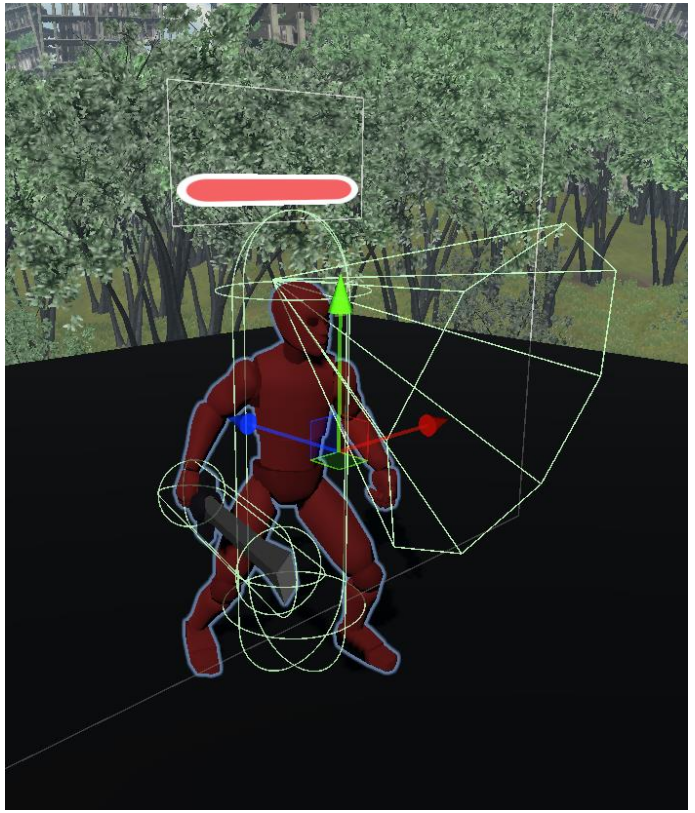
SURVIVE THE GAME

By, Jeremy Labelle

The idea

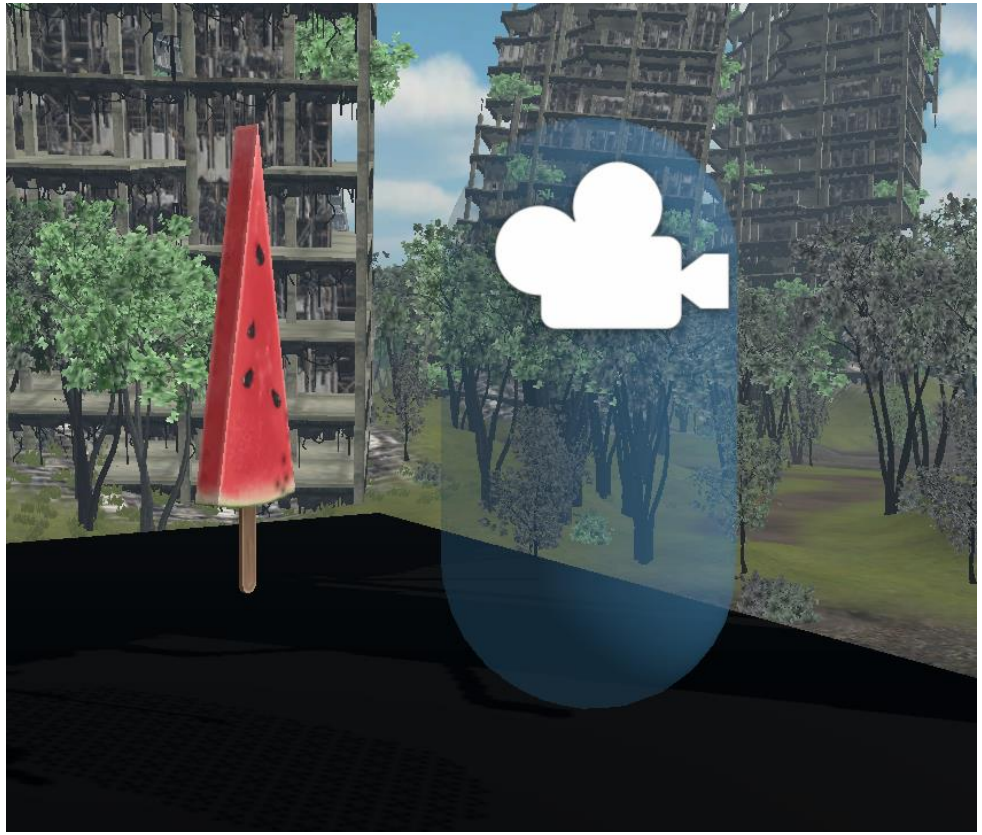
- Make a sword fighting Ai that Attacks, Parries, Blocks, and Dodges
- Make a first person controller for sword combat
- Make the enemy intelligently respond to your combat moves

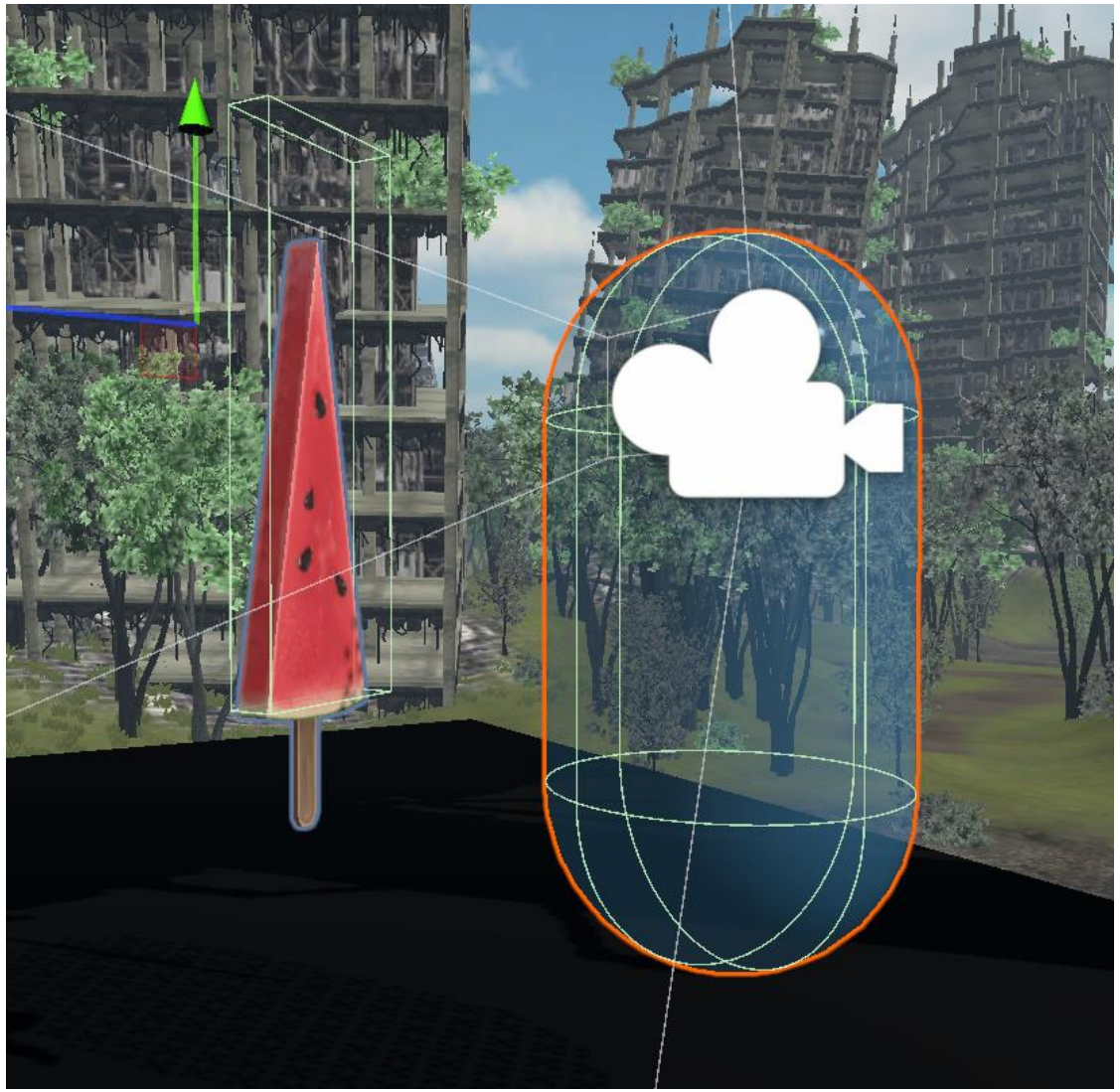
Creating the Fight area and Enemy



*The sword has a rigidbody with isKimematic checked

The Player





Player Scripts

```
Unity Message | 0 references
private void Start()
{
    Cursor.lockState = CursorLockMode.Locked;
    rb = gameObject.GetComponent<Rigidbody>();
}

Unity Message | 0 references
private void Update()
{
    if (Input.GetKey(KeyCode.Space) && Physics.Raycast(rb.transform.position, Vector3.down, 1 + 0.001f))
        rb.velocity = new Vector3(rb.velocity.x, 5.0f, rb.velocity.z);

    if (Input.GetKey(KeyCode.LeftShift)) walkSpeed = 10f;
    if (Input.GetKeyUp(KeyCode.LeftShift)) walkSpeed = 5f;
    look();
}

Unity Message | 0 references
private void FixedUpdate()
{
    movement();
}

1 reference
void look()
{
    pitch -= Input.GetAxisRaw("Mouse Y") * sensitivity;
    pitch = Mathf.Clamp(pitch, -90.0f, 90.0f);
    yaw += Input.GetAxisRaw("Mouse X") * sensitivity;
    Camera.main.transform.localRotation = Quaternion.Euler(pitch, yaw, 0);

}
1 reference
void movement()
{
    Vector2 axis = new Vector2(Input.GetAxis("Vertical"), Input.GetAxis("Horizontal")) * walkSpeed;
    Vector3 forward = new Vector3(-Camera.main.transform.right.z, 0.0f, Camera.main.transform.right.x);    //expensive operation so call in fixedUpdate as opposed to u
    Vector3 wishDirection = (forward * axis.x + Camera.main.transform.right * axis.y + Vector3.up * rb.velocity.y);
    rb.velocity = wishDirection;
}
```

```
0 references
public void Clash()
{
        CanAttack = false;
        Animator anim = sword.GetComponent<Animator>();
        anim.SetTrigger("Clash");
        StartCoroutine(ResetAttackCooldown());
}

1 reference
public void SwordAttack()
{
    CanAttack = false;
   // if () { Clash(); }
        Animator anim = sword.GetComponent<Animator>();
        anim.SetTrigger("Attack");
        StartCoroutine(ResetAttackCooldown());


}

1 reference
public void Block()
{
    CanAttack = false;
    Animator anim = sword.GetComponent<Animator>();
    anim.SetTrigger("Block");
    StartCoroutine(ResetBlockCooldown());
}

2 references
IEnumerator ResetAttackCooldown()
{
    yield return new WaitForSeconds(AttackCooldown);
    CanAttack = true;
}

1 reference
IEnumerator ResetBlockCooldown()
{
    yield return new WaitForSeconds(BlockCooldown);
    CanBlock = true;
}
```

Enemy Scripts

```
[SerializeField]
GameObject Enemy;
public bool CanAttack = true;
public bool CanBlock = true;
public float BlockCooldown = 2.0f;
public float AttackCooldown = 1.0f;
Unity Message | 0 references
private void OnTriggerEnter(Collider other)
{
        if (CanAttack)
        {
            Swing();
        }

        if (!CanAttack && CanBlock)
    {
        Block();
    }

        else if (CanBlock)
    {
        Block();
    }
}

1 reference
public void Swing()
{
    Animator anim = Enemy.GetComponent<Animator>();
    anim.SetTrigger("SwingForPlayer");
    StartCoroutine(ResetAttackCooldown());
}

2 references
public void Block()
{
    Animator anim = Enemy.GetComponent<Animator>();
    anim.SetTrigger("Evade");
    StartCoroutine(ResetBlockCooldown());
}
```

Enemy Fight Pattern

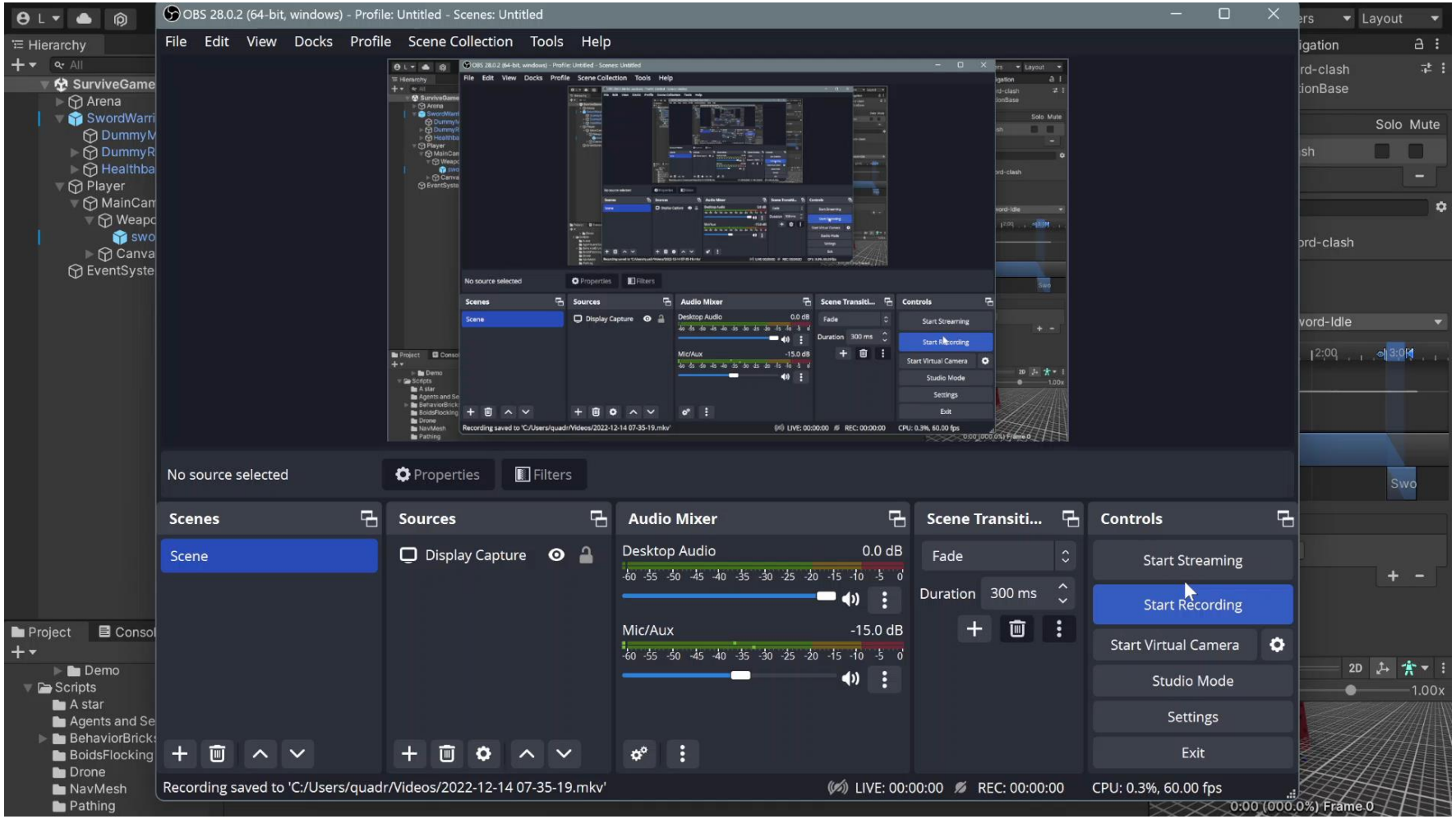
←

Enemy

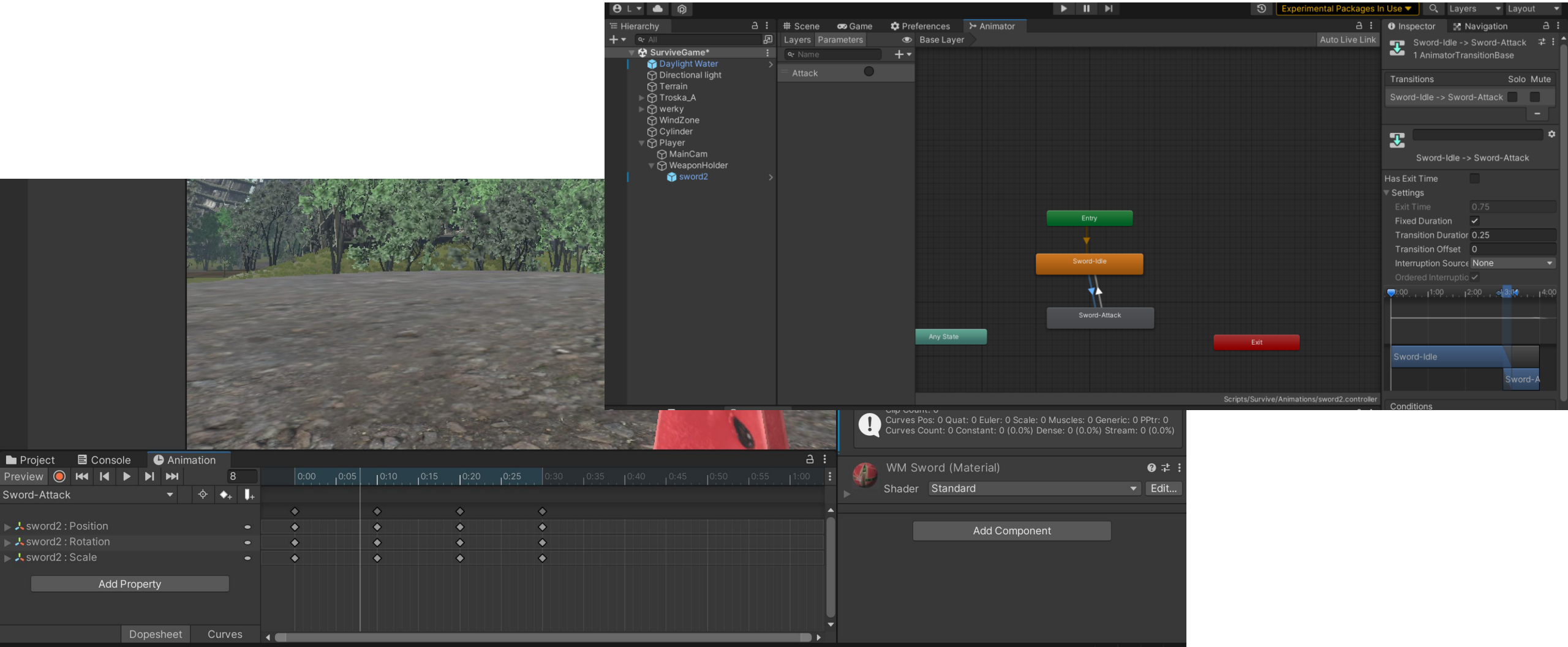
Damage Controller →

```
public class DamageController : MonoBehaviour
{
    [SerializeField]
    private Image HealthBar;
    public float Health = 100f;
    [SerializeField]
    float damage;
    [SerializeField]
    GameObject Enemy;

    Unity Message | 0 references
    void OnTriggerEnter(Collider other)
    {
    HealthBar.fillAmount -= damage / Health;
        takeDamage();
    if (HealthBar.fillAmount <= 5.0f) Death();

    }
    1 reference
    public void Death()
    {
        Animator anim = Enemy.GetComponent<Animator>();
        anim.SetTrigger("Death");
    }
    1 reference
    public void takeDamage()
    {
        Animator anim = Enemy.GetComponent<Animator>();
        anim.SetTrigger("takeDamage");
    }
    }
```

Progress report



The Sword animations

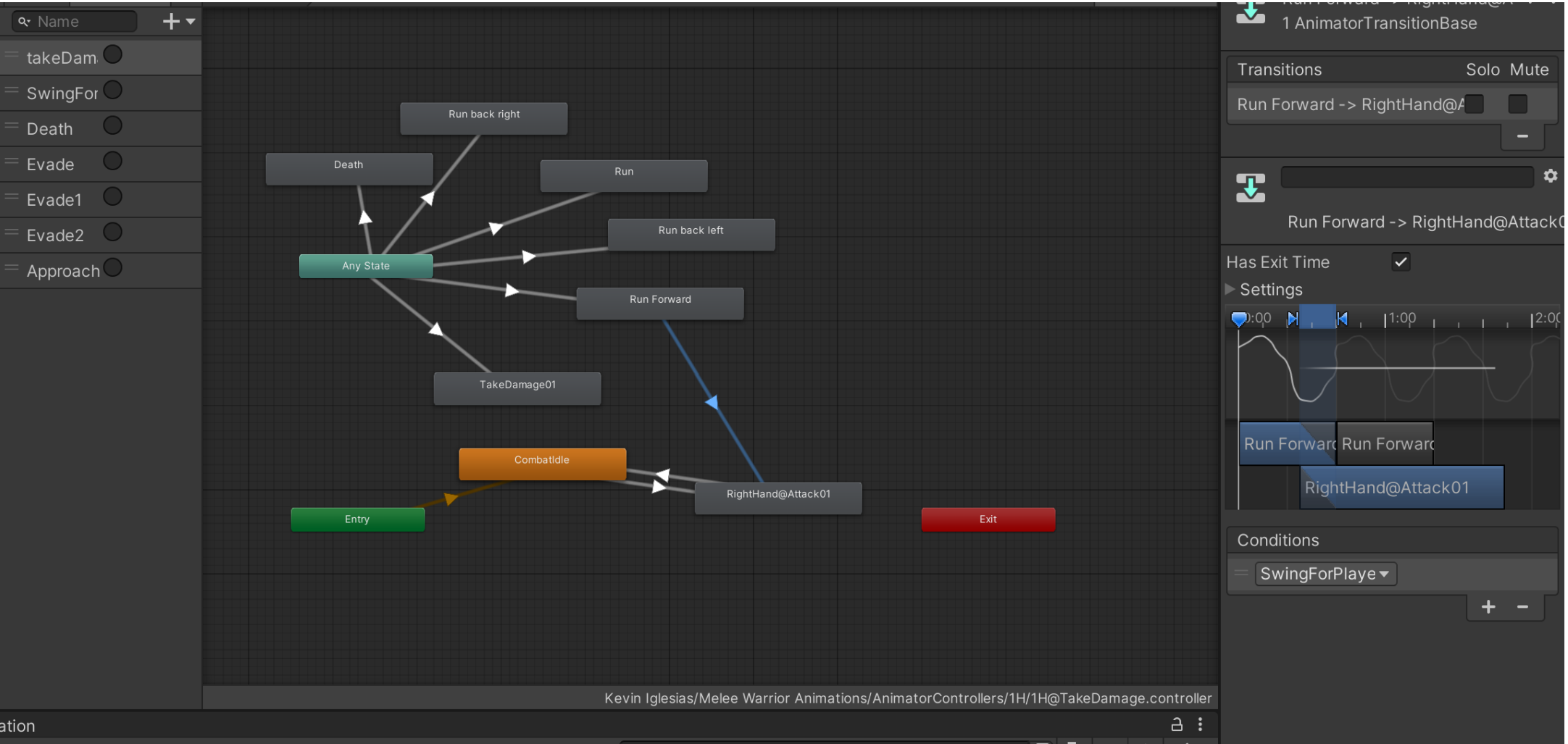


Additions to the Enemy Fight pattern

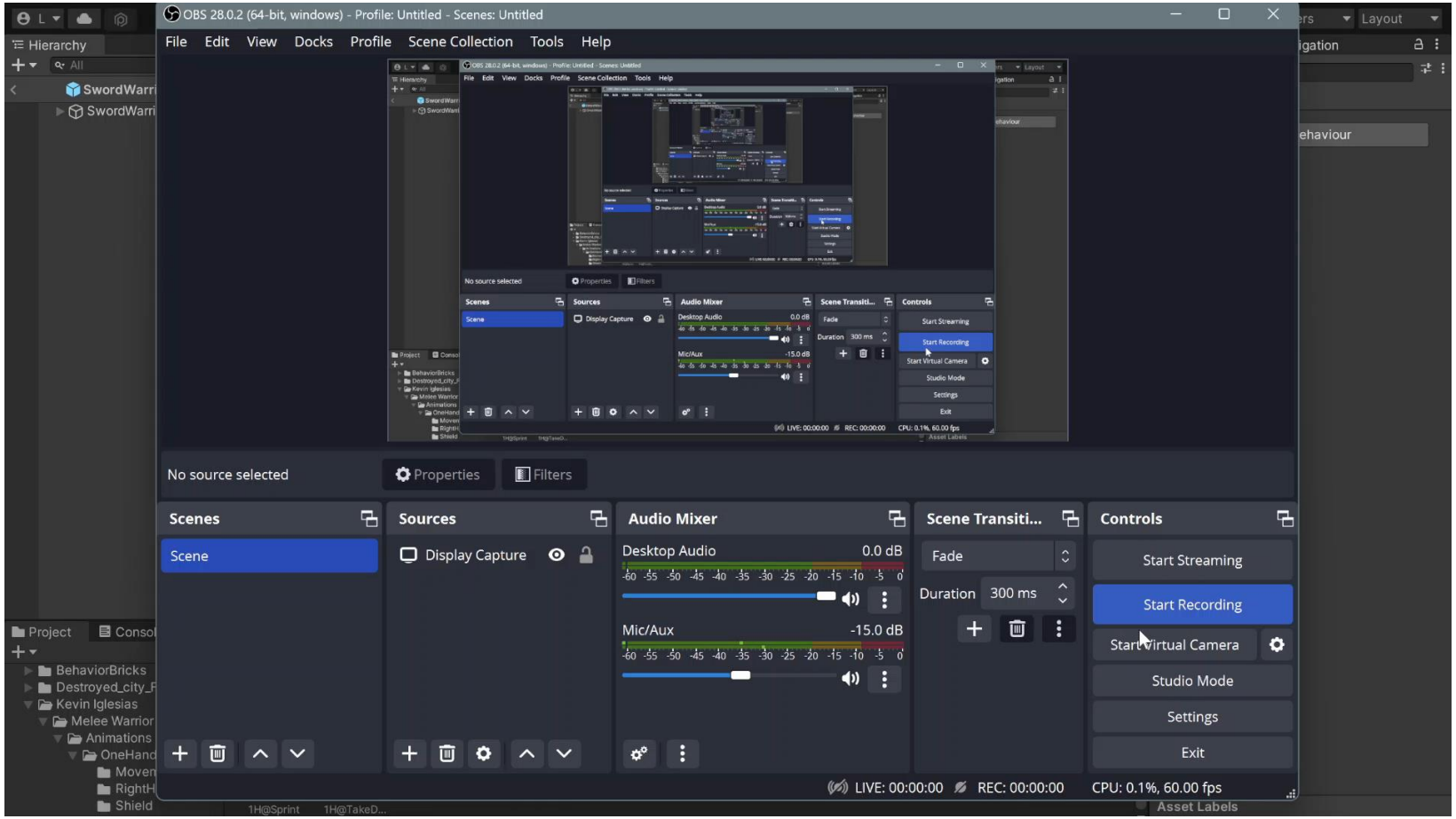
- Added an Approach animation
  - With a respective cooldown coroutine
- Added random evasion directions

```
Unity Message | 0 references
private void OnTriggerExit(Collider other)
{
    if (CanApproach)
    {
        Approach();
    }
}

1 reference
void Approach()
{
    Animator anim = Enemy.GetComponent<Animator>();
    anim.SetTrigger("Approach");
    StartCoroutine(ResetApproachCooldown());
}
1 reference
public void Swing()
{
    Animator anim = Enemy.GetComponent<Animator>();
    anim.SetTrigger("SwingForPlayer");
    StartCoroutine(ResetAttackCooldown());
}

2 references
public void Block()
{
    int choice = Random.Range(1, 4);
    Animator anim = Enemy.GetComponent<Animator>();
    if (choice == 1) anim.SetTrigger("Evade");
    else if (choice == 2) anim.SetTrigger("Evade1");
    else if (choice == 3) anim.SetTrigger("Evade2");

    StartCoroutine(ResetBlockCooldown());
```

The Enemies animations



WHAT I ENDED UP WITH



REFLECTION

- I was not able to:
  - Add Parrying
  - Fine tune the hitboxes so blocking worked reliably
- Programming responsive behaviors is difficult
- With the use of visual sensors you can crudely mimic real time combat
  - The enemy ends up resembling a zombie
- I ended up using the animator as a behavior tree
- With a combination of colliders and collisions I handled damage.
  - Definitely not an amazing way but it worked

Conclusion

- Game dev is hard
- Animation is hard
- Making accurate “hitboxes” is hard
- I have a new respect for games like Elden ring or other fast paced combat games

Where’d I get my assets?

- https://assetstore.unity.com/packages/3d/animations/melee-warrior-animations-free-165785
- https://assetstore.unity.com/packages/3d/environments/sci-fi/destroyed-city-free-6459
- https://assetstore.unity.com/packages/3d/props/weapons/watermelon-sword-191078

ANY QUESTIONS?