Virtual Reality Course Project Guidelines (2024)

This document provides the necessary information for the successful realization of the project for the VR Course. Students are to refer to these guidelines for clarification for how projects will be evaluated, what is expected from the project and deadlines of said project.

Deadlines

1. Delivery Project Proposal **(26.02)**
2. Project Pitch **(18.03)**: 3min followed by 5min Q&A
3. Q&A Sessions during the "Project" slots in the planning (blue color code)
4. Basic Game interactions peer assessment **(15.04)**: ~10 min (to be confirmed)
5. Project Playtesting with peer assessment **(13.05)**: ~10 min (to be confirmed)
6. Delivery of the Game Executable and the Report: **22.05, 23h55min**
7. Project Oral Presentations (twice): **week 27-31.05** ; 15min per group + 5min per person

Project Requirements

All students must abide by the following requirements to successfully complete the project. Please note that **no exceptions** will be made.

1. Projects will be implemented using **Unity 3D**.
2. Projects must use the **Oculus Framework**.
3. Projects must use the **Oculus Quest** using both the standard oculus controllers or/and the Oculus Finger Tracking, **exclusively**. Potential features that may appear over the course of the project will not be accepted. Only the default features of the controllers and finger tracking may be used.
4. Students must **deliver a self-contained build** of their game, **uploaded and running exclusively on the Oculus Quest**. Games running on the Unity 3D editor **will not be accepted**.
5. **The Game must be Stable**: it must be fully playable from start to end **without crashing.**
6. **All projects must include a detailed report** justifying the decisions on gameplay mechanics, implementations, **and how the project fits the requirements.**
7. **All the code and respective methods must be commented** detailing exactly what each method does.
8. Developed games **must not contain references to drugs, sex or explicit violence** (e.g. showing blood and gore)
9. The **games developed must be primarily for entertainment** but can have a serious component to them. **A serious game does mean it has to be a "boring" game.**
10. **Each student must be able to answer all questions relating to their project** including technical, gameplay or otherwise.
11. **All groups must create a video showing a playthrough of the game**, and highlighting the most interesting features of their game, no exceptions.
12. The game should **not include works protected by copyrights** (e.g. copyrighted music)
13. We strongly encourage the use of ready-made assets that can be found in the Unity Asset Store or online. Do not waste valuable time creating 3D models.

Project Description

Students must build a fully playable game with at least 5 types of player interactions (4 for groups of 3 students) and 2 locomotion methods (i.e. player displacement; one must be teleportation). The game itself is freeform, meaning that students may build whatever game they wish, so long as it respects the previously stated requirements and the game is built around **5 virtual interactions (4 for groups of 3 students)** among the following :

- **Throwing**: The player must have the ability to throw certain objects using their hands. Throwing should take into consideration the amount of force used by the player.
- **Filling and Grabbing a Container:** The player must have the ability to place certain objects within another (e.g. such as a bag, jar or basket). Players must then be able to pick up the container with the objects inside and hold on to it (i.e. grab container).
- **Swing and Hit:** Players must have the ability to swing certain objects and have said objects react accordingly when hit. This interaction should take into consideration the amount of force used by the player. Example: Hitting an object with a sword, hitting a ball with a golf club,...
- **Alternate Grab:** Players must be able to grab and hold objects from the environment into their virtual hands. Students must use an alternate method for this interaction than the ones implemented during the TP class. Examples: Magnetic Grab (objects fly towards the players hand), Extended Grab (players may extend their arms to grab objects further away).
- **Joint Interaction:** The game must feature an object with a joint component which the character needs to interact with. Accepted joint types are hinge and spring joints. Example: a door or a chest being opened and closed by grabbing and pulling (without animation), pulling an object attached by a spring/cord (that may break if the force is big enough) which gives haptic feedback as it becomes harder to pull, …
- **Two Hand Interaction:** Players must have an interaction that requires both hands interacting on the same object simultaneously. An interaction that can be performed with both hands separately does not count as a two-hand requirement such as grabbing with either hand. Two separate hand interactions for an action does not count if it is not on the same object. For example, throwing an arrow is not accepted because the bow and the arrow are separate objects. Examples: Pull apart an object by pulling from two sides, hold an object with one hand and rotate with the other, An object that can only be pulled/pushed/grabbed with both hands (balancing a tray),...
- **Shooting System:** Players should physically aim, shoot, and manually reload firearms. This system can be combined with haptic feedback, ammo system (weapon reloading) etc.
- **Gesture-Based Spell Casting:** Players can cast spells or perform special actions by making specific gestures with their hands. The system would recognize gestures, allowing for a

wide range of possible spells or abilities, each with unique effects on the environment or enemies.

- **Precision Crafting:** Players can manipulate small components or materials to create or repair items. For example, assembling a clock or fixing a device. The mechanic would require precise hand movements or could utilise haptic feedback to simulate the tactile sensation of handling different textures and resistances.
- **Puzzle Solving:** Players should engage in physically interactive puzzles, like aligning components or manipulating lights for problem-solving. Puzzle solutions can unlock new areas, reveal story elements, or change the environment.

The game must also include a **Teleportation Method and an Alternate Locomotion method:** Players must be able to move around the virtual environment using a type of displacement mechanism (i.e. Locomotion). **Students must provide a teleportation method AS WELL AS another locomotion method than the ones implemented during the TP class.** Furthermore, for the purposes of this project Jumping/Crouching will not be considered even though it is a type of locomotion.

Students must also implement a **Tutorial System** for their game detailing the mechanics and interactive methods on how to play their game. **This tutorial must be self-sufficient**, meaning that players must be able to understand how the game is played simply based on this tutorial without any external help.

Lastly, students must **build a 3D environment** reflecting the needs of the game and allow players to experience the implemented interactions developed

Project Limitations (What can or cannot be used)

Currently a lot of software exists that helps developers in implementing certain interactions. However, certain limitations are imposed so that students may learn how to implement these features themselves. Here is a list of what can be used and what cannot be used for this project.

Usable

- **Head and Oculus Controller Tracking:** Tracking the Head Mounted Display (HMD) and the Controllers in a 3D environment is the most basic feature of the Oculus Framework, and thus it can be used for this project.
- **3D Models, Animations, Audio and Texture Assets:** For the purposes of creating a nice-looking game art asset such as 3D modelling, textures, audio and animations can be used from external sources such as the Asset Store or others.
- **Interactions Implemented during the VR Course:** Students can use all interactions developed during the TP classes as additional interactions to the ones asked for implementation, if they so wish. Keep in mind that all interactions asked must be substantially different than the ones previously implemented during class.

Non-Usable

Interaction Frameworks: Students may not use popular frameworks such as VRTK or even copy examples out of the Oculus/SteamVR applications. All code relating to both Locomotion, Interactions and the Game itself must be fully implemented by the students. Thus, all students must comment their code detailing their methods correctly.

Evaluation Criteria

Evaluation will take into consideration several aspects considered important for each objective of the project.

Evaluating Interactions

Interactions will be evaluated based on the following 3 criteria:

- **Basic Functionality**: Does it achieve the basic intended action? and how well is it achieved? Students were careful to make the interaction stable.
- **Quality / Usability** : Is the interaction as intuitive as possible within the context of the game? Is the way the interaction is triggered coherent with the other interactions implemented?
- **Feedback** : Is the interaction visually pleasing, is the audio feedback appropriate ? Students used passive/rumbling haptics as a mechanism to improve the interaction and its usability.

Evaluating Locomotion

Locomotions will be evaluated based on the same 3 criteria, with an additional **Motion Sickness Inducing** criteria: Is the motion highly susceptible to motion sickness ? Students were careful to consider the theoretical concepts relevant to motion sickness induction and attempted to develop something around it.

Evaluating the game

Projects will be evaluated by both previous mentioned criteria and by the following 4 criteria:

- **Creativity and Quality of the Game:** How original is the game? (e.g. is it just a copy of an existing game?) How well the interactions are integrated into the game? Is the game fun to play? Is the soundscape appropriate ?
- **Tutorial Quality:** Can the game teach us how to play? How well does the game explain its concepts to the player and how self-sufficient it is without any external help?
- **User Friendly:** How friendly is the game to new players. Is play intuitive? Are the controls easy to use (i.e. not frustrating)?
- **Visual Fidelity / Quality:** How does the game look, and does it have a thematic consistency. Did students take some degree of care about the visuals of the game?

The criteria will be considered based on all material provided by the students including: the final game, the presentation during the group oral exam, the final report and the video (all these elements will have an impact on the project grade). All materials should be used to justify their methodology accordingly and why certain decisions were made. Note that the weight of their justification will be subject to our own evaluation based on experience and theory.

Two oral exams (group and individual)

Group oral exam (15min) The group oral exam will be conducted with all members of the group present during the exam. Students are expected to be capable of answering questions related to all aspects of their projects, including in the source code organization.

Individual oral exam (5min) An independent schedule will be set to meet each member of a group but alone for a 5min presentation of some functionality of the project source code ; the functionality is chosen by the TA: *how is functionality X implemented ? where exactly in the source code ? etc…* The performance of the student during this exam will determine their individual grade.

Project Pitch session

All groups are asked to present (total of 3 minutes) their project pitch to the lecturers and tutors before fully committing to an idea. This allows students to obtain feedback on their idea, its feasibility and if it goes inline with the project guidelines. We strongly recommend that all groups partake in this session as it will provide invaluable information to the group. Furthermore, all groups should have their ideas ready and rehearsed so that it is easily communicated during this session.

Questions and answers

The use of edStem is recommended for asking questions and documenting bugs and solutions. Please keep in mind that we cannot guarantee a fast answer to questions made outside the Monday project Q&A time slots.

Contacts

- Dr. Ronan Boulic (ronan.boulic@epfl.ch) – INJ 141
- Loën Boban (loen.bodan@epfl.ch) – INJ 138
- Fatma-Betül Güres (fatma-betul.gures@epfl.ch) – INF 216