Peak Visualization First Iteration

Tyler Newlin, Connor Pflederer, Dan Chepkwony, Cody Lockridge

Client Information

Alicia Mckoy - Peak Mind
- alicia@joinpeakmind.com

Mentor Feedback

- Feedback on getting more feedback and requirements from the client.
  - Mostly useful for the collecting requirements for the Second iteration.

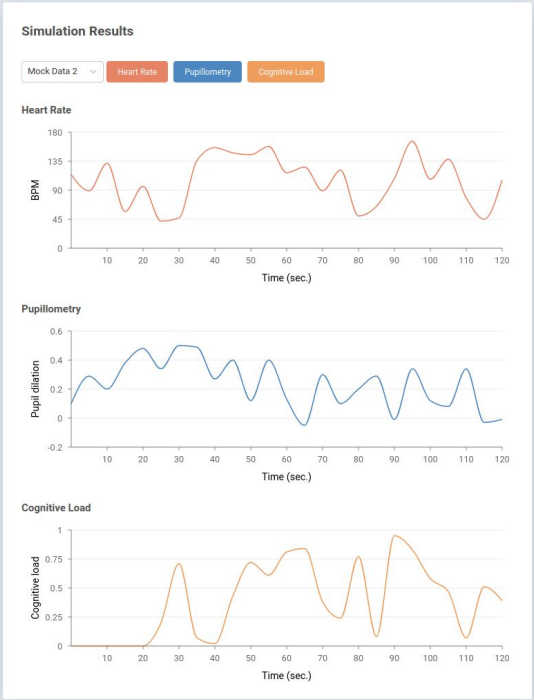
- Better testing practices, specifically for the webapp.

- Idiomatic code examples for C# for Unity application.

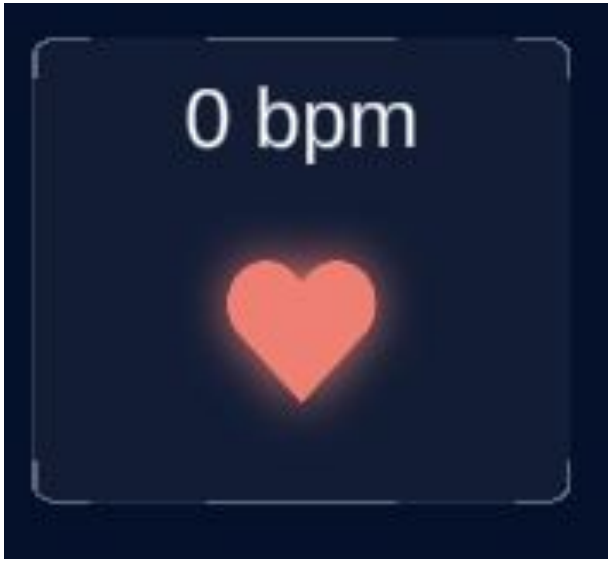
Client Feedback

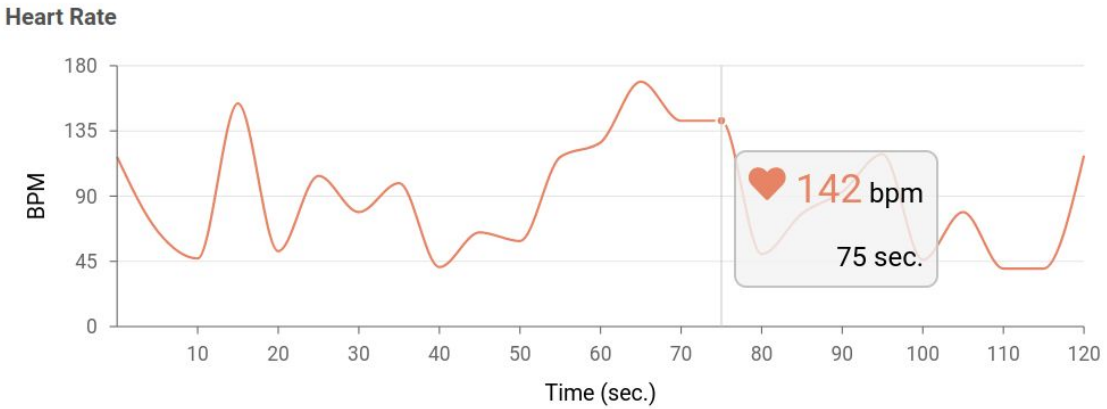
- To add some sort of peak visualization branding to the webapp.
- Adding some spacing to the webapp for possible additions of representing parts of the simulations the data is relating to.
- Adding more animations to the Unity application.
- Some form of input field is needed for user id and numbers for the Unity app to relate the collected data set to them.

Webapp Dashboard

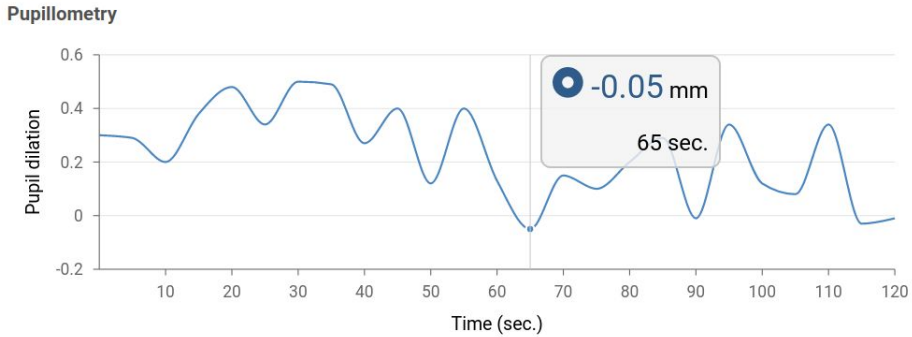


Visualization of Heart Rate data

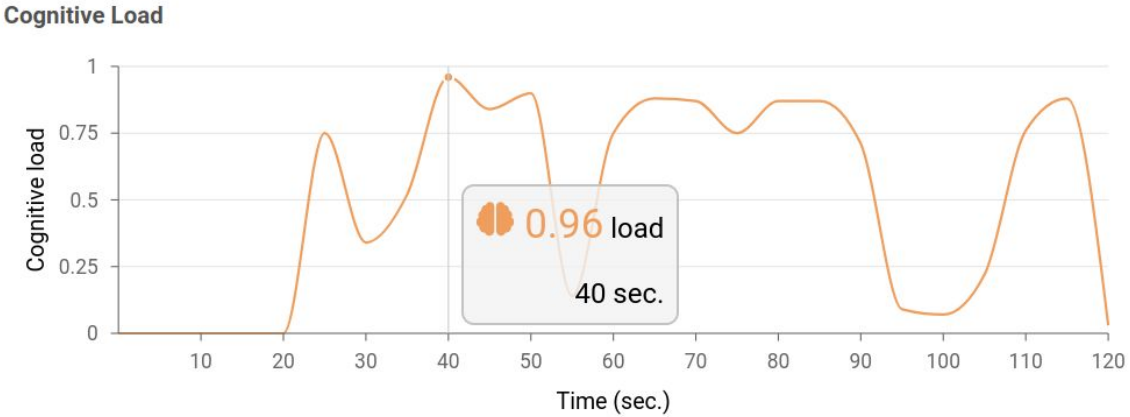




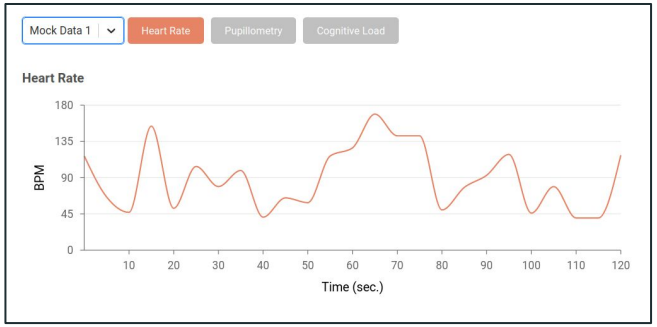
Visualization Of Pupillometry

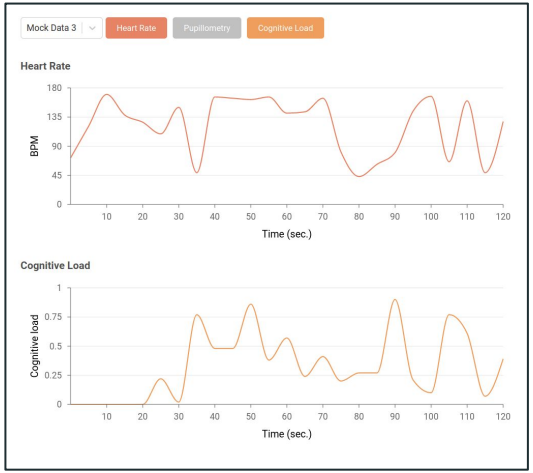


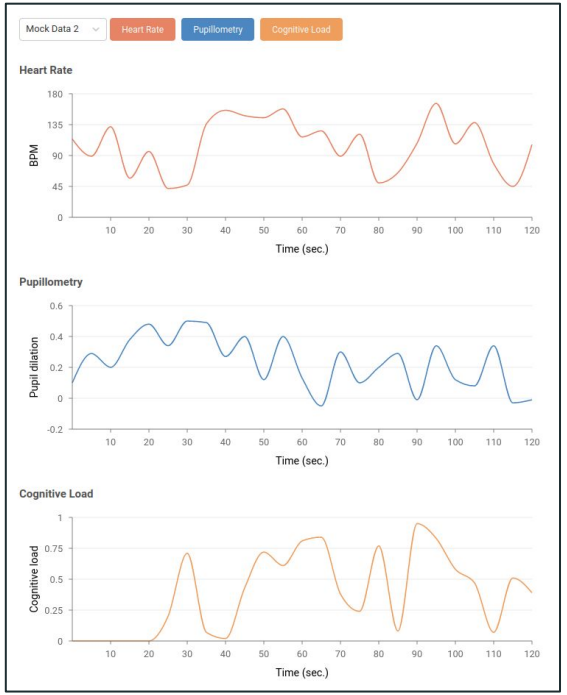
Visualization of Cognitive Load



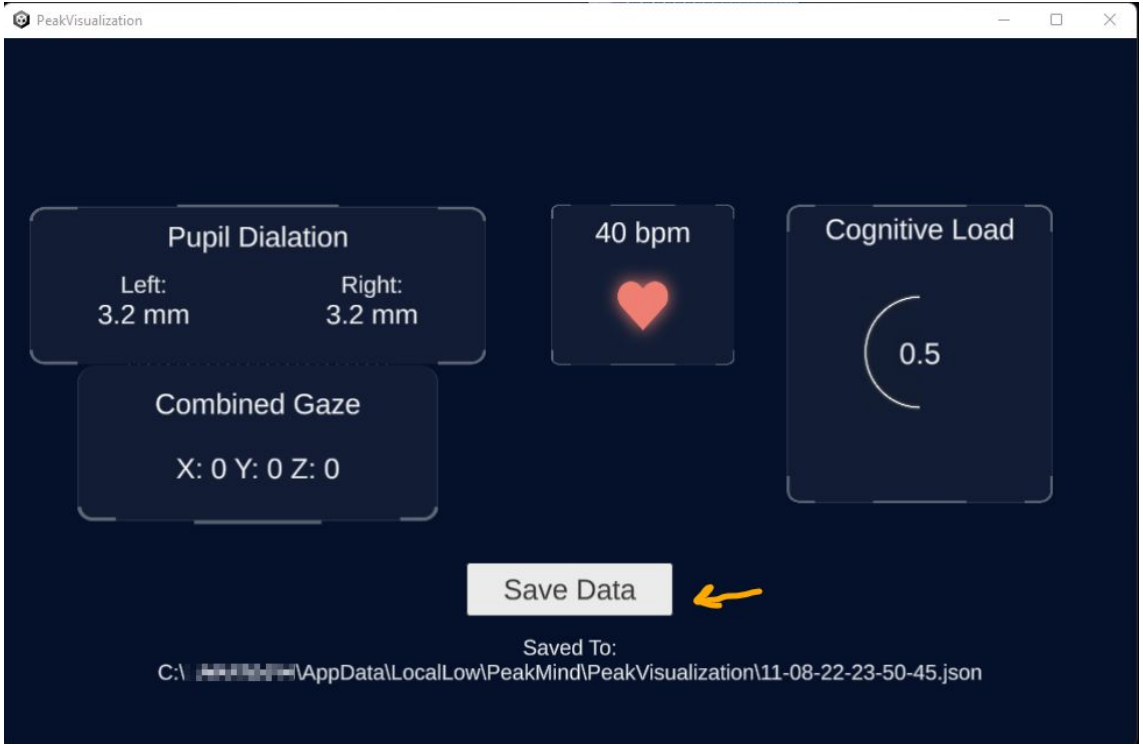
Selecting Which Data is Being viewed

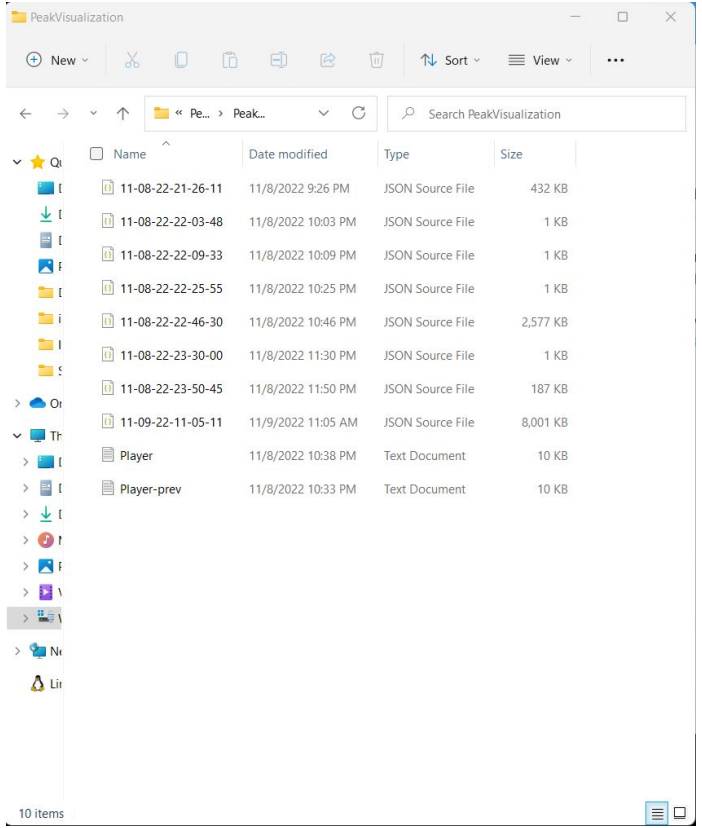






Saving Omnicept Data in JSON format





Planned Second Iteration Features

- Adding a baseline to the graphs in the WebApp based on data provided by Peak Mind to compare to the data collected during the vr simulation.
- Integrating with the Peak Mind's AWS platform to upload data sets from Unity application and get data sets to visualize in WebApp
- 2d animation to roughly display the combined gaze of a user in Unity app.
- Circular animation that resize to show the pupil dilation of a user in the Unity app.
- Adding effects to represent spikes and lows in data variation, focused mostly on spikes.

C# Reference

Summary

Members

`namespace HP::Glia::Messaging::Protobuf`

`namespace HP::Omnicept`

`namespace HP::Omnicept::Errors`

`namespace HP::Omnicept::Messaging`

`namespace HP::Omnicept::Messaging::Messages`

`namespace HP::Omnicept::Messaging::Protobuf`

`namespace HP::Omnicept::Messaging::Zmq`

`class HP::Omnicept::Messaging::Channel::ChannelBuilder`

`class HP::Omnicept::Glia::GliaRegistry`

`class HP::Omnicept::Messaging::Protobuf::proto_SessionControlSignal::Types::SessionLicense`

`class HP::Omnicept::Messaging::Protobuf::proto_ConnectionStatusSignal::Types`

`class HP::Omnicept::Messaging::Protobuf::proto_BiovaultAction::Types`

HP's documentation is the joke.