

## CS-444 Project Description

This project requires a team of 3 to 4 students to collaborate on developing a fully playable and stable VR game.

The game concept is open-ended, as long as it respects the outlined requirements.

### Requirements

#### Tools

You have to:

- use the Unity game engine
- submit a `.apk` file compatible with one of the Quest, Quest 2 or Quest 3 devices.

Other technical choices are up to you. In particular:

- You can use your own headset
- You can use `Meta XR SDK` instead of the `XRIT`

However, we will only provide support for recommended tools and headsets.

#### Code

Your code must be extensively documented. In particular:

- Any downloaded, copied, or adapted scripts/assets from external sources must be clearly sourced.
- An external resource without significant modification cannot satisfy mandatory feature requirements.

You must clearly distinguish:

- Original code: Produced entirely from scratch.
- Adapted code: Complemented or modified from an external resource.
- Unmodified code: Used as-is without any changes.

#### XRIT features

Your game must incorporate the following built-in XRIT features:

- Continuous move
- Teleportation
- Continuous rotation
- Snap rotation
- Climb
- Grab move
- Near interaction
- Far interaction
- Poke interaction

- Socket interaction

Each feature should represent a meaningful part of your gameplay. However, they don't all need to be active simultaneously. For instance, you can allow players to toggle between continuous move and teleportation modes.

**Feedback Requirement:** Include comprehensive feedback (visual, audio, or haptic) for each interaction. Interactions should be intuitive and pleasant for the user.

**Cybersickness Consideration:** Your game needs to consider the cybersickness felt by the player and try reducing it as much as possible. Do not hesitate to pinpoint which elements were used towards this goal.

### Custom features

You must implement **three** additional custom features (or **four** for teams of 4 students). These features must be designed and implemented by your team, meaning merely following online tutorials or downloading scripts/assets without significant modification is not allowed. If you're unsure if your idea is appropriate, do not hesitate to ask the opinion of a staff member.

**Movement Interaction Requirement** At least one custom feature should provide a movement interaction beyond simple joystick navigation (arm movements / positioning, steerwheel, ...)

Here are some examples:

- Flying (arm movements, not joystick only)
- Gliding (arm movements, not joystick only)
- Driving (steerwheel, not joystick only)
- Hand jets (See [Lone Echo](#))
- Grab the environment (See [Lone Echo](#))
- Gorilla walk (See [Gorilla Tag](#))
- Parkour (with hands placement)
- ...

**Non-Movement Interaction Requirement** At least two custom features should offer sufficiently complex non-movement interactions. Here are some examples:

- Sculpting
- Painting
- Magic casting
- Archery
- Fishing
- Musical instruments
- Cooking
- Gardening / Farming

- Puzzle-solving
- Pet care
- Crafting systems
- ...

**Feedback Requirement:** Include comprehensive feedback (visual, audio, or haptic) for each interaction. Interactions should be intuitive and pleasant for the user.

**Cybersickness Consideration:** Your game needs to consider the cybersickness felt by the player and try reducing it as much as possible. Do not hesitate to pinpoint which elements were used towards this goal.

## Submission and Grading

### GitHub repository

On Moodle, submit the link to a **public** GitHub repository containing:

- The complete Unity project, including all source files.
- A `Readme.md` file at the root, detailing:
  - Produced scripts and assets: Created entirely by your team.
  - Adapted scripts/assets: Modified from external sources.
  - Unmodified scripts/assets: Used as-is.
  - Relevant content (scripts, assets, scenes) for each custom feature.

Since the repository needs to be public, remember to license it properly. You can typically use an MIT license (simply copy our **LICENSE** file).

If you wish to keep your repository private, please contact the class staff.

### Build of the game

On Moodle, submit a build of your game as a `.apk` file compatible with Quest, Quest 2, or Quest 3 devices. Name the file as `<team_number>_<device>.apk`, where `<device>` is one of: `quest`, `quest2`, `quest3`.

We will test this build on our own headsets: ensure the game is fully playable without crashes or errors.

### Report

On Moodle, submit a **detailed report** named `<team_number>.pdf`, containing:

- **Synopsis:** A brief summary of the game, including an illustration (e.g., a screenshot).
- **Scenario description:** Concisely outline the game's narrative or progression.
- **Interaction descriptions:** Explain implemented interactions, including:
  - How to trigger them

- Their functionality and feedback
- A brief description of the implementation (diagrams are highly appreciated)
- **Playtesting feedback:** Summarize player feedback and any resulting changes, especially regarding motion sickness or complexity.
- **Focus areas:** Highlight graphical care, sound design, thematic elements, or other crucial development aspects.
- **Team contributions:** Describe each group member's contribution.
- **Acknowledgments:** Optionally mention playtesters, friends, family, or external resources.

**Length requirement:** No more than 7 double-column pages or 10 single-column pages.

### Video

On Moodle, submit a **short video** (under 5 minutes) showing a **complete** playthrough of the game.

- No editing is required.
- Submit in a common format or as a link to a video-sharing platform if the file is too large.
- If you do not want the video shared on public platforms, please notify any staff member

### Group presentation

Each team will present their project in a **10-minute oral** presentation covering:

- Game overview
- Key design elements
- Required interactions and their implementation

Focus on interaction functionality, adherence to project guidelines, and associated feedback (visual, audio, haptic). Live demos are not permitted due to time constraints.

A **5-minute Q&A session** will follow, where staff will ask questions about your presentation and your game

### Individual presentation

Each student will give a brief **5-minute presentation** of their individual contributions. Questions will typically focus on technical aspects, such as code implementation.