***Moonbase - Technical Specifications***

Jun Yen Leung - junyenle@usc.edu

Ethan Barker - esbarker@usc.edu

Jake Leventhal - jglevent@usc.edu

Uri Rahimi - [urahimi@usc.edu](mailto:urahimi@usc.edu)

**Section 0: Misc Notes**

* Platform: Moonbase will run on both Mac OS X and Windows Operating Systems and will be designed on both Mac OS X Windows 10.
* Moonbase will use keyboard controls.

**Section 1: Title Screen, Login, Registration**

* Title Screen / Login / Registration (8 Hours)
  + The main title screen needs to have three buttons: login, register, and continue as guest.
    - If the login button is clicked, the user needs to be brought to a screen with a form that takes two text fields: username and password. Here, there must be a back button which returns the user to the previous screen, and a login button that submits the form. Clicking this login button should check the username and password inputs against the database. If the username and password are valid, the user will be logged in. If not, an error message should be displayed.
    - If the register button is clicked, the user needs to be brought to a screen that has a username field, password field, register button, and back button. The back button should send the user back to the previous screen.
    - If the “continue as guest” button is clicked, we will let the user proceed and to the main game as a “guest user” that can only play the 1-Player Game.
    - Once a user is logged in, there should be two additional options on the menu screen, 1-Player Game and 2-Player Game.

**Section 2: Agents**

**2a: Player Character (PC)**

PC Actions (8 Hours)

* + The PC should have the following abilities within the game world:
    - If the up button is pressed, the PC should jump. Holding down the button should result in a higher jump.
    - If the ‘left’ button is pressed, the PC should move to the left.
    - If the ‘right’ button is pressed, the PC should move to the right.

PC Art/Animations (8 Hours)

* PC ART HAS BEEN SIMPLIFIED TO A SIMPLE SHAPE.

**Section 3: Environment**

**3a: Player-acting environment**

Environment Design (8 Hours)

* + The level will be created by making a grid of tiles in a tilemap. Tiles need to be drawn 120x120 pixels in size.

**3b: Non-player-acting environment**

Environment Design (8 Hours)

* + Non-player-acting means environment features that are purely aesthetic.
  + There needs to be a painted background to add an element of world immersion to the game.

**3c: Player / Environment Bridging**

By this we mean the elements that help connect the player on the other side of the screen to the environment we have created.

Player Camera (8 Hours)

* + The Player Camera (PLAYCAM) determines which elements of the LEVEL are visible on the player’s screen.
  + The PLAYCAM should be centered on the PC and move with the PC.
  + The width of the PLAYCAM should be the width of the player’s screen, up to a maximum of 1920 pixels.
  + This will be done by updating the x and y positions of the PLAYCAM’s center every few frames and setting it equal to the PC’s x and y positions.
    - We do this every few frames and not every frame so that we can achieve a delayed camera following effect (for style purposes).

**Section 4: Additional features**

**4a: GAME OVER**

GAME OVER Event (2 Hours)

* + GAME OVER occurs when the player wins.
  + The player should be taken to a screen that says “Game Over” with an animation showing the player traveling back to Earth.

**Section 5: Database**

Database (8 Hours)

* + We will use a MySQL database hosted on one of the project member’s desktop computer.
  + The database (DB) will have one collection:
    - It will consist of all registered players and their login details. For instance,

{

username : “ttrojan”,

password : “fighton”

}

This collection should be indexed by username for quick querying when players try to log in.