Tutorial; Player movement and camera controls

Setup

To begin with, open 3D mode. Then, create a plane from the game object selection bar and set the coordinates to 0,0,0 and set the sizing to 5. Create a material for the terrain so you can see the plane more clearly as well as the player's movement across the terrain. Add a capsule game object and set the co-ordinates to 0, 2, 0 and renamed it "Player". Then, add an empty script, called it player and added it to the game object player in the hierarchy.

Coding player movement

The first part of code was created for the player movement and speed. For this, set up three public floats in the script under MonoBehaviour;

```
public class Player : MonoBehaviour
{
    public float movementSpeed = 30;
    public float turningSpeed = 60;
    public float damage = 0;
```

The purpose of creating a public float is so we can adjust the player speed accordingly in the hierarchy in game. On void update, set the movement speed accessing the information from the float using this code;

```
  // Update is called once per frame
    void Update()
    {
        float horizontal = Input.GetAxis("Horizontal") * movementSpeed * Time.del-
taTime;
        float vertical = Input.GetAxis("Vertical") * movementSpeed * Time.deltaTime;

        transform.position += new Vector3(horizontal, 0, vertical);

}
```

This gives the player controls where the horizontal axis (the left-or-right keys) turn the player around, while the vertical axis (up-or-down keys) move the player forward and backward.

Coding player camera

Next, attach the camera to the player by creating a script called LookAtCamera. Then, attach to the main camera in the hierarchy and use the code in the script;

```
public class LookAtCamera : MonoBehaviour
{
    public GameObject target;
    // Start is called before the first frame update
    void Start()
    {
```

```
    }

    // Update is called once per frame
    void LateUpdate()
    {
        transform.LookAt(target.transform);
    }
}
```

Next, create a script which has a fixed position on the player as it moves around the terrain. Call this script “Dungeon Camera” in reference to the Dungeon crawler game genre that uses this method of third person controlling;

```
public class DungeonCamera : MonoBehaviour {

   public GameObject target;

   Vector3 offset;


   void Start() {

      offset = transform.position - target.transform.position;

   }

void LateUpdate() {

   Vector3 desiredPosition = target.transform.position + offset;

   tranform.position = desiredPosition;

}
```

Once again drag the script into the MainCamera in hierarchy and then in the public float, add the player game object from the hierarchy into the public float, so the player becomes attached to the script.