# Number

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
public class Number : 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;
}
}
}
```

## Grid

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class GRID : MonoBehaviour
{
public GameObject squarePrefab;
public int gridWidth = 5;
public int gridHeight = 5;
public float padding = 1.3f;
void Start()
{
SpawnGrid();
}
void SpawnGrid()
{
for (int x = 0;x<gridWidth; x++)
{
for (int y = 0;y<gridHeight; y++)
{
Vector2 spawnPosition = new Vector2(x * padding, y * padding);
Instantiate(squarePrefab, spawnPosition, Quaternion.identity,
transform);
}
}
}
}
```

## SPAWN

```
using System.Collections;
using UnityEngine;
public class Spawn : 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);
}
}
}
```

## Color

```
using System.Collections.Generic;
using UnityEngine;
public class Color1 : 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];
}
}
}
}
```

## Clock

```
using System;
using UnityEngine;
public class Clock : 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);
}
}
```

## Char-move

```
using UnityEngine;
public class Char : 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);
}
}
}
```