

## 1. Instantiate a 5 x 5 grid of squares using sprites and C# Code

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Grid2D : MonoBehaviour
{
    public GameObject squarePrefab; // Assign your square sprite in the
    inspector
    public int gridWidth = 5;
    public int gridHeight = 5;
    public float padding = 1.3f; // Space between sprites
    void Start()
    {
        SpawnGrid();
    }
    void SpawnGrid()
    {
        for (int x = 0; x < gridWidth; x++)
        {
            for (int y = 0; y < gridHeight; y++)
            {
                // Calculate position for each sprite in the grid
                Vector2 spawnPosition = new Vector2(x * padding, y * padding);
                // Instantiate the sprite at the calculated position
                Instantiate(squarePrefab, spawnPosition, Quaternion.identity,
                transform);
            }
        }
    }
}
```

## **2.Randomly instantiate sprites and disappear after a specific amount of time**

```
using System.Collections;
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)
        {
            // Spawn a square at a random position within the camera's view
            Vector2 spawnPosition = new Vector2(Random.Range(-8.0f, 8.0f),
            Random.Range(-4.0f, 4.0f));
            GameObject square = Instantiate(squarePrefab, spawnPosition,
            Quaternion.identity);
            // Wait for the specified interval, then destroy the square
            yield return new WaitForSeconds(spawnInterval);
            Destroy(square);
        }
    }
}
```

**3.Square should change its color when clicked. Colors should cycle through the VIBGYOR colors.**

C/C++

```
using System.Collections.Generic;
using UnityEngine;
public class ColorChanger : MonoBehaviour
{
    // List of colors
    public List<Color> colors;
    private int currentColorIndex = 0; //current color
    void Update()
    {
        if (Input.GetMouseButtonDown(0)) //check for mouse click
        {
            Vector2 mousePosition =
                Camera.main.ScreenToWorldPoint(Input.mousePosition); //get mouse
                position
            RaycastHit2D hit = Physics2D.Raycast(mousePosition, Vector2.zero);
                //Cast
            ray from mouse position
            if (hit.collider != null && hit.transform == this.transform) //check if ray
            hits square
            {
                // Change to the next color in the list
                currentColorIndex = (currentColorIndex + 1) % colors.Count;
                GetComponent<SpriteRenderer>().color = colors[currentColorIndex];
            }
        }
    }
}
```

#### **4.Create a 2D character and write a C# script which will allow the player to move and jump**

C/C++

using UnityEngine;

public class CharacterController : MonoBehaviour

{

public float moveSpeed = 5f;

public float jumpForce = 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 \* moveSpeed;

if (Input.GetKeyDown(KeyCode.Space))

{

rb.AddForce(new Vector3(0f, jumpForce), ForceMode2D.Impulse);

}

}

}

## 5. Create a simple “Guess the Number” game using Unity’s UI Features

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

public class GameController : MonoBehaviour
{
    public Button button1;
    public Button button2;
    public Button button3;
    public Button button4;
    public Button button5;
    public Button button6;
    public Button button7;
    public Button button8;
    public Button button9;
    public int answer;
    public Text answerTxt;
    int targetNumber = 0;

    void Start()
    {
        targetNumber = Random.Range(1, 9);
        button1.onClick.AddListener(() => ButtonInputBehaviour(1));
        button2.onClick.AddListener(() => ButtonInputBehaviour(2));
        button3.onClick.AddListener(() => ButtonInputBehaviour(3));
        button4.onClick.AddListener(() => ButtonInputBehaviour(4));
        button5.onClick.AddListener(() => ButtonInputBehaviour(5));
        button6.onClick.AddListener(() => ButtonInputBehaviour(6));
        button7.onClick.AddListener(() => ButtonInputBehaviour(7));
        button8.onClick.AddListener(() => ButtonInputBehaviour(8));
        button9.onClick.AddListener(() => ButtonInputBehaviour(9));
    }

    public void ButtonInputBehaviour(int answer)
    {
        targetNumber = Random.Range(1, 9);
        if (answer == targetNumber)
```

```

{
    answerTxt.text = "Congrats!!";
}
else
{
    answerTxt.text = "Try again, Value was : "+targetNumber;
}
}
}

```

## 6.Create a simple 2D clock with moving seconds arm

```

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

```