CODE A GAME

FOR A DUMMY, BY A DUMMY

MANAAL ASIM

woohoo!

WHAT IS CODING?

Code is a fancy language for computers.
It is a task based langiuage. you tell the computer what to do in the right terms, and it does it.

Electronic devices like cell phones, laptops, and tablets require code to function properly. Coding allows humans to communicate with these devices. Modern technology such as traffic lights, calculators, smart TVs, and cars use internal coding systems.

There are many coding languages, they are called programming languages, such as:
python, c++, java, lua, COBOL, PHP, R, Excel and go. are the top programming languages, all used for different reasons.

Thats a lot right?

Computers follow the set of instructions given to perform tasks, it involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in a programming language.

Users interact with websites and apps with the help of these coded instructions. The code itself does not show up on the screen. Instead, people see the words and images a developer or programmer asks the computer to display.

UNITY

“All great games begin with an idea. Make yours come to life for eager players around the world - with the confidence that you have the powerful tools, on-demand services, growth plan, and support you need.” [unity]

Unity is a software which allows users to build games, apps, or immersive experiences. If you’ve played games like Call of Duty Mobile or Pokémon Go, you’ve played a Unity game.

This software lets you assemble assets, which are pieces used in creating games such as objects, animations, sounds , backgrounds, etc. it also allows you to write code to interact with these assets; an important part to making games.

in order to make a game, one has to be able to actually play with the objects right? that is what unity does.

LETS DESIGN A GAME: WHAT YOU NEED

So now it's time to make a game where characters appear and you have to collect them to score points.

Fun concept!

Before you begin coding:
Pick your characters- What characters do you want?

Pick your score font- Something cool, techno maybe?

Pick your backgrounds- Do you want it in mixed reality? do you want it in VR? Pick a vibe.

Make a story-What are you trying to achieve? Think of this book of codes as a template. Play around with it

SPAWN CONTROLLER CODE

This code actually allows the characters to show up on the screen. When the code is entered, you place the characters into it using unity.

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

public class CubeSpawner : MonoBehaviour
{
    // Start is called before the first frame update

    public GameObject prefab;
    public float spawnDelay = 5f;

    void Start()
    {
        StartCoroutine(SpawnObjectRepeatedly());
    }

    private IEnumerator SpawnObjectRepeatedly()
    {
        //while (true)
        //{
            yield return new WaitForSeconds(spawnDelay);
            //for(int i = 0; i < 50; i++)
            //{
                float randomx = Random.Range(-2f, 2f);
                float randomz = Random.Range(-2f, 2f);
                Vector3 spawnPosition = new Vector3(randomx, 0, randomz);
                Instantiate(prefab, spawnPosition, Quaternion.identity);
            //  }


      //  }
    }

    // Update is called once per frame
    void Update()
    {

    }
}

```

SCENES CHANGING

If you have a start page and a main game page, this code actually tells thed computer to shift from one page to the other when an action is done.

```
using UnityEngine;
using UnityEngine.SceneManagement;

public class Control : MonoBehaviour
{
    public void NextScene()
    {
        SceneManager.LoadScene("SampleScene");
    }
}
```

DEBRIS CONTROLLER

This is the main control for the game. every game has a main code, the foundation to the actions.

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

public class DebriController : MonoBehaviour

{

    public static int score = 0;


    public float maxRaycastDistance = 5f; // Maximum distance for raycast

    // Start is called before the first frame update
    void Start()
    {
        Debug.Log(score);
    }

    // Update is called once per frame
    void Update()
    {
        Vector3 controllerPosition = OVRInput.GetLocalControllerPosition(OVRInput.Controller.LTouch);
        Quaternion cr = OVRInput.GetLocalControllerRotation(OVRInput.Controller.LTouch);

        Vector3 rayDirection = cr * Vector3.forward;

        RaycastHit hit;

        if(Physics.Raycast(controllerPosition, rayDirection, out hit, maxRaycastDistance))
        {
            score = score + 1;
            Debug.Log(score);
            Destroy(hit.collider.gameObject);
        }
        //if (OVRInput.Get(OVRInput.RawButton.LIndexTrigger))
        //{
        //    transform.position += new Vector3(transform.position.x, 1, transform.position.z);
        //}
    }
}

```

SCORING

Now a game isn't fun without keeping track of the scores, how else will the player know how good they are?

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4   using UnityEngine.UI;
5   using TMPro;
6
7   public class ScoreGUI : MonoBehaviour
8   {
9       public TMP_Text labelText;
10      // Start is called before the first frame update
11      void Start()
12      {
13
14
15      }
16
17      // Update is called once per frame
18      void Update()
19      {
20          string t = ""+DebriController.score;
21          labelText.text = t;
22
23      }
24  }
25
```

DETECT OBJECTS

The game won't actually work if the controller cannot detect the characters. Tt needs to destroy them!

It is actually the part of the code in the debris controller.

```
    {
        Debug.Log(score);
    }

    // Update is called once per frame
    void Update()
    {
        Vector3 controllerPosition = OVRInput.GetLocalControllerPosition(OVRInput.Controller.LTouch);
        Quaternion cr = OVRInput.GetLocalControllerRotation(OVRInput.Controller.LTouch);

        Vector3 rayDirection = cr * Vector3.forward;

        RaycastHit hit;

        if(Physics.Raycast(controllerPosition, rayDirection, out hit, maxRaycastDistance))
        {
            score = score + 1;
            Debug.Log(score);
            Destroy(hit.collider.gameObject);
        }
```

TEST IT OUT!

Maybe it works maybe it doesn't. Coding is a process, make sure you keep testing and keep learning.

Coding is also a pain so make sure every detail is written correctly otherwise it won't work!