

# STEAL LIKE AN ARTIST

## 1. **Game Introduction**

Steal Like an Artist is a cross-platform game based on your experience as an artist, trying to find elements you can steal from paintings at the museum. But unfortunately, security is on its way! find 3 to 6 elements before time ends to escape and create your own paintings! For the demo version, you will walk around the museum in order to steal colors from the art pieces. If you want to escape from the police you need to find between 3 to 6 colors before the time ends. In the complete version, you will be able to steal elements from the art and you will have to move them around a canvas, mix them, and create your own personal paint! Share it with your family and friends through the mobile version or VR.

---

### **Genre**

It's a family game, a demo version 3D, and future mobile and VR production.

### **GamePlay**

At each level, the player starts in a museum and must steal elements from paintings and sculptures to build their own painting before the police catch them.

### **Target audience & platforms**

Our primary target audience is teenagers and young adults.

### **Look and Feel**

You will be in the middle of a museum night, soon you will find this museum is built like a maze, would you be able to escape before the police catch you?

- Story

Steal Like an Artist is a game based on your experience as an artist, trying to find elements you can steal from paintings at the museum. Unfortunately, security is on its way! find 3 to 6 elements before the time ends to escape and create your own paintings!

- How to Play

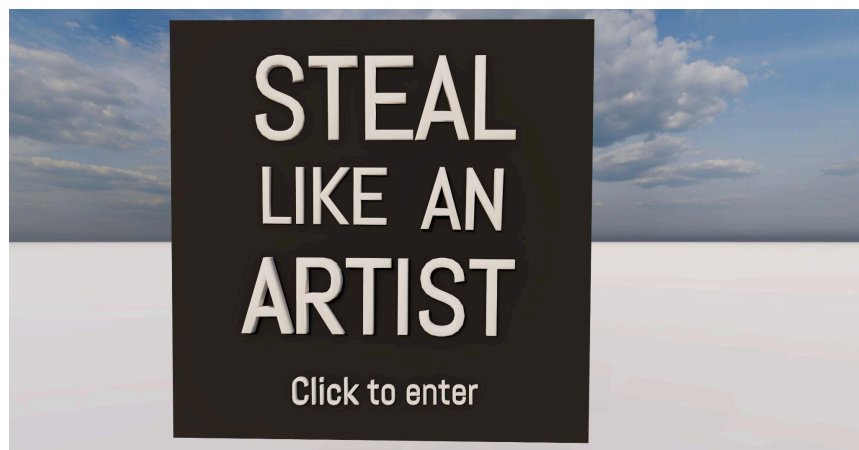
- To begin with the demo version, you must first click enter.
- For the demo version, on pc, you can move around the maze using the arrows. In order to move your view, use the mouse.
- You will “find” something to steal by walking directly to the art piece.
- Use the mouse to steal an element. In the demo version, you will recognize the element as a circle in a color related to the piece.
- Move as fast as possible to find at least 3 objects to steal if you want to escape from the police.

- A mobile version will include arrow buttons, a VR version you will move by pointing an arrow and pressing the trigger button to the point where you want to go.
- 

## 2. Artwork (design)

As it is a museum, the quality and design of the art will include from three-dimensional parametric compositions to flat pointillism paintings or conceptual art. The museum is an exhibition piece itself as it is treated as an element of three-dimensional sculpture. You can walk around while you get lost in the art on display as well as in the intricate labyrinths of the museum. It will take you to passageways with no exit but that allows you to Discover the hidden pieces of art. All the artwork was made in Virtual Reality thinking about the experience of the user in a future VR game.

Start View



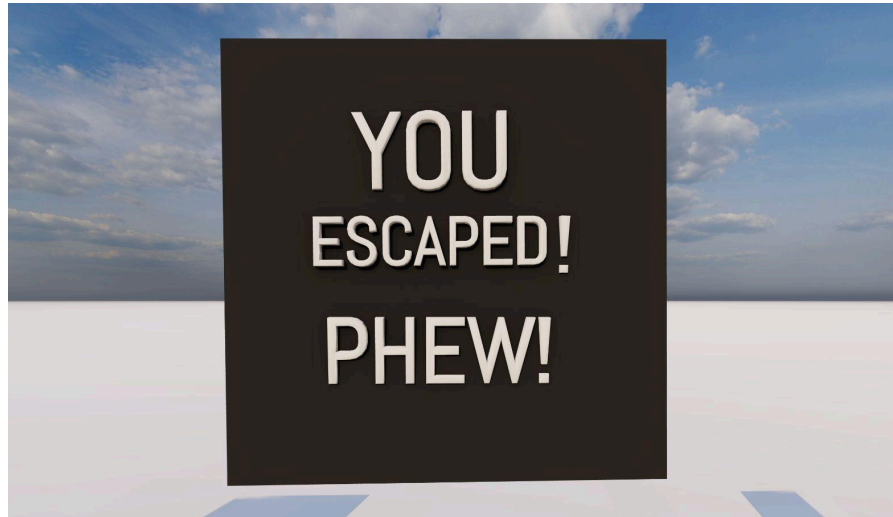
The start view is a 3d text that allows you to start the game. it has also the name of the game on it.

Busted View



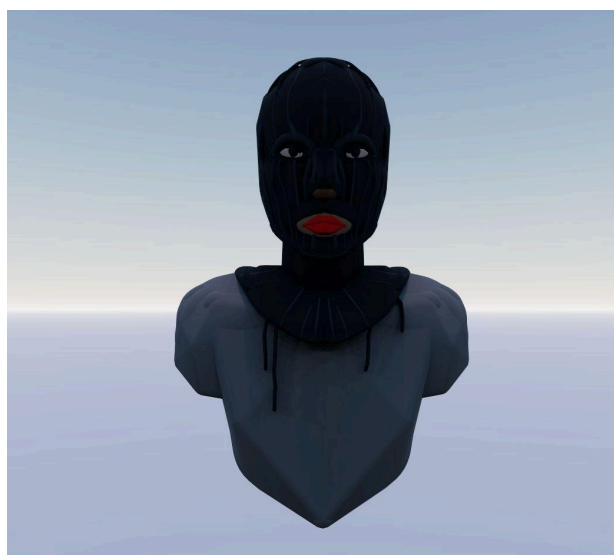
In case you couldn't find enough art pieces in the time this 3d text will appear telling you that you failed.

Escaped view



This 3d text appears when you succeed in stealing the art pieces.

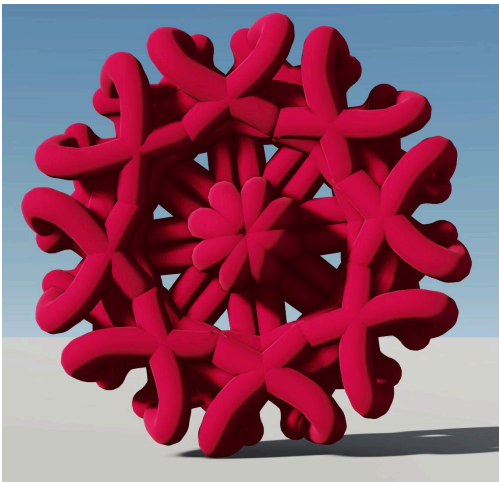
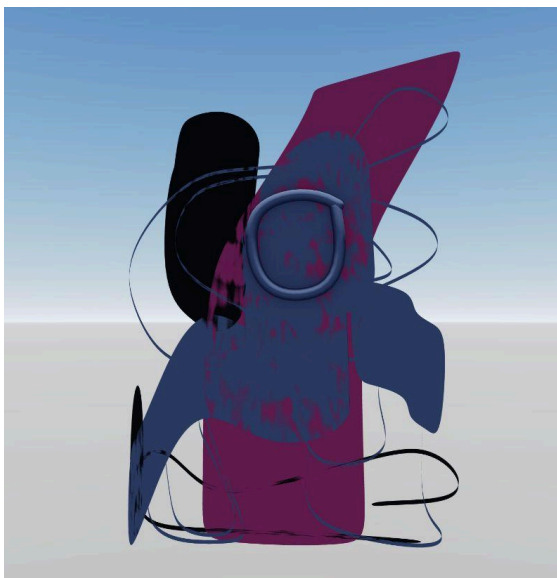
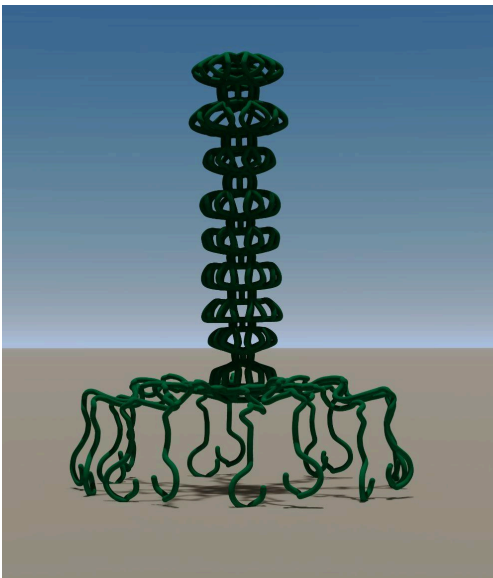
The Avatar

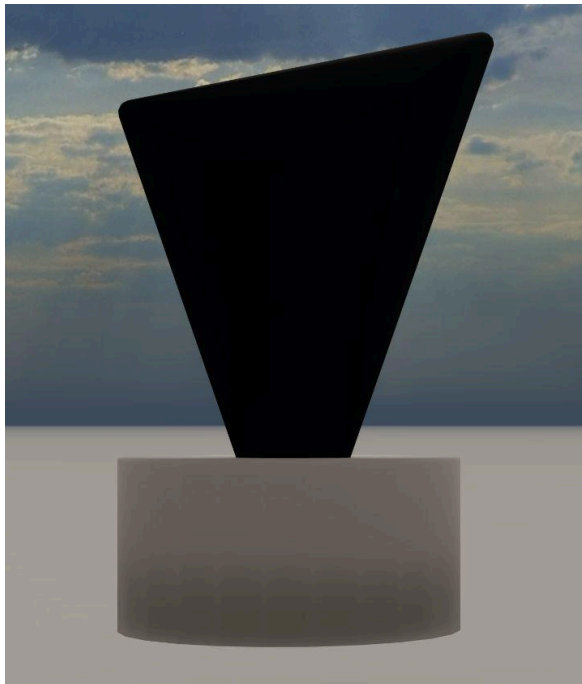
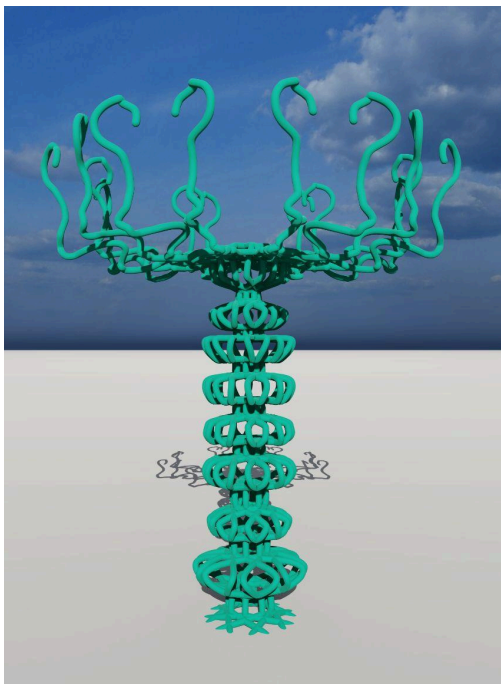
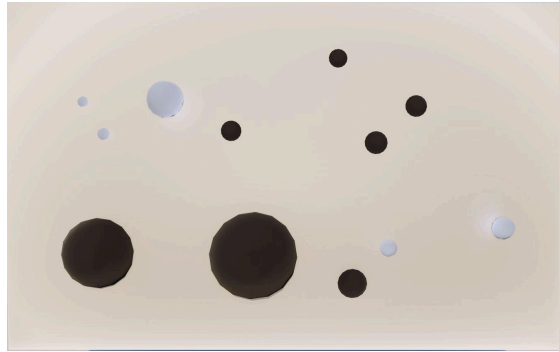


As a thief, you are trying to be as stealthy as possible, to not be discovered you are using a mask that protects you from any camera in the museum.

Modern Art in the Museum

The texts, the avatar, and the art pieces for the museum were made for our Artist Olivia Barron in Virtual Reality.





- 
- Technical Requirements  
The art pieces at the museum will be created as hand-painted in VR.
  - Characters
    - Character 1

You are the principal character, a famous artist who now looking for some inspiration decided to steal some ideas from the museum pieces.



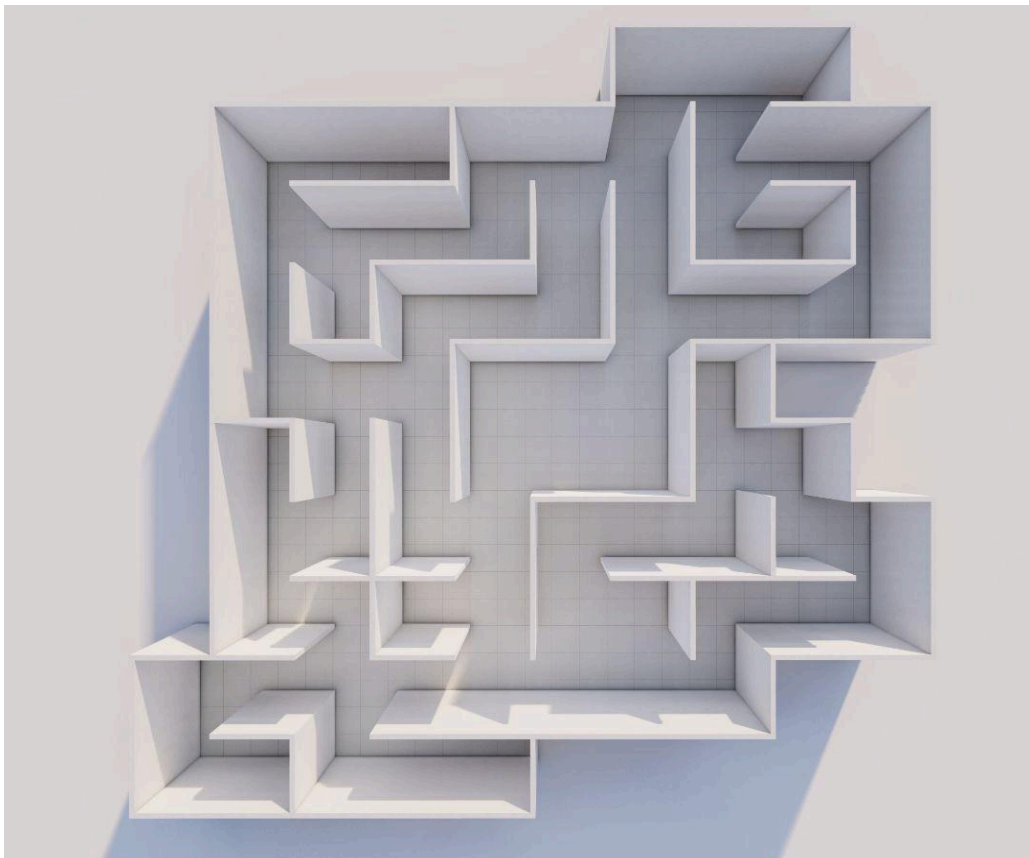
- Character 2

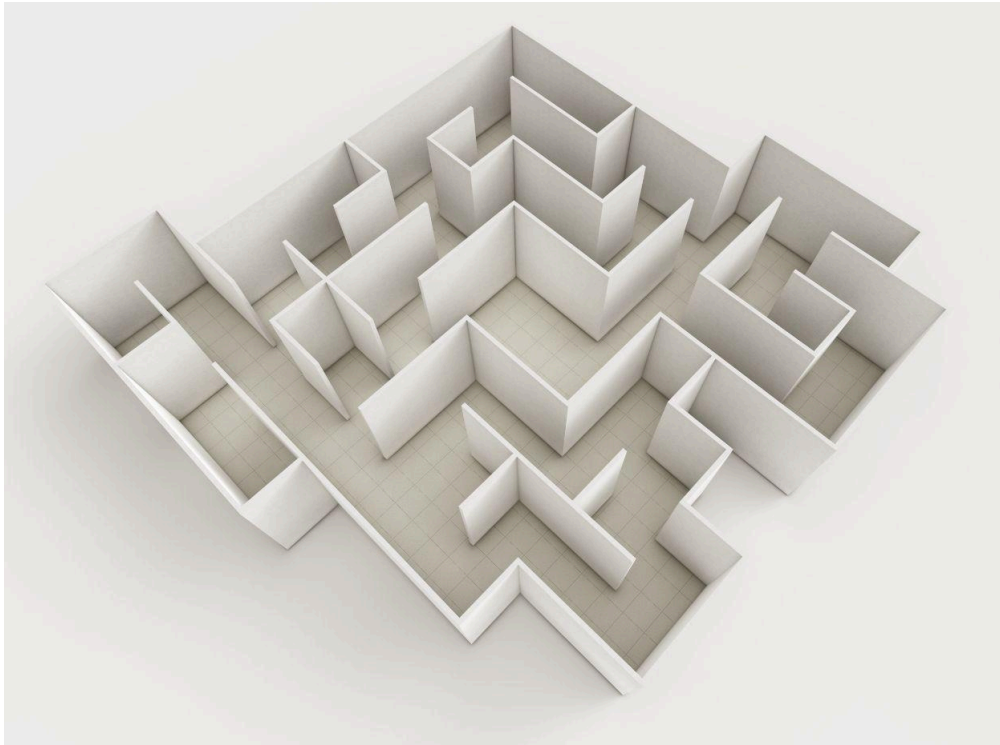
The police! In the demo version, the police will arrive once your time is finished. For future VR versions, the police will chase you around the museum.



- \_\_\_\_\_
- Level Design

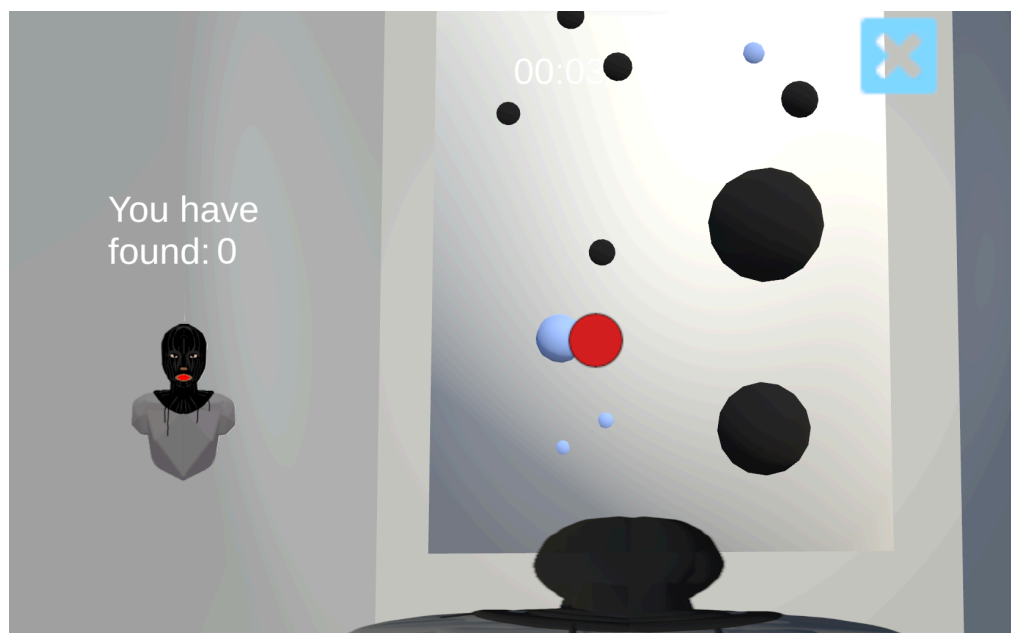
### The Museum



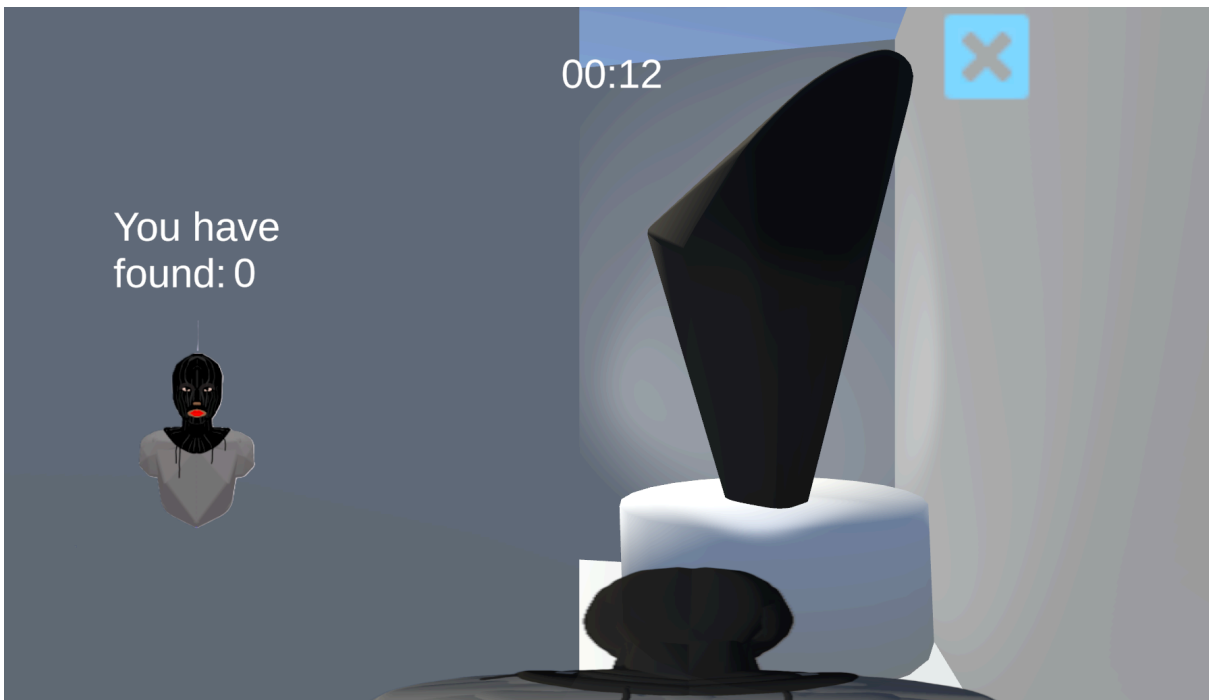


The museum is a labyrinth that allows you to get loose into the art. It can be walked in many ways giving you the chance to collect different art pieces. The museum is designed based on a grid, mixing different kinds of spaces like narrow hallways, big rooms, or dead ends. Every single space is design thinking in the experience of the player, allowing him to explore in a unique way. The museum was designed by our Architect Oscar Escalera.

- Player View
  - The player once has found something to steal



- The player in the museum walks in front of an art piece to steal the white color.



- Video Game Play
    - [https://drive.google.com/file/d/1CGAzwxWkDi7eiamE-WhNxYZh21waFyrY/view?usp=drive\\_link](https://drive.google.com/file/d/1CGAzwxWkDi7eiamE-WhNxYZh21waFyrY/view?usp=drive_link)
  - Download Demo Version
    - [https://drive.google.com/drive/folders/1ASea-3Dt-m9VHN\\_NnAFcLPM1Igd3Xd8s?usp=sharing](https://drive.google.com/drive/folders/1ASea-3Dt-m9VHN_NnAFcLPM1Igd3Xd8s?usp=sharing)
3. Audio & Sound F/x
- The use of 360 Audio in the VR version would be required, including environment “silence” sound and your steps. For the Demo version, you are at night without a sound.
-

#### 4. Scripts.

Made by Camila Claros

- loadScene

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

using UnityEngine;
using UnityEngine.SceneManagement;

public class loadScene : MonoBehaviour
{

    public void LoadNextScene()
    {
        int currentSceneIndex = SceneManager.GetActiveScene().buildIndex;
        SceneManager.LoadScene(currentSceneIndex + 1);
    }

}
```

- counter

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

public class counter : MonoBehaviour
{

    public int count = 0;
    public TextMeshProUGUI countText;

    public void Increment()
    {
        count++;
        UpdateText();
        if(count==6)
        {
            LoadNextScene();
        }
    }

    void UpdateText()
    {
        countText.text = count.ToString();
    }

    public void LoadNextScene()
    {
        int currentSceneIndex = SceneManager.GetActiveScene().buildIndex;
        SceneManager.LoadScene(currentSceneIndex + 1);
    }

}
```

- interactable

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

public class interactable : MonoBehaviour
{
    public GameObject interactionCanvas;
    public GameObject movementObject;

    void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.tag == "Player")
        {
            interactionCanvas.SetActive(true);
            movementObject.SetActive(false);
        }
    }
}
```

- mouseLook

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

public class mouseLook : MonoBehaviour
{
    public float mouseSensitivity = 100f;

    public Transform playerBody;

    float xRotation = 0f;

    void Start()
    {
        Cursor.lockState = CursorLockMode.Locked;
    }

    void Update()
    {
        float mouseX = Input.GetAxis("Mouse X") * mouseSensitivity * Time.deltaTime;
        float mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity * Time.deltaTime;

        xRotation -= mouseY;
        xRotation = Mathf.Clamp(xRotation, -90f, 90f);

        transform.localRotation = Quaternion.Euler(xRotation, 0f, 0f);
        playerBody.Rotate(Vector3.up * mouseX);
    }
}
```

- mouseMovement

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

public class mouseMovement : MonoBehaviour
{
    public float speed = 10f;

    void Update()
    {
        Cursor.lockState = CursorLockMode.None;

        if (Input.GetKey(KeyCode.UpArrow))
        {
            // Get mouse position
            Vector3 mousePosition = Input.mousePosition;

            // Convert screen point to world point
            Vector3 worldPoint = Camera.main.ScreenToWorldPoint(mousePosition);

            // Get direction to mouse
            Vector3 direction = worldPoint - transform.position;
            direction.y = 0f;

            // Move towards mouse
            direction = direction.normalized;
            transform.position += direction * speed * Time.deltaTime;
        }
    }
}
```

- PlayerCollision

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

public class PlayerCollision : MonoBehaviour
{
    private Rigidbody rb;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
    }

    void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.tag == "Obstacle")
        {
            rb.velocity = Vector3.zero;
            rb.angularVelocity = Vector3.zero;
        }
    }
}
```

- playerMovement

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

public class playerMovement : MonoBehaviour
{
    public float speed = 5f;

    void Update()
    {
        Cursor.lockState = CursorLockMode.None;
        float x = Input.GetAxis("Horizontal") * speed;
        float z = Input.GetAxis("Vertical") * speed;

        Vector3 movement = new Vector3(x, 0, z);
        movement = Vector3.ClampMagnitude(movement, speed);

        transform.Translate(movement * Time.deltaTime);
    }
}
```

- quitButton

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

public class quitButton : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
    }

    public void quitGame()
    {
        Application.Quit();
    }

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

- TimerScript

```
using UnityEngine;
using UnityEngine.SceneManagement;
using TMPro;

public class TimerScript : MonoBehaviour
{
    public float timeLeft = 60f;
    public int count2 = 0;
    public string sceneToLoad;

    public TextMeshProUGUI timerText;

    void Update()
    {
        if (timeLeft > 0)
        {
            timeLeft -= Time.deltaTime;
            UpdateTimerText();
        }
        else
        {
            if (count2 >= 3)
            { sceneToLoad = "escaped"; }
            else sceneToLoad = "police";
            SceneManager.LoadScene(sceneToLoad);
        }
    }

    public void Increment()
    {
        count2++;
    }

    void UpdateTimerText()
    {
        int minutes = Mathf.FloorToInt(timeLeft / 60);
        int seconds = Mathf.FloorToInt(timeLeft % 60);

        timerText.text = string.Format("{0:00}:{1:00}", minutes, seconds);
    }
}
```