Unity
Lab 1

Move Game Object
Method 1

Procedure

1. Start a new 3D project
2. In Hierarchy, rename the SampleScene as **Move**
3. Select the Main Camera object, and set the following parameters in the Camera component
    a. Clear Flags: Solid Color
    b. Background: White
4. Add a 3D Cube object to the scene, and rename it to 'Player'
5. Set the following parameters for the Player's Transform component:
    a. Position: X=0, Y=0, Z=0
    b. Rotation: X=0, Y=0, Z=0
    c. Scale: X=4, Y=4, Z=1
6. Create a new Material and rename it as 'PlayerMat'
    a. Select the Blue from the Color palette
    b. Drag the 'PlayerMat' to the Player's cube in the Scene
7. In the Inspector, select Add Component
    a. Select 'New script'
    b. Enter the script name 'Move'
    c. In Visual Studio, enter the source code Move.cs
8. Run the game by clicking on the Play button
9. Select Player from the Hierarchy, then go to the Inspector and change the targetPosition coordinates

Source Code

```
// Move.cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Move : MonoBehaviour
{
  public float maxVelocity = 1.0f;
  public float Mass = 1.0f;
  public float Friction = 0.05f;
  public Vector2 Acceleration = Vector2.zero;
  [HideInInspector] public Vector2 currentVelocity;
  public Vector2 targetPosition = Vector2.zero;

  private void Start()
  {

  }

  void Update()
  {
    float distance = Vector3.Distance(transform.position, (Vector3)targetPosition);
    Vector2 desiredVelocity = (targetPosition - (Vector2)transform.position).normalized;
    desiredVelocity = desiredVelocity * maxVelocity;
    currentVelocity = desiredVelocity - currentVelocity;
    currentVelocity += Acceleration / Mass;
    currentVelocity -= currentVelocity * Friction;

    if (currentVelocity.magnitude > maxVelocity)
        currentVelocity = currentVelocity.normalized * maxVelocity; // truncate currentVelocity

    if (distance <= 0.01f)
      desiredVelocity = Vector2.zero;
    else
      transform.position = transform.position + (Vector3)currentVelocity * Time.deltaTime;
  }
}
```

Move Game Object
Method 2

Procedure

1. Under Projects, go to the Assets page
2. Duplicate the scene Move (using CTRL+D) and rename it as Move2
3. In the Inspector, remove the old script component
4. In the Inspector, select Add Component
    a. Select 'New script'
    b. Enter the script name 'Move2'
    c. In Visual Studio, enter the source code Move2.cs
5. Run the game by clicking on the Play button
6. Select Player from the Hierarchy, then go to the Inspector and change the targetPosition coordinates

Source Code

```
// Move2.cs
using UnityEngine;

public class Move2 : MonoBehaviour
{
    public float speed = 1.0f; // positive = toward, negative = away
    public Vector2 targetPosition = Vector2.zero;

    void Start()
    {

    }

    void Update()
    {
        float step = speed * Time.deltaTime;
        transform.position = Vector2.MoveTowards(transform.position, targetPosition, step);
    }
}
```

Additional Exercises

Duplicate the Move2 scene and rename the new scene as **Move3**. Modify the code from Method 2 and make your Player move to several different positions when the SpaceBar is pressed once.

Source Code

```
// Move3.cs
using UnityEngine;

public class Move3 : MonoBehaviour
{
    public float speed = 3.0f;
    private Vector2 targetPosition = Vector2.zero;
    public Vector2[] nextPosition;
    private int index = 0;

    void Start()
    {

    }

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            targetPosition = nextPosition[index];
            index++;
        }

        float step = speed * Time.deltaTime;
        transform.position = Vector2.MoveTowards(transform.position, targetPosition, step);
    }
}
```

END OF LAB EXERCISE