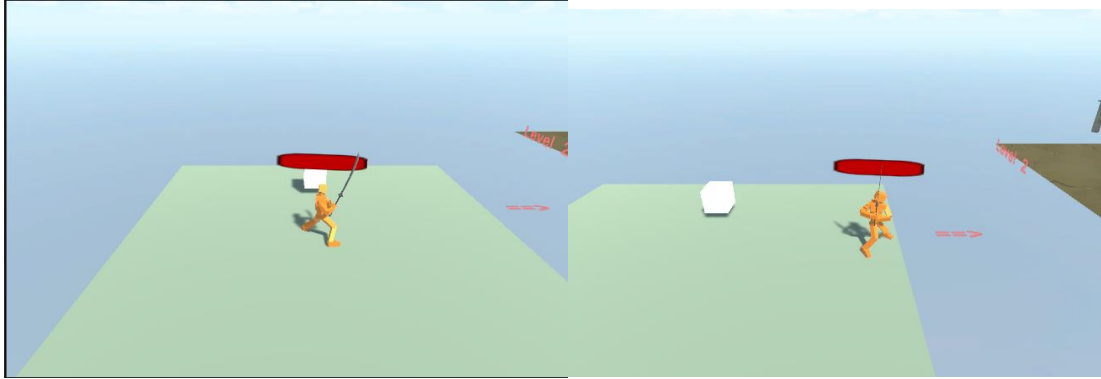
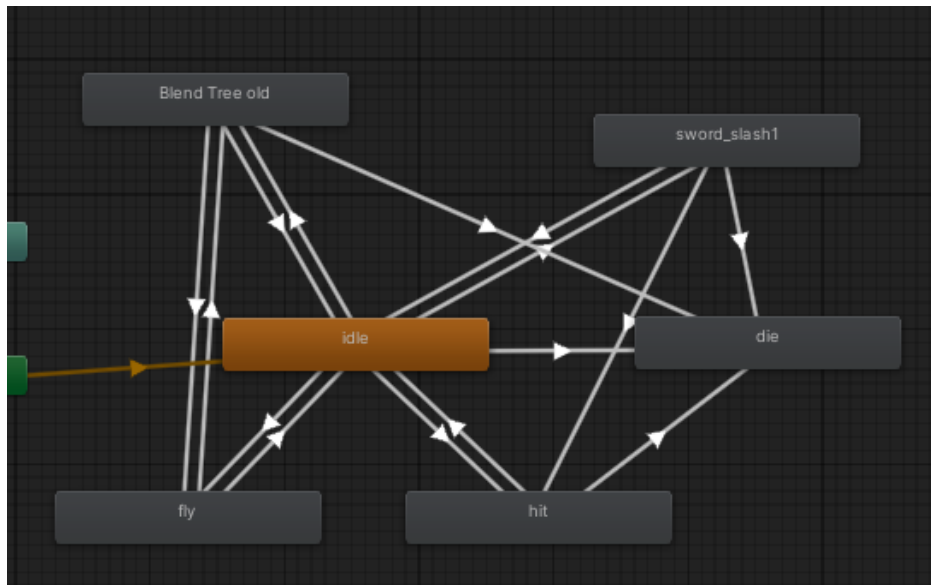


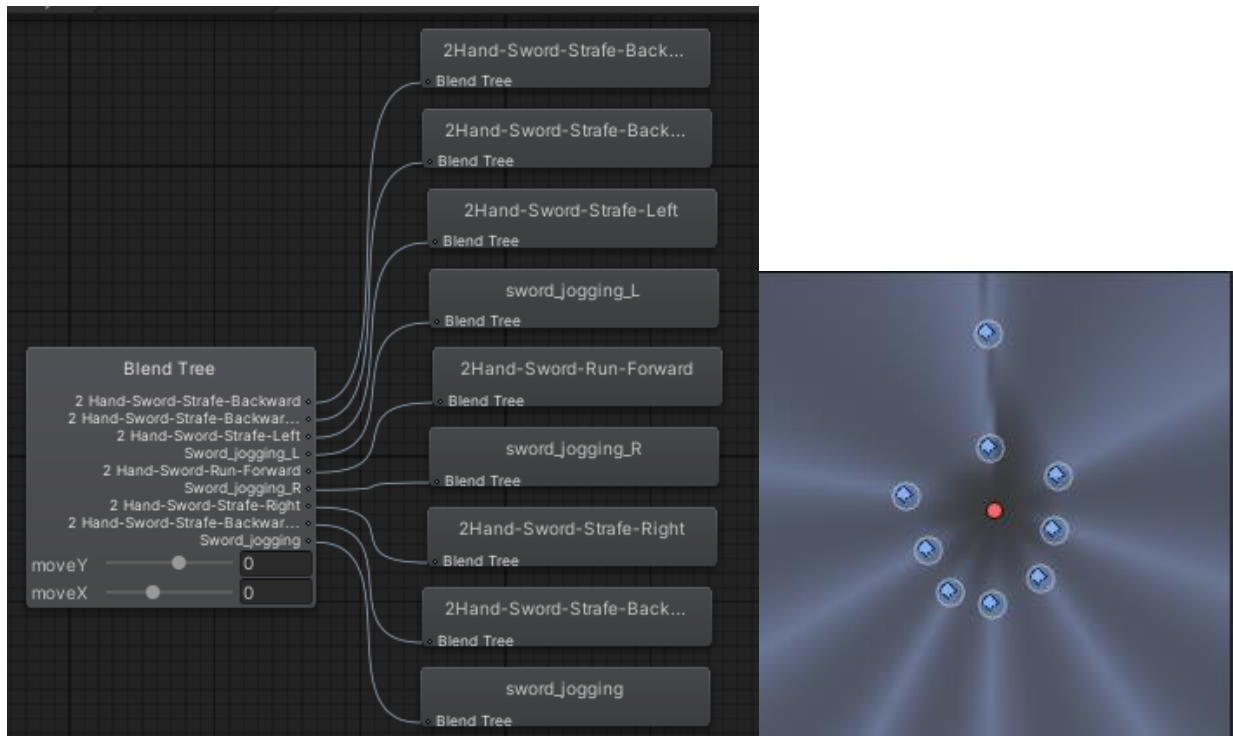
RPG game

1) Camera control (15%) –2) Basic movement and control (15%).



3) Animations





- Enterance: $\text{velocity.magnitude} > 0.01$
- 9 points of blend tree: walk and run and 45 degrees
- The agent's velocity, acceleration, and angular velocity has optimized for best reaction.

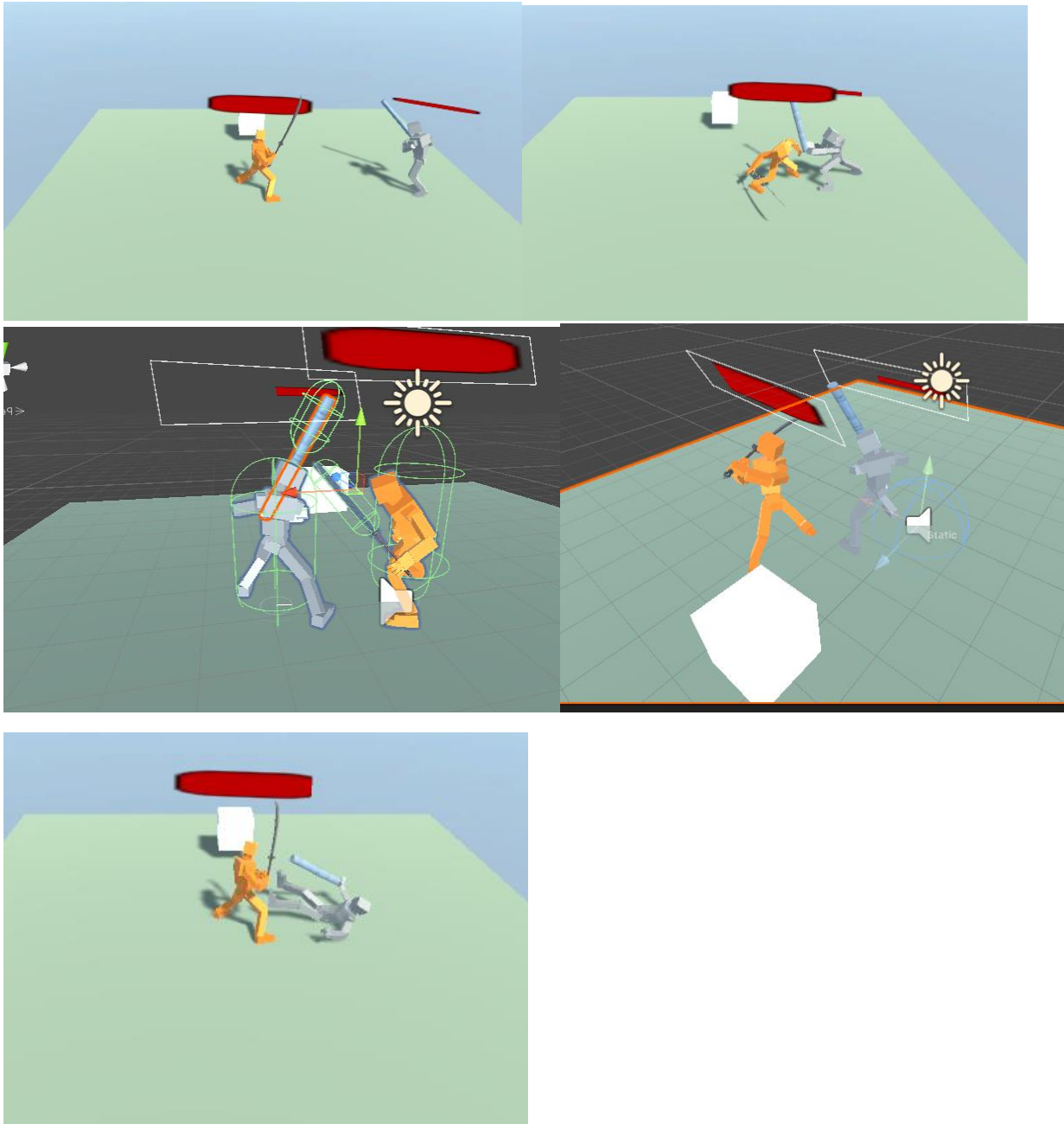
```
//---- movement for blender
if ((agent.velocity.magnitude / agent.speed) > 0.1)
{
    Vector3 normalizedMovement = agent.desiredVelocity.normalized;
    Vector3 forwardVector = Vector3.Project(normalizedMovement, transform.forward);
    Vector3 rightVector = Vector3.Project(normalizedMovement, transform.right);
    float forwardVelocity = forwardVector.magnitude * Vector3.Dot(forwardVector, transform.forward);
    float rightVelocity = rightVector.magnitude * Vector3.Dot(rightVector, transform.right);
    anim.SetFloat("moveX", forwardVelocity);
    anim.SetFloat("moveY", rightVelocity);
}
```

Code Descriptions of blend tree:

- Blend tree gets moveX and moveY which are the project of the desired agent velocity on the current transform of the agent. The desiredVelocity is a nav mesh agent component, based on the destination of agent that clicked by user or the place of player for enemies.

4) combat system

point and clicking - run and attack - enemy fight back - collision detection - additional animation - dying

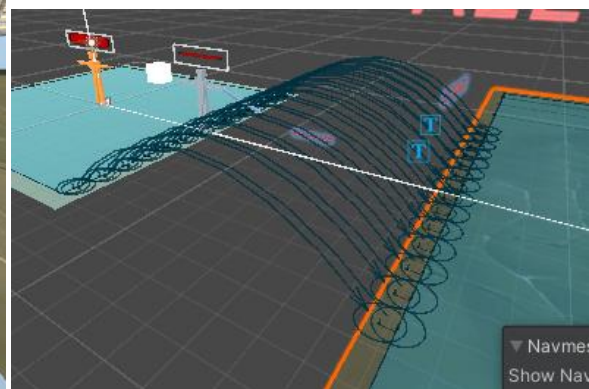
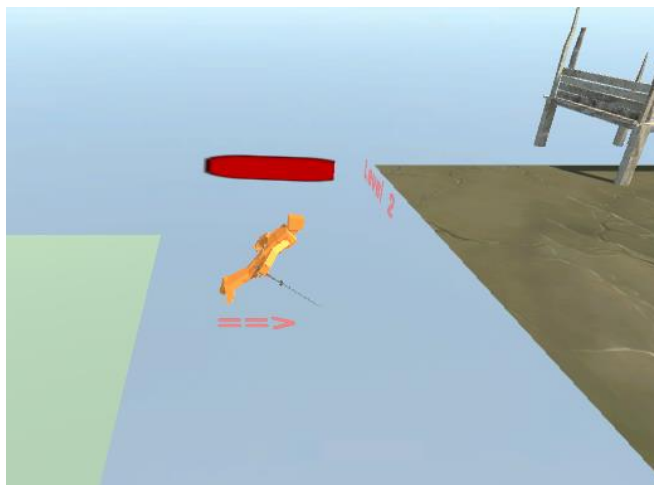
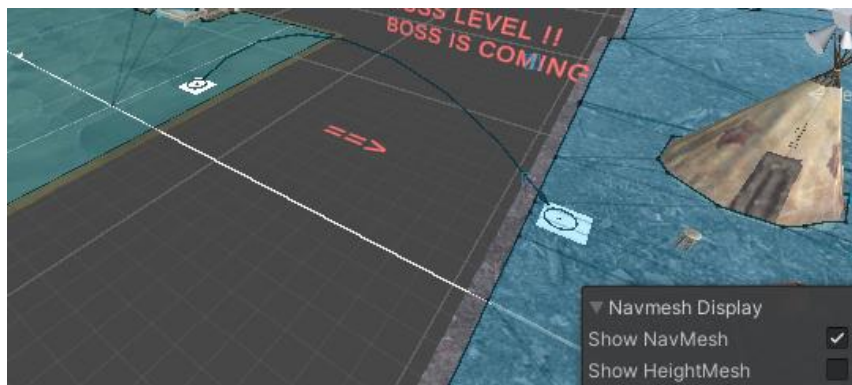


5) User interface – main menu



-- Bounces parts:

- off-mesh links with animation:



- multiple enemies challenge:



- Leveling system: the Player became bigger and stronger at level 2 after killing some enemies,



- 3 different 'zones', which comes after each other, and a final orc boss at the end.



Code Descriptions:

Player controller: Movement is by ray casting. And hit layer.

```
if (Input.GetMouseButton(0))
{
    RaycastHit hit;
    Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    if (Physics.Raycast(ray, out hit, 100.0f))
    {
        hitPoint = hit.point;
        if(hit.transform.tag == "Level")
        {
            agent.destination = hit.point;
            attacking = false;
        }
        else if (hit.transform.tag == "enemy")
        {
            enemyTransform = hit.transform;
            enemyObject = hit.transform.gameObject;

            Vector3 between = agent.transform.position - enemyTransform.position;
            attacking = true;
            if ((between.magnitude) > 2)
            {
                agent.destination = between.normalized * 2f + enemyTransform.position;
            }
        }
    }
}
```

Enabling to attacking near enemies on click:

```
if (enemyObject != null && attacking == true && (enemyTransform.position - agent.transform.position).magnitude < 4)
{
    anim.SetBool("attacking", true);
    agent.transform.LookAt(enemyTransform.position);
}
else
{
    anim.SetBool("attacking", false);
}
```

Camera : camera transform following playe + zooming and panning

```
Camera.main.transform.position += agent.transform.position - prevPos;
prevPos = agent.transform.position;
```

```
///---camera
if (Input.GetMouseButton(1))
{
    float mouseX = Input.GetAxis("Mouse X") * 500 * Time.deltaTime;
    float mouseY = Input.GetAxis("Mouse Y") * 500 * Time.deltaTime;
    xRotation -= mouseY;
    yRotation = mouseX;
    xRotation = Mathf.Clamp(xRotation, 90, 90);

    if (Mathf.Abs(mouseX) > Mathf.Abs(mouseY))
        Camera.main.transform.RotateAround(agent.transform.position, transform.up, yRotation);
    else
        Camera.main.transform.RotateAround(agent.transform.position, Camera.main.transform.right, mouseX);
}

if (Input.GetAxis("Mouse ScrollWheel") != 0)
{
    float scroll = -Input.GetAxis("Mouse ScrollWheel");
    Vector3 ray = Camera.main.transform.position - agent.transform.position;
    Camera.main.transform.position += ray * scroll;
}
```


Player life script: Initialize health at start and in update check for dying or not:

```
void Start()
{
    life = 150;
    anim = GetComponent<Animator>();
}
void Update()
{
    if (life == 0)
    {
        anim.SetBool("die", true);
        Destroy(gameObject, 1.5f);
        Invoke("QuitGame", 3.0f);
        death.Play();
    }
}
```

Also Checks what is triggering the collider for different damages + adjust the health bar UI:

```
private void OnTriggerEnter(Collider other)
{
    if (other.transform.name == "woodE")
    {
        AAY.Play();
        anim.SetBool("gethit", true);
        life = life - 1;
        lifebar.fillAmount = (float)life / 150;
    }
    if (other.transform.name == "sword")
    {
        anim.SetBool("gethit", true);
        life = life - 2;
        lifebar.fillAmount = (float)life / 150;
    }
    if (other.transform.name == "bil")
    {
        bil.Play();
        anim.SetBool("gethit", true);
        life = life - 3;
        if (life < 0) life = 0;
        lifebar.fillAmount = (float)life / 150;
    }
}
private void OnTriggerExit(Collider other)
{
    anim.SetBool("gethit", false);
}
```


Enemies' agents behavior: The player (main character) Find() and locked on as enemy of the agent.

```
agent = GetComponent<NavMeshAgent>();
anim = GetComponent<Animator>();
enemyObject = GameObject.Find("Player");
}

void Update()
{
    anim.SetBool("run", (((agent.velocity.magnitude / agent.speed) > 0.01) && agent.remainingDistance > 0.1));

    if (enemyObject != null && attacking == true && (enemyTransform.position - agent.transform.position).magnitude < 3)
    {
        anim.SetBool("attacking", true);
        agent.transform.LookAt(enemyTransform.position);
    }
    else
    {
        anim.SetBool("attacking", false);
    }

    enemyTransform = enemyObject.transform;

    Vector3 between = agent.transform.position - enemyTransform.position;
    attacking = true;
    if ((between.magnitude) > 2)
    {
        agent.destination = between.normalized * 2f + enemyTransform.position;
    }
}
```

Enemei's life script: On start: pre-amount for life (base on level), deactivating the next level and vector for player upgrade after killing this enemy. + In update: check if the enemy should die or not. If yes, the next level activated and the player upgraded.

```
void Start()
{
    player = GameObject.Find("Player");
    nextL = GameObject.Find("planeC");
    nextL.SetActive(false);
    life = 5;
    a = 0;
    agent = GetComponent<NavMeshAgent>();
    anim = GetComponent<Animator>();
    scaleChange = new Vector3(1f, 1f, 1f);
}
```

```

void Update()
{
    if (life == 0)
    {
        if (a == 0)
        {
            player.transform.localScale += scaleChange;
            player.GetComponent<NavMeshAgent>().speed += 4;
            a = 1;
            death.Play();
            nextLaudio.Play();
        }
        anim.SetBool("die", true);
        nextL.SetActive(true);
        Destroy(gameObject, 1.5f);
    }
}

```

```

private void OnTriggerEnter(Collider other)
{
    if (other.transform.name == "wood")
    {
        AAY.Play();
        anim.SetBool("gethit", true);
        life = life - 1;
        lifebar.fillAmount = (float)life / 5;
    }
}

private void OnTriggerExit(Collider other)
{
    anim.SetBool("gethit", false);
}

```

Off-mesh link animation:

```
if (agent.isOnOffMeshLink)
{
    anim.SetBool("fly", true);
}
else
{
    anim.SetBool("fly", false);
}
```

Health bars' transcript:

```
void Update()
{
    transform.LookAt(Camera.main.transform);
}
```

Thank you kindly for your time!

Finish.

Finish.

Green Salar
Salar.rezayani.com