downloads/sounds/index.php
http://www.hark.com/collections/bpmslgxkzj-super-mario-brothers