Lesson 1
General Setup

Create a platform for the character to move over and color it

Create a basic character

Write a script to control the character

Write a script to control the camera
Creating a Platform
Creating a Material
Creating a Player
Player Movement
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Camera_Controller : MonoBehaviour
{
    void Start()
    {
        // Start is called before the first frame update
    }

    void Update()
    {
        // Update is called once per frame
    }
}
void Update()
{
    Vector3 velocity = new Vector3(0, 0, 0);
    this.transform.position = this.transform.position + velocity;
}
Movement with Input

```csharp
void Update()
{
    Vector3 velocity = new Vector3(0, 0, 0);
    if (Input.GetKeyDown(KeyCode.D))
    {
        this.transform.position = this.transform.position + velocity;
    }
}
```
void Update()
{
    Vector3 velocity = new Vector3(0, 0, 0);
    if (Input.GetKeyDown(KeyCode.D))
    {
        this.transform.position = this.transform.position + velocity;
    }
    else if (Input.GetKeyDown(KeyCode.A))
    {
        this.transform.position = this.transform.position - velocity;
    }
}
Movement with Axis Input

```csharp
void Update()
{
    Vector3 velocity = new Vector3(0, 0, 0);
    velocity.z = Input.GetAxis("Horizontal");

    this.transform.position = this.transform.position + velocity;
}
```
Make a Camera Follow the Player
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Camera_Controller : MonoBehaviour
{
    public GameObject player;

    // Start is called before the first frame update
    void Start()
    {
    }
}
public GameObject player;

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

// Update is called once per frame
void Update()
{
    this.transform.position = player.transform.position + new Vector3(0, 0, -10);
}
Player Movement with a Rigidbody
Rigidbody

1. Physics
2. FixedUpdate
Movement with a Rigidbody

rigidbody rb;  // A reference to the Rigidbody attached to this object

void Start()
{
    Rb = this.GetComponent<RigidBody>();
}

void Update()
{
    Vector3 velocity = new Vector3(0, 0, 0);
    velocity.z = Input.GetAxis("Horizontal");

    rb.AddForce(velocity);
}
Movement with a Rigidbody

```csharp
rigidbody rb;       // A reference to the Rigidbody attached to this object
float moveSpeed = 5f;

void Start()
{
    Rb = this.GetComponent<Rigidbody>();
}

void Update()
{
    Vector3 velocity = new Vector3(0, 0, 0);
    Velocity.z = Input.GetAxis("Horizontal") * moveSpeed;

    rb.AddForce(velocity);
}
```
Jumping with a Rigidbody

```csharp
rigidbody rb; // A reference to the Rigidbody attached to this object
float moveSpeed = 5f;
float jumpForce = 5f;

void Start()
{
    rb = this.GetComponent<Rigidbody>();
}

void Update()
{
    Vector3 velocity = new Vector3(0, 0, 0);
    Velocity.z = Input.GetAxis("Horizontal") * moveSpeed;

    rb.AddForce(velocity);

    if(Input.GetKeyDown(KeyCode.Space))
        rb.AddForce(Vector3.up * jumpForce);
}
```
Note on Different Types of Forces

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Force</strong></td>
<td>Add a continuous force to the rigidbody, using its mass.</td>
</tr>
<tr>
<td><strong>Acceleration</strong></td>
<td>Add a continuous acceleration to the rigidbody, ignoring its mass.</td>
</tr>
<tr>
<td><strong>Impulse</strong></td>
<td>Add an instant force impulse to the rigidbody, using its mass.</td>
</tr>
<tr>
<td><strong>VelocityChange</strong></td>
<td>Add an instant velocity change to the rigidbody, ignoring its mass.</td>
</tr>
</tbody>
</table>
using System.Collections.Generic;
using UnityEngine;

public class Player_Controller : MonoBehaviour
{
    Rigidbody rb;

    public float jumpStrength = 5;
    public float moveSpeed = 5;

    // Start is called before the first frame update
    void Start()
    {
        rb = GetComponent<Rigidbody>();
    }

    // Update is called once per frame
    void Update()
    {
        rb.AddForce(Vector3.right * Input.GetAxisRaw("Horizontal") * moveSpeed, ForceMode.Force);

        if (Input.GetKeyDown(KeyCode.Space))
        {
            rb.AddForce(Vector3.up * jumpStrength, ForceMode.VelocityChange);
        }
    }
}