Shark Bait

Concept

Shark Bait is an underwater survival game where the player must take on increasingly difficult rounds of enemies. The main enemy type are the Sharks who have increasing speed, health and damage as the wave go on, they also increase in numbers. Another enemy type are the jellyfish who act as a zoning obstacle where is the player touches one it will damage them however, they are not actively pursuing the player. Additionally, dolphins and turtles roam about the level with dolphins flocking together and providing the player a speed boost when in proximity and turtles being killable for a shield boost. While fighting the incoming waves the player must also be attentive to their oxygen meter and ammo count, managing it well enough to survive the ocean.

Assets used:

|  |  |  |  |
| --- | --- | --- | --- |
| Asset | Source | Accessed | License |
| Shark model/anim/texture | https://www.cgtrader.com/free-3d-models/animals/fish/realistic-animated-great-white-shark | 21/11/22 | Royalty Free License |
| Waterbox Texture | https://opengameart.org/content/ocean-hdriskybox | 23/11/22 | Public Domain |
| Harpoon gun and Harpoon model/ texture | https://sketchfab.com/3d-models/harpoon-rifle-5456adeff51947d5bf66a4563bb3821f | 23/11/22 | CC Attribution |
| Bubble Emitter model/texture | https://www.cgtrader.com/3d-models/plant/other/corals | 24/11/22 | Royalty Free License |
| Height map | Generated myself using: https://tangrams.github.io/heightmapper/#7.20833/26.401/-59.591 | 23/11/22 | Public Domain |
| Ammo crate texture | https://3dtextures.me/2021/02/12/wood-crate-001/ | 23/11/22 | Public Domain |
| Sand Texture | https://open3dmodel.com/3d-models/coral-reef-scene\_595273.html | 24/11/22 | Free for personal use |
| Red background for health pack and crossfade | https://www.needpix.com/photo/1384018/square-red-background | 01/12/22 | Public Domain |
| Jellyfish model | https://creazilla.com/nodes/3281-set-of-jellyfish-3d-model | 06/12/22 | Creazilla Open-Source License |
| Turtle model and texture | https://free3d.com/3d-model/-sea-turtle-v1--427786.html | 05/12/22 | Personal Use License |
| Menu Background | https://unsplash.com/photos/XexawgzYOBc | 07/12/22 | Free to use |
| Boat model and Textures | https://www.turbosquid.com/3d-models/boat-pbr-model-1522670 | 21/11/22 | https://blog.turbosquid.com/turbosquid-3d-model-license/ |
| Dolphin model/Texture/Anims | https://www.cgtrader.com/free-3d-models/animals/mammal/realistic-animated-dolphin | 09/12/22 | Royalty Free License |
| Rock Texture | https://www.freepik.com/free-photo/cut-tile-shade-durable-wallpaper\_1066914.htm#query=rock%20texture&position=4&from\_view=keyword | 10/12/22 | Free license |
| Game background audio | https://pixabay.com/music/build-up-scenes-talaso-extasis-119659/ | 10/12/22 | Simplified Pixabay License |
| Bubbling Audio | https://pixabay.com/sound-effects/bubbling-6184/ | 10/12/22 | Simplified Pixabay License |
| Shark Bite | https://pixabay.com/sound-effects/monster-bite-44538/ | 10/12/2022 | Simplified Pixabay License |
| Gun Shot | https://pixabay.com/sound-effects/gunshot-37055/ | 10/12/2022 | Simplified Pixabay License |
| Menu Music | https://pixabay.com/music/ambient-dark-cinematic-suspenseful-ambient-111682/ | 10/12/2022 | Simplified Pixabay License |

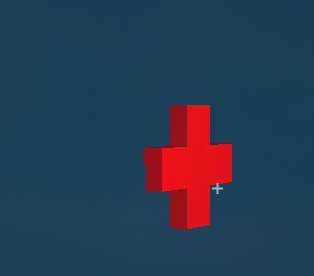
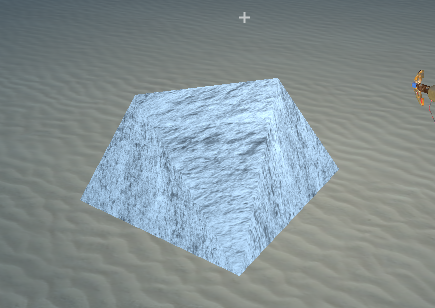
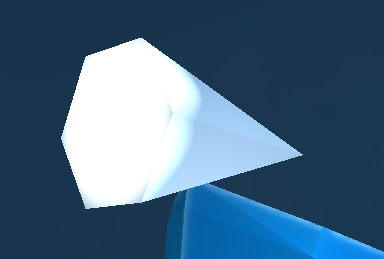
I produced both the death and main menus using the website Fotor.com, I placed in the cited image as a back drop and added the appropriate text to each menu, exported and resized the image accordingly.

Features Implemented:

Part 1:

As mentioned above I created 2 types of menu and implemented them into the game by rendering a quad in the orthographic camera for each menu if the corresponding menu flag was active. For the main menu the flag is active by default and deactivated by a key press when viewing the menu. For the death screen the flag is set to false by default but once the player’s health reaches 0 or below the flag is triggered displaying the death menu, the menu is then deactivated on key press event and the game level is reset to its default state.

For the primitive shapes I constructed and implemented:

* Plus Icon:
  + 
  + Repeated and transformed multiple times across the game world
  + Correctly lit
* Prism:
  + 
  + Textured correctly and placed in the game world
  + Correctly lit
* Hexagon based pyramid:
  + 
  + Placed and transformed in the gameworld
  + Correctly lit

These were all created by plotting the vertices, joining the vertices correctly to form the current indices and creating the faces to display correctly with the correct lighting normal. Furthermore there is a primitive cuboid shape repeated, textured and lit correctly in the form of the collectable ammo crates, as well as a primitive sphere shape as the oxygen bubbles which is repeated and transformed within the game world and lit correctly with a translucency.

For game audio I have implemented 2 different music tracks as well as 3 spatialized audio sounds. I have 1 music track playing whenever a menu is displayed and the other as background music to the gameplay. The spatialized sounds trigger on events with: bubble sound being played when a bubble is created by an emitter, with the sound also being emitted by the emitter. The bite sound playing when the shark bites and damages the player which is emitted from the shark’s position. The gun-shot sound playing when the gun successfully shoots a projectile, the sound is emitted from the gun’s position.

The players HUD displays their: Health, Shield, Ammo, Oxygen level as well as the current wave of the game. All these values are displayed on the left hand side of the screen and are rendered as text in the orthographic camera.

Part 2:

For camera motion, the camera is bound to the waterbox and terrain area with traditional WASD movement as well as LShift to decent and Space to ascend.

I have several mesh based objects in my game scene with 2 of which being animated:

Animated:

* Shark
  + Repeating enemy type
  + Transformed
  + Textured
  + Lit – Mostly correct, some odd artefacts when the animations are applied, potentially due to animation export settings
  + Reactive Animations the play under different scenarios
    - Swimming
    - Faster Swimming
    - Biting
    - Dying
    - Death
* Dolphin
  + Repeating NPC type
  + Animated correctly
  + Transformed
  + Textured
  + Lit – Mostly correct, some odd artefacts when the animations are applied, potentially due to animation export settings

Static:

All lit, textured and transformed correctly

* Shipwreck
* Gun
* Turtle
* Jellyfish
* Harpoon Gun
* Harpoon

In the game I have 5 point lights at the locations of the bubble emitters emitting an orange glow around the object to draw the player in and alert them of the location. A directional light mimicking the sun. 2 spotlights, 1 of which is of higher intensity and is emitting a white light from the shipwrecks lamp towards the centre of the level as well as a smaller white directional light with higher cut-off moving with the player’s gun and pointing in that direction acting as a flashlight/torch.

For special effects I have a blue fog which is only displayed if the player is underwater of which the density was adjusted in the shader settings to give the desired effect. The jellyfish have a level of transparency applied to them as well as the oxygen bubbles. Furthermore there is crossfading implemented between the camera view of the level and a red quad when the player take damage to their health, the code for the crossfade class was sourced from the FX example provided in week 5.

Part 3:

I was unable to implement game physics as when I added my game objects to the physics manger they were transported far beyond the clipping plane for unknown reasons, furthermore as the game is set underwater I don’t think their would have been much opportunity to implement interesting physics besides just velocity to the game objects and acceleration as implementing buoyancy would have made the game less fun as controlling the player would be difficult.

I have 4 npcs (Shark, Turtle, Jellyfish and Dolphin) all of which using state machines to adjust their behaviour with shark being the most intricate. The shark has 5 states with the transitions being as such: Passive -> Attack on random probability if no more than 2 sharks are attacking the player -> Strike when close to the player -> Flee when it has successfully attacked the player. If the shark’s health slips blow 0 it enters the dying state which then transitions to the death state once the animation is complete. The dolphins use a flocking mechanism to move around the map once they are in a pod, otherwise they select a random point in the game world and move towards it, like the turtle, jellyfish and shark in its passive state.

For gameplay elements I have implemented multiple types of pick-up item (oxygen bubbles, health and ammo crates) as well as a speed boost when in proximity to a dolphin and multiple timer elements to manage cooldown events such as damage causing events and shooting.

Conclusion:

Unfortunately, I did not achieve all the features I wished to add: story elements, caustics, optional endgame mostly due to time constraints but some due to the game engine limitations or my understanding of them but I am very proud of the game I produced and plan to carry on building out this game to add to my portfolio as well as potentially adapting it into a demo in unity or unreal.