


Lesson 2

Moving the Player with the Camera

▼  SampleScene* ☰

 Directional Light

 Cube

▼  Capsule

 Main Camera

▼  Transform ? ☰ ⚙️

Position X Y Z

Rotation X Y Z

Scale X Y Z

Player Controller

```
public class PlayerController: MonoBehaviour
{
    float yaw;

    void Update()
    {
        yaw += Input.GetAxis("Mouse X") * 8;
        transform.eulerAngles = new Vector3(0, yaw, 0);

        Vector3 velocity = new Vector3(0, 0, 0);
        velocity.x = Input.GetAxis("Horizontal") * 10 * Time.deltaTime; //Switch .x and .z to look like this
        velocity.z = Input.GetAxis("Vertical") * 10 * Time.deltaTime;

        if (Input.GetKeyDown(KeyCode.Space))
            rb.AddForce(Vector3.up * 5, ForceMode.Impulse);

        transform.Translate(velocity);
    }
}
```

Camera Controller

```
public class cameraController : MonoBehaviour
{
    public GameObject player;

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        //Dont Do Anything!
    }
}
```

Moving Platforms


```
public class MovingPlatform : MonoBehaviour
{
    public GameObject waypoints[];
    int index = 0;

    void Start()
    {
        index = 0;
        transform.position = waypoints[0].transform.position;
    }

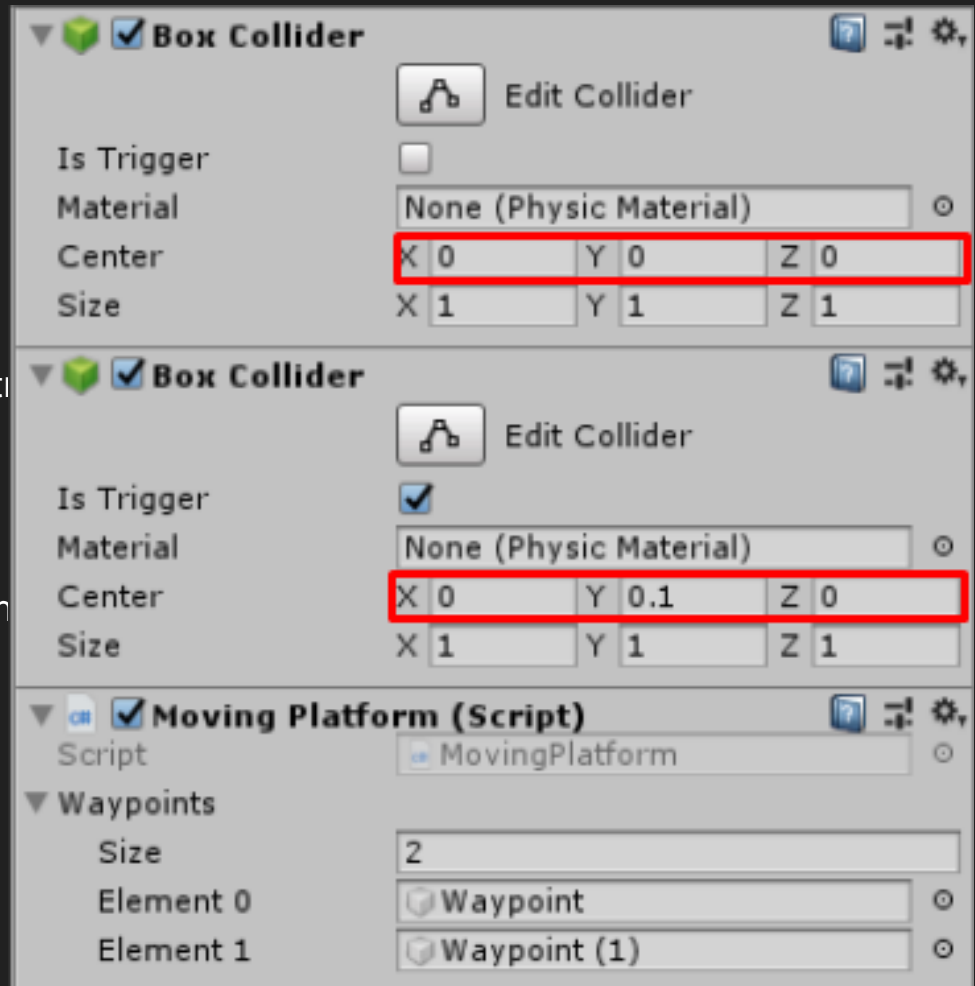
    void Update()
    {
        if (Vector3.Distance(transform.position, waypoints[index].transform.position) < .1)
        {
            index++;
            if(index > waypoints.Length - 1)
            {
                System.Array.Reverse(waypoints);
                index = 1;
            }
        }
        else
            transform.position = Vector3.MoveTowards(transform.position,
waypoints[index].transform.position, 0.05f);
    }
}
```

```
public class MovingPlatform : MonoBehaviour
```

```
{  
    void OnTriggerEnter(Collider other)  
    {  
        other.transform.parent = tr    }  
}
```

```
    void OnTriggerExit(Collider other)  
    {  
        other.transform.parent = n    }  
}
```

```
}
```



Stop the Player from
Jumping Infinitely

Check if the Player is Touching a Floor

```
public class PlayerController: MonoBehaviour
{
    bool isGrounded()
    {
        //Shoot a ray down from center of player to see if we are touching the ground
        //Length of ray is height from center of player to bottom of player (0.5 in this case) + 0.1 to reach
the ground
        return Physics.Raycast(transform.position, Vector3.down, 0.6f);
    }

    void Update()
    {
        //...

        if (Input.GetKeyDown(KeyCode.Space) && isGrounded())
            rb.AddForce(Vector3.up * 5, ForceMode.Impulse);

        //...
    }
}
```

BONUS: Invisible Cursor

```
public class playerController: MonoBehaviour
{

    // Start is called before the first frame update
    void Start()
    {
        rb = GetComponent<Rigidbody>();
        Cursor.visible = false; //Press ESC to make visible
        Cursor.lockState = CursorLockMode.Locked;
    }

}
```