Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

Prism: Final Report

Jira Sprint: Issues, Menu, and Boss

Features: Jonathan Rodriguez

Task: Fixing Demo Issues

Balance Attacks

Decreased amount of damage the enemy does for both red slime and yellow slime balls. EnemyStats.cs deals with red damage while SlimeBallBullet.cs deals with yellow slime ball damage.

Link (EnemyStats.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/e6ef455dc4075097601336e62cdf5f58db9a9dbe#diff-3b9c7172ffac81e9502272fdeec8ef6642b0d0cfbb87e7be12a3f18efc72801a

```
healthbar.hit(10);
healthbar.hit(2);
```

Link (SlimeBallBullet.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/9813c8083f46eb02ab8668454136b36faf10b540#diff-fd573ad25b98cce0c0aafeace2a6c1f7d7c05c940394e9ccd8f8df4e7caf43ec

```
pHealth.hit(4);
```

Assign attacks to 123

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

Switched from keys Q and E to switch left or right to 1,2,3. This was done in SwapBars.cs by just switching left or right depending on the current state and key chosen instead of rewriting ColorStats.cs.

Link (SwapBars.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/e6ef455dc4075097601336e62cdf5f58db9a9dbe#diff-3843213c7d1fba4dec17bf1d366fb9c05ea4bf84b81645d16ea4dec829dc5523

```
private void changeWeapon(int num1, int num2){
    if(stats.changeState == num1){
            stats.colorSwapFoward(true);
        }
        else if(stats.changeState == num2){
            stats.colorSwapFoward(false);
        }
        //Debug.Log("space key was pressed");
        selected();
}
```

Fix dash from clipping through walls

With the help of a YouTube tutorial I used a raycast and layermask with walls and objects to position the player right before touching a wall or object in PlayerMovement.cs.

Link (PlayerMovement.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/e6ef455dc4075097601336e62cdf5f58db9a9dbe#diff-e4a119584c97350c5432078f4ca08d355511b07ed01c0c00479b9e3ae7798c51

Youtube

```
RaycastHit2D raycastHit2d = Physics2D.Raycast(transform.position, moveDir, dashAmount, dashLayerMask);
if (raycastHit2d.collider != null) {
    dashPosition = raycastHit2d.point;
}
```

PlayerMovement.cs

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

```
+           RaycastHit2D rc = Physics2D.Raycast(transform.position+ new Vector3(0,.25f,0), dir*teleport, teleport, dl);
+
            if (rc.collider != null){
-               dp = rc.point;
-               dashCooldown = 0;
+               dp = rc.point - (Vector2)dir/2;
```

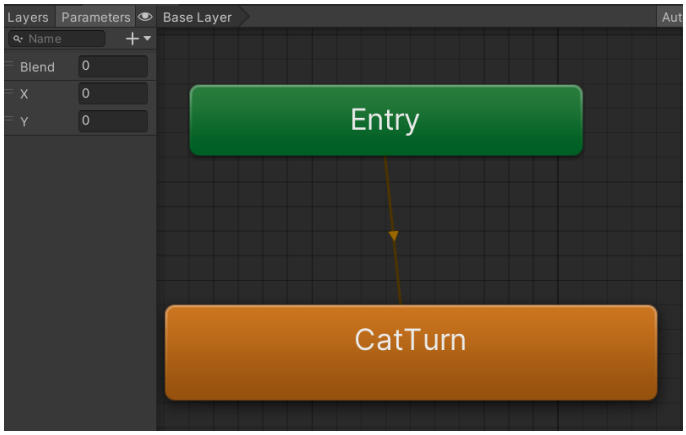
Task: Animations

Yellow Boss Animation

Animations are done using an animator and parameters. There are 5 different animations for the boss these include; crystal balls, lightning attacks, cat boss, teleport, and when boss it hit or dead.

Cat boss turns with BossTurning.cs script. It detects the x and y values using its previous location for direction.

Link (BossTurning.cs):https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/291a08491d86d21f8090f960b70ec0b0d94bb176#diff-3d85b47f59929162dc43a1f1ac6496d0af7dac4f03e3101ee6e3129351f3a1f3



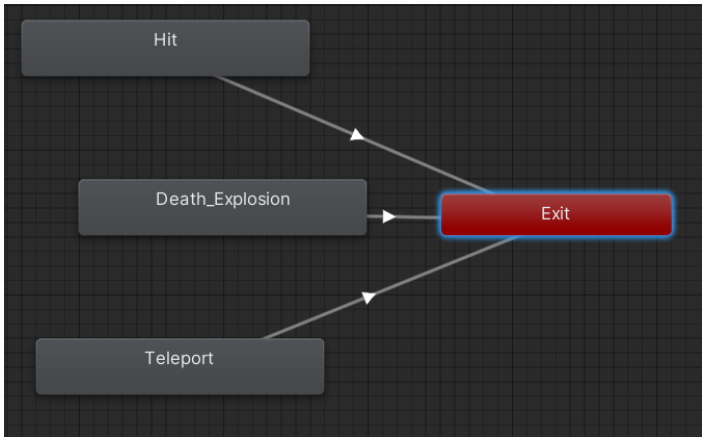
The boss has a parent with explosion animations when hit in BossHitAnimation.cs script.

Link (BossHitAnimation.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/aa9fb9f6108f293d5a49d162e5ee05a6f4bf5f7a#diff-36881b74d4c8ccf87f3e2064447deb6b9b870fee0d8df176308abc1824e94107

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

Teleport activates the thunder animation then turns it off right after it is used in BossTeleport.cs script.

Link (BossTeleport.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/9ce7a17246111f12a89a5a9b3ac592cdf32da9e2#diff-464aa7ec7e93d085dab00790c3a3924dfb6d7b976691e721348449a541ad2303



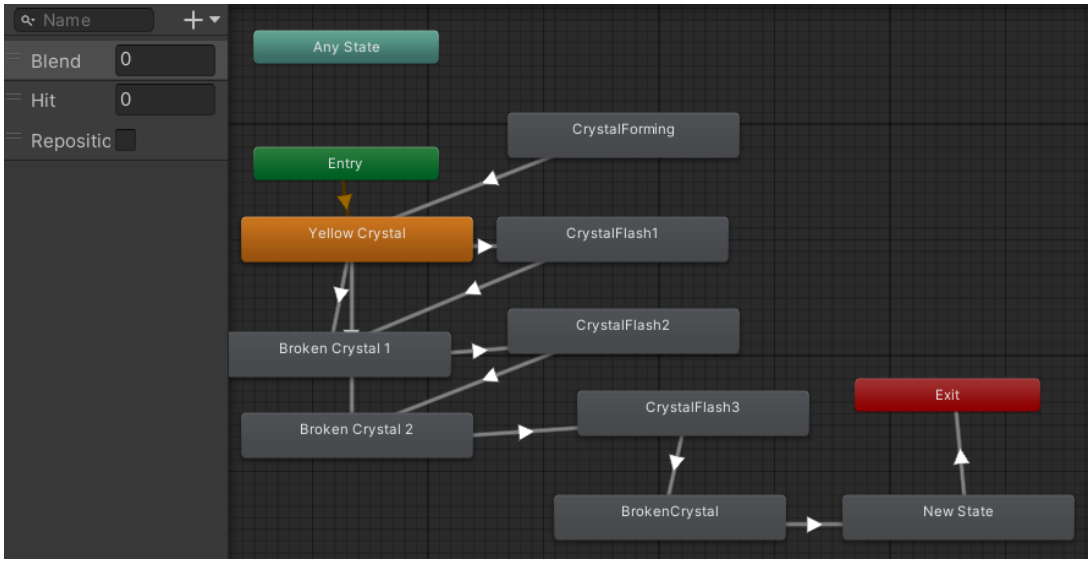
Crystals rotate around the parent using transform.Rotate in CrystalMovement.cs. The crystals use CrystalHealth.cs to determine its state of animation.

Link(CrystalHealth.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/aa9fb9f6108f293d5a49d162e5ee05a6f4bf5f7a#diff-15dc90ab8a7613ab2bdcf55e0aa9ececff98bb93d01d4971460189319a1a7e96

Link(CrystalMovement.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/291a08491d86d21f8090f960b70ec0b0d94bb176#diff-f4448b39b053f96b1d4b7ccd4744d

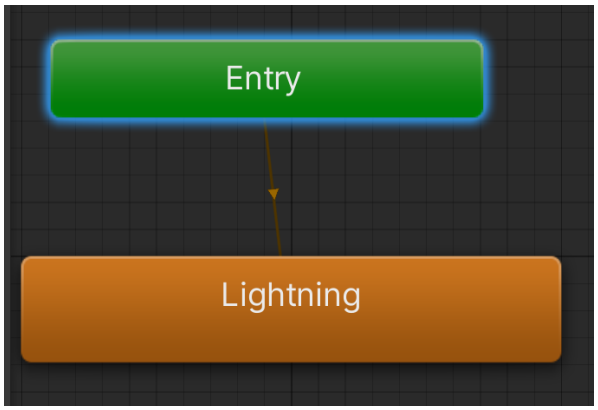
Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

082db261107cf2911f70970d49dcaf6f5f5



Lightning attack plays animation looping for a certain time after it is enabled and stopping when disabled in HitArea.cs

Link (HitArea.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/083d2346a140c4f9f7b86987bb7d0ea5b142b2bf#diff-b4ffffc22e0fbdebac5ab7bb1d1b0db0e99b6646e356c699648668f979b042f3

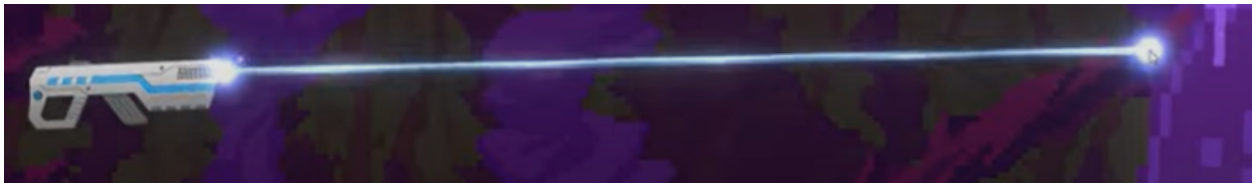


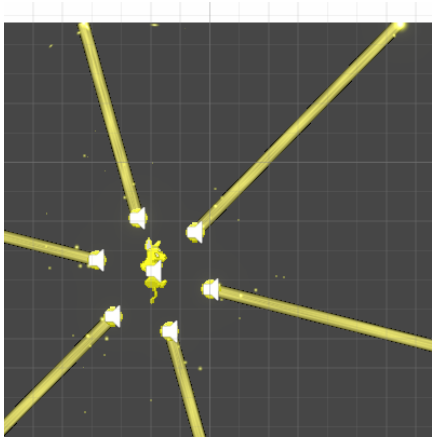
Laser Animation

Followed a YouTube tutorial on how to make lasers shoot towards mouse click. Only used code in the tutorial to enable and disable the laser and particles. Uses shaders, line renderer, and particle system to create the animations.

YouTube Link: https://youtu.be/S6eRVwAtfOM

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco



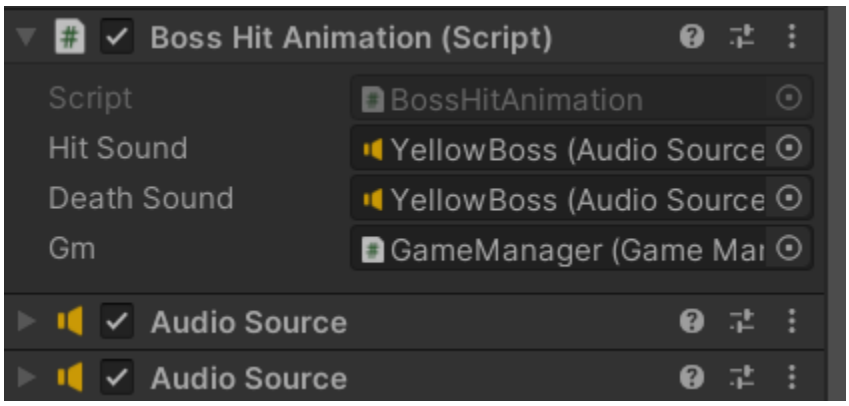


Task: Sounds

Sound for Boss and Crystal Balls

Incorporated audio sources with the animations in the main parent for when hit or dead called when crystal is destroyed.

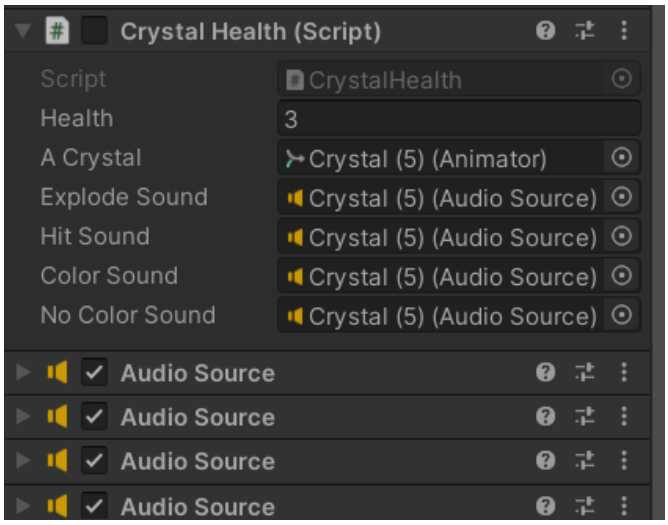
Link (BossHitAnimation.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/9ce7a17246111f12a89a5a9b3ac592cdf32da9e2#diff-dcc458c0de7f7b34d00c86e789ff27ca5d3432cfa670d5d6d661a215afb0657f



Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

Each crystal has an audio in Crystal Health script when hit by different colors or destroyed.

Link (CrystalHealth.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/9ce7a17246111f12a89a5a9b3ac592cdf32da9e2#diff-15dc90ab8a7613ab2bdcf55e0aa9ececff98bb93d01d4971460189319a1a7e96



Stories:

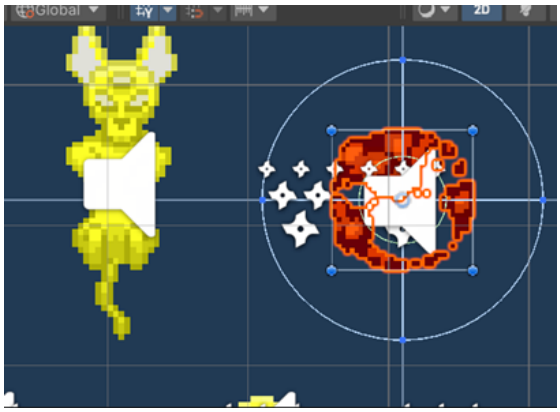
Yellow Boss Health System

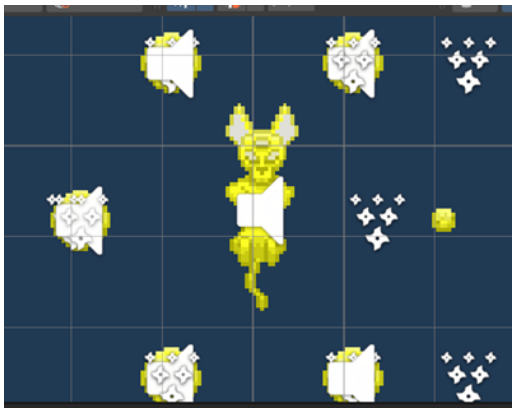
System for health consist of CrystalHealth.cs in each crystal calling YellowBossHealth everytime a crystal ball is destroyed to damage player and track it's health. It then goes into BossHitAnimation.cs for animations and sounds to play in the parent gameobject. The player can hit the enemy with any weapon but only the same color would do more damage. Each crystal ball can reform but the boss will only do so 3 times. Once all 9 are destroyed, the boss gets destroyed.

Link (CrystalHealth.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/com

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

mit/083d2346a140c4f9f7b86987bb7d0ea5b142b2bf#diff-15dc90ab8a7613ab2bdcf55e0aa9ececff98bb93d01d4971460189319a1a7e96

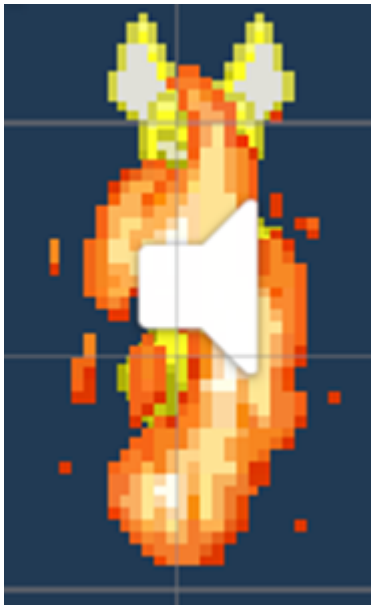




Link (YellowBossHealth.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/aa9fb9f6108f293d5a49d162e5ee05a6f4bf5f7a#diff-5b68f5e6ba872baae4984052d10185d8a84b382f6644dee6f4ac296768eee004

Link (BossHitAnimation.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/aa9fb9f6108f293d5a49d162e5ee05a6f4bf5f7a#diff-36881b74d4c8ccf87f3e2064447deb6b9b870fee0d8df176308abc1824e94107

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco



Yellow Boss Attack #1

Updated a YouTube tutorial by making it a charged attack with the particle system. The shooting is done . Shoots and stops lasers to raycast to shoot in direction away from the boss in LaserBeam.cs.

In LaserAttack.cs I enabled each child laser and disabled each health, then I disabled lasers and enabled health again after a cooldown.

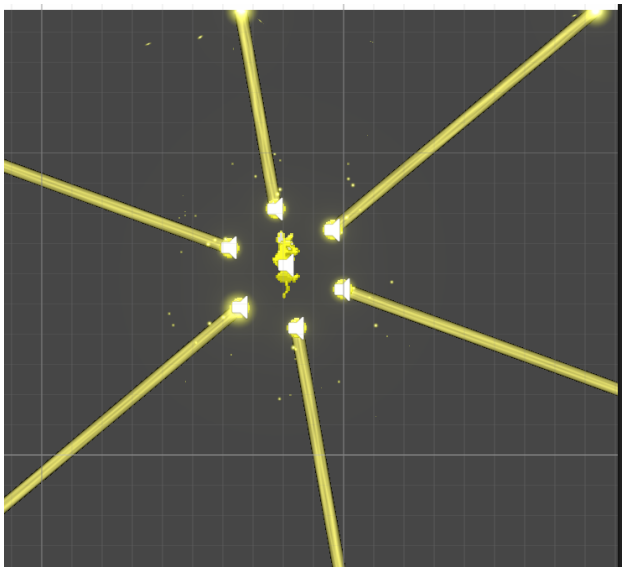
LaserVisuals.cs is used to rotate the lasers around the player.

Link (LaserBeam.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/4a763664993440fbcdbeee79d570fb72f6cf183d#diff-a1966294d16fe68d2350e2ea84a29c1775ee442dc97b0b407483b5845332ce13

Link (LaserAttack.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/4a763664993440fbcdbeee79d570fb72f6cf183d

Link (LaserVisuals.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/4a763664993440fbcdbeee79d570fb72f6cf183d#diff-cd289d310e019873b73ac1e78b76ba69a278f41e0abebaada485314ab2295791

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco



Yellow Boss Attack #2

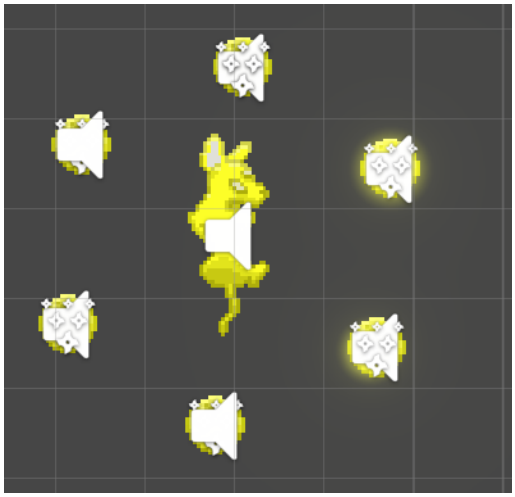
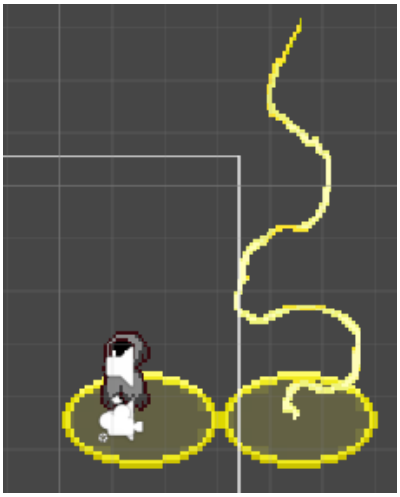
LightningAttack.cs charges crystals, enables attacks, and disables health and enables health once attacking is done.

HitArea.cs unparents and teleports to player's location before activating. It attacks the player after a given time using a collider on the area then parents again or destroys if the parent is gone.

Link (LightningAttack.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/083d2346a140c4f9f7b86987bb7d0ea5b142b2bf#diff-811afda76863d5172b0542c386058b5a80c951133671983a3c5a3efe65e691cd

Link (HitArea.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/083d2346a140c4f9f7b86987bb7d0ea5b142b2bf#diff-b4ffffc22e0fbdebac5ab7bb1d1b0db0e99b6646e356c699648668f979b042f3

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

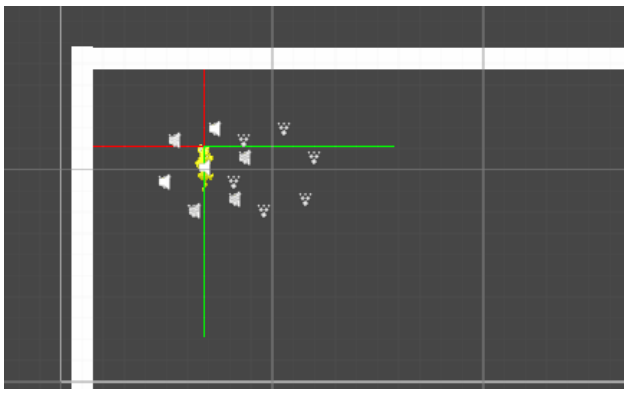
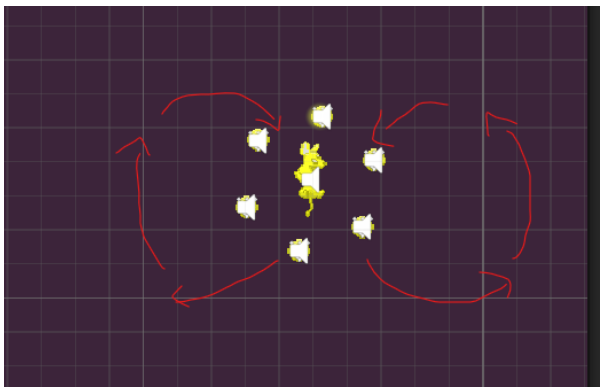
Yellow Boss Movement #1

If the boss is too close to the walls, MovementCorrector.cs will position it a certain distance from the wall giving it enough space for movement.

In MovementInfinityPath.cs the boss travels in an infinity symbol path using sin and cos Mathf library for a certain time.

Link (MovementInfinityPath.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/e349b31b8c2a7cf6363be607488a04178f235332#diff-c0919b797b5e434b145537fdca861071d33231fa8fd3118d2aa9a35509f06f9a

Link (MovementCorrector.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/fc1f3d661f5e891b0ecfd014db9d54d582b6e0a3#diff-c3f9b675a97eb74b33de2316af4949a80ce7900191bd8a9e24fcf459b9712adf

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

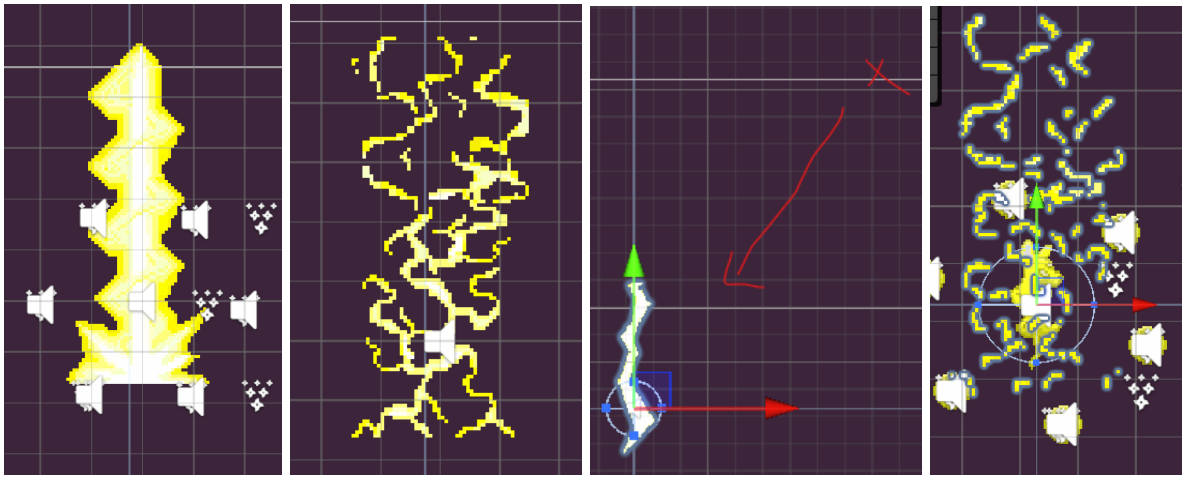
Yellow Boss Movement #2

The cooldown of the teleport is controlled by the Teleport Control while Boss.

Teleport keeps checking if there is a location available within a square shaped room that is not on top of the player. Using coroutines it plays lightning animation, deactivates boss, teleports the boss somewhere else, activates boss, then stops lightning animation.

Link (TeleportControl.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/9ce7a17246111f12a89a5a9b3ac592cdf32da9e2#diff-80280b0fe5dc141844b03f1c9175f572c7e6c5b15963b7fcad25c6e5bdedf643

Link (BossTeleport.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/9ce7a17246111f12a89a5a9b3ac592cdf32da9e2#diff-464aa7ec7e93d085dab00790c3a3924dfb6d7b976691e721348449a541ad2303



Boss' AI Controller

In AttackController.cs the boss controls the attacks and movements depending on its remaining health, player distance, and active weapon of the player. There is also a 20 percent chance to change them. When an attack stops the movement plays and vice versa.

Using chatgpt I got a random number generator that I'm trusting.

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

```
1 reference
public int GenerateRandomInt(int min, int max)
{
    using (var rng = new RNGCryptoServiceProvider())
    {
        var buffer = new byte[4];
        rng.GetBytes(buffer);
        var result = BitConverter.ToInt32(buffer, 0);

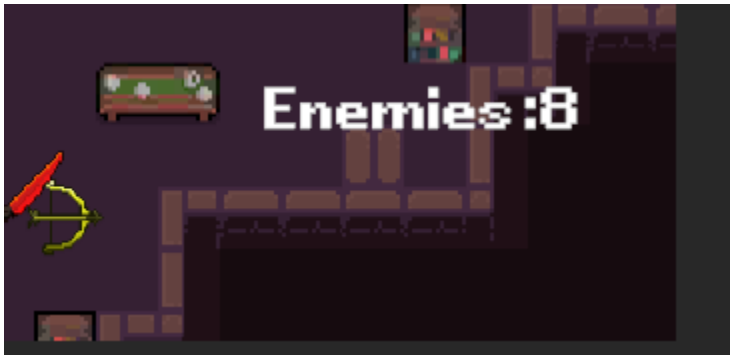
        return Mathf.RoundToInt(Mathf.Lerp(min, max, (result / (float)int.MaxValue)));
    }
}
// Update is called once per frame
```

Link (AttackController.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/dda6fe91ef814e6d29d42419d86c5a271e642612#diff-457e1157d539790bdd9e340a90d5c2682f43056e61f368c2e8e7c0c674b6cb0e

Show Remaining Enemies Till Boss Fight

GameManagerScript adds a count to enemies which is displayed on the UI. EnemyStats.cs adds a counter at the start and subtracts when enemies get destroyed. It updates the canvas TextMeshPro text in the UI with the current number of enemies. When it reaches 0, the boss scene starts.

Link (GameManagerScript.cs):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/d6454a1f5e7e20dc9c82c91f384f448927f0428f#diff-50601cbafc65639d8a163ec9cb3653e4d0d08c6d3d46081493e82df685d9f113



Features: Juan Garza

Task:Corridor Size

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

Fix Corridor Size

As previously shown the corridor was established just as to traverse from one room to the other, in testing the player (depending on the generation of the map), could get stuck with an enemy in the corridor making it challenging to make it to the next room. The algorithm changes this by expanding the corridor. I followed the following youtube video to achieve this.

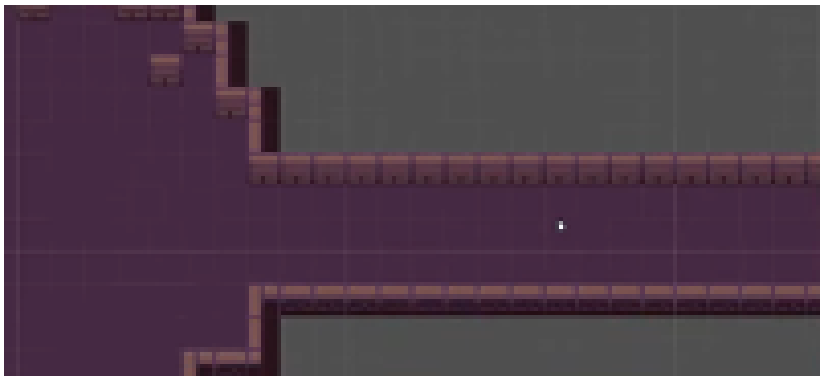
Link (Increase Corridor Width):

https://youtu.be/jCVa4O4RksU

Link (corridor update):

https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/c4e37d3dca15d0195a775f7a5d7ee9ee211a0ff8

```
public List<Vector2Int> IncreaseCorridorBrush3by3(List<Vector2Int> corridor)
{
    List<Vector2Int> newCorridor = new List<Vector2Int>();
    for (int i = 1; i < corridor.Count; i++)
    {
        for (int x = -1; x < 2; x++)
        {
            for (int y = -1; y < 2; y++)
            {
                newCorridor.Add(corridor[i - 1] + new Vector2Int(x,y));
            }
        }
    }
    return newCorridor;
}
```



Task:Balance Room (High Level System)

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

Create a High Level System

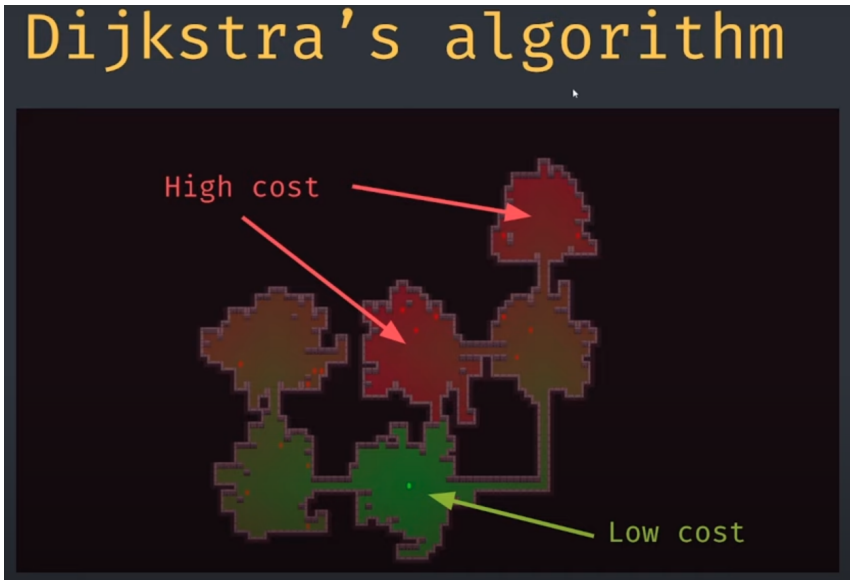
The player is placed in the dungeon with items alongside the player. The player room ensures that the player has all the necessary things needed for the room. While I implemented the algorithm to make this functional, Jonathan was able to utilize it fully. Dijisktra's Algorithm was used to calculate the cost of movement to each tile from the start position.The following youtube video was used in the implementation.

Link (How to Place Items procedurally):

https://youtu.be/t1a1QBcfRlM

Link (High Level System):

https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/c8919dd7dfb88464d42dffae309212fde8a7c822



Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

```
public void RunDijkstraAlgorithm(Vector2Int playerPosition,IEnumerable<Vector2Int> floorPositions)
{
    graphReady = false;
    graph = new Graph(floorPositions);
    dijkstraResult = DijkstraAlgorithm.Dijkstra(graph, playerPosition);
    highestValue = dijkstraResult.Values.Max();
    graphReady = true;
}


private void OnDrawGizmosSelected()
{
    if (graphReady && dijkstraResult != null)
    {
        foreach (var item in dijkstraResult)
        {
            Color color = Color.Lerp(Color.green, Color.red, (float)item.Value / highestValue);
            color.a = 0.5f;
            Gizmos.color = color;
            Gizmos.DrawCube(item.Key + new Vector2(0.5f, 0.5f), Vector3.one);
        }
    }
}
```

Task:Corridor Size

Game Over Screen

A simple game over screen was implemented to ensure the three functions helped the player get to where they needed to get after they were slayed. As shown the player has three options to use from. A Game Manager script was implemented to ensure that the game loaded the screen at the right time. The following video was used to implement.

Link (Increase Corridor Width):

https://youtu.be/pKFtyaAPzYo

Link (Game Over Screen):

https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/50178a097e36b9d67da9e870548c5077ca2c2327

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco


```
public void gameOver()
{
    gameOverUI.SetActive(true);
}
public void restart()
{

neManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
}
public void mainMenu()
{
    SceneManager.LoadScene("TitleScreen");
}
public void quit()
{
    Debug.Log("Quit has been pressed");
    Application.Quit();
}
```

Task:Boss Dungeon Arena

Boss Arena

While Jonathan was working on the boss, we were able to use a previous generator that we had utilized in the creation of the dungeon called room first generator. With this, we were able to generate a boss room to allow the player to fight the boss at the end of the dungeon's main objective. The following video was used to implement the room first generator.

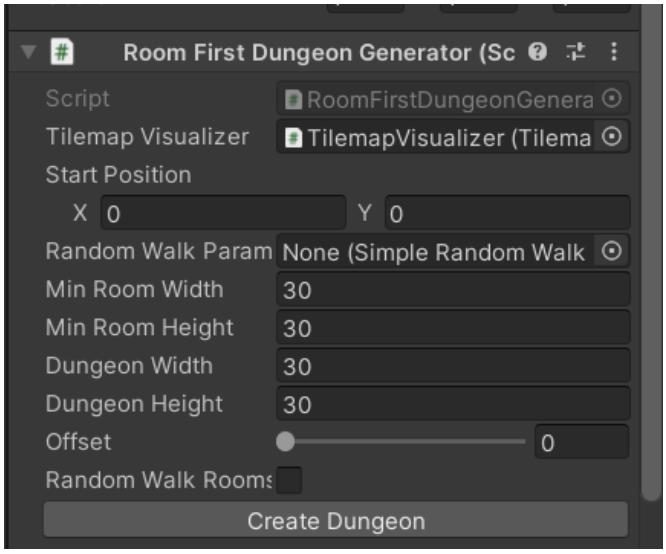
Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

Link (procedural 2D Dungeon):

https://youtu.be/-QOCX6SVFsk

Link (Dungeon Update):

https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/c8919dd7dfb88464d42dffae309212fde8a7c822




```
private void CreateRooms()
{
    var roomsList = ProceduralGenerationAlgorithms.BinarySpacePartitioning(new BoundsInt((Vector3Int)startPosition, new Vector3Int(dungeonWidth, dungeonHeight, 0)), minRoomWidth, minRoomHeight);

    HashSet<Vector2Int> floor = new HashSet<Vector2Int>();

    if (randomWalkRooms)
    {
        floor = CreateRoomsRandomly(roomsList);
    }
    else
    {
        floor = CreateSimpleRooms(roomsList);
    }


    List<Vector2Int> roomCenters = new List<Vector2Int>();
    foreach (var room in roomsList)
    {
        roomCenters.Add((Vector2Int)Vector3Int.RoundToInt(room.center));
    }

    HashSet<Vector2Int> corridors = ConnectRooms(roomCenters);
    floor.UnionWith(corridors);

    tilemapVisualizer.PaintFloorTiles(floor);
    WallGenerator.CreateWalls(floor, tilemapVisualizer);
}
```

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

Task:Background Music

Background Music

While implementing the entire game, music was something that we wanted to ensure we got it right from the start. We were able to find three options for the main menu, in dungeon music, and also boss battle. I was able to implement them using a couple of empty objects and using the audio component. Following a simple YouTube video, I was able to complete this task. The following three songs were chosen from a royalty free website:

Ava Low - Into the Prism

https://www.epidemicsound.com/track/JusvqlEULL/

Lexica – Helios

https://www.epidemicsound.com/track/HLklhDSNTw/

Lupus Nocte – Astrophage

https://www.epidemicsound.com/track/t40NyOP3s9/

Link (Music):

https://youtu.be/KOf3P5y19Bw

Link (Music Addition):

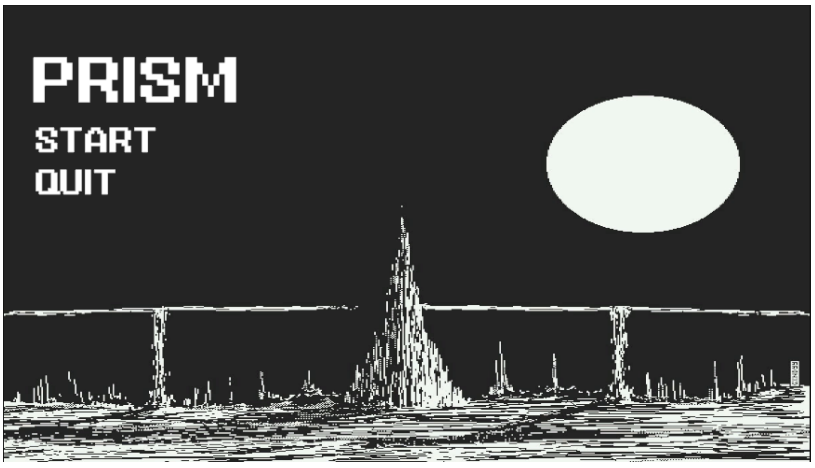
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/9760e321b331849b3a0f7ce49a27cd0e9d48a6d2

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

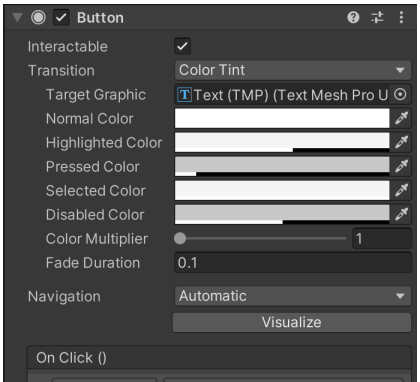
Features: Eduardo Cruz

Branch: titleScreen

Task: Game Menu



Built the title screen for the project following a tutorial. Started with creating two buttons on a new scene using Text Mesh Pro. Imported the Font from an open source website. While trying to import the font I learned that the font has to be imported as a TrueType Font in order to work properly with TMP. Using the



build editor, I set up the scenes and connected the buttons to switch between the scenes in the MainMenu.cs script and added effects to the buttons when hovering over them in the editor. Lastly I added the background Image to set the scene for the game.

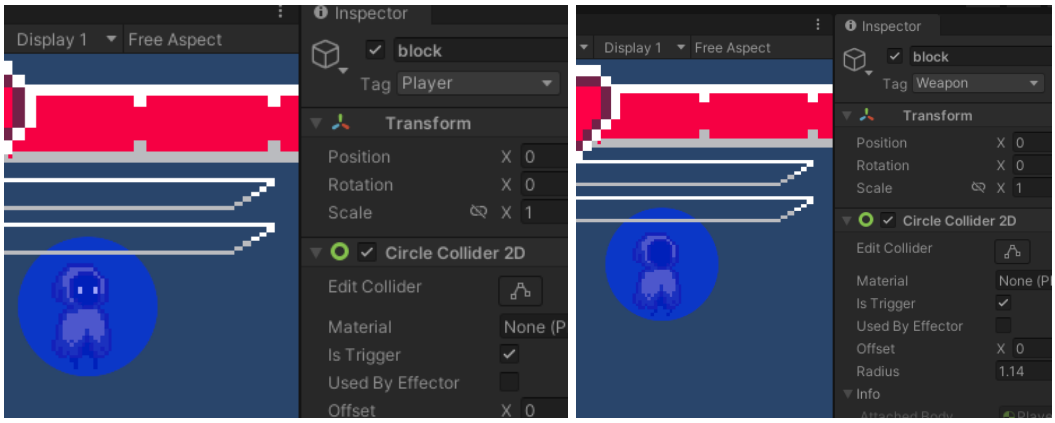
Youtube Link: https://youtu.be/-GWjA6dixV4

Github:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/e3075adb716523e9af2f17d7eb0eff8e7c56955a#diff-ff79cf706dad49900bac9c73daa3a72ea8b22ad6270b2c0fa6b1adfad850e96b

Branch: batt

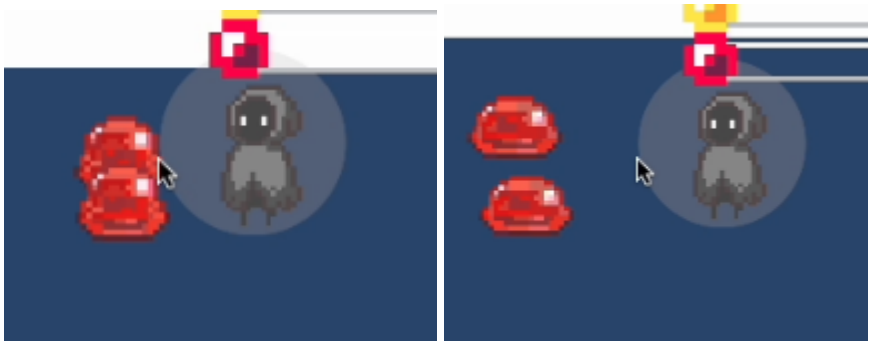
Task: Player Shield Attack

For this task I worked on Shield.cs. In this script I created a new series of code to change the blocks tag to 'Weapon' from 'Player' when using the attack. I changed it from being right click to use to left click for easier simplicity to just use block. When the User clicks both the left and right mouse button, this triggers the tag of the shield to swap to weapon and triggers the knockback script on the enemy which determines if it collides with a weapon it is then knocked back.



Left: Only Left clicked pressed. Right: Both Left and Right click pressed.

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco



Shield Knockback

Cleaned up enabled shield into a function to avoid repeating code in the file. And Lastly I added a OnTriggerEnter2D collider to detect and damage enemies referencing the Enemy Stats files.

Shield.cs:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/pull/61/commits/8840f4d8e98f6d614be43e5b7d165365f3de70f4#diff-0b351d5279325d543c36695801b7811e063a8c8593fd686a98cb3623820495b0

Task: Fix Sword Hit One Enemy Per Swing

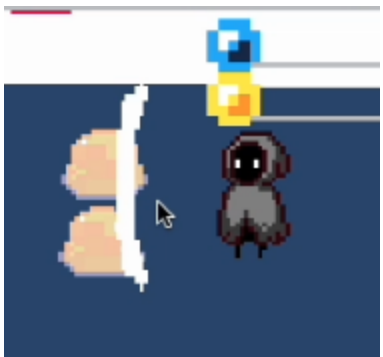
First in this commit I fixed an issue where the shield attack would attack randomly and fail to block. This issue was caused by Swapbars.cs not being set to Awake() and instead was set to Start().

SwapBars.cs:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/pull/61/commits/55d07b3ea7a744feb0a9f4d1c1bf697e48f446cf#diff-3843213c7d1fba4dec17bf1d366fb9c05ea4bf84b81645d16ea4dec829dc5523

Once that issue was addressed I looked up the different types of trigger Colliders in Unity C#. There I found an article which explained what each OnTrigger Collider did and how it stores information.

Article: https://www.patrykgalach.com/2019/09/05/understanding-ontrigger-methods/

From there I decided to change it from OnTriggerStay2D to OnTriggerEnter2D. In this function I stored the information of the game objects that entered the area onto a list. From that list, in the main function I sorted through each to find



those with the enemy tag and dealt damage to them as well as still apply the knockback in Sword.cs. Also I wanted to make sure that the list would remove existing objects that have been dealt with and those who leave the area. To do this I made another function for OnTriggerExit2D which removes game objects. Also set a cooldown using a coroutine function for the hits.

Sword.cs:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/pull/61/commits/55d07b3ea7a744feb0a9f4d1c1bf697e48f446cf#diff-53909357bad7634d2166c7ae109f475aa7ede019d8132c40b97e0a4b42262e91

The same was done for Shiled.cs in order for the shield to also detect multiple enemies. The collider was already a OnTriggerEnter, so only alist was needed to keep track and damage everyone in the area. Also created a cooldown using coroutine so it wouldn't spam the damage dealt on the enemy.

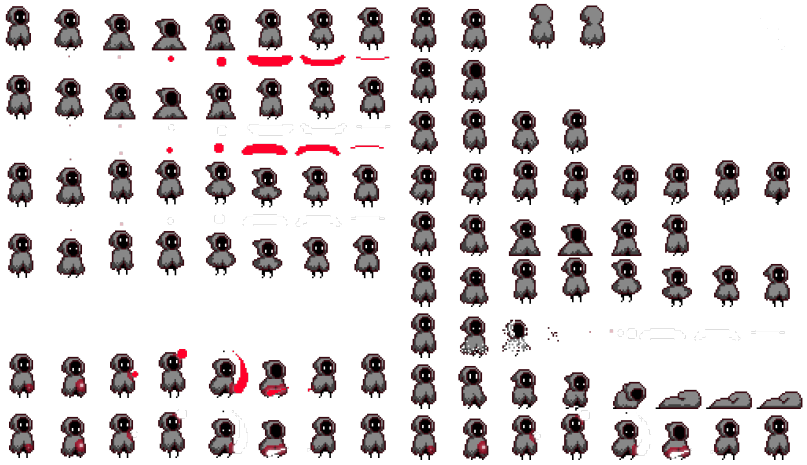
Shield.cs:

https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/pull/61/commits/55d07b3ea7a744feb0a9f4d1c1bf697e48f446cf#diff-0b351d5279325d543c36695801b7811e063a8c8593fd686a98cb3623820495b0

Branch: newAnimation & fixArrow

Task: Player Animation

Found player sprite sheet from itch.io to replace the therion sprite sheet from a Nintendo game.

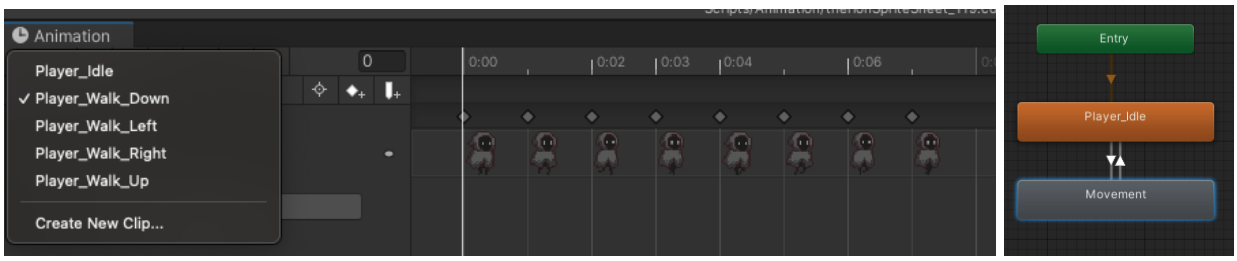


Took a while but I made new sprite sheets and had to redraw a couple because it would not notice the opacity of some. Recolored the sprites and made new swing animations for the melee attack.

Link:https://penzilla.itch.io/hooded-protagonist

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

For the new animations, in order to avoid messing with the code and have function properly due to time constraints, I replaced all the movement sprites with a simple walking motion.



This would resolve me having to mess with the animator and work more on coding.

Based on the input from the user, the function Flip() in Player Movement is called. There was an issue where the sword would not flip according to the direction the player was facing, that was adjusted through various tests and updated in the PlayerMovement.cs file.

PlayerMovement.cs:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/pull/65/commits/aad8fe878cbf7e89adbf24f68e7c9d14e37b00e4#diff-e4a119584c97350c5432078f4ca08d355511b07ed01c0c00479b9e3ae7798c51

The first issue that would come from this is the Player's attacks not flipping properly based on the direction given. So in the PlayerMovement.cs script I added a variable that checks if the player is facing right or not.

PlayerMovement.cs:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/pull/65/commits/7d962f3abf002239c64db36c028472f223a6c870#diff-e4a119584c97350c5432078f4ca08d355511b07ed01c0c00479b9e3ae7798c51

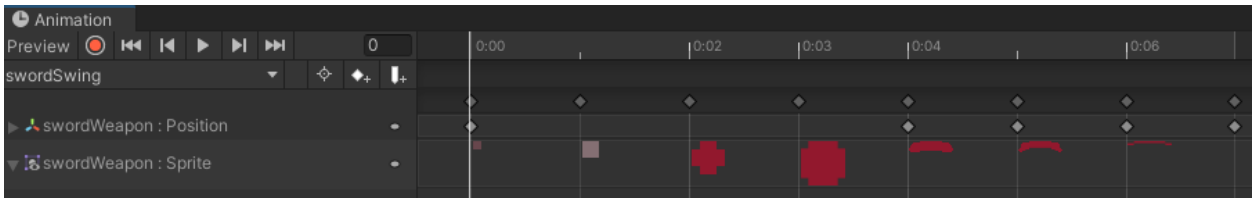
This is referenced to fix an issue involving the Arrow attack where the direction of the range attack was not aligning with the direction you were facing, and was inverting based if you were flipped. So if you were facing right the input from the mouse location would be inverted to give the accurate location no matter if you were flipped or not.

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

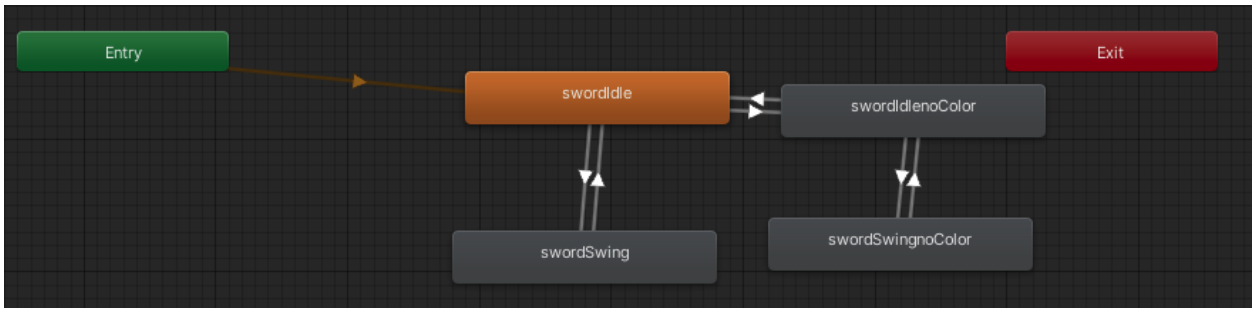
RangeAttack.cs:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/pull/65/commits/7d962f3abf002239c64db36c028472f223a6c870#diff-ef61682c31ef7a22258f2cf92f8b135f45504694f1dfc337807afd1891163805

Task: Fix Combat Animation

The animations fo combat were updated using the new sprites. There was an issue that i was having in the sword swing where the sprite would not align, instead I created a new sprite sheet solely for the swing attack.



Using this new animation sprite sheet I edited the attack range to be more broad and noticeable. Then in the animation I moved the spirited to create a cool swing effect that seemed efficient and noticeable for the player.



The animations were fixed for all the objects in the animator. After this the prefab for the bow and arrow were changed to match with the rest of the attacks.

Commits:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/pull/65/commits/7d962f3abf002239c64db36c028472f223a6c870

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

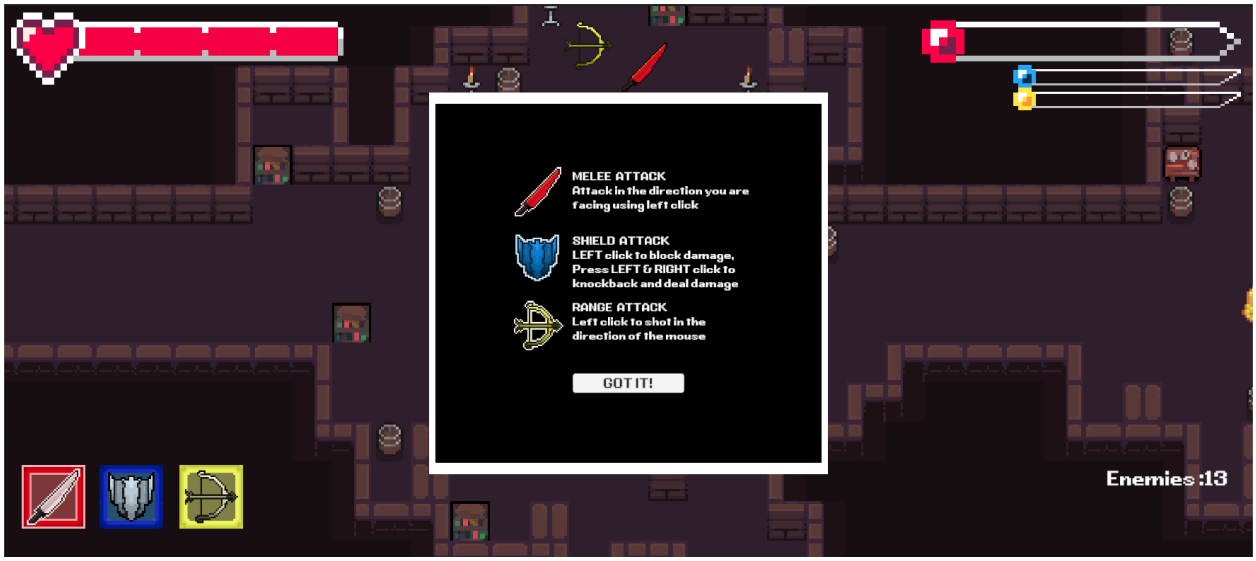
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/5a3f61f24e5cec1c81ae104630f624d4dfc5fb53

Branch: Instructions & bossMsg

Task: Add Attack Instructions

I found an article that gives a step by step process of how to create a UI pop-up window to set the instructions of how the attacks work to inform the player.

Article:
https://answers.unity.com/questions/1327989/open-a-popup-window-with-text-and-images-when-clic.html



The PopupMessage canvas holds a panel which is set to be transparent so you could still see the game behind it.

The following two Canvases Frame and Popup, simply just hold the color wanted and act as the box that displays the information.

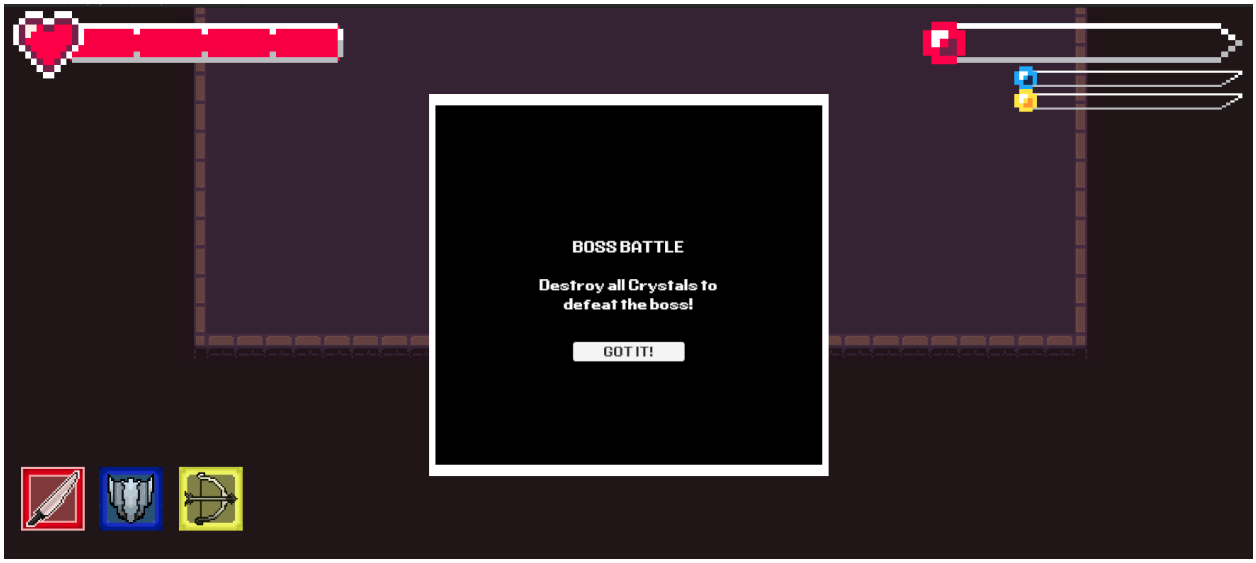
Then with the UI raw images and text were added to add the information needed.

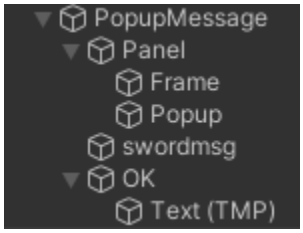
Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

The Button ‘OK’ is set to close the PopupMessage using the Script PopupMessage.cs Script located in the GameManager object.

PopupMessage.cs:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/bcb9fd7936bc33a35cc08ea0fa9f6e8068abb43f#diff-56b4da7ffaba68855682ec5b332e04bb6828bb35a5512b496b308f3124a33df1

Task: Boss Message

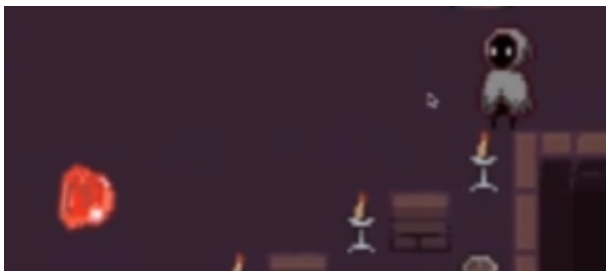




The same thing was set to the boss scene where a short message pops up to give instructions on how to defeat the enemy. This uses the same script as before so no new scripts were needed.

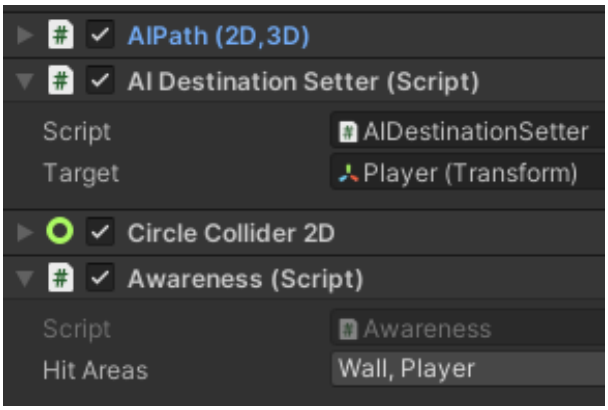
Branch: redfix, Flash, & PIVOT

Task: Fix Red Enemy



There was an issue where the red enemy was being pushed away from the parent and kept spiraling and moving toward a direction. This would happen whenever the

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco



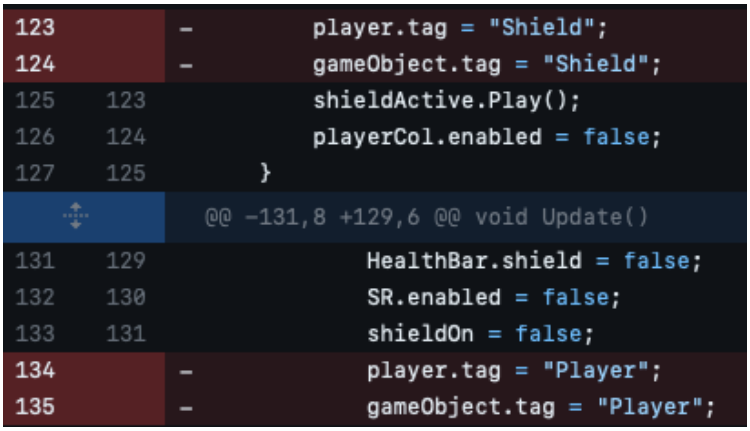
player got close and walked over the red enemy

The problem that was found was that the prefabs constraints and properties had been disabled causing the slime not to move towards the player and be knocked around.

The target for the player was also removed. And the rigibody was also removed.

Task: Fix Combat

Another Issue that was found is that the sword and shield were no longer attacking the Enemies. Looking through the code the tag for shield was being changed and not considered a weapon, because of that it was no longer harming the enemy because it no longer saw Weapon, it instead saw Shield. I removed the tag changes to fix the issue in the shield.



Shield.cs:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/dc1e8d5775d042fe0bdbd03ec225a1d958ed9c2e#diff-0b351d5279325d543c36695801b7811e063a8c8593fd686a98cb3623820495b0

After some time the shield was no longer knocking back as supposed to, this was later fixed in a new branch called ShieldKB where I moved the position of where the tag was being changed to Weapon.

Shield.cs:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/0bb302f10f3b55ccf6c9eb220017b38a8ba5d946#diff-0b351d5279325d543c36695801b7811e063a8c8593fd686a98cb3623820495b0

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

For the Sword Attack the PolyCol.enabled was not working properly. I removed it and from what I could tell it was working fine.

Sword.cs:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/dc1e8d5775d042fe0bdbd03ec225a1d958ed9c2e#diff-53909357bad7634d2166c7ae109f475aa7ede019d8132c40b97e0a4b42262e91

Then there was an issue that the sword area was being recognized as the player so when the player held the sword if an enemy object entered the proposed attack area, it would harm the player. This was fixed in the coroutine to switch the tag to sword instead of player.

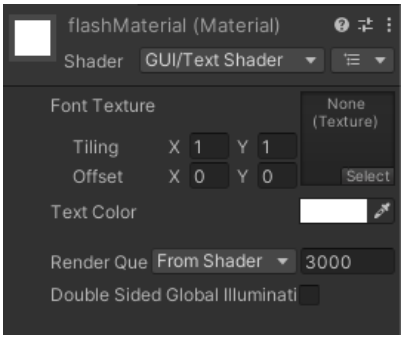
```
IEnumerator wait()
{
    yield return new WaitForSeconds(1f);
    gameObject.tag = "Player";
    gameObject.tag = "Sword";
}
```

Sword.cs:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/285bb2b825ccc97991981ef2f166792efbb8b4a8#diff-53909357bad7634d2166c7ae109f475aa7ede019d8132c40b97e0a4b42262e91

Task: Player Damage Indicator

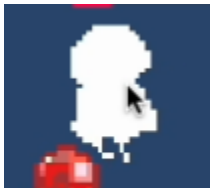
For this task I followed a youtube tutorial to create an indicator which reveals when the player is being hit and damaged.

Youtube Link: https://youtu.be/9rZkiEyS66I



Following the tutorial I created a new Material with the flash color wanted, in this case white. Then on the player I created the FlashDMG.cs script which will change the color by changing the material of the sprite for a brief second when activated.

FlashDMG.cs:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/co



mmit/285bb2b825ccc97991981ef2f166792efbb8b4a8#diff-f8f61a50330f7500b3a4670c09dd1c04867140a8afb1e9426b8819d18e360503

This is set to the HealthBar.cs script since it is what takes account if the player is damaged. So by referencing the FlashDMG.cs script if you lose health, it calls the Flash() function to make the player flash momentarily.

HealthBar.cs:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/285bb2b825ccc97991981ef2f166792efbb8b4a8#diff-6334354973e14f62589ae54c806c1947750febdf76afcab559a63681f1457f62

Task: Fix Yellow Enemy

The yellow enemy's range attack was seemingly colliding before it hits the player. This was a simple issue to fix as I was looking closer at the interaction and saw that the slime ball hits the player and then the animation of it exploding would seemingly teleport it a little further back. To fix this I looked at the sprite sheet and found that the pivots of the sprites were sporadic, after centering them all the animation looked better between the interaction and avoided the random teleports, except for the feet, but Jonathan said he will fix that issue.

Commit:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/7690e8e8eb8a36aa02ef0e9182f70339b6307a25

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

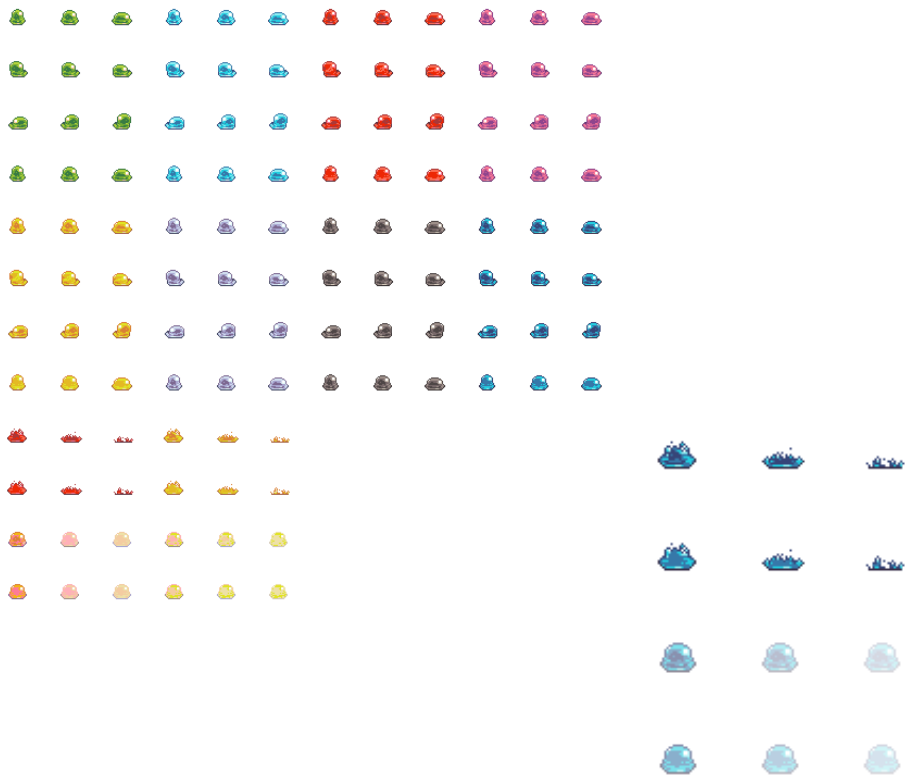
Features: Samuel Pacheco

Task: Animations

YouTube Link (For all Animations): https://www.youtube.com/watch?v=hkaysu1Z-N8

Animation and sprite updates were made for both the blue and red enemy. The red enemy received a darker toned red for itself, which then required the animations to be updated to the correct sprites. On the other hand, the blue enemy was completely new at this point in the development process and due to its attacking mechanics needed a different approach than the red enemy.

Link (Initial):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/81cac41ce1412b2a673c3de91ad254e4e87613a6

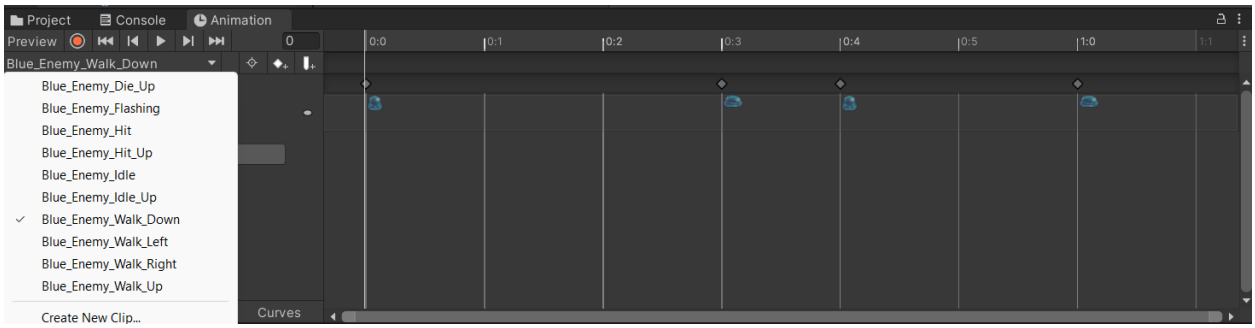


-

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

This initial commit created the animations for the blue enemy slime that included the hit, moving, idle, and death clips.

Link:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/81cac41ce1412b2a673c3de91ad254e4e87613a6

- 

This second commit was created to update the sprites for the red enemy due to errors that were brought up in the above commit for this enemy. A darker shaded red sprite for the slime was created so that it could be distinguished clearly as red. With this new sprite being added the animations for the enemy were needed to be updated for each clip.

Link (Updates for red enemy):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/f33c302650d9432a1dbff4b196fd747d5c669bdc

- 

This third commit was made to create an explosion for the blue enemy once it self-destructs. To make this work the animator ws worked on to satisfy that the enemy would go through with this animation if the attached/self-destruct variable was found to be true.

Link (Updates for blue enemy death):

https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/ad0f4d8b54f97ed4dbd76d3bc1e6ceda25b1e545

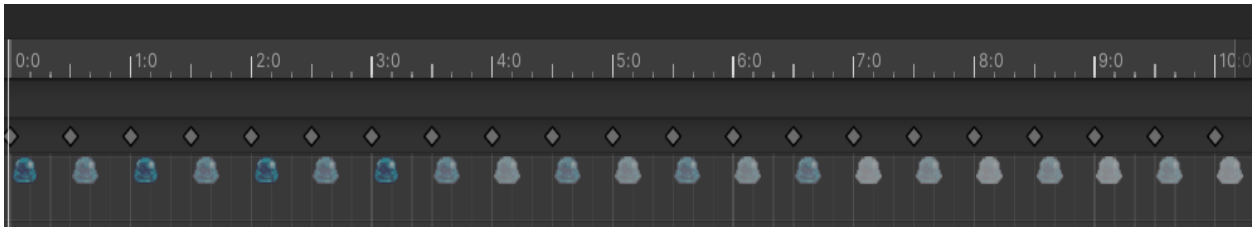
Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

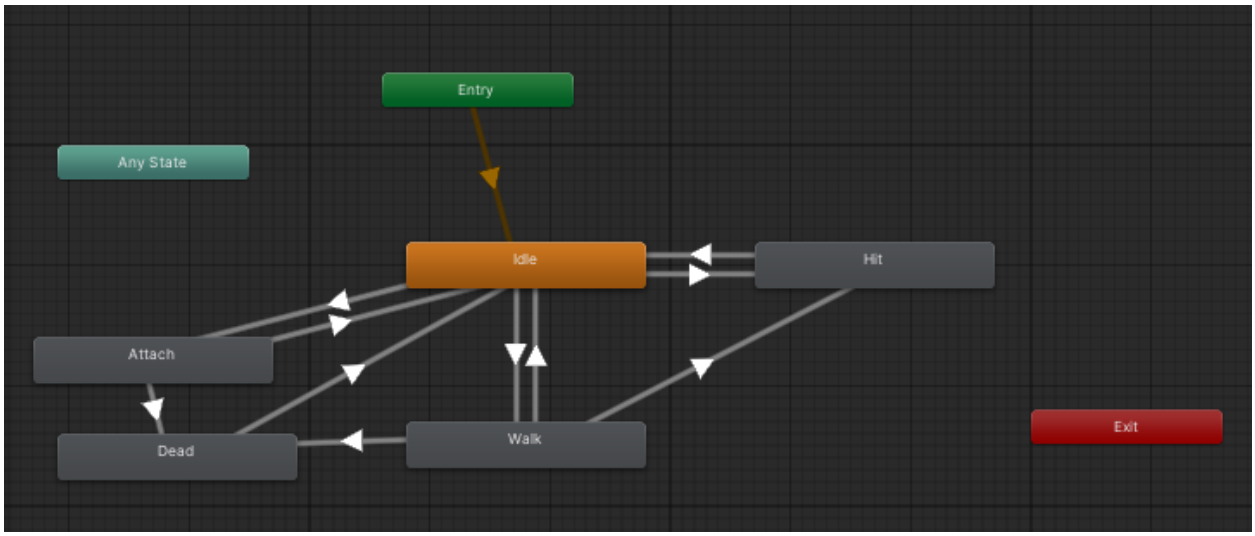
- 

This final commit was made to create a noticeable effect on the blue enemy once it is attached to the player. The blue enemy will begin to flash once it is attached to the player and eventually self-destruct and die.

Link (Updates for blue enemy flash) (FINAL UPDATE):

https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/ab21a71092da92f8787c1bfecc5ba49641cc1c2c

- 

- 

Task: Stories

Blue enemy movement: Similar to the red enemy, the blue enemy uses Astar and awareness to track down the enemy. The code below parents the enemy slime onto the player, which causes the slime to stay on the player.

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

Link (LeechAttack.cs)(Initial):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/205584ce6f8d69edfb338d6f137bd0a14532145c

```
void Update() {
    transform.GetChild(0).position = transform.position;
}

public void OnTriggerEnter2D(Collider2D other){
    if(other.tag == "Player"){
        stats.dontmove = true;
        attack = true;
        aiPath.maxSpeed *= 2;
        aiPath.canMove = false;
        stats.onsight = true;
        stats.idle = true;
        transform.parent.parent = player.transform;
        healthbar.hit(40);
    }
}
```

Initially the goal was to try and unparent the enemy slime, but I was unsuccessful in that task. So I shifted to the slime following the player closely by setting the transform child position to the transforms position.

Link (LeechAttack.cs)(Final):
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/ab21a71092da92f8787c1bfecc5ba49641cc1c2c#diff-421a8e7951c0d175ac032973241aada448472facb02c30bb308dee89cf1ba52a

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

```
void Update() {
    if(attack){
        count += Time.deltaTime;
        transform.GetChild(0).position = transform.position;

        if(count >= 5){
            stats.selfDestruct();
            this.enabled = false;
        }
    }
}

public void OnTriggerEnter2D(Collider2D other){
    if(other.tag == "Shield"){
        count = 0;
        //attack = false;
        stats.animator.SetBool("Attached", false);
        aiPath.maxSpeed *= 0;
        aiPath.canMove = false;
    }
    if(other.tag == "Player"){
        stats.animator.SetBool("Attached", true);
        stats.dontmove = true;
        attack = true;
        aiPath.maxSpeed = 10;
        aiPath.canMove = false;
        stats.onsight = true;
        stats.idle = true;
    }
}
```

Blue enemy attack: The blue enemy will attack by attaching itself to the player and acting as a trigger for the counter. Once the timer is completed the enemy will self destruct causing the player to lose health.

Link:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/ad0f4d8b54f97ed4dbd76d3bc1e6ceda25b1e545

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

```
void Update() {
    if(attack){
        count += Time.deltaTime;
        transform.GetChild(0).position = transform.position;

        if(count >= 5){
            stats.selfDestruct();
            this.enabled = false;
        }
    }
}

public void OnTriggerEnter2D(Collider2D other){
    if(other.tag == "Shield"){
        count = 0;
        //attack = false;
        stats.animator.SetBool("Attached", false);
        aiPath.maxSpeed *= 0;
        aiPath.canMove = false;
    }
    if(other.tag == "Player"){
        stats.animator.SetBool("Attached", true);
        stats.dontmove = true;
        attack = true;
        aiPath.maxSpeed = 10;
        aiPath.canMove = false;
        stats.onsight = true;
        stats.idle = true;
    }
}
```

-

```
public void selfDestruct() {
    selfDestructed = true;
    health = 0;
    healthbar.hit(25);
    Debug.Log("Enemy self destructed");
}
```

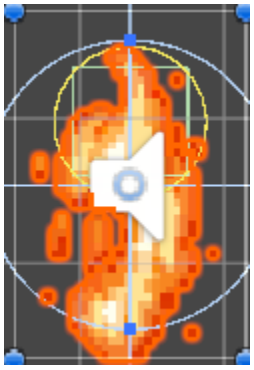
-

Blue enemy self destructs: The blue enemy self destructs only when the player is within its awareness area. This is caused by a timer being completed that is triggered by the player being in the proximity.

Link:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/ad0f4d8b54f97ed4dbd76d3bc1e6ceda25b1e545

YouTube Link (For all Animations): **https://www.youtube.com/watch?v=hkaysu1Z-N8**

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

- 

```
public void selfDestruct() {
    selfDestructed = true;
    health = 0;
    healthbar.hit(25);
    Debug.Log("Enemy self destructed");
}
```

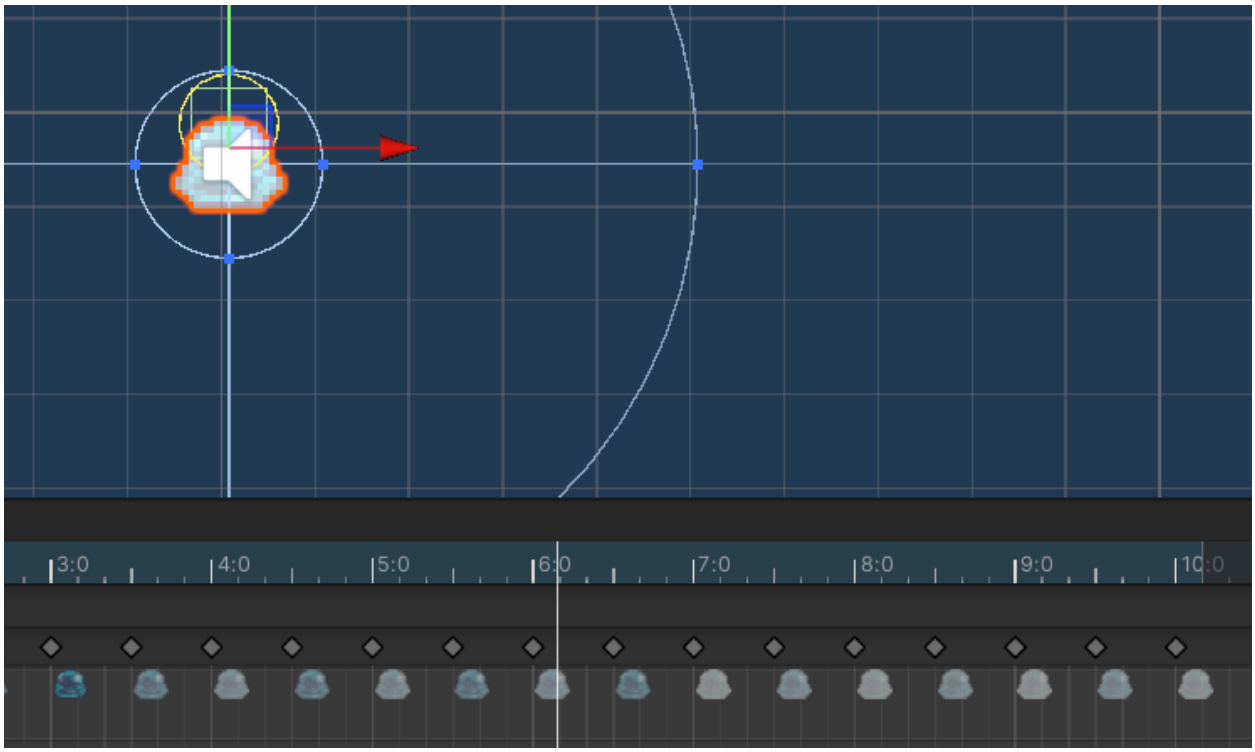
Blue enemy flashes before self destruction: To add an effect so that user's are able to understand what is happening a flashing animation was added to the blue enemy to act as a notice to players that something is going to happen. This flash begins once the enemy attaches to the player triggering the counter as well.

Link:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/ab21a71092da92f8787c1bfecc5ba49641cc1c2c

YouTube Link (For all Animations): https://www.youtube.com/watch?v=hkaysu1Z-N8

```
0 references
public void OnTriggerEnter2D(Collider2D other){
    if(other.tag == "Weapon" && other.gameObject.name == "block"){
        count = 0;
        attack = false;
        stats.animator.SetBool("Attached", false);
        aiPath.maxSpeed *= 0;
        aiPath.canMove = false;
    }
    if(other.tag == "Player"){
        stats.animator.SetBool("Attached", true);
        stats.dontmove = true;
        attack = true;
        aiPath.maxSpeed = 10;
        aiPath.canMove = false;
        stats.onsight = true;
        stats.idle = true;
    }
}
```

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

-
```
0 references
public void OnTriggerExit2D(Collider2D other){
    if(other.tag == "Player"){
        count = 0;
        attack = false;
        stats.animator.SetBool("Attached", false);
        stats.dontmove = false;
        aiPath.canMove = true;
        aiPath.maxSpeed = 2.5f;
        count = 0;
        stats.onsight = false;
        stats.idle = false;
        stats.hitbox.enabled = true;
    }
}
```
- 

Fix enemies from clipping to walls: This was done by creating a script that would push the enemy back and after a set time would return back to its previous velocity. With that in mind, to make this function correctly the tag “Weapon” was to be used by the attacking weapon for the knockback feature to work.

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

Link:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/commit/1ad2c6ad0df632cc691bf2f1e73abe69d1db6af0
YouTube Link: https://www.youtube.com/watch?v=RXhTD8YZnY4

- 
- ```
  0 references
  private void OnTriggerEnter2D(Collider2D other)
  {
      if(other.tag == "Weapon")
      {
          StopCoroutine(reset());
          Vector2 dir = (transform.position - other.transform.position).normalized;
          rb.AddForce(dir * force, ForceMode2D.Impulse);
          StartCoroutine(reset());
      }
  }

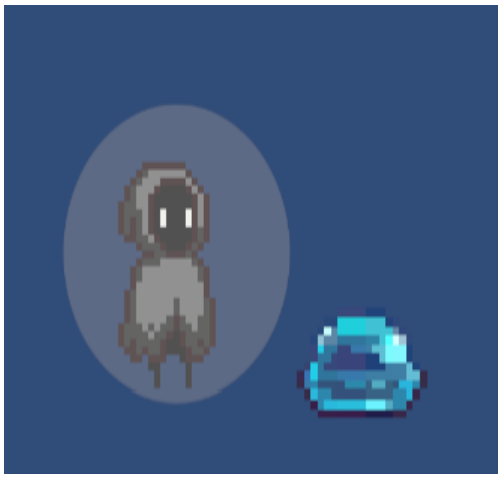
  2 references
  private IEnumerator reset()
  {
      yield return new WaitForSeconds(1f);
      rb.velocity = Vector3.zero;
      transform.parent.parent.position = transform.position;
      transform.localPosition = Vector3.zero;
  }
  ```

Blue enemy releases player: The release function of this enemy is done in the leech attack script. In this script, if the enemy detects a shield trigger then it will stop following the player. Along with this, if the player is able to escape the enemy by dashing and leaving its awareness proximity it will be released.

To make this work the player tag is updated to change to weapon only when the shield is enabled. The first set of code shown below used a previous “Shield” tag that caused the enemies to not function correctly due to conflicting parts in the shield and leech script. The second set of code shows the fix was to make the shield flip to the “Weapon” tag when on, therefore a switch over to the “Weapon” tag was necessary on the OnTriggerEnter.
Link:
https://github.com/UTRGV-CS-Projects/202320-spring-2023-projects-prism-bossrush/com
  ```

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

mit/ad0f4d8b54f97ed4dbd76d3bc1e6ceda25b1e545#diff-421a8e7951c0d175ac032973241aada448472facb02c30bb308dee89cf1ba52a

-  
-
```
void Update() {
    if(attack){
        count += Time.deltaTime;
        transform.GetChild(0).position = transform.position;

        if(count >= 5){
            stats.selfDestruct();
            this.enabled = false;
        }
    }
}

public void OnTriggerEnter2D(Collider2D other){
    if(other.tag == "Shield"){
        count = 0;
        //attack = false;
        stats.animator.SetBool("Attached", false);
        aiPath.maxSpeed *= 0;
        aiPath.canMove = false;
    }
    if(other.tag == "Player"){
        stats.animator.SetBool("Attached", true);
        stats.dontmove = true;
        attack = true;
        aiPath.maxSpeed = 10;
        aiPath.canMove = false;
        stats.onsight = true;
        stats.idle = true;
    }
}
```

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

```
public void OnTriggerEnter2D(Collider2D other){
    if(other.tag == "Weapon" && other.gameObject.name == "block"){
        count = 0;
        attack = false;
        stats.animator.SetBool("Attached", false);
        aiPath.maxSpeed *= 0;
        aiPath.canMove = false;
    }
    if(other.tag == "Player"){
        stats.animator.SetBool("Attached", true);
        stats.dontmove = true;
        attack = true;
        aiPath.maxSpeed = 10;
        aiPath.canMove = false;
        stats.onsight = true;
        stats.idle = true;
    }
}
```

Jonathan Rodriguez
Juan Garza
Eduardo Cruz
Samuel Pacheco

Summary

Requirements:
Based on our initial requirements we failed to develop multiple stages where certain enemies would spawn out as well as multiple bosses. We did however make a map that is generated randomly as well as the number of enemies that spawn in each map. We also were able to create different enemies and attacks to have the player interact with. We also implemented the color system attacks that were proposed in the beginning of the semester. These attacks deal damage to the 3 corresponding enemies more effectively and diminish with use. There is a complete iteration of the game that is playable for the User to experience. Although it differs from our first proposal, it still adequately meets most of the criteria we proposed and is enjoyable for the User experience.

Changes along the way:
Due to lack of time, the number of bosses that we were able to make was set to 1. Also changes in animations were made to better suit the game and environment. This was also made to please comments from the showcase where it was hard to tell what the player was doing. We also decided not to repool the enemies and instead destroy them. In addition, the smaller enemies' behavior was also changed periodically through development.

Lessons Learned:
In the project something that we learned is how many things are accomplishable in a set amount of time. Also when things are being created, making sure that everything still flows properly without any issues.

Project Management Obstacles, solutions:
The project was managed using Jira to assign tasks to certain team members as well as communicating through discord. Some members had more experience and time than others so they voluntarily took on more tasks and roles. If there was an issue of some sort, Discord was our mode of contact to address any issues or to ask others for help.

Teamwork:
The team was readily available in discord to help each other out and take criticism if needed. The Team was very vocal about the things that they were working on as well as any questions or concerns that arose.

Lesson Learned:
As long as there was communication there were no issues in managing the contents of the project. Also if there was an issue, it could easily be addressed. One important thing is to make sure you test changes, not only with what you are working on but in the overall project. There were instances where a script was changed to fix an issue that resulted in other features to break.