**Double Jump Tutorial**

**15/10/19**

I started by creating a few variables, "**public float speed;**" to be able to control the speed of the player in the level. I then created "**public float jumpForce;**" to be able to control the height the player can jump and edit it later. Lastly, I created a "**private float moveInput;**" to check whether the keys are being pressed.Next, I wrote:

**private Rigidbody2D rb;**

**void Start(){**

**rb = GetComponent<Rigidbody2D>();**

This was done so that the scripts can be used with the player model. The next line of code I placed was:

**void FixedUpdate(){**

**moveInput = Input.GetAxis("Horizontal");**

**rb.velocity = new Vector2(moveInput \* speed, rb.velocity.y);**

This allows the player to move left and right. This also prevents the player moving on the y axis making it completely 2D.

After, I placed the script onto the player and gave it a speed value. I gave the player model a Rigidbody 2D and froze the rotation and gave it a default Box Collider 2D, as well as the platforms on the level, so the player wouldn't fall through them.

Once I had the basic player movement complete I moved onto creating the jumping script. I started by writing the variable "**private bool isGrounded;**". This line checks whether or not the player is currently on the ground. I then wrote the variables “**public Transform feetPos;**”, “**public float checkRadius;**” and “**public LayerMask whatIsGround;**”. Next, I created a new void update:

**voidUpdate(){**

**isGrounded = Physics.OverlapCircle(feetPos.position, checkRadius, whatIsGround);**

**if(isGrounded == true && Input.GetKeyDown(KeyCode.Space)){**

**rb.velocity = Vector2.up \* jumpForce;**

These lines of code check if the player is able to jump in their current position. I then created a **feetPos object** and parented it to my player, and positioned it at the players feet. Next, I moved the object to the Feet Pos variable created on the player.