3D Game Development Report

Lee Murray

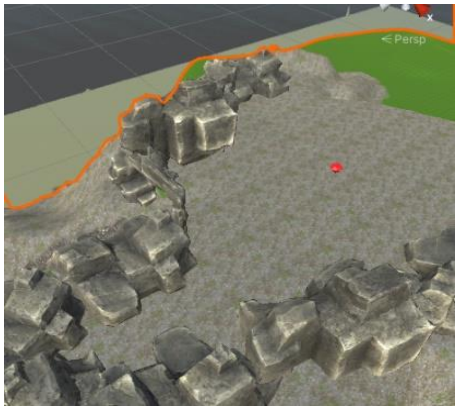
Lee Murray
[Email address]

Introduction

The following game was developed in unity using techniques learned from tutorials and lectures in the model 3D Game Development. the game is a simplified version of turn based combat rpgs taking inspiration from both jrpg and western rpgs both retro and modern. The overworld and combat camera perspective was inspired by the dungeon crawling scene in early ultima games. The initiation of combat however is inspired by games like Pokémon and shin Megami Tensei where combat is initialised after the player spends a varying amount of time in a specified area. the combat is a simple take on turn based combat without the party system. This document describes in detail the techniques used in the creation of the 3d environment and also details on the gameplay process.

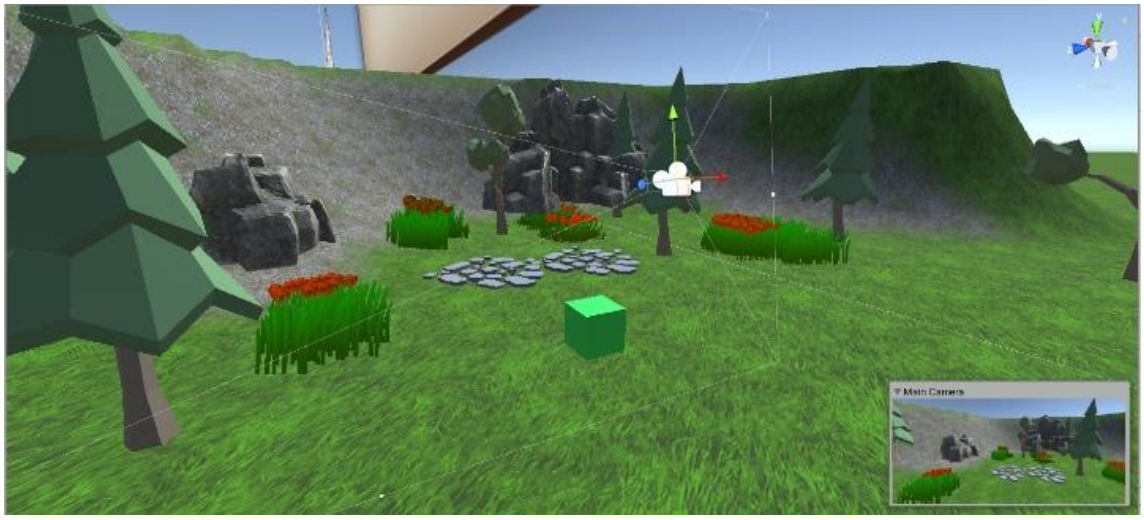
Overworld Scene

The games scenes consist of two major scenes. These are the overworld and battle scene. the battle scene is relatively empty consisting of just a floor, background, camera, player object and user interface. The overworld however is more complex and contains many models and a system of sounds at specified locations such as the use of ambient noise in the village area. the battle scene is designed to show the enemies to give the user a visual representation of gameplay. This is done through the use of collision detection and an integration of the gameplay system through the use of the ui component. This will be discussed in depth further in this document. The overworld consists of several areas these being the village area, maze, castle, mountain, grasslands, and graveyard. The level is designed in a way that funnels the user down a specific route. Exploration is allowed but the user will only get so far before having to backtrack and try another route. The overworld uses walls and gates as well as a combination of strategically placed collision boxes in order to stop the player from progressing in directions until they have filled the requirements to access other areas. The route planned out is as follows; maze, castle, front grasslands, graveyard, mountain, back grasslands village.



Battle scene

The battle scene is a simplistic scene that is not interacted with by the player physically but instead acts as a background. As a result colliders were not important in this scene and the camera would not be moved from its initial location. As a result I was able to construct the scene easily using techniques and models used in the creation of the overworld. The scene is designed in a way that allows for the instantiation of up to four enemies while remaining in frame the green cube represents the player and although the use of a empty game object could also have been used I found that using the cube allowed for me to visualise the battle scene more accurately during the development process, it also made selecting the object easier while applying different parameters to the main player script that holds most of the battle logic and menu processing as well as the players information



Village audio

Instead of having audio for each individual npc in a scene that would play in the background a single audio file was attached to the object containing the village. By doing so I was able to set up the audio component to play audio in a certain range, this range being the range of the village. The benefit of doing this is that it sounds clearer than having an audio component on ever npc and stops audios from conflicting with the dialogue system.

Dialogue Audio

Dialogue audio was recorded by myself using a combination of audacity and voice changing software in order to give some variation in the sound of the voices used. Dialogue is given to game Objects with the tag of “NPC” activation of these.

Battle and ambient music

Both the battle music, ambient music and main menu music use royalty free music. Both the battle and menu music were created by Alexander Nakarada. More music by the artist as well as the music used can be found at www.serpentsoundstudios.com. All music is attached to a mixer. The mixer is attached to the menu script that allows the volume to be adjusted by the menu system.

Particle system

Particle systems are used for a variety of different purposes. One such purpose is the use of particle systems from an aesthetic and stylistic choice in the main menu. Another use is the visualisation of status effects that can be inflicted upon the enemies. In the latter case each enemy prefab contains three objects that are disabled upon there initialisation into the scene. These objects are set to active depending on the status effect inflicted.

Movement and collision

Collision is handled using a combination of different primitive colliders and the use of mesh colliders where primitive ones cannot. One of the main uses of collision was in order to stop the player from colliding through objects such as npc's structures and the overall environment.

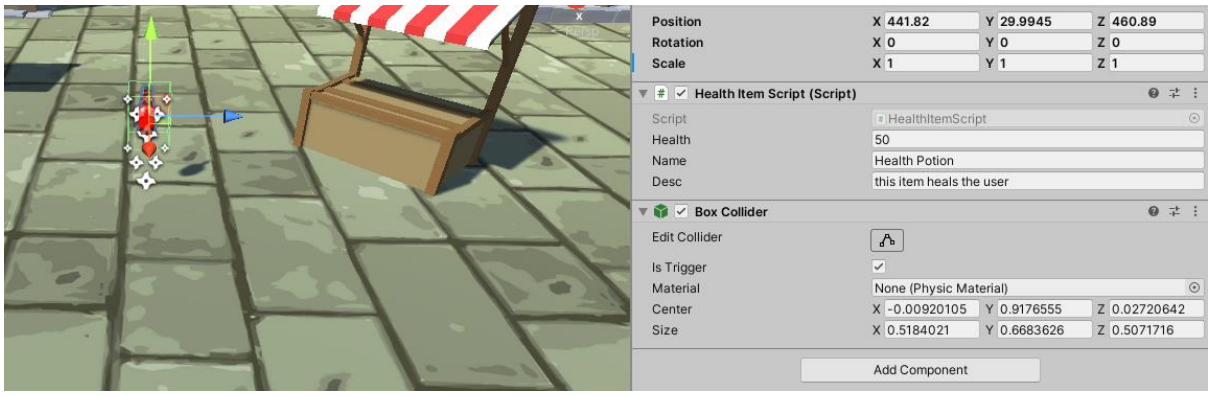
Player movement

Movement is handled by the fps controller prefab. This prefab was taken from the tutorials for this module. Movement is controlled using wasd keys. These keys are generally used for movement in pc games and as a result are used because the player will most likely know them through muscle memory. The camera is controlled using the mouse movement. In order to open the menu the tab button was used, the escape key would have been used however this conflicted with the controller.

Item collision

Items use a simple system of collision boxes that when on trigger add a custom class item to the inventory of the player. Using a class instead of holding the game Object itself means that the behaviour of the item is easily defined.

Health items and mana items



Enemy combat spawning

The gameplay of the game takes inspiration from turn based rpg combat systems such as shin Megami Tensei, ultima and final fantasy to name a few. Combat is initiated through the use of a box collider that initiates a new combat scene setting a timer. The box collider also holds a list of Game

Object prefabs. The list of enemy prefabs is passed to a script that holds static variables. On the load of the combat scene the list is passed to a factory class that initialises the enemies to the combat field. The benefit of using this system is that it allows me to dynamically decide where enemies are spawned and what enemies that are spawned through the use of prefabs.

Gameplay

Overworld Gameplay

Overworld Gameplay can be defined by three possible actions the user can take during there exploration of the world. Action one is the collection of items in the world.

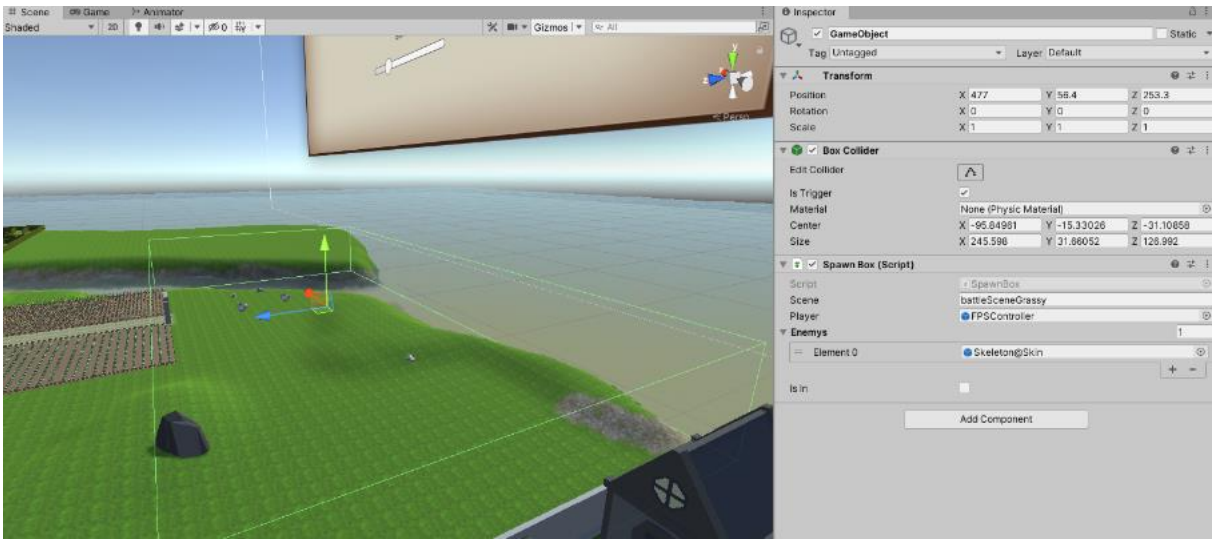
The collection of these items uses a simple collision script similar to the one used in the lecture and tutorial materials. The difference is the way the information is processed. When the user collides with a collectable object a Object from the Item class is created. There are two types of items that can be created. Basic Items using the base class and a consumable item using the consumable class that extends the item class in order to give the item some kind of behaviour while in the inventory.

Interaction with npc's

Interacting with npcs was done through the use of ray cast. The ray cast script was attached to the player and upon colliding with a game Object with the tag of npc would tell the player to click the g button in order to interact with the npc. Upon this the npc would change the animation through the use of a animator controller containing a Boolean parameter of Idle. The npc would then switch animations accordingly and then loop through a list of dialogue.

Explore the environment

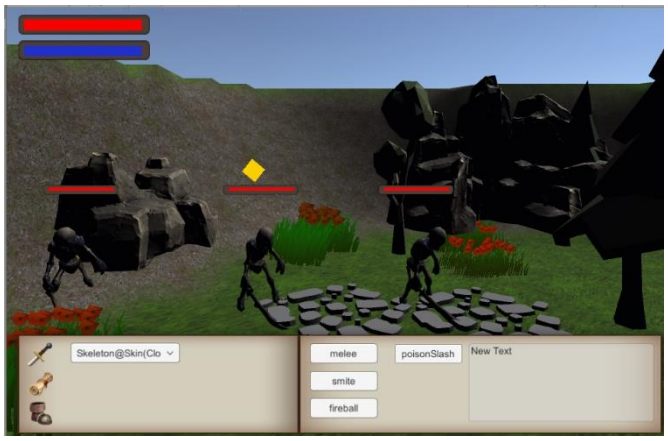
The environment is divided into sections, these sections are defined by game objects containing box colliders that cover the area, when entering these areas scripts are triggered.



Funnelling the player

The games layout is designed in a way that both allows exploration but tells the user where to go through the use of player placement and the placement of npcs. For example, when the game starts the first npc they see is a waving man. Placing this man in front of the player and making him wave tells the player that they should approach them. After that the dialogue tells the user what they should be doing. On top of these locations and routes are locked of until the player obtains key items.

Combat



the combat system is a simplistic implementation of a turn-based combat system inspired by retro and modern turn based rpg's. the perspective was inspired by the ultima dungeon crawling perspective while the combat is inspired by early turn based rpgs such as final fantasy and shin megami tensie. The combat phase has four possible paths, path one is use an item and then attack which will automatically end the users turn. Path two is the use of just an attack. Path three is the use of an item and then a pass turn. And path four is for the user to pass turn with out doing anything. The enemies then attack in sequence. In order to prevent the user from exploiting the menu and in order to limit the number of attacks per turn the menus are hidden during the enemy turn. When the enemies have all completed there turn the menu is displayed again and it is once again the players turn. This continues until the player defeats all enemies, is defeated themselves or retreats from battle. On the completion of combat the player is spawned back into the overworld in their last location.

Win condition

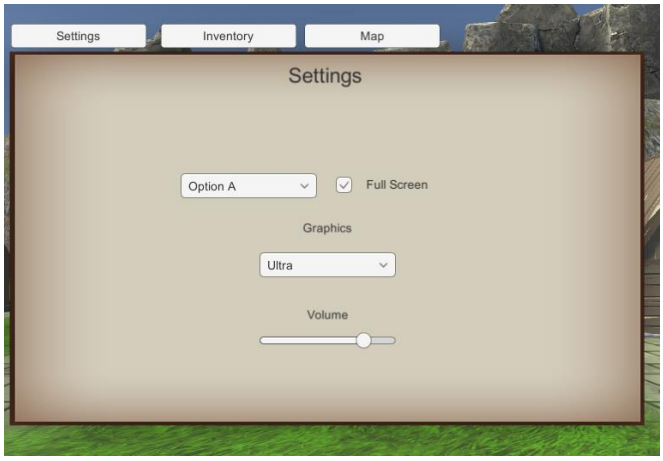
The win condition of the game is to get to the crystal on the top of the mountain and bring it back to the village. The way in which this condition is completed is by collecting the crystal and adding it to the players inventory. After doing this the player must then make there way back to the village. Once the user enters the village with the crystal in inventory the win screen will display, and the user will be booted back to the main menu scene allowing them to either exit from the menu or start again from the beginning.

Inventory

The inventory holds several things, such as mana/health potions and key items. During combat the inventory gets filtered through and two lists are created one for mana potions and one for health potions. These are added to a drop box for each allowing the user to use the items. Key items are kept in inventory and are only ever used to check conditions such as whether the user has a key to a door or the crystal that is the win condition.

Menus

Overworld Menu

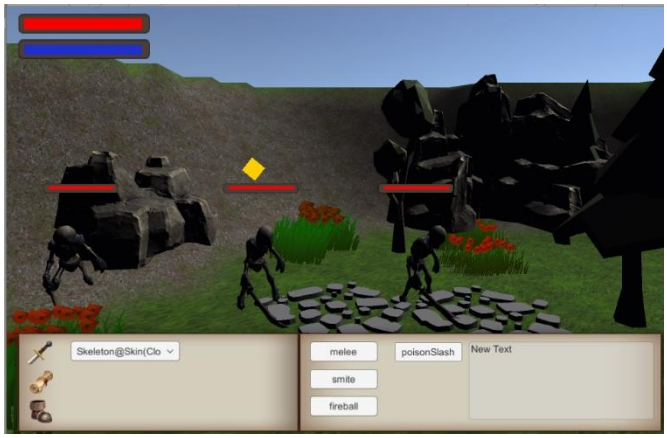


When the player is in the overworld they have access to three sub menus in the pause menu. These menus are , settings and map. The settings menu consists of several options that the user can modify, graphics gives the player a choice between ultra-very low. The volume slider controls the volume of the game full screen toggles the full screen of the game on and off and a resolution dropdown allows the user to select the resolution they wish to display the game at.

Map Menu

The map menu uses a camera and a raw image in order to display what that camera is seeing to the user. In order to display to the user there current location a pane was attached to the fpscontroller prefab and shadow was turned off in the render settings preventing the object from casting down a shadow. The mesh pane was placed atop the player. Since the pane is a child of the fpscontroller object its location is moved in relation to that of the mouse. The effect that this has is that the pane will never be infront of the main character and so the player will never see it. A material with a vibrant color was then picked so that it can be easily spotted when rendered on the map.

Gameplay Menu



Since the gameplay in this game is turn based the use of menus is the players primary form of interaction. The menus for the combat section are divided into two sections. On the left side is the primary menu that holds four options. These options are attacks, items run and the enemy selector. On the right side the menu system changes depending on what button from the left menu has been selected. If the attack button is clicked, then the attack menu is displayed. The attack menu displays up to 6 attacks that the user can have potentially. For this version of the game the user has

4 attacks assigned to themselves. Another feature added to the combat menu is the use of hover over. When the user hovers over an attack button the information in the text box is set to the information of the attack. A similar system is used with health items. Since there are only two usable items the complexity of the item's menu can be reduced by having two buttons for the items and an amount value next to each of them. Once again hovering over the item in the menu displays the information of that item in the text box.

Main menu

The main menu is a simple menu using a combination of a simple scene and a particle effect in order to make the menu more interesting. The scene that is the backdrop of the menu uses models and textures that are representative of the game. The dragon in the background for example is the same model used in the main game. An animator component with a loop is attached to this model. The main menu holds three options these being the start game option the exit option and the settings option. The settings option is the same as the one used in the main games over world menu allowing the user to change the resolution, the graphics setting, toggle full screen and adjust the volume of the game.