Into the Dungeon

Jekko Syquia

March 4, 2021

Contents

1 Introduction

Into the dungeon is a dungeon monster slayer VR game targeted at the Oculus Quest device that is tethered a PC machine. The objective of the game is to go through each level and try to slay all the monsters present in each level of the game. Each level introduces new weapons, and with each new level the user is introduced to new enemies. New enemies deal harder damage as the game progresses. Each enemy AI is able to target the user and will shoot projectiles at the user when the user is at a closer range to the player. Every enemy also has their own score value that increases the users total score in the game.

This game main objective is to allow the users to experience streamlined games like *Doom* and other first-person shooter games in VR, thus enabling a tensed gaming experience. Unlike other streamlined VR games however, the user has a longer health bar than usual to allow the user to play at a more relaxed phased in the game. If the user dies, the user is respawned at the starting position of that level and the enemies are respawned. Users can also expect to keep a score count in the game to show off their friends. It is also important to notice that this game does not require the user to finish each level as they have the option of skipping to the next level if they would like. Therefore, it is a "finish your game at your own phase" and the score is dependent on the play time of the user.

2 VR Application Design

One of the most important key design decision when designing this game was balancing between physical engagement and motion sickness prevention.

Balancing is most notable in the exclusion of weapons that require physical contact with the enemies i.e. swords, hammers etc. This decision was important as too much physical engagement such as swinging the sword could disorient users by having them move and swing at the same time. This can make users uncomfortable and also would require more space to work around in the game.

The grabbing system in the game has also been changed. While the user can still interact with objects closely, interaction ray has been introduced to allow users to point at a weapon at which it will be redirected to their hands at any distance. This design decision prevents the need for users to physically reach their weapon when it is dropped on the floor.

Gameplay wise, users have infinite amount of ammo, this decision was taken into account to prevent users from needing to reload and prevents added complexity of loading their weapons which would make this game less arcade feeling.

The map design simplicity was also reduced greatly. Instead of moving around the map so much, the level was simplified to prevent users from getting lost in the game.

3 Integration

For the build-process, I primarily worked with C# Unity and Blender as the tool for modeling my enemies and level design. The weapons in the game was provided through the asset store in Unity.

The tool mainly used for handling VR interaction in the game was the inclusion of *XR Interaction Toolkit* through the package manager. This tool provided the tools necessary to handling the VR device. For example, interaction with objects in the game and player in game setup was handled by this tool. Mechanics such as

hand presence still had to be scripted and written individually.

Models in the game was built using Blender. This allowed for custom map models in the game and designing custom enemies not found in the asset store. The creation and baking of textures was also handled through Blender. In turn these were imported through Unity as an asset in which each enemy object has an enemy script attached to them to handle their behavior.

4 User Experience

Into the Dungeon has many design considerations to allow for a relaxed user experience. In this section we talk about the design decisions that make this game much more enjoyable. It is important to note that the game was meant to replicate arcade games like *Doom* as mentioned earlier. However the game is also meant as a way for users to physically more intense, but each level is not required be completed as the user can skip through each level as they would like.

4.1 Navigation

Users can expect to move in the game via teleportation, controller or physical movement. Since the game utilizes *Six degrees of Freedom (DOF)* users are not limited to teleportation. Teleportation is introduced in the game to mitigate the unconformability when moving with a joystick in the game. Furthermore, if the user has the space perhaps a large room, the users can use their physical space. While the games utilizes all three, they are not restricted to one and can choose any option as they which.

To ease the movement even further, the user has the option for *Snap Rotation* functionality. Snap Rotation allows users to flick the right analog stick left or right to a fixed degree of 45°.

An important design decision when it comes to moving from one level to another was the exclusion of doors. To make the game even easier to move around, I allowed

users to navigate to each level using the UI. This prevents the need to interact with doors to go to the next level of the game.

4.2 Guiding Audio/Sound Effects

Every weapon and enemy interaction is present in the game. For example, the handgun has handgun sound effects for firing to indicate it is firing. Additionally, once the shot has been fired sound effects will be present in the location of the collision. Therefore, there is a sense of 3D spatial sound in the game. Once the enemy is defeated the game also lets the user know the enemy is dead by indicating an explosion sound depending on the enemy.

Enemies in the game have their own individual sound as well. Each sound source will be relative to the location of the player. By doing this, we can have spatial awareness in the audio. Therefore, enemies shot far away will have a more muted sound, enemies shot closer will sound closer. This rule applies to all elements in the game to further enhance the presence of enemies in the game.

Levels also has background music that is playing throughout the game. Furthermore, when a player finishes a level, a sound cue is played to indicate to the player that level is done and will be moving to the next level.

4.3 Text Instructional

In the starting room the user will be given a tutorial on how to move about the room i.e. how to use the teleportation. Additionally in the beginning room the user will be provided with instructions on how to use the weapons by initializing UI elements that indicates what everything is. Whenever the user enters a new room an instruction on how to use that weapon will appear. Once the user has read and tested the item the user can ceed to walk forward to engage the enemies which they can immediately notice.

4.4 Instructional Signs/Wayfinders

Each room contains an instruction on how to use the weapons as well as how to move to the next room as the user is spawned in front of the instructions.

4.5 Six Degrees of Freedom

Six Degrees of Freedom (DOF) is present in the game. This design decision was an important implementation as moving with guns around the game would be a lot harder if the user does not have full control of their weapons.

5 User Guideline

5.1 Windows Installation

To use the game for Windows. Extract the *Windows Build V 1.2.7z* to their desired location using **7zip**. Assuming the user has already installed the necessary drivers to connect their VR to a PC. Simply execute the file *Into The Dungeon VR.exe* located in the Windows build folder. There are no further installations needed as the application is built.

5.2 Android Installation

To use the game for Android, make sure that the user is in Developer mode and install the oculus ADB driver on their PC.

Oculus. Once the user has downlaoded this, using the command prompt or terminal *cd* to the location of the adb and run the following:j

```
adb install -r C:\Users\YourName\Downloads\MyCoolNewApp.apk
```

On their device, users can then navigate to the app library and go to unknown sources to play use the application. The game will proceed to placing you in a tutorial setting where you can view the controls. For best experience use the teleportation (left trigger)

6 Conclusion

Considering the initial design works what work was the room to room dynamics of the game. However, the physical movement from one room to another was removed to prevent the user from getting lost. However, I do believe implementing UI room navigation was a lot more effective that indoors. However, what did not work was implementing swords and such as since it was a lot more complicated to implement physical damage through user collisions. What does need work is the level design, as I focused more on reducing unnecessary movements this also dumb down the levels in the game.

7 Suggestions for Further Improvements

To improve the game even more, perhaps the introduction of more complicated map designs could make the game more engaging and increase the repeatability factor of the game. Also including more weapons in the game can increase the difficult as well as being able to carry multiple weapons can be helpful for the user. To enhance the experience it would be helpful to optimize the models even further to be compatible with the Oculus Quest unlinked. This application runs smoothly connected to a PC.

A Lessons Learned

In designing this VR application my vision for what I want was pretty clear and achievable, however some complexities had to be cut down due to time constraints. The biggest challenge I faced was implementing enemy AI behaviour, collisions, and user movements.

AI behavior was certainly one of the biggest hurdles trying to implement in the game. As the enemy does not only move in 2D but have to compensate for 3D ray casting from the user in different direction. One of the struggles I initially had was calculating the path way for the enemy to go in the event of an obstacle in the way. This was however solved using NavMeshAgent that is built in Unity. This allowed me to implement enemies without having the need to calculate if the next step is possible as the component calculates the next possible step to the destination. Within this script I was also able to implement different states for the enemy with the following: Patrol, Chase, Attack. This was also a struggle to implement as it was hard to calculate the threshold of when the patrol, chase and attack occurs. However, I found that 20, 10, and 5 to be the best spot for attacking.

The next struggle in the game was calculating the Collision and when it should happen. I discovered that the method *OnCollisionEnter(Collision)* and *OnCollisionStay(Collision)* were two different things. When calculating enemy damage I used on *OnCollisionEnter(Collision)* as using the alter would keep decreasing my health when a projectile collided with the player especially if the projectile had a slower velocity. This was the main reason I excluded using swords in the game as I did not know when the damage should keep inflicting or when an enemy actually attack the enemy opposed to having the sword just touching the enemy.

User movement was a struggle however, when I found out about *XR Interaction Toolkit* the implementation was simplified. Once the rig was setup it was primarily implementing the behavior of weapon objects and applying scripts to them that correspond to user button press such as the shoot button.

B Feedback to the Instructor

For the VR Dev platform, it was very helpful to use Unity with C# as the learning was transferrable to other game developments that use the same platform. The grouping is pretty helpful, however I think it would be helpful to have at least one demo of a feature or code in class to help the class get started. For example, showing a simple instruction as to how to grab an object in a game in Unity. In addition, implementing a simple VR setup such as showing the player in game would be a helpful assignment to have and keep everyone at the same phase and prevent them from being discourage to develop in Unity as I certainly found it overwhelming at first if I had not known anything about Unity or Blender experience.

The collaboration groups are pretty helpful as it allowed us to reflect on our applications and evulate VR design factors that would help our applications work effectively. Thus, collaboration reveal weaknesses and strengths in our applications.

C Classmates' VR App Evaluation