readme.txt

# Starting Scene File

Assets/Level/Scenes/Intro Scene.unity

The Intro Scene contains the main menu. Multiple scenes are used in play mode. The tutorial scene holds the map/terrain, but we do use other scenes like PlayerCameras & RockScene which hold the player and other in-game objects.

# How to play

### Mouse/Keyboard:

To move: Use “WASD” keys, or your directional keys

To look around: Use your mouse

To aim: Hold “*Right Click*”

To attack: “Left Click”

To activate Timespell: Press “Q”

To activate Jump Spell: Press “spacebar”

To activate Flare Spell: Press “R”

To increase camera sensitivity: Press “. ” (Period)

To decrease camera sensitivity: Press “, ” (Comma)

### Controller: (Playstation controller)

To move: Use “Left Joystick”

To look around: Use “right Joystick”

To aim: Hold “Left Bumper” (R2 on PS5)

To attack: Press “West Button” (Square on PS5)

To activate Flare Spell: Press “Right Trigger + North Button” (R2 + Triangle)”

To activate Timespell: Press “Left Trigger + East Button” (R2 + Circle)

To activate Jump Spell: Press “Right Trigger + South Button” (R2 + X)”

### Technology requirements

* **Objective**: The player can win by using the flare spell after collecting the four books required and activating it when a ship is in the green
* **Failure**: If the player dies or quits without successfully launching the flare spell
* **AI-controlled agents**: Wolves are implemented through a combination of the Anim system, a state machine and a nav mesh agent. The state machine controls the destination of the navy mesh agent and the wolf behavior. At the same time, it provides constant feedback to the animation controller. The controller uses this feedback to dynamically blend between the different animation layers, much like the player controller. Rabbits also use the navmesh system to pick random destinations, and the anim system to drive the animation states. Both both and rabbits react to the player’s distance to them; with wolves proceeding to chase and attack, and rabbits to run away.
* **3D Character (Player)**: Humanoid rig with skeletal animation implemented through Unity’s Anim system. Contains various animation layers to support locomotion and combat. Uses blending for locomotion animations during both walking, running, sprinting, and aiming states. Movement and combat use the new Input System, which is event-driven.
* **Camera**: 3 separate cameras using Unity’s Cinemachine system, with blending between them. One camera is used as de default player camera. Another camera is used for the aiming system. Lastly, a third camera is used as a cinematic camera to provide a clear view of the flare as it launches into the air. Both Aiming and Player cameras are controlled via the new Input System, and support both mouse and controller inputs.
* **Animation:** Both player and wolfs use animation layers, sub-state machines and blending. Animation layers are used to differentiate locomotion and combat animations, while sub-state machines are used to encapsulate combination of animations that must happen in sequence. The player jump is such an example of a sub-state machine, where there are 3 separate animations for preparing to jump, hovering mid air, and falling.
* **Audio:** The game uses a combination of spatial audio, event-driven audio, and background music. When the player is near the beach, they will hear seagulls and the sound of waves. As they enter the forest, they will start to hear birds and different critters. Footsteps can also be heard as the player walks, as well as the sound of rocks as they fall from a cliff.
* **UI:** The player can pause the game at any time. Upon death, they have the option to restart the game, or go back to the main menu. Additionally, when the player picks up a new spell, or a flare fragment, they will receive a notification with instructions on how to them.

# Known problem areas

* The player can momentarily get stuck on a tree when landing from a high jump
* Jump is difficult to control as the camera does go down enough to view the ground.
* There is no indicator of how many flares are left as the player picks them.
* The animation blend between running and sprinting is not entirely seamless
* Wolves can push the player around, making it difficult to move if they warm them.
* Player can sprint while jumping.
* The player’s body can clip through the terrain on the death animation.
* Camera might behave unexpectedly in certain extreme angles.

# Manifest

1. Kiran: Worked on enemy wolf prefab (WolfPrefabAudio) that will aggro on the player

when they come within a certain range and attack. Worked on the animator for this and

the WolfController.cs code which manages the actions and allows the wolf to take

damage and play a barking sound. The wolf will stop barking/chasing the player if they

player gets out of range. Also worked on the prefab for the health, energy, and mana

bars (HealthEnergyManaBarsPrefab) to be displayed on the UI. Wrote the code for these

to make it so that the energy slowly decreases over time and once it reaches zero, the

health starts to decrease slowly as well. Mana slowly replenishes over time. Added ability for player to control camera sensitivity with “,” and “.” keys. Fixed terrain near wolf pack area so as not to allow player to exploit it.

2. Lada: Worked on player attack prototype that uses projectile prefabs and Projectile script when cast that disappear after some distance. Worked on final flare spell that needs to have all of it’s pieces collected first to work (FlarePiece.cs +PlayerController). It can also only be cast during the time period the ship is in reach of the flare spell from the island. That is indicated by the bar (part of the UI prefab) that shows the location of the ship related to “in reach” area (ShipBar.cs + GameTimeManager.cs). Wrote the code for it so that it only allows the player to cast the spell when all pieces are collected and ship is within green area of the UI, shows win text when successfully cast and plays animation. Added animations and character input for the spells. Modified the ship to take longer for a full circle and fixed the bug to not allow casting it when ship is not in reach. Worked on the jumping spell puzzle, modified the rock assets to make into steps, but didn’t add them as a prefab. Also fixed issue of being able to walk on the mountain around the rocks. Helped with fixing the bugs discovered in playtesting stage, and level design, placing items around the world. Worked on the trailer.

3. Quan: Worked on the Time spell that stops objects around the player.

Worked on the Maze Puzzle, the player needs to find the path through the maze to get a reward. The game over screen that displays on player death. The game manager that controls states of the game including player death. The spell tip that pops up on spell pickup.

4. Alejandro: All aspects of player control. This includes the entire animation anim system, for blending between locomotion and combat, and the different animation layers. Set up animations for walking, running and sprinting, as well as the attack sequence, jump spell, time spell, and flare. Added and configured two virtual cinemachine cameras for regular player follow and aim, and coded the logic for controlling switching between them. To support all controls, implemented the new Input System, which allows for easy setup for both control and mouse/keyboard gameplay. Additionally, created the player Interface that allows for easy interaction with any of the player stats.

Designed and coded the Spell System, which is supported by the Event Manager taken from the Milestone projects. The system allows for synchronized animation events for firing spells, and includes scriptable objects to easily create new spells. Additionally supports spell stats, such as cooldowns, and controls how often a player can fire them.

Designed the event-driven Pickup System, which allows for simultaneous updates of game logic and UI; and is used across several components, ranging from Wolf AI and Spell Manager to Game Controller.

Designed a new revamped Wolf AI, to include a state machine for controlling the behavior, as well as a new Anim system with animation layers and animation blending; both of which work in tandem to provide a cohesive visual behavior, and also integrate with the rest of the systems. Also created scriptable objects to store enemy configuration and allow for variety without coding implementation.

Created multiple events for different aspects of the game, this includes a player damage event, a spell pickup event, a food pickup event, player death.

Revamped the Game Manager system to support the new event-driven architecture. The manager controls the energy over time, as well as the mana consumption and regeneration depending if the player is sprinting or not; both of which interact with the Player Interface.

Designed and modeled the initial island area of the game. Worked on the trees and grass, and added a water shader to simulate the island setting. Added 3d audio across the initial terrain, to simulate waves, seagulls, and forest birds.

Added palm trees from the Unity asset store, and configured shadow rendering and their shaders. Also configured the shaders for the rest of the trees to work properly with URP.

Created the flare spell cast sequence, which includes the animation, particle effects, lighting, camera blending, and flare model. There is a third camera that is used to follow the flare as it flies through the fair.

Created the death sequence, where the death animation plays and the game over screen is synchronized with the end of it.

Added new models from sketchfab for the different spells and flare pickups, including point lighting to make them stand out from the environment. Additionally, added the Sketchfab extension from the unity store so that others could grab any models and import them.

Added post-processing to the main camera as well as the player and aim cameras. The aim camera contains a chromatic aberration effect and motion blur.

Designed and created the UI Manager, which handles all interactions with the player UI; and is state driven. Modified the tooltips when picking up spells to appear and disappear to the side of the screen in order to avoid description of the gameplay flow.

Created the intro scene, pause menu, and final game scene.

Revamped the time spell to support pausing of any game object, as well as return them to their original state after the effect wears out.

Created the 3d model for the basic attack.

Created prefabs for the food pickups, using Scriptable objects

5. Mini: Worked on building the terrain and environment details using the Unity Terrain tool and terrain assets. Worked on incorporating a skybox to the environment. Worked on the falling rocks path prefab which uses the file FallingRock.cs which utilizes complex

colliders which will trigger rocks to fall as the player enters and deal damage to the

player’s health. Worked on defensive AI enemy which is represented by the rabbit.

The navMesh and using the navMesh agent to run away from the player as he approaches and would lead player to an objective. The code can be found

in Rabbit.cs script file.

### Assets:

1. Kiran:

* Assets/Prefabs/WolfPrefabAudio
* Assets/Prefabs/HealthEnergyManaBarsPrefab
* Assets/Sounds/0006AF\_M\_24\_Dog Bark One
* Assets/WolfAggro/Scripts/WolfController.cs
* Assets/WolfAggro/Scripts/WolfAggro.cs
* Assets/Scripts/UI/HealthBar.cs and HealthBarTest.cs
* Assets/Scripts/UI/EnergyBar.cs and EnergyBarTest.cs
* Assets/Scripts/UI/ManaBar.cs and ManaBarTest.cs

2. Lada:

* Assets/Prefabs/Projectile
* Assets/Prefabs/UIBarsPrefab (Assets/Prefabs/Ship UI for separate view)
* Assets/Sprites/sliderShip
* Assets/Sprites/ship
* Assets/Prefabs/Flare
* Assets/Animations/Player/JumpSpell.fbx
* Assets/Animations/Player/FlareSpell.fbx
* Assets/Scripts/CharacterControl/FlarePiece.cs
* Assets/Scripts/CharacterControl/Projectile.cs
* Assets/Scripts/UI/ShipBar.cs
* Assets/Scripts/UI/GameTimeManager.cs
* RockScene

3. Quan:

* Assets/Animations/maze
* Assets/Prefabs/Maze Prototype/Maze
* Assets/Scripts/Utility/MovingMaze.cs
* Assets/Scripts/CharacterControl/TimeSp.cs
* Assets/Scripts/Managers/GameManagerScript.cs
* Assets/Level/Prefabs/UI/Spell Tip Window.prefab
* Assets/Level/Prefabs/UI/GameOverScreen.prefab

4. Alejandro:

* Assets/Animations/Player/Idle.fbx
* Assets/Animations/Player/Run.fbx
* Assets/Animations/Player/Walk.fbx
* Assets/Animations/Player/Standing 1H Magic Attack 01
* Assets/Level/Prefabs/Pickups/FlarePickup.prefab
* Assets/Level/Prefabs/Pickups/FoodPickup.prefab
* Assets/Level/Prefabs/Pickups/FoodPickupXL.prefab
* Assets/Level/Prefabs/Pickups/JumpSpellPickup.prefab
* Assets/Level/Prefabs/Pickups/TimeSpellPickup.prefab
* Assets/Level/Prefabs/Projectiles/ProjectilePrefab.prefab
* Assets/Level/Prefabs/Player.prefab
* Assets/Level/Prefabs/PlayerCamera.prefab
* Assets/Level/Prefabs/AimCamera.prefab
* Assets/Level/Prefabs/AimCanvas.prefab
* Assets/Level/Prefabs/Main Camera.prefab
* Assets/Level/Prefabs/Controllers/UI Manager.prefab
* Assets/Level/Prefabs/Controllers/EventManager.prefab
* Assets/Level/Prefabs/Controllers/AudioManager.prefab
* Assets/Level/Prefabs/PauseMenu.prefab
* Assets/Level/Prefabs/Projectiles/Flare.prefab
* Assets/Level/Scenes/Intro Scene.unity
* Assets/Level/Scenes/Tutorial Scene.unity
* Assets/Level/Scenes/PlayerCameras.unity
* Assets/Scripts/CharacterControl/AimController.cs
* Assets/Scripts/CharacterControl/PlayerController.cs
* Assets/Scripts/CharacterControl/PlayerInput.cs
* Assets/Scripts/CharacterControl/TimeSpell.cs
* Assets/Scripts/CharacterControl/Projectile.cs
* Assets/Scripts/EnemyControl/EnemyInterface.cs
* Assets/Scripts/Events/PlayerDamageEvent.cs
* Assets/Scripts/EnemyControl/EnemyController.cs
* Assets/Scripts/CharacterControl/PlayerInterface.cs
* Assets/Scripts/Events/FoodPickupEvent.cs
* Assets/Scripts/Events/PlayerDeathEvent.cs
* Assets/Scripts/Events/PlayerWinEvent.cs
* Assets/Scripts/Events/SpellCastEvent.cs
* Assets/Scripts/Events/SpellPickUpEvent.cs
* Assets/Scripts/Managers/SpellManager.cs
* Assets/Scripts/Managers/UIManager.cs
* Assets/Scripts/Scriptable Objects/EnemyScriptableObject.cs
* Assets/Scripts/Scriptable Objects/SpellScriptableObject.cs
* Assets/Scripts/Utility/FlareController.cs
* Assets/Scripts/Utility/SpellPickup.cs
* Assets/Scripts/Utility/FoodPickup.cs
* Assets/Scripts/Spells.cs
* Assets/Scripts/Wolf/WolfBehaviour.cs
* Assets/Scripts/Scriptable Objects/FlareSpell.asset
* Assets/Scripts/Scriptable Objects/JumpSpell.asset
* Assets/Scripts/Scriptable Objects/TimeSpell.asset
* Assets/Scripts/Scriptable Objects/Enemies/Wolf\_lvl1.asset
* Assets/Scripts/Scriptable Objects/Enemies/Wolf\_lvl2.asset
* Assets/Scripts/Scriptable Objects/Enemies/Wolf\_lvl3.asset
* Assets/Art/Animations/AnimationControllers/PlayerAnimController.controller
* Assets/Art/Animations/Player/death\_anim.fbx
* Assets/Art/Animations/Player/JumpSpell.fbx
* Assets/Art/Animations/Player/Aiming/idle\_strafe.fbx
* Assets/Art/Animations/Player/Aiming/walking\_back\_strafe.fbx
* Assets/Art/Animations/Player/Aiming/walking\_forward\_strafe.fbx
* Assets/Art/Animations/Player/Aiming/walking\_left\_strafe.fbx
* Assets/Art/Animations/Player/Aiming/walking\_right\_strafe.fbx
* Assets/Art/Animations/Player/Locomotion/Idle.fbx
* Assets/Art/Animations/Player/Locomotion/Run.fbx
* Assets/Art/Animations/Player/Locomotion/Sprint.fbx
* Assets/Art/Animations/Player/Locomotion/Walk.fbx
* Assets/Art/Animations/Player/Spells/basic\_attack.fbx
* Assets/Art/Animations/Player/Spells/basic\_attack\_reverse.fbx
* Assets/Art/Animations/Player/Spells/time\_spell.fbx
* Assets/Art/Animations/Player/Spells/power\_jump.fbx
* Assets/Art/Animations/Player/Spells/fall\_to\_land.fbx
* Assets/Art/Animations/Player/Spells/flare\_spell.fbx
* Assets/Art/Animations/Player/Spells/hover\_mid\_air.fbx
* Assets/Art/Animations/AnimationControllers/Wolf.controller
* Assets/Art/Models/food\_model\_meshes
* Assets/Art/Models/flare\_meshes
* Assets/Art/Models/flare\_book\_meshes
* Assets/Art/Textures/flare\_book\_textures
* Assets/Art/Textures/flare\_textures
* Assets/Art/Textures/food\_model\_textures
* Assets/Art/Textures/player
* Assets/Art/Textures/spellbook
* Assets/Art/Textures/UI
* Assets/Audio/Sound/beach\_ambience.aiff
* Assets/Audio/Sound/Footstep.mp3
* Assets/Audio/Sound/Forest\_Ambience.mp3
* Assets/Audio/Sound/seagulls.wav
* Assets/ThirdParty/FlexibleCelShader-SRP-main
* Assets/ThirdParty/Lean GUI
* Assets/ThirdParty/Simple Water Shader
* Assets/ThirdParty/Trees/Coconut Palm Tree Pack

5. Mini:

* Assets/ New Terrain 2.asset
* Assets/Rabbits/Materials/Rabbit Eyes.mat
* Assets/Rabbits/Animations/Rabbit@Run.fbx
* Assets/Prefabs/Defensive Enemy.prefab
* Assets/Prefabs/Falling Rocks Path.prefab
* Assets/Prefabs/FallingRock.prefab
* Assets/Prefabs/FallingRock2.prefab
* Assets/Sound/rock-destroy-6409.mp3
* Assets/Fantasy Skybox
* FREE/Cubemaps/FS017/FS017\_Day\_Sunless\_Cubemap.mat
* Assets/FallingRock.cs
* Assets/Rabbit.cs

### Script files:

1. FallingRock.cs (Mini)
2. UIManager.cs (Alejandro)
3. Projectile.cs (Alejandro)
4. PlayerInterface.cs (Alejandro)
5. AimController.cs (Alejandro)
6. FoodPickupEvent.cs (Alejandro)
7. PlayerDeathEvent.cs (Alejandro)
8. PlayerWinEvent.cs (Alejandro)
9. EnemyInterface.cs (Alejandro)
10. SpellCastEvent.cs (Alejandro)
11. SpellManager.cs (Alejandro)
12. EnemyScriptableObject.cs (Alejandro)
13. SpellScriptableObject.cs (Alejandro)
14. SpellPickUpEvent.cs (Alejandro)
15. FlareController.cs (Alejandro)
16. Spells.cs (Alejandro)
17. WolfBehaviour.cs (Alejandro)
18. SpellPickup.cs (Alejandro)
19. FoodPickup.cs (Alejandro)
20. PlayerDamageEvent.cs (Alejandro)
21. EnemyController.cs (Alejandro)
22. Rabbit.cs (Mini)
23. FlarePiece.cs (Lada)
24. ShipBar.cs (Lada)
25. GameTimeManager.cs (Lada)
26. PlayerController.cs (everyone)
27. WolfController.cs (Kiran)
28. WolfAggro.cs (Kiran)
29. HealthBar.cs/HealthBarTest.cs (Kiran)
30. EnergyBar.cs/EnergyBarTest.cs (Kiran)
31. ManaBar.cs/ManaBarTest.cs (Kiran)
32. MovingMaze.cs (Quan)
33. TimeSpell.cs (Quan, Alejandro)
34. GameManagerScript.cs (Quan)

Credits:

* Wolf model/animations by Dzen Games (<https://assetstore.unity.com/publishers/14408>) from <https://assetstore.unity.com/packages/3d/characters/animals/wolf-animated-45505>
* Wolf sound by Porovaara (<https://assetstore.unity.com/publishers/12675>) from <https://assetstore.unity.com/packages/audio/sound-fx/animals/dog-barks-1-0-35370>
* bar.png sprite used for UI bars from <https://github.com/Brackeys/Health-Bar/tree/master/Health%20Bar/Assets/Sprites>
* Ship png icon used for handle sprite on ship location bar <https://www.flaticon.com/free-icon/cargo-ship_870107>
* Footstep sounds and background: <https://www.fesliyanstudios.com/>
* Lean UI (UI Framework/Library): <https://assetstore.unity.com/packages/tools/gui/lean-gui-72138>
* Model Animations: <https://www.mixamo.com/#/>
* Reticle UI: <a href="https://www.flaticon.com/free-icons/miscellaneous" title="miscellaneous icons">Miscellaneous icons created by Barudaklier - Flaticon</a>
* Spellbook: "Ornate Book" (https://skfb.ly/REER) by N8 is licensed under Creative Commons Attribution (<http://creativecommons.org/licenses/by/4.0/>).
* Meat Pickup: "Meat" (https://skfb.ly/UrRv) by Xillute | Dev is licensed under Creative Commons Attribution (<http://creativecommons.org/licenses/by/4.0/>).
* Flare spell book: "Paladin's book" (https://skfb.ly/6sN7z) by AlexChuchvaga is licensed under Creative Commons Attribution-NonCommercial (<http://creativecommons.org/licenses/by-nc/4.0/>).
* "Flare" (https://skfb.ly/6zvIU) by lionclaw0612 is licensed under Creative Commons Attribution (<http://creativecommons.org/licenses/by/4.0/>).
* Beach Sounds: <https://freesound.org/s/18363/>
* Forest sounds: <https://opengameart.org/content/forest-ambience>
* Seagull Sounds: <https://freesound.org/people/juskiddink/sounds/98479/>
* Flare Launch: <https://freesound.org/people/kyles/sounds/637537/>

Spoiler Walkthrough

Go to the left, where the group bunnies are. You will find the Time Stop spell there. Sprint forward and then make a left into the wolf’s den.

Use the Time Spell to stop the entire pack, and pick up the Jump Spell on the ground. Use the spell to pick up the red Flare that’s right next to it.

Exit through the back of the den, towards the initial hill as seen at the beginning of the game. At the top of the hill, there is another Flare Spell. Use the jump spell again to pick up that flare.

Go back to the wolf’s den, and right in front of it, there will be a group of 3 bunnies. Use the jump spell to get into that area.

Use sprint to run pass the rocks, and immediately use the Time Stop spell to prevent the rocks from falling down and killing you.

Continue walking forward, until you find the entrance of the maze, there will be a red Flare at the end of the maze.

After you exit the maze, go back to where you started, and you should see rock platforms on the side of a mountain. Use the platforms to climb the mountain (using the jump spell). The last red Flare pickup will be at the top.

Wait until the ship icon is in the green section of the ship bar, and then cast the flare spell to finish the game.