

## Assignment 03 (Individual) – GAMES DEVELOPMENT

SE4031 – Games Development

Title – *Immersive VR Sci-Fi Survival Experience: ORBITAL FAILURE*

Assignment Created By: Mr. Aruna Ishara Gamage, Mr. Nushkan Nismi

**End Digit 8**

Assignment Weight: **30%**

## OBJECTIVE

### *Orbital Failure*

**Design and develop an immersive Virtual Reality (VR) sci-fi survival experience application using Unity, where the player is trapped inside a malfunctioning space station.**

The player must navigate the station, interact with futuristic systems, issue voice-based system commands, and survive escalating failures through a voice-driven emergency challenge

## ASSIGNMENT STRUCTURE (2 Parts)

- **Part A (60%) – Based on Tutorials + Lab Sheets**

Students can score Part A using concepts and techniques covered in tutorials and lab sessions

- **Part B (40%) – Self-Learning + Advanced Work (Required for High Grades)**

Part B must include voice-controlled mechanics and a unique survival challenge developed through independent research

## CORE REQUIREMENTS

### Part A Requirements (60%)

#### 1. VR Movement & Locomotion

- Use teleportation or joystick-based movement
- Allow smooth navigation across multiple station areas (corridors, control rooms, engineering bays)

#### 2. Futuristic System Interactable

- Implement at least three (3) of the following:
  - **Control panels or terminals:** Activate or monitor station systems
  - **Valves, levers, or switches:** Control power, pressure, or flow
  - **Doors or airlock mechanisms:** Open, close, or lock access points
  - **Repairable system components:** Fix damaged systems using interaction

#### 3. VR UI Elements

- Include a HUD or VR-friendly UI displaying:
  - Station system status (power, oxygen, integrity)
  - Alerts or warnings
  - Timers or objectives

#### 4. Basic Environmental Feedback

- Alarm sounds, flickering lights, warning sirens
- Environmental feedback reacting to player actions or failures

## Part B Requirements (40%) – Self Learning

### 5. Voice-Based System Command Control (Voice Input Only)

- The player must control **at least three (3)** station systems using voice commands only

#### Example Commands (DO NOT USE):

- “Restore Oxygen”** – Stabilizes oxygen levels
- “Seal Breach”** – Closes hull damage
- “Redirect Power”** – Transfers power to critical systems

#### **⚠️ IMPORTANT NOTE**

- The system command phrases listed below are **EXAMPLES ONLY**
- Students **MUST** create their **own original** command phrases
- Using the example commands exactly as written will result in **0 marks** for this component

#### The system must:

- Accurately detect commands using voice input tools (e.g., Windows Speech API (wit.ai))
- Provide visual/audio feedback upon successful commands recognition
- Show a response or warning for unrecognized commands
- Please note that the Windows Speech API will NOT work when the application is built and executed inside a VR environment

### 6. Voice Command Integration

- Integrate a real-time** speech recognition system
- Provide visual and/or audio feedback for:
  - Successful** system commands
  - Incorrect** or **unrecognized** commands
- Voice system must **remain responsive** during gameplay

### 7. Dynamic Station Reaction System

- Voice commands** and player actions must cause clear system-level reactions, such as:
  - Oxygen depletion or restoration
  - Emergency lighting or power loss
  - Alarm escalation or shutdown.

### 8. Emergency Challenge Area (Mandatory)

- Include a dedicated emergency sequence where the player must:
  - Use voice** commands under time pressure
  - Interact** with station systems to survive
  - Complete a repair, stabilization, or evacuation challenge**

#### **⚠️ Part B Creative Challenge Note**

**Part B** includes a creative challenge, and it must be unique (**not copied from other students**) to score marks.

## **Folder Structure & Code**

- Unity project must follow the folder structure.
- Scripts and assets must be well-named and organized.

## **Game Documentation**

- Submit a PDF with:
  - Title, student name, and IT number
  - Game summary and spell list
  - Screenshots of gameplay
  - Control guide (movement, voice usage)
  - Credits for any assets/tools used

## **Submission Requirements**

- **Windows .exe Build**  
Include .exe and Data folder, playable with a VR headset.
- **Zipped Unity Project Folder**  
Must follow the folder structure.  
Upload to a shared Google Drive folder in Courseweb.
- **Gameplay Demo Video**
  - 5 minutes showcasing:
    - Voice-activated spellcasting
    - Movement
    - Item collection and target interaction
    - Spellbook interaction
    - Challenge zone gameplay

## **PLAGIARISM / ORIGINALITY VERIFICATION (VIVA)**

- A mandatory one-to-one viva will be conducted
  - Students must clearly explain:
    - Voice command logic
    - System interaction behavior
    - Emergency challenge design
- Failure to justify originality may result in mark deductions or zero marks

### **Assessment Rubric (Part A + Part B)**

#### **Part A (60 Marks)**

<b>Criteria</b>	<b>Excellent</b>	<b>Good</b>	<b>Satisfactory</b>	<b>Poor</b>	<b>Marks</b>
<b>VR Movement &amp; Locomotion</b>	Smooth and immersive navigation implemented	Navigation implemented with minor issues	Basic navigation implemented with limitations	No or broken navigation	12
<b>System Interactable</b>	All 3 interactable types fully implemented and functional	2 interactable working with minor issues	1 interactable working or buggy	No meaningful interactable	12
<b>VR UI Elements</b>	Clear VR-friendly HUD with system status, alerts, and indicators	UI mostly clear with minor issues	Basic UI with limited usefulness	No VR UI elements	10
<b>Folder Structure &amp; Code</b>	Fully structured project with well-named assets and scripts	Mostly structured with few misplacements	Some structure but inconsistent	Disorganized or missing structure	8
<b>Game Documentation</b>	Full PDF with all required sections, clear and informative	Mostly complete PDF with minor missing details	Basic write-up, lacks detail	No documentation submitted	8
<b>Part A Total</b>					<b>60</b>

## **Part B (40 Marks) – Self Learning**

<b>Criteria</b>	<b>Excellent (Full Marks)</b>	<b>Good</b>	<b>Satisfactory</b>	<b>Poor</b>	<b>Marks</b>
<b>Voice-Based System Commands (Voice Only)</b>	3 unique system commands accurately triggered via voice with clear VFX/SFX	2 working voice commands, 1 partially functional	Only 1 working command or unstable voice control	Commands missing or not voice-controlled	20
<b>Voice Command Integration</b>	Voice system is stable, responsive, and provides clear feedback & captions	Minor delays or detection issues	Frequently unresponsive or inconsistent	Not implemented or unusable	12
<b>Environmental Reaction + Emergency Challenge Area</b>	Strong sci-fi immersion with dynamic station reactions and a fully playable timed challenge	Good immersion with a working challenge and minor issues	Basic visuals/reactions with a weak or non-challenging sequence	Poor or missing immersion and challenge	8
<b>Part B Total</b>					<b>40</b>

## **Plagiarism / Originality Verification (Viva)**