**Introduction to Unity**

**Tutorial - Creating the Tilting Board Prototype**

Here are the steps to create the Tilting Board basic prototype.

**Creating the base**

1. Start a new 3D project called Tilting Board.
2. Create a simple cube (GameObject🡪3D Object🡪Cube) and resize it so that its scale in the x and z axes is 100 units. Scale the y axis to a value of 1.
3. Reposition the cube to 0, -0.5, 0 if it is not already (this puts the top surface at y=0).
4. Make a new Physic Material (Asset🡪Create🡪Physic Material), rename it Metal and set the static and dynamic friction to 0.
5. Call the cube you have just made ‘Board Base’ (without the quotes).
6. Attach a Rigidbody to it (Component🡪Physics🡪Rigidbody).
7. Make sure Use Gravity is not ticked.
8. Turn it into a Kinematic Rigidbody by ticking the 'Is Kinematic' checkbox.
9. Change its collision detection to Continuous Dynamic.

**Make the scene camera**

1. Add a camera object (GameObject🡪Camera) to the scene, call it ‘Main Camera’ and set its tag to MainCamera.
2. Move the camera so that it is positioned at 0, 100, 0.
3. Rotate the camera to 90, 270, 0.
4. Change the far clipping plane to at least 1000 and the field of view to at least 55.

**Add a directional light source**

1. Add a directional light to the scene (GameObject🡪Light🡪Directional Light).
2. Position the light at 0, 100, 0.
3. Rotate the light to 90, 270, 0.

**Creating the walls**

1. Create a cube and size it appropriately to the base e.g. 100, 5, 5.
2. Position the wall where you want it on the base, noting that if you want it to rest on top of the base then its y-position should be set to 2.5.
3. Feel free to name the wall something useful.
4. Now parent the wall to the base by dragging it over the Board Base object in the hierarchy and letting go (notice what happens to the position of the wall when you parent it).
5. Make three other outer walls like this and continue in this fashion to make any other walls you need (note: you can use Edit🡪Duplicate to copy and paste an object or the shortcut key combination CTRL+D).

**Creating the ball**

1. Create a sphere (GameObject🡪3D Object🡪Sphere) whose scale is 5.
2. Change its material in its collider to Metal (you can change this if you wish).
3. Rename the sphere ‘Ball’.
4. Attach a Rigidbody to it and make sure its 'Use Gravity' checkbox is ticked and 'Is Kinematic' is unticked.
5. Change its Collision Detection to ‘Continuous Dynamic’.
6. Place the sphere so that it is resting on the Board Base object (e.g. 0, 2.5, 0).

**Changing the global Physics Manager settings**

1. Open the Physics Manager (Edit🡪Project Settings🡪Physics).
2. Change the gravity to -98.1 (you can change this later but why is the original value -9.81?).

**Changing the Input Manager settings**

1. Open the Input Manager (Edit🡪Project Settings🡪Input) and have a look at the settings for the Horizontal and Vertical Axes.
2. If you want to change the default keys/buttons that are currently bound to an axis then do so here (they are per-project settings).

**Creating the tilt script**

1. Create a custom script folder and make a new C# file called Tilt (this is case-sensitive).
2. Double click on it to open the Tilt script file to open it in the Visual Studio editor and change the contents to be the following:

using UnityEngine;

using System.Collections;

public class Tilt : MonoBehaviour

{

public float smooth = 1.5f;

public float maxTiltAngle = 30.0f;

// Update is called once per frame

void Update ()

{

// check if the user is pressing left or right, up or down on the keyboard

// by default Unity assigns the "a" and "d" key & the left and right arrow keys to the

// horizontal axis and the "w" and "s" key & the up and down arrows keys to the vertical axis.

float tiltAroundX = Input.GetAxis("Horizontal") \* maxTiltAngle;

float tiltAroundZ = Input.GetAxis("Vertical") \* maxTiltAngle;

Quaternion target = Quaternion.Euler(tiltAroundX, 0, tiltAroundZ);

transform.rotation = Quaternion.Slerp(transform.rotation, target, Time.deltaTime \* smooth);

}

}

1. Check that both the class’s name is Tilt and the filename is Tilt.cs (these must match in order for the file to compile).
2. Add the Tilt script to the Board Base object by dragging and dropping it from the Project panel over the Board Base object in the Hierarchy.

**You can now try to run the basic Tilting Board prototype by pressing the play button.**

**Until we have looked at how to add custom behaviours via script, we can't do much more than this.**

**Make sure you save the scene and the project and take it with you!**

**Go through the steps to rebuild this basic prototype at home so you understand how the scene is constructed.**