**Table of Contents**

1. **Player**
   1. **Player Attacks**
      1. **FinisherAreaController**
      2. **FinisherMode**
      3. **RunicBeamBlast**
      4. **PlayerSwordAttack**
      5. **PlayerSwordHit**
   2. **Camera**
      1. **CameraCollider**
      2. **CameraFollow**
      3. **CameraMovementController**
   3. **PlayerHealthController**
   4. **PlayerMovementController**
   5. **PlayerTriggerCollider**
   6. **PlayerUpdater**
2. **Enemies**
   1. **AI**
      1. **EnemyAI**
      2. **EnemyGroup**
      3. **GroupDirector**
   2. **Enemyhp**
   3. **EnemyMovementController**
   4. **EnemySword**
   5. **EnemySwordRange**
3. **UIScripts**

**Temp Scripts: Throw Limb(throw blue mask) Throw Part(cut in half physics)**

**Player**

**PlayerAttacks**

**FinisherAreaController – FinisherArea**

When you enter finisher mode, push nearby enemies away, or make them pause

**FinisherMode - Player**

Checks if finisher meter is full when you press F, then grabs closest enemy ( need to change to grab the guy your looking at and is near, with a popup), then enter finisher mode. Then starts up some coroutines to enter the finsher mode frezes the enemy, and changes variables to get you into the mode. After that update handles the next user input, once the user inputs a combo, and confirms with rt later, it goes to Execute finisher coroutine, which exectus the attack, like cutting the body in half. It also calls FailFinisherMode if you fail to execute with a period of time, which lets the enemy go.

**RunicBeamBlast – RunicBlastBeamObject prefab**

Just kills enemies it touches and destroys itself after a set time

**PlayerSwordAttack – Sword Arch**

Handles player inputs that will be light and heavy attacks, right now just enables the sword object and rotates around the player to simulate a sword attack.

**PlayerSwordHit - Sword**

Damages enemies on hit

**Camera**

**CameraCollider – MainCamera**

Helps make the camera act as a dolly, to push camera inward if about to collide with a wall. **MORE TESTING AND REFINING NEEDED.**

**CameraFollow - CameraBase**

**Takes mouse or controller input and moves the camera centered around and following the palyer**

**CameraMovementController - CameraTargetPos**

Moves the camera parent object from OOC position, InCombatPosition, or FinisherCameraLocation

**PlayerHealthController - PlayerModel**

Here you can set the max health of the player and it will decrease this value when PlayerHit is called. It also updates the health slider on the UI and will be in charge of restarting from last checkpoint when the health is zero. **NOTE: we need to add a small timer here to prevent the player from being damage too often from simultaneous hits, or at least make sure an enemies weapon cannot deal damage too often. i.e. get hit, and somehow walk into the weapon again to activate trigger during the same attack animation, reset when that animation section is completed.**

**PlayerMovementController - Player**

The wheel of our car. Allows the player to walk, **RUN?**, jump, dash, and get knocked back or prevented from moving or turning. There is some complicated stuff that allows movement while airborne, but restricts certain actions unless grounded. It is nearly ready to add fall damage if we want that. It correctly prevents dashing while out of combat, and prevents jumping while in combat. **NOTE: we may need a similar knockback cooldown, so you aren’t getting batted around indefinitely.**

**PlayerTriggerCollider - PlayerModel**

Simply starts player knockback when it detects it was hit by a sword, this is done here because it is buggy when activated by the enemy for currently unknow reasons.

**PlayerUpdater - Player**

Currently used to enter and exit combat state, I also updates and Immune Counter and Poise Counter used by PlayerHealth and Movement Controllers to prevent being hurt or knocked back in a few seconds. It also may be used for other useful things in the future.

**Enemies**

**EnemyAI - Enemy**

EnemyAI allows each enemy to handle itself, its given a role from the director via Change Status, and then the enemy detects that its updated status is different in update, and tries to switch the current status to that one, then performs the actions associated with that state

**EnemyGroup - Class**

Used by director to send orders to the enemies, and like wake up enemies, setAllToWait, or SetAllToAttack. Things in the future will focus setting a Primary Attacker, ArcRunners, Surrounders, SpecialRoles, etc.

**GroupDirector - GroupDirector**

Initialize the list of enemies, and formulate the orders for the group over the course of the fight. End the encounter when all the enemies are dead.

**Enemyhp – Enemy**

Controls enemy health, player uses it to deal damage to the enemy, and Calls KillEnemy in EnemyAI when the health reaches 0

**EnemyMovementController – Enemy**

Given a target, it pathfinder towards it, updating every frame. Also has a knockback method to move stun enemy away from player, may need to add an additional pause period after the knockback.

**EnemySword – EnemySwordAttack**

Given a damage value and duration, it damages the player on trigger, player knocks itself back for now.

**EnemySwordRange - EnemySwordRange**

Sets the EnemySwordAttack object to active on trigger.