Culminating Project X

James McDonagh

**Overview :** 3D Roguelike Dungeon Fighting Game

*Roguelike*: a subgenre of role-playing video games, characterized by a dungeon crawl through procedurally generated game levels, turn-based gameplay, tile-based graphics, and permanent death of the player-character.

**Game Description :** A Dungeon Crawler that implements simple combats mechanisms (shooting, melee, etc.) to defeat enemies. The game will progressively become more difficult (Stronger Enemies with more health, more attack power, different attack patterns, etc.) with each new “Depth” (explained later). Other additions-such as stats, relics, sprites, “sulthers”-will be collected/gained as you continue through the game. However, upon death Ultimately the goal is to defeat the final boss, reclaim the “Soul Breaker Sword” and clear the dungeon.

**Expectations :**

|  |  |  |
| --- | --- | --- |
| Concept | Implementation | Explanation |
| ===================== | ===================== | ===================== |
| Unity Environment | Unity Game Engine | Use Unity to make game |
| Variables | Ints, Floats, Doubles  Bools  Transforms  Game Objects  “For” loops | Basic use (storing values that will be used/altered throughout the game) and use of the “.find” command to assign values without using public every time. |
| If Statements | Switches  Locks  Buffers | Used to turn things on / off (using if (true) or if (!true) and break;). Used to lock certain parts of the program until the parameter of the if statement is true (using update and a “buffer” so that the command isn’t activated more than once). |
| Methods | Execute multiple commands at the same time | Generally, just using them to activate large blocks of code without typing them multiple times. |
| Loops | Enemy Spawning  Obstacle Spawning  Item Spawning | Checking for positions, spawning items, continuously running things not in update. |
| Arrays | Prefab Storage  Spawned Object Storage | Store Prefabs that are made, Store multiple game objects and their positions. |
| Classes | Accessing Reference Variables  Accessing Methods  Instantiation | Used on the occasion for UI and raycast detection. |
| Instantiate | Spawning Enemies  Spawning boosts  Spawning Objects | Spawning anything that is not already in a given scene. |
| Prefabs | Premade Objects | Used in instantiation. |
| Tags | Collisions  Assignment | Detect different kinds of enemies, used with “.find” for assigning values. |
| Controller | Character Controller | Moving the character, using physics and “.Move” for jump as well as strafe. |
| Collisions | Damage  Item Pickup | Checking for hits, using “OnTriggerStay” since most objects are not just sessile objects. |

Development Plan :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DP | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 |
| Monday | Basic UI coding | Learning  Blender  Basics | Verifying / Unity Transfer | Coding | Coding | Final Touches |
| Tuesday | Basic UI coding | Learning  Blender  Basics | Setup Model in Unity | Coding | Coding | Test Runs |
| Wednesday | Advanced UI coding | Making Character Model | Using Premade Models in Unity | Coding | Coding | Revisions |
| Thursday | Advanced UI coding | Making Other Models | Using Premade Models in Unity | Coding | Coding | Final Build |
| Friday | Multiple Scene manipulation | Rigging | Controller / Camera Setup | Coding | Coding | Hand In |
| Saturday | Model Design | Animations | Unity Animations Code | Coding | Coding |  |
| Sunday | Model Design | Blend shapes | Environment | Coding | Coding |  |

Coding will be a creative process that comes as the game is developed. Therefore, I have no idea as to what I will be working on for now. Will be completed as time goes on.

“Actual Game” ShoGunDow.

**Overview :** A first Person shooter survival game.

Game Description : A survival shooter where a player must traverse through a room that has randomly spawned “pillars” that act as barriers and steps. The main goal of the game is to win by completing one of the following, surviving for 1h or getting a score of 10000 points. This game has the unique limiter of only being able to rotate horizontally forcing the player to jump / climb to damage different types of enemies.

Player Mechanics : The Player has a set health of 100, speed of 10, and jump height that is “relatively” low. The player must use the WASD keys to move and Spacebar to Jump to traverse the terrain. The Player is equipped with two different guns that will be explain later.

Enemy Types :

* Chaser (always 4) : The chaser is a fast moving enemy that tracks the player at a high speed forcing the player to run or die, although limited in damage and health it has the fastest hit rate making it very dangerous if not dealt with. It deals 1 damage when colliding with player, is faster than the player, and has a health of 2.
* Soldier (always 4) : The soldier is a slower yet still fast enemy that shares many things in common with the chaser. However, these enemy deal 3 times the damage to player at a slightly reduced hit rate. It has a health of 9 and is the same speed as the player.
* Titan (always 2) : The titan is slow yet is the most dangerous of all the enemy types. It deals a whopping 40 damage if it collides with the player. Not only that it fires 3 types of shots based on the players position (Ground = 10, Mid = 20, Top = 30). These bullets die on impact, after reaching target or after their lifespan. To kill a titan, you must hit it in the head or weak spot (eye). To do this you must jump onto a shattered pillar. This may seem easy as Chasers and Soldiers cannot hit you while you are on these Pillars but the constant pressure of being hit by these bullets makes it much more stressful. Killing titans also grants the player a boost spawn which will be further explained.

Side note : All enemies have a 1/50 chance to drop a health pack on death that will fully restore a players health.

Guns : Guns are the only way to kill incoming enemies.

* Left Gun : The light pistol, is a fast shot rate gun that allows the player to shoot a lower damaging bullet a long range. It has a clip size of 7 and a shot buffer of 0.25 sec. As for damage; chasers take max 2 shots (2 to head, 1 to eye), soldiers take max 9 shots (9 to body, 3 to head / face (takes extra damage)), and titans take max 10 shots (10 to head, 5 to eye).
* Right Gun : The hand cannon, is a slower shot rate gun that allows player to nearly one shot enemies at a close range. It has a clip size of 4 and a shot buffer of 0.5 sec. As for damage: Chasers take max 1 shot (1 to head, 1 to eye), soldiers take max 3 shots (3 to body, 1 to head / face), and titans take max 2 shots (2 to head, 5 to eye).

Side notes : The only way to reload guns is to pickup an ammo crate. You can hold down the mouse button to auto fire after buffer.

Ammo (always 3) : Spawns randomly throughout the map and reloads gun clips.

Healthpacks (Only 1) : Spawns randomly throughout the map and restores player to full health.

Boosts (Only 1 of each) : Spawns randomly throughout the map and gives player a powerful boost.

* Invinciboost : Makes player invincible for 10 seconds on impact.
* Burstboost : Unlocks guns fire rate and gives player unlimited ammo for 10 seconds (empties clip when done).
* Shield Boost : Gives player a shield that increases health by 100 (taking damage is reduced through shield before player health).

Side note : for boosts of limited duration a timer will appear letting you know how long you have left as well as a change in armband color. These boosts are only activatable when the other is not (not including shield).

UI : Hitting escape in game will pull up a game menu allowing for quitting by clicking the F6 button on screen and sensitivity changing.

Daily Record :

|  |  |
| --- | --- |
| Daily Log | Week 1 |
| Monday | Learnt More about Unity’s in application UI interface (A butt tons…). |
| Tuesday | Properly Coded a working button as well as slider that were able to call functions from other scripts (using classes). |
| Wednesday | Looked into making a Menu and Pause Menu Scene that will be able to work interchangeably. |
| Thursday | Looked into making the menus incorporate interactable UI that can call functions from other scripts (using classes). |
| Friday | Using Scene Management, Switched Between Multiple Scenes and Scene pausing. |
| Saturday | Drew out VERY rough character designs. |
| Sunday | Drew out VERY rough other designs (enemies, weapons, relics, etc.). |

|  |  |
| --- | --- |
| Daily Log | Week 2 |
| Monday | Watched YouTube videos on Model Design with Blender / other. |
| Tuesday | Watched YouTube videos on Model Design with Blender / other. |
| Wednesday | Began Making Character Design. |
| Thursday | Continued to work on character design. |
| Friday | Realized how hard character modelling is and decided to make simple enemies. |
| Saturday | Weapon models, much easier. |
| Sunday | Rigging had to remake core components of model to properly function. |

|  |  |
| --- | --- |
| Daily Log | Week 3 |
| Monday | Rigging continued. |
| Tuesday | Started Animations (Character Movement, Weapon Swinging, etc.). |
| Wednesday | Animations Continued. |
| Thursday | Began Transfer to Unity. |
| Friday | (Finally) Setup player in a scene (basically a prefab that was instantiable). |
| Saturday | Now can be controlled by controller (.Move) and have multiple cameras (POV, Third Person, Omnipotent). |
| Sunday | Began Testing for Animations (Using mouse clicks and other inputs). Little to no success (little fluidity and lots of buggy collisions). |

|  |  |
| --- | --- |
| Daily Log | Week 4 |
| Monday | Remade animations and used different hitboxes for Models in Blender, successfully rigged and animated new models. |
| Tuesday | Began designing dungeon layers, started by making square rooms (10x10 plains) that would randomly spawn next to each other based on what portal is collided with (N, S, E, W). Stored them as prefabs that would spawn different enemies, items and obstacles. |
| Wednesday | Continued to design the layouts of dungeons. Began testing ways to check positions and whether they have been used or not.  Decided on passing spawn points between scripts. Began Player Script development as well. |
| Thursday | Finished basic layouts and decided to use in game time as well as stages completed (public int) to buff enemies. Final room will occur when 3 things have happened (health == 1000 && Stages Completed == 100 && Relics collected == 100). This guarantees a fair fight entering the boss room. PlayerScript values are now properly displayed as UI and are controller with its script. |
| Friday | Boss AI, many issues with target selection (Decoy relic not functioning all the time (Fix. I was using a tracking system that did not work when called in update the way I wanted it to. Change to a NavMesh Agent controller for all other moving objects but Player and used the NavMesh from then on)). Other problems occurred but were dealt with quickly enough (not hitting boss, increased range and lowered avoidance in Agent Script. |
| Saturday | Now began the repetitive process of coding the rooms for different enemy spawns, item drops and obstacles. Issues with “checkforposition” and items spawning on each other. Still need to check for possible reasons why. |
| Sunday | Completed all! Normal rooms and created a master script for room spawning, stat tracking and overall control on the game components. |

|  |  |
| --- | --- |
| Daily Log | Week 5 |
| Monday | Began testing Overall game with many errors along the way. Objects that were instantiated must have a “(clone)” along with their .name component when checking for collisions. Relics didn’t affect the player sometimes, Player Script now controlled when relics have been obtained and activated. Combat was simplified (Melee was a simple lunge attack rather than multiple swing combo). |
| Tuesday | Continued Play testing (got my friends to try) games went by decently, remade health system with a shield rather than defense stat (DamageTaken = DamageTaken / Defence became Shield - = DamageTaken) to simply gameplay. Balanced weapons (Mainly the gun clips, too much ammo and not enough punishment for spamming fire). Swords (lunges) were balanced (risking positioning for large amounts of damage). One of the biggest problems was enemy scaling, people took wayyyyy to long to complete 100 rooms and the result was enemies that became to strong. Decided to make rooms cleared the ultimate factor for enemy buffing. |
| Wednesday | Went through scripts taking out unnecessary lines and fixing any older issues that had become apparent during testing (didn’t always need to make a while loop have a parameter that involved a premade variable (used while (true) and break;). |
| Thursday | Began setting up the menu scene with multiple options like a settings scene for sensitivity and overall appearance. Didn’t think of a good way to change a value from one scene to another so I just put settings in the game scene and the menu was purely for looks. |
| Friday | Created UI for the scenes and added a pause menu during the game that allowed for sensitivity changes and quitting. Overall, I finished the majority of the UI elements and am looking to make a final test phase for the game. |
| Saturday | Error with models, unable to load game in unity, went on all my backups (I had been making separate copies each hour so that if such a thing were to happen I could easily salvage my work) and all of them came up with the same error. Troubleshooting. |
| Sunday | Still unable to salvage game, all previous copies are lost, can’t get to scripts, prefabs nor scenes so I don’t know what I’m going to do. |

|  |  |
| --- | --- |
| Daily Log | Week 6 |
| Monday | Restarted from scratch because of extremely unfortunate error with my models… ☹ |
| Tuesday | Used test scripts made when making previous game to begin design for my new game “ShoGunDow”, a first-person shooter. |
| Wednesday | Made basic creations for player / enemies. |
| Thursday | Shooting is now controlled with ray casts and has a more CS : GO style vibe to shooting (no bullet drop and instant hit at every distance). |
| Friday | Made a large room that the player would have to maneuver through with randomly spawned Pillars, Ammo Boxes and (kind of) Enemies, which spawned at a random portal at each side of the room. |
| Saturday | Finished shooting coding with enemy hit and death tracking (increasing score and drop chance of boosts). |
| Sunday | Finished all Spawning components and moved on to game mechanics like enemy health, speed, damage, etc. |

|  |  |
| --- | --- |
| Daily Log | Week 7 |
| Monday | Polished game by adding visuals for shooting, shot buffers, boost activation, etc. Fixed slight errors with different enemy tracking (reduced obstacle avoidance to improve tracking ability and used multiple layers of NavMesh to customize each AI). |
| Tuesday | Created an end game sequence that allowed for restart. Added Menus (as previously made in other game with similar functionality). |
| Wednesday | Began play Testing for any serious bugs and or issues that would cause the game to run poorly or crash. |
| Thursday | Polished Scripts and created a new game description as well as explanation for how to play game. |
| Friday | Handed in! |

Other issues were solved relatively quickly (small mistakes in syntax, calling and setup) and were not worth mentioning. Overall the second games development went by much faster “Thanks To” the other one getting Destroyed.