1. You’ll first need a simple sprite in Unity. Make sure it has a unique identifier so that you can see if it flips; I simply put a triangle on the right-hand side of a square and dragged the triangle into the square in the hierarchy so that they were attached.

Set the scale of the sprite to 2 by 2, so that the coding is easier later.

1. Create a new C# script. You can call it whatever you like, but I called it Player. Once you’ve named it, open it.
2. Under void Update, type

*transform.Translate(Input.GetAxis("Horizontal")\* 15f \* Time.deltaTime, 0f, 0f);*

This is what will allow the character to move, with the Y and Z values both set to 0.

Adding the *\* 15f \* Time.deltaTime* sets the character movement to a reasonable speed that aligns with the speed of the computer being used to avoid delay.

1. This next block of code is what will flip the character sprite.

*Vector3 characterScale = transform.localScale;*

*if (Input.GetAxis(“Horizontal”) < 0) {*

*characterScale.x = -2;*

*}*

*if (Input.GetAxis(“Horizontal”) > 0) {*

*characterScale.x = 2;*

*}*

*transform.localScale = characterScale;*

The program will see if the character is moving to the left or right and will re-scale the sprite to match the direction.

1. Finally, attach the script to the player sprite. (Drag and drop the script onto the player sprite in the hierarchy.