Prism: Quarter 2

# Testing:

Features required experimentation by creating scenes , updating latest prefabs, and ultimately merging everything together into a working scene for the midterm demonstration.   
  
Documentation:

For task organization, our team engaged in a brainstorming session to on create a sprint on Jira consisting of how the player would interact with enemies inside a room. The number of features grew to 34 as we added more features to showcase during the midterm.

My contribution to the poster is on the features I incorporated into the game, including the player’s dodge and bow, UI color meters and icons, enemy attacking player, and items such as color drops and weapons.

We collectively discussed how to approach features that require working with the same prefabs for better understanding of functionality, prevent complications during merging, and avoid interfering with each other’s progress.

# Jira Sprints: Player Fights Enemies

I was responsible for addressing 14 issues involving implementing sounds, animations, and mechanics in the project.

Graphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, application

Description automatically generated

# Features:

## Task: Sounds

### Sound for UI icon change & Sound for UI bars

Incorporated audio elements into the UI for when changing weapons and increasing color amount in the color bars. I configured the audio listener in the camera and modified ColorStats.cs in the canvas to accommodate four audio sources: three for the color bars and one for changing the weapon icons.

***Link:*** <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/08d548d1bb182dc56853dd8ee646d0cb0ac62a2e#diff-4021b4b015820ca7ee9678d996502f706a6cdea11a3af624eada9e54776e6416>

Graphical user interface

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

### Player Dash Sound

Implemented an audio for the player’s dash mechanic, which involved modifying PlayerMovement.cs script to call and play the sound. The configuration for the player only required the addition of an audio source for the dash sound.

Graphical user interface

Description automatically generatedGraphical user interface

Description automatically generatedLink: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/f33e8f266b01a4d5afe680acf4095faedfa5c5a9#diff-e4a119584c97350c5432078f4ca08d355511b07ed01c0c00479b9e3ae7798c51>

## Stories Animations:

### Enemy Animation & Yellow Enemy Animation

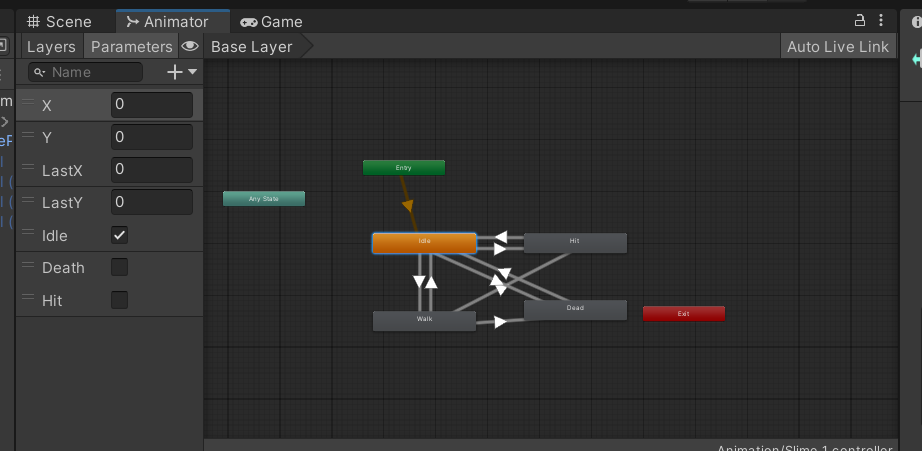
Created an animator for the red enemy’s movement and yellow enemy’s position relative to the player. Modified EnemyStats.cs to take into account both the current and previous locations of the enemy. The animator should choose the best animation with that data.

Link for idle and movement: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/414f8122b81fefb04a2f8a60132024ad64d14ac7#diff-3b9c7172ffac81e9502272fdeec8ef6642b0d0cfbb87e7be12a3f18efc72801a>

Link for hit and death animation: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/f20a25680d109d5754a046f71bc7a8567604ed28#diff-3b9c7172ffac81e9502272fdeec8ef6642b0d0cfbb87e7be12a3f18efc72801a>

*A screenshot of a video game

Description automatically generatedBase Layer of Red Enemy Animator: Idle, Hit, Walk, Dead*.

*Base Layer of Yellow Enemy Animator: Idle, Hit, Walk, Dead.*

### Bow Animation

Changed the color of bow and arrow’s aura when yellow color meter is above zero. I also modified ArrowScript.cs to activate condition when hitting a target, and it retrieves information from ColorStats.cs that checks if bow and arrow have color.

Link: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/979f720ba56c4337130a2693b6ea1866a90b76f7#diff-e82a24e9a84c0275c844455adf4e48bfbf57dbaa0ef8f10bc8cf8ae10037d258>

*A picture containing graphical user interface

Description automatically generatedArrow animation No Glow, Glow, Fly, Hit*

Stories Features:

### Player Dodge Attacks

In PlayerMovement.cs the dodge mechanic teleports player a certain distance and leaves a trail behind. To ensure that the player does not teleport past walls or other objects with colliders, the script checks for collision before executing the dodge. The player itself has a trail renderer component that creates the trail when teleporting.

Link: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/07062bbfe72d88db290676ef0cfb9968a7abfd0b#diff-e4a119584c97350c5432078f4ca08d355511b07ed01c0c00479b9e3ae7798c51>

Graphical user interface

Description automatically generated

### Enemy Moves Straight to Player (For Testing)

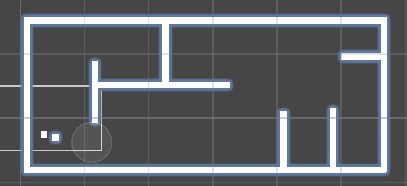
Created an enemy gameobject and implemented the EnemyMovement.cs to enable it to move towards player’s location at a normalized speed.

Link: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/1a63515df112e6eb5c0c23215fe80b003e4ff09b#diff-c074dfd35c7d07f8cb58562efcc1cae71b0b4724aa0e6ef68217293b6a8751d9>

### Spawn Enemy (For Testing)

EnemySpawn.cs script facilitates the spawning of random amount of enemies withing a given sized room from a random range of positions. To reuse the enemies once they are defeated, I implemented an enemy repooling system within the script This was to engage battling multiple enemies simultaneously.

Link: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/9e6ba853379393334b35ef0fd365dd190df1da4c#diff-71d497013960bd0ea7d7f079c40e981c9919d7cb67be09edb7d9461d8515c297>



### Enemy Stats

The script EnemyStats.cs has the health system for the enemies and can damage player or receive damage from player. Consists of ids to differentiate enemy’s color. First made to interact with player and updated with specific information based on the enemy.

Link of when file was first made: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/2378fd2ad24b27c35ebcad15a9422f253326c2e8#diff-3b9c7172ffac81e9502272fdeec8ef6642b0d0cfbb87e7be12a3f18efc72801a>

Link of current file: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/feea86df0593c4c81eb6f827469f73384f80aee0#diff-3b9c7172ffac81e9502272fdeec8ef6642b0d0cfbb87e7be12a3f18efc72801a>

### Enemy Attacks

Used an ontrigger collider for enemy to attack player when player’s collider is detected. Used EnemyStats.cs to determine when to attack and CloseAttack.cs to define how enemy should attack. Red slime enemy stops moving, jumps towards player, and returns to its original location. For awareness it uses a collider to detect the player and a raycast to determine if the player is visible to execute attack.

Link for enemy attack made: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/4d773e37673f483b141e1c49b75dcc54396601a3#diff-1d5cbd27e5d697c51833e1c6003cc728ff78f5fd30c9de8bb277544d0c4a221c>

Link for most updated enemy attack (Midterm Scene): <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/9813c8083f46eb02ab8668454136b36faf10b540#diff-1d5cbd27e5d697c51833e1c6003cc728ff78f5fd30c9de8bb277544d0c4a221c>

*Diagram

Description automatically generatedOuter circle is awareness of player.*

*Inner circle is enemy attack range.*

### Player Arrow Attack Damage

To optimize arrow usage, I created a pool of arrows called ArrowPool. Automatically re-pools after a certain time or after hitting a wall, object, or enemy. Additionally, ArrowScript.cs uses an ontrigger collider allowing the arrow to detect an enemy and parents whatever it hit.

Link: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/979f720ba56c4337130a2693b6ea1866a90b76f7#diff-e82a24e9a84c0275c844455adf4e48bfbf57dbaa0ef8f10bc8cf8ae10037d258>

Text

Description automatically generated

Graphical user interface

Description automatically generated

### Yellow Enemy Attack

The yellow slime enemy functions similarly to the bow and arrow, using a gameobject to fire from and rotates towards the player. In addition, it utilizes a raycast to detect the player’s location. The slime ball projectile is fired towards the player but can still hit walls and objects. Once fired, the bullets automatically re-pool back to enemy after a certain time or after popping. EnemyRangeAttack.cs fires the bullets while SlimeBallBullet.cs manages the rotation, hitting, and re-pooling functionalities.

Link: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/013b452aca961d62e7bf84a7a1eb7918ddf8221c#diff-60bfd692afe84db4bfa3b7732532976d4e4a3c813d523ea8bf26b07741622113>

Graphical user interface

Description automatically generated

### Enemy Drop Items

Created pools of energy that the player can pick up with unique IDs in PickUp.cs. When an enemy’s health reaches 0, EnemyStats.cs gives it a random chance to drop health, matching color energy, or nothing. The items are then re-pooled back into their respective pools when collected.

Link: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/0809758e0c7b285d876ccdb54f504d77d5c9dff9#diff-3b9c7172ffac81e9502272fdeec8ef6642b0d0cfbb87e7be12a3f18efc72801a>

Graphical user interface, application

Description automatically generated

# Scene for Demo

Midterm Demo

We combined all the features we previously worked on by updating prefabs and merging them together. My contribution involved integrating and fixing errors in features I worked on, as well as assisting with any errors in Map integration.

Link for branch: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commits/MidtermScene>

Link for errors fixed before Midterm: <https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/96d962cfb3f96e8bc3c3f2597f4660e4928a0bbc>

Graphical user interface

Description automatically generated