CS CAPSTONE EXECUTIVE SUMMARY

FEBRUARY 28, 2020

DESIGN AND IMPLEMENTATION OF A FRAMEWORK FOR BIO-INFORMED 3D USER INTERACTION

PREPARED FOR

OREGON STATE UNIVERSITY COLLEGE OF ENGINEERING

RAFFAELE DE AMICIS ______ Signature ______ Date

PREPARED BY

GROUP 51

BIOMR

LEY ALDINGER ______ Signature ______ Date

AYUSH CHOUDHURY ______ Signature ______ Date

KYLE HIEBEL ______ Signature ______ Date

ZUNYUE QIU ______ Signature ______ Date

Abstract

VR is gaining new applications, especially in manufacturing. With people spending a long time in VR, our customer wants to study how a VR environment can affect a user's time perception. To do this, we were tasked with designing a system architecture which can modify a VR scene based on biometric sensor data. This data includes eye tracking, skin conductance, and brain activity. It is thought that data from these sensors quantitatively correlates with time perception. Creating this system architecture will allow our client to design a study where they can manipulate parameters in the VR scene, and measure the resulting time perception.

CONTENTS

1 PURPOSE

Over the last few years, virtual reality (VR) and other immersive technologies have rapidly grown in popularity. A new usage for this technology is emerging in manufacturing shops throughout America. Smart machines, with enhanced visualization capabilities, allow for remote operation through a virtual interface. Workers using these new interfaces can expect to be in a virtual environment for up to 8 hours a day. Due to this large amount of time spent in VR, our customer would like to study the effects of certain environmental conditions on VR users, and use this information to improve VR working environments.

To gain insight on how to improve VR work environments, our customer wants to study time perception in VR. Particularly, he wants to study how lighting, weather, movement, and sound can impact an individual's time perception. The customer intends to measure time perception using biometric sensors such as eye tracking, galvanic skin response, and electrocardiogram. The insight gained will be used to optimize VR work environments to improve development costs, time-to-market, employee safety, and general efficiency for an industrialized company.

2 SUMMARY OF OUR PROJECT

To allow a researcher to gain insight on how environmental factors affect a VR user's perception of time, we will develop a system architecture in which a researcher can perform such a study. The researcher will interact with our system, named BioMR, through a custom API which will serve as middleware between biometric sensors and a VR scene. A subject will interact with the VR environment through a HTC Vive headset while biometric sensors will quantitatively measure the subject's perception of time.

The API will provide the researcher a user interface to control and monitor the study. Using the interface, the researcher can modify effects in the game engine using value sliders. Additionally, the researcher can view plotted biometric sensor data in an existing application, iMotions. The BioMR API will connect to iMotions to read the plotted biometric sensor data. The researcher can set triggers in the BioMR API which will automatically change the VR scene based on a sensor level. For example, if the subject's heart rate goes higher than 120 beats per minute, the VR scene can automatically become darker to calm down the user.

Additionally, we will create two test scenes, one in Unreal Engine 4 (UE4) and another in Unity to validate that the system architecture is collecting meaningful data. Each of these scenes will contain controllable lighting effects, weather effects, and dynamic objects which can be modified by the researcher. Testing with two different game engines ensures that the API will be portable to any future game engine–a requirement of the client.

3 OVERVIEW OF DESIGN

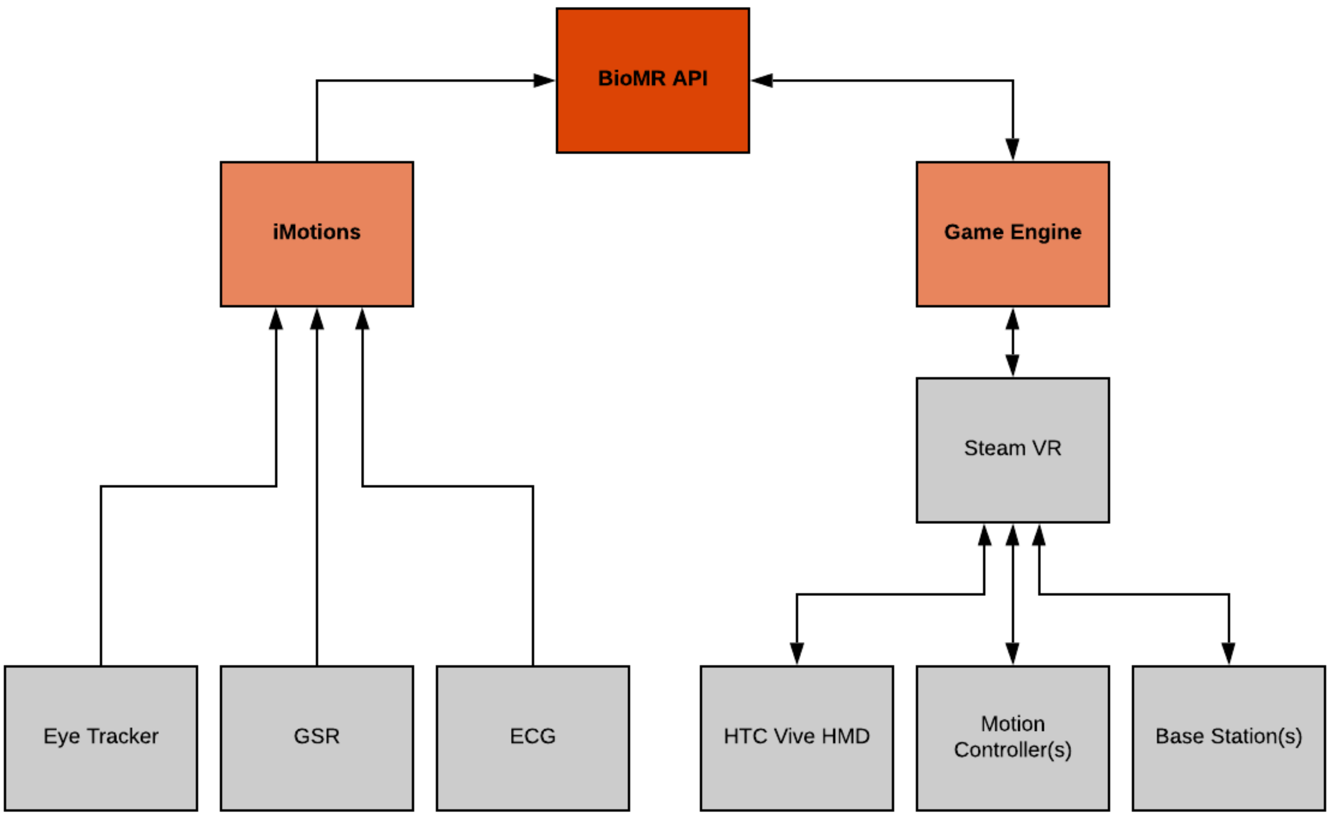

Fig. 1: Dataflow of our system architecture. The sensors send biometric data from the subject to an existing application, called iMotions. iMotions stores the data, then sends a copy to the BioMR API. Our API reads the data and determines if any parameters need to be changed in the game engine. If so, the BioMR API sends a signal to the game engine (Unity or Unreal Engine 4) which tells it to make the change. The game engine communicates with the HTC Vive through steamVR and responds to user input.

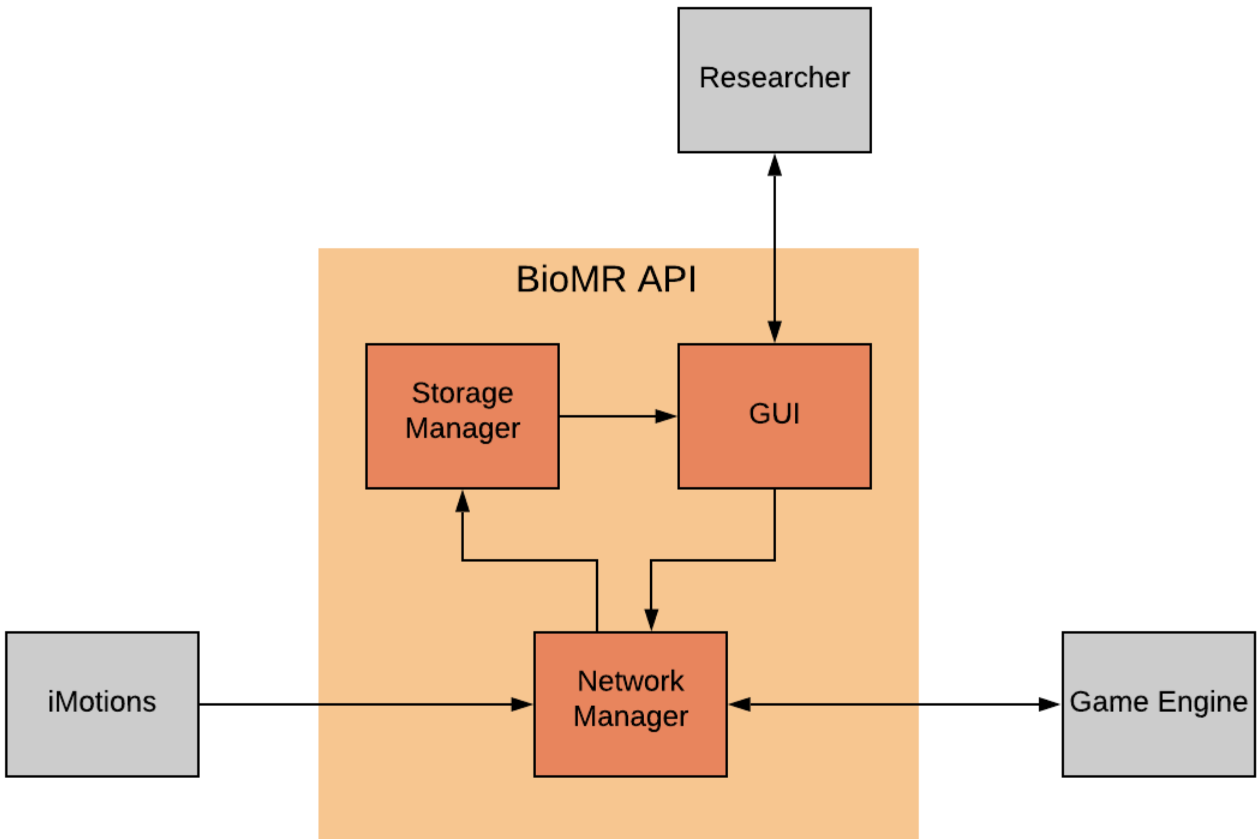

Fig. 2: Design of the BioMR API. iMotions and the game engine communicate with a special module called the network manager. This module consists of 2 UDP senders and 2 UDP receivers. The network manager stores all communications in short term memory in the storage manager. The storage manager updates the GUI based on incoming data. The GUI can automatically trigger a change in the game engine, or respond to researcher manual input. This command is sent to the network manager where it will be sent out over sockets.

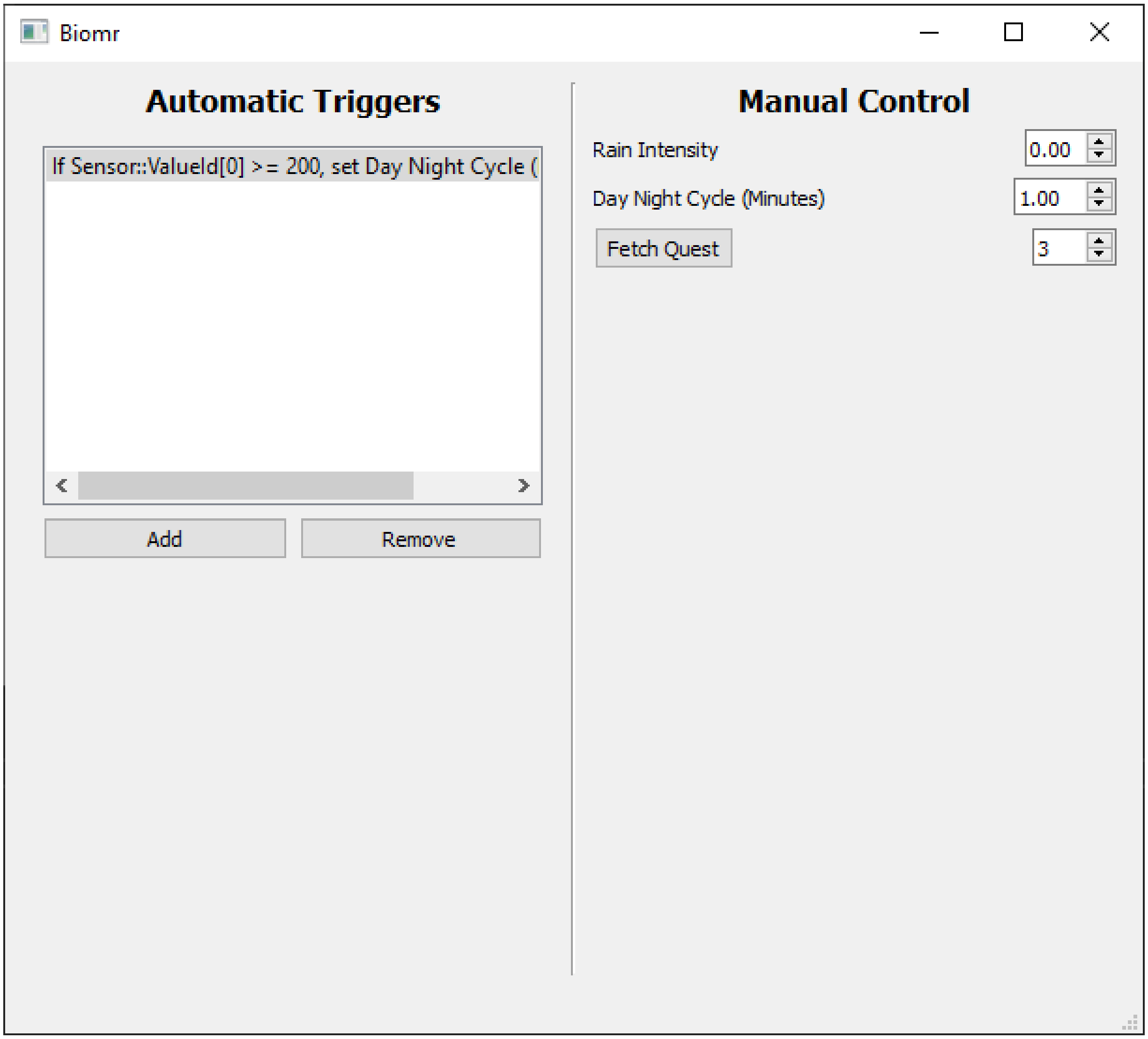

Fig. 3: BioMR API layout. The left side allows the researcher to set automatic triggers. See 4. The right side allows the researcher to manually change parameters in the game engine. For example, the rain intensity can be modified in real time by simply changing the value in the spinbox.

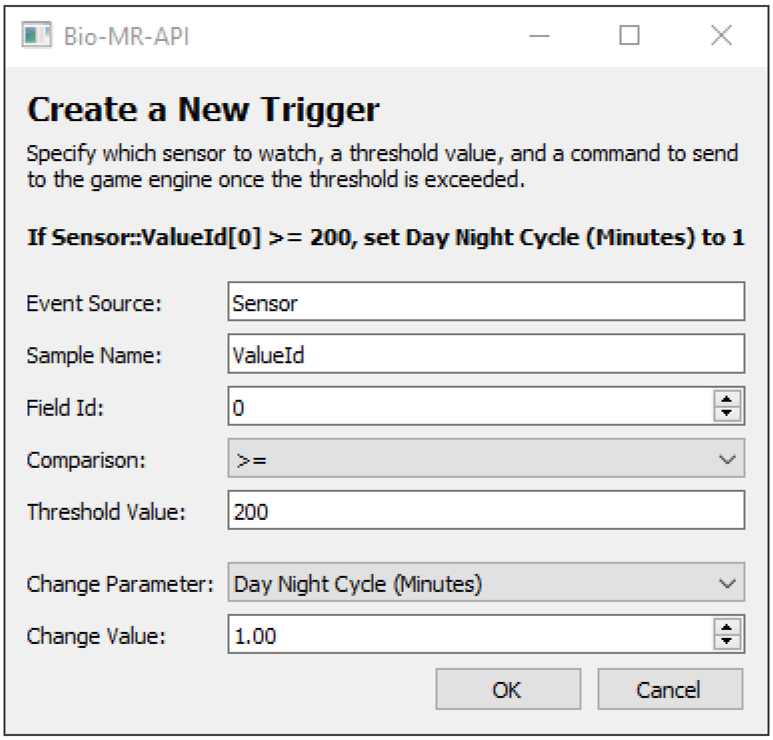

Fig. 4: This widget allows a researcher to set a new trigger. The format is a little strange because it is based on the iMotions data format. The event source indicates the name of the sensor (i.e. eyeTracker), the sample name is the name of the particular data (i.e. gaze length), and the field ID indicates which value in the raw data to compare to. This alpha version is difficult to use and convoluted. The researcher can select a comparison function, and a value to compare to. Once the biometric data passes this threshold, the "Change Parameter" will be set to "Change Value".