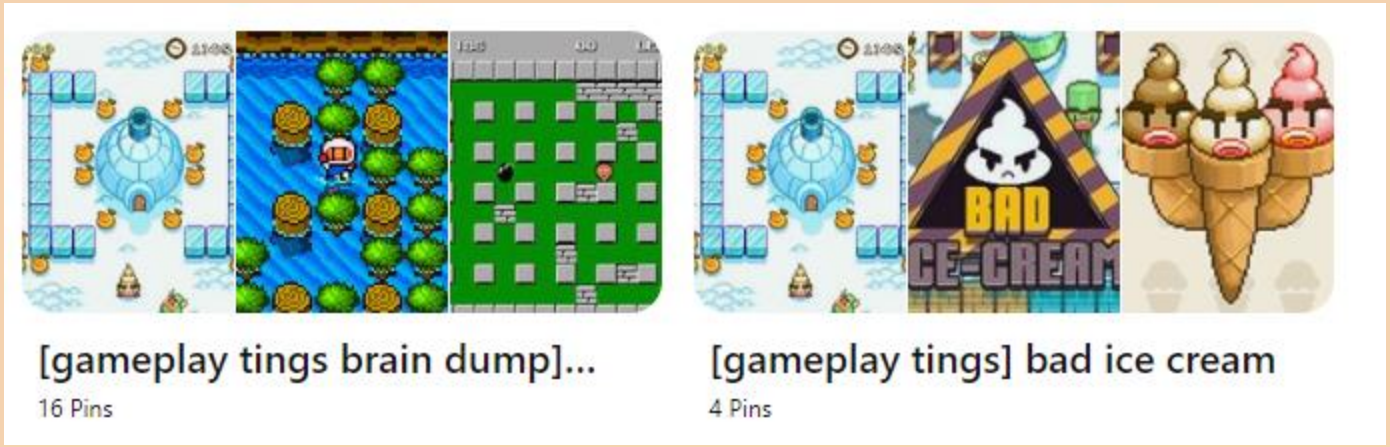
Individual Documentation

Nadine Carillo

Week 1: Brainstorming & Planning



Our group set up a Pinterest board so we could gather all of our ideas together in one place. These screenshots show the contribution I made to our collaboration board. Sharing ideas for potential gameplay that our game's direction could take. I knew our team wanted to do either 2D OR 2.5D so I picked a few nostalgic games within this category.

Although our game didn't take inspiration of any of these games, gameplay or style, I liked the Bad Ice Cream, Fat Princess and Gun Mayhem ideas the most because I felt like with our coding abilities we would be able to create a game similar to these genres.

Our finalised idea ended up being a 2D pixelated Dungeon Crawler set in Halloween 1751.

Week 1: Assigning tasks

nadddine. 08/05/2024 11:55 AM

@everyone

GOALS FOR THIS WEEK

~~PRIORITY attack system done (nadine)~~

- PRIORTY animations done and put into Unity (Theo, Cath)
- PRIORITY map design (Theo)
- ~~camera movement (Aurelia)~~
- weapons done (Theo)
- ~~pause menu (Nadine, Brianna)~~
- ~~code title page to cutscene to game (Nadine)~~

Maybe (priority next week)

- quest manager / item pick up
- minor
- timer
- Teleport
- hert - make player flash red when hit yayaya

(edited)

nadddine. 08/12/2024 12:13 PM Jump

WEEK 5 GOALS @everyone

- ~~PRIORTY animations done and put into Unity (Cath)~~
- ~~PRIORITY map design 1 (Theo)~~
- ~~PRIOTIY Quest Manager (Aurelia, Nadine)~~
- ~~PRIORITY Updated Camera Movement (Brianna)~~
- ~~PRIORITY Mini map (Brianna)~~
- ~~PRIORITY Dialogue (Cathy)~~
- ~~Teleport (Nadine)~~
- ~~Hert (Aurelia)~~

nadddine. 08/19/2024 10:48 AM

WEEK 6 GOALS

- maps (theo)
- ~~dialogue in cutscene (Cathy)~~
- ~~healing zone (nadine)~~
- ~~box colliders (Aurelia)~~
- setting up quests (Aurelia and Nadine)
- ~~point system (Nadine Brianna)~~
- ~~pause menu issues (nadine)~~

(edited)

nadddine. 08/25/2024 4:18 PM Jump

things to do tomorrow (group stuff)

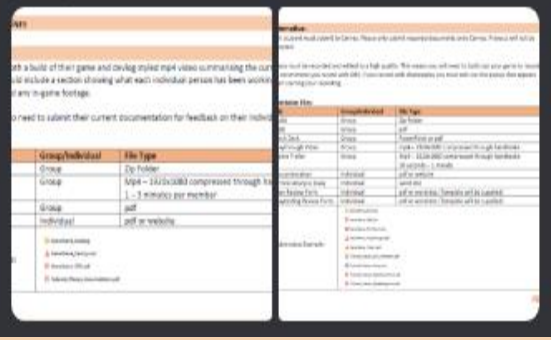
- devlog vid (can b edited during da week b4 friday)
- putting levels together (PRIORITY)
- playtest
- kill counter (Aurelia)

yuh (edited)

nadddine. 07/31/2024 10:28 PM

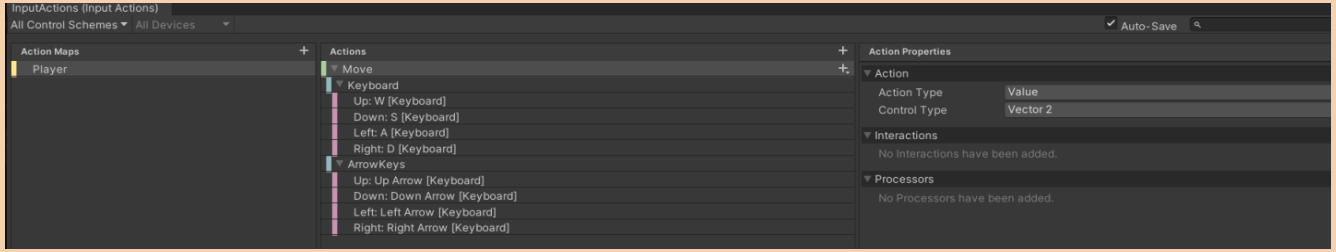
SUBMISSION DEETS

(edited)



Every week, I set up a to-do list of goals we need to achieve so that our group can keep on track.

Week 2: Coding Movement and local multiplier




```
public class MoveWASD : MonoBehaviour
{

    Rigidbody2D Rb;
    public float speed = 5f; // Default speed value
    float MovementX = 0;
    float MovementY = 0;

    void Start()
    {
        Rb = GetComponent<Rigidbody2D>();
    }

    void Update()
    {

        if (Input.GetKey(KeyCode.W))
        {
            MovementY = 1;
        }
        else if (Input.GetKey(KeyCode.S))
        {
            MovementY = -1;
        }

        if (Input.GetKey(KeyCode.A))
        {
            MovementX = -1;
        }
        else if (Input.GetKey(KeyCode.D))
        {
            MovementX = 1;
        }

        Vector2 movement = new Vector2(MovementX, MovementY).normalized;
        Rb.velocity = movement * speed;
```

Summary: Made a movement script for all 3 players. DPS: MoveArrows.cs, Healer: MoveWASD.cs, and Support: PS4Controller.cs.

In class, we were taught how to create controls for a multiplayer game using the input system. But since I wasn't there due to a driving test, I had to find tutorials to learn how to use the input system. Initially we decided for our controls to be “WASD”, “Arrow Keys” and the left joystick of a PS4 controller. I was having trouble making WASD and Arrow Keys work on the same keyboard, when I tried to put those controls into the input system it would both control one player only.

I tried changing the value of the action type, changing controlType, renaming the Actions to different names. I decided to code the movement controls manually for our Local Multiplayer. The issue was the Input system was controlling one GameObject with WASD and ArrowKeys.

Going back to the basics, I added variable references at the top covering movement and speed. T hen applied a Rigidbody2D reference as well, so C# knows that we're coding for a 2D game. In the Start function, I wrote:

`Rb = GetComponent<Rigidbody2d>( );`

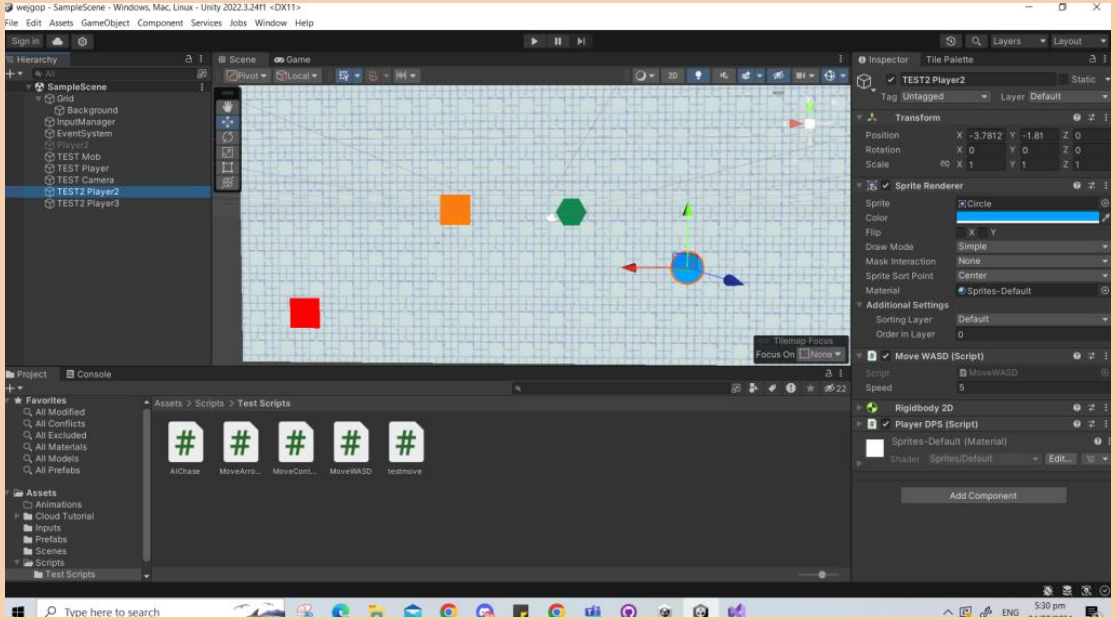
This allows the script to find the Rigidbody component that's attached to the GameObject so it can be used later. Then, I wrote if statements in our update function so that movement is being called every second in the game.

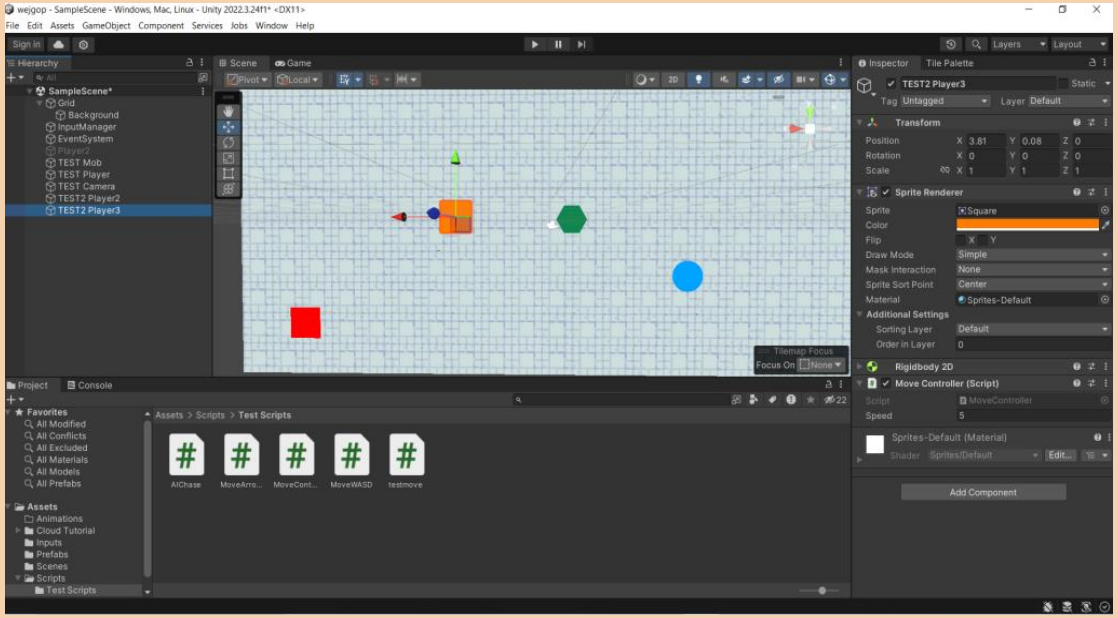
The statements are saying, if this specific key is pressed, then Movement X/Y will be set to 1/-1. The value 1 represent moving up and forward. The value 1 represents moving down and backward.

The code at the bottom with the if statements, under the Update function is saying I want to move in this direction at this speed(whatever the value of speed is set at). .normalized makes the movement consistent, allowing you to remain at the same speed no matter what direction you're going at. Vector 2(MovementX, MovementY) is creating a direction using the values in the bracket.

The movement script is written the same way for all 3 players.

Week 2: Coding Movement and local multiplier





SUMMARY: Setting up the player gameobjects in Unity. Adding list into AIChase.cs.

Here are screenshots of how the players are set up on the Unity Editor. the players all have their own scripts specific to their class. I attached them all to the 3 characters. Since the character sprites weren't ready during this time, I created temporary placeholder sprites and colour coded them. The blue circle is Wylla, the orange square is Cerwyn and the green hexagon is Harry.

The red square is an AI mob set by Aurelia. We worked together to combine the scripts so that the mob would follow the players I made.

All we had to do was create a list in the AIChase.cs script:

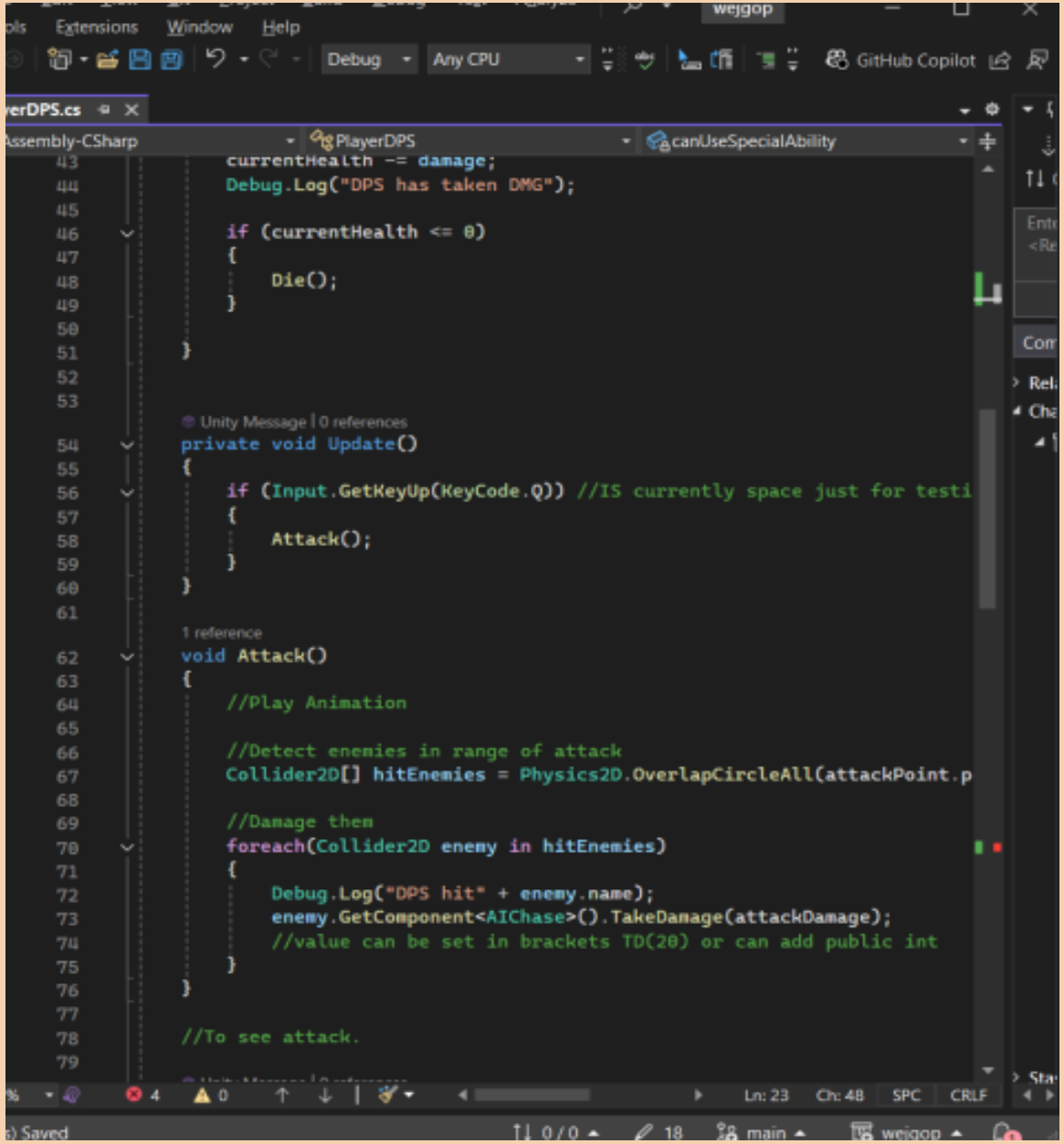
```
public List<GameObject> players;
```

Then on the mob game objects that the script was attached to, I dragged the player gameobjects into the list. Which allowed the AI to detect the players.

This was a good solution in the early stages of the game but as the game got more complex; we decided to call game objects in code using tags and layers. Each player game object is on the "Player" layer.

I think something we also could have done was put the movement all onto the PlayerParent.cs. But I chose not to, at this time because I was focused on getting the multiplayer controls to just work so that next week, I'd be able to add attacks into the game.

Week 3: Coding Base Attacks



SUMMARY: Created PlayerDPS.cs, PlayerHealer.cs and PlayerSupport.cs. Coded base attacks for the players.

Aurelia coded the base attack for Harry. (DPS)

From my understanding, she's calling the Attack() function using an if statement covering the button "Q." She sent a tutorial covering AI mobs to me later; which helped me more to understand the Attack() function.

Collider2D[] hitEnemies = Physics2D.OverlapCircleAll(attackPoint.position, attackRange, enemyLayers);

I think this tells me that it's detecting all enemies within a specified circular area around a designated point. That Physics2D.OverlapCircleAll is used to find enemies within that circle, putting them into a list for later use. I know attackRange covers the radius size and enemyLayers is there to show what layer we need to check to make the code function.

```
foreach(Collider2D enemy in hitEnemies)
{
   Debug.Log("DPS hit" + enemy.name);
   enemy.GetComponent<AIChase>().TakeDamage(attackDamage);

}
```

The log is there to tell us if the player game object is damaging the mob game object. Also, it's getting the behaviour from AIChase component attached to the mob game object and tells it to take damage based off of the attackDamage value.

After processing the code, I copy and pasted the functions covering attack into the PlayerHealer.cs and the PlayerSupport.cs. then set up the keycode for Wylla's base Attack as (') and charged attack as (Enter). Cerwyn's base attack is (Square) and charged attack as (Circle). Wylla's keybinds were originally "KeypadEnter" and "KeypadPeriod" but I changed it because it's catered to left-handed peopled and may feel unnatural for right-handers. So, I think switching to controllers will be good later on.

I think I'm going to change Cerwyn's attack buttons to the right and left triggers later on too because it's the buttons that are normally assigned for attacking when playing on PS4. Depending on what feedback we get for playtesting, I'm keeping the controls as is for now but if time comes and we find that it's better to have 3 PS4 controlled players then we'll change it.

Week 3: Coding Charged Attacks

SUMMARY: Updated PlayerDPS, PlayerHealer and PlayerSupport scripts so that they have specific charged attacks

```
1 reference | notaurelie, 14 days ago | 2 authors, 6 changes
void Attack()
{
    //Play Animation

    //Detect enemies in range of attack
    Collider2D[] hitEnemies = Physics2D.OverlapCircleAll(attackPoint.position, attackRange, enemyLayers);

    //Damage them
    foreach(Collider2D enemy in hitEnemies)
    {
        Debug.Log("DPS hit" + enemy.name);
        enemy.GetComponent<AIChase>().TakeDamage(attackDamage);
        //value can be set in brackets TD(20) or can add public int

        totalDamageDealt += 40;

        if (totalDamageDealt >= damageThreshold)
        {
            canUseChargedAbility = true;
            skillImage.SetActive(true);
            Debug.Log("DPS Charged ability is now available!");
        }
        // notifies quest system
        //GoBattle();
    }
}
```

```
1 reference | - changes | -authors, -changes
void Attack()
{
    // Detect enemies in range of attack
    Collider2D[] hitEnemies = Physics2D.OverlapCircleAll(attackPoint.position, attackRange, enemyLayers);

    // Damage them
    foreach (Collider2D enemy in hitEnemies)
    {
        //Debug.Log("Healer hit " + enemy.name);
        enemy.GetComponent<AIChase>().TakeDamage(attackDamage);

        totalDamageDealt += attackDamage;

        if (totalDamageDealt >= damageThreshold)
        {
            canUseChargedAbility = true;
            skillImage.SetActive(true);
            // Debug.Log("Healer Charged ability is now available!");
        }
    }
}
```

```
    AIChase aiChase = enemy.GetComponent<AIChase>();
    if (aiChase != null)
    {
        aiChase.TakeDamage(attackDamage);
        aiChase.ApplyStun();
        totalDamageDealt += attackDamage; // Track the total damage deal

        // Check if the threshold is reached
        if (totalDamageDealt >= damageThreshold)
        {
            canUseChargedAbility = true;
            skillImage.SetActive(true);
            Debug.Log("Charged ability is now available!");
        }
    }
}

if (totalDamageDealt >= damageThreshold)
{
    foreach (Collider2D enemy in hitEnemies)
    {
        AIChase aiChase = enemy.GetComponent<AIChase>();
        if (aiChase != null)
        {
            aiChase.ApplyStun();
        }
    }
}
```

HARRY'S CHARGED ATTACK

The way it works is that there's a damage threshold that needs to be met before the charged attack is available for use. damageThreshold is set at 120. attackSkill value is set to 100. attackSkill is the amount of damage they deal when they do their charged attack. The int was originally called "chargedAttackdamage" but it came up with errors because attackSkill was the value set in the references, so we just changed the code from referencing chargedAttackdamage to attackSkill. I think either way it would have been fine. We decided to leave it. The values will be subject to change depending on future game development. The attackDamage int is currently set at 40. I set the charged attack button for our DPS to be "E" for now, but should we choose to use controllers later it'll be the right trigger on a PS4 controller. Aurelia and I worked together to find the issue of the code. Being in call allowed her to understand how the code was being made.

WYLLA'S CHARGED ATTACK

I set up Wylla's charged attack the same as Harry's the only difference is she's supposed to attack long range. This has not been playtested properly yet. I think it'll be good to have charged attack animations done soon so there's a visual representation when the healer does their charged attack. The good news is her charged attack does work and if coding her charged attack to be long range doesn't work, I'll just change the damage threshold for her, so she doesn't do as much damage as our DPS to balance out our gameplay. attackDamage = 15. AttackSkill = 30. DamageThreshold = 120;

```
totalDamageDealt += attackDamage;


if (totalDamageDealt >= damageThreshold)
{
    canUseChargedAbility = true;
    skillImage.SetActive(true);
    // Debug.Log("Healer Charged ability is now available!");
}
```

Attack() Example:
+= line of code is saying that attackDamage is adding to the totalDamageDealt. Which keeps a running total of the damage being dealt.

CanUseChargdAbility= true; is a flag which allows the gameobject to use the charged attack The same goes for skillImage.
Also, CanUseChargedAbility is set false by default. (Referenced at top.)

Basically, this code is saying that the charged attack can be activated once you deal the right amount of damage.

CERWYN'S CHARGED ATTACK

Cerwyn's charged attack which was a bit more complicated because I had to update our "AIChase" script to be able to stun the mobs when Cerwyn's charged attack was active. attackDamage = 40. attackSkill = 100. damageThreshold = 120. AIChase aiChase = enemy.GetComponent<AIChase>(); finds the component in the mob game object. Then I made 2 if statements, within if (aiChase != null), aiChase.TakeDamage(attackDamage); and aiChase.ApplyStun(); both call methods from the aiChase script.

Week 3: Coding Charged Attacks 2

```
public override void PerformAbility()
{
    base.PerformAbility();

    {
        // Play Charged Attack Animation


        Collider2D[] hitEnemies = Physics2D.OverlapCircleAll(attackPoint.position, attackRange, enemyLayers);


        foreach (Collider2D enemy in hitEnemies)
        {
            Debug.Log("DPS Charged attack hit" + enemy.name);
            enemy.GetComponent<AIChase>().TakeDamage(attackSkill);
            // Reset ability
            canUseChargedAbility = false;
            skillImage.SetActive(false);
            totalDamageDealt = 0;

        }
```

SUMMARY: Updated PlayerDPS, PlayerHealer and PlayerSupport scripts in the Performability() function.

PERFORMABILITY () for all 3players

In the Performability() function for all 3 players, the code is set up the same way. A debug log is set to check if the charged attack has been used under foreach (Collider2D enemy in hitEnemies), finds each enemy in the radius. Collider2D[] hitEnemies = Physics2D.OverlapCircleAll(attackPoint.position, attackRange, enemyLayers); Finds all enemies that have a specific attackPoint, attackRange and enemyLayer. CanUseChargedAbility is turned off again here, so it resets the charged attack. As well as the skillImage and totalDamageDealt is back to 0. base.Performability(); is saying to do the functions from the parent class first before doing these. The function is also set as public override void so that I could add this method to a class that already exists.

```
public override void PerformAbility()
{
    Debug.Log("Support used charged ability!");


    Collider2D[] hitEnemies = Physics2D.OverlapCircleAll(attackPoint.position, attackRange, enemyLayers);
    foreach (Collider2D enemy in hitEnemies)
    {
        AIChase aiChase = enemy.GetComponent<AIChase>();
        if (aiChase != null)
        {
            aiChase.TakeDamage(attackSkill); // Deal charged attack damage
            aiChase.ApplyStun();
            skillImage.SetActive(false);
        }
    }


    totalDamageDealt = 0;
    canUseChargedAbility = false;
}
```

PERFORMABILITY () for all 3players

Cerwyn's Performability() is slightly different from the other two.
It just has another if statement under foreach (Collider2d enemy in hitEnemies)

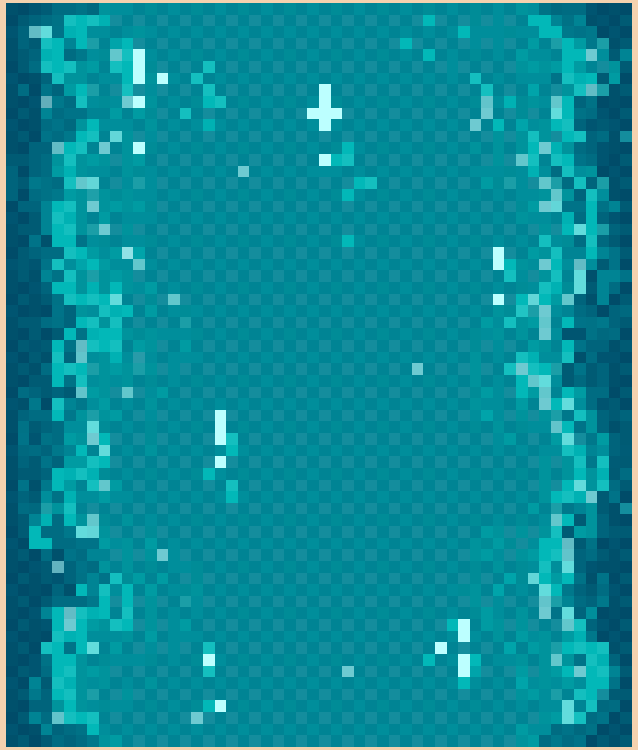
```
foreach (Collider2D enemy in hitEnemies)
{
   AIChase aiChase = enemy.GetComponent<AIChase>();
   if (aiChase != null)
   {
      aiChase.TakeDamage(attackSkill); // Deal charged attack damage
      aiChase.ApplyStun();
      skillImage.SetActive(false);
   }
}
```

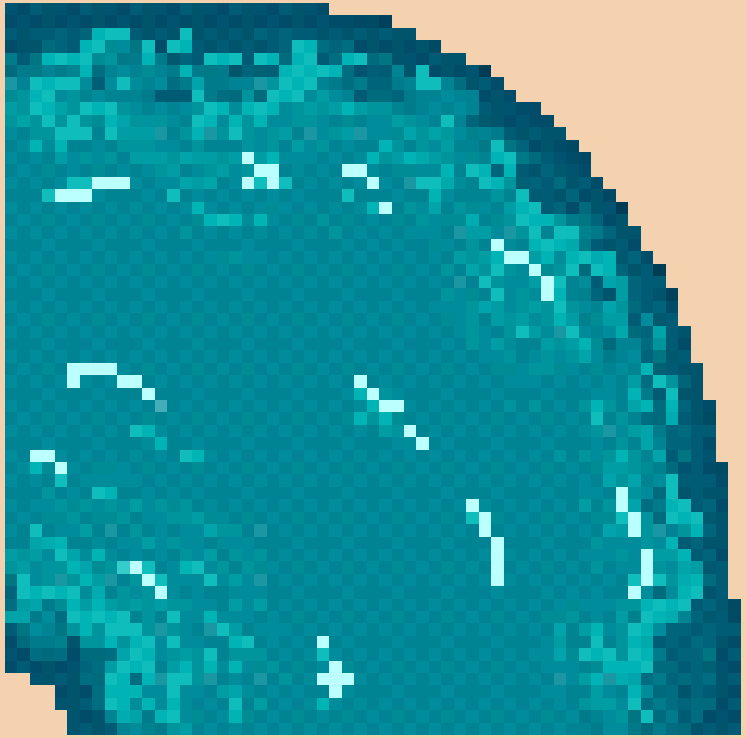
So basically, aiChase.TakeDamage(attackSkill) damages enemy when charged attack key is pressed. AiChase.ApplyStun() stuns the enemy when the charged attack is pressed. SkillImage.The rest of the code shown in the screenshot works the same way as the other Performability() in the PlayerDPS.cs and PlayerHealer.cs scripts.

Week 3: Pixel Art

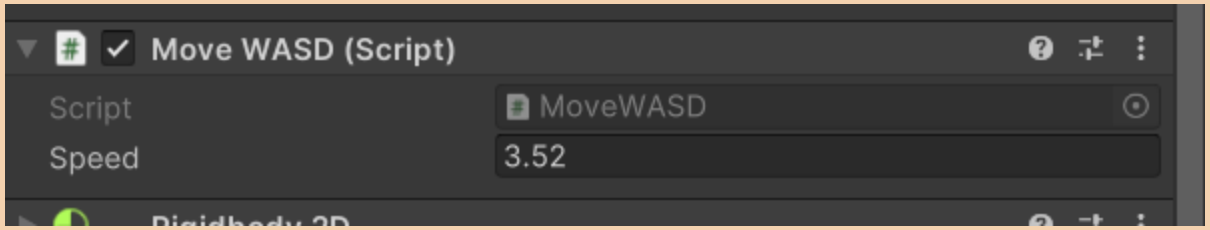
RIVER

The only sprites I made for Hallowed Three was a river. After discussion with my group abou how we wanted to create the map design; I asked them if they wanted it as part of a tile set so that they can put the river together in Unity or if they wanted on big river to add directly into the map. My teammate said probably best for the river to be broken up into tiles so we could change the direction of the river later if we wanted to. I aimed to finish this as soon as possible so that my teammate could import the river nto Unity. He made a temporary placeholder river to mark where the river should be on the maps. I followed the colour palette of our game and made the river this shade of blue that's almost like turquoise to add to the spooky and ghastly atmosphere of the game. To animate these, I used the river mood board I made in our group's Pinterest board (Link in Group Documentation).





Week 4: Coding Passive Abilities




```
1 reference
private void HealAlliesInRange()
{
    // finds all GameObjects within the healing radius
    Collider2D[] allies = Physics2D.OverlapCircleAll(transform.position, healingRadius);

    foreach (Collider2D ally in allies)
    {
        if (ally.CompareTag("Ally")) // checks for tag
        {
            HealthBar allyHealthBar = ally.GetComponent<HealthBar>();
            if (allyHealthBar != null)
            {
                // increases ally's health but won't exceed the max
                int newHealth = Mathf.Min((int)allyHealthBar.slider.maxValue, (int)allyHealthBar.slider.value + healingAmount);
                allyHealthBar.SetHealth(newHealth);
                //Debug.Log("Healed " + ally.name);
            }
        }
    }
}

1 reference
private IEnumerator PassiveHealing()
{
    while (true)
    {
        HealAlliesInRange();
        yield return new WaitForSeconds(healingInterval);
    }
}
```

I haven't coded Cerwyn's passive ability because the quest system needs to be done first before this.

SUMMARY: Updated MoveWASD.cs and Updated PlayerHealer.cs,

HARRY PASSIVE ABILITY

Following the Game Specifications workflow, I made our DPS have more speed for his passive ability compared to Wylla and Cerwyn. They're just slightly faster. Speed value is set to 3.52 for now. This speed variable is referenced in the MoveWASD.cs script.

WYLLA'S PASSIVE ABILITY

Once my teammate made the health bars and attached them to the player game objects. I ran into some issues with Wylla. My teammate tried to help; Wylla did heal but the players were susceptible to dying after being healed. Turns out the problem was that the health bar scripts weren't attached to the other players, tagged as "Ally" that's why the healing function wasn't working so that's what I did - it also fixed the issue of Wylla doing immense damage to the mobs after healing her allies so now the gameplay is a lot more balanced. She currently heals 20 per second and her healing radius is at 2.

The way this code works is that Wylla has a healing radius (collider) and basically if the players in the party are tagged as Ally, the HealingAlliesRange() will function for them. HealingAlliesRange() and PassiveHealing() calls HealingAlliesinRange() repeatedly every healingInterval seconds. OnDrawGizmos() is made so that the collider is visible showing the healing radius.

It took me a while to understand this line:
int newHealth = Mathf.Min((int)allyHealthBar.slider.maxValue, (int)allyHealthBar.slider.value + healingAmount);

allyHealthBar.SetHealth(newHealth);

From my understanding, (int) allyHealthBar.slider.maxValue looks at the max value a gameobject tagged ally can have. It's a bit different for each player. Wylla's maxHealth is 2000, Harry's is 1000 and Cerwyn's maxHealth is at 1000 too. (int)allyHealthBar.slider.value + healingAmount takes the current health and adds healing to it. If ally has 50hp left and Wylla heals 20 every second. It'd be 50 plus 20 totalling to 70hp. Math.Min((int) I think this is used to compare new health with max health so like if an ally gets healed at 80hp ,and the max hp is 100, it'll stays as 80hp because it's less than the max health. If an ally gets healed 30hp more than 100hp. It'll stay at 100hp because that's the max hp in this example. allyHealthBar.SetHealth(newHealth); keeps track of the health updates to an ally health bar visually.

The private IEnumerator PassiveHealing() function is basically saying while true allows the code within this function to loop until it is stopped. It just keeps passively healing allies within the radius of Wylla. HealAlliesInRange(); is calling the function. yield return new WaitForSeconds(healingInterval) basically waits for a few seconds, healingInterval = 1. When time ends it'll loop HealingAlliesInRange() again. The private IEnumerator PassiveHealing() function allows this to keep running over time.

Week 4: Coding the Main Menu and the Pause Menu

SUMMARY: Created MainMenu.cs, Updated PauseMenu.cs, fixed buttons

MAIN MENU

I learned my lesson, from my last assignment, and made sure to attach the main menu script to a game object before attaching it to the onclick function so I'll be able to access the MainMenu.cs script. For the main scenes like cutscene to main scene I attached the code to the Play button, to transport player from Title Page to Cutscene and to a temporary Start button to transport the player from the cutscene to the main game.

```
public void PlayCutscene()
  {
     // loads scenes in order within the build settings
     SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
  }
```

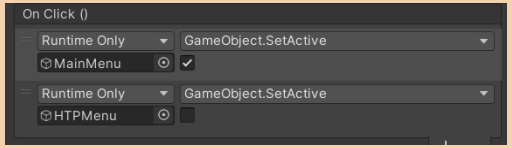
I chose to write the code like this, in order of the build settings for the public void PlayCutscene() so I don't have to keep coding all the names of the levels after the cutscene.

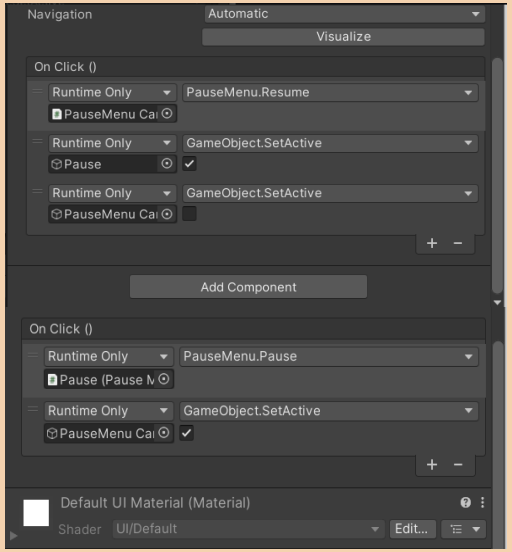
```
public void StartGame()
{
    SceneManager.LoadScene("game");
}
```

```
public void StartGame()
{
    SceneManager.LoadScene("Level_01");
}
```

The only changes I made was after I changed the names in the editor was change ("game") to ("Level_01") to help my teammates navigate all the levels in order.


```
+  {
8  +    public void PlayCutscene()
9  +    {
10 +        // loads scenes in order within the build settings
11 +        SceneManager.LoadScene(SceneManager.GetActiveScene().buildInde
       x + 1);
12 +    }
13 +
14 +    public void QuitGame()
15 +    {
16 +        Application.Quit();
17 +    }
18 +
19 +    public void StartGame()
20 +    {
21 +        SceneManager.LoadScene("Game");
22 +    }
23 +
24 +    public void BackToTitlePage()
25 +    {
26 +        SceneManager.LoadScene("Title Page");
27 +    }
28 + }
```





PAUSE MENU

I did not make them separate scenes for options menu and pause menu like in my Space Invaders Remake from semester 1. I just used the Gameobject.SetActive function under the Onclick component within the inspector as true or false and attached the script to the canvas the buttons are in. My teammate coded the Pause Menu script, and I helped her set it up on the editor using my knowledge from making the main menu multiple times. The pause button in Level_01 broke later on too but all that needed to happen was the GameObject.SetActive needed to be checked on the right gameobjets which were ticked for Pause, when pressing the back button and PauseMenuCanvas when pressing the Pause button.

Week 5: Quest System, Teleports

SUMMARY: Updated QuestGoal.cs, Created Teleport.cs, added new scenes

QUEST SYSTEM

Continued trying to fix the quest system with Aurelia. The issue was that after the quest is activated the mob wouldn't get registered as currentAmount so the quest couldn't be completed. I tried to register the mobs by referencing their tag Enemy and by their layer Mob, but it didn't work. Then I tried to attach the script to an NPC (GameObject) because I noticed that it was attached to a button and tried to set the quests like how it was last year for the 2.5 adventure game, where you attach the scripts to game objects and call them in the "OnClick" method in the editor. It didn't work either because the quests weren't set up as "MonoBehaviour" scripts.

Brooke made us realise that there was nothing wrong with our code. The problem lay in the editor. We just didn't have the scripts attached to the mobs so that's why the quests never got completed. So that meant we could have referenced the mob game objects by their layer or tag and the code would have still worked.

TELEPORTS

Firstly, I made new scenes and cleaned up the Unity editor so that it's organised. Before I coded the teleports for the different levels. I also created the rest of the levels too for efficiency later when we start to put our assets and scripts together.

Then I used on trigger colliders to teleport our players. Mobs don't get teleported because they have their own tag separate to the player. The script initially only called the tag Ally but after the healing zone got implemented in Week 6, I changed Wylla's tag from Ally to Wylla since Wylla has her own Healing zone, the gravestone, if I didn't change her tag the healing zone would have healed everyone and we made the decision that wouldn't be the case.

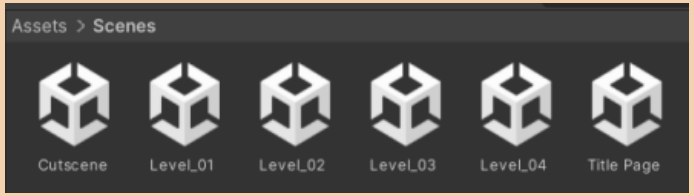
```
public bool IsReached()
{
    return currentAmount >= requiredAmount;
}
```

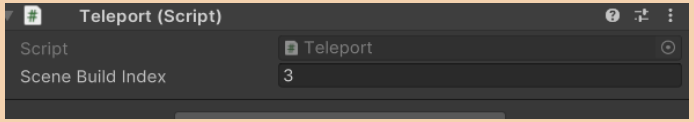
```
private void OnTriggerEnter2D(Collider2D other)
{
    print("Trigger Entered");

    if(other.tag == "Ally")
    {
        //player has entered so move level
        print("Switching Scene to" + sceneBuildIndex);
        SceneManager.LoadScene(sceneBuildIndex, LoadSceneMode.Single);
    }
```

```
private void OnTriggerEnter2D(Collider2D other)
{
    print("Trigger Entered");

    if (other.CompareTag("Ally") || other.CompareTag("Wylla"))
    {
        //player has entered so move level
        print("Switching Scene to" + sceneBuildIndex);
        SceneManager.LoadScene(sceneBuildIndex, LoadSceneMode.Single);
    }
}
```

So, to call both tags, I changed if(other.tag == "Ally") to if(other.CompareTag("Ally") || other.CompareTag("Wylla")) so that mobs still can't enter the teleport and Wylla is the only onee being healed at the gravestones scattered all around the levels. || means or and checks if condition is true. Other.CompareTag is checking if the gameobject has the specific tag on them.

print("Switching Scene to" + sceneBuildIndex); SceneManager.LoadScene(sceneBuildIndex, LoadSceneMode.Single); Allows the scene to switch to the next one based off the Build Index. All my teammates have to do to make this work is write down the number of the next level, on the game object this script is attached to.



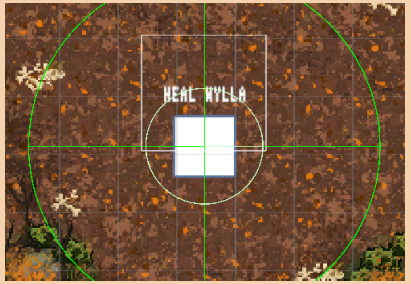


Week 6: Healing Zone

SUMMARY: Created HealingZone.cs, Created Parent GameObject ---HEALING ZONE---, Created healingzoneforwylla1 child gameobject

```
Unity Message | 0 references
private void OnTriggerEnter2D(Collider2D col)
{

    if (col.CompareTag("Wylla"))
    {

        StartCoroutine(HealingCoroutine(col.gameObject));
    }
}

Unity Message | 0 references
private void OnTriggerExit2D(Collider2D col)
{

    if (col.CompareTag("Wylla"))
    {
        StopCoroutine(HealingCoroutine(col.gameObject));
    }
}
```

```
private IEnumerator HealingCoroutine(GameObject ally)
{
    while (true)
    {
        HealAlly(ally);
        yield return new WaitForSeconds(healingInterval);
    }
}

1 reference
private void HealAlly(GameObject ally)
{

    if (ally.CompareTag("Wylla"))
    {

        HealthBar healthBar = ally.GetComponent<HealthBar>();
        if (healthBar != null)
        {
            // Increase ally's health, but do not exceed max health
            int newHealth = Mathf.Min((int)healthBar.slider.maxValue, (int)healthBar.slider.value + healingAmount);
            healthBar.SetHealth(newHealth);
            Debug.Log("Healed " + ally.name + " to " + newHealth);
        }
        else
        {
            Debug.LogWarning("HealthBar component not found on " + ally.name);
        }
    }
    else
    {
        Debug.LogWarning(ally.name + " is not tagged as 'Wylla'");
    }
}
```



HEALING ZONE

I knew what I wanted the function to be: Since Wylla can't heal herself in game but can heal others and the Point of Death locations (gravestones) of all the MCS were just an easter egg to sight see. I decided that to make the gravestones as healing zones for Wylla. It balances out her gameplay and makes the P.O.D. spots interactable for her. To create this, I looked at the passive ability code, healing allies within her radius for reference. I put a temporary sprite to mark the gravestone in the map and created text for it too, helps my teammates understand what the white square does.

Made int healingAmount = 25, healingInterval = 1f, healingRadius = 3f.

The OnTriggerEnter2D (Collider2D col) and OnTriggerExit2D (Collider2D col) cover the function that when a gameobject with a specific tag enters and exits the collider; The HealingCoroutine begins which is basically the healing method for Wylla and Stops when the gameobject exits the collider.

private IEnumerator HealingCoroutine(GameObject ally) kept as while (true) will keep running until it's stopped. HealAlly(ally) calls the method HealAlly which doesn't heal the other allies, only Wylla, it is called this because Wylla is still an ally even if she's not tagged as one. The healingInterval for the healing zone is also at one second for now like the healing radius that surrounds Wylla's gameobject.

Under the HealAlly (GameObject ally) function, compare tag is used again to check that the ally being healed is tagged as Wylla. HealthBar healthBar = ally.GetComponent<HealthBar>(); is here again to track Wylla's health bar. if (healthBar != null) checks if wylla has the healthbar attached to her gameobject. Then the same line of code that was made for Wylla's passive ability to heal is used again.
int newHealth = Mathf.Min((int)healthBar.slider.maxValue, (int)healthBar.slider.value + healingAmount);
healthBar.SetHealth(newHealth); It's basically letting her heal without exceeding the max amount. Her max HP right now is 2000. Harry and Cerwyn only have 1000. These values work for now because Wylla needs the extra HP to stay alive before she reached the next gravestone. She also heals more for other compared to herself. Also, I may change the healing zone from giving her 25HP every second to 100HP every second to make sure the gameplay still matches the brief since each level should only be 1-2 minutes long.

Week 6: Score System

SUMMARY: Game Manager.cs made and attached to Game Manager game object. QuestGoal.cs updated under "EnemyKilled" function. Score Text added in ---UI--- .

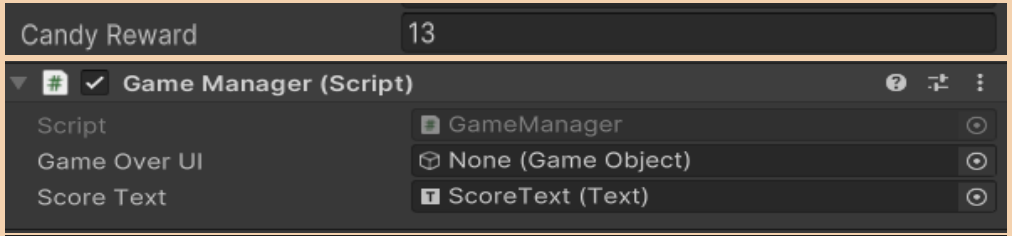
SCORE SYSTEM

We discussed as a group that we wanted a simple reward system going for now before making the quests more complex. My teammate worked on this first while I tested my own candy score system on a new project. She tried to implement a kill counter and call it "candy counter" instead. Though it was a good idea, it didn't work because she failed to reference the counter to the right scripts. When it was my turn to try to implement a score system into our game, I chose to utilise the knowledge of score systems I learned when I made a Space Invaders remake for our Atari project back in semester 1.

Candy Reward = 13 for story reasons.

In GameManager.cs, I put [SerializeField] private GameObject gameOverUI; [SerializeField] private Text scoreText; first. Now that i'm looking back, i'm going to delete the gameOver one because I'm going to make checkpoints to respawn at instead. It'd also ruin the balance of the gameflow if the players died and had to restart from the very beginning. public static GameManager Instance { get; private set; } was something new I learned. Instance is a variable that belongs to the entire game and since GameManager is shared everywhere; instance is being used here. {get; private set;} get means other parts of the game can access this Instance, which is GameManager and private set means that GameManager is the only one that can change or set the Instance.

if (Instance != null && Instance != this) and else { Instance = this; }: firstly he if statment is checking for any other Game Manager, called Instance and if there is another one it destroys it. The second line is saying that this is the right GameManager, use this.




```
public int score { get; private set; } = 0;

private void Awake()
{
    // ensure only one instance of GameManager exists
    if (Instance != null && Instance != this)
    {
        Destroy(gameObject);
    }
    else
    {
        Instance = this;
    }
}

private void Start()
{
    SetScore(score); //shows candy score display
}

// this updates score ui
public void SetScore(int score)
{
    this.score = score;
    scoreText.text = score.ToString().PadLeft(4, '0');
}
```

Week 6: Score System 2

SUMMARY: Game Manager.cs made and attached to Game Manager game object. QuestGoal.cs updated under "EnemyKilled" function. Score Text added in ---UI--- .

SCORE SYSTEM

SetScore(score); calls SetScore, a new function I made for updating the candy score. In public void SetScore(int score), within GameManager.cs. this.score = score; takes the number you assign and store the variable. scoreText.text = score.ToString().Padleft(4, '0'); converts the score into a string, meaning its visible ingame. PadLeft(4,'0') means score will show 4 digits all the time. So if score is set at 13, which it is, it'll show as "0013".

Under public void UpdateCandyScore(QuestGoal questGoal) has an if statement again to check if questGoal has given a quest, not empty. I think SetScore(score + questGoal.candyReward); says if quest is there, it'll add the candy reward value that was put in the hiearchary of the questGoal child gameobject and adds the scores to a total; SetScore. The debug.log warning is there to tell us if something goes wrong.

Then, I had to update the QuestGoal.cs to reference the GameManager. I only updated QuestGoal.cs because it covers the tasks and quest completion. private GameManager manager; is added and GameManager.Instance.UpdateCandyScore(this); is added under public void EnemyKilled(). GameManager.Instance reference the GameManager. UpdateCnadyScore(this) is telling us that UpdateCandyScore function in GameManager.cs is passing to this quest and it sends the info back to GameManager.

```
// updates score with "Candy Reward" in the wuest goal script
public void UpdateCandyScore(QuestGoal questGoal)
{
    if (questGoal != null)
    {
        SetScore(score + questGoal.candyReward);
    }
    else
    {
        Debug.LogWarning("QuestGoal is null in UpdateCandyScore!");
    }
}
```

```
if (IsReached())
{
    Debug.Log("Goal Reached!");
    // Handle quest completion
    teleport.gameObject.SetActive(true);
    Debug.Log("Teleport is now active!");
    quest.Complete();

    GameManager.Instance.UpdateCandyScore(this); // nadine added this thats it
}
```

Week 7: Planning for mid-semester break

FUTURE FUNCTIONS

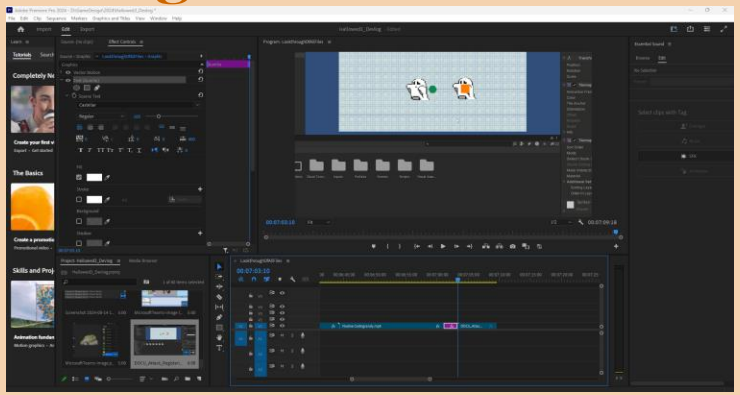
After testing out the game this week and receiving feedback for our playtest, I realised we needed more functions to balance the game. Functions like checkpoints scattered all over the map where players can respawn when they die. We considered other options like a game over scene but that will ruin the experience of the 2D Dungeon Crawler and create more frustration for the player. We also plan to use the candy to buy items that can help the player. I'm also going to code the movement for the players and the mobs so that they flip when they turn left or right as well as add their animations. We're also planning to add cooldown to the top of the health bar.

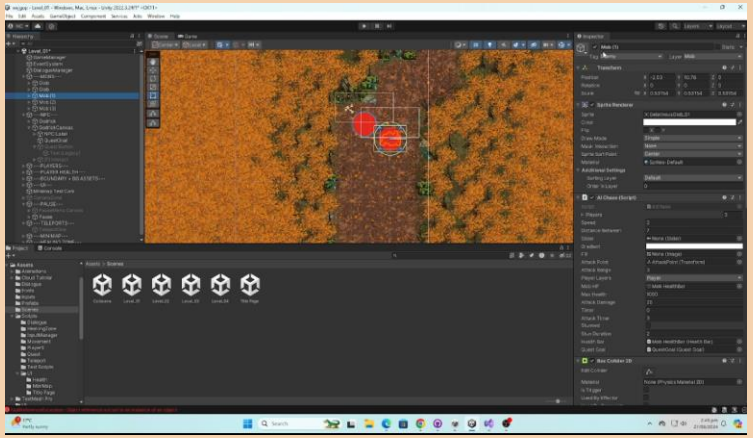
Our boss battle function is going to be one of our most important updates in the second half of semester. I'm also going to finally add cerwyn's passive ability. Our mechanic for Cerwyn's passive ability is going to change from being the inventory holder to lower the enemies attack for a 4 second interval which can only occur when enemies are within Cerwyn's radius. Then there'll be a 13 second cooldown before the passive ability starts up again.

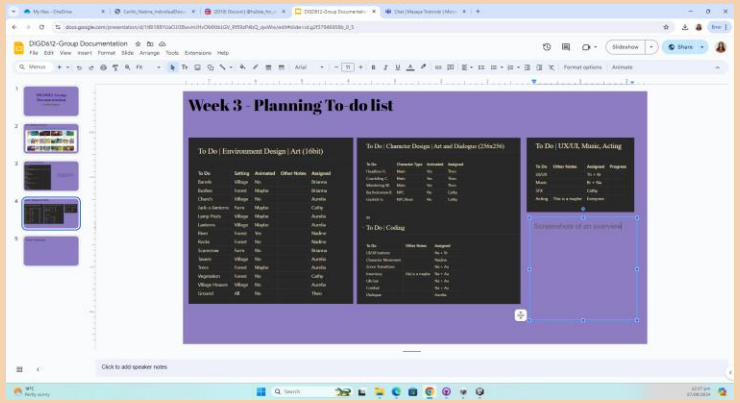
ESSAY

I plan to write the draft of my essay during the break since I have more free time by then. I've decided to either pick question 1 or question 4 because I found them very interesting. I ran my idea by Michaela and have decided to write about an inactive fandom that destroyed the development of an old popular game in ROBLOX.

Week 7: Devlog and GDD





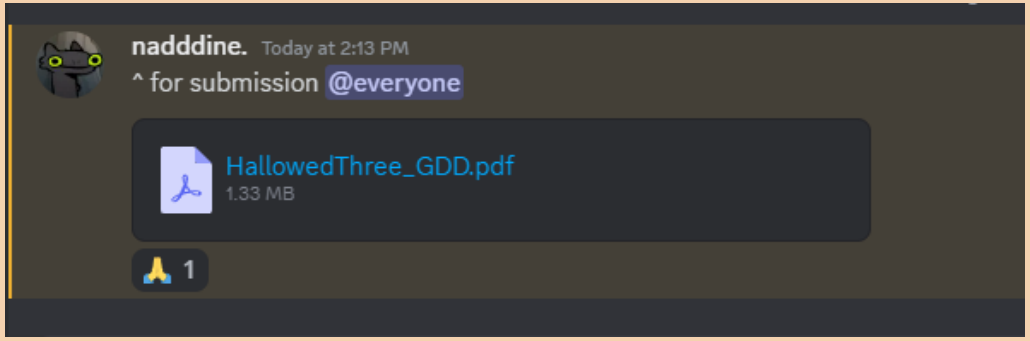


DEVLOG

I assisted Cathy in importing in videos for the Devlog video as well as setting up the premiere file for it. I set up a devlog script document for us to put our lines into as well.

GROUP DOCUMENTATION

I created the shared presentation file for our group documentation pdf needed for submission. It was a bit difficult finding evidence for each week since majority of the game was created in pairs during the previous weeks. Most of our group work was mostly working individually together in class and after hours. This documentation covers our planning, mood boards, brainstorms and playtests in a bit more detail.



Break: Respawn Checkpoints

SUMMARY: Created Checkpoint.cs, Created PlayerPos.cs, Created ---CHECKPOINTS--- Parent game object, created child gameobject "Checkpoint1", attached Checkpoint.cs to Checkpoint 1, Created tag for Game Manager "GM", Updated PlayerDPS.cs, PlayerSupport.cs, PlayerHealer.cs and added Respawn() function and new references.

```
ivate void Update()

  if (Input.GetKeyDown(KeyCode.Space))
  {
      SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
  }
```

```
Unity Message | 0 references
void Start()
{
    gameManager = GameObject.FindGameObjectWithTag("GM").GetComponent<GameManager>();
    transform.position = gameManager.lastCheckPointPos;
}
3 references
public void RespawnPosition()
{
    if (gameManager != null)
    {
        transform.position = gameManager.lastCheckPointPos;
    }
}
```

```
Unity Message | 0 references
void OnTriggerEnter2D(Collider2D other)
{
    if (other.CompareTag("Ally") || other.CompareTag("Wylla"))
    {
        gameManager.lastCheckPointPos = transform.position;
        Debug.Log("Checkpoint reached:" + gameManager.lastCheckPointPos);
    }
}
```

CHECKPOINTS

I started off by creating a game object called "---CHECKPOINTS---" then made a child game object called checkpoint 1 then attached the Checkpoint.cs onto checkpoint1. In GameManager.cs, I added public Vector2 lastCheckPointPos; which basically helps keep track of the x and y coordinates. In PlayerPos.cs script', I made a reference to GameManager, private GameManager gameManager and in private void Update() I had set an If(Input.GetKeyDown(KeyCode.Space)) line to respawn the character through the build index again. This was good at the time, but it didn't let our player respawn when the player died; it only lets them respawn when space is pressed.

So, I made changes: gameManager = GameObject.FindGameObjectWithTag("GM").GetComponent<GameManager>(); finds the game manager attached to the gameobject with the tag GM. transform.position = gameManager.lastCheckPointPos; Is asking where the last checkpoint was and moves your object to those coordinates. These lines were set in Start() so it runs automatically when game starts.

```
public void RespawnPosition()
{
   if (gameManager != null)
   {
      transform.position = gameManager.lastCheckPointPos;
   }
}
```

The RespawnPosition() is basically a method that checks if the GameManager is set up properly before it moves your object to the last checkpoint they entered.

In Checkpoint.cs, gameManager = GameObject.FindGameObjectithTag("GM").GetComponent<GameManager>(); again in the Start() to find the component. Then below it is the code covering the OnTrigger Enter Event, it has an if statement, to check for the game objects with the tag Ally and Wylla and if they do have these tags they'll be teleported to their last checkpoint position. The Compare tag lines are what I learned from making the healing zone for Wylla and gameManager.lastCheckPointPos = transform.porsition; is written the same way as it was in the Game Manager calling that same line and referencing it back to Game Manager.

Break: Respawn Checkpoints 2

SUMMARY: Created Checkpoint.cs, Created PlayerPos.cs, Created ---CHECKPOINTS--- Parent game object, created child gameobject "Checkpoint1", attached Checkpoint.cs to Checkpoint 1, Created tag for Game Manager "GM", Updated PlayerDPS.cs, PlayerSupport.cs, PlayerHealer.cs and added Respawn() function and new references.

CHECKPOINTS 2

Then I also had to make changes to the player scripts: the code here has been updated into PlayerDPS.cs, PlayerHealer.cs, and PlayerSupport.cs.

In the start() function for the player scripts I just put playerPosScript = GetComponent<PlayerPos>(); to reference the PlayerPos.cs script so that it can be used here.

Then In the Die() function I made a null if statement to check for the playerPosScript and just put playerPosScript.RespawnPosition();
Then Invoke ("Respawn", 1f); The if statement is calling the Respawn Position method from the PlayerPos.cs script and Invoke("Respawn,1f) means that after 1 second the Respawn method will run. It basically gives it a time interval between respawns.

After that, I created a new function called Respawn() which has this.gameObject.SetActive(true); GetComponent<Collider2D>().enabled=true;, currentHealth = maxHealth; and halthBar.SetHealth(currentHealth); which is all saying that when the game object respawns, the game object gets set active, the component Collider2D is actived, and their health bar returns to the max health.

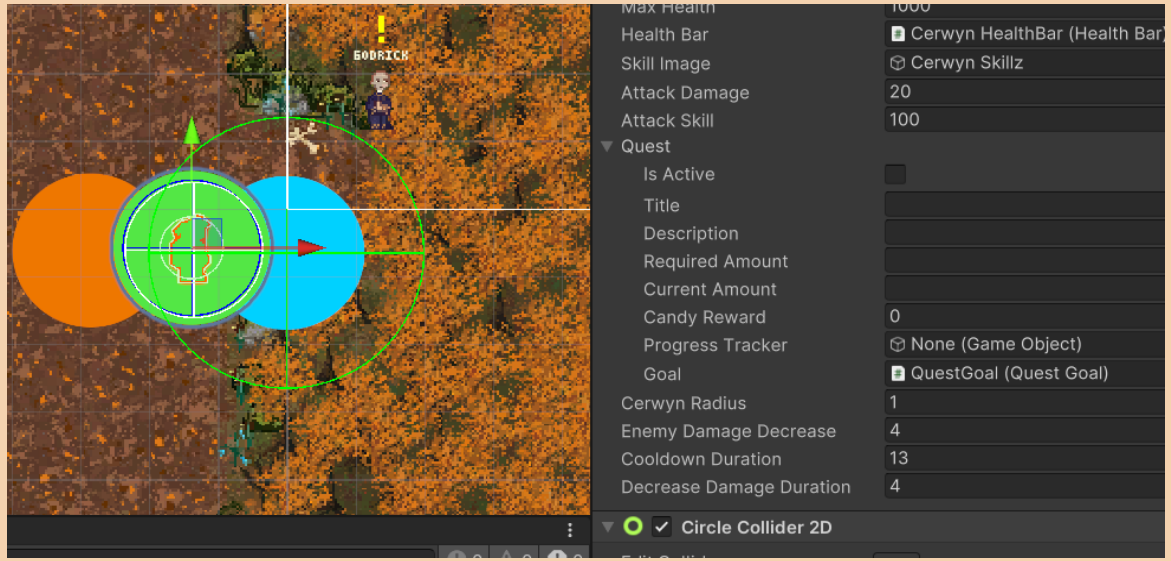
Making all these changes ensures that when the player dies after triggering the collider on the checkpoint, the player will respawn at the last checkpoint position and have their collider2D, sprite, health back as new. Then for my teammates, I dragged and dropped ---CHECKPOINTS--- into the Prefabs folder so that they can transport it easily to the next levels.


```
Unity Message | 0 references
void Start()
{
    currentHealth = maxHealth;
    healthBar.SetMaxHealth(maxHealth);
    playerPosScript = GetComponent<PlayerPos>();
    // references player pos script - nadine

}

2 references
```

```
    // nadine - made changes here
    if (playerPosScript != null)
    {
        playerPosScript.RespawnPosition();
    }

    Invoke("Respawn", 1f);
}

0 references
void Respawn() // nadine added this
{
    this.gameObject.SetActive(true);
    GetComponent<Collider2D>().enabled=true;
    currentHealth = maxHealth;
    healthBar.SetHealth(currentHealth);
}
```

Break: Cerwyn's Passive Ability

SUMMARY: Updated PlayerSupport.cs, Added DecreaseDamageAbility()



DECREASE ENEMY DAMAGE WITHIN RADIUS

Created public float cerwynRadius = 2f;
[SerializeField] int damageDecrease, public float cooldownDuration = 13f; and private bool cerwynAbilityOnCooldown = false; the bool isn't currently being used right now so it's currently just noted "//". I wanted to make sure the ability works first before I add in a cool down. Cerwyn's passive ability is: When enemies enter his radius, they do less damage, value is currently set as 4. I used what I learned during making Wylla's Healing radius to make Cerwyn's radius.

Under private void DecreaseDamageAbility(), I put int ogDamage = mobAI.attackDamage; It's a line I wanted to test, it basically a reminder about how much attack damage the mob should be doing. Collider2D[] enemies = Physics2D.OverlapCircleAll(transform.position, cerwynRadius); is a line of code that is referenced from PlayerHealer.cs in HealAlliesInRange(), it helps find specific game objects within their radius.

```
private void DecreaseDamageAbility(AIChase mobAI)
{
    int ogDamage = mobAI.attackDamage;
    Collider2D[] enemies = Physics2D.OverlapCircleAll(transform.position, cerwynRadius); // finds each enemy within cerwyn's radius

    foreach (Collider2D enemy in enemies)
    {
        if (enemy.CompareTag("Enemy")) // using tag to call the mobs
        {
            AIChase aIChase = enemy.GetComponent<AIChase>();
            if (aiChase != null)
            {
                ogDamage -= aiChase.attackDamage; // original damage - attackDamage in aichase script
                Debug.Log("cerwyn decreases enemy attack");
            }

        }
    }
}
```

Then under foreach (Collider2D enemy in enemies), I put an if statement, using CompareTag to check for the "Enemy" tag when calling mobs. Followed by a GetComponent <AIChase>; so, I can call variables for use from AI Chase.cs. ogDamage -= aiChase.attackDamage; is basically saying to minus the attackDamage, currently set at the value of 4, from the original damage, differs for each mob; can be changed in any of the mob's inspector.

Week 8: Cerwyn's Passive Ability 2

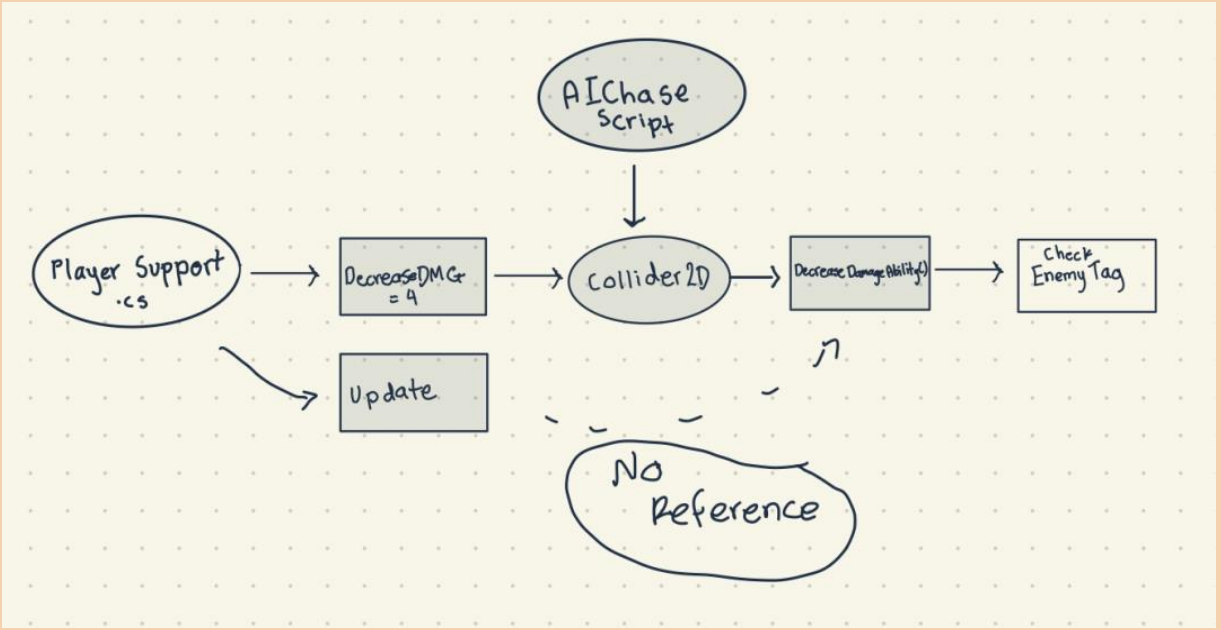
SUMMARY: Updated PlayerSupport.cs, Created mind map for breakdown, Changes to DecreaseDamageAbility()

DECREASE ENEMY DAMAGE WITHIN RADIUS 2

Unfortunately, the DecreaseDamageAbility() I created didn't work based off HealAlliesinRange() from PlayerHealer.cs, so I created a mind map to break down the function visually. The error became clear this way, yes, I was using variables from AIChase.cs but the DecreaseDamageAbility() wasn't referenced or called anywhere in the PlayerSupport.cs script.

On the first day back, Brooke showed me another way to reference the function. The screenshot below features the changes she made. I also got advised that Collider2D[] enemies = Physics2D.OverlapCircleAll is used a lot in this script, for better optimization, I'm going to combine the functions that are getting called with that specific line of code. The main changes that were made was (int dam) is being used instead of (AIChase mobAI) because we're already grabbing the component with AIChase aIchase = enemy.GetComponent<AIChase>(); so, it got changed to that which is a shortened name for damage. Then aiChase.attackDamage -= dam; replaced ogDamage -= aiChase.attackDamage; The new line is saying that whatever attackDamage is in AIChase will be decreased depending on the value that was set for dam. It's still set at 4 for testing purposes. Then the last thing I had to do was call the function in Update so that it's being referenced. Under Update() I put DecreaseDamageAbility(4). By doing this it references the function (4) is the value that int dam is set as.

By doing this, the function works but it decreases by 4 every second which is not ideal because it's making the players invincible since no damage is hitting them. So, I decided it's probably best to call DecreaseDamageAbility() elsewhere because what I want to happen is damage decrease should only be deducted once when they enter Cerwyn's collider and the mob's damage should revert to doing their original damage when they leave the collider.




```
1 reference
private void DecreaseDamageAbility(int dam)
{
    Collider2D[] enemies = Physics2D.OverlapCircleAll(transform.position, cerwynRadius);

    foreach (Collider2D enemy in enemies)
    {
        if (enemy.CompareTag("Enemy"))
        {
            AIChase aIChase = enemy.GetComponent<AIChase>();
            if (aIChase != null)
            {
                aIChase.attackDamage -= dam;
                Debug.Log("cerwyn decreases enemy attack");
            }

        }
    }
}
Unity Message | 0 references
```

Week 8: Cerwyn's Passive Ability 3

```
private void OnTriggerEnter2D(Collider2D collision)
{
    Collider2D[] enemies = Physics2D.OverlapCircleAll(transform.position, cerwynRadius

    foreach (Collider2D enemy in enemies)
    Debug.Log($"Collider {collision.name}");

    if (collision.gameObject.tag == "Enemy")
    {
        if (enemy.CompareTag("Enemy"))
        if (!globs.Contains(collision.gameObject.GetComponent<AIChase>()))
        {
            AIChase aIChase = enemy.GetComponent<AIChase>();
            if (aIChase != null)
            {
                aIChase.attackDamage -= damageDecrease;
                Debug.Log("cerwyn decreases enemy attack");
            }
```

Assets\Scripts\Movement\AIChase.cs

```
@@ -29,6 +29,8 @@ public class AIChase : MonoBehaviour
29 29
30 30
31 31       public int attackDamage = 5;
   32 +     public int decreaseDamage = 4;
   33 +     public bool attackDecreased = false;
32 34
33 35       //timer
34 36       public float Timer;
@@ -101,7 +103,10 @@ public class AIChase : MonoBehaviour
101 103            foreach (Collider2D player in hitPlayers)
102 104            {
103 105                Debug.Log("Mob hit " + player.name);
104     -              player.GetComponent<PlayerParent>().TakeDamage(attackDamage);
    106 +              //Short hand for if statement
    107 +              player.GetComponent<PlayerParent>().TakeDamage(attackDecreased == true ? attackDamage - decreaseDamage : attackDamage);
    108 +
    109 +
105 110              //value can be set in brackets TD(20) or can add public int
106 111          }
107 112      }
```

SUMMARY: Updated PlayerSupport.cs, added OnTriggerEnter2D and OnTriggerExit2D, Updated AIChase.cs

DECREASE ENEMY DAMAGE WITHIN RADIUS 3

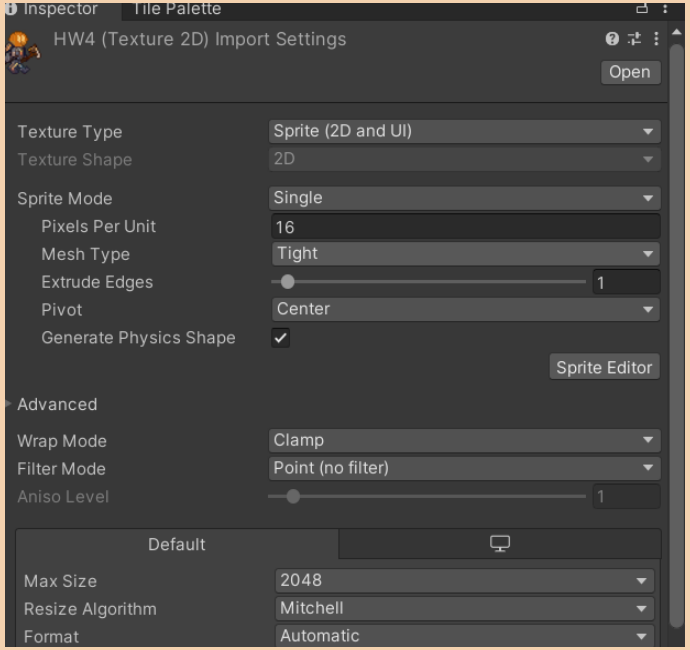
After making the decision to put the DecreaseDamageAbility() elsewhere. I put the code from the previous slide into OntriggerEnter2D and OnrtriggerExit2D functions. It no longer uses (int dam) and instead I tried to call my damageDecrease variable which was already set at 4 at the top of the script. In OntriggerEnter2D it says aIChase.attackDamage -= damageDecrease; and under OntriggerExit2D I put aiChase.attackDamage += damageDecrease; hoping that this would allow the AIChase attackDamage to go revert to its original attackDamage. But instead, these new changes made the attackDamage change whatever value the mob was set at to 4 and then add 4 every time they enter Cerwyn's radius. I also tried aiChase.attackDamage = 0 and 1 to see what would happen. Messing around either made the values go up or down to negative numbers.

Eventually, my lecturer came to help and created a list for the mobs that enter Cerwyn's radius. To be able to run this ability. It was a new Debuff() function that used references that were set in AIChase. Basically, all of this debuffs a glob when they enter Cerwyn's collider. Instead of setting up the code in PlayerSupport.cs. He set the code up in AIChase. Something I learned here was a shorthand for an if statement "player.GetComponent<PlayerParent>().TakeDamage(attackDecreased == true ? AttackDamage – decreaseDamage : attackDamage); Below is code that got added in PlayerSupport.cs to call this code from the AIChase.cs script.

```
public List<AIChase> globs = new List<AIChase>();

Void Debuff()
{
Foreach (AIChase glob in globs)
{
Glob.attackDecreased = true;
}

}

Then he added,
If(!globs.Contains(collision.gameObject.GetComponent<AIChase>()))
{
Globs.add(collision.GetComponent<AIChase>());
```

Week 8: Uploading Animations for Players

SUMMARY: Added Harry.anim, updated PlayerDPS.cs, Added HarryWalk.anim

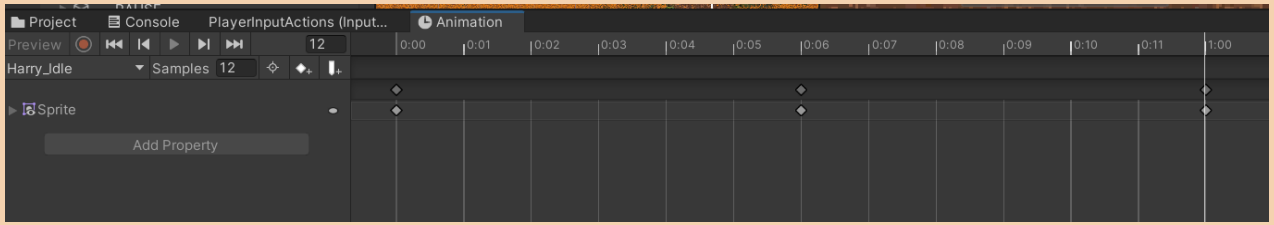


Idle and Walk

Firstly, I made sure that all the sprites were at 16 pixels per unity and the filter mode was Point(no filter) so that they'd be the same sizes, and the quality will be crisp. Then created animations for Harry's idle and Walk sprites that my teammate made.

Ran into an issue with the sprite sizes because they were at 100 pixels per unit the whole time for all the three players, so I had to delete my clone of the file and redownload it because I didn't want to push through the errors I made.

So, I tried to rescale the sprites to 16 again and checked if that changed their position again before updating Harry's Animator and adding Idle and Walk Sprites to him. Making his pixels per unit 16 made him tiny so I had to manually scale him up using the scale tool. His idle animations work in game now but they're too fast, so I had to space out the keyframes in his idle animation. I spaced the keyframes out, so they loop every half second before the difference was at 0.2 milliseconds so that's why his idle was looping quickly. Then I copy and pasted the first keyframe at 1 second so that his animation will loop more smoothly.



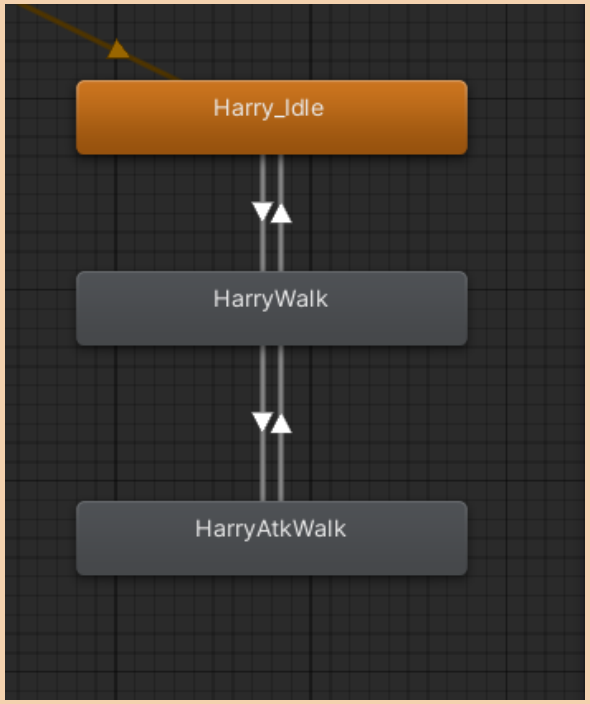

```
0 references
public void Walk()
{
    if (Input.GetKey(KeyCode.W) || (Input.GetKey(KeyCode.S) || (Input.GetKey(KeyCode.A) || Input.GetKey(KeyCode.D){
        update.isWalking = true;
    }
} }
```

Week 9: Player Movement Transition 1

SUMMARY: Updated PlayerInput.cs, added WASD into InputControls,

```
float MovementY = 0;

if (Input.GetKey(KeyCode.W))
{
    MovementY = 1;
    anim.SetBool("IsWalking", true);
}
else if (Input.GetKey(KeyCode.S))
{
    MovementY = -1;
    anim.SetBool("IsWalking", true);
}

if (Input.GetKey(KeyCode.A))
{
    MovementX = -1;
    anim.SetBool("IsWalking", true);
}
else if (Input.GetKey(KeyCode.D))
{
    MovementX = 1;
    anim.SetBool("IsWalking", true);
}

Vector2 movement = new Vector2(MovementX, MovementY).normalize
Rb.velocity = movement * speed;
```

```
{
    public int playerID;
    public Vector2 Movement;
    InputControls inputControls;
    public float speed = 3;

    public bool attackButton;
    public bool chargedAttackButton;
    public bool northButton;
    public bool westButton;
    public bool southButton;
    public bool eastButton;

    private Animator anim;
```



Transition to new movement controls

So, with the help of our lecturers, my teammate was able to change the player movement. What they did was implement the player movement into input manager. This was to help optimize our gameplay cause having two of people on the keyboard wasn't ideal anymore because we wanted gamers to be able to all see the screen and not be squished next to each other trying to fight for space on the keyboard. They also made an AssignInputs() function in PlayerInput.cs which covers all the buttons, controllers to be assigned to the player upon joining.

CODING WALK ANIMATIONS FOR THE PLAYERS

So, after importing in the animations that Theo made and setting them up in Unity, I created references to Harry's walk cycle animator first in MoveWASD.cs because that's where the movement methods are. I also made a bool "IsWalking" which is set to false because I don't want the walking to occur by default, I want it to be triggered by movement.

Under the if statements for "W", "A", "S" and "D" I put anim.SetBool("IsWalking", true) so that when any of these buttons are pressed, the walking animation should play because there's a transition going from HarryIdle to HarryWalk with the Boolean set as true and when the button is not getting pressed the bool returns to false because that's what it's set as in the transition from HarryWalk to Harry Idle.

This had to be optimised cause although it works for the keyboard player, it didn't work for the players with PS4Controller.cs when I added the same changes into the script because an error would occur when this script is attached to multiple game objects. The PS4 controller would control two players at once. So, I had to take a different approach to this.

Week 9: Player Movement Transition 2

SUMMARY: Updated PlayerInput.cs, added WASD into InputControls, updated moveWASD.cs

Transition to new movement controls 2

So, what I had to do was implement the boolean and animation code into the PlayerInput.cs script under private void Update(). After referencing the animator again my lecturer Brooke, taught me how to optimise the animation code that I put into MoveWASD.cs into this script. I had to make a reference for the sprite renderer too because I needed the sprite to change directions when turning left or right. The if statement is basically covering movement and using if (Movement.x > 0) to flip the sprite. We were accessing the "flipX" option under sprite renderer and flipping it in code by using true or false statements. Then I wrote anim.SetBool("IsWalking", false); under else and wrote it as true under if(canMove), it's basically saying that if you can move your player, IsWalking boolean will be turned on and will be turned off because it's under else.

I had to make references in Start() to call spriteRenderer, animator and I also made a new bool called canMove to make this all work. It's set as true at the start because we want the players to be able to move when the game begins. Also, the screenshot below is stopping the character when movement input is released but when it's performed the character moves in the direction based on the player's input.

```
Unity Message | 0 references
private void Update()
{
    if (canMove)
    {
        transform.position += (Vector3)Movement * Time.deltaTime * speed;
        if (Movement != Vector2.zero)
        {
            anim.SetBool("IsWalking", true);
            if (Movement.x > 0)
            {
                spriteRenderer.flipX = false;
            }
            else
            {
                spriteRenderer.flipX = true;
            }
        }
        else
        {
            anim.SetBool("IsWalking", false);
        }
    }

}
```

```
1 reference
private void MovementCanceled(UnityEngine.InputSystem.InputAction.CallbackContext obj)
{
    Movement = Vector2.zero;
}

1 reference
private void MovementPerformed(UnityEngine.InputSystem.InputAction.CallbackContext obj)
{
    Movement = obj.ReadValue<Vector2>();
}
```

```
// Start is called before the first frame update
Unity Message | 0 references
void Start()
{
    InputManager.instance.onPlayerJoined += AssignInputs;
    anim = GetComponent<Animator>();
    spriteRenderer = GetComponent<SpriteRenderer>();

    canMove = true;
```

Week 9: Player Attack Animations Code

SUMMARY: Updated PlayerInput.cs, Updated PlayerHealer.cs, Updated PlayerDPS.cs, Updated PlayerSupport.cs

```
inputControls.MasterActions.AttackButton.performed += ctx => attackButton = true;
inputControls.MasterActions.AttackButton.canceled += ctx => attackButton = false;

inputControls.MasterActions.ChargeAttackButton.performed += ctx => chargedAttackButton = true;
inputControls.MasterActions.ChargeAttackButton.canceled += ctx => chargedAttackButton = false;
```

```
public void UpdateAttackBindings()
{
    if (inputControls != null)
    {
        inputControls.MasterActions.AttackButton.performed += ctx => attackButton = true;
        inputControls.MasterActions.AttackButton.canceled += ctx => attackButton = false;

        inputControls.MasterActions.ChargeAttackButton.performed += ctx => chargedAttackButton = true;
        inputControls.MasterActions.ChargeAttackButton.canceled += ctx => chargedAttackButton = false;
    }
}
```

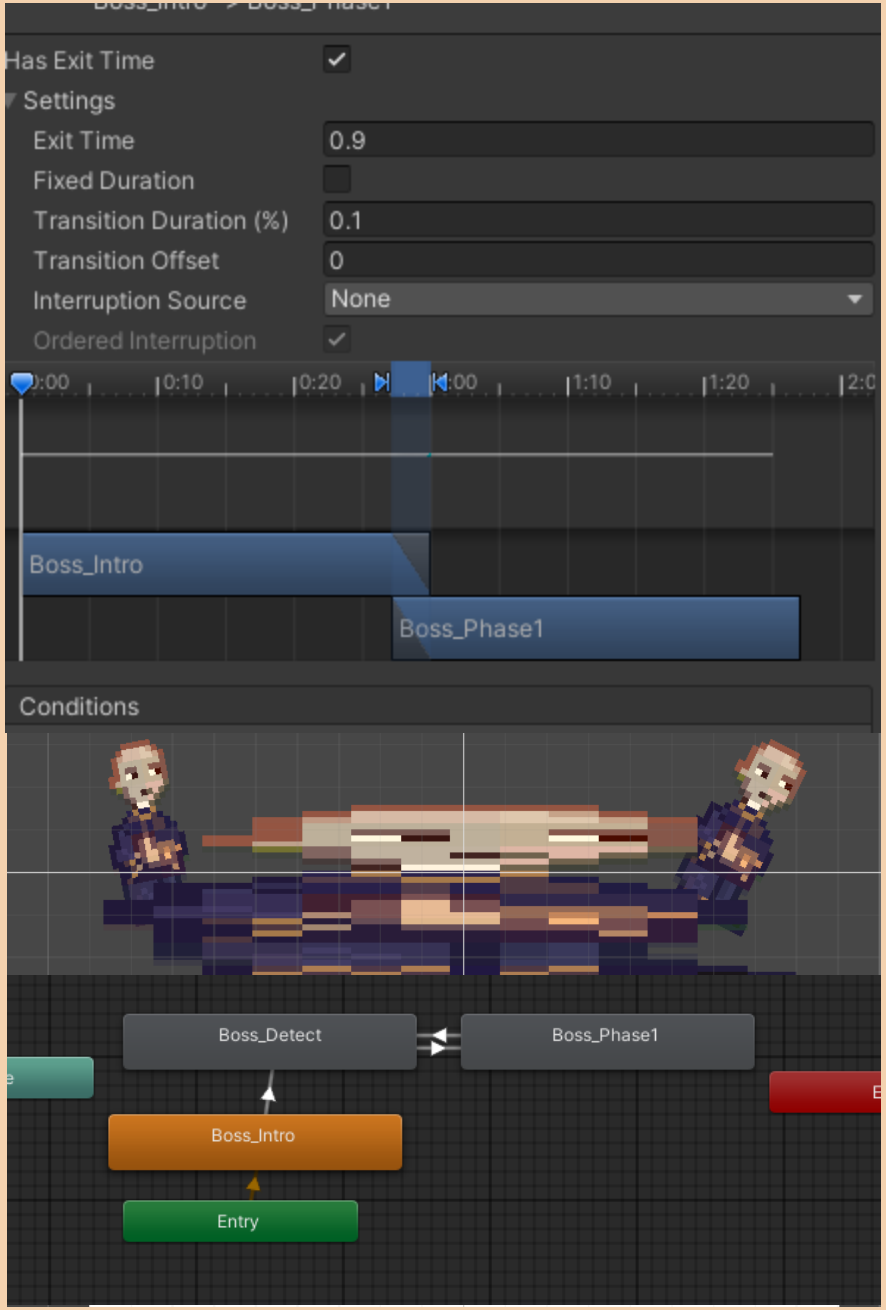
```
Unity Message | 0 references
private void Update()
{
    if (playerInput.attackButton) //changed for all three players are now referenci
    {
        Attack();
        playerInput.attackButton = false;
        anim.SetBool("IsAttacking", true);
    }
    else
    {
        anim.SetBool("IsAttacking", false);
    }

    if (canUseChargedAbility && playerInput.chargedAttackButton)
    {
        PerformAbility();
        playerInput.chargedAttackButton = false;
        anim.SetBool("IsChargedAttacking", true);
    }
    else
    {
        anim.SetBool("IsChargedAttacking", false);
    }

    //Call function

}
```

Base Attack and Charged Attack Changes

Firstly, I had to set the new attack buttons again through input manager. The controls are split between attackButton and chargedAttackButton So, on keyboard it's "E" for attack and "R" for charged attack. On any console controller it's LeftTrigger for charged attack and RightTrigger for normal attack. Under AssignInputs() I added inputControls.MasterActions.AttackButton.perform += ctx => attackButton = true and false for both charged attack and normal attack.

Then on PlayerDPS.cs under Update() I had to change the if statement again because it was following the original player movement scripts. It used to be (GetKeyUp...) in the brackets so I had to change it to reference the code in the playerInput.cs script. Referenced in the top and in Start() For example, shown in the third screenshot, covering attack, the if statement is now if(playerInput.attackButton) which gets the buttons that were assigned under Attack Button in the input manager. Then a boolean was made "IsAttacking", which was also added in the animator attached to this player. It's set as true when pressed and set as false when it's not being used. PlayerInput.attackButton = false under here because it resets the state after the buttons get pressed, so that it'll be ready for use when pressed again. Same method added for if(canUseChargedAbility...). I made the same changes for PlayerSupport.cs and PlayerHealer.cs too.

Week 9: Level_04 Boss Battle



SUMMARY: Updated Level_04 scene, Added Boss gameobject, Added child gameobjects under boss, Added Background parent game object, Added ---PLAYERS--- prefab to scene, Made Boss Battle Script Folder, Made Boss.cs, Made Boss_PhaseTwo.cs, made Boss_Animator

The boss battle

In terms of visuals, story and ambience, we wanted Godrick to be the final boss in terms of lore. Showcasing betrayal to the players because Godrick is their guide in the beginning. I made separate game objects for his body, his head and his hands and attached these child game objects onto the Boss Game Object. Then I added SpriteRenderer components to each part so that when my teammate is done with making the assets for the boss, they'll be able to find where to add the sprites in the Unity file. How the boss battle is currently planned as is that he's going to have 3 different phases, Phase 1 is just normal attacks, Phase 2 will have waves of mobs spawning that Godrick summons, and Phase 3 will have holy water rain down on the players which deals damage. All of these functions will stack up on each other and will all work in Phase 3.

I made a start following Brackeys' tutorial on this because it's not something I've learned before. I created folders for it to organise the files, everything boss battle related is under the "Boss Battle" folder. Then I created the states under the boss animator starting from Boss_Intro to Boss Detect which goes to BossPhase1. One part I found super helpful for our other animations was setting up the exit time as 0.9, unchecking fixed duration and making transition offset into 0. This made the attack player animations work more smoothly when the buttons get pressed. I also put placeholder sprites of godrick for his body and his hands to test the scripts out. It didn't work because Godrick isn't set up as the same way in the tutorial. Our Boss isn't going to follow the player through their body, they track the player through their hands.

Boss Battle used to be under my workload but because my teammate wanted to take up more coding, so I offered this to her instead so that I can be available for developing other important functions in the game that I have already made like checkpoints, teleports, adding healing indicators, adding damage indictors and putting together the other levels.

Week 10: Respawn and PlayerInput.cs issues

SUMMARY: Updated Checkpoints.cs, Updated PlayerPos.cs, Updated PlayerInput.cs, updated PlayerHealer.cs, PlayerSupport.cs and PlayerDPS.cs

UPDATED PLAYER POS.CS

So, one error I encountered when I playtested the checkpoints again was that when the player dies and respawns, the player couldn't attack anymore and I knew it's because the controls weren't being reassigned when they respawn. In PlayerPos.cs, under RespawnPosition(), I referenced the Player Input script and tried to reassign the inputs for movement and attacks using playerinput.AssignInputs(playerInput.playerID) and UpdateAttackBindings but this did not work. With help from my lecturer, we found that it was just because it had this.enabled = false; on all three player scripts. All we had to do was delete it. I think I overcomplicated this issue by trying to disable the assigned Inputs through PlayerInput.cs. That section is all noted now.

In PlayerSupport.cs, they didn't have anything under Respawn() so I quickly added in this.gameObject.SetActive(true);, GetComponent<Collider2D>().enabled=true;, currentHealth = maxHealth and healthBar.SetHealth(currentHealth); again, because it got deleted at some point, probably by accident

```
public void RespawnPosition()
{
    if (gameManager != null)
    {
        transform.position = gameManager.lastCheckPointPos;

        //PlayerInput playerInput = GetComponent<PlayerInput>();
        //if (playerInput != null)
        //{
        //    playerInput.AssignInputs(playerInput.playerID);
        //    playerInput.UpdateAttackBindings(); // reassign attacks

        //}
    }
}
```

```
Unity Message | 0 references | sealsaremermaidsindogworld, 7 days ago | 2 authors, 3 changes
private void OnDisable()
{
    if (inputControls != null)
    {
        InputManager.instance.onPlayerJoined -= AssignInputs;
        inputControls.MasterActions.Movement.performed -= MovementPerformed;
        inputControls.MasterActions.Movement.canceled -= MovementCanceled;

        //"unsubscribe" attack events
        inputControls.MasterActions.AttackButton.performed -= ctx => attackButton = true;
        inputControls.MasterActions.AttackButton.canceled -= ctx => attackButton = false;

        inputControls.MasterActions.ChargeAttackButton.performed -= ctx => chargedAttackButton = true;
        inputControls.MasterActions.ChargeAttackButton.canceled -= ctx => chargedAttackButton = false;
    }
    else
    {
        InputManager.instance.onPlayerJoined -= AssignInputs;
    }
```

```
0 references
void Respawn() // nadine added this
{
    this.gameObject.SetActive(true);
    GetComponent<Collider2D>().enabled=true;
    currentHealth = maxHealth;
    healthBar.SetHealth(currentHealth);


}
```

```
// nadine - made changes here
if (playerPosScript != null)
{
    playerPosScript.RespawnPosition();
}

Invoke("Respawn", 1f);
```

```
// Disable the script and collider
GetComponent<Collider2D>().enabled = false;
this.enabled = false;
this.gameObject.SetActive(false);


//checkpoints
```

Week 10: Working on our Pitch Deck

SUMMARY: Presenting our Pitch Deck

PITCH DECK

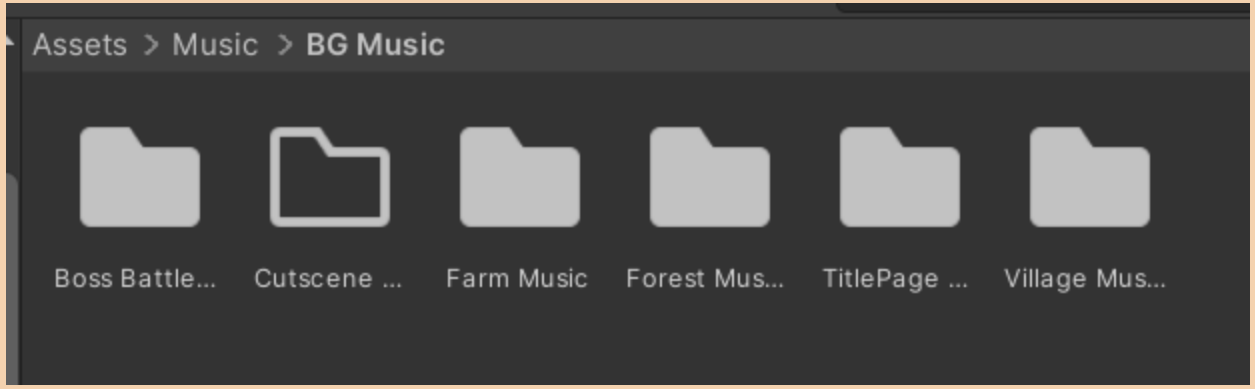
We had to present our Pitch Deck today, to the class. Since we have a big group, I suggested we allocate the slides to each of us. Everyone had 2 slides that they had to talk about in the end. I suggested this so we wouldn't talk above each other by accident when we were pitching our game.

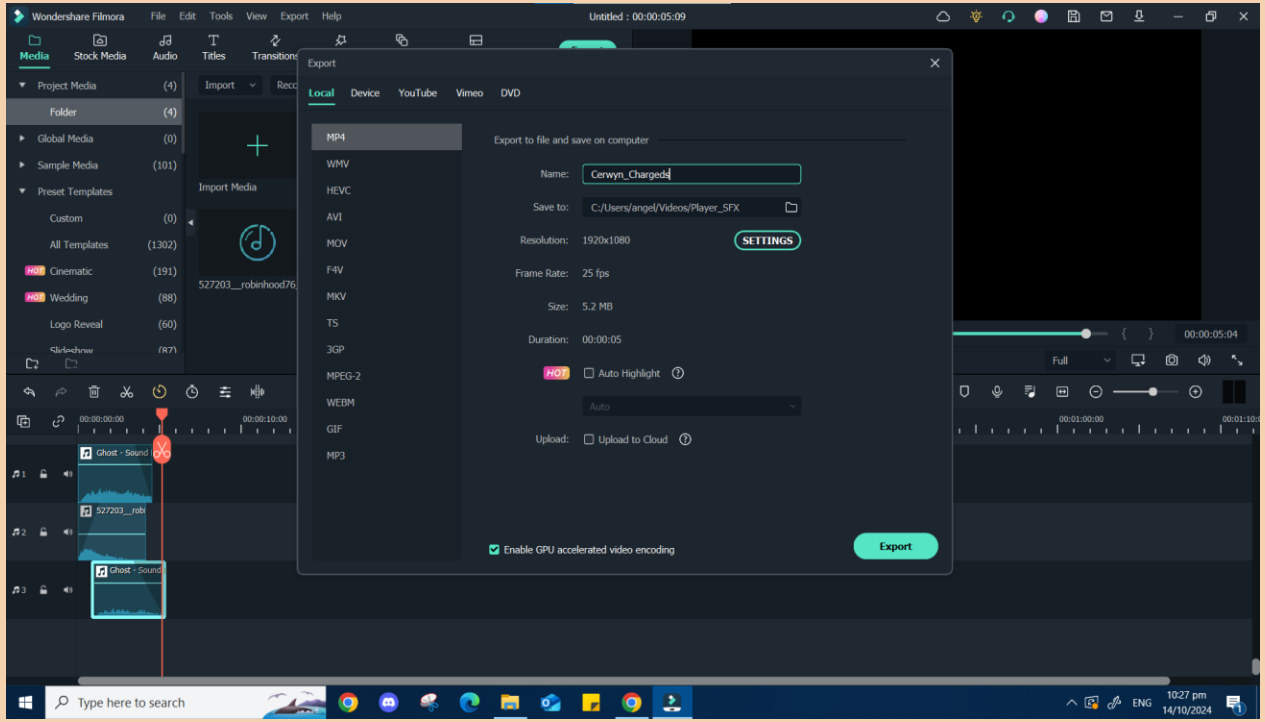
When we were writing our pitch deck, I contributed to a few slides like everyone else. I wrote parts of the Game Mechanics and the Target Audience slides. When we presented, some slides' fonts didn't load and they

Michaela gave us a lot of good feedback about making some slides covering the launching campaign for our game. We had market analysis but no launch plan. So, I took time later on to make a release plan slide covering a launching timeline for our group between October 23rd to Halloween 2025 next year.

Week 11: Adding Music

SUMMARY: Added Cutscene , Forest, TitlePage, Village, Boss Battle, and Farm Music Folders





BG MUSIC

Since submission is coming soon, I decided to import in the music that Brianna and I selected two weeks ago. Here is a direct link to the document: https://docs.google.com/document/d/1z1MzxzpnduPFkQ5pBuTuhx4wFvPCp3_rAalTgn4y-ik/edit?usp=sharing

After discussing what type of ambience, we wanted our game to project through the music, we finalised that music outside the game, the title page and significant cutscenes, would have eerie horror music box tunes playing and that in-game we would play 8-bit spooky Halloween themed tunes to match the 2D art style. We hope that it'll help the players feel nostalgia and cozy spooky vibes when playing our game.

Our other options were to play normal spooky music on the title page, but this never reached the final because the in-game music is already so upbeat, I wanted the title screen music to keep the players curious on what the game is going to be about by creating suspense through the eerie music box melody.

SFX

After adding in all the background music, I edited the SFX that I chose for all the players and imported those in as well when they were a good length. Each player has a different combat SFX for their normal attack and charged attack. After editing the music, I named the files with their names so that my teammates won't get confused when they need to find the SFX. Harry and Cerwyn's weapons both have a swing/slash SFX and Wylla's is potion bottles breaking. I used Wondershare Filmora to edit the SFX, either to shorten the SFX files or to combine SFX that I wanted like for example to create a wave sound effect with haunting noises to showcase Cerwyn's charged attack ability, I got a ghost sound effect and a wave crashing sound effect and combined the two together to create the charged attack sound for Cerwyn. I also layered the ghost sound effect to make it sound like there's multiple ghosts because that's how it was animated by my teammate.

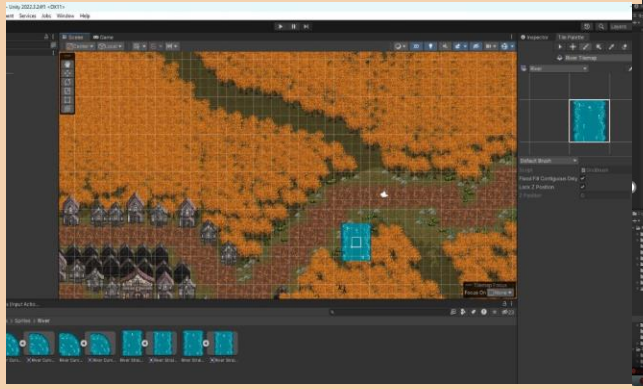
Week 11: Setting Level 2's River

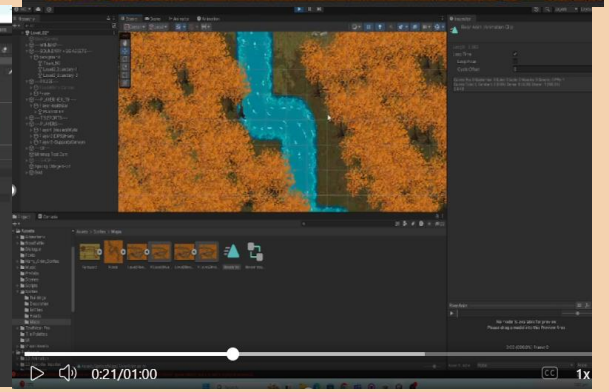
SUMMARY: Updated Scene Level_02, Added River Folder, Added River Curved 1 and 2, Added River Straight 1 and 2. Added River Animator amd River_Anim

FIXING LEVEL_02 BOUNDARY COLLIDERS AND TRIED TO ADD RIVER INTO UNITY

I started fixing the minor issues in Level_02 before adding the prefabs in. I realised the river hadn't been imported in yet, so I tried to add it in using the tile map method that got mentioned in a 2D Dungeon Crawler Tutorial I watched in Week 1 when I was coding player movement. I know where I went wrong here, I didn't make a tile set of my river sprite instead I just imported the sprites into the tile map as well as the dimensions were off too, the dimensions for the river sprites were 32x32. The sprites ended up being bigger than the path of the river that was mapped out by my teammate.

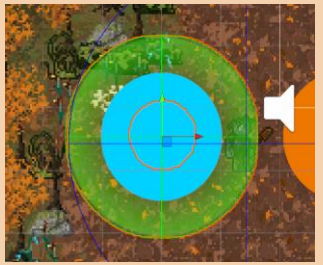
So, to save time, I gave my sprites to Theo because he has the photoshop files of the various maps within the game and told him that it's best to add the river onto the map through photoshop, both variations and then give it back to me so that I can attach an animator to the background's game object to toggle the two variations of the river on the map so my animation of the river can be seen.





Week 11: Healing Indicators

SUMMARY: Updated PlayerHearler.cs, Added Heal Game object under Player1(Healer)/Wylla game object.



```
public GameObject healImage;
```

```
void Attack()
{
    // Detect enemies in range of attack
    Collider2D[] hitEnemies = Physics2D.OverlapCircleAll(attackPoint.position, attackRange, enemy

    // Damage them
    foreach (Collider2D enemy in hitEnemies)
    {
        //Debug.Log("Healer hit " + enemy.name);
        enemy.GetComponent<AIChase>().TakeDamage(attackDamage);

        totalDamageDealt += attackDamage;

        if (totalDamageDealt >= damageThreshold)
        {
            canUseChargedAbility = true;
            skillImage.SetActive(true);
            // Debug.Log("Healer Charged ability is now available!");
        }
    }
}

healImage.SetActive(false);
```

```
1 reference | sealsaremermaidsindogworld, 34 days ago | 2 authors, 4 changes
private void HealAlliesInRange()
{
    // finds all GameObjects within the healing radius
    Collider2D[] allies = Physics2D.OverlapCircleAll(transform.position, healingRadius);

    foreach (Collider2D ally in allies)
    {
        if (ally.CompareTag("Ally")) // checks for tag
        {
            HealthBar allyHealthBar = ally.GetComponent<HealthBar>();
            if (allyHealthBar != null)
            {
                // increases ally's health but won't exceed the max
                int newHealth = Mathf.Min((int)allyHealthBar.slider.maxValue, (int)allyHealthBar.slider.value + healingAmount);
                allyHealthBar.SetHealth(newHealth);
                //Debug.Log("Healed " + ally.name);
                healImage.SetActive(true);

            }
        }
    }
}
```

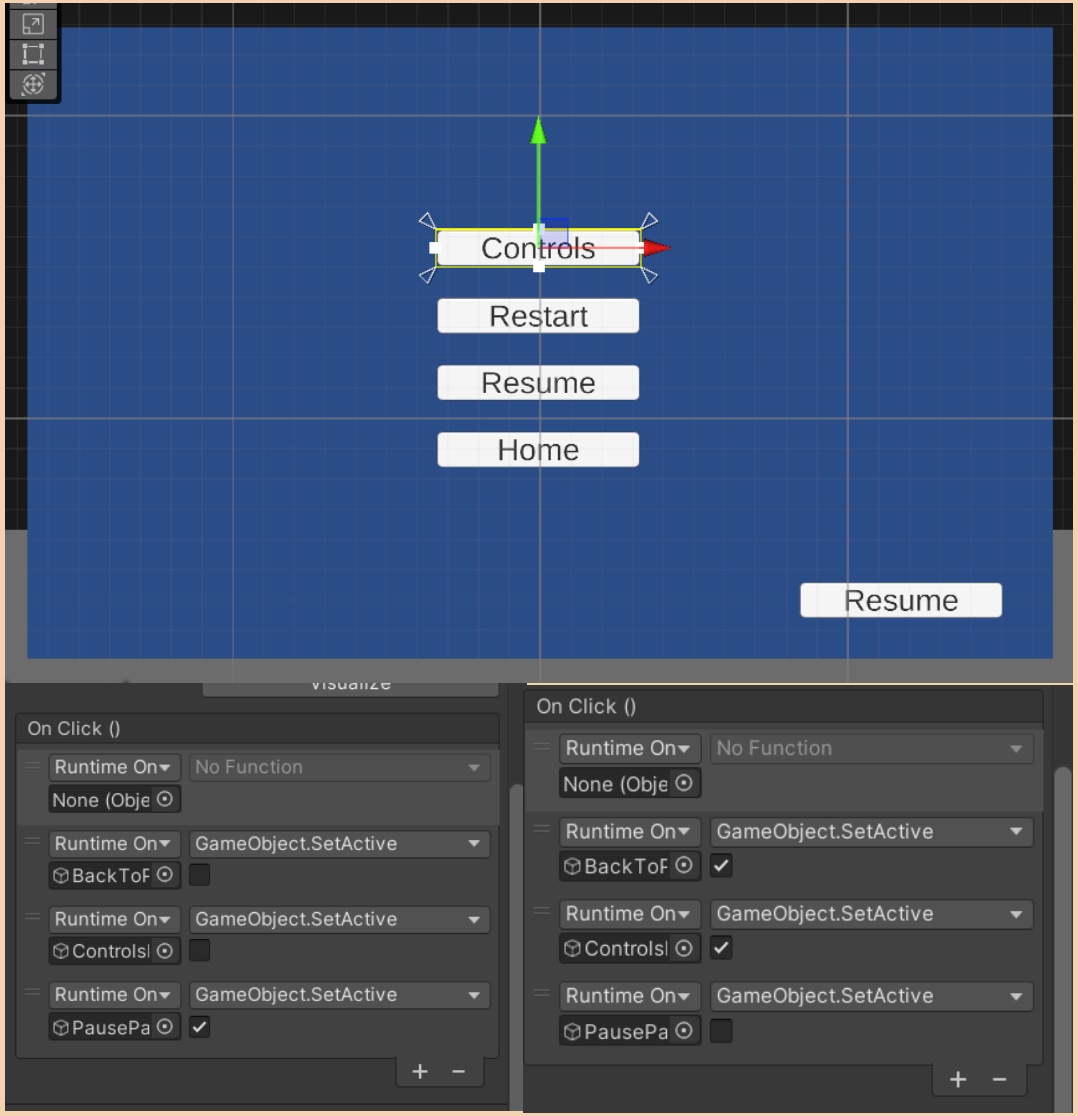
```
{
    isHealing = true;
    anim.SetBool("IsHealing", true);
    // finds all GameObjects within the healing radius
    Collider2D[] allies = Physics2D.OverlapCircleAll(transform.position, healingRadius);

    foreach (Collider2D ally in allies)
    {
        if (ally.CompareTag("Ally")) // checks for tag
        {
            HealthBar allyHealthBar = ally.GetComponent<HealthBar>();
            if (allyHealthBar != null)
            {
                // increases ally's health but won't exceed the max
                int newHealth = Mathf.Min((int)allyHealthBar.slider.maxValue, (int)allyHealthBar.slider.value + healingAmount);
                allyHealthBar.SetHealth(newHealth);
                //Debug.Log("Healed " + ally.name);



            }
        }
    }

    anim.SetBool("IsHealing", false);
    isHealing = false;
}
```

HEALING INDICATORS

Firstly, I made a reference to the health indicator image that my classmate made into PlayerHealer.cs then I had to make a gameobject for the heal image, so I could drag and drop it into the inspector of Player1 (Healer)/Wylla. Then back on PlayerHealer.cs, under HealAlliesInRange(), I tried to use set active lines of code to activate the heal image and deactivate the image when there's a player tagged ally that enters her healing radius. I tried putting healImage.SetActive(true); under if(allyHealthBar != null) but the issue here was that it just kept staying active which was not what we want. I also tried to put else { healImage.SetActive(false); } outside of the foreach(collider2D ally in allies) to try to turn it off but the issue here was that it flickers and doesn't turn on when we want it to. Then I tried to do the same thing this time under the Attack(). After that, I tried to make a boolean to turn the heal image on and off instead because my classmate also made an idle animation with the healing radius attached to Wylla. The reference was called anim.SetBool("IsHealing",); which I set as (,false); outside of the HealAlliesInRange() function and was set as true inside the function: anim.SetBool("IsHealing", true); I also had to add an isHealing = true next to the setbool line as well as one saying it's false next to the set bool line outside.

Apparently, the only changes I had to make was add playerLayers at the end of Collider2D[] allies = Physics2D.OverlapCircleAll(transform.position, healingRadius); without playerLayers, the healImage would activate in the wrong time because it was detecting the players but also other gameobjects that we didn't want to be detected because they weren't tagged as allies. Adding playerLayers completed the line of code so that it would be able to detect just the players when they enter the healingRadius. Then instead of healImage.SetActive(false); being outside, I put it under else{}. This ensures that the sprite will deactivate if there are no allies being detected in the healing radius.

Week 12: Updating PauseMenu



SUMMARY: Added Controls Button, Added BackToPause Button, Updated ---PAUSE--- gameobject, Added ControlPanel gameobject

ADDING IN-GAME PAUSE MENU

Following the playtest feedback we got, someone mentioned that it'd be good to have the controls accessible in game as well, not just through the title page. This makes sense because when you return to the title page your progress gets reset in game. So, for better gameplay, I implemented a controls panel in the pause menu, that our players can access so they don't have to go back to the title page to check.

The way I set it up was that I duplicated one of the buttons already in the pause menu and renamed it as Controls and another one called BackToPause. Then for BackToPause button's OnClick() I added the ControlsPanel game object, the PausePanel game object and the BackToPause game object. I selected GameObject.SetActive functions to all three and set Pause Panel as true and the other two as false. This ensures that The Pause Panel game object will be activated again when the BackToPause button is selected.

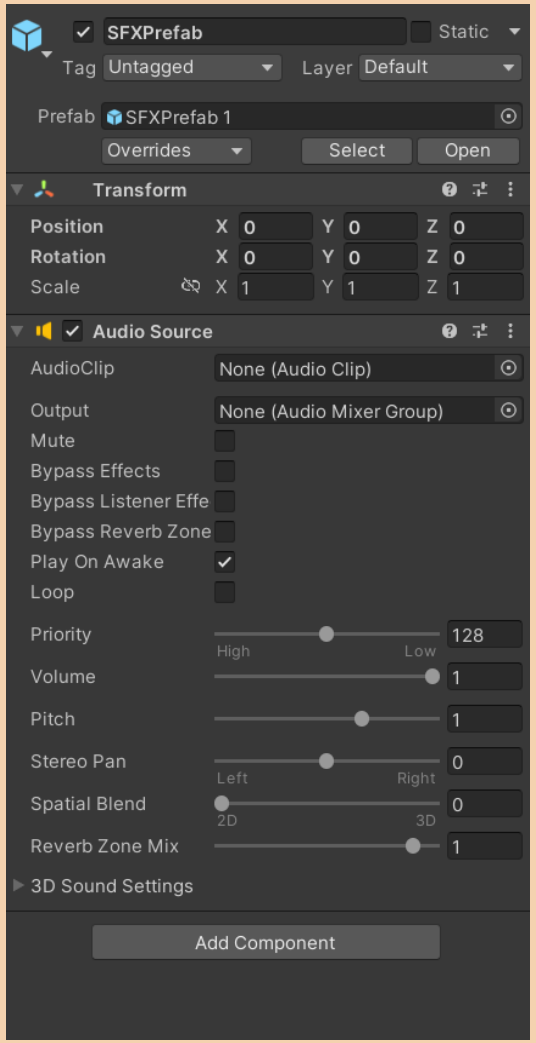
For the Controls button OnClick(), I made ControlsPanel and BackToPause button activated. Pause Panel game object is turned off. This makes the player see the controls panel and be able to return to the pause menu because of the BackToPause button.

Week 12: Adding SFX to Players 1

SUMMARY: Created AudioManager.cs, added SFXPrefab, updated PlayerDPS.cs, PlayerHealer.cs and PlayerSupport.cs

```
// for audio
private AudioClip Cerwyn_BaseATK_SFX;
private AudioClip Cerwyn_ChargedATK_SFX;
```

```
private void Update()
{
    // Check if the Fire1 button is pr
    if (playerInput.attackButton)
    {
        Attack();
        CerwynBaseSFX();
        playerInput.attackButton = fal
        anim.SetBool("IsAttacking", tr
    }
    else
    {
        anim.SetBool("IsAttacking", fa
    }

    // Check if the Fire3 button is pr
    if (canUseChargedAbility && player
    {
        PerformAbility();
        CerwynChargedSFX();
        playerInput.chargedAttackButto
        anim.SetBool("IsChargedAttacki
    }
```



ADDING SFX TO PLAYERS AS DAMAGE INDICATORS 1

Taking our feedback from playtesting into consideration, I wanted to make damage indicators for our game. Since out gameplay features a straightforward hack and slash and there are no combos in the attack system made for the players, I decided to use audio for the damage indicators instead of text. This will help add to the sound design of the gameplay and make the combat system be less mundane. I was also encouraged to make our game better as much as I could during polishing because of what I've learned when I attended NZGDC24. We don't have time to add new mechanics, so I'm spending time on other aspects, like audio, to make our game a bit more interesting for submission.

Firstly, I tried to make the audio be an animation event, my classmate recommended an audio tutorial for me to reference from by _____. This video taught me how to add audio to my animation using "PlayClipAtPoint", below is an example of how I set it up in Wylla's script (PlayerHealer.cs):

```
public void WyllaAttackSFX()
    {
        //plays audio
        AudioSource.PlayClipAtPoint(Wylla_BaseATK_SFX, transform.position, volume * 10f);
    }


    public void WyllaChargedAttackSFX()
    {
        //plays audio
        AudioSource.PlayClipAtPoint(Wylla_ChargedATK_SFX, transform.position);
    }
```

It worked but it came with issues like how the audio was too quiet when the attack button was pressed or how it was looking for a receiver even if I had referenced the correct audio source. I also tried to increase the volume by making a reference at the top of the script and tried to multiply the audio initially by 5 and then eventually by 10f.

Week 12: Adding SFX to Players 2

SUMMARY: Created AudioManager.cs, added SFXPrefab, updated PlayerDPS.cs, PlayerHealer.cs and PlayerSupport.cs

ADDING SFX TO PLAYERS AS DAMAGE INDICATORS 2

I asked my lecturer for some suggestions on what I could do, and we discussed that using an Audio Manager would be useful in my case. This is what I did, I created an SFXprefab (screenshot in the previous slide) and attached an audio source component to it, leaving the Audio Clip as none.

Then I created a new Audio Manager script, adding public static AudioManager instance; I learned that doing this will make it easier to manage all my player's audios in one place. Instance = this; lets Unity know that this is the AudioManager to use.

Public void PlaySFX() is a function that covers making a new SFXprefab (Instantiate(SFXPrefab) to play audio based off its name (audioSource.clip = clip; and audioSource.Play();). Letting you adjust the volume through audioSource.volume = volume; and destroy the audio when the audio clip that was assigned is done playing. This method fixed all the errors that I ran into and made managing my audio a lot better.

Then all I had to do was add new functions in the player scripts and call the correct audioclips through there when the attack button and charged attack buttons are pressed. Here's an example in Wylla's script:

```
public void WyllaBaseSFX()
{
   AudioManager.instance.PlaySFX(Wylla_BaseATK_SFX, 1.0f);
}


public void WyllaChargedSFX()
{
   AudioManager.instance.PlaySFX(Wylla_ChargedATK_SFX, 1.0f);
}
```

```
using UnityEngine.Audio;
using UnityEngine;
using System.ComponentModel;

Unity Script (1 asset reference) | 7 references
public class AudioManager : MonoBehaviour
{
    public static AudioManager instance;
    public GameObject SFXPrefab;
    Unity Message | 0 references
    void Start()
    {
        instance = this;
    }

    6 references
    public void PlaySFX(AudioClip clip, float volume)
    {
        GameObject obj = Instantiate(SFXPrefab);
        AudioSource audioSource = obj.GetComponent<AudioSource>();
        audioSource.clip = clip; // assigns clip to audio source ga
        audioSource.Play();
        audioSource.volume = volume;
        Destroy(obj, audioSource.clip.length); // destroys clip whe
    }
}
```

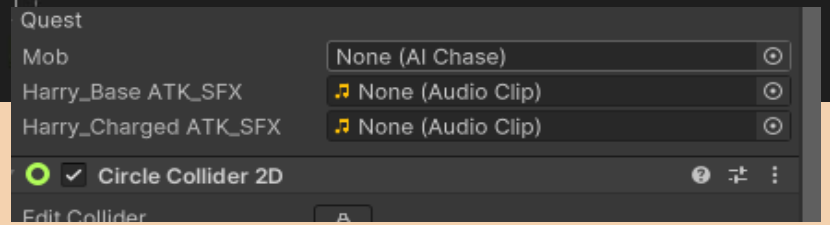
I found out that I couldn't add this line of code, AudioManager.instance.PlaySFX(_____, 1.0f);, to the existing Attack() and Performability() functions. To save time, I just created these functions to run the audio code and called them in the update function. WyllaBaseSFX() is under if(playerinput.attackbutton) and WyllaChargedSFX() is under if(canUseChargedAbility && playerinput.chargedattackbutton). This ensures that when the attack button is pressed, the base attack audio clip will play and the same goes for the charged attack audio clip.

Week 13: Fixing bugs

SUMMARY: Updated MainMenu.cs, Updated PlayerSupport.cs, Updated PlayerDPS.cs, Updated PlayerInput.cs. Updated PauseMenu.cs

```
    public PlayerInput[] players;

0 references | sealsaremermaidsindogworld, 72 days ago | 1 author, 1 change
public void PlayCutscene()
{
    // loads scenes in order within the build setting
    SceneManager.LoadScene(SceneManager.GetActiveScen
}

0 references | sealsaremermaidsindogworld, 72 days ago | 1 author, 1 change
public void QuitGame()
{
    Application.Quit();
}

0 references | notaurelle, 55 days ago | 2 authors, 2 changes
public void StartGame()
{
    SceneManager.LoadScene("Level_01");

    foreach(PlayerInput p in players)
    {
        p.canMove = true;
    }
```

```
52      -   private AudioClip Cerywn_ChargedATK_SFX;
    50  +   private AudioClip Cerwyn_ChargedATK_SFX;
53  51
```



FIXING BUGS

The first bug I ran into, was that I was getting a null reference error when I was playtesting Harry and Cerwyn's SFX in Level_01. I couldn't see the null reference anywhere. Turns out, the reason why is because I made the Audio Clip references at the top private, so by doing this I couldn't see that in the PlayerSupport.cs and the PlayerDPS.cs component had null references. All I had to do was change them from private to public. After that, I was able to assign the audio clips successfully.

For the second bug, I needed help from my lecturer because the error mostly lay in the PlayerInput.cs and one line of code in the PauseMenu.cs script. The errors was that when players clicked the home button and returned to the title page then tried to start the game again; players wouldn't be able to rejoin. I tried to figure it out by myself first, I added foreach(PlayerInput p in players) { p.canMove = true; in MainMenu.cs under StartGame() which was a function attached to the Start button in the first cutscene when the players first play the game. It worked momentarily but as I kept replaying the game loop (going in game, to going back to cutscene, pausemenu and back) I ran into more errors. Below is what my lecturer did:

Assets\Scripts\InputManager\PlayerInput.cs

```
@@ -28,7 +28,14 @@ public class PlayerInput : MonoBehaviour
28  28      // Start is called before the first frame update
29  29      void Start()
30  30      {
31      -       InputManager.instance.onPlayerJoined += AssignInputs;
    31  +       if (InputManager.instance.players.Count >= playerID + 1)
    32  +       {
    33  +           AssignInputs(playerID);
    34  +       }
    35  +       else
    36  +       {
    37  +           InputManager.instance.onPlayerJoined += AssignInputs;
    38  +       }
32  39          anim = GetComponent<Animator>();
33  40          spriteRenderer = GetComponent<SpriteRenderer>();
```

Assets\Scripts\UI\Title Page\PauseMenu.cs

```
@@ -21,7 +21,7 @@ public class PauseMenu : MonoBehaviour
21  21
22  22      public void Restart()
23  23      {
24      -       InputManager.instance.players.Clear();
    24  +       //InputManager.instance.players.Clear();
25  25          SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
26  26          Time.timeScale = 1;
27  27      }
```

Basically, in the playerinput.cs he made an if statement, under Start(), that covers detecting when a new player joins, checking if the players we want are already there. Then it waits to make sure player joining and assigning inputs only happens once. So, we don't have to do it again. Then I just had to remove InputManager.instance.players.clear(); in PauseMenu under Restart() because that was what was unassigning the players to the controls.

Week 13: Updating Cerwyn's charged Attack

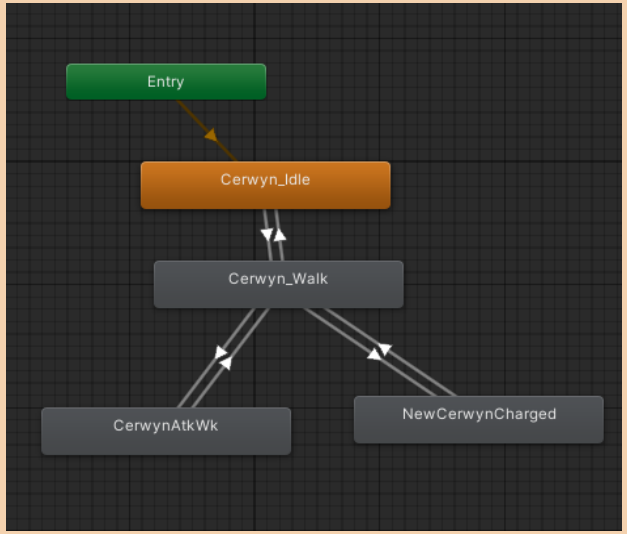
SUMMARY: Updated MainMenu.cs, Updated PlayerSupport.cs, Updated PlayerDPS.cs

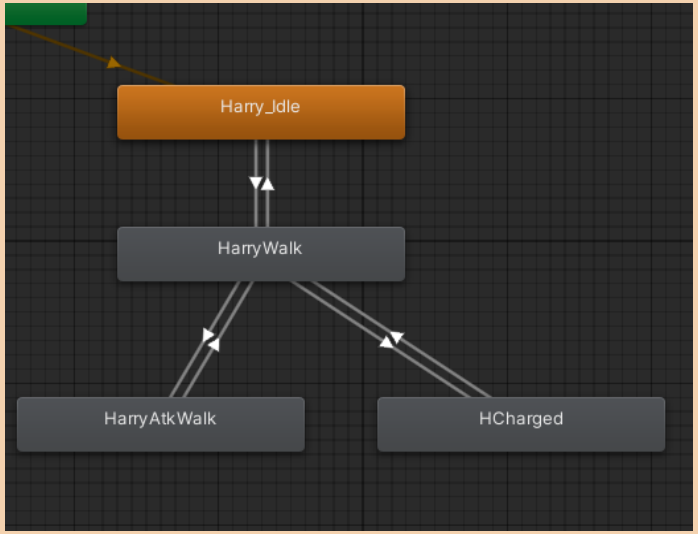
UPDATING CERWYN/HARRY CHARGED ATTACK

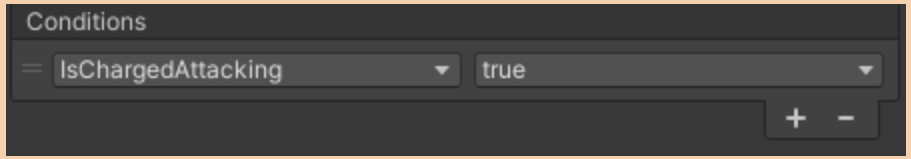
When Theo was finally done updating Harry and Cerwyn's charged attacks. I had to redo them in their individual animators. Thankfully, this time, Theo imported the animated frames properly (16x16) so doing this didn't take a lot of work. Since he had already made the animation files for them both in Unity. The only thing he forgot to do was check "Loop" for both animations. It was easy to find the animations because he renamed Cerwyn's "CerwynNewCharged" and Harry's was just in the Charged folder under Harry_Anims, so it wasn't too bad.

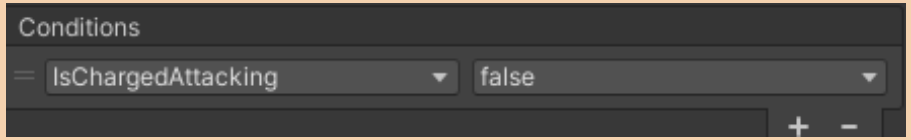
In the animator, all I had to do was drag and drop the new charged animations for Harry and Cerwyn. Then did the same process I did before, made a transition going from their walk animation to the charged animation with an "IsChargedAttacking" bool set as true. Then added another transition going from the charged animation to the walk animation with the bool being set to false. Has Exit time is unchecked going to the charged anim and it is checked when charged goes back to the walk anim.

There's still some weird timings that occur when the charged attack button is pressed for Harry. It works, but the animation is delayed sometimes. Since it's so close to the due date, I've decided to just leave it for now because it works anyway.



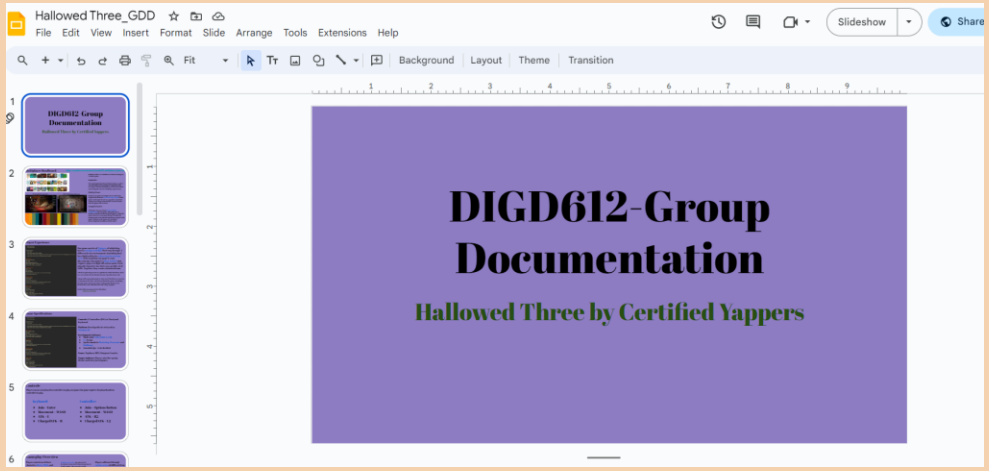






Week 13: Updating GDD

SUMMARY: New Format for GDD



UPDATING GDD

Since it's polishing season, I decided to spice up our GDD and just make it neater overall. I did this by using the template we had used for Pitch Deck. It was a free-to-use template that my teammates and I chose together. This didn't take too much time, but I did find that there were still some extra slides that were missing.

All I did was duplicate the pitch deck, shared it with my teammates. Then began replacing the pitch deck slides with the GDD content from the old purple slides. I wanted to do this for our group so that our group aesthetic was consistent.

Using the pitch deck template made the GDD less messy and a look a lot more polished and professional for our final submission.

Week 13: Null References and reassigning them all in Level_02 and Level_03

SUMMARY: Updated Scene Level_02, Updated DialogueManager.cs, Updated Level_03

NULL REFERENCES AND MISSING REFERENCES

Naturally, we ran into a lot of errors when importing prefabs from Level_01 to Level_02 and Level_03. I went ahead and reassigned any null references in them. In Level_01, there was a missing reference when the dialogue triggers in the map were entered in. I just had to note out startButton.SetActive(true); and it made the game flow better.

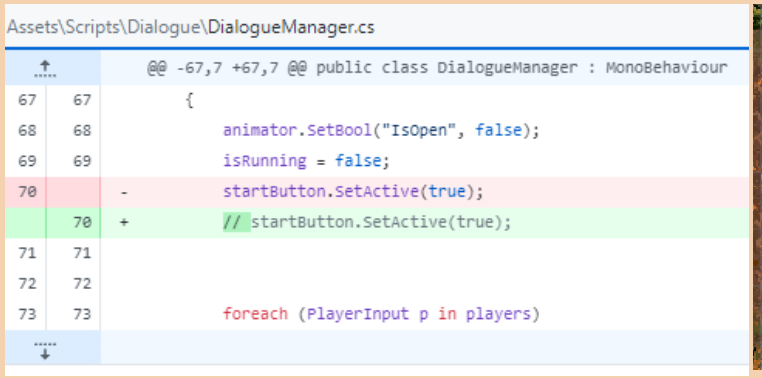
After that, I found null references to Wylla's Health Bar for some reason, so it wouldn't show her health indicators. I just had to copy and paste the component from Level_01 to Level_02. It's weird how it didn't just do this automatically because the other components were not null. I had to reassign the same for Harry and Cerwyn's Health Bars. The same thing happened to Level_03.

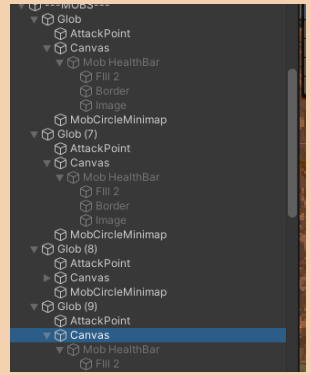
I also noticed my team mates still hadn't added in the Game Manager from Level_01 into Level_02 so, I had to quickly make it a prefab and import it into Level_02 and Level_03.

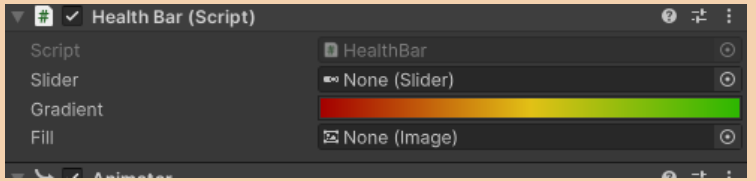
My point system wasn't showing up and I finally found the issue. Score Text wasn't assigned under the GameManager Component. For some reason it got unassigned a few weeks ago. I proceeded to add the Game Manager to Level_03 as well.

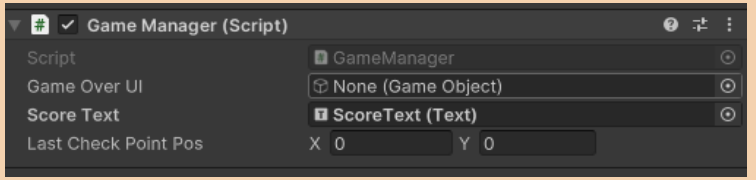
I also had to add in an AudioManager to Level_02 and Level_03 to stop the Null Reference errors I was getting because it was looking for the SFX prefab.

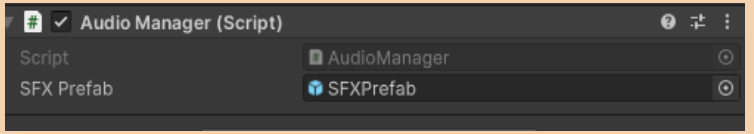
I didn't have to but I imported my teammate's barn into level_03.

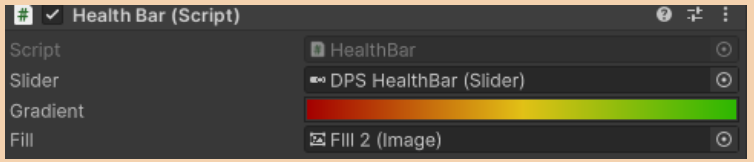






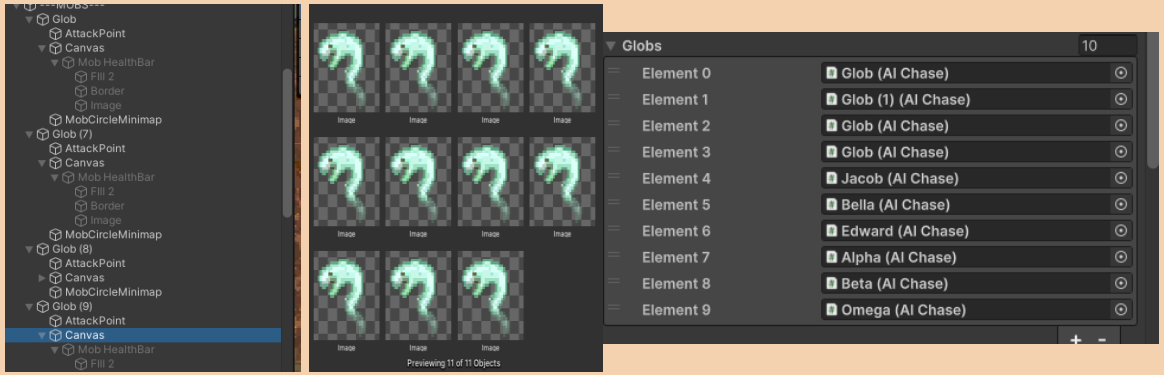






Week 13: Updating Mob sprites in Level_02 and more bugs

SUMMARY: Updated Scene Level_02, Updated DialogueManager.cs, Updated Level_03, Updated PlayerPos.cs, Updated PlayerSupport.cs, Updated PlayerDPS.cs




```
190     void Die()
191     {
192         Debug.Log("Player DPS died!");
193
194         //Disable enemy script as they have 'died'
195         GetComponent<Collider2D>().enabled = false;
196         this.enabled = false;
197         this.gameObject.SetActive(false);
198
199         // nadine - made changes here
200         if (playerPosScript != null)
201         {
202             playerPosScript.RespawnPosition();
203         }
204
205         Invoke("Respawn", 1f);
```

UPDATING MOB SPRITES IN LEVEL_02

Cathy wanted the globs in Level_02 to be Ghosties so I went ahead and helped Aurelia change the sprites of her mobs and assigned the health bars to all of them. We ran into a few minor issues like the pixels per unit of the Ghostie being 100 pixels per unit when they were supposed to be 16. Thankfully, Aurelia already put the mobs where she wanted them in Level_02 so I didn't have to do any other extra work. I decided to help Aurelia because adding mobs into the scenes take a while and we all wanted the game to be done as soon as possible.

I passed the rest to Cathy and got to work on another bug that appeared. When you respawn at the checkpoints, you can't attack again, and this is because the PlayerPos.cs and Checkpoint.cs scripts haven't been updated to cater to the changes that were made in the PlayerInput.cs script. Then I had to assign the mobs into Cerwyn's stun list.

MORE BUGS WITH PLAYERINPUT.CS AND CHECKPOINTS

So, the same issue with the home button happened with the checkpoints, you can't attack after you die and respawn at the checkpoint. I tried to fix it by making an if statement to call the AssignInputs(), trying to reassign it. Unfortunately, some weird bugs happened when me and my classmates were working on the game together. Our changes that we pushed and pulled almost broke the game 4 times this week and the code got lost. I lost my changes.

It wasn't too bad on my end because I was just experimenting. It was bad for Cathy because she had important stuff. We concluded that if we couldn't fix this checkpoint bug, in time for submission, we'd keep it in the game anyway because the players are still able to respawn and move around, just not attack and the game resets the inputs anyway when you teleport into the next level.

Update: My lecturer, helped me find the issue. It was the same that happened before when the Checkpoints() first got implemented in. It's because there's a line of code in the Die() in PlayerSupport.cs and PlayerDPS.cs that says this.enabled = false;. Which is disabling the player script when they die. So, all I had to do was remove these and re-enable them in Respawn(). It was a weird bug because it was working before even if I didn't make these changes. I also had the option to write down this.enabled = true; into Respawn() and it would have fixed the issue the same way.

Week 13: What didn't make it to the final build

POINTS AND SHOP

The point system works but I noticed that points get reset when players join a new round. I decided to leave this be for the final build. This is because, there isn't time to figure out why it's not just adding up the points in the next levels. We set up our different scenes in late stages. This was due to a team mate not completing one of the core mechanics that they wanted to incorporate into the game, they asked me to wait for them, which put the team behind in combining everything together. But unfortunately, these were just ones that we didn't have time to polish in the end. The main point though, is that the game is still playable, and the points don't really do anything at all. They just work but reset when entering a new level. The shop wasn't my workload but the points system I made was supposed to be the currency you can use to upgrade skills for the players.

SFX IN THE PAUSE MENU

If I had a bit more time to clear all the major bugs that we came across, I would have worked on adding a music panel to the pause menu. This function would just have a master volume slider for every audio, a BG music slider for the music, and an SFX slider to adjust the volume of the sound effects that you can hear in game. I didn't have time to do this because I was helping my teammates set up the levels because submission is coming up.

HEALING IINDICATOR FOR WYLLA'S HEALING ZONE

I wanted to add the healing sprite that my teammate made for Wylla's healing radius to Wylla's healing zone. But this implementation wasn't a priority because it already has text saying "Heal Wylla" above every gravestone and dialogue is triggered before enterin Wylla's healing zone that lets the player know that Wylla can be healed near the gravestone.

CERWYN'S STUN AND PASSIVE ABILITY

It's hard to say if Cerwyn's stunning ability did make it to the final build because I hardly ever get to test him because I don't own a controller at home that I can use to test him out. But I believe that he only stuns the Gelatinous Globs, but not the other mobs. It's not too obvious, so I decided to leave it. I also didn't have time to implement an indicator that the mobs do 4DMG less than their overall attack damage. I think adding a visual indicator that the debuff was happening would have been nice to let the players know that Cerwyn has a passive ability. But the code works. Update: Cerwyn does stun but it's not obvious because the mobs slide anyway.

BOSS BATTLE

Teammate did not complete it in time. We have a cutscene implemented instead to finish off the game and return back into Home screen.

References

https://youtu.be/-7I0slJyi8g?si=NWjwryi90xV653qu
https://discussions.unity.com/t/objects-become-invisible-depending-on-camera-rotation/930256
https://discussions.unity.com/t/objects-become-invisible-depending-on-camera-rotation/930256
https://discussions.unity.com/t/should-we-be-using-coroutines-to-manage-things-like-cooldowns-or-buff-durations/900742/2
https://discussions.unity.com/t/running-animation-with-keyboard-and-or-buttons/697870
https://stackoverflow.com/questions/42737573/change-flip-player-sprite-direction
https://www.youtube.com/watch?v=N73EWquTGSY
https://www.youtube.com/watch?v=qGdf1T4Ew8I
https://www.youtube.com/watch?v=ofCLJsSUom0
https://www.youtube.com/watch?si=NWjwryi90xV653qu&v=-7I0slJyi8g&feature=youtu.be