***Basic player Movement Tutorial in unity***

***Introduction:***

The purpose of this tutorial is to demonstrate how to move a game object based on player input. This will involve using basic coding skills to implement the functionality. The tutorial utilizes Unity as the game development platform and C# as the programming language, alongside Unity's built-in engine libraries.

***Setup:***

First you will need to create a new unity project or open an existing one.

A screenshot of a computer

Description automatically generated

***Create object for movement:***

After creating a new Unity project, navigate to the GameObject menu, then go to 3D Object, and select any type of 3D game object you want. For this example, a simple cube has been created.

A screenshot of a computer

Description automatically generated

***Creating a C# Script***

To enable your object to move, you need to create a C# script. This script is the core component of the tutorial.

1.Navigate to Assets:

Go to the Unity editor's Assets menu, then select Create > C# Script.

2.Locate the Script:

After creation, the script will appear in the Assets folder.

3.Name the Script:

Rename the script to match the name of your game object. For example, if your object is called "PlayerMovement," rename the script to PlayerMovement.

Important: Ensure the script name matches the name of the game object exactly, as consistency helps with organization and readability.

4.Attach the Script to the Game Object:

Once named, drag the script onto the game object in the Hierarchy panel or use the Inspector to attach it.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

***Script explanation***

Below is an explanation about this script that makes a player object move in unity. The player move’s based on input from the keyboard (arrow keys or WASD)

The number 1. and number 2. lines of code are like toolboxes Unity can use, though in this case, they are not doing much. They are included by default when creating a script in Unity. Number 3 line is essential which is the “Using UnityEngine”. It tells Unity that we are using its tools to create things like movement, physics, and visuals. The number 5 line defines a class called more specifically “PlayerMovement” Think of a class as a blueprint that controls the behaviour of your player. The “Monobehaviour” part means this script can be attached to a game object in Unity to make it do stuff.

Next the number 8 line of code the “SerializeField” means the “speed” value can be adjusted in the Unity editor without making it public. “private Float speed=20;” sets the player's movement speed to 20. If you change the number, the player will move faster or slower. The number 9 line of code “private Vector3 movement= new Vector3(10, 20);” sets up a Vector3, which is just a way to store 3 numbers (X, Y, and Z). In this line, it starts as (10, 20, 0), but this “movement” is not used in the “Update” method, so it does not affect anything in the game.

Next line number 12 “Void Update()” the “Update” function runs every frame of the game. If your game runs at 60 frames per second, this code will execute 60 times every second. This is where we handle real-time input and movement. Within the update function the Float moveX = Input.GetAxis(“Horizontal”); on number 14 checks if the player is pressing the left or right arrow key (or A/D). It also returns a value between -1(moving left) and 1 (moving right). Line number 15 Float movez=Input.GetAxis(“Vertical”); is like line number 14 but it checks for up/down movement (W/S keys). Line number 17 Vector3 movement= new Vector3(moveX,0, movez) \* speed \*Time.deltaTime;. This creates a new Vector3 using the input values moveX for left/right and movez for forward/backward. The \*speed makes the player move faster or slower based on the speed value. The \*Time.deltaTime ensures movement is smooth and consistent, even if the game’s frame rate changes. Lastly, Line number 18 the transform.Translate(movement);. This moves the player in the direction specified by the movement vector.

In conclusion this script allows a player to move by reading input from the keyboard. The unity’s update method constantly checks for input and moves the player object. The movement speed and direction depend on the speed value and the Horizontal and Vertical input axes.

A computer screen shot of a black screen

Description automatically generated

***Demonstrations of the Movement being used***

Once the script is created, you can test it by clicking the Play button in Unity. However, before doing this, follow these steps:

*Attach the Script to the Game Object:*

Select the game object (e.g., the cube you created).

Drag the script from the Assets folder into the Inspector panel.

You will see the script appear in the Inspector under the game object's components.

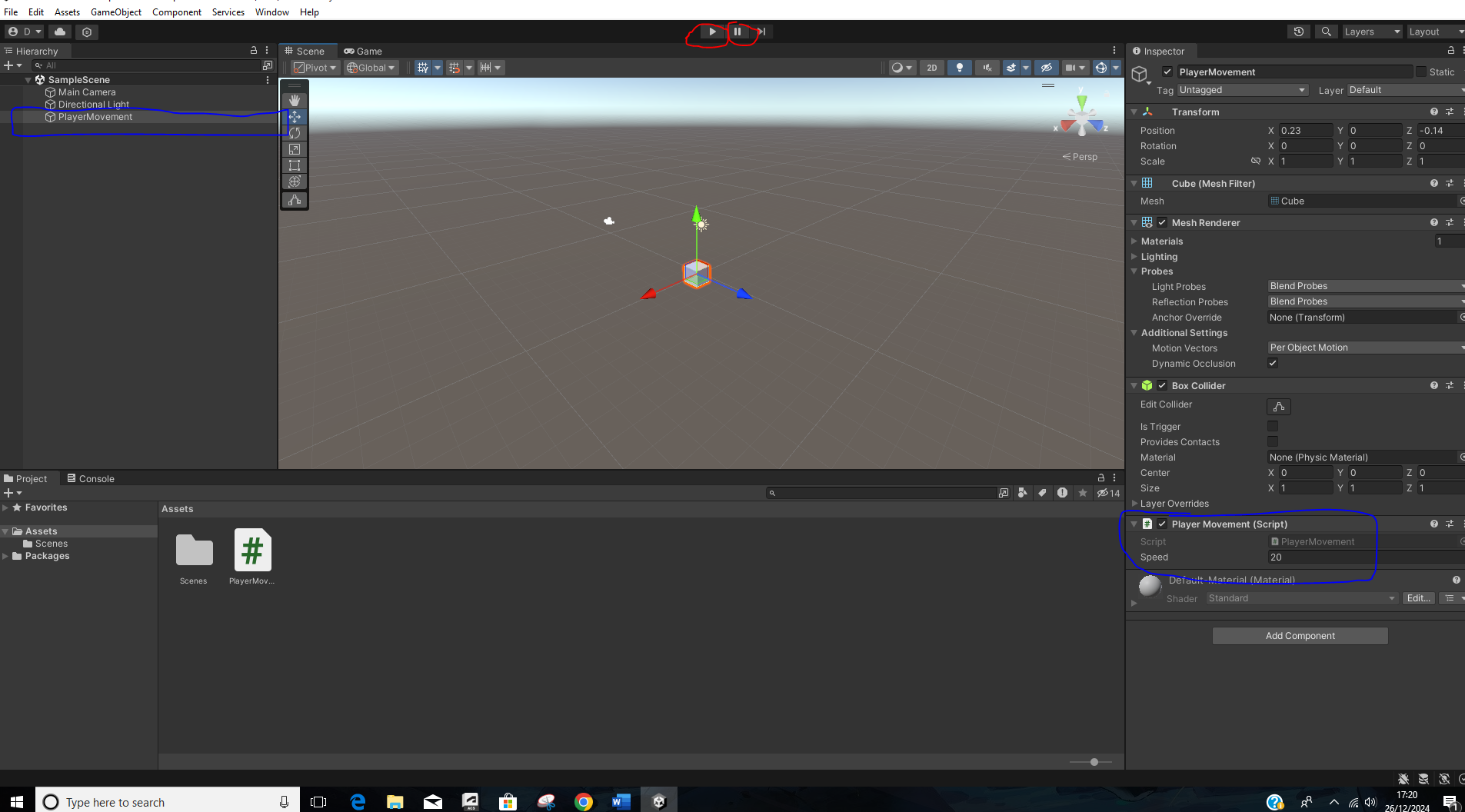
*Run the Game:*

Click the Play button (circled in red) to start the game and test the movement script.

Use the assigned keys to move the object, and it should respond as expected.

*Pause the Game (Optional):*

If needed, you can pause the game using the Pause button, also circled in red.



A screenshot of a computer

Description automatically generated

A computer screen shot of a white cube

Description automatically generated