REM—a third-person mini game based on Unity3D

Developer intent

The game draws materials from a Japanese animation and the player will play the role of one of the characters in this animation. The game is designed for mobile platforms and should be a mini one which means simple and easy to control. The player should use the weapon and skills to kill enemies and the game will become increasingly harder as time goes on which will be reflected by levels on the screen. Just survive as much time as possible. The levels are also seen as your result.

Scope

C# language Unity3D engine MonoDevelop IDE

MagicaVoxel for modeling

User guide

Use the touchpad on the screen to move and slide to turn your view (**Input and output**). Try to swing your bola (your weapon) to destroy enemies which will keep showing up and moving towards you. The enemies will be killed by one hit but you also will take damage if touching with them. The difficulty increases with the levels but you will get fully healed every time the levels +1. You can use the skill RUSH to make a faster move in a short time but there is a cooldown of 10 seconds. The ultimate skill RAM is available every time you kill 10 enemies. Try to save it for dangerous situation.

Code

**//Detect the collision between bola and enemies. Destroy enemies on hit and change values in another class**

public class Ball : MonoBehaviour  
{  
  
    void OnTriggerEnter (Collider coll)  
    {  
        if (coll.tag == "Enemy") {  
            coll.GetComponent<FFFCtrl> ().Hurt ();  
            GameObject.Find ("SP").GetComponent<Ultimate> ().count++;  
        }  
    }  
}

**//Alike the class above. Detect the collision between magic balls (Ultimate skill) and enemies. Destroy both on hit and show the death effect.**

public class contactEnemy : MonoBehaviour  
{  
    *//public int Speed;*  
    public GameObject blood;  
  
    void OnTriggerEnter (Collider other)  
    {  
  
        if (other.tag == "Enemy") {  
            Destroy (other.gameObject);  
            Destroy (this.gameObject);  
            Instantiate (blood, transform.position, Quaternion.identity);  
        }  
    }  
  
}

**//Control the move of enemies and define a class to make self-destroyed.**

public class FFFCtrl : MonoBehaviour  
{  
    public Transform player;  
    public float rotSpd;  
    public float movSpd = 7.0f;  
    public GameObject blood;  
  
    *// Use this for initialization*  
    void Start ()  
    {  
        player = GameObject.Find ("REM").transform;  
    }  
      
    *// Update is called once per frame*  
    void Update ()  
    {  
        Vector3 targetDir = player.position - this.transform.position;  
        float step = rotSpd \* Time.deltaTime;  
        Vector3 newDir = Vector3.RotateTowards (transform.forward, targetDir, step, 0.0f);  
        transform.rotation = Quaternion.LookRotation (newDir);  
        transform.Translate (Vector3.forward \* Time.deltaTime \* movSpd);  
    }  
  
    public void Hurt ()  
    {  
        Destroy (gameObject);  
        Instantiate (blood, transform.position, Quaternion.identity);  
    }  
}

**//Control the move of the player and also detect the collision between the player and enemies. Define a class to take damage on hit.**

public class RemCtrl : MonoBehaviour  
{  
    *// Use this for initialization*  
  
    public float Speed = 4;  
    *//public float rotSpeed = 120;*  
    public RawImage logo;  
    public GameObject blood;  
    public Slider HPStrip;  
    public int HP = 10;  
  
    void Start ()  
    {  
        Time.timeScale = 1;  
        HPStrip.value = HPStrip.maxValue;   
    }  
      
    *// Update is called once per frame*  
    void Update ()  
    {  
        transform.position += transform.forward \* Speed \* Time.deltaTime;  
  
    }  
  
    void OnTriggerEnter (Collider coll)  
    {  
        if (coll.tag == "Enemy") {  
            Instantiate (blood, transform.position, Quaternion.identity);  
            HealthDmg (2);  
            logo = logo.GetComponent<RawImage> ();  
            logo.color = Color.red;  
        }  
    }  
  
    void OnTriggerExit (Collider coll)  
    {  
        if (coll.tag == "Enemy") {  
            logo = logo.GetComponent<RawImage> ();  
            logo.color = Color.white;  
        }  
    }  
  
    void HealthDmg (int damage)  
    {  
        HP -= damage;  
        HPStrip.value = HP;   
        if (HP == 0) {  
            GameObject.Find ("SceneManager").GetComponent<ExitManager> ().GameOver ();  
        }  
    }  
}

**//Control the move of RAM (Ultimate skill) and define a class to fire magic balls to kill enemies.**

public class RamCtrl : MonoBehaviour  
{  
    public Transform player;  
    public float rotSpd;  
    public float movSpd = 6.5f;  
    public GameObject magicBall;  
    public int count;  
  
    void Start ()  
    {  
        player = GameObject.Find ("REM").transform;  
        InvokeRepeating ("Fire", 0, 0.5f);  
    }  
  
    void Update ()  
    {  
        Vector3 targetDir = transform.position - player.position;  
        float step = rotSpd \* Time.deltaTime;  
        Vector3 newDir = Vector3.RotateTowards (transform.forward, targetDir, step, 0.0f);  
        transform.rotation = Quaternion.LookRotation (newDir);  
        transform.Translate (Vector3.forward \* Time.deltaTime \* -movSpd);  
  
    }  
  
    void Fire ()  
    {  
        GameObject bullet1 = Instantiate (magicBall, this.transform.position + new Vector3 (0, 2, 0), Quaternion.identity) as GameObject;  
        GameObject bullet2 = Instantiate (magicBall, this.transform.position + new Vector3 (0, 2, 0), Quaternion.identity) as GameObject;  
        GameObject bullet3 = Instantiate (magicBall, this.transform.position + new Vector3 (0, 2, 0), Quaternion.identity) as GameObject;  
        bullet1.GetComponent<Rigidbody> ().velocity = (transform.forward + new Vector3 (1, 0, 0)) \* -10;  
        bullet2.GetComponent<Rigidbody> ().velocity = (transform.forward + new Vector3 (0, 0, 1)) \* -10;  
        bullet3.GetComponent<Rigidbody> ().velocity = this.transform.forward \* -14;  
        Destroy (bullet1, 4);  
        Destroy (bullet2, 4);  
        Destroy (bullet3, 4);  
        count++;  
        if (count > 5) {  
            this.CancelInvoke ();  
            count = 0;  
        }  
    }  
}

**//Manage the level and make enemies show up randomly**

public class levelManager : MonoBehaviour  
{  
  
    static public levelManager lm;  
    public Transform player;  
    public GameObject enemy;  
    public Text Level;  
  
    public float rateTime = 1.8f;  
  
    private float countTime;  
    private float totalTime;  
    private int level = 0;  
  
    void Awake ()  
    {  
        lm = this;  
    }  
    void Update ()  
    {  
        countTime += Time.deltaTime;  
        totalTime += Time.deltaTime;  
  
        Level.GetComponent<Text> ().text = "Lv." + level.ToString ();  
  
        if (countTime > rateTime) {  
            Vector2 r = Random.insideUnitCircle.normalized \* 30;  
            Instantiate (enemy, player.position + new Vector3 (r.x, 0, r.y), Quaternion.Euler (new Vector3 (0, Random.Range (0.0f, 360.0f), 0)));  
            countTime -= rateTime;  
        }  
        if (rateTime > 0) {  
            if ((int)totalTime % 10 == 0) {  
                rateTime -= 0.1f;  
                level++;  
                GameObject.Find ("REM").GetComponent<RemCtrl> ().HPStrip.value = 10;  
                GameObject.Find ("REM").GetComponent<RemCtrl> ().HP = 10;  
                totalTime += 1;  
            }  
        }  
    }  
}

**//Manage the cooldown and effects of skills**

public class Skill : MonoBehaviour  
{  
    public Image icon;  
    public float coolDown = 10.0f;  
    public GameObject speedLine;  
  
    private float currentCoolDown;  
    private Button skillButton;  
  
    void Start ()  
    {  
        this.skillButton = this.GetComponent<Button> ();  
        skillButton.onClick.AddListener (() => this.RemSkill ());  
        currentCoolDown = coolDown;  
    }  
  
    void Update ()  
    {  
        if (currentCoolDown < coolDown) {  
            currentCoolDown += Time.deltaTime;  
            this.icon.fillAmount = currentCoolDown / coolDown;  
        }  
    }  
  
    public void RemSkill ()  
    {  
        if (currentCoolDown >= coolDown) {  
            GameObject.Find ("REM").GetComponent<RemCtrl> ().Speed += 6;  
            speedLine.SetActive (true);  
            Invoke ("SpeedDown", 3.0f);  
            currentCoolDown = 0;  
        }  
    }  
  
    void SpeedDown ()  
    {  
        GameObject.Find ("REM").GetComponent<RemCtrl> ().Speed -= 6;  
        speedLine.SetActive (false);  
    }  
}

**//Define the interaction and effects of the ultimate skill**

public class Ultimate : MonoBehaviour  
{  
    public Image icon;  
    public float coolDown = 10.0f;  
    public GameObject Ram;  
    public GameObject player;  
    public float count;  
    public GameObject textAnim;  
    public GameObject Effect;  
  
    private Button skillButton;  
    private GameObject assist;  
  
    *// Use this for initialization*  
    void Start ()  
    {  
        this.skillButton = this.GetComponent<Button> ();  
        skillButton.onClick.AddListener (() => this.UltimateSkill ());  
    }  
      
      void Update ()  
    {  
        if (count <= 10) {  
            this.icon.fillAmount = count / coolDown;  
        }  
        if (count >= coolDown) {  
            textAnim.SetActive (true);  
        }  
    }  
  
    public void UltimateSkill ()  
    {  
        if (count >= coolDown) {  
            assist = Instantiate (Ram, player.transform.position + new Vector3 (2, 0, 0), Quaternion.identity)as GameObject;  
            Instantiate (Effect, assist.transform.position, Quaternion.identity);  
            textAnim.SetActive (false);  
            count = 0;  
            Destroy (assist, 5.0f);  
                  }  
    }  
}

**//Animation of the player and enemies to make movements more vivid**

public class TextAnim : MonoBehaviour  
{  
  
    public AnimationCurve ac;  
    public float playSpeed = 1.0f;  
    float timeOffset = 0.0f;  
    Vector3 sca;  
    *// Use this for initialization*  
    void Start ()  
    {  
        sca = transform.localScale;  
        timeOffset = Random.value;  
    }  
      
    *// Update is called once per frame*  
    void Update ()  
    {  
        timeOffset += Time.deltaTime;  
        float tens = ac.Evaluate (timeOffset \* playSpeed);  
        transform.localScale = new Vector3 (sca.x \* tens, sca.y \* tens, sca.z);  
    }  
}

Illustration





