**Learning Journal**

16/02/2021

Hand Following Mouse

The first thing I started to program for the game was the movement for the mouse. The game I was aiming to make was to have a rhythm game where the user would control the players hand with the mouse. The first thing I wanted to get done was the players hand following the mouse.

The task was difficult because due to lacking the knowledge. I spend some time looking into the issue and was able to find out the solution. I had to ensure firstly, that the code would be using the distance of the Main camera in the game. I then had to add code to track the position of the mouse. This was done using a vectors and getting the input of the mouse’s position. It was tricky to understand initially but I think with time it will make more sense.

20/02/2021

Beat Scroller.

Another important section for my game was the beat scroller, this was a script which would allow the notes to scroll along the screen. In order for this to work correctly I needed to know the tempo of the song I created; I was able to grab it from the program I used to make music. Once I had the tempo it needed to be divided by 60 to show that it is beats per minutes (bpm), this would need to be set in the start function.

beatTempo = beatTempo / 60f;

The next step was to add a transform position which would allow the notes to scroll to the same tempo of the music. This didn’t take me very long because I had some knowledge of using the beat scroller.

transform.position -= new Vector3(0f, beatTempo \* Time.deltaTime, 0f);

28/02/2021

Player Movement

The player movement took some time because I investigated movement for both 2D and 3D games and I have little knowledge of movement. The first type of character movement I started with was for 3D games. I had the player turn on an axis, the W and S keys were to go forward and backward. The A and D keys were used to turn, so if the player were to press the “A” key, they would spin to the left on the spot.

if (Input.GetKey(KeyCode.A))

{

transform.Rotate(Time.deltaTime \* TurnSpeed \* Vector3.down);

}

if (Input.GetKey(KeyCode.W))

{

transform.position += Time.deltaTime \* MoveSpeed \* transform.forward;

animator.SetBool("Running", true);

}

This worked for the game of that I was producing, however there were better ways to allow the player to move. I wanted to use a different movement style, the code for this was much simpler and it allowed the player to move in all directions. (up, down, left, right and diagonal)

void Update()

{

float xDirection = Input.GetAxis("Horizontal");

float zDirection = Input.GetAxis("Vertical");

Vector3 moveDirection = new Vector3(xDirection, 0.0f, zDirection);

transform.position += moveDirection;

For the 2D movement there were two different ways I went about it, there was the movement style where the player moves in all directions. This would be used in more in rpg’s.

void Update()

{

movement.x = Input.GetAxisRaw("Horizontal");

movement.y = Input.GetAxisRaw("Vertical");

rb.MovePosition(rb.position + movement \* moveSpeed);

}

The next type of movement was the more platformer 2d movement. This consists of moving left and right as well as the vertical jump. The “Mathf.Abs(rb.velocity.y) < 0.001f)” this is use to check to see if the player is touching the floor, if not then the player will not jump if they are in the air.

void Update()

{

var movement = Input.GetAxisRaw("Horizontal");

transform.position += new Vector3 (movement,0,0 \* Time.deltaTime \* MovementSpeed);

if (Input.GetButtonDown("Jump") && Mathf.Abs(rb.velocity.y) < 0.001f)

{

rb.AddForce(new Vector2(0, JumpForce) ,ForceMode2D.Impulse);

}

}

01/03/2021

Random Instantiation

The random instantiation was for a package I was creating. The random instantiation uses an array to store the items which would be randomly instantiated. The script randomly selects an object from the array and instantiates it in the spawn position which would need to be specified.

public void SpawnRandom ()

{

randomInt = Random.Range(0, Spawnees.Length);

Instantiate(Spawnees[randomInt], spawnPos.position, spawnPos.rotation);

}

02/20/2021

Health Package

The package that I made for health was something I had a few issues with. The first few things I had to do was to set up the integers which would hold the max health and the current health. The current health would need to be set as the max health in the start function so that the player would start with the max health.

The next step was to have functions which allowed the player to take damage and be healed. I had to ensure there were things to stop the players health from exceeding the maximum and for the health to specify that the player has died once their health goes below zero.

public void TakeDamage(int amount)

{

CurrentHealth -= amount;

HealthText.text = "Health: " + CurrentHealth;

if (CurrentHealth <= 0)

{

print("Dead");

}

}

public void Heal(int amount)

{

CurrentHealth += amount;

HealthText.text = "Health: " + CurrentHealth;

if (CurrentHealth => MaxHealth)

{

CurrentHealth = MaxHealth;

print("Full Health");

}

}

16/04/2021

Homing Missile

The Homing missile was another package I created for the module. There are two different ways I went about finding the target that the missile would aim at. The first was by making it a public variable and the second was to find the game object which had the tag of player.

public Transform target;

target = GameObject.FindGameObjectWithTag("Player").transform;

The missile starts off by finding the target, then it looks for the direction that it needs to go towards to be facing the target. The variables within the code are the speed, which controls how fast the bullet goes and the rotate speed, this is for how fast and tight the missile will turn.

void Update()

{

Vector2 direction = (Vector2) target.position - rb.position;

direction.Normalize();

float rotateAmount = Vector3.Cross(direction, transform.up).z;

rb.angularVelocity = - rotateAmount \* rotateSpeed;

rb.velocity = transform.up \* speed;

}

I linked this code with the health script so that the player would take damage when it collides with them.