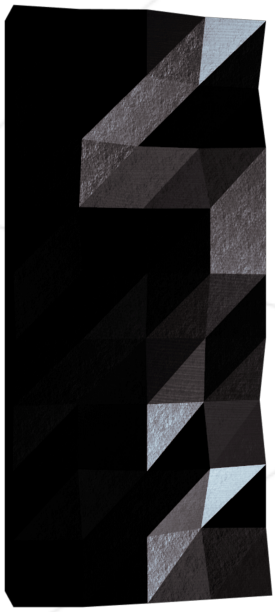

Freddie Mercury

He wants to break free

Summary: In this project, you will code a brand new concept in VR! The player will have to resolve subtle puzzles and mysteries to unlock new steps and finally break free! Ok, it's an escape game in VR...

Version: 1.1

Contents

Chapter I

Preamble

Here we are, born to be kings
We're the princes of the universe
Here we belong, fighting to survive
In a world with the darkest powers

Why is this project called Freddie Mercury? Well, it's not based on That song...

Chapter II

Introduction

Your goal is to code an Escape Game in VR.
This project mixes game design, some 3D, et some VR-specific interactions.

You will need a VR headset at some point in this project, and at least for the evaluations! You can use your own. On some Campuses, you might be able to borrow one for the project duration.

Chapter III

Objectives

This project aims to master the basics of VR controls and VR game development.

Chapter IV

Instructions

General instructions :

- You must use a Game Engine: Unity, Unreal Engine, Godot, CryEngine, ... or any other game engine.
- You can use any 3D Asset (meshes, animation rigs...), image assets and audio assets, as long as the copyrights and/or intellectual property rules attached to the asset are fully respected. The graphical quality is not an important part of the evaluation.
- Assets are not directly evaluated but the overall look and feel of game must be as professional as possible.
- You must use default official APIs and scripts to handle VR (provided by the game engine or the headset SDK), no advance packages are allowed.
- You must code objects interaction, rotation, and movement using the rawest API for your game engine/headset.

Chapter V

General instructions

Your Escape Game must come to the player with the following constraints:

- Full VR game.
- Compatible with at least one VR headset: Vive, Meta/Oculus, Apple/VisioPro, Playstation, ... or any other VR headset with at least hand tracking.
- A lobby: a start zone before the game, showings controls, tutorials, options.
- Entering the first level must be done with an action in the lobby and with an adequate transition (screen fade or loading animation...)
- Elements in the lobby or in the first level start telling a little story (like in an escape game !)
- The gamer must understand by the situation and surrounding space that he is locked and must escape the room/area.
- The gamer can move in the space (teleport is ok) and interact with some objects. At least :
    - grabs objects with his hands, manipulate, turn, passing them hands to hands, drop them, throw them.
    - interacts with rotative elements, on one or more axes, like doors, lockpads, bottle caps...
    - interacts with elements by push/pull.
- The gamer can make some objects interact with other objects: a key with a keylock, or smoke with fire-detection, water with electricity... .
- Some actions create automatic events.
- The game has coherent sounds according to events and global ambiance.
- The game contains at least 2 levels.
- The game / adventure / story have an end, on a dedicated level or screen, or when back to the lobby.

- Other considerations:
  - The game must be polished, no weird bugs or glitches, no crappy controls or collisions.
  - VR comfort is a must have: if motion sickness is too high, nobody will play and evaluate you.
  - Hand controls must be as natural as possible.
  - The game FPS is stable and fluid on every occasion.

Chapter VI

Bonus Part

Anything that makes the game more professional can be part of the bonuses. Some examples:

- The game handles multiple VR headsets.
- A LOT of contents.
- Multiplayer mode.
- Coherent and beautiful world.
- Interaction with in-game characters.
- Saving/loading system to continue later.

Chapter VII

Turn-in and peer-evaluation

As usual, turn-in your work on your repo `GiT`. Only the work included on your repo will be reviewed during the evaluation.
If your assets are over 1Go, it will be a good idea to not push them directly in your repo and replace your asset folder with a clean md5 recursive hash of the folder, to ensure that nothing has changed before the evaluation.