Sami Nachwati

000879289

12/4/2023

# Overview of Unity Game

## Elements and Goals of the Game)

A game made in unity where a knight player is stranded on a piece of land somewhere. The goal of the game is for the knight to avoid the 2 moving sprite objects falling from the sky, representing sparks that are trying to eliminate him. The more time the player is on the land without getting hit by the sparks, a score is incremented, representing the score of his survival.

## Controls, How Elements Move, and How Parts Interact)

The game initially starts with no movement except for the background music always playing, as well as an instruction’s text, containing the instructions for the user to procced with the game. In order for the user to start the game, they will have to click the S key bind located on their keyboard. In addition, the only movements allowed by the player are moving left and moving right. To move left, the player will have to click the left arrow key bind. To move right, the player will have to click the right arrow key bind. If the player loses the game, the instruction menu will reappear, locking any other movements and the player’s movement until they press S, which in this context, is to restart the game again.

## 

## Important Code Snippets)

**Detecting collision:**

/// <summary>

/// The method used to determine the collision between the player and the spark object

/// </summary>

/// <param name="other">spark object being collided</param>

private void OnTriggerEnter2D(Collider2D other)

{

if (other.gameObject.CompareTag("Spark"))

{

sparkSound.Play();

FindObjectOfType<GameManager>().GameOver();

}

}

In the Code above, collision is checked between the player and the Spark. When the player and Spark collide, a hit sound is played, followed by the GameOver method, which does the following as seen below:

/// <summary>

/// Method used to stop the game

/// </summary>

private void Stop()

{

instructions.enabled = true;

running = false;

Time.timeScale = 0f;

player.enabled = false;

generator.enabled = false;

CancelInvoke(nameof(IncrementScore));

}

/// <summary>

/// Method used to end the game

/// </summary>

public void GameOver()

{

Stop();

}

This GameOver method called the Stop() method, which sets the enabling of the Sparks movement and the player’s movement to false, as well as canceling the incrementation of the score in the game.

**Code for the Movement of the player:**

// Update is called once per frame

private void Update()

{

if (Input.GetKey(KeyCode.RightArrow) && transform.position.x < -boundary.x)

{

transform.position += Vector3.right / 5f;

knightPos.flipX = false;

if (isMoving == false)

{

isMoving = true;

animator.SetBool("isMoving", isMoving);

}

}

if (Input.GetKey(KeyCode.LeftArrow) && transform.position.x > boundary.x+3)

{

transform.position += Vector3.left / 5f;

knightPos.flipX = true;

if (isMoving == false)

{

isMoving = true;

animator.SetBool("isMoving", isMoving);

}

}

if (!Input.GetKey(KeyCode.LeftArrow) && !Input.GetKey(KeyCode.RightArrow))

{

isMoving = false;

animator.SetBool("isMoving", isMoving);

}

}

In this code snippet above, the Right Arrow and Left Arrow keys are checked along with a check to ensure that the player is within proper boundaries. When the user clicks Left arrow or Right arrow, as long as they are within range, the player’s movement is followed through, as well as an addition of the moving animation for the player, and their position horizontally is rotated, making them face the direction where they are moving. In addition, a check is there to ensure that if the user does not select the Left arrow and Right arrow, the animation for moving is not played.

**Generation of the Spark Objects**:

/// <summary>

/// Method used to Spawn the spark objects, placing the objects at random places on the scene

/// </summary>

private void Spawn()

{

int rand = Random.Range(0, sparks.Length); // integer value used to represent the random selected value used to instantiate the sparks

GameObject s = Instantiate(sparks[rand], transform);

s.transform.position += Vector3.left \* Random.Range(-13, 13);

}

In the code above, the method Spawn() is used to ensure the generation of multiple spark objects at a time. When it is called, a random place on the scene is selected, and the instantiation of that spark object, is made. This is done precisely with the Vector3 physics, used to pinpoint the player in the desired place for generation.