

VRUE 2024

Assignment #4: Final Project

Goals of the task:

Apply learned skills and combine functionality from previous assignments; select and integrate the new features into a VR app on a chosen topic.

Assignment preparations:

- Make sure that the mandatory parts of the previous assignments are working – you will need this functionality again for your game.
- After **reading the full assignment document**, have a brainstorming session together to refine your application idea.

Project file structure in Unity

- Scenes should be in the folder *Assets -> Scenes*
- Scripts should be in the folder *Assets -> Scripts*
- Custom prefabs should be in the folder *Assets -> Resources -> Prefabs*
- Custom materials should be in the folder *Assets -> Resources -> Materials*
- Custom models (.fbx, .obj) should be in the folder *Assets -> Resources -> Models*
- External assets from Unity Asset Store and alike [Assets -> External](#)

Make sure that:

- you are using the correct Unity version (2021.3.44f1 LTS);
- your hierarchy and project files are well organized;
- every script serves a dedicated purpose and can be reused in the future with minimal changes;
- avoid creating a “jack of all trades”;
- your scripts are attached to the game objects they are most related to;
- all builds are really working – test them!

Assignment Description:

Implement a small multi-player, preferably collaborative, VR application for 2+ players (2 main actors and others – only observers). Tip: you can test extra players with running an extra build instance.

It is up to you to define the topic/genre of your application – it may also be a simulator, educational use-case, or anything else with a **purpose**. You should aim to design interactions and choose the features that “make sense” for your specific application.

The following are the mandatory requirements.

Multi-user	Support 2+ users. The application task or scenario can be limited to two users. The other users will have to be “ghost” observers (with limited functionality, e.g. with ability to navigate, but not interact, semi-transparent avatars). On disconnection of the main user, one of the observers might get that role (optional).
Users’ avatars	Each user should have an avatar: some sort of a networked representation of a head and hands, torso is optional.
Environment: Skybox, Terrain/Indoor scene/Water surface	The surroundings of the users should be represented with a complete and properly designed environment: no visible “end of the world” (unless required by concept) and other blank spaces.
Lighting	The scene and all objects in it should be properly lit and visible, unless otherwise required by the concept.

Basic UI	Provide the users with a basic UI and enable them to: connect, disconnect, reset, and exit. Feel free to add extra features if necessary.
Clean networking	All connects/disconnects should be handled properly and leave no garbage in the scene.
Minimum feature set	Choose a number of features listed below. Each feature has a weight associated with it. If the feature's weight is set as a range – you can clarify the final value after the Milestone submission. The goal is to have the cumulative weight of all features equal 3 or more. You cannot choose two features that belongs to the same category.

Feature List

Category	Feature	Weight
World builder tweaks		
	Change skybox & light parameters with UI (at least 3 different changes)	1
	Create & modify objects with UI (at least 3 different objects)	1
	Group operations with objects (at least 3 operations)	1
Avatar		
	Customize VR avatar runtime: choose hand, leg, hair...	2
	Adjust VR avatar to manually adjust to the user (rescale individual body parts)	2-3
	Hand tracking with Leap Motion or Oculus hand tracking	2
	Hand-controller animation beyond standard buttons	1
	Inventory system	0.5
Locomotion		
	Locomotion metaphor with a helper object (moving parts should be moving somehow): car, scooter, magic carpet...	2
	Locomotion metaphor without a helper object, bound to movement (flying, climbing) should not match Assignment 3	1-2
	Teleportation	0.5
Communication		
	Voice chat	1.5
	Text chat between users (e.g. simplified UI - text like on a mobile or alike)	2
	Sign exchange for communication (like smiles 😊)	1

Interaction		
	Gesture recognition with hands or controllers (min. 3 gestures)	2
	Haptic prop with real tracked object (e.g. using a vive tracker)	2
	Complex interaction (two-handed = 1, multi-component /multi-tool [min 3 components/tools with different ways to interact with player/environment] = 2)	1-2
	Networked object that can be passed hand to hand. A falling object should fall correctly for all users	1
	2-Player simultaneous collaborative interaction with environment (two players should do something simultaneously)	1
Effects		
	Particle systems	0.5
	Spatial audio (background+ a couple of sound effects = 0.5, more complex with multiple sounds in 3D space = 1)	0.5-1
	Haptics with vibration (notifications = 0.5, more complex use like material simulation etc. = 1)	0.5-1
	Camera manipulations (FOV, pose manipulations, spectator view...)	0.5
	Mini-map of the virtual environment (should show your current location)	0.5
	Complex physics interaction or simulation (similar to assignment #2 = 1 point), or advanced character physics (physics based puppet =2)	1-2
	Objects or NPCs animation (applying an existing animation, trigger it and stop without artifacts = 1, add smooth transitions between animations = 2)	1-2
	Custom self-implemented shaders (1 complex or at least 3 simple ones), shader graph is allowed	1

If your chosen feature has a weight set as range, its final weight will be defined based on complexity of your usage. The complexity depends on the following criteria: 1) difficulty of application, 2) variety of application cases, 3) additional code needed, 4) difference from default configuration, 5) creativity of application. Complex and self-implemented features will have a maximum weight. 1 weight points = 15 points, maximum you can get is 45 points (3 weighted points).

Submissions:

For the final project you will need to make the following **two submissions** by the respective deadlines.

1. Early prototype video (Milestone): Deadline **Thu. 8.12.2024** – upload as .zip to TUWEL.

- Create a brief video (2 minutes max) that demonstrates the functionality of your early prototype ("**Group#_Milestone_Video.mp4**"). At this point you should be able to show the implementation of the mandatory requirements and at least one chosen feature. Include your game title and group number.
- A refined version of the concept idea with a list the features that you have chosen to implement ("**Group#_Milestone_ExtConcept.txt**"). Mark the current state of the features (0%-100% ready) and expected weight points to confirm them (add details to define the complexity). Mind that the cumulative weight of all features should be ≥ 3 to be able to score a full grade for the project.

2. Game project + video: Deadline **Sun. 15.12.2024**

- a. Create an archive (.zip) with the name "*Group#_FinalProject.zip*" containing
 - your **complete** Unity project; do not remove any folders. **In the project, configure PhotonServerSettings to use your AppID;**
 - **build(s) of your Unity project for the Vive & Quest users (.exe & .apk).**

Due to the file size of submission, upload your project to your Google Drive, Onedrive, or OwnCloud and submit a public link to this project to [TUWEL](#). Make sure that the visibility of the file is set to anyone with the link can view. Please do not use other file sharing platforms.

- a. Create a short video (2 minutes) demonstrating your gameplay and the integration of the chosen features ("*Group#_FinalProject_Video.mp4*"). Make sure to include your game title and group number and upload this to [TUWEL](#).
- b. Extend the concept idea with details of your implementation: briefly describe the goals, interaction, and implemented features ("*Group#_FinalProject_Description.txt*"). Please, specify if you used the ready assets or implemented something yourself.
- c. Write a brief description of your implemented *bonus* in a .txt file ("*Group#_FinalProject_Bonus.txt*"). Upload this to [TUWEL](#).