



Kin Fantasia

Culmination Project

Su Huan Li



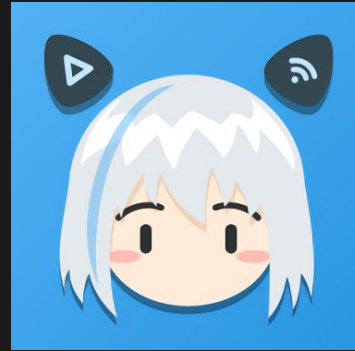
Project About

Create a simple basic RPG-style combat system. That includes one player character movement, attack, heal, and have an AI monster such as a non-boss and boss that detects the player and combat with them and each monster will have different attack skills and damage.

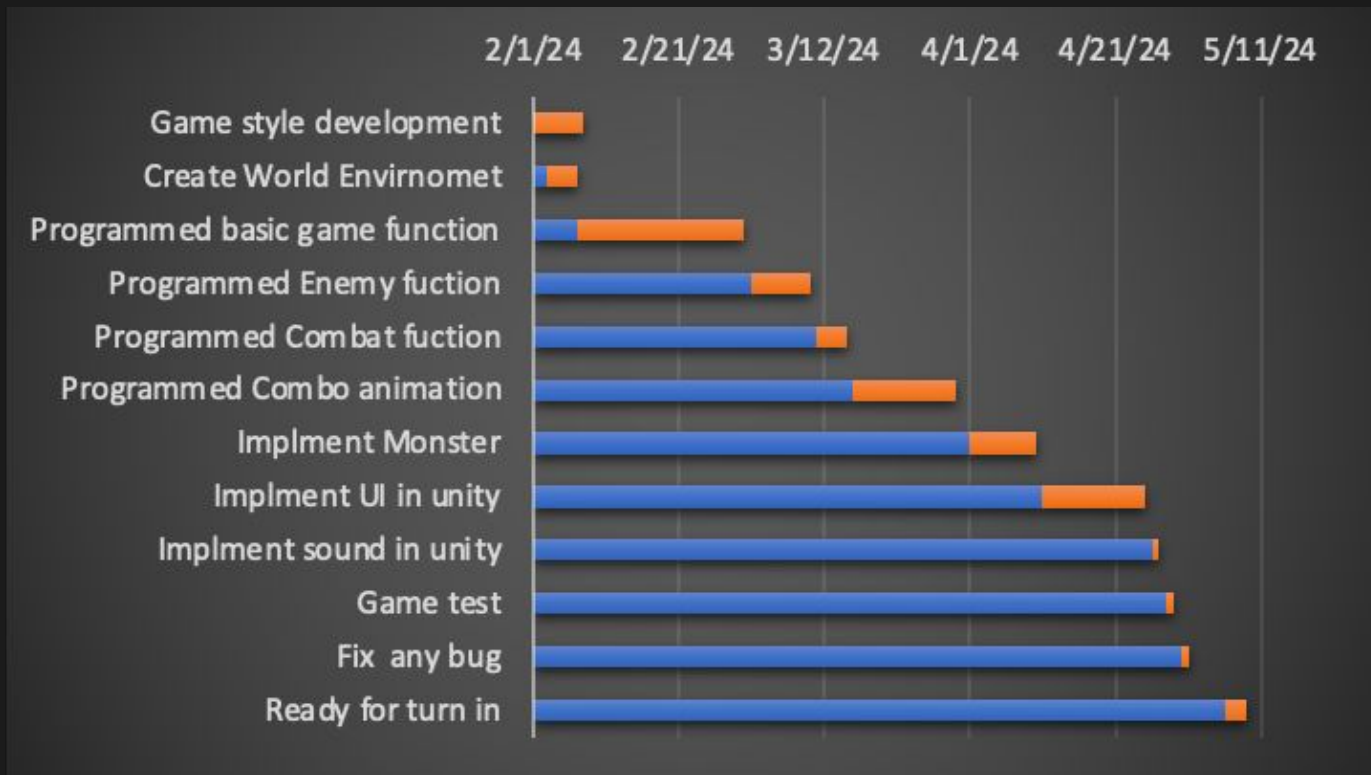


RESOURCES

- Unity
- Unity Asset Store
- Blender (3d model)
- House of model
- Mixamo
- Freesound
- A PC
- Google



Schedule



AI Components

Navigation

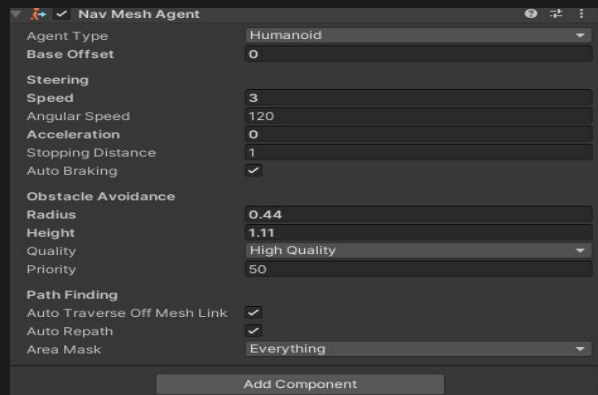
The navigation system enables you to create characters that can intelligently move around the game world. It uses navigation meshes that are automatically generated scene geometry.

The blue area indicates the walkable area.



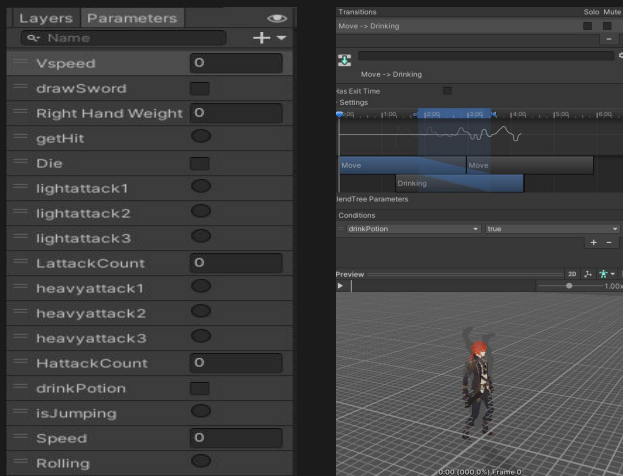
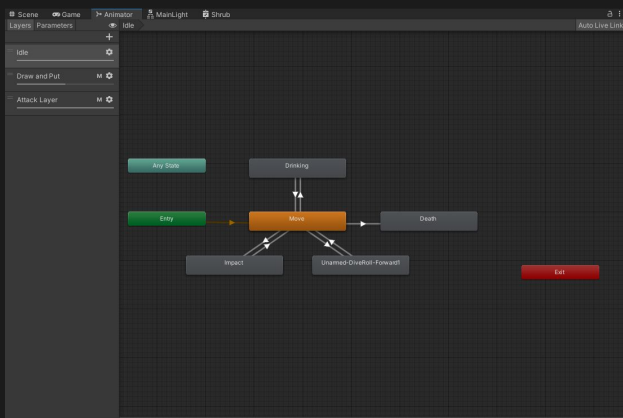
NavMeshAgent

This component helps characters navigate a scene using the NavMesh to avoid obstacles while moving toward their goal. It also assists in setting up a target destination, auto pathfinding if available, and determining the stopping distance when they are a certain distance away from something.



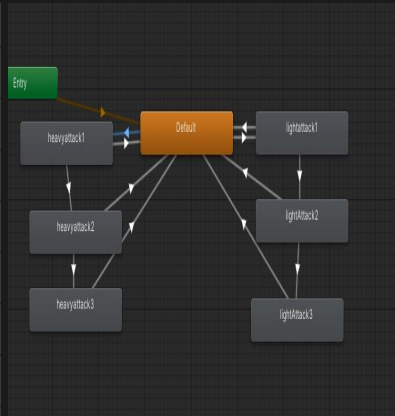
Animator Controllers

The Animator Controller manages animation clips and transitions for a character or object. It controls the logic of an animated GameObject. For example, you can use an Animator Controller to switch between animations when certain game conditions occur. For instance, you could switch from a walk Animation Clip to a jump Animation Clip whenever the spacebar is pressed.



Player Combat Logic

lightattack1	<input type="radio"/>
lightattack2	<input type="radio"/>
lightattack3	<input type="radio"/>
LattackCount	0
heavyattack1	<input type="radio"/>
heavyattack2	<input type="radio"/>
heavyattack3	<input type="radio"/>
HattackCount	0



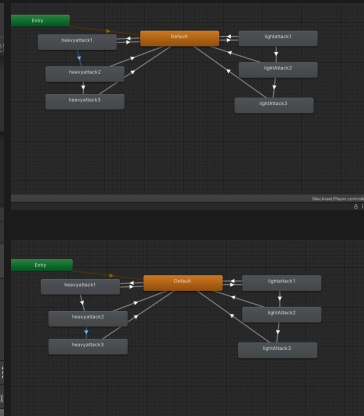
Settings

Default

Conditions

heavyattack1

Preview



Conditions

heavyattack2

HattackCount

Preview

```

private void LightAttack()
{
    if (drawSword && !isAttacking) // Check if the sword is drawn and no attack animation is currently playing
    {
        if (Time.time - lastLightAttackTime > lightCooldownTime)
        {
            ResetAttackCounts(); // Reset attack counts if cooldown time has passed
        }

        if (LattackCount >= 4)
        {
            ResetAttackCounts(); // Reset attack counts if maximum attacks reached
        }
        else
        {
            Debug.Log("Light attack on cooldown!");
            StartCoroutine(PlayLightAttackAnimations()); // Start coroutine to play light attack animations
        }
    }
}
    
```

```

private IEnumerator PlayLightAttackAnimations()
{
    isAttacking = true; // Set attacking flag to true

    LattackCount++; // Increment the attack count
    animator.SetInteger("LattackCount", LattackCount); // Set the attack count parameter in the animator

    // Play light attack animations based on attack count
    if (LattackCount == 1)
    {
        animator.SetTrigger("lightattack1");
        yield return new WaitForSeconds(0.3f);
    }
    else if (LattackCount == 2)
    {
        animator.SetTrigger("lightattack2");
        yield return new WaitForSeconds(0.3f);
    }
    else if (LattackCount == 3)
    {
        animator.SetTrigger("lightattack3");
        yield return new WaitForSeconds(1f);
    }

    // Wait for the animation to start before proceeding
    yield return new WaitForSeconds(0.1f); // Adjust delay as needed

    lastLightAttackTime = Time.time; // Update the last attack time
    isAttacking = false; // Set attacking flag to false after animations complete
}
    
```

```

private void ResetAttackCounts()
{
    LattackCount = 0; // Reset light attack count
    animator.SetInteger("LattackCount", LattackCount); // Update the attack count parameter in the animator

    HattackCount = 0; // Reset heavy attack count
    animator.SetInteger("HattackCount", HattackCount); // Update the attack count parameter in the animator
}
    
```



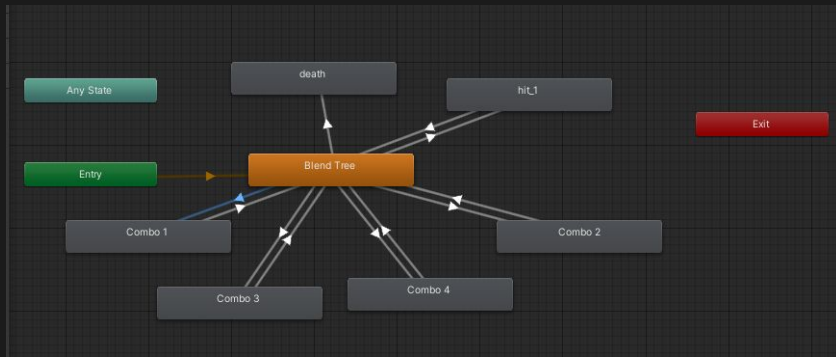
Enemy Combat Logic

Scene Game Anim

Layers Parameters

Name +

- speed 0
- hit
- death
- kick
- Combo1
- Combo2
- Combo3
- Combo4



Blend Tree -> Combo 1

Has Exit Time

Settings

10:00 1:00 2:00 3:00 4:00 5:00 6:00 7:00

Blend Tree

Combo 1

BlendTree Parameters

Conditions

Combo1

Preview

2D 1.00x

```
foreach (var target in colliders)
{
    if (target.CompareTag("Player"))
    {
        attackTarget = target.gameObject;
        //animator.SetFloat("speed", 1);
        agent.destination = attackTarget.transform.position;

        float distanceToTarget = Vector3.Distance(transform.position, attackTarget.transform.position);

        if (distanceToTarget > 1.5f && distanceToTarget < sightRadius)
        {
            agent.isStopped = false;
            sightRadius = 50f;
            agent.SetDestination(attackTarget.transform.position);
            animator.SetFloat("speed", 1);
            agent.speed = 3f;
            agent.acceleration = 3f;
            if (!hasPlayedfoundlayerAudio && audioSource != null && foundlayerAudio != null)
            {
                audioSource.PlayOneShot(foundlayerAudio);
                hasPlayedfoundlayerAudio = true; // Set the flag to true after playing the audio
            }
            // Rotate towards the player's position
            Vector3 direction = (attackTarget.transform.position - transform.position).normalized;
            Quaternion lookRotation = Quaternion.LookRotation(new Vector3(direction.x, 0, direction.z));
            transform.rotation = Quaternion.Slerp(transform.rotation, lookRotation, Time.deltaTime * 5f);
        }
    }
}
```

```
if (distanceToTarget <= 2f)
{
    agent.stoppingDistance = 2f;
    animator.SetFloat("speed", 0);
    agent.isStopped = true;
    agent.speed = 0;
    agent.acceleration = 0f;
    agent.velocity = Vector3.zero;
    agent.angularSpeed = 0f;
    if (canAttack && currentCooldown <= 0)
    {
        // Randomly select the attack animation
        int randomNum = UnityEngine.Random.Range(1, 5);

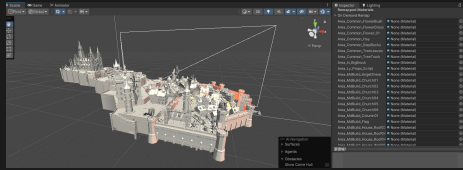
        switch (randomNum)
        {
            case 1:
                animator.SetTrigger("Combo1");
                break;
            case 2:
                animator.SetTrigger("Combo2");
                break;
            case 3:
                animator.SetTrigger("Combo3");
                break;
            case 4:
                animator.SetTrigger("Combo4");
                break;
            default:
                break;
        }
        //animator.SetTrigger("attack");
        currentCooldown = attackCooldown;
        canAttack = true;
    }
}
```


Game Overview



Challenges

Importing Assets



Enemy AI behavior



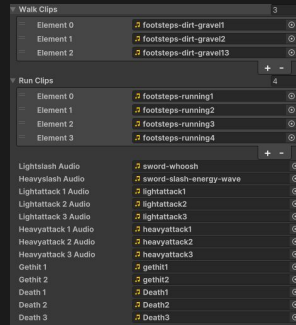
NavMesh



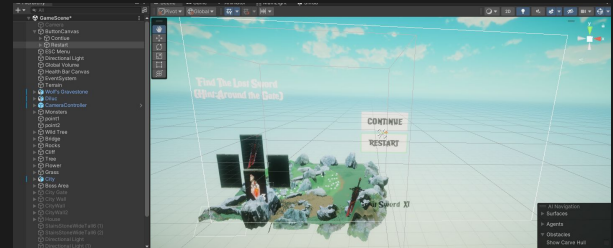
Animation Effect



Audio



UI



Conclusion

Creating Kin Fantasia was a valuable learning experience that provided me with essential knowledge in developing 3D games. The skills I gained are applicable to 2D game development as well, including scene construction, animation creation and adjustment, UI design, special effects, sound effects, character behaviors, and more. The challenges I encountered during the project have helped me improve my problem-solving skills and deepen my understanding of game development. Although time constraints prevented me from completing the combat system and turning the project into a full game, I am eager to continue working on it to add more interactive features, expand the terrain, introduce additional characters, enemies, weapons, animations, story elements, and more to make it a more complete RPG game.





THANK YOU!

