Introduction to C# programming and Unity

Week 3 - Exercise 12

<u>Move that game object</u>

The Physics 2D Settings define limits on the accuracy of the physical simulation. To be specific, it is where you will set all of the default behaviors for your physics objects. **Rigidbodies** are components that allow a GameObject to react to real-time physics. This includes reactions to forces and gravity, mass, drag and momentum.

1. *<u>Adding game objects</u>*

A game object i.e., the gold ball is inserted in the game scene and the Rigidbody 2d component is added to it. By default due to the action of gravity, the ball falls down. By setting the Y component of gravity (found in Edit->Project Settings) to 0, the ball stays still.

2. *<u>Move the game objects</u>*

GetComponent will return the first component that is found (if the game object has one attached, null if it doesn't) and the order is undefined.

AddForce method adds force to the game object in world space, at the center of mass of the rigid body. The method gets two inputs as parameters as follows.

- <u>Components of force</u> - a vector in 2d space(Vector2d) or in 3d space(Vector3d) and add the force according to the values specified in corresponding axes.
- <u>Method</u> - the method by which the force is applied. The default method is 'Force' which adds force to the rigid body in 2d plane using its mass. The other option available is 'Impulse' which adds an impulse to the rigid body in 2d plane using its mass

GetComponent<Rigidbody2D>().AddForce(new Vector2(0, 5), ForceMode2D.Impulse); - The object moves upward in this case since we specified positive Y value as the parameter.

GetComponent<Rigidbody2D>().AddForce(new Vector2(-5, 0), ForceMode2D.Impulse) The object would move left in this case since we specified negative X value.