SW Engineering CSC648/848

Mesekai

real-time full-body 3D avatar and virtual room web application

Section 04 Team 04

Caelen Wang (Team Lead)

Vasudevan Venugopal (Frontend Lead)

Matthew Madore (Backend Lead)

Jose Miguel Atienza (Scrum Master)

Eugene San Juan (Git Master)

Mohammad Abdelrahman

"Milestone 3"

November 8, 2021

Revision History Table

Appendix 1

UI and Functionality Feedback
Instructor's comments on functionality for your demo:

The functionality and technical sophistication is adequate. The professor's main suggestion was to consider specific customer needs and use cases for our product. We are taking a laissez-faire approach by simply creating a customizable sandbox for Users to choose from Avatars, Worlds, and Objects. The use case will be up to them. In its current single player form, example use cases would be a virtual meeting through screen share (like the live demo in class), and Twitch streaming using an avatar instead of actual face. For a future multiplayer version, Users can host a gathering in a shared virtual space built by their own creativity.

Instructor's comments on UI:

The UI is nice and smooth. (since we were running out of time in class we did not get much feedback on it)

Brief Review of Coding, GitHub, Database

Branch policy:

The master branch is for production. Any changes to master should be ready for deployment to the live site. During the initial phase of testing the software stack and building the vertical prototype, we used the following development branches:

- ui - used to develop frontend user interface and styling
- backend - used to develop APIs for connection to database
- avatar - used to develop pose detection and avatar tracking

The “ui” and “backend” branches have since been merged with master and are removed, but they can be found in the repository history. The “avatar” branch remains to serve as an experimental playground for quick testing, without having to worry about frontend/backend interactions. At the completion of the vertical prototype, we successfully established frontend and backend connection using Next.js. Future development will take place on the branches:

- next - used for frontend styling, backend APIs, and User authentication
- three - used for Avatar, World, Objects, and Interface

Code review policy:

Each member is responsible for making their own features production ready. Those comfortable are encouraged to directly push to master or make pull requests from their feature branch. Any concerns are first discussed with their subteam, then with the Git Master to facilitate the merge. The Team Lead will supervise all changes to the master branch. To ensure proper deployment, we test that all features work locally before pushing to master. Additionally, pull requests to master from other branches should also reflect production ready code. Vercel provides an automatic build check on all branches, so before merging to master we should observe that all checks have passed, signaled by a green arrow next to the most recent commit.

Project Status

Teamwork:

The team is working well together. We have separated into separate groups in alignment with our GitHub branches to develop different elements of the app. Each day we let each other know in our Discord server how our progress is going and give each other updates. These updates consist of what we have accomplished and problems we have run into. Our team lead reviews these updates and helps us with any problems we have.

On Mondays we decide what we need done by the end of the week and then we meet again on Thursdays to show what we have done so far that week. On Thursdays we fix the issues we ran into during the week and work on Milestone work. Sometimes, when we have a lot to do that week, group members meet in between the regularly scheduled meetings to work on their assignment for that week to make sure everything is running smoothly and on time. Every group member has been attending each meeting for the most part. If someone cannot make a meeting, has to leave early, or come late, they inform the group or the scrum so that there are no delays in the team meeting. Those who couldn't make it to a meeting or had to leave early their group members will inform them on what they missed to keep them up to speed.

Risks:

- Teams not being able to complete their weekly assignment on time. We can resolve this through scheduling an additional meeting and solving any issues within that time.
- Running into refactoring problems when integrating the different parts of the app. As a group we can resolve this through looking through whatever documentation we need to read through and brainstorm together to fix those issues.
- If a certain part of the app is out of place or not working correctly then we will take another approach. Perhaps a simpler way but a less efficient, whichever works.
- New frameworks that we use may break some of the things we already have in place. To fix that we must do refactoring until things are running smoothly.
- Visual and physical fidelity of the result might be computationally expensive. We will optimize as much as possible and provide settings for users to adjust based on their hardware.
- Avatar tracking is inaccurate and gets stuck in the "uncanny valley". We will consult research papers and other work for better approaches.

List of P1 Features Committed for Delivery

ID	Functional Requirement	Details
1	Account creation	Store User provided information such as username, email, and password in database
2	Log in and log out of account	Log in to account using credentials stored in database provided during account creation or password reset
3	Reset password to account	Link on login page that lets User enter their email address, to which a password reset link will be sent
4	Select Avatar from presets	User chooses from list of available Avatars from Interface: ● Remy ● Stefani ● YBot ● … choice stored in Account
5	Select World from presets	User chooses from list of available Worlds from Interface: ● Room ● Forest ● Castle ● … choice stored in Account
6	Select Objects from list	Based on User choice from Interface, different Objects will be spawned in ● Chair ● Bed ● Desk ● Sword ● … choices stored in Account
7	Avatar tracking	User's body controls Avatar's face, limbs, and fingers
8	Avatar / Object Interaction	● Avatar pushes Object through collision ● Avatar picks up Object using fingers ● Object physics
9	Transform Objects around World through Interface	Click, drag, drop Objects to be placed in desired location. Translation, rotation, scale stored in Account