G01: A Night on the Docks

Writeup

Michael Kelleher (mjk289), Jackie Peng (jtp224), Kevin Pratama (kp432), Ritika Poddar(rp477)

December 6, 2021

INFO 5340 / CS 5650: Virtual and Augmented Reality
Cornell Tech

Project description

You wake up in the middle of the night abandoned on a mysterious dock to the sound of a booming voice. Before you know it you're in a fight for your life. Zombies have risen from the dead and are attacking you from all sides. Luckily, a mysterious purple spirit is equipped with a variety of weapons and defense tools to help you survive. Still, it will be a challenge to defeat the undead swarm who seem to be spawning out of nowhere. Can you survive until the morning?

A Night on the Docks is a fast paced virtual reality game that puts you right in the middle of this dramatic tale. A perfect experience for someone new to VR, the game begins with an easy-to-follow, engaging tutorial that teaches you everything from how to teleport to how to detonate a grenade. From there, it is up to you to fight off the zombies and make it out alive.

Instructions

***Menu/UI*:** The player can toggle the pause menu using the menu button on the left controller.
***Locomotion*:** To teleport, the player can use the left joystick and aim to a target location. They use the same joystick to select a target orientation and release it to teleport. To get to the elevated platform, they can press the glowing yellow elevator button with their virtual hands.
Weapons and Shield: The player can use the grip button on either controller to distance grab any of the weapons from their toolbelt. They can use the trigger button to shoot the crossbow and can physically swing the sword. To throw the grenade, the user has to release the grip button while swinging the controller, mimicking throwing objects in real life. The player reaches to their upper back and presses the grip button with either controller to grab the shield. They can then use it to reflect enemy bullets.

Features

Locomotion*:* The most basic mode of locomotion is simply walking around the space. Teleportation is the primary mode of moving large distances. The player can aim using the controller, orient themselves with the thumbstick, and release to teleport.
Elevator Button: There is a second level that can be accessed by operating an elevator. It is operated by a physical button the player presses with their hand. Then the platform they are standing on rises up or down, and there is a haptic rumble to add further movement cues.
Weapon Toolbelt*:* All the player's tools (except for the shield) are located on a toolbelt that follows the player around.
Sword: The player can physically swing the sword to attack zombies. This is done by grabbing the weapon and swinging the sword directly in contact with the zombies. Swinging the sword at the limbs will cut those specific body parts, and will kill the zombie if enough body parts have been cut off or if the chest and head are targeted.
Crossbow: The player can interact with the crossbow by grabbing the controller using the grip button and press the trigger button to shoot the arrows. If the arrow hits the zombie, it will damage the zombie by cutting off that specific body part, and will kill the zombie if it has hit the

zombie enough times or hit the head and the chest. Otherwise, the arrow will get stuck at the terrain before disappearing.

***Shield*:**The shield is located behind the player, near their upper back, and it is out of view when it is not grabbed. It can be grabbed via distance grab, similar to the tools on the toolbelt, when the player reaches over their right or left shoulder with either hand and presses the grip button on their controller. The shield deflects bullets back to the zombies by hitting them back in the normal direction of the shield face, which can then incur damage on the zombies if the bullets hit them. The controller holding the shield will vibrate if a bullet is successfully deflected by it (if it bounces off the surface of the shield).

Zombie: The zombies spawn randomly from 10 locations around the landscape, with only 3 alive at a time. With each frame they always rotate to look at the player and slowly start to make their way towards the player. The zombie has five general hit zones, if it is hit in the head/chest it dies instantly. If it is hit in any of the limbs, on any part of the limbs, that limbs disappears. The zombie is a rigid body that abides by gravity so if both its legs disappear it falls to the ground but can still move towards you until it is hit a third time and killed.

Player Hands (Distance Grabber): The player's hands can grab the weapons from a distance to make it more ergonomic for the user.

Player Health: The player can be shot up to 25 times before losing the game. This is visualized with 5 green health bars on the player's left hand. After each fifth hit, one bar is removed. There is also a red flash of light after each hit to provide visual feedback of damage.

Win/Lose Conditions: The player wins the game when they kill all 10 zombies. They must either hit the zombie directly in the chest/head to kill instantly, or hit it three times in any of its limbs to kill it. When the player wins, the win game menu appears that says "You won the game" and gives them the option to restart, go to the menu, or open settings. The player loses the game when they run out of all 5 health bars (hit 25 times). When the player loses, the loss game menu appears that says "You lost the game" and gives them the same menu options as the win game menu.

Pause Menu: When the player presses the menu button, time stops and a menu is brought up directly in front of the player. Locomotion is disabled, and the menu remains stationary until the player either clicks the menu button on the controller again or clicks the resume button in the menu. They can also restart the game, or go back to the main menu. Finally, they can open a settings menu, which allows them to use a slider to change the background music volume.

System Control:In every menu, there is a laser pointer that shows the user where their selection is. When a button is hovered over it changes color and gives haptic feedback when pressed.

Tutorial: This is a separate scene with a simpler, more controlled gameplay. Each feature is rolled out one at a time, giving the player a chance to learn the game mechanics. As the stages of the tutorial progress, more and more features are turned on. To progress to the next stage, a player has to finish a specific task, such as kill a zombie with the crossbow. Instructions are delivered by a purple spirit figure with custom voiceover with auditory effects.

Background Music: A free, open license audio track was used to provide a more exciting environment and set the thrilling tone we were looking for.

Haptic and Auditory Feedback: Most game features have sound effects and/or haptic feedback to give the user more presence and make it clear when actions have occurred.

Technical contribution

Environment Design/Lighting: An external asset was used for the environment, but it looked very flat with the general ambient lighting. Thus, we used a collection of in-game light sources, primarily point lights associated with lamp objects, to make a more natural feel. The water was lit separately with a directional light to mimic moonlight. The sky is rendered using a sky box. To avoid a hard edge at the end of the ocean, we built a fog particle system around the perimeter.
Locomotion: Locomotion was largely implemented using the Oculus Integration, with a few modifications. The first was changing the look of the destination prefab and assigning new materials. Second, there was a major bug where the player could get stuck at off angles, including upside down. This was fixed with a modification of the source code to correctly orient the player. To control where the player could access and prevent them from teleporting onto walls, a layer called walkable was assigned to the relevant floor game objects. This prevents them from teleporting into the building or the water as well.
Elevator Button: This uses a Configurable Joint component between two cylinders to allow the button to be pushed and return to its base position. This triggers a smooth lerping of the elevator to the higher or lower position. This could in theory cause some vection, so we added further cues like a haptic rumble, and kept the duration of the movement short.
***Weapon Toolbelt*:** The toolbelt is essentially an invisible player body that is located under the PlayerController during the game, implemented as a cylinder that is a child of the PlayerController GameObject. In order to keep the toolbelt from rotating with the player's head, the orientation of the body is constantly set to a fixed value at each frame and this fixed value is updated only after locomotion or snap turns. When the player grabs an object from the toolbelt, the object is removed as a child of the player body. When the player drops an object (or in the case of the grenade, when the grenade is finished exploding), the object is automatically returned to its original position on the toolbelt. Each GameObject that belongs on the toolbelt has a ReParent function, which sets the isKinematic property of the Rigidbody component of the object to false, sets the parent of the object to the player body again, and updates the local position and rotation of the object to its original values to fix it on the belt.
Sword: The sword is implemented by importing a 3D sword model from the Unity Asset Store, and is implemented through RigidBody and box colliders. Each part of the sword is represented with the box collider, allowing for accurate collisions with the enemies. Moreover, the sword can be grabbed by distance grabbing the sword using the grip button with either controller, which snaps to the player's hands. The sword can be swung at the zombies while holding the grip button, which triggers the audio and haptic feedback upon collision with the zombie body parts.
Crossbow: The crossbow grab works in a similar manner with the sword. The collision of the bow is also implemented using box colliders, with the difference being that it does not kill or damage the zombies. The crossbow shoots an arrow when the respective controller's trigger button is pressed down. The arrow is implemented with a rigidbody (gravity enabled) with box colliders, which falls off as the arrow goes further away. The arrow damages or kills the zombie upon impact.

Grenade: The grenade is implemented using the box collider and rigidbody to enable grabbing as well as the motion and Physics of throwing the grenade. Moreover, when the grenade is thrown, it has the timer countdown which checks for any zombies nearby upon explosion. If a zombie is caught, it will be set to inactive and another zombie instance with no skeletons will be created. This new instance has the mesh collider and rigidbody for each of the body parts. Explosion force which is applied to the parts will launch the body parts away from the explosion point. The debris will also be destroyed after five seconds.
Shield: The shield is implemented with the imported 3D asset, represented with a box collider and rigid body. The shield can be grabbed in a similar fashion with the weapons, and blocks the enemy projectiles by checking if such collision happens with the enemy bullets. The enemy bullets then will be reflected upon collision, which is implemented by reverting the direction / orientation of the bullet's force. As such, the reflected bullet is able to damage the enemy back.
Zombie (Body Parts, Physics): The zombie was implemented using a free asset from the Unity Asset store. It is made up of about 15 separate body parts. There is also a black particle system surrounding it as smoke. It is implemented as a rigid body that abides to gravity. There are five different capsule colliders on the zombie. One covers the head/chest/hips, if the player hits this collider the zombie dies instantly (set inactive). There are two capsule colliders for each of the arms, each covering the palm/forearm/upper arm for the left and right side. And there are two capsule colliders for each of the legs, each covering the thigh/calf for the right and left side. If any of the four limb colliders are hit, all of the body parts that are tagged with that collider are set to inactive. At every frame the zombie rotates to look at the player, and moves slowly towards the player up to 0.5 meters away.
Player Health: Each player has a hit count, and for every 5 hits a health bar goes inactive. The health bars are children of the left hand and can be seen at all times on the back of the hand.
Win/Lose Conditions: Each zombie has a hit count, on the third hit it is set to inactive and when all 10 zombies have been set to inactive the win game menu appears. This menu looks similar to the pause menu but only has the options to restart, go to menu, or settings. When all five health bars are inactive the game is over and the loss game menu appears. This is implemented the same as the win game menu.
Pause Menu: This has the primary function of pausing the game by setting the time scale to 0. This prevents enemies from moving. It also is scripted to prevent player locomotion as the pause menu will be world stabilized. The menu is positioned based on a fixed offset from the player transform and looking-direction.
System Control: This is largely done utilizing the Oculus framework to bring in a laser pointer. This interacts with a UI panel where buttons and sliders control game functionality. Many buttons are scripted to load new scenes.
Tutorial: This is built as a separate scene and includes many of the features listed separately. The player is guided by a new character with spatialized voice over audio. The tutorial is divided into 13 phases, each with a different completion check. When each is reached, it triggers a new voice over and sets the new game objects being explained as active.

Team Member Contributions

- Michael

  - Environment setup and additional design, all menus, locomotion and player controller, lighting, system control, tutorial design and logic, sound effects, assistance on all zombie behaviors.
  - ButtonHaptics, Elevator, PhysicsButton, SceneController (Components dealing with pausing, haptics, and loading scenes), TutorialController
- Ritika
  - Zombie: Spawning timing and locations , Zombie particle system, Behavior, rotation and movement, damage/kill logic. Player damage, health bar deactivation. Win and lose conditions, Win and lose menus.
  - Scripts: SceneController(contributions to zombie spawning, win/lose conditions, win/lose menus), PlayerBehavior(player health/damage), ZombieDie(zombie damage/death), ZombieBehavior(contributions to zombie movement)
- Jackie
  - Weapon toolbelt and grabbing (BodyBehavior, contributions to CrossbowShoot), shield (ShieldBehavior), sword (SwordBehavior), zombie damage sound effects (contributions to ZombieDie and SceneController)
- Kevin
  - Crossbow (Haptic & audio feedback, arrow physics), grenade (Physics, explosions, debris physics upon explosion), player Damage (Vignette), changed shaders and emissions to improve visibility.
  - CrossbowBehavior, CrossbowShoot, GrenadeBehavior, PlayerBehaviorZombieBehavior, ZombieDie

Other comments

Overall, this was a rewarding project. We approached it with a clear vision of what we wanted the final project to look like, and while there were changes made along the way, it was satisfying to see that vision become reality. One key challenge was collaboration with Unity. Merge conflicts were abundant when working with branches and often very difficult to resolve. We also found that often the work of one member would unwittingly affect the logic of a different feature, so that had to be resolved. Another, more general challenge/learning was just how detail-oriented VR development has to be. You have less control over what the player sees and have to take more care in making the experience comfortable. This leads to focusing on a lot of small details that can take a lot of time and focus to get right.

If we had more time, some things we would have liked to implement are:

- Evolution of weapons, for example the sword becomes a flaming sword
- Different levels of difficulty, such as introducing a larger even harder to kill enemy once enough of the zombies have died.
- Introduce some animation into the zombie behavior
- Different amounts of damage done to zombies based on weapon used
- Work on the overall balance of the utility of the weapons.

ADDENDUM - EXTERNAL ASSETS

Music

Epic by Bensound: https://www.bensound.com/royalty-free-music/track/epic

Environment

Old Sea Port by Christopher Fantauzzo:
https://assetstore.unity.com/packages/3d/environments/old-sea-port-environment-36897

Grenade

FC User Interface SFX [Lite] by fcaudio:
https://assetstore.unity.com/packages/audio/sound-fx/fc-user-interface-sfx-lite-167083
Grenade Sound FX by MGWSoundDesign:
https://assetstore.unity.com/packages/audio/sound-fx/grenade-sound-fx-147490
Grenade Pack Free by Black Light Studio:
https://assetstore.unity.com/packages/3d/props/weapons/grenade-pack-free-83440

Crossbow

Classical Crossbow by RyuGiKen
https://assetstore.unity.com/packages/3d/props/weapons/classical-crossbow-196127
Bow and Hammer Sound Effects by MGWSoundDesign
https://assetstore.unity.com/packages/audio/sound-fx/weapons/bow-and-hammer-sound-effects-163948
Whoosh & Grab Sound Effect by
kiGZkeViNhttps://www.youtube.com/watch?v=qZTOiwRFCDk

Sword

Seven Swords - Seven Stylized Swords by Perpetual Divergence

https://assetstore.unity.com/packages/3d/props/weapons/seven-swords-seven-stylized-swords-14304

Drawn sword sound effects (free) by John Doe

https://www.youtube.com/watch?v=U34MegCHlig

Swish, Swoosh, Cutscene Sound Effect by Animal Jam Giveawayhttps://www.youtube.com/watch?v=EFY9TX2Fghg

Best Sword Sounds (Slice/Slash/Crash/Swoosh/etc):

https://www.youtube.com/watch?v=EgRvVq8mStE

Shield

skull shield by Vitar Publishing

https://assetstore.unity.com/packages/3d/props/clothing/armor/skull-shield-76967

Captain America Shield Sound Effect HD - Royalty Free Sound by Creator Effects

https://www.youtube.com/watch?v=1UltDCRu-2Q

Characters

Zombie by Pxltiger:

https://assetstore.unity.com/packages/3d/characters/humanoids/zombie-30232

Ghost: https://sketchfab.com/3d-models/ghost-a0f01623a7c1418caaee24fa1c685c5e

Visual Effects

Unity Particle Pack 5.x:

https://assetstore.unity.com/packages/essentials/asset-packs/unity-particle-pack-5-x-73777

Player Sound Effects

Male Pain Grunts Sound Effects SFX Free and Royalty Free:
https://www.youtube.com/watch?v=5RW4Hfz-MJg