COMP-4478-WA - Game Programming Project II

Game Explanation

The game is called Dude. It is a platform based coin collecting game while avoiding or killing the obstacle.

How to Play:

Go to D:\Dude\Exe and play the Dude Exe

Controls

▶	Move player Right
◀	Move player Left
Space	Make Player Jump
Alt	Shoot Bullets

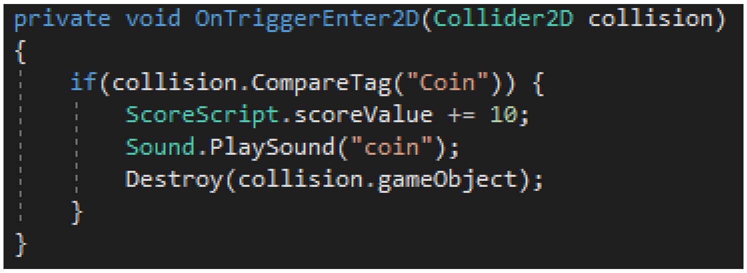
Game Objects

A. Platform : The platforms are stored as prefab

[redacted]

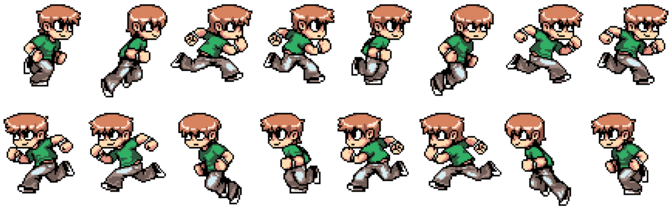
B. Coins: The coins are given an animation making them constantly rotate

When the player collides with the coin a function is triggered that increases the static score value by 10 and deletes the coin object

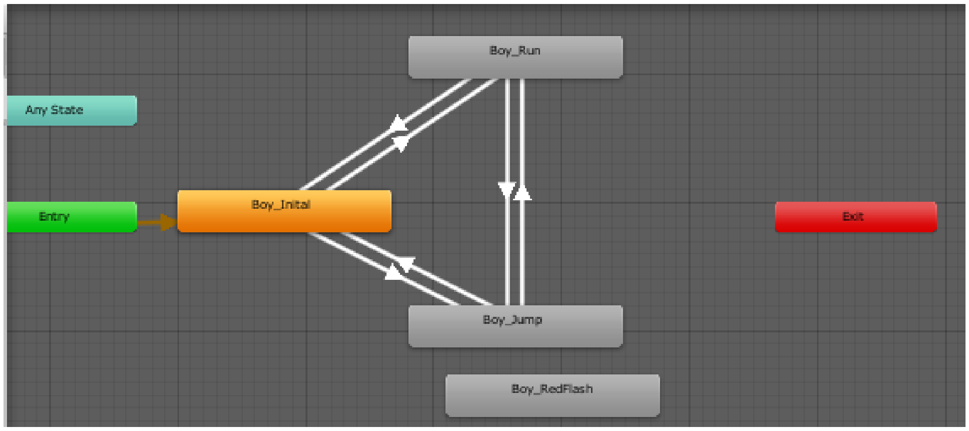


```
private void OnTriggerEnter2D(Collider2D collision)
{
    if(collision.CompareTag("Coin")) {
        ScoreScript.scoreValue += 10;
        Sound.PlaySound("coin");
        Destroy(collision.gameObject);
    }
}
```

C. Player: The Running boy sprite shown below is divided into multiple sprites with the help of sprite editor. Using the sprite the 4 animations were created transition between which is parameters based. The 2 parameters are float value speed and bool value grounded.



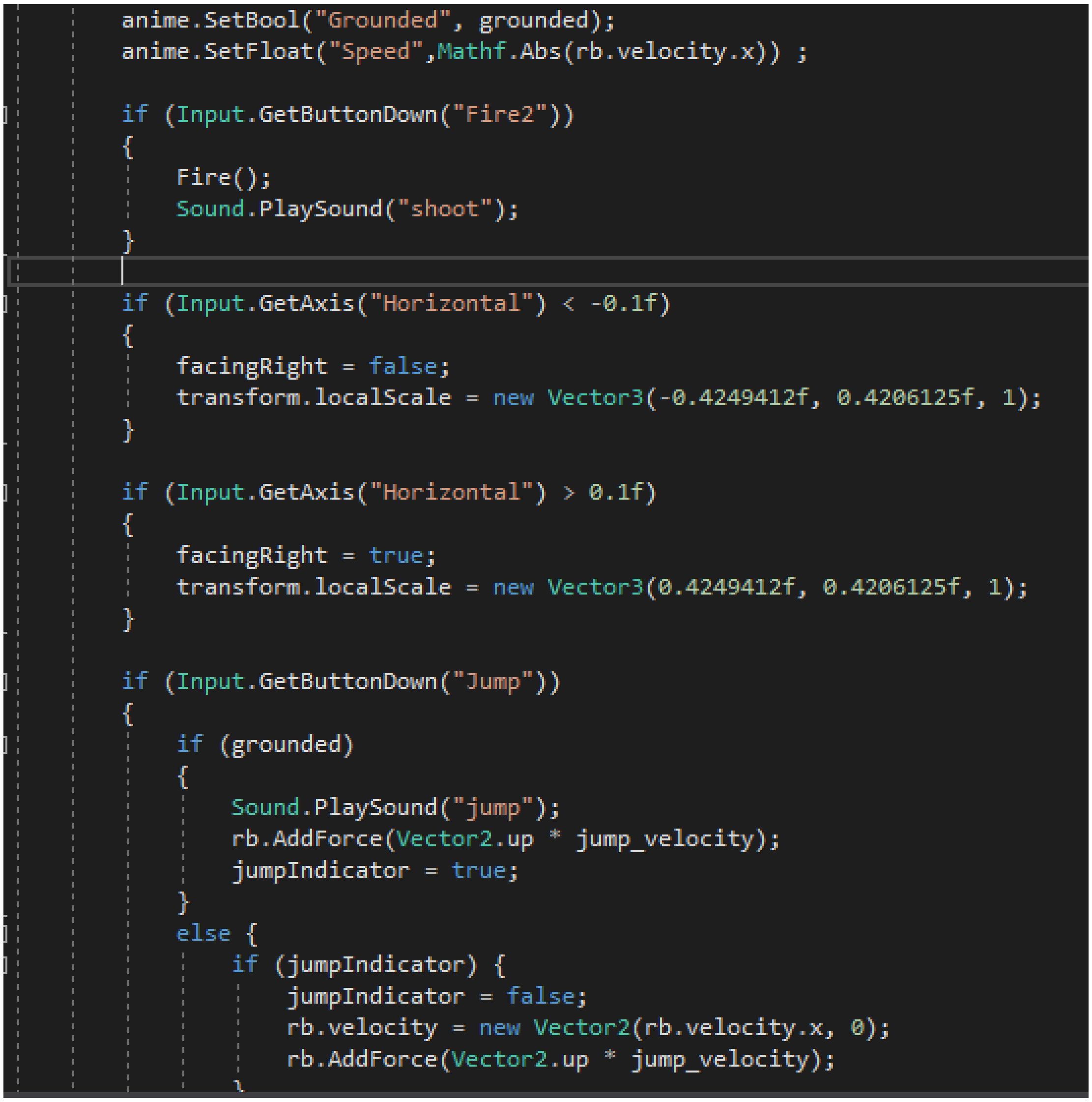
The animator chart is as shown



The player goes to run state if he is grounded and speed >0.1
The player goes to jump state if player is not grounded.
The player goes to initial state if speed is 0 and player is grounded
The RedFlash animation is played when the player hits an obstacle other than coin

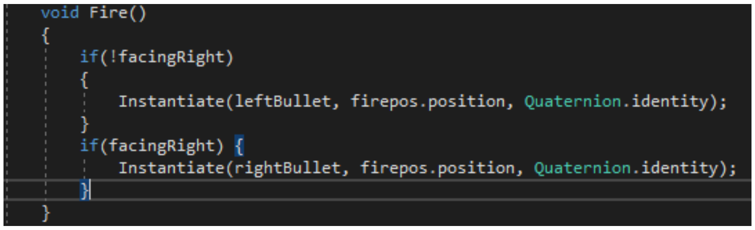
The player local scale is transform based on user input.

The code for player movement and shoot is shown below



```
        anime.SetBool("Grounded", grounded);
        anime.SetFloat("Speed",Mathf.Abs(rb.velocity.x)) ;

        if (Input.GetButtonDown("Fire2"))
        {
            Fire();
            Sound.PlaySound("shoot");
        }

        if (Input.GetAxis("Horizontal") < -0.1f)
        {
            facingRight = false;
            transform.localScale = new Vector3(-0.4249412f, 0.4206125f, 1);
        }

        if (Input.GetAxis("Horizontal") > 0.1f)
        {
            facingRight = true;
            transform.localScale = new Vector3(0.4249412f, 0.4206125f, 1);
        }

        if (Input.GetButtonDown("Jump"))
        {
            if (grounded)
            {
                Sound.PlaySound("jump");
                rb.AddForce(Vector2.up * jump_velocity);
                jumpIndicator = true;
            }
            else {
                if (jumpIndicator) {
                    jumpIndicator = false;
                    rb.velocity = new Vector2(rb.velocity.x, 0);
                    rb.AddForce(Vector2.up * jump_velocity);
                }
```

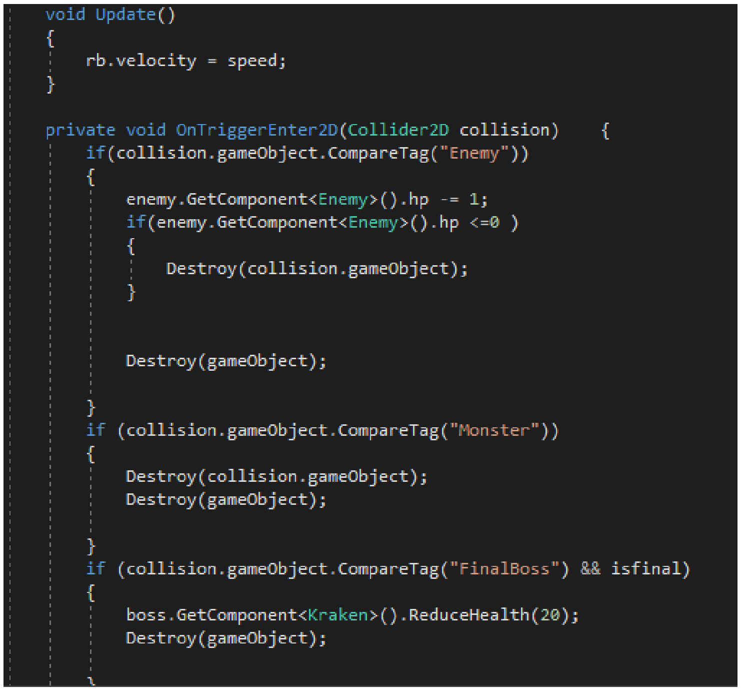
The fire function instantiates the bullet in the direction of the firepos which is a player child game object that always point in the direction of the player.



```
    void Fire()
    {
        if(!facingRight)
        {
            Instantiate(leftBullet, firepos.position, Quaternion.identity);
        }
        if(facingRight) {
            Instantiate(rightBullet, firepos.position, Quaternion.identity);
        }
    }
```

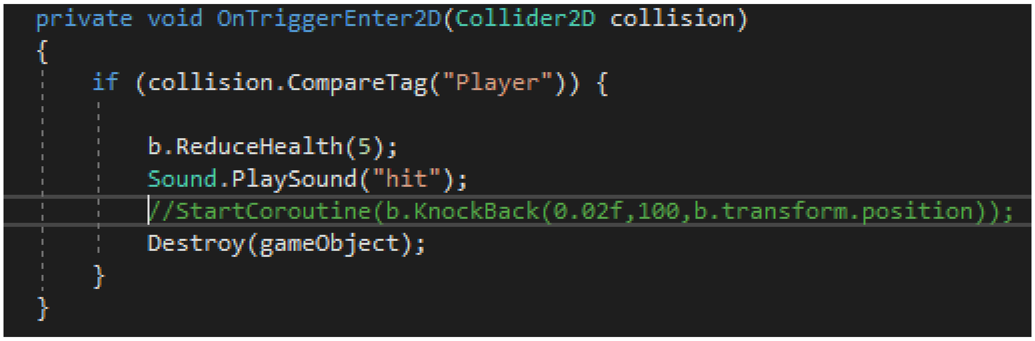
The firepos is as shown

D. Player Bullet:
The Player bullet is assigned a serialized field speed and sets off trigger on collision with different obstacles. An additional script works in the background to destroy the bullet after a delay.



```
void Update()
{
    rb.velocity = speed;
}

private void OnTriggerEnter2D(Collider2D collision)    {
    if(collision.gameObject.CompareTag("Enemy"))
    {
        enemy.GetComponent<Enemy>().hp -= 1;
        if(enemy.GetComponent<Enemy>().hp <=0 )
        {
            Destroy(collision.gameObject);
        }


        Destroy(gameObject);

    }
    if (collision.gameObject.CompareTag("Monster"))
    {
        Destroy(collision.gameObject);
        Destroy(gameObject);

    }
    if (collision.gameObject.CompareTag("FinalBoss") && isfinal)
    {
        boss.GetComponent<Kraken>().ReduceHealth(20);
        Destroy(gameObject);
```

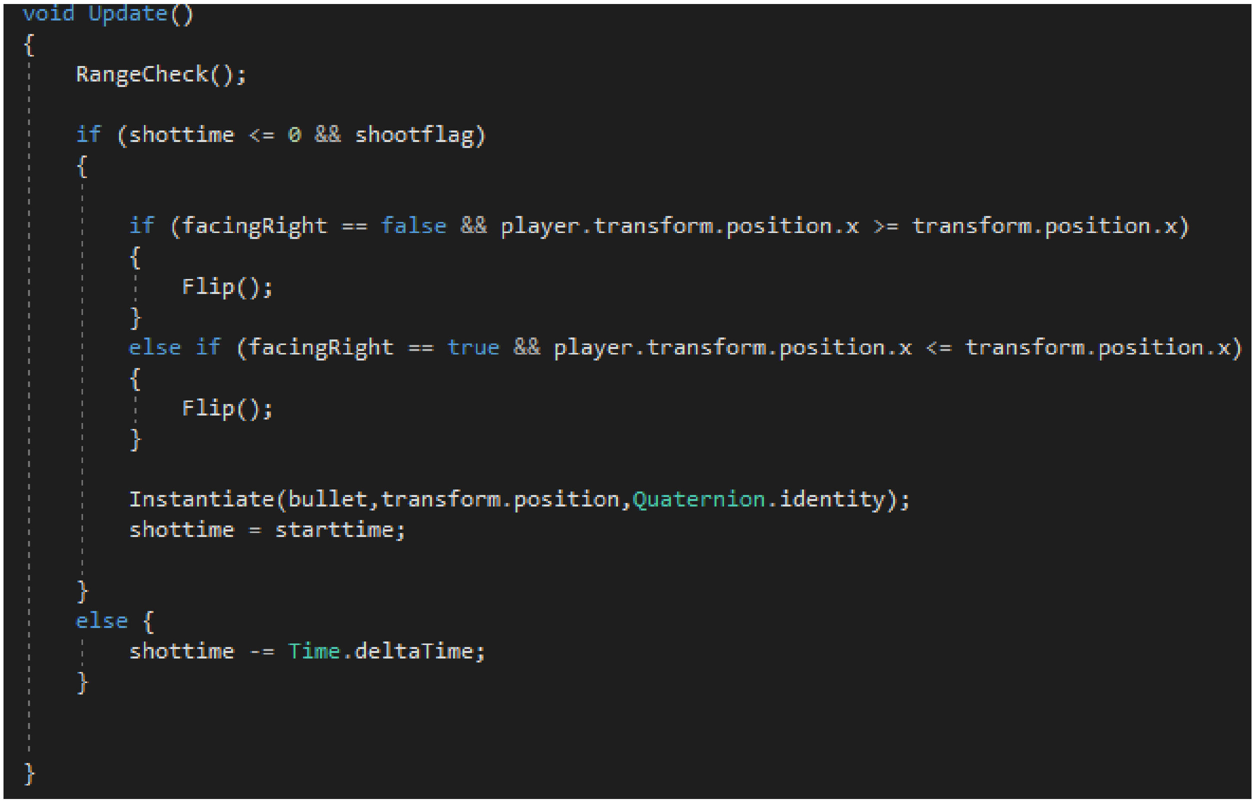
E. Monster: The player health is reduced by 5 on collision with the monster. When the player collides with the monster a function is triggered, in which player health is reduced and the monster object is destroyed.



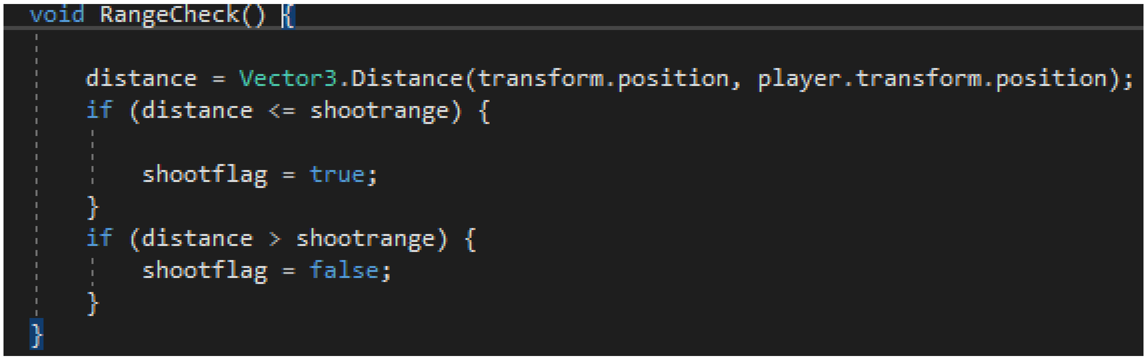
```
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.CompareTag("Player")) {

        b.ReduceHealth(5);
        Sound.PlaySound("hit");
        //StartCoroutine(b.KnockBack(0.02f,100,b.transform.position));
        Destroy(gameObject);
    }
}
```

F. Enemy: This is a shooting enemy that appears in level 2. When the player is in a set range it looks and fires at a particular interval in the direction of the player making it difficult to escape the bullet. The Enemy has a health of 2hp.

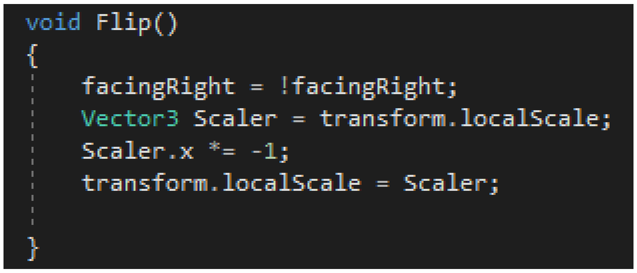
So here it checks the range first using the RandomCheck function shown ahead. If in range the The enemy flips in the direction of the player and Instantiates a Bullet.



```
void Update()
{
    RangeCheck();

    if (shottime <= 0 && shootflag)
    {

        if (facingRight == false && player.transform.position.x >= transform.position.x)
        {
            Flip();
        }
        else if (facingRight == true && player.transform.position.x <= transform.position.x)
        {
            Flip();
        }

        Instantiate(bullet,transform.position,Quaternion.identity);
        shottime = starttime;

    }
    else {
        shottime -= Time.deltaTime;
    }


}
```

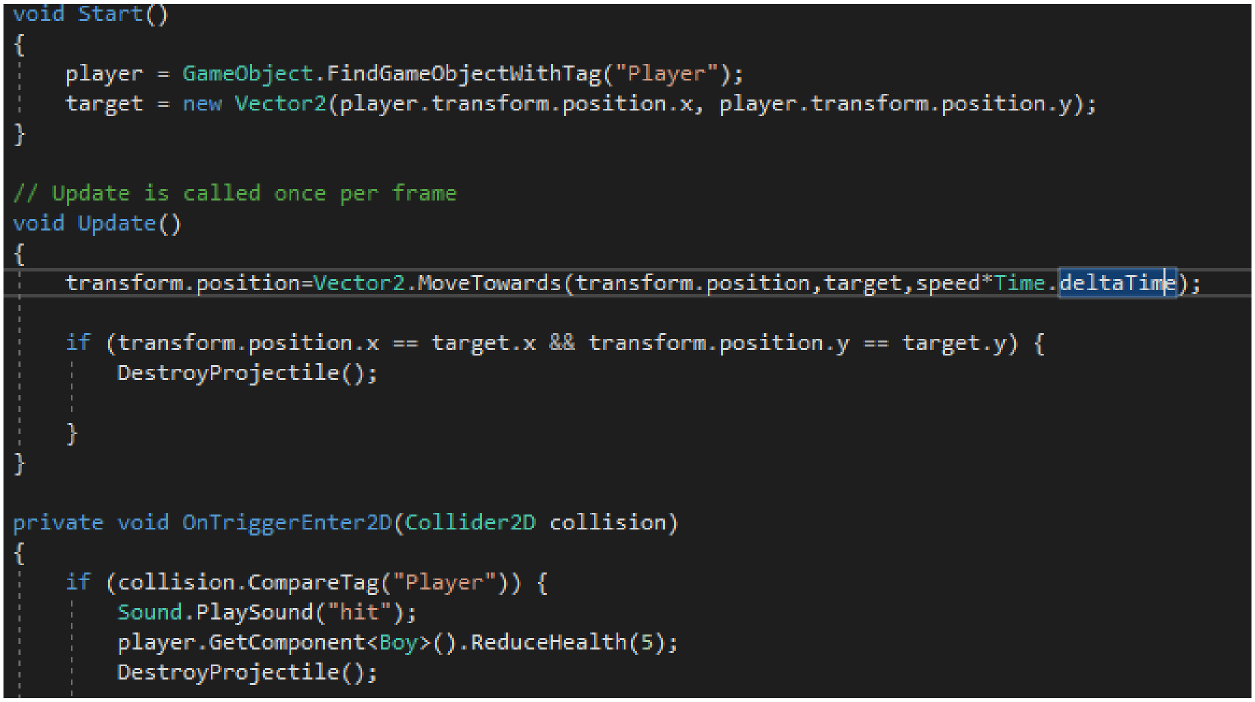


```
void RangeCheck() {

    distance = Vector3.Distance(transform.position, player.transform.position);
    if (distance <= shootrange) {

        shootflag = true;
    }
    if (distance > shootrange) {
        shootflag = false;
    }
}
```



```
void Flip()
{
    facingRight = !facingRight;
    Vector3 Scaler = transform.localScale;
    Scaler.x *= -1;
    transform.localScale = Scaler;

}
```

G. Enemy Bullet: The initiated enemy bullet is move towards in the current direction of the target ie the player. If the bullet collides with the player a function is trigger that reduces the player health by 5 and destroys the bullet game object.

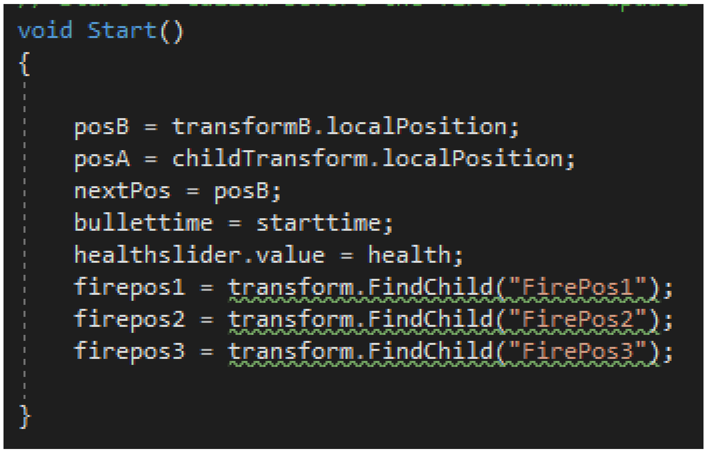


```
void Start()
{
    player = GameObject.FindGameObjectWithTag("Player");
    target = new Vector2(player.transform.position.x, player.transform.position.y);
}

// Update is called once per frame
void Update()
{
    transform.position=Vector2.MoveTowards(transform.position,target,speed*Time.deltaTime);

    if (transform.position.x == target.x && transform.position.y == target.y) {
        DestroyProjectile();

    }
}

private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.CompareTag("Player")) {
        Sound.PlaySound("hit");
        player.GetComponent<Boy>().ReduceHealth(5);
        DestroyProjectile();
```

H. Final Boss : The Final Boss has a hp of 500 and moves around vertically along with constantly firing bullets in a pattern. When the final Boss health is reduced to 200 the color of the final boss changes and also the firing pattern increasing the level of difficulty for the player .

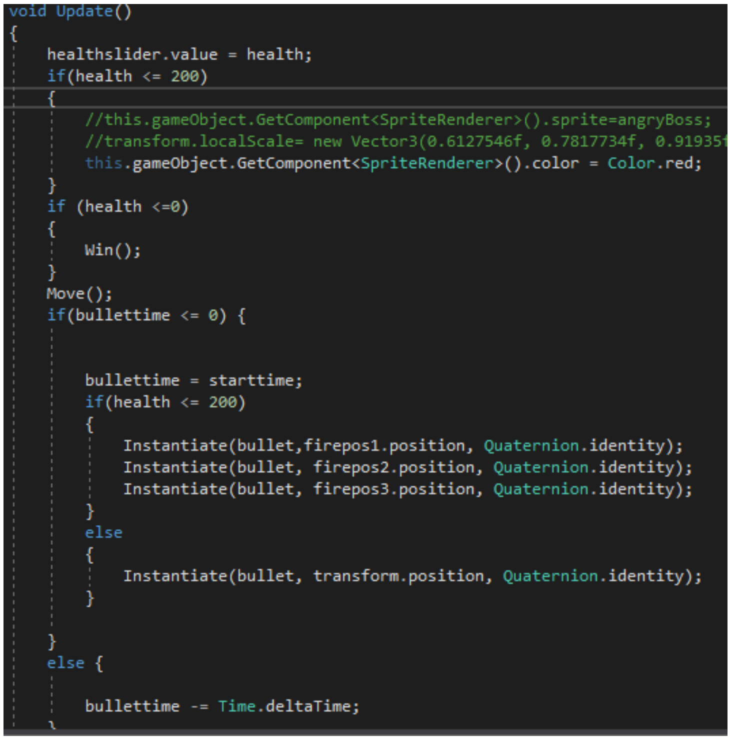
Here posB is the Boss position and posA is the child game object position towards which the boss moves.

Firepos 1,2 &3 are different shooting directions of boss bullet.



```
void Start()
{

    posB = transformB.localPosition;
    posA = childTransform.localPosition;
    nextPos = posB;
    bullettime = starttime;
    healthslider.value = health;
    firepos1 = transform.FindChild("FirePos1");
    firepos2 = transform.FindChild("FirePos2");
    firepos3 = transform.FindChild("FirePos3");

}
```

A health drops below 200 sprite color changes and more bullets are instantiated than before

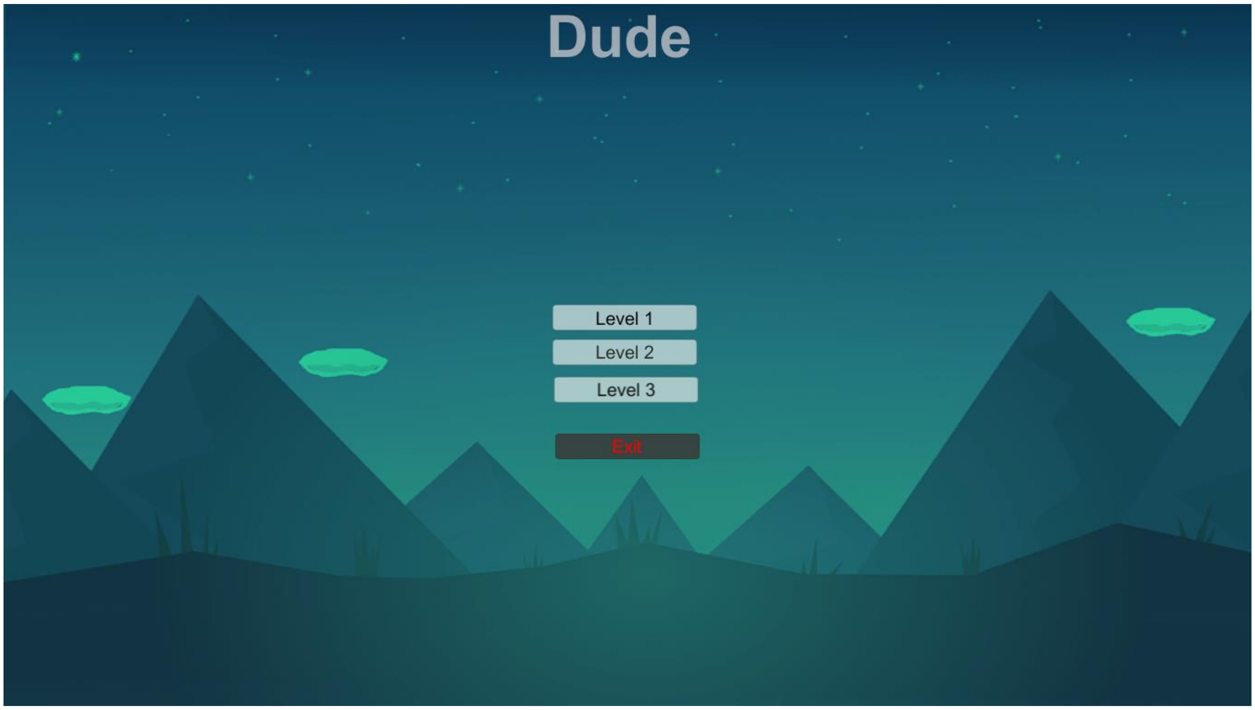


```
void Update()
{
    healthslider.value = health;
    if(health <= 200)
    {
        //this.gameObject.GetComponent<SpriteRenderer>().sprite=angryBoss;
        //transform.localScale= new Vector3(0.6127546f, 0.7817734f, 0.91935
        this.gameObject.GetComponent<SpriteRenderer>().color = Color.red;
    }
    if (health <=0)
    {
        Win();
    }
    Move();
    if(bullettime <= 0) {


        bullettime = starttime;
        if(health <= 200)
        {
            Instantiate(bullet,firepos1.position, Quaternion.identity);
            Instantiate(bullet, firepos2.position, Quaternion.identity);
            Instantiate(bullet, firepos3.position, Quaternion.identity);
        }
        else
        {
            Instantiate(bullet, transform.position, Quaternion.identity);
        }

    }
    else {

        bullettime -= Time.deltaTime;
    }
```

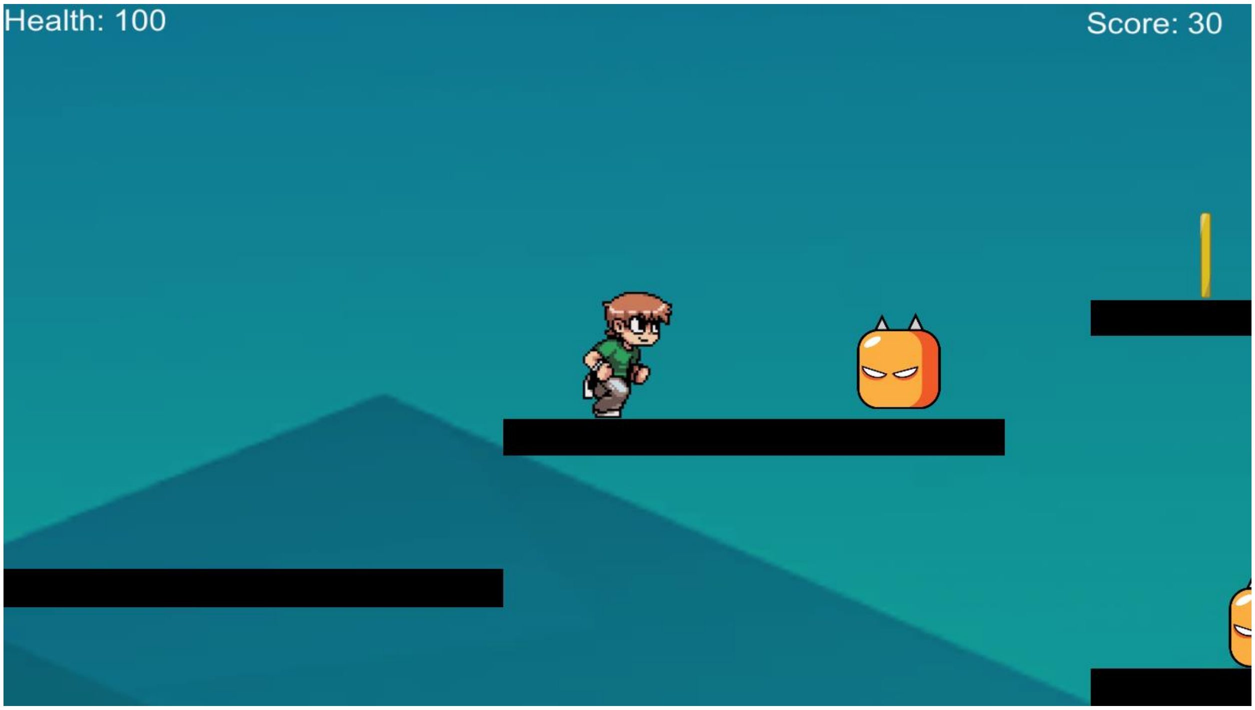
The motion of the boss is handled using the Vector3. MoveForward function used in the Move().

Output:

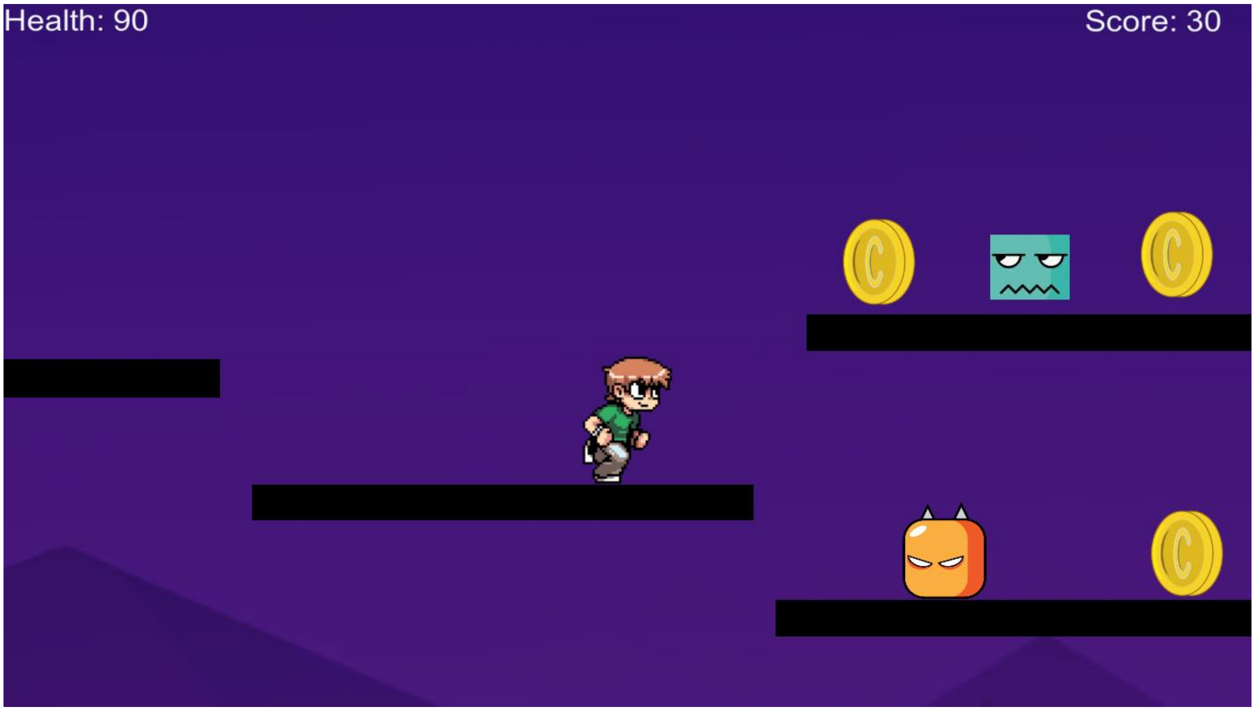
Menu



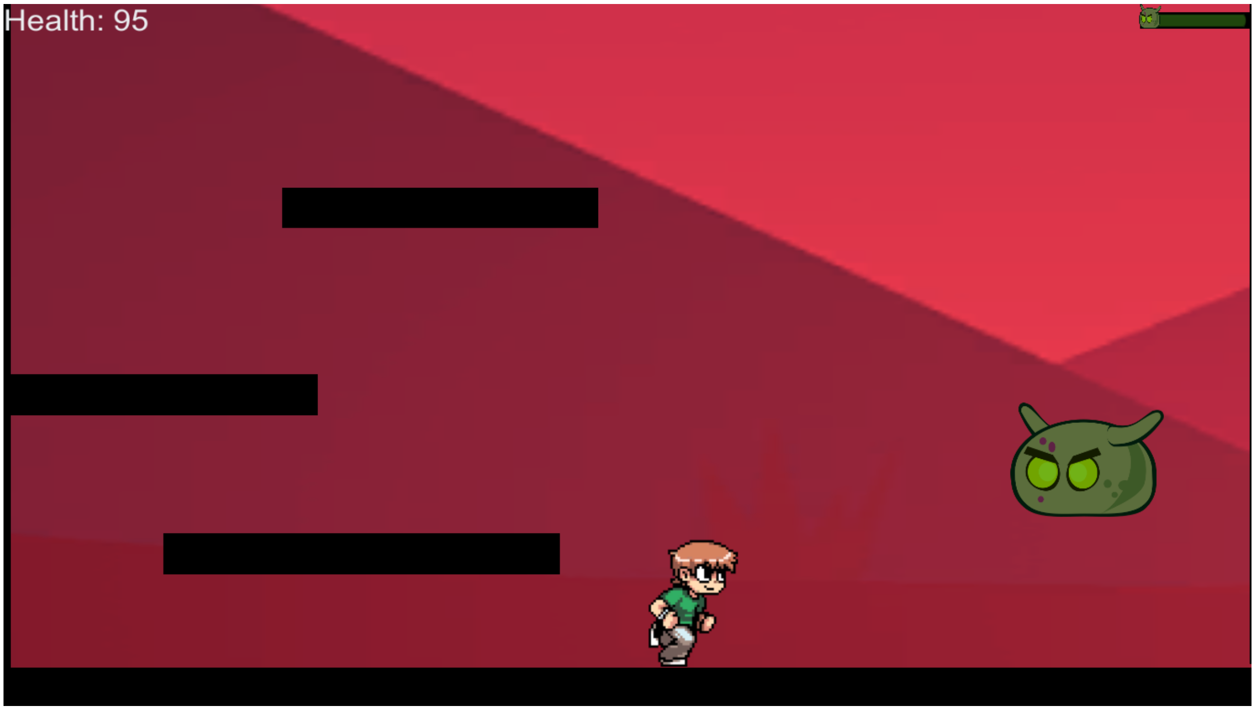
Level 1



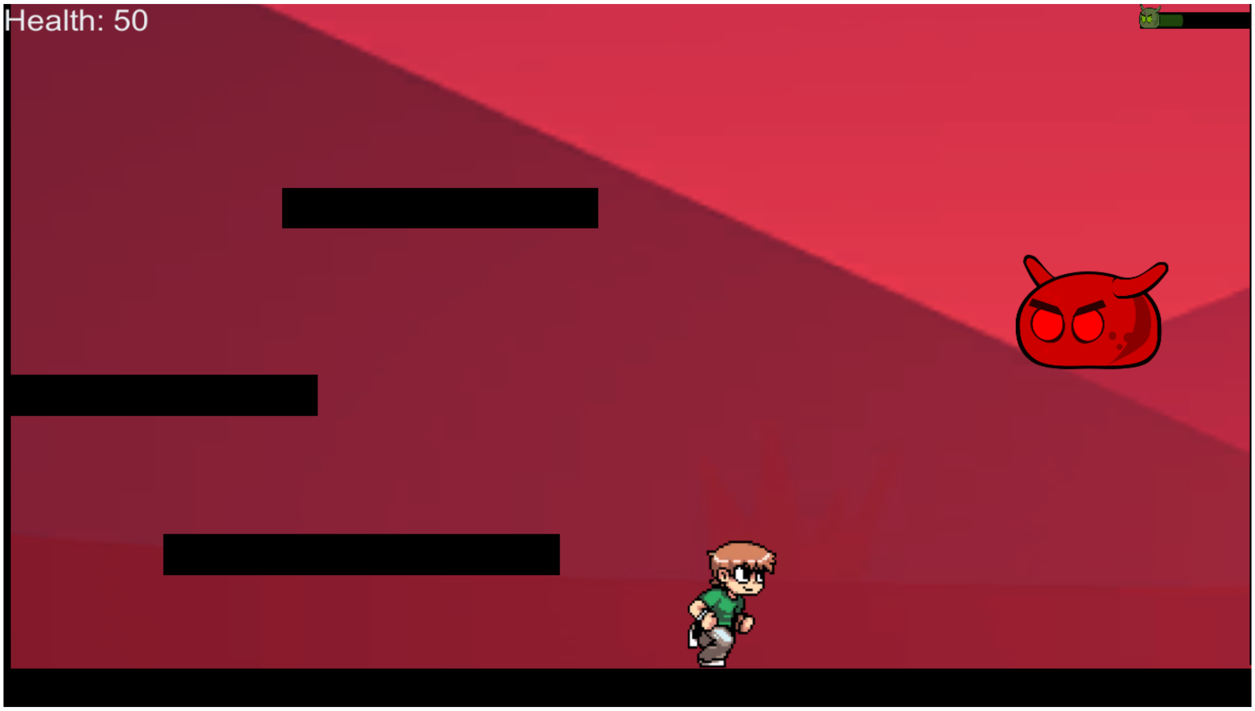
Level 2



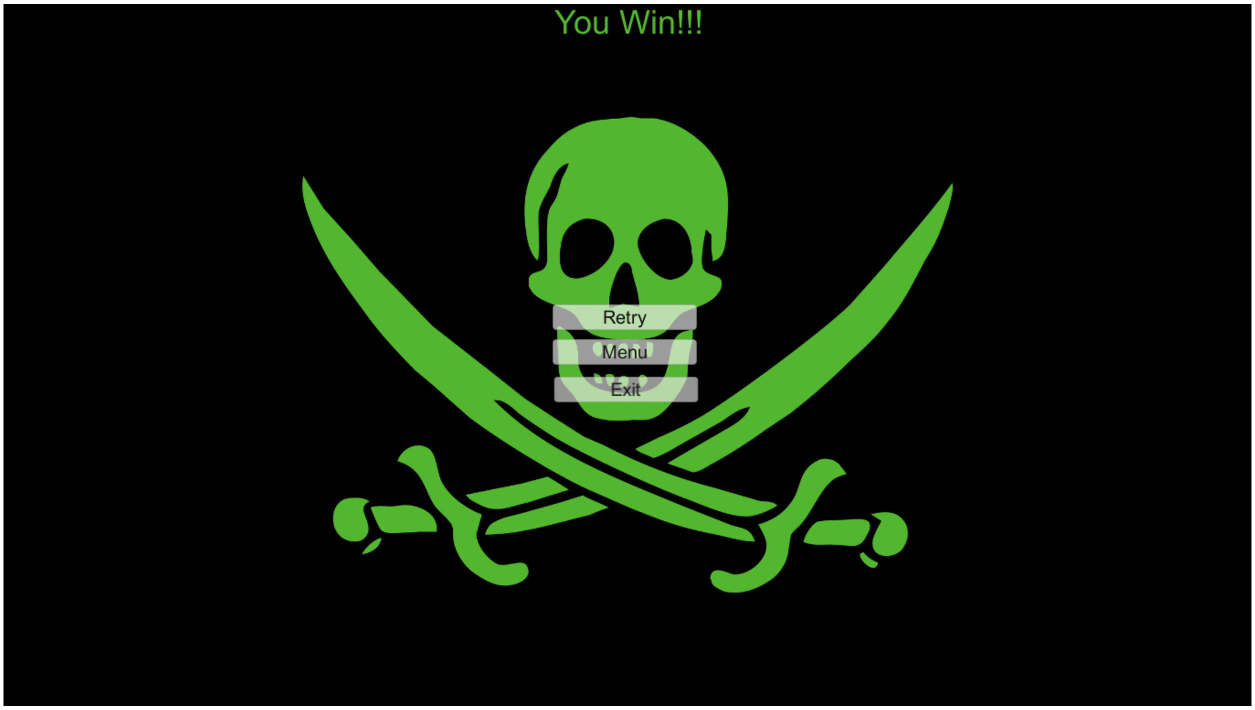
Level 3



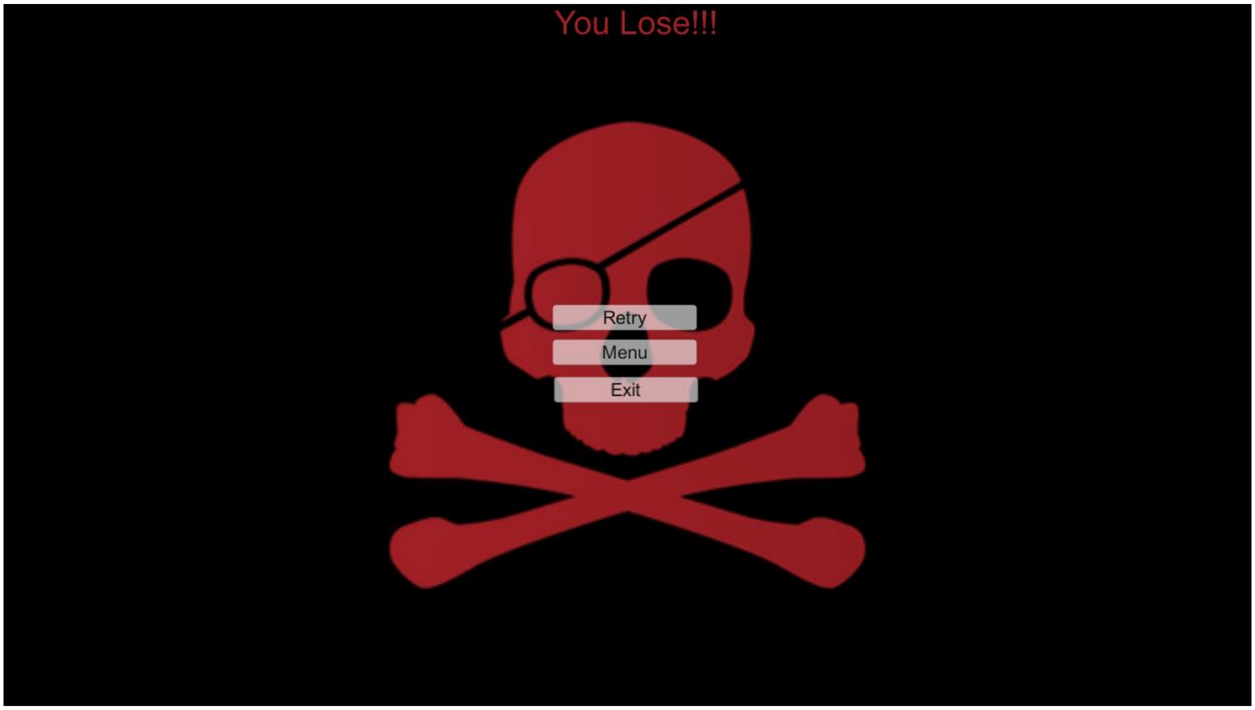
Level 3 Boss upgraded



Win



Die



You Lose!!!
Retry
Menu
Exit