REEF RUNNER

High Concept Document

Dylan Ramsden

Table of Contents

Title

Reef Runner

Level Designs

Level 1:

Assets:

- Beautiful coral reef blocks
    - Boarders the running space
- Giant rocks passing over the running space
- Plenty of small fish and sea creatures
    - Sun fish
    - Little Tuna
    - Whales
    - Dolphins
    - Schools of colorful fish
- Abundance of coral
- Plenty of shells

Dangers:

- Hungry sharks will not resist a snack if you get in the way of their path
- Spiky sea urchins
- Ocean trenches leading to endless depths of the ocean
- Strong currents

Interactables:

- Friendly turtles willing to lend a shell
- Bubbles to help sustain your oxygen level
- Shells

Colors:

- Very bright and vibrant colors
    - Aimed to create a beautiful and exciting atmosphere
- Ocean floor is very brown-very similar to a typical coral reef and the water color is very blue

Light:

- Lots of sunlight, the reef being close but unreachable to the surface

Music/sound:

- Very calming and relaxing music
    - Aimed at creating a very relaxing experience at the start of the coral reef
- Large sea creatures each have their own sounds that is activated once the player gets closer to it

Level 2:

Assets:

- Very plain dark rocks
    - Boarders the running space
- Very scarce amount of sea creatures
    - Shrimp
    - Hermit crabs
    - Angler fish
- Scarce plain coral
- Particle systems resembling neon plankton
- Very few shells

Dangers:

- Lack of oxygen bubbles
    - Due to the depth of the reef
- Aggressive angler fish
    - Scared off by bright lights
- Glowing jelly fish

Interactables:

- Batteries
    - For your torch
- Oxygen bubbles
- Friendly hermit crabs willing to lend a shell
- Shells

Colors:

- Very dark and neon like colors
- Ocean floor is very dark and water is a very dark green color

Light:

- Minimum sunlight as this level takes place in a deep area of the reef

Music/sound:

- Very cautious and relaxing music
    - Emphasizes the unknown and what dangers lie in the dark

Level 3:

Assets:

- Sand Dunes
    - Very open dunes with plenty of rocks and coral
- Shipwrecks
- Plenty of fish
    - Schools of fish
    - Sharks
- Colorful sea urchins

- Turtles
- Sea birds
- Bubbles

Dangers:

- Sharp blanks of wood
- Falling debris from a current shipwreck
- Hungry sharks
- Spiky sea urchins

Interactables:

- Friendly turtles
- Hungry seabirds
    - Used to boost player jumps and movement
- Bubbles

Colors:

- Very bright and vibrant colors
    - Very colorful coral and sea creatures
- Sand color is a light gold color
    - Much closer to the surface than any of the other levels
- Water color is a light blue

Light:

- Plenty of sunlight
    - The level and its assets are very visible and lit

Music/sound:

- Very calm and ambient music
- Sounds of splashing
    - From diving sea birds

Pick-ups/Interactables

General Pick-ups/Interactables:

- Bubbles:
    - Due to the diver's constant loss of oxygen-bubbles are used to sustain oxygen once picked up
    - Bubbles, must be constantly picked up throughout the entirety of the run
    - Bubbles have a decreased spawn rate the longer the player runs for
- Shells:
    - Shells can be picked up to add points to the player's score throughout the run
    - Shells come in many different forms and each different form adds a different points value
        - Blue shells = 30 points
        - Yellow Shells = 50 points
        - Gold Shells = 100 points
    - The more points a shell gives the less chance it has to spawn throughout the run

Level 1:

- Turtles:
    - Turtles are very friendly creatures found within the reef
    - When a turtle is picked up-it provides the player with a shield
        - This shield is used to block shark and sea urchin attacks
    - If a player already has a shield and they pick up more turtles, they gain a score multiplier
        - Every turtle a player picks up while having a shield-increases the multiplier by *1*
        - If the multiplier is active, the players increasing score *(per a second)* is multiplied by the turtle multiplier
    - The turtle multiplier is active until the player loses their shield-it then resets the multiplier back to 1
- *All general pick-ups have a standard spawn rate*

Level 2:

- Batteries:
    - Once the player enters 'level 2' they are able to use their torch mechanic
        - Used as a light source in the dark stage
        - Used to scare of angler fish
    - The torch has a power supply that constantly decreases when being used
    - Batteries are used to add more power to the torch
        - Once the torch dies the player has to maneuver blindly through the stage
        - With more batteries the play can easily maneuver with light
    - Batteries spawn very scarcely
        - Players must use their torch wisely
- Hermit crabs:
    - Hermit crabs are used the same way turtles are
        - Hermit crabs provide the player with a shield that can block a jellyfish sting or angler fish attack

    - Hermit crabs also activate the score multiplier
        - Once the player has obtained a hermit crab shield, picking up more will increase the multiplier
- *Oxygen bubbles have a decreased spawn rate*

Level 3:

- Hungry seabirds
    - Hungry seabirds dive into the ocean looking for fish
    - A player is able to grab onto a seabird
        - The seabird will increase their jump height as well as movement speed
    - The seabird pick-up lasts for 10 seconds
    - Hungry seabirds spawn very rarely
        - Have an increased spawn rate during the ‘level’ 3 ‘boss’ interaction

Boss mechanics

The bosses get progressively harder throughout the different levels. This is done to allow the player to gradually adapt to the complexity and increasing difficulty of the game while still having fun and being able to explore and observe the different stages.

Level 1:

- Strong currents
    - The 'boss' the player encounters in level 1 is strong currents
    - The strong currents constantly push the player to the *left*
        - This makes movement and dodging obstacles much more difficult
        - If the player is moving in the opposite direction of the current, they will move much slower and with less force
        - If they are moving in the direction of the current, they will move much faster- making the controlling of the player much harder
    - Implementing strong currents in the level makes the reef feel much more realistic and interactive by the player
        - The boss also adds a new sense of difficulty to the players run, while still leaving room for enjoyment and relaxation

Level 2:

- Angler fish 'boss'
    - The 'boss' the player encounters in level 2 is a group of aggressive angler fish
    - This boss implements a new danger/mechanic rather than one individual threat
    - The player finds themselves encountering very aggressive and angry angler fish that will attack if the player gets in the range of them
        - The player will only ever find one angler fish on its own at a time
    - Counters:
        - The player can use their torch to scare of an angler fish if they are in range of it
            - Causing it to swim away rather than attack
        - Hermit crab shields can block an angler fish attack
            - Hermit crabs are very scarce during the 'boss' interaction

Level 3:

- Shipwreck 'boss'
    - The 'boss' the player encounter in level 3 is falling debris from a shipwreck above the reef
    - This boss essentially adds more obstacles/dangers for the player to dodge and maneuver around
        - Large pieces of debris from the shipwreck fall swiftly onto the players path
        - Being hit by a piece of debris will cause the player to immediately die
    - Counters:
        - Seabirds can be used to maneuver quickly around and over the falling debris- making it easier to survive this 'boss'
        - A turtle shield is able to block one hit from a piece of debris

Game Concept

Reef Runner takes the player on a breathtaking journey along a beautiful and ambient reef, interacting with turtles, hermit crabs and hungry seabirds to fend of the dangers of the deep and defeat suffocation.

Rules

Player/User Interface:

- Player enters their name at the end of each run
    - Once the player has entered their name-their score is stored in a database
- The player can view the scores database in the leaderboards menu
    - Players can search for their name and look what their current scores are in the database
    - Players can sort the database to view the high scores and where they stand in the leaderboard
- Players cannot reset the database
    - Only the database administrator (Dylan James Ramsden) will be able to alter and reset it
    - The database will be reset once a month

Sound:

- Players can alter with the overall game volume-tweak it to their preference
- All sounds and music can be muted by the players

General gameplay:

- Players score is constantly increased
    - This is done with a timer
- Player oxygen level constantly decreases
    - Can be increased by picking up a bubble
    - When the players oxygen is depleted, they die

Gameplay UI:

- The players score will be visible at all times throughout the gameplay
- The current high score will also be visible
    - Visually this will be shown as a school of gold fish during the players run, to assure they are aware once they have beaten the current high score
- An icon will display whether they player has a shield or not (either a turtle or hermit crab)
    - The score multiplier will be displayed next to the shield icon
- The players oxygen level will constantly be displayed during gameplay
- If a player picks up a shell, the bonus score it provides will be displayed next to the players score for a short amount of time-then added to the players score

Player Mechanics:

- The player moves forward by default
    - The players forward movement speed gradually increases the longer the player runs for
- The player can move left and right (A and D keys)
- The player can jump (Space key)

- The player has two interactive mechanics:
    - The player must click the E key in order to pick up a turtle shield or hermit crab shell
        - The player must be near the creature in order for it to work
    - The player must click and hold the Right Mouse key in order to use their torch

Levels:

- A level change takes place every 2000 points
    - The level change is calculated off of how many points the player has (the players score)
- A levels 'boss' is activated at 1500 points
    - Just before the 'boss' is activated (1350 points)-the player will be given a warning message e.g. "Strong currents ahead"
    - A levels 'boss' will be activated until 1950 points-just before the level change
- When the level change occurs, a random level is chosen to generate

Difficulty Increase:

- The difficulty of the game increases gradually the longer the player runs for
    - The gradual difficulty increase is very important as the game must get harder but in a way for the player to still enjoy it-no matter how hard it gets
- Players speed increases over time
- Every boss encounter-no matter which boss it is, will add more complexity when encountering it
    - Current 'boss' – the current will get stronger every time it is encountered
    - Angler fish 'boss' – more angler fish will be spawned
    - Shipwreck boss – more debris will fall onto the running space and at a much faster space
        - Less seabirds will spawn
- Over the course of the run-less oxygen bubbles will be spawned
    - This makes oxygen bubbles a much higher priority in the later stages of the run
- Less interactables will be spawned
    - Turtles
    - Hermit crabs
- More general obstacles will be spawned/dangers
    - Sharks
    - Jellyfish
    - Sea urchins
    - Trenches

Game classes and objects

Primary Game scripts:

GameManager:

Location: Stored in the *MainMenu* scene

- *The reason the script is stored in the MainMenu scene is because it ensures the singleton is created when the game first loads up.*

Purpose:

- The GameManager script is a singleton-this means the script is never destroyed throughout each scene
- The script is also defined as static to ensure there is only ever one instance at a time
- The script is responsible for storing general values and connecting the managers of each scene together
    - Storing the player score between the GameplayScene and DeathScene to display the right score on death
    - Storing the players score and name to pass to the DatabaseManager to store in the database
    - Storing the volume level to ensure the volume is set to the same user defined value between scenes

RunManager:

Location: Stored within the GameplayScene

Purpose:

- The RunManager script is responsible for storing the primary gameplay information during the players run
    - Player score
    - Player shield multiplier
    - Oxygen value
    - Level
    - The players current alive/dead Boolean
- The run manager is destroyed at the end of each run and re-created with a new run
    - This is because there is no need for it to be accessed outside of the gameplay scene
- When the player dies the RunManager passes its game information to the GameManager

ButtonChange:

- The ButtonChange script holds all of the methods each button uses in the game
    - These methods are accessed by the button through the Event Trigger component attached to them

Gameplay scripts:

CameraMovement:

Location: On the Main Camera

Purpose:

- The CameraMovement script is responsible for ensuring the Main Camera follows the player object on the Z axis
- The reason for the Main Camera not being a child of the player object is due to mechanics in the game affecting the player objects stability, which causes the Main Camera to shake

Player:

Location: On the player object

Purpose:

- Controls the movement of the player
- This script is also responsible for implementing the mechanics of 'boss's' and power-ups/interactables that affect the players movement and values
- Responsible for detecting the collision between the player and pick-ups/interactables or enemies

WorldGenerator:

Located: On an empty game object

Purpose:

- Generates all the entities of the running space
    - Platforms
    - Power-ups/Interactables
    - Enemies
- All entities spawned vary on the different level the player is on
- All generation/spawning is random

gameplaySoundManager:

Located: On an empty game object

Purpose:

- Responsible for controlling all of the sounds within the gameplay scene
    - Volume
    - Methods to play the sounds

DeleteAssets:

Located: Main Camera

Purpose:

- Deletes all game objects that collide with the Main Camera's box collider
- Used to optimize the game by deleting game objects, once the player has passed them

GameplayCanvasManager:

Located: Gameplay Canvas

Purpose:

- Fetches information from RunManager script and displays it on canvas objects
    - Score
    - Oxygen
    - Whether the player has a shield or not
    - Shield multiplier

SharkMovement:

Located: On a Shark game object

Purpose:

- Controls shark movement and mechanics

BubbleScript:

Located: On a Bubble game object

Purpose:

- Controls bubble movement

Death Scene scripts:

DeathCanvasManager:

Located: Canvas

Purpose:

- Fetches the values from the players previous run (stored in GameManager) and displays it for the player to see

Options Scene scripts:

OptionsManager:

Located: Canvas

Purpose:

- Sends the volume value to the GameManager
    - Ensures the volume level stays the same throughout all scenes

Game methods

Primary Game Scripts:

GameManager:

- Start()
    - When the script is first running, the Start method instantiates the GameManager and ensures it is only ever created once

RunManager:

- FixedUpdate()
    - This method is used to:
        - Increase the players score-the longer they run for
        - Decrease the players oxygen count
        - Initiate boss mechanics at certain scores within the run
        - Use the DeathControl method when the player is dead
- OxygenControl()
    - If the player has no more oxygen, it sets the player as dead
- DeathControl()
    - When the player is dead, their score is sent and stored in the GameManager script-to be used in another scene\

ButtonChange:

- hoverOverButton() and hoverLeftButton()
    - Used to show and hide Button1 and Button2 when the mouse hovers over it and leaves it
- PlayClick()
    - Changes the scene to the Gameplay scene
- reTryClick()
    - Changes the scene to the Gameplay scene
- BackClick()
    - Changes the scene to the MainMenu scene
- QuitClick()
    - Closes the application
- OptionsClick()
    - Changes the scene to the Options scene

Gameplay scripts:

CameraMovement:

- Update()
    - Moves the main camera at the same speed as the player, on the Z axis

Player:

- Update()

    - This method controls player movement based off player input as well as all of the different mechanics that effect the player:
        - Boss mechanics that directly affect player movement
        - Obstacles/Pick-ups that the player interacts with
        - Increasing forward speed of the player, the longer they run for
- SpeedIncrease()
    - Increases the players forward speed gradually-based on their score
- Boss1Mechanics()
    - Implements level 1's 'boss' mechanics on the player
    - The reason it is stored in the Player script is because it affects the players movement directly
- OnCollisionEnter()
    - Detects collision between the player game object and interactables as well as enemies
        - Depending on what object the player collides with-their respective mechanic/effect is implemented

<u>WorldGenerator:</u>

- Update()
    - This method detects what level the player is on and runs the correct level spawning script (e.g. if the player is currently running in level 1, the Level1Generator is called)
- Level1Generator()
    - Generates the terrain and assets for Level 1

<u>gameplaySoundManager:</u>

- Start()
    - Sets all of the sound components in the gameplay scene to the same volume as the global sound volume within the game (stored in the GameManager)
- muteSound()
    - Sets all of the gameplay sounds to 0
- playSound()
    - Sets all of the gameplay sounds to the global sound volume, stored in the GameManager

<u>GameplayCanvasManager:</u>

- Update()
    - Controls all of the UI in the gameplay scene
        - Displays whether or not the player has a shield and turtle multiplier
        - The players oxygen level
        - The score
- bossText()
    - Displays the warning message for the current boss the player will soon encounter

DeleteAssets:

- OnCollisionEnter()
    - Destroys all game objects that collide with the game object the DeleteAssets script is stored on
        - Collision is detected through box colliders

BubbleScript:

- bubbleMoveUP()
    - Adds force to the game object it is stored on, pushing it upwards
- bubbleMoveDown()
    - Adds force to the game object it is stored on, pushing it downwards

SharkMovement:

- Update()
    - Gradually moves the shark forward towards the player
    - Detects how far the shark is from the player and if close it speeds up the shark

Death Scene Scripts:

DeathCanvasManager:

- Update()
    - Displays the players run information after the score
        - Score
        - Turtle multiplier
        - Final score (Score*Turtle multiplier)
    - Controls the score displaying effect as well
        - E.g. players score gradually increases until it reaches their actual score from the run

Options Scene Script:

OptionsManager:

- Update()
    - Detects whether the player has changed the value on the volume slider, if so then it changes the global sound volume within the game