SCENE LOADER

Testproject

As a developer at Innerspace you get the responsibility to implement the scene-loader for the VR-Trainings. The product owner crafted the following user stories for this project.

USER STORIES

As a user, I want to see a loading screen, while the main-scene loads, so that I get an indicator that the program is not stuck.

As a user, I want to get one randomized hint/tip/quote about VR-Trainings in general (effectivity, etc.), while the main-scene loads so that I can learn something, while waiting. (be creative, with finding useful quotes)

As a user, I want to see some kind of progress while waiting, so that I can guess approximately how long it could take, until I can start the training.

As a user, I want to have a smooth transition between loading screen and the scene itself. (be creative, with the transition… but keep in mind you build it for VR-Users)

As a product owner, I want to change the hints/tips/quotes outside of unity, so that I don't need to have coding-skills.

BONUS:

As a VR-User, I want to have a loading-screen, which follows the movement of the headset with a slight delay, as static HUDs are not very comfortable. Robo-Recall does this very good in this sample at minute 7:00 https://youtu.be/Tjzcuz7PN5k?t=420

IMPLEMENTATION

- Setup a private or public Github Project, where you give access to the following user: https://github.com/bfercher before starting with the implementation
- Develop as you would in a team (feature branches, commit often, etc.)
- Setup a Unity project with the latest stable Unity version
- Simulate/build a heavy-weight main-scene to see a loading progress (third party assets are totally fine)

- The canvas with the loading-screen must be a world canvas, to use it with VR
    - Take care of setting up the canvas/distance correctly
    - More information here: https://www.ryanhinojosa.com/2018/01/07/device-independent/
- Make it reusable for various projects
- Do not use third party libraries (from github/asset store etc.)
- Write automated tests (Unit and/or Integration tests)
- Licensing: https://unlicense.org

DELIVERY

- Before implementation
    - Time estimation before starting the project for each user story
    - Send time estimates to bernhard.fercher@innerspace.at
    - Optional skype-call for requirements clarification
- Implementation
    - Functional build for x64 machines
    - Access to Github Project
- After implementation
    - Time documentation for each user story
    - Review for the project
    - Send report to bfercher

EVALUATION CRITERIA:

- Structure of project/hierarchy and code
- How and which software patterns you use
- Maintainability and cleanliness of your code
- How you test your code
- Creativity
- Development time (effectivity)

Questions at any time to bernhard.fercher@innerspace.at