Unity Project

Design Specification

Members: Anthony Jamora, Jonny Nabors, Thomas Burke

Draft 1

*CSC 4420*

*Unity Group*

*2015*

[1 Game Description 5](#_Toc413432131)

[1.1 Design Goals 5](#_Toc413432132)

[1.2 Influences and Sources 5](#_Toc413432133)

[1.3 Target Audience 5](#_Toc413432134)

[1.4 Target Customer 5](#_Toc413432135)

[2 Functional Specifications 5](#_Toc413432136)

[2.1 Core Game Play 5](#_Toc413432137)

[2.2 Game Flow 6](#_Toc413432138)

[2.3 Characters 6](#_Toc413432139)

[2.4 Game Play Elements 6](#_Toc413432140)

[2.5 Game Physics and Statistics 6](#_Toc413432141)

[2.6 Artificial Intelligence 7](#_Toc413432142)

[3 User Interface and Menus 7](#_Toc413432143)

[3.1 Main Game Menu 7](#_Toc413432144)

[3.2 GUI Objects 9](#_Toc413432145)

[4 Art and Video 9](#_Toc413432146)

[4.1 Overall Goals 9](#_Toc413432147)

[4.2 3D Art and Animations 10](#_Toc413432148)

[4.3 GUI 10](#_Toc413432149)

[4.4 Terrain 10](#_Toc413432150)

[4.5 Game Play Elements 10](#_Toc413432151)

[4.6 Special Effects 10](#_Toc413432152)

[4.7 Marketing and Packaging Art 10](#_Toc413432153)

[4.8 Assets Pipeline 10](#_Toc413432154)

[5 Sound and Music 11](#_Toc413432155)

[5.1 Overall Goals 11](#_Toc413432156)

[5.2 Audio Sources 11](#_Toc413432157)

[6 Story 13](#_Toc413432158)

[6.1 Player Characters 13](#_Toc413432159)

[6.2 Secondary Characters 13](#_Toc413432160)

[6.3 Enemy Characters 13](#_Toc413432161)

[6.4 Story Theme 13](#_Toc413432162)

[6.5 Visual Theme 14](#_Toc413432163)

[6.6 Story Outline 14](#_Toc413432164)

[7 Level Requirements 14](#_Toc413432165)

[7.1 Level Diagram 14](#_Toc413432166)

[7.2 Asset Revelation Schedule 14](#_Toc413432167)

[7.3 Level Design Seeds 14](#_Toc413432168)

[8 Technical Specifications 14](#_Toc413432169)

[8.1 Game Mechanics 14](#_Toc413432170)

[8.2 Game Engine 14](#_Toc413432171)

[8.3 Naming Convention 14](#_Toc413432172)

[8.4 Platform and OS 16](#_Toc413432173)

[8.5 External Code 16](#_Toc413432174)

[8.6 Code Objects 16](#_Toc413432175)

[8.7 Control Loop 16](#_Toc413432176)

[8.8 Game Objects Data 16](#_Toc413432177)

[8.9 Data Flow 16](#_Toc413432178)

[8.10 Artificial Intelligence 16](#_Toc413432179)

[9 Production Schedule 16](#_Toc413432180)

[9.1 Scope 16](#_Toc413432181)

[9.2 Scheduling 16](#_Toc413432182)

[9.3 Dependencies 16](#_Toc413432183)

[9.4 Cost Estimate 16](#_Toc413432184)

[10 Use Cases 16](#_Toc413432185)

[11 References 16](#_Toc413432186)

[12 Document Revision History 16](#_Toc413432187)

[13 Appendix 17](#_Toc413432188)

1. Game Description
   1. Design Goals

The goals this game aims to achieve for its target audience is to provide an interactive, immersive, and entertaining experience for the player to enjoy. This standalone PC game will be available within seconds whenever the customer wants to play it. The main focus is to draw the player in with an awe-inspiring world, stimulating combat, and a rich story. The game will provide for the user a glimpse into a fantasy world where civilization has been established among people who unknowingly live within a massive creature.

* 1. Influences and Sources

A team of interdisciplinary students attending the College of Creative Studies were the original creators of the games artwork and lore. They came up with an idea for an RPG where the main protagonist explores different imaginative environments to discover that he and his people live within this immense organism. They created multitudes of artwork, ranging from sketched concept art, to video and animations, and also 3D models. They created the models using tools such as Zbrush and Maya. These artwork assets were then handed to the programmers to be implemented within Unity.

* 1. Target Audience

Our target audience for the game are gamers who specifically play action-adventure roleplaying games. This audience makes up 13.9% of the entire U.S. gaming market which consists of 211 million people. This becomes 20 million plus customers within the United States alone. While the age of the average gamer lies at 31, the age of the average RPG players lies at 26. (1)

* 1. Target Customer

Our target customer is a male or female whose age fall within the range of 22-30 with the average age being 26. This customer takes many precautions before making the decision to purchase a new game. Being so integrated with the digital age, our target customer has a tendency to retrieve information on potential purchases through blogs and other online media outlets. Much of their decision is centered on reviews of the games by others who have played it. When our customer chooses to purchase a game, they expect three criteria to be met:

1. The ability to customize the character they interact with to some extent.
2. The freedom to explore the setting in which they are placed.
3. Last and most importantly, a strong storyline with characters they can empathize and connect to. This customer enjoys the option of choice and takes pride in the characters they create. The power to choose what their character wears, says, and does, places them in a position where they gain a sense of vested interest in the character. (1)
4. Functional Specifications

This section will go over our game mechanics.

* 1. Core Game Play

Explanation of what our user will experience during the game and how.

* 1. Game Flow

Explanation of how our game will flow from launch 🡪 exit.



* 1. Characters

**The Hero** – The main character who is controlled by the player, he is an emissary who was created by the massive creature so that, through you, it could communicate with the cities of people who live now within it.

**Friendly Old Man NPC** – Interacts with the main hero through a short dialogue, providing the player information about the island, how to fight enemies, and how to collect/use potions and essences.

**Enemy Insect NPC** – The main antagonist in the game, a simple minded but fierce insectoid. They are the main combatants in the game for the player to interact and fight with. They will have small patrol routes and react to the player walking within a certain proximity before they aggress onto the player.

* 1. Game Play Elements

**Health** – A simple health points counter which will determine whether the player has taken enough damage to reach the threshold for in-game death. This will be displayed as a sliding red bar on the UI, and the player will start with 200 health points. A health count greater than 0 will allow the player to maintain control over the hero. Conversely, once it reaches 0, the hero will play a death animation and the player will be directed towards a continue screen.

**Stamina** – Another player counter visually represented by a (green) bar on the UI. The hero will start with a maximum 150 stamina points, and will always recover 10 pts/sec (5 if actively blocking). If stamina reaches 0, then there will be a 2 second delay before recharging resumes. If the player’s stamina bar is depleted, their movement actions will be limited to the standard walk.

The player’s stamina count will be deprecated by each of the following player actions:

* Sprinting: 15 pts/sec
* Attacking: 30 pts/ea.
* Blocking: Equivalent to incoming attack
* Rolling: 30 pts/ea.
* Sidestepping: 15 pts/ea.

**Essence** – Small pick-ups which will automatically fly to the player then disappear upon collision when the player is within a short proximity. Each essence will increase the counter on the UI by 1, and after a pre-determined threshold is met, a special attack will be unlocked, and available for the player to perform. ¹

**Potion** – Collectable items found throughout the game world which can be consumed by the player to recover health and stamina on-demand. Potions will have a dedicated animation where the player takes a drink whenever the player decides to use one. Using a potion will restore 80 health points, and 70 stamina points. They will not be able to restore points over the pre-defined maximums.

¹ This functionality is currently a stretch-goal for the demo

**Treasure Chest** – An interactive model in-game which the player can approach and use an action key to open it, and collect 5 potions from within.

**Boulder** – An interactive game-world object which will be released after solving the puzzle. It can roll around the island, and damage enemies who come into contact with it.

**Puzzle** – There will be a puzzle intended to briefly vary the style of gameplay, requesting the player to step on three different raised tiles in the correct order. When a tile is stepped on in the right order, it will remain lowered. Once the correct order has been inputted, the boulder will be released. Upon stepping on a tile in the wrong order (whether it be the first or last), the tiles will all raise and reset to their original positions. The purpose of which is to instantly tell the player something went wrong due to that specific action, and to encourage them to try again.

* 1. Game Physics and Statistics

The in-game physics is controlled by Unity’s built-in physics engine. We first create objects (3D model, light source, etc.), then assign components to those objects to add basic functionality and behaviors to that object. Essentially, components are the “nuts and bolts”, or inner workings, which we manipulate to change our objects behavior. The rigidbody component, for example, adds a built-in functionality for an object to react to basic physics (e.g. mass, drag, gravity). These new properties can be adjusted during development, and also harnessed by scripts to be changed during run-time. Along with the rigidbody component, other core components which are built-in to the Unity engine are: colliders, raycasting, joints, force, and torque.

* 1. Artificial Intelligence

Explanation of how our AI will work for all three types of our enemies. Aggro range, deaggro range…

1. User Interface and Menus

Similar to how a vehicle has a dashboard that shows you the critical, need to know information about the status of your vehicle while driving, the game must provide the user with a simple and intuitive user interface and heads-up-display as well as an easy to navigate game menu.

* 1. Main Game Menu

Upon loading the game from the executable, the player will be presented with a screen containing the Capstone Team’s logo. The game will be loading for the duration of this on-screen graphic and will not move to the main menu until this has completed. After successful load of the game data, the player will then be taken to the game’s main menu screen.

Menu Items:

1. New Game
   1. By selecting this the player will launch a fresh instance of the game and will be brought immediately to the game world.
2. Options
   1. Change Resolution
      1. From this selection the player will be able to change the resolution from what is currently selected to something that better fits their computer monitor. The aspect ratio is limited to widescreen or the 16:9 aspect ratio.
   2. Adjust Brightness
      1. The player will be able to adjust how bright the game is from this option in case of low light or if they are sensitive to the bright lights in the game.
   3. Adjust Graphical Quality
      1. In the case that the player is unhappy with the game’s default video settings, they can select one from here that better fits their computer’s video output capabilities.
3. Control Scheme
   1. By selection this option the player will view an image of a keyboard depicting clearly what each key does in the game and how to control the player character from the game.
4. Credits
   1. From the main menu, the player can choose an option to see the list of names of all the people that have worked on the game.
5. Exit to Desktop
   1. In the case that the user does not want to start a game or change an option, they can choose this selection to return them to their desktop.

  
Main Menu Flow Diagram

Should also list each “page” in the game with a brief description.

Splash: Intro screen with CCS/Wayne logos

Main Menu: First interactive user screen with standard menu options

Then have diagrams/mockups of each “page”

Splash:

SPLASH LOGO

* 1. GUI Objects

Description and list of user interactive menu items

Description and list of HUD items

Attach Screenshot

Screenshot

1. Art and Video

Description of the role art and video play within the game

* 1. Overall Artistic Goals

The game will be produced in a “Low Poly” style. This translates to low polygons. These polygons determine how much detail will be sculpted into the 3d models for the game’s assets. The 3d models being referred to will be created using Zbrush and Maya (industry standard development software). Rather than raising the polygon count on these models, our team will focus on making our models interesting through the use of colors and textures. Lowering the amount of these polygons per model will decrease the overall time necessary to develop the game, which in turn cuts down on production costs.

To really make the models stand out, a major aspect of our production will focus on the lighting and coloring of the assets found within the game. This approach to production will take the game towards a stylized direction. This in turn will place more emphasis on the game’s storytelling and interaction. (1)

* 1. 3D Art and Animations

After the three-dimensional models have been textured, they will then be sent over to the animators for rigging and animation. Rigging is the process in which the models are given joints that enable animators to move the models for the task of animation. The rigging and animations will be created within the animation program known as Maya. (1) Some pictures of our raw 3D models and animations can go here as well.

  
3D Render of Player Character Concept

* 1. GUI

List of our GUI items and some images

* 1. Terrain

List of terrain items categorized by interactive, background(skyboxes), undecided.

  
Terrain Concept Art

* 1. Game Play Elements

List Description of different game play elements, health pots, stamina…

* 1. Special Effects

Description of special effects in game, ie when the boulder goes crashing into the ground or character. When the enemy hits the player and vice versa.

* 1. Marketing and Packaging Art

The primary marketing strategy for the game will be centered on the creation of a strong demo. The demo will be a presentational tool, solidifying the core features of the game with consumers.

Image of the spash art used at the start of the game along with an icon for our executable.

* 1. Assets Pipeline

Should create a layout of how our assets and hierarchy should be organized.

Hierarchy

Enviornment

Island

Navmesh

Characters

Player

Enemy

Bug1

UI

Also include a description of how our Assets/prefabs/hierarchy items should be titled.

1. Sound and Music

Music and sound effects for the game will be developed in parallel with the game art and assets. Sounds used within the game will all be custom made for the sole purpose of the game itself. Each sound and music clip’s intended use is for game enhancement and as such the player could still play the game without any audio if they so desired without compromising the quality of the gameplay.

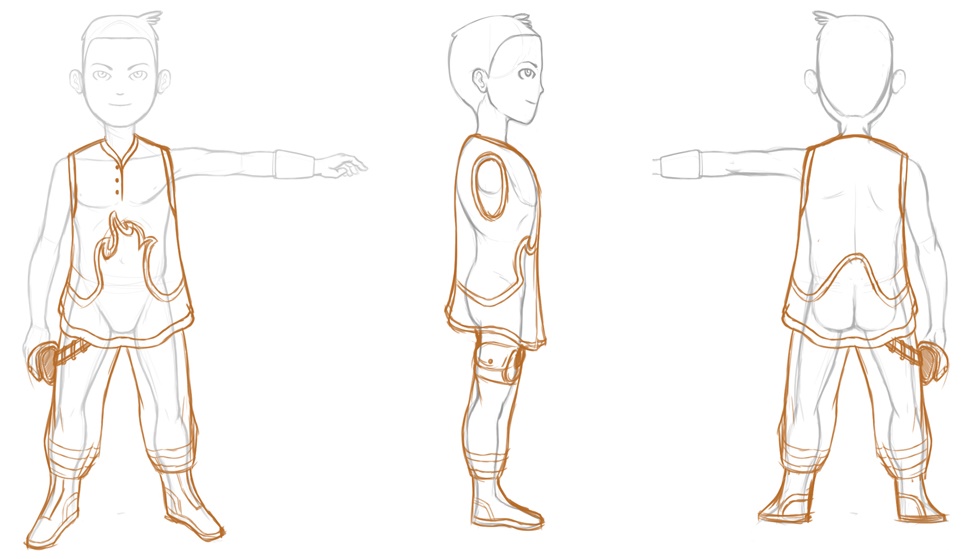
* 1. Overall Goals

The overall goal of the Sound and Music of the game is to add to the level of immersion the player should feel while playing the game. Similar to a good movie, the audio in the game should lend to the sense that the player is actually in a real, living place. Music should swell and change during moments of tension, weapons should make noises when colliding with enemies, and the player should know when they have selected a specific option in the main menu.

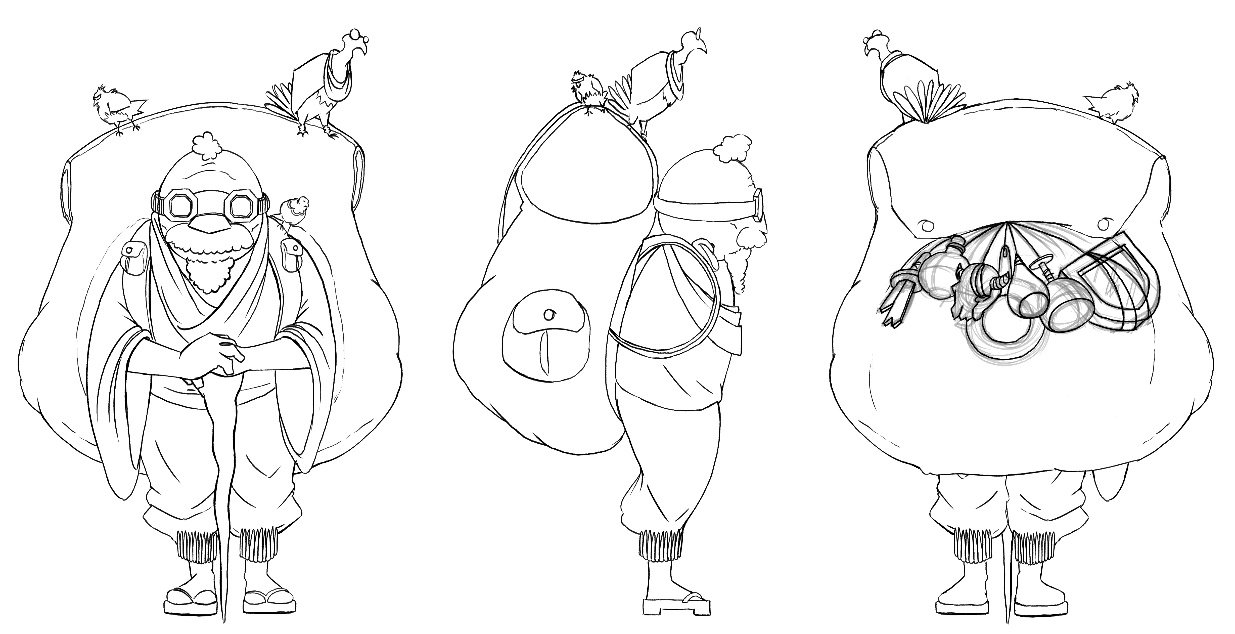
* 1. Audio Sources

The below list exists to give a brief overview of the various elements in the game that are related to sound or will trigger the change of music. All audio sources can be attached to a specific game object and may be controlled by triggered events from the game scripts.

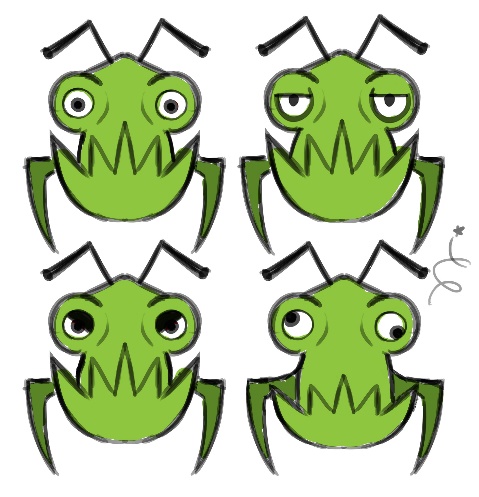
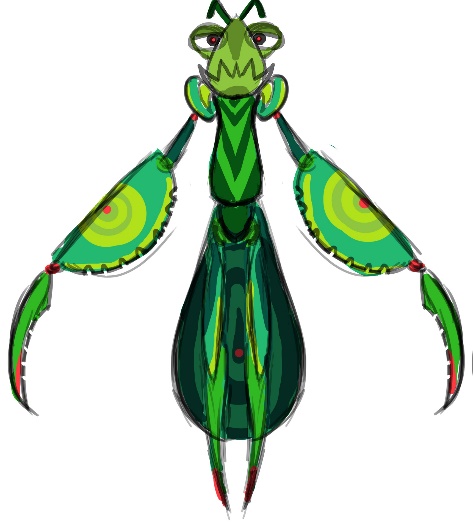
1. Music
   1. Main Menu Music – Specific audio to be played during the main menu of the game.
   2. Ambient Game Music – During the normal flow of the game, this will be the default music heard in the background. It will be an easy to listen to audio clip that loops continuously.
   3. Tension/Action Music – When the player comes into contact with an enemy, the music will transition from the ambient track to one with more urgency to it.
2. Sound
   1. Option Menu Move – While in any options menu, if the player changes the currently selected choice or moves their mouse over the option choice, an audio cue will play.
   2. Option Menu Selection – Similar to the above audio, this clip will be triggered when any option is actually selected from the menu by either mouse click or key press.
   3. Player footsteps – During regular movement, the player will have footstep noises that play during the movement animation. These footsteps can be specific to the terrain type, such as grass or sand. This audio will be changed based off the movement speed of the player character.
   4. Player Attack Action – While attacking, the player will make a noise similar to a grunt while swinging his sword.
   5. Player Attack Collision – If the player’s sword comes into combat with an enemy or an in-game object, an audio cue will be played to signify the type of collision made. There will be one generic sound played for the sword colliding with the terrain & assets, and one sound for when the player’s sword successfully makes contact with an enemy.
   6. Enemy Movement – The enemy character in the game will use the same footstep audio as the player character.
   7. Enemy Activation – When the player character has come close enough to an enemy to trigger combat, an audio cue will play originating from the enemy character. This may loop as a continuous noise as a base representation of the sound the enemy makes at all times such as a buzzing.
   8. Essence Collection – Upon the player picking up an essence token dropped by the enemy character, a sound similar to a chime will be played.
   9. Drink Potion – When the player presses the button to drink a potion, a drinking or swallowing sound will be played.
   10. Open Chest – The treasure chest in the game world will have a sound attached to it that will trigger when the chest is opened. It will be similar to a creaking door opening.
3. Dialogue
   1. Non-Player Character Speech – When the player comes into contact with the non-player character and interacts with him, various audio cues may be played. These will not be voiceovers, but more like vocal cues similar to grunts or other noises that represent what the non-player character may sound like.
4. Additional Audio
   1. Ocean Sounds – When the player is near the water, an audio clip will loop the sound of the ocean.
5. Story
   1. Player Characters

  
Player Character Concept Art

* 1. Secondary Characters

  
Non-Player Character Concept Art

* 1. Enemy Characters

  
Concept Art of Enemy Character

* 1. Story Theme

Countless years have passed since the first humans took refuge within the creature. Little is known of the outside world, few speak of it, and most ensure that there is no such place. This immense organism, a being, godlike in size, seems transient to the humans. It wishes to connect with those who have taken refuge within its body. It needs a way to personally interact with these humans. It needs you. You, who embodies the essence of this creature. You, who now stands at a crossroads between two worlds. You, who now has the power to change the world. A great struggle lies ahead, but fear not, for you will never walk alone. (1)

* 1. Visual Theme
  2. Story Outline

1. Level Requirements
   1. Level Diagram
   2. Asset Revelation Schedule
   3. Level Design Seeds
2. Technical Specifications
   1. Game Mechanics
   2. Game Engine
   3. Naming Convention

The Unity Project must remain consistent with a uniformly used and closely monitored naming convention for all game objects and assets.

|  |  |  |
| --- | --- | --- |
| Identifier Type | Rules for Naming | Examples |
| Player Object | All assets in the game that refer to the player character object must be identified as player\_PlayerDetail, where PlayerDetail explains the purpose of the particular player object. There is typically only one player object per scene. If there are other copies of the player character in one scene, they are to refer to the player object's current location in the game world. | player\_MainPlayer, player\_UnderwaterPlayer |
| Physical Environmental Assets | Objects and assets in the game that are a part of the active game world, meaning that the player can collide with or come into contact and interact with these objects, will be identified by env\_ObjectName. In this case, ObjectName is a succint description of the type of object this asset represents. | env\_Island, env\_RollingBoulder, env\_Water |
| Enemy Characters | Enemies that have been added into the prefab folder will be identified as enemy\_EnemyType\_Number where EnemyType refers to the type of Enemy it represents and Number is identifying which particular enemy this is in the game world. Since there may be multiple copies of a particular enemy type, this will alleviate confusion with multiple enemies being placed on the island. | enemy\_EasyEnemy\_1, enemy\_Boss\_2, enemy\_MediumEnemy\_7 |
| Lighting Elements | Elements and objects that give light to the game world will be named light\_LightType, where LightType refers to the type or light or purpose that the light has in the game world. There are many different light sources in the game that serve various purposes based on time of day, so there will be several different lights placed in the playable game world. | light\_Sun, light\_TreasureChest, light\_BossSpotlight |
| Camera Objects | Instances of the camera in game should be named as camera\_CameraType where CameraType refers to whate the particular camera object does. These camera objects will typically be children of a player object. | camera\_ThirdPerson, camera\_BossFocus, camera\_FirstPerson |
| Heads Up Display Elements | The static, on-screen heads-up-display elements as well as any pop-up text boxes or messages shall be named as hud\_HUDType where HUDType is a brief description of what this particular heads-up-display element achieves in game. | hud\_HealthBar, hud\_StaminaBar, hud\_EssenceCounter |

* 1. Platform and OS
  2. External Code
  3. Code Objects
  4. Control Loop
  5. Game Scripts

Each player and non-player object will need a script in order for any action to take place. Along with any objects that are interactive in anyway such as, Item Container, Potion, Weapons, Ladder, and even particles.

|  |  |  |  |
| --- | --- | --- | --- |
| Game Object | Attached Scripts | Description | Public variables |
| Player | Potion | Controls the animations and the health bar when a potion is used. | -potion animation  -health gain |
| Container | Controls the animation and item received when a container is opened. | -item inside  -open animation |
| Attack | Controls the player model, particles, stamina bar, and damage dealt when the player attacks. | -attack animation  -speed  -damage  -range |
| Block | Controls the player model, particles, stamina bar, and damage blocked when the player blocks. | -stamina value  -block amount  -block animation |
| Health | Controls the health bar when player is dealt damage an enemy or the environment. | -Total health  -health slider  -player death |
| Movement | Controls the player model and corresponding animations. | -Speed  -turn speed  -mouse turn speed |
| Climb | Controls the player model and animation when climbing objects. | -Speed  -climb animation |
| Interact | Controls the player model and animation when interacting with game objects. | -range  -interaction animations |
| Stamina | Controls the stamina bar when player is using any stamina reducing abilities. | -total stamina  -stamina slider  -exhaustion animation |
| Collect | Controls the essence model and animation when player “collects” essences. | -Speed |
| Enemy | Movement | Controls the enemy model and corresponding animations. | -Speed |
| Aggro | Controls the ‘aggro’ range and ‘deaggro’ range of an enemy. | -Aggro range  -De-aggro range |
| Attack | Controls the enemy model, particles, and damage dealt when the enemy attacks. | -attack animation  -speed  -damage  -range |
| Health | Controls the health of the enemy when damage is dealt by the player or environment. | -total health  -death clip  -essence value |
| Menu | In-Game Menu | Controls the in-game menu and all submenus. | -return  -main  -options  -controls  -exit |
| Main Menu | Controls the main menu and all submenus. | -start  -options  -controls  -credits  -exit |
| HUD | Quick Access | Controls what items are displayed in the quick access menu |  |
| Game Over Manager | Controls the ‘Game Over’ screen as the player dies. | -player health  -restart timer |
| Characters | Enemy Manager | Controls spawn points and spawn rates of enemies. | -enemies  -spawn locations  -spawn speed |
| NPC Manager | Controls the friendly NPC. | -chat bubble time  -chat bubble |
| Environment | Day-Night Manager | Controls the rotating light and skyboxes. | -Skyboxes  -rotation speed  -day length |
| Music Manager | Controls what music to play depending on what is occurring in the game. | -Music tracks  -volume |
| Camera | Close Fade | Eliminate character meshes when camera is too close | -Character Meshes  -fade distance  -hide distance |
| Camera Follow | Will follow the character appropriately | -ground check  -rotation speed  -distance update speed  -max angle  -max distance |

* 1. Game Objects Data

Every object in the game is considered to be a GameObject in Unity. GameObjects need special properties before they become characters, environments, or special effects. In short, GameObjects are containers that can hold different pieces called components. (2) All game objects have a Name, Tag, Layer, and Transform Component. The name is simply what we call the GameObject. A tag is a word that is linked to one or more GameObjects to make them easier to sort and find as well as being able to group GameObjects as a certain type. A layer is used by the camera to render specific parts of the GameObject and can also be used in testing for collision. The transform is the physical location of the GameObject in the game world as well as its rotation and scale on the X, Y, and Z axes. Lastly, a GameObject can have any number of attached components, such as an Audio Listener, a Physics RigidBody, a Collider, and a Camera.

  
The Sun GameObject, Component, and Script

When a component is created, there may be one or more scripts attached to it. A script is a specific type of Component that the user is creating and giving functionality to. Once a script is attached to a component it will begin working when the game runs. See below a visualization of the GameObject – Component – Script architecture and relationship.

Each GameObject in the game will have these Components and Scripts and can be tweaked and changed during the game’s development.

  
Visualization of the GameObject and its Components

* 1. Data Flow
  2. Artificial Intelligence
  3. Controller Map

|  |  |  |  |
| --- | --- | --- | --- |
| Class | Action | Key | Description |
| Movement | Forward up | W | Player runs north with forward mechanic |
|  | Forward Left | A | Player runs west with forward mechanic |
|  | Forward Right | D | Player runs east with forward mechanic |
|  | Forward Back | S | Player runs south with forward mechanic |
|  | Turn left | \*Mouse | Player turns left |
|  | Turn right | \*Mouse | Player turns right |
|  | Sidestep left | Q | Player dashes to the left (\*Stamina) |
|  | Sidestep right | E | Player dashes to the right (\*Stamina) |
|  | Jump | Spacebar | Player jumps (\*Stamina) |
|  | Roll | R | Player rolls/tumbles (\*Stamina) |
|  | Sprint | Shift | Player movement is increased (\*Stamina) |
| Combat | Attack | Left Click | Player swings sword (\*Stamina) |
|  | Block | Right Click | Player raises shield (\*Stamina) |
| Interaction | Use item | C | Player uses item |
|  | Interact | F | Player interacts with environment or other characters |
| Menu | Pause game | Escape | Pauses game and brings up menu options |

1. Production Schedule
   1. Scope
   2. Scheduling
   3. Dependencies
   4. Cost Estimate
2. Use Cases
3. References

|  |  |  |  |
| --- | --- | --- | --- |
| Doc Number |  | Doc Version | Doc Name & Location |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Author | Changes |
|  |  |  |  |
|  |  |  |  |

1. Appendix

Material including referenced documentation the web or elsewhere, as well as alternative designs or items/ideas for future improvements.