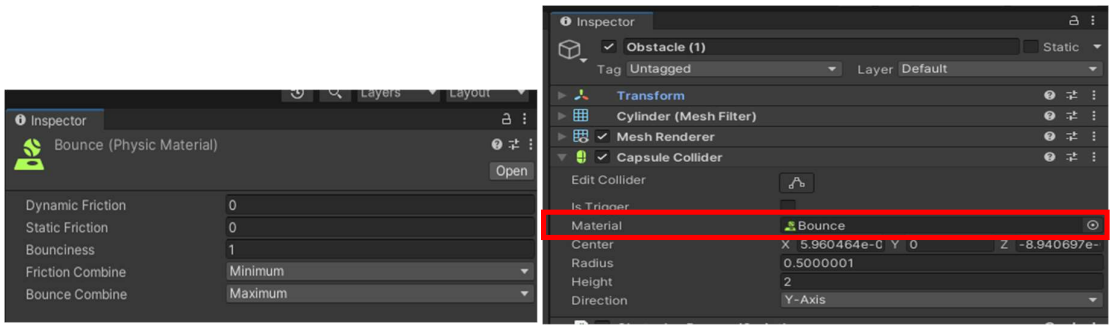
CM3045 3D Graphics & Animation

Mid-Terms Coursework Exercise 2

An Interactive 3D Scene

Written By: Ng Xuan Min

I created a pinball and added force to it such that when it is spawned, it will shoot upwards. The pinball will bounce off the obstacles, when I add a physic material component. Physic material allows me to adjust the friction and bouncing effects of the colliding GameObject(s). This allows the pinball to be able to bounce off any GameObject(s) it collides with. To create a physic material, select **Assets** (on the tools bar) -> **Create** -> **Physic Material**, change the properties to desired values and drag and drop to the GameObject(s).

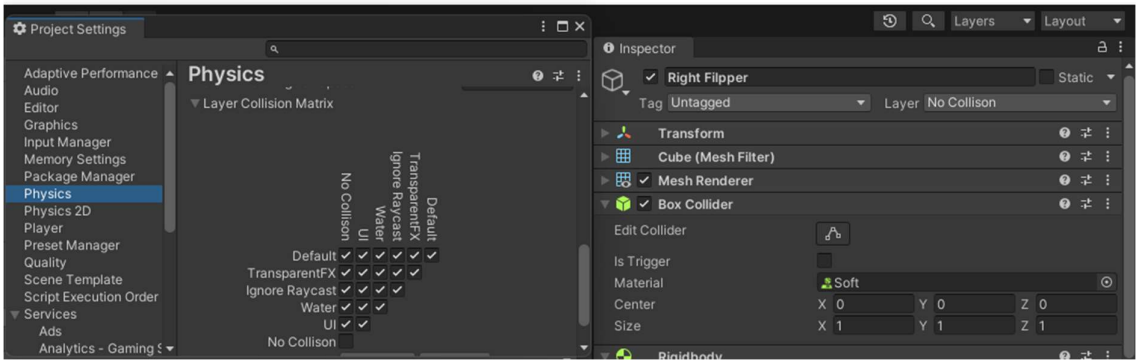


Above is an example of **Physic Material** I've created, I then add it into the GameObject by dragging and dropping it into the **Material** properties (red box) in the inspector column.

To spawn the pinballs, I've created an empty GameObject (spawn point) placed at the bottom right-hand corner where the pinballs will be dispensed and shot out. A script named "game" is written to add force to the pinball such that it can be shot out. The pinball prefab is instantiated at the spawn point and will be shot out when the space bar is pressed. In addition, I added a script such that the light colour changes, when the ball is dispensed. I also added **ForceMode.VelocityChange** to the pinball. This adds an instant velocity change to the pinball's rigidbody and on the other hand, ignoring it's mass.

```
1 reference
void ShootBall()
{
    pinBall pinBall = Instantiate(pinBallPrefab, pinBallSpawnPoint.position, pinBallSpawnPoint.rotation);
    pinBall.rb.AddForce((pinBallSpawnPoint.forward * maxForce), ForceMode.VelocityChange);
    lc.RandomLights();
}
```

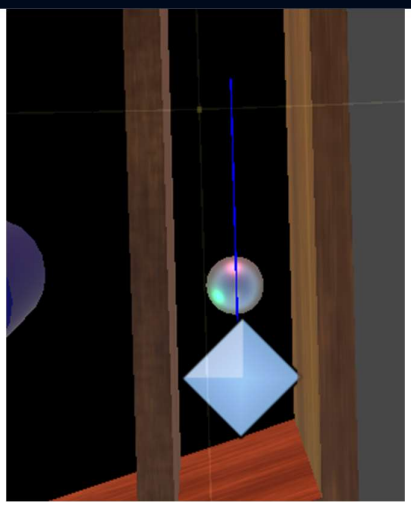
Each of the different GameOject(s) will have collisions occurring if it has a **Rigidbody** component. I layered surfaces that aren't supposed to collide with each other, named "No Collision".



As shown above, I unchecked the box for “No Collision”. This layer is not a preset in Unity, I added it by clicking on layer (in the inspector column) > Add Layer… > User Layer (No.) . After adding the layer, I select it for each GameObject(s) that are not supposed to collide. To uncheck the layer, I have to go to Project Settings > Physics > Layer Collision Matrix, to check or uncheck the layers.

I made use of the Unity’s built in functions to determine the direction of the pinball when it is shot from the spawn point. A script named “debugDirectionBorder” is written to attain this function.

```
Unity Message | 0 references
void OnDrawGizmos()
{
    Gizmos.color = Color.blue;
    Gizmos.DrawLine(transform.position, transform.position + (transform.forward * length));
}
```



As seen above, the is a blue line pointing out of the spawn point (the blue diamond below the pinball).

Project Link:

https://hub.labs.coursera.org:443/connect/sharednninbild?forceRefresh=false