UCLA Mario

**Team Members**

Natalie Salem - 904018474

Tushar Shrimali - 804070047

Andrew Soucie - 804001803

**System Specifications for final run of project**

For best results:

Windows 7 using Chrome

- Need to run the command prompt to get the textures going:

"complete filepath to chrome.exe" --allow-file-access-to-files "complete filepath to \Code\cs174a\_term\_project.html"

**Game Mechanics**

**Press UP to jump, LEFT for moving left and RIGHT for moving right.**

UCLA Mario has a simple gameplay with the objective being crossing the UCLA campus from start position while at the same time **avoiding obstacles** spread throughout and **accumulating points by walking over collectibles**.

The obstacles might be in the form of USC trojan characters and/or Mario themed plants that cause out character to die and respawn. The player dies by falling on top of the either of the two enemy objects but is blocked by the Trojan/plant object if approaching from the side.

The level is covered with bricks that the user jumps over/ducks under. These bricks have been textured with the same patterns and are spread at random locations.

**Concepts/Ideas Implemented**

* Heavy use of collision detection with almost everything. The bear needs to land on objects and avoid contact with enemies
* Texturing for various objects rendered on screen (including transparency/blending for the bear and enemies)
* Music files play when the bear collides and/or dies
* Model View transformations and projection transformations for all movements and rendering on the screen (objects move left as the bear runs right at the center of the screen)
* Lighting for all non transparent objects

**Notes on Gameplay**

- When the player dies 3 times game over displays on the bottom, but we didn't set the page to refresh, so the collectibles won't reset.

- When the player reaches the end of the level to win, the bottom display updates, but nothing else happens, so you'll have to refresh the page to start again.

**Who Did What:**

**Natalie:**

* Setup brick, enemy, respawn points and collectible arrays. Applied textures for bricks/enemy/collectibles/floor/grass.
* Collision Detection (essentially using squares, since the game is played in 2D). Sets collectibles invisible when touched, and player to dead when top of enemy is touched.
* Player died animation. Bear flashes red twice and then respawns.
* Lighting for the non-2D objects, applied in the fragment shader.
* Multi-key press for running and jumping

**Tushar:**

* Setup main world and object locations (floor, background and camera positioning)
* Texturing and placement of all base objects (background images, floor and the bear)

**Andrew:**

* Sound effect (.wav) and background music (.mp3) collection
* Background music (only played once; could not figure out how to loop or restart upon character death within the time constraints)
* Interaction between collision detection module and sound effects
* *style.css*: #canvas-wrap, .scorefooter, .livesfooter, .youwinfooter
* Implementation of *style.css* in HTML code