**Lab 6: - Unity – Prefabs**

**Aim of this lab session:**

To learn and practice how to create your own prefabs through the Unity editor and instantiate game objects spawned from the prefabs using C# scripts. If time allows, you may also practice different types of colliders, a continuation of the last lab session.

1. Start the Unity and create a new **2D project** called Lab6\_Unity. Note that this time we are going to work on 2D.
2. Import the “Lab 6\_Asset\_Package.unitypackage” into your project (“Project window” – Right Click – “Import Package” – “Custom Package” – “Import”.
3. Add one player prefab to your scene somewhere near the bottom.
4. Create an empty game object underneath ‘player’ in the hierarchy (as a child object of the ‘player’), call it ‘pongBallSpawn’.
5. Click back on the player and click ‘Override’ (and then ‘Apply’) in the Inspector to save your changes back to the prefab. Double click the ‘player’ in the prefab folder (in the Project) and you shall see the child object ‘pongBallSpawn’ under ‘player’ in the Hierarchy.
6. Add a circle to your scene (by dragging the ‘circle’ from the folder ‘Images’ to the scene). Scale / color it (clicking the ‘Color’ option in the ‘Inspector’ under ‘Sprite Renderer’) to be the pong ball you like. Rename this circle as ‘PongBall’ (in the Hierarchy).
7. Add Rigidbody2D (and set the gravity ‘0’) and ‘Circle Collider 2D’ to the PongBall (by clicking ‘Add Component’ in the ‘Inspector’, then ‘Physics2D’).
8. Now add a new script to the PongBall called ‘PongBallController’, open the script.
9. To start with, give it an ‘m\_speed’ (as a public variable) and get it translating upwards every update (hint: inside the *Update* method, using the function: *transform.Translate*(…)).
10. Back to Unity editor and ‘Play’…your pong ball should fly up.
11. Now drag the object (in the Hierarchy) PongBall into the prefabs folder and then delete the original one.
12. Add the PlayerController script to the Player. Playtest…...you shall be able to move the Player left and right by pressing ‘A’ or ‘D’ key.
13. Open the PlayerController script and add a public m\_pongBallPrefab with the type ‘GameObject’. Save and back to Unity Editor.
14. Click the Player object in the Hierarchy. With the new attribute in the Inspector (for the Player), drag the PongBall prefab from the Prefabs folder over the attribute box.
15. In the PlayerController script, add handling of the ‘Space’ key that instantiates a PongBall object at the position of your pongBallSpawn object with no rotation. <http://docs.unity3d.com/ScriptReference/Object.Instantiate.html> (Hint: use the method getKeyDown(…) to check if a key is pressed, and compare it using the method getKey(…))
16. Back to Unity editor and ‘Play’… your pong balls (being instantiated by pressing the key Space) should fly out and upwards.
17. Now add some Blocks (prefabs) to your scene (by dragging the ‘Blocks’ from the Prefabs folder to the scene)
18. Add 2D collision enter handling to your PongBallController script. This should check the colliding object for a tag (as ‘Block’) and destroy it if it is a Block. Note that, since this is a 2D game, you need to use *OnCollisionEnter2D (Collision2D)* instead. You may need to add a ‘tag’ in the ‘Inspector’ for the Block. (Hint: use the method *Destroy*(…) to make the block which the ball hits disappear)
19. Back to Unity editor and ‘Play’… you should be able to destroy blocks by shooting them.

**Task 6.1**: Finish the steps above and zip two folders, Assets and Project Settings, (Not Include other folders), and email to hit\_submission2023@163.com

You may want to review and practice the steps above. If you are comfortable what you’ve done so far, go ahead to try the followings (**Bonus Points!**):

1. Add bouncing behaviour to your pongBall when it hits a block.
2. Add three walls using the Square sprite (top, left, right) off which the pongBall should bounce.