GAME DEVELOPMENT CODES

LP1:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CharacterController : MonoBehaviour
{
   public float move = 5f;
   public float jump = 5f;
   private Rigidbody2D rb;
   void Start()
   {
      rb = GetComponent<Rigidbody2D>();
   }
   void Update()
   {
      Vector3 movement = new Vector3(Input.GetAxis("Horizontal"), 0f, 0f);
      transform.position += movement * Time.deltaTime * move;
      if (Input.GetKeyDown(KeyCode.Space))
      {
         rb.AddForce(new Vector3(0f, jump), ForceMode2D.Impulse);
      }
   }
}
```

LP2:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class GameController : MonoBehaviour
{
   public Button b1;
   public Button b2;
   public Button b3;
   public Button b4;
   public Button b5;
   public Button b6;
   public Button b7;
   public Button b8;
   public Button b9;
   public int msg;
   public Text message;
   int target = 0;
   void Start()
   {
      target = Random.Range(1, 9);
      b1.onClick.AddListener(()=>action(1));
      b2.onClick.AddListener(()=>action(2));
      b3.onClick.AddListener(()=>action(3));
      b4.onClick.AddListener(()=>action(4));
      b5.onClick.AddListener(()=>action(5));
      b6.onClick.AddListener(()=>action(6));
      b7.onClick.AddListener(()=>action(7));
      b8.onClick.AddListener(()=>action(8));
      b9.onClick.AddListener(()=>action(9));
   }
   public void action(int msg)
   {
      target = Random.Range(1, 9);
      if (msg == target)
      {
         message.text = "Congrats!!";
```

```
        }
        else
        {
            message.text = "Try again, Value was : "+target;
        }
    }
}
```

LP3:

```
using System.Collections;
using System.Collections.Generic;
using System;
using UnityEngine;
public class ClockController : MonoBehaviour
{
   const float sec = -6f;
   public Transform Pivot;
   void Update()
   {
      var time = DateTime.Now;
      if (Pivot != null)
      Pivot.localRotation = Quaternion.Euler(0f, 0f, sec *time.Second);
   }
}
```

LP4:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Grid2D : MonoBehaviour
{
   public GameObject square;
   public int Width = 5;
   public int Height = 5;
   public float padding = 1.3f;
   void Start()
   {
   SpawnGrid();
   }
   void SpawnGrid()
   {
      for (int x = 0; x < Width; x++)
      {
         for (int y = 0; y < Height; y++)
         {
            Vector2 spawn = new Vector2(x * padding, y * padding);
            Instantiate(square, spawn, Quaternion.identity,transform);
         }
      }
   }
}
```

LP5:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class SpawnSquare : MonoBehaviour
{
   public GameObject squarePrefab;
   public float spawnInterval = 2.0f;
   void Start()
   {
      squarePrefab.transform.position = new Vector2(100, 100);
      StartCoroutine(SpawnAndDestroy());
   }
   IEnumerator SpawnAndDestroy()
   {
      while (true)
      {
         Vector2 spawnPosition = new Vector2(Random.Range(-8.0f, 8.0f),
         Random.Range(-4.0f, 4.0f));
         GameObject square = Instantiate(squarePrefab, spawnPosition,
         Quaternion.identity);
         yield return new WaitForSeconds(spawnInterval);
         Destroy(square);
      }
   }
}
```

LP6:

```
using System.Collections.Generic;
using UnityEngine;

public class ColorChanger : MonoBehaviour
{
   public List<Color> colors;
   private int index = 0;

   void Update()
   {
      if (Input.GetMouseButtonDown(0) &&
Physics2D.Raycast(Camera.main.ScreenToWorldPoint(Input.mousePosition), Vector2.zero).collider !=
null)
         GetComponent<SpriteRenderer>().color = colors[index = (index + 1) % colors.Count];
   }
}
```

(OR)

```
using System.Collections.Generic;
using UnityEngine;
public class ColorChanger : MonoBehaviour
{
   public List<Color> colors;
   private int currentColorIndex = 0;
   void Update()
   {
      if (Input.GetMouseButtonDown(0))
      {
         Vector2 mousePosition =
         Camera.main.ScreenToWorldPoint(Input.mousePosition);
         RaycastHit2D hit = Physics2D.Raycast(mousePosition, Vector2.zero);
         if (hit.collider != null && hit.transform == this.transform)
         {
            currentColorIndex = (currentColorIndex + 1) % colors.Count;
```

```
            GetComponent<SpriteRenderer>().color = colors[currentColorIndex];
        }
    }
  }
}
```