Evaluating Different Techniques for

Improving Computer Games using Affective Computing

Aidan Eyre

1304010

Computer Games Technology, 2018

School of Design and Informatics

Abertay University

Table of Contents

1. Abstract

Context: Affective computing has been researched for several years but has not seen wide use in computer games where it could have the potential to vastly improve a game by making it more immersive, interesting and give them much more impact.

Aim: To develop and evaluate different techniques of measuring player's emotional level using different hardware and software that would be available to the average gamer. To also see if these techniques could be used together in order to build a more accurate affective map of how a player is feeling.

Method: Research several pieces of technology to see what emotional data could be taken from them and then create applications that shall gather this data. The applications then shall be tested alongside each other to track a player while they play a game. The applications will gather data that shall then be analysed to see how to body reacts to different emotions and compare the different inputs to see how they can be used together.

Results: The results from this project have shown that it is possible to develop techniques for gathering affective data using technology that is available to the average gamer. Test data and questionnaire answers have shown that slight changes in the body can be seen by the input devices depending on the emotional state of the player. Questionnaire answers also shown that adding an affective element to a game could make it more interesting.

Conclusions: While it is possible to track affective data with these techniques, more testing is required on a wider test group and with more effective ways of eliciting certain emotions in order to determine if these techniques could be used to support each other and create a more accurate affective map. Some techniques also require more development in order to be better integrated with each other.

2. Introduction

The ability for a computer game or piece of software to read, recognise and interpret a human's emotions, mood or personality is often described under the term *Affective Computing* (Picard, 2000). While affective computing has been researched for many years, it has not been researched much for use in the games industry as it often requires special equipment which would make it difficult to have them implemented in any fashion into a computer game. The use of affective computing has great potential in computer games to aid the player's immersion and make a game feel more exciting. For example, a game featuring an non playable character (NPC) or artificial intelligence (AI) that has the ability to smile or laugh back at a player's real life actions is much more interesting than an NPC that would just follow a player around (Hamdy and King, 2017).

In modern day computer games, artificial intelligence has become a very important part of a game. It can be used to add to the story telling or environment of a game through different methods such as co-op companions or intelligent environmental systems which can change part of the in game world such as light or weather. If a game can react to the emotions of a player such as having an in-game character show sympathy when the player is feeling sad or nervous, or have an environment become lighter when the player is feeling anxious or excited, It could make a game massively more engaging and immersive. It could also be used to affect the music within the game, changing the tempo or volume depending on how the player is feeling. This will also help a computer game or artificial intelligence feel more realistic, which is the ultimate aim of an AI.

Only a few game studios have looked into the idea of making use of affective computing or taking biometric data from a player while they play a game. One of the few games that has implemented and researched affective computing is the horror game Nevermind (Reynolds, 2013) which modifies different aspects of the game by taking the heart rate of the person playing the game. Using the player's heart rate, the game can change the music or lighting depending on if the player's heartrate is too high, indicating that they may be anxious or nervous. The game collects the user's heart rate data through a watch that contains a heart rate sensor.

The most important part of affective computing is reading a user's physical data to try to calculate their emotional state while they are using the software or playing a game. There have been many different techniques that have been developed over time to try and track someone's emotional state. The issue with many of these techniques is that they often require specialist equipment, this means that they can be potentially expensive, bulky or require specialist training to operate. These sorts of methods would not be suitable for a gaming scenario as big, distracting equipment would take away from the fun and enjoyment of a computer game. An example of this type of method is through tracking how much a person is sweating using a galvanic skin response sensor (GSR). The technology would be unrealistic for use in games as it would require the user acquiring equipment that is generally not easily available and it could potentially be disruptive if the player is using a controller as the sensor is usually applied to the arm or hand.

The main goal of this project is to evaluate different techniques and technologies to see if it is possible to accurately measure a player's affective state while they play a game. The technology that is being used must be suitable for the average gamer and be a piece of technology that the player is likely to own. The affective data will be collected from multiple sources that will be used together in order to collect a more accurate picture of how the player is feeling. The technologies that have been investigated for this project are a standard games console controller which shall be used to time the player's inputs. A smart watch to track a player's heart rate as most new smart watches feature a built in heart rate monitor. The final technology that has been used is facial recognition using a standard webcam that is used to read a player's emotion through their facial expressions.

<u>The key aims of this project are:</u>

- To develop and evaluate 3 different techniques for gathering affective data:
    1. Controller input
    2. Heart rate
    3. Facial Expressions
- To develop an application using the Unity game engine to bring the inputs together and potentially fuse them into a single input.

- Conduct a test using the application where participants shall play a game in order to gather test data and analyse all 3 inputs.
- To make recommendations on how the final inputs could be used together in a game to help improve the immersion, interest and reality of a game.

Chapter 3 will look at the literature and research that was examined and evaluated on each of the three technologies that were used, controller, heart rate and facial recognition. This chapter will also show how this research helped back up some of the ideas and planning that went into this project. Chapter 4 shall look at the process of how each technology was implemented individually and then brought together in the Unity engine. It will also focus on some of the technical challenges faced during the development of the application. Chapter 5 shall look at how testing was carried out for the project, the results from each data source and the results from the final testing conducted using play tests and the final application. Chapter 6 will be a discussion about the final results from testing, answers found on the questionnaire the participants were asked to complete and a discussion on the project as a whole. Finally chapter 7 shall discuss some of the conclusions that can be drawn from the project and how the project may be taken forward.

3. Literature Review

This chapter is a record of the research that has been carried out for this project. Many of the decisions that were made for this project are a result of some of the research into past studies and experiments. While research on affective computing has been carried out in the past to determine if a computer can detect and react to a user's emotions, it has not seen much research when applied to the games industry. This section shall be split into three parts, one for each of the chosen technologies for this project, a games console controller, a heart rate sensor in a smart watch, and facial recognition system using a camera to read facial expressions.

3.1 Controller

Computer games controllers have been a key input for games since the very early days of games consoles. Controllers have remained one of the most popular inputs for games, with games consoles taking up 41% of the overall games market (Newzoo, no date) and the vast majority of these consoles requiring a controller to operate. This means that a good share of gamers own at least one games controller. Some of the popular console controllers have also been made available for use on PC which means that many PC gamers also use these controllers as a method of input, increasing the overall share of gamers owning these controllers. Over the years, controllers have advanced greatly in terms of design, sensitivity of the controls and the amount of readings that can be taken from them (such as motion control using accelerometers or IR sensors). While previous research has been carried out to see if input devices such as mouse and keyboards can be used to detect a user's affective state, very little research has been carried out to see games console controllers could be used.

One previous test using standard console controllers was carried in 2003. The aim of the test was to see if controllers could be used to calculate if a player was feeling frustrated, annoyed or aroused while playing a game (Brown and Sykes, 2003). The research proposed that if a player was feeling more frustrated or engaged with the game they were playing, they would apply more pressure when pressing the buttons on the gamepad. During the test, participants were asked to play three levels of the classic game 'Space Invaders' while using a Playstation 2 controller. Each level of the game would be more difficult and intense than the last. While the players were playing the game, the inputs on the controller were recorded and compared across each difficulty level. The results of the test showed that between the easy and medium difficulties, there was very little difference. However between the medium and hard difficulties, the user was putting more pressure on the buttons as the game became more intense and more difficult. This shows that a gamepad can be used in some limited capacity to try and track the affective state of a player.

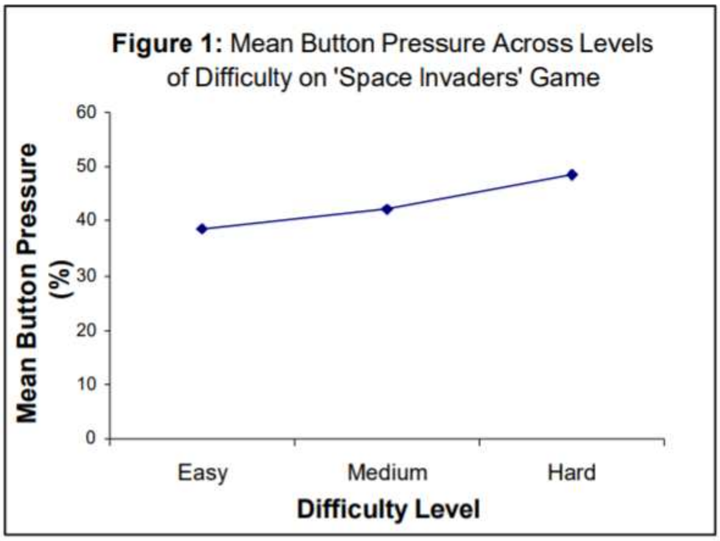


Figure 1 - Mean button pressure across levels of difficulty on 'Space Invaders' game

While the previous test shows that the pressure applied to a button press could potentially be used for tracking a user's emotions, another piece of research looked another method of capturing affective data from common input devices and combining the data. The main aim of this research was to see if combining the inputs of three different devices would provide a higher accuracy in emotions recognition (Bakhtiyari, Taghavi and Husain, 2015). The devices being analysed were all common input devices such as keyboard keystrokes, mouse (touch pad) movements and touch screens. For the keyboard keystrokes, the researchers took the length of time between each key press, the length of time to press down a key and the length of time to release a key. For the mouse and touch screen, the length of time and distance between the start point and the end point were extracted and examined. Test participants were asked to state which emotion they were feeling and then were monitored while they interacted with the input device. The emotions that the test was attempting to track were neutral emotion, fright, sadness and nervousness. Overall the test managed to detect the intended emotion 93.20% of the time which is a reliably accurate result. This shows that combining input devices can lead to a fairly accurate result.

3.2 Heart Rate

One of the rising trends in the past few years is the increase of the number of smart watches and fitness trackers being purchased as they become more popular. Between November of 2015 and February of 2016 the number of wearable devices being owned by people in the UK alone grew by 750,000 with the majority being smart watches (YouGov, 2016). In Q1 of 2016, 71% of all wearable devices owned in the UK were fitness trackers with the rest being either smart watches or other devices. The release of new smart watches such as the Apple Watch (Watch, no date), Google's Android Wear (Android Wear, no date) and multiple Fitbit watches (Fitbit, no date) have been the main reason for this rise. Many of these watches have multiple uses such as handling text and phone calls when they are linked to smart phones, playing music or GPS location tracking. One of the biggest draws however is the inclusion of heart rate monitors in many of the watches. While having a heart rate monitor is a smart watch has obvious applications when it comes to fitness tracking, it could also have other uses in applications that make use of biometric data. One such application could be the video game industry. As smart watches popularity rises, it is more likely that gamers will own such smart watches that include heart rate monitors, allowing game developers to potentially make use of these devices. One such game that made use of biometric data, as mentioned earlier, was the game Nevermind, which changed aspects of the game depending on the player's heart rate.

One of the questions raised with using heart rate monitors on smart watches is the accuracy of such devices compared to dedicated technology used such as an ECG machine. Fortunately some research has been carried out to attempt to answer this question. A test was carried out where 112 hospital in-patients with various heart issues were asked to wear either the Apple Watch or Fitbit Blaze while they are being monitored by a 5-lead ECG machine for thirty minutes (Koshy *et al.,* 2017). During that time, 54,030 HR values were recorded from all devices. The mean value across all heart rates showed that the Apple Watch showed the strongest correlation to the ECG in HR estimation with a value of 86 compared to the ECG which had a value of 87. The Fitbit Blaze did not do so well with an average value of 80 but these values show that the smart watch heart rate monitors are reasonably accurate compared to an ECG machine depending on the device being used.

3.3 Facial Recognition

One of the key ways to read someone's emotional state or to convey to someone how you feel is through facial expressions. A human's face can give off many ranges of expressions, both voluntary and involuntary (Ilves et al., 2014). Involuntary facial expressions occur spontaneously when someone experience emotions like when they become frightened surprised or when they are angry. These emotions can occur very quickly hence why they can cause involuntary facial expressions. Voluntary facial expressions can be used to affect how another person is feeling; for example, a smile can be used to convey that you are happy and then can be used to cheer someone else up or show friendliness.

Since facial expressions are so pivotal in communications between humans, it is a subject that has been widely researched in how computers can read expressions and how they may react in turn to the user's emotional state. Facial expression tracking could provide a great benefit to computer games in many different ways. An artificial companion that could react to a player's voluntary expression such as a smile or smirk would help make a character in a game much more believable and realistic. It would also help to make a game much more immersive and make the player more invested. This is another aspect that facial expression tracking could be a useful tool in a video game, tracking how much attention a player is paying to the game and how invested they are. In 2008, research was carried out to see if a facial recognition application could be developed to track a player's eye movements to see how much attention a player was paying to a game (Da Silva *et al.,* 2008). The hypothesis of this research is that someone who is interested in the game will focus their eyes on the certain aspect of the game such as the in game avatar, enemy or text on the screen. If a person is not looking at the screen for a certain length of time, then it would be assumed that they are not interested and the application would react to the player's inattention. The application would also track the position of the player's head in order to gauge if they were looking at the screen or not. Much like this project, this application was designed to work for a wide range games and be low cost so that it would not interrupt the running of whatever game it was working with.

As facial expression systems can be difficult and time consuming to create, it was decided that a pre-existing system and database would be used in order to save time and resources in creating a new system. After researching different facial recognition applications, the system that would be used for this project was the Affdex plug-in for Unity created by Affectiva (Affectiva, 2017). This plug-in was chosen as it was known to run with Unity and had plenty of pre-existing documentation online to help with implementing and setting up the system which made adding it to the project simple and less time consuming. Another reason Affdex was chosen was due to its output system while tracking facial expressions.

Affdex will simply output a number between 0 and 100 for each emotion being tracked, which is helpful for when it is being passed on to the next stage of the project.

4. Methodology

The application for this project has been made using several different programs. The main application has been created using the Unity (Unity Technologies, 2018) game engine, the task of this main application is to bring together the three sources of data and fuse them into a single input which can potentially be used in a computer game.

An Application will also be made in Unity for tracking a user's input on the controller. This will take the form of timing how quickly a user presses or pushes a thumb stick or trigger.

Two additional programs have also been created using Android Studio for the Android Wear watch and smart phone. The purpose of these applications were to read the heart rate from the watch, send it over to the smart phone where the data will be saved, ready to be accessed by Unity.

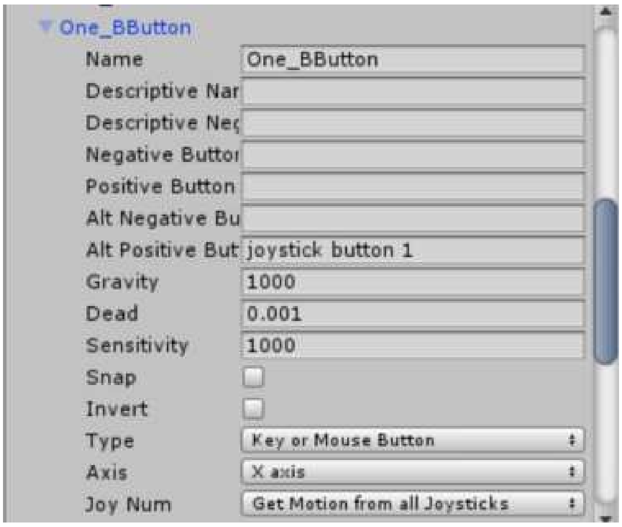
4.1 Controller

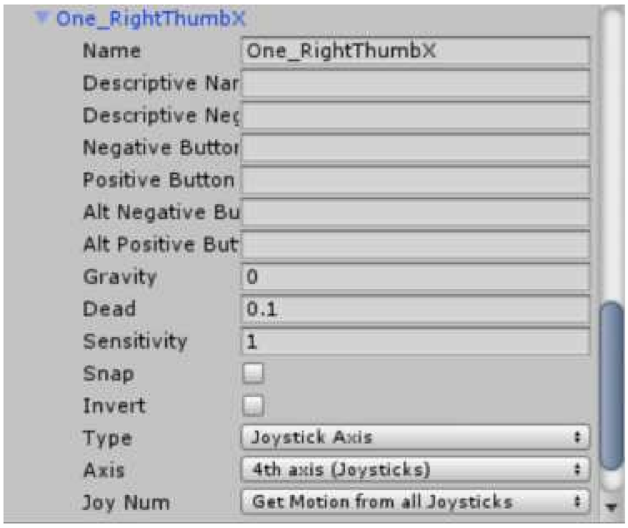
For the console game controller side of this project, an Xbox One controller has been selected. The controller will be connected to the application using a USB port rather than wireless communication to ensure a more reliable connection. It has also been selected as it integrates well with a PC, it can simply plug into a computer and it shall work.

In order to extract data from the controller while the user is playing a game, Scripts have been created in the Unity engine using the C# language.

Before data could be collected however, the controller must first be configured so that Unity identify what each button, thumb stick and trigger is when they are pressed or activated (Unity Answers, 2017). This also makes programming each input on the controller much easier as you only have to call the tag that you have given each button on the controller. In order to configure the controller, each input on the Xbox One controller is given a number or axes which can then be accessed in the input tab in Unity's project settings. For example as shown in the figures below, The B button on the controller is number 1 which can then be accessed in Unity by assigning an input to "joystick button 1".

In order to access the X value of the right thumb stick, an input in unity must be assigned "joystick axis 4".





Figures 2 and 3 - The input tab in Unity

It is worth mentioning that this controller button map is for Windows 10, it was not tested to see if it would work correctly if used on Mac or another operating system.

Once the controller had been configured, the scripts for collecting controller data can then be added. The data that is being collected from the controller is the length of time that is taken for a player to press a thumb stick or trigger from its neutral position to being fully pressed. A thumb sticks input values range from -1 to 1. For example, when the left thumb stick is pressed all the way down, its X value is -1 and while it is 1 when it is pressed fully up. When sitting at a neutral position, the value of the thumb stick should remain at zero but this is not always the case as shall be explained later in the chapter.

When the thumb stick or trigger are pressed and their values are no longer zero, an internal timer shall start that shall run until the value reaches 1 or returns to zero, at which point the timer shall be reset ready for the next input. Once a successful time has been calculated, it will then be sent forward ready for analysis. To ensure that miss-presses or incorrect data is filtered out, only times that are above a certain value are allowed to pass on to the next stage.

The time shall then be saved into a text file located in the asset folder of the Unity project. The trigger and thumb stick data shall also be recorded into the text file along with the current time and day so that the data can be analysed later on.

Over time, controllers can suffer from wear and become less sensitive as a result. An example of this is the thumb sticks, as they wear, they can sometimes stick or not return exactly to the correct position. This can cause the controller to think that the stick is being pressed slightly. This caused some issues with the scripts created for timing the movements of the thumb sticks as the values for the controller would not sit at zero and would cause the script to take abnormal readings. This issue was fixed using a technique known as a dead zoning (Third Helix, 2013). This is where there an area of motion of the thumb stick is set so that it will remain at zero no matter where it is physically in that area. In terms of code, it is very simple to implement and add to the script.

```
if (tempLThumbX < -0.2 || tempLThumbX > 0.2)
{
    tempLThumbX = 0;
}

if (tempLThumbY < -0.2 || tempLThumbY > 0.2)
{
    tempLThumbY = 0;
}
```

Figures 4 and 5 – Code used to create dead zone and graphical representation of dead zone

By adding these two if statements, if the value of the thumb stick is between -0.2 and 0.2 on both the X and Y axis, it will be reset to zero. This means that the timer will not start as it has not changed from zero.

In order to test the data collection using the controller, a small test was created to collect base line data. The test itself takes the form of a simple two dimensional game where the player controls a small triangular ship that rotates on the z axis. At the edge of the screen there will be enemies that spawn and will travel toward the player's position. The player's objective is to rotate the ship toward the enemies and fire bullets at them and stop them from getting too close. The player must use the left thumb stick to rotate the ship and the right trigger to fire bullets. To make sure that a consistent distance of the trigger has been pressed for every time reading, a bullet shall only be fired when the value of the trigger is one. This means that the player must fully press the trigger for a bullet to be fired. The thumb sticks must also be fully pressed left or right for the ship to rotate around, this again ensures that consistent times are taken. While the test is being played, the timings from the controller will be taken and recorded.

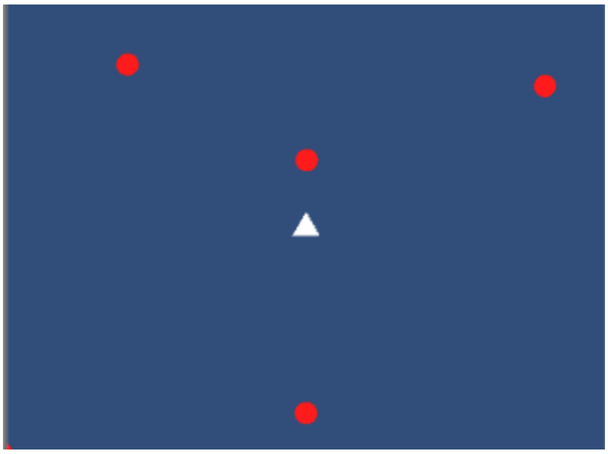


Figure 6 – Screenshot of controller test

Like the test carried out back in 2003, in order to look for emotions, the game has three difficulty levels, each more difficult than the last. When an enemy is destroyed, it shall respawn after a random amount of time between values that is determined by the difficulty. On the easiest difficulty there is a large amount of time between the enemies respawning which makes the game reasonable simple and laid back. On the harder difficulties, there is very little time between respawns which makes the game much more frantic and potentially frustrating. The idea of the test is that if a player is feeling frustrated or agitated, they shall push the thumb sticks and triggers quicker.

4.2 Heart Rate

The heart rate aspect of this project has been the most challenging part of creating the application. In order to receive the heart rate data, a plan had to be devised in order to collect the data from a heart rate sensor and pass it into Unity. Research was also carried out in order to see which smart watches would be suitable for tracking the heart rate for this project. The smart watch obviously had to feature a heart rate sensor, but also had to be; not too expensive, so it is not out of reach of the average gamer. Lightweight, so that it does not get in the way of the gamer or break their immersion. Finally the smart watch had to have its own software development kit so that we can create our own application for reading the heart rate and then sending the data.

For this project, the smart watch that was chosen was the Huawei Watch 2(Huawei, no date). Fortunately for the project, after some discussions with the fitness tracker company Fitbit, they have kindly donated one of their smart watches for use in this project. This meant there were two watches to develop for which would allow a comparison of the two watches in terms of accuracy and ease of use.

The original plan for gathering the heart rate featured several steps to get the data from the smart watch and into Unity. The first step was to read the heart rate from the smart watches in built sensor. This was accomplished by creating a separate app that would activate the heart rate sensor when it is run and would send the data over to a connected phone or Bluetooth device.

Developing an application for the Fitbit Ionic involves using Fitbit's new development software, Fitbit Studio. The development software works as an online application that you access through a browser. While this makes development fairly accessible, it does add the issue of requiring an internet connection in order for it to work. The development kit is also fairly new so had a few issues such as connecting the watch to the development server so you could upload and test your created apps. The development kit does have helpful documentation however which aided me in collecting the heart rate data.

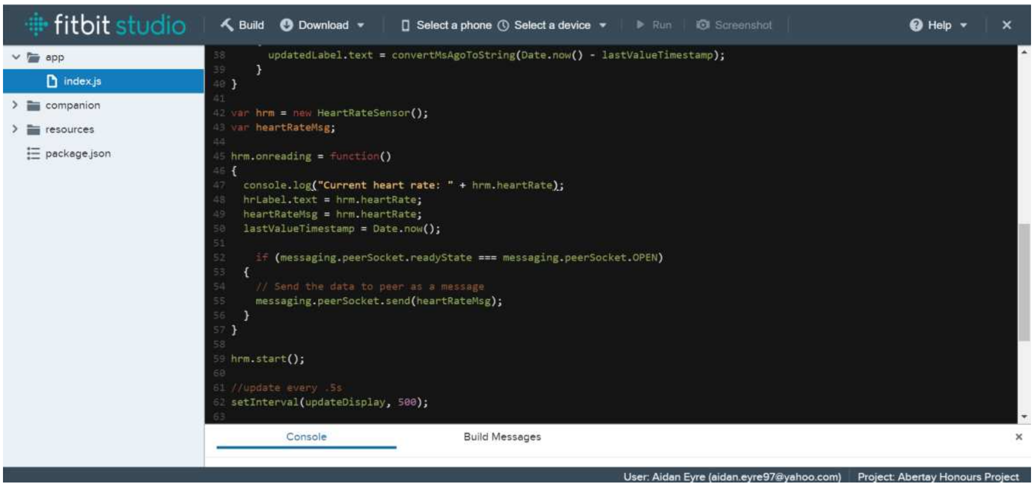


Figure 7 – Screenshot from Fitbit Studio

Fitbit studio uses the Java language in order to program the watch and companion. In order to use the Ionic's heart rate sensor, it simply has to be called and the function defined so the watch knows what it must do with the data once it has been called. When the heart rate function is called, it shall first update the display to show the user what their current heart rate. It shall also give the user the length of time since the last successful reading. The watch application will finally send the heart rate number over to the phone application using Fitbit's own messaging system.

Creating the application for the Android wear watch was much more complicated than the Fitbit Ionic. Creating android wear applications using android studio requires a lot of setting up before you can correctly test and use an app.

```
<uses-permission android:name="android.permission.BODY_SENSORS" />
```

Figure 8 – Code for giving permission to application

The first stage of creating the application is to give permissions so that the watch will be able to collect the correct data and function in the correct way. The first permission that the application requires is permission to use the watches body sensors, which includes the heart rate sensor. Without this permission, the application would crash whenever a function related to the sensor is called. Another important permission that the watch will require is in order for the watch to stay awake while the appliction is running. Usually, an android watch will put the screen and CPU to sleep after a few seconds in order to preserve battery life. However, for this project to work correctly, the watch has to stay awake so it will constantly be reading the user’s heart rate. The next step, is to create a listener and configure it so that it will be listening for any changes on the heart rate sensor.

```
sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
heartSensor = sensorManager.getDefaultSensor(Sensor.TYPE_HEART_RATE);
sensorManager.registerListener( this, heartSensor, SensorManager.SENSOR_DELAY_FASTEST);
```

Figure 9 – Initilising the sensor listener

After the listener has been set up, all the application has to do is wait for the users heart rate to change. If the sensor detects that the heart rate has changed, then a function shall be called that will record the beats per minute (BPM) and send the number over to the connected phone.

```
@Override
public void onSensorChanged(SensorEvent event) {
    Log.d(TAG, "" + (int) event.values[0]);
    float HeartRateFloat = event.values[0];
    HeartRate = Math.round(HeartRateFloat);
    textView.setText(Integer.toString(HeartRate));
    sendHeartRate(HeartRate);
}
```

Figure 10 – Code for when data on the sensor changes

The next step was receiving the data on a connected phone. The data shall be sent wirelessly, this meant that another application had to be created for the phone so it knows what data to expect from the watch. For both of the watches, when a watch app is created, by default a companion app is also created in their respective development kits. The only purpose for this companion app is to sit and wait for the heart rate data to be sent. Once the heart rate data has been obtained, the companion application shall then save that data to a text file along with a timestamp to a location on the phones external storage.

This is where problems with the Fitbit were encountered. Unfortunately due to the development kit still being new, the feature to save data to a text file is currently unavailable. This makes recording the data for the Fitbit more difficult as the data would have to be recorded by hand, considering the amount of readings that would have to be taken, this would be unreasonable.

Fortunately the Android watch does allow its companion app to save text files on to the phone so it can be used to record data. The mobile application for Android, like the Fitbit application, shall sit and wait for data to be sent from the connected smart watch. Transferring data between phone and smart watch does not work quite as you would expect in terms of just sending a message from one to the other. Instead, the two devices will synchronise data between each other, in this case the heart rate. So when the heart rate changes on the watch, that data shall be compared to the heart rate variable on the phone. If the heart rate is different than zero, then the variable on the phone shall be changed so it matches up with the variable on the watch.

Once the data has been changed on the phone, the display shall be updated to show the heart rate on the screen and the data shall be saved to a text file located on the phone. Along with the heart rate value, the time at which the reading was taken is also saved into the text file so that it can be synchronised with the data gathered from the other inputs.

Figures 11 and 12 – Output screens for watch and mobile applications

4.3 Facial Recognition

The facial recognition system is probably the most important part of the emotion tracking system as facial expressions are the key method of telling how someone is feeling. Luckily, as this is a well-researched area, finding and implementing a facial expression system into Unity was relatively easy. The recognition system being implemented in the project is the Affdex plug-in by Affectiva. Implementing the plug-in into Unity was simply a case of downloading the package and importing it into the asset file of the Unity project and using the scripts included (Affectiva, no date). The plug-in also contains some example projects to show how Affdex works and how it could be used. These were very helpful in setting up the recognition system and collecting the data necessary for the project.

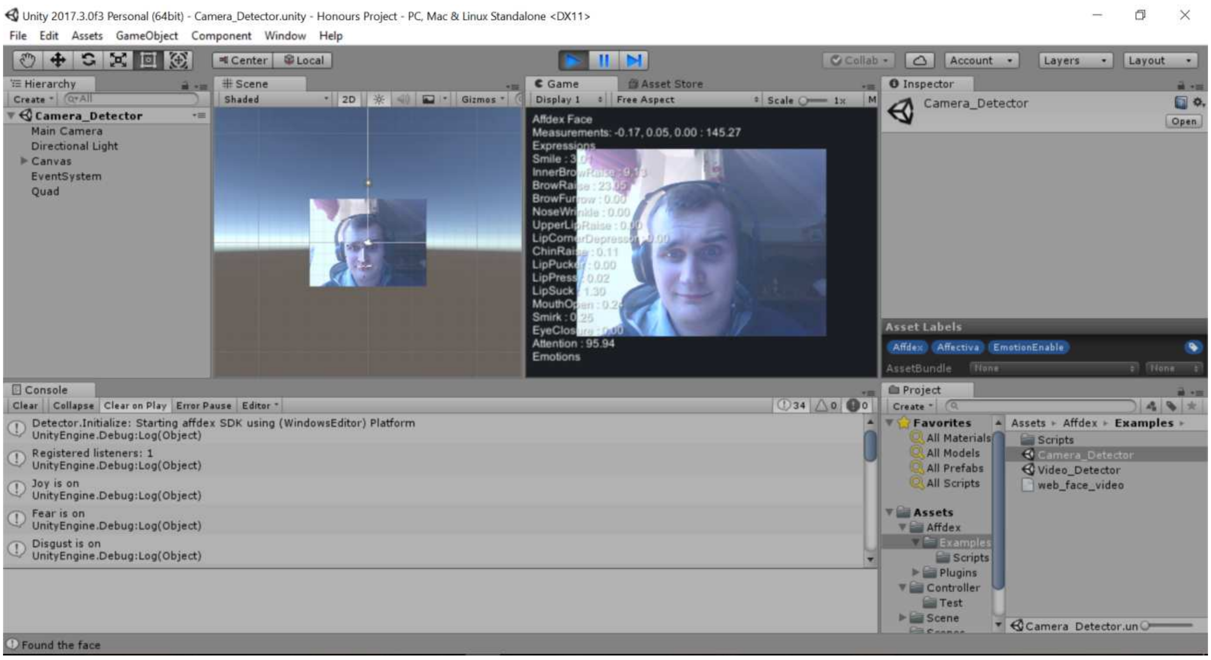


Figure 13 - One of the Affdex Example programs

In order for the Affdex system to work, a few scripts that were created by Affectiva must be added to the main camera in Unity so that it will be able to track the users face. The first script to be added is the Camera Input; this script will activate the any webcam or camera that is connected to the computer. The second script to be added to the main camera is the detector. The Affdex works by taking a still image from the webcam and then processing that image to detect for emotions and expressions, the detector script is what processes this still image from the webcam. This detector script can be calibrated by the use to tell the

Affdex which emotions or expressions you are looking for. The expressions and emotions that you ask to the system to look for can have an effect on the projects performance, the more that you try to track, the more calculations the system has to carry out. Another script called view camera must also be added, the only purpose of this script is to feed the webcam image into the detector. The final script that must be added is one that must be created by the user in order to handle the data for each specific emotion being tracked.

```
public override void onImageResults(Dictionary<int, Face> faces)
{
  //  Debug.Log("Got face results");

    foreach (KeyValuePair<int, Face> pair in faces)
    {
        int FaceId = pair.Key;  // The Face Unique Id.
        Face face = pair.Value;    // Instance of the face class containing emotions, and facial expression values.

        //Retrieve the Emotions Scores
        face.Emotions.TryGetValue(Emotions.Joy, out currentJoy);
        face.Emotions.TryGetValue(Emotions.Fear, out currentFear);
        face.Emotions.TryGetValue(Emotions.Disgust, out currentDisgust);
        face.Emotions.TryGetValue(Emotions.Sadness, out currentSadness);
        face.Emotions.TryGetValue(Emotions.Anger, out currentAnger);
        face.Emotions.TryGetValue(Emotions.Surprise, out currentSurprise);
        face.Emotions.TryGetValue(Emotions.Valence, out currentValence);
        face.Emotions.TryGetValue(Emotions.Engagement, out currentEngagement);

        featurePointsList = face.FeaturePoints;

    }
}
```

Figure 14 – Code for retrieving facial expression values

The above code shows which emotions the application is to track for this project. Once the detector is told which emotions are being tracked, the above functions will return a value between 0 and 100. If the value for a single an emotion is zero, then that means that the Affdex detection system is not seeing or tracking that emotion and vice versa if the value is one hundred then that emotion is fully being detected.

In the application that has been created, while the option is available to show the webcam feed to the user, it was decided it would not be. This was in order to help improve the performance of the application and that the program itself would likely be covered up by the game being played.

While the facial recognition system is tracking the players face, it shall also be saving the emotion value to individual text files for each of the emotions being tracked. Like the other emotion tracking inputs, the current time shall be passed to the text file as well so it can be synchronised with the other inputs.

A small test was carried out to examine the accuracy of the facial recognition system when presented with certain emotions. 9 participants of the test were similar to those in the final test, were asked to show different facial expressions in front of the webcam to align to a certain emotion such as smiling for joy or frowning for sadness. While this expression is being pulled, the recognition system will show the value for that emotion which is then recorded.

5. Testing

In order to test how the accuracy of the project, a group of 12 participants from ages between 21- 29 and mixed genders were asked to play three levels of a game while the created applications would attempt to track their biometric data. They would then be asked to fill out a questionnaire about their experience with the levels and which emotions they felt during the experiment and how strongly they felt them.

The game being played by the testers is the popular building/sandbox game Minecraft (Mojang, 2011). The reason for this game being chosen is due to just how customisable it is and the amount of ways available to play the game. Minecraft can be used as a way to create artistic structures or creating adventure maps for users to play. The game also features an extensive survival mode where the player has to survive for as long as possible by building houses, defending themselves from monsters and creating farms for food. This mode also features lots of things for the player to do from farming to potion brewing. The great range of things to do and create in Minecraft made it perfect for creating different scenarios that can elicit different emotions.

The participants targeted for this test were ideally people who have had previous experience playing a wide range of games and using console controllers. The reasons for this is that someone who has had experience playing games using controllers are less likely to become confused with aspects such as controls. This would likely take their attention away from the game and more on which buttons they were pressing. Someone who is used to playing games also knows what to expect from playing a game and how they operate, this would help the test be completed more smoothly as less input and instruction would be required from the researcher.

The three maps used for this test are designed to try and elicit different emotions from the player as they play. The scenarios had to take around 5 or 10 minutes to complete so that the tester would not become bored or lose engagement. The levels also had to be fairly simple and easy to follow in order for the tester not to become lost or confused.

The first scenario tasks the tester to carry out some very simple and sedate aspects of the Minecraft survival mode. This includes animal farming, planting crops and then brewing potions. The aim of this scenario is make the player feel relaxed and happy. It also helps form base line data for the person playing the game.

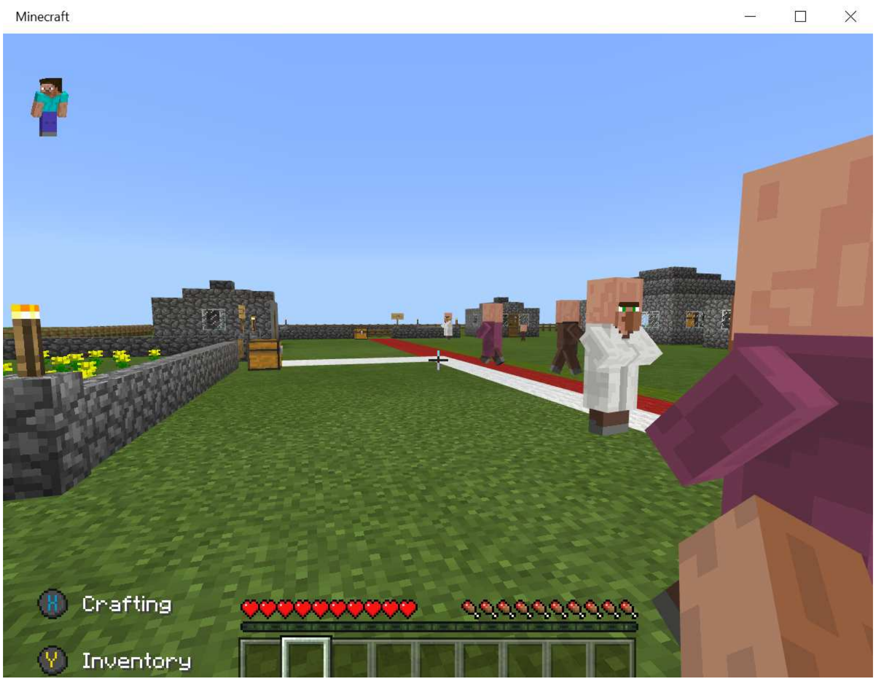


Figure 15 – Screenshot from Scenario 1 in Minecraft

The second scenario is designed to try and trigger the more involuntary emotions such as excitement and anger. In this test, the player must defend themselves and the animals from the monsters in Minecraft. The hope shall be that this will make the player feel tense, excited and potentially frustrated.

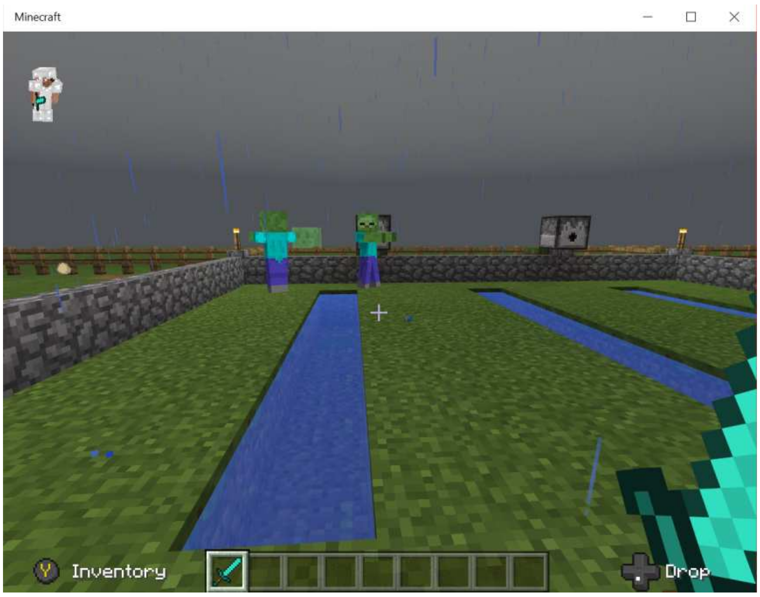


Figure 16 – Screenshot from Scenario 2 in Minecraft

The third and final scenario is designed to try eliciting emotions such as surprise and fear. The map takes the form of a horror map created by Creeper GamerZXZ (MCPLE DL, 2017) featuring jump scares to frighten and surprise the player. The reason for downloading a map shared online instead of it being created is that some of the mechanics of creating a horror map (such as jump scares, tense movement and the general atmosphere) can be complicated and time consuming to create. So a readymade horror map was found online instead.

Figure 17 – Screenshot from Scenario 3 in Minecraft created by Creeper GamerZXZ

Once the players played all three scenarios, they were then asked to answer the questionnaire. The questionnaire first asked to tester for a quick profile just so an idea how familiar each of the participants are with games, asking how long and how often they play computer games. It also asks the participant how many games platforms they own such as PC, Xbox One (Microsoft, 2017) or Nintendo 3DS (Nintendo, no date). The testers will then be asked to answer questions about each scenario. For each scenario there is a quantitative question where the participant are asked which emotions they felt while playing the map and how strongly they felt them, 0 for not at all and 5 for very strongly. They are then asked if they felt any other emotions during the scenario. At the end of the testing, participants are then asked two open questions if they have any positive or negative comments about the scenarios or the equipment used during the test. The facial recognition and heart rate data for some of the emotions of one of the participants from the test and the results of questionnaire can be found below (Other data samples may be found in the appendix);

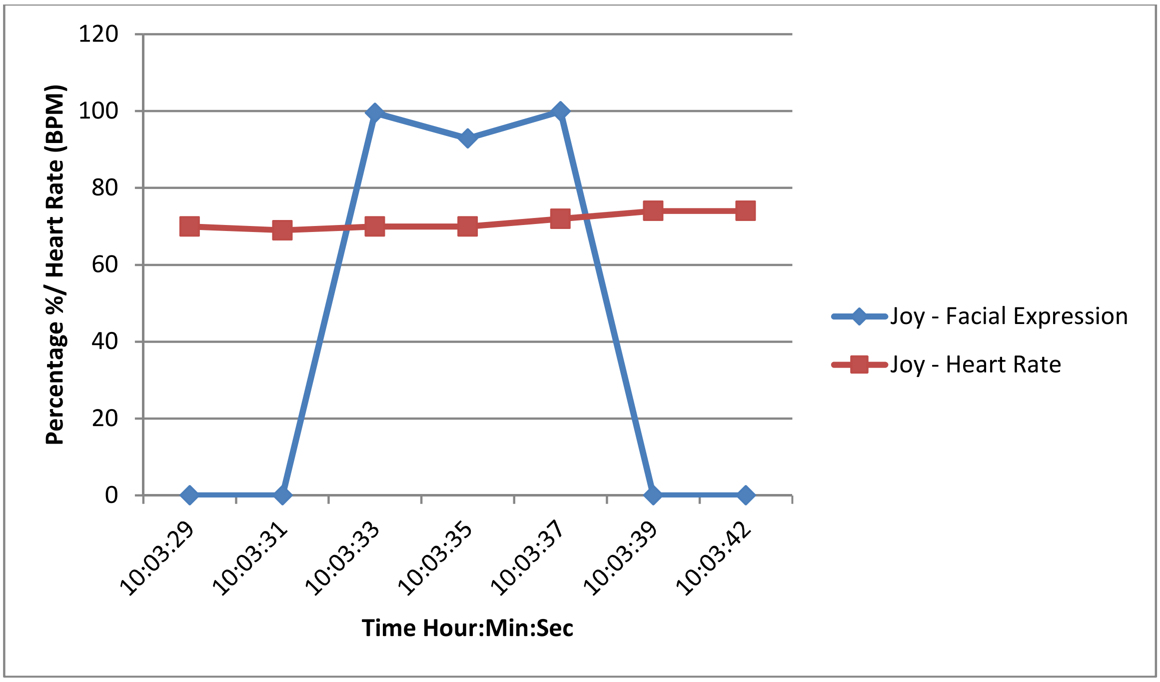


Figure 18 – Joy facial expression and heart rate value taken during Scenario 1

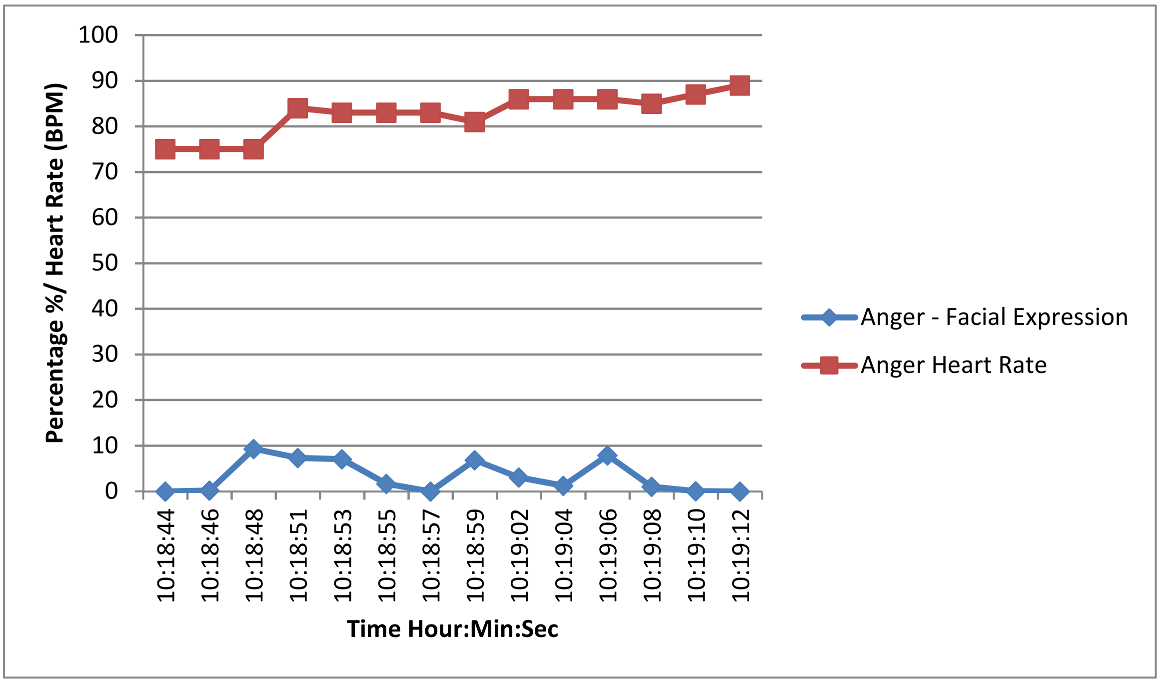


Figure 19 – Anger facial expression and heart rate value taken during Scenario 2

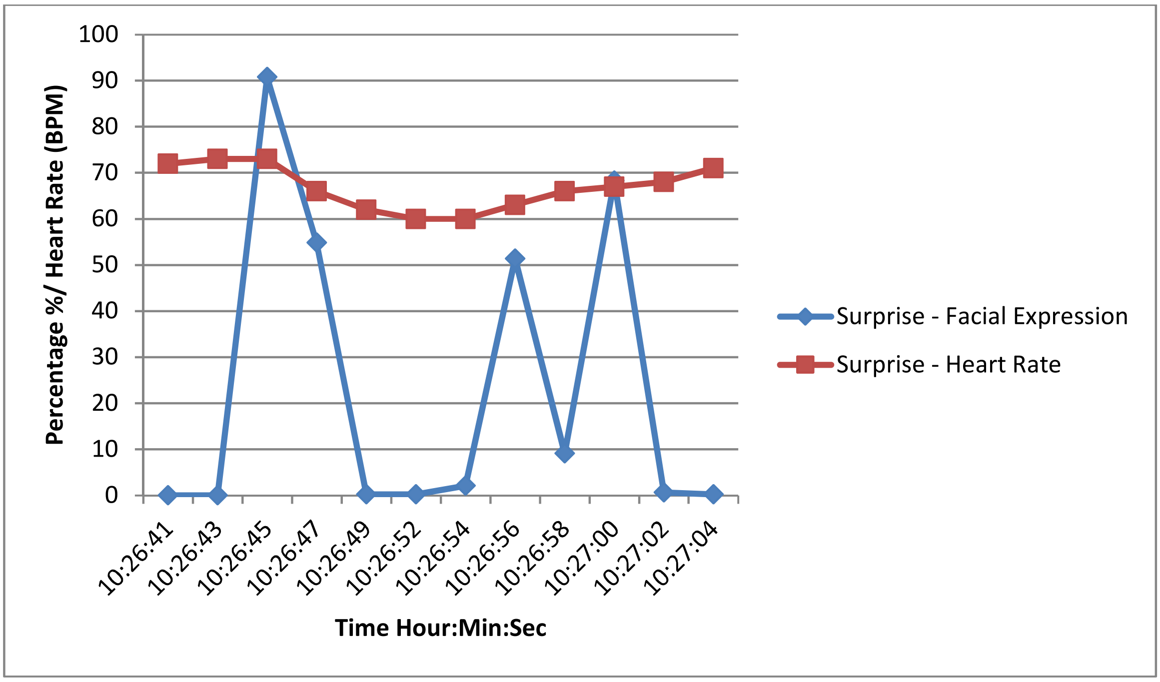


Figure 20 – Surprise facial expression and heart rate value taken during Scenario 3

Question 1: Please indicate which emotions you experienced during Scenario 1 (The farming map) with 0 being not at all and 5 being very strongly

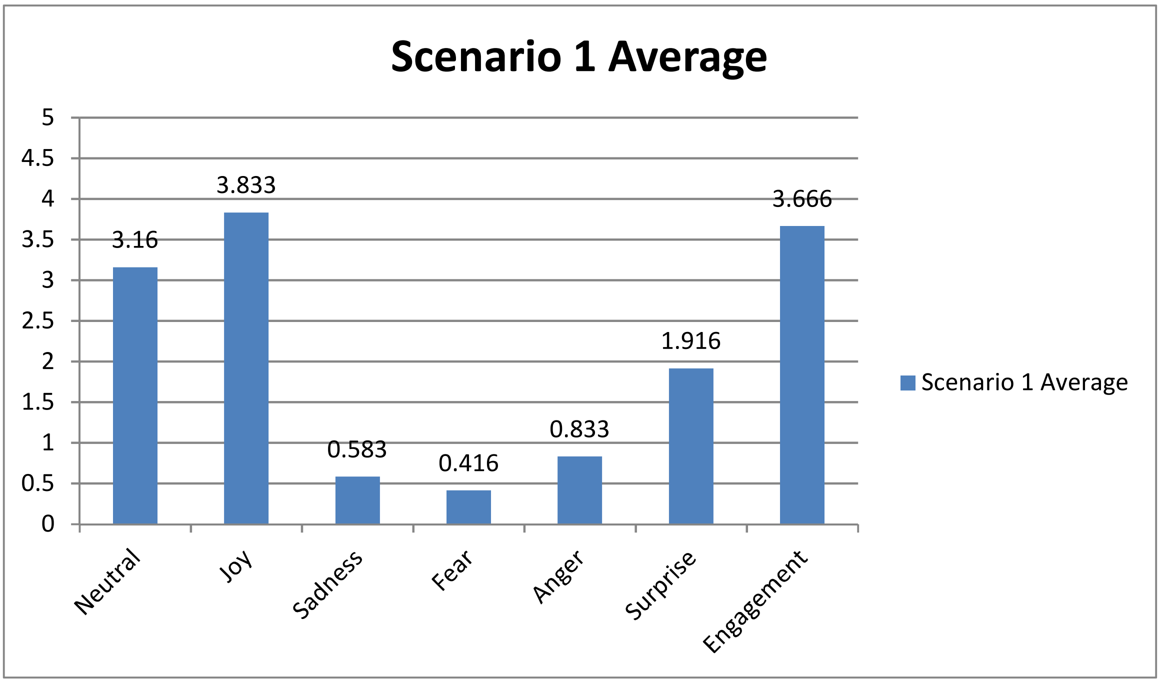


Figure 21 – Question 1

Question 3: Please indicate which emotions you experienced during Scenario 2 (The village attack) with 0 being not at all and 5 being very strongly

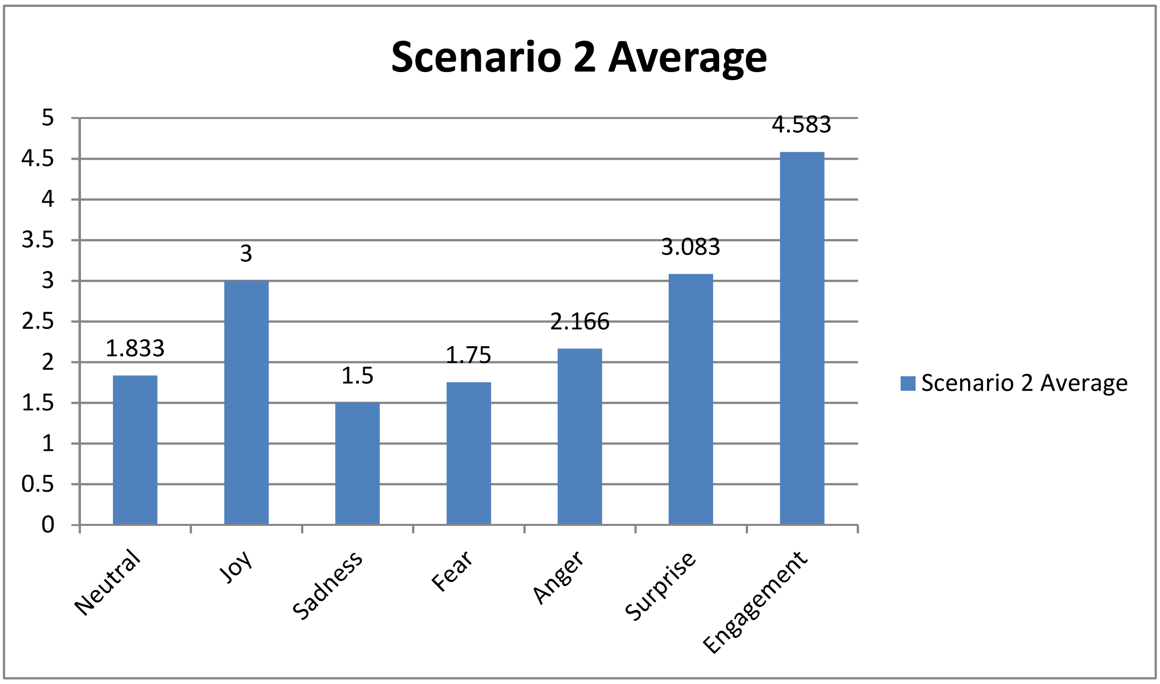


Figure 22 – Question 3

Question 5: Please indicate which emotions you experienced during Scenario 3 (The horror map) with 0 being not at all and 5 being very strongly

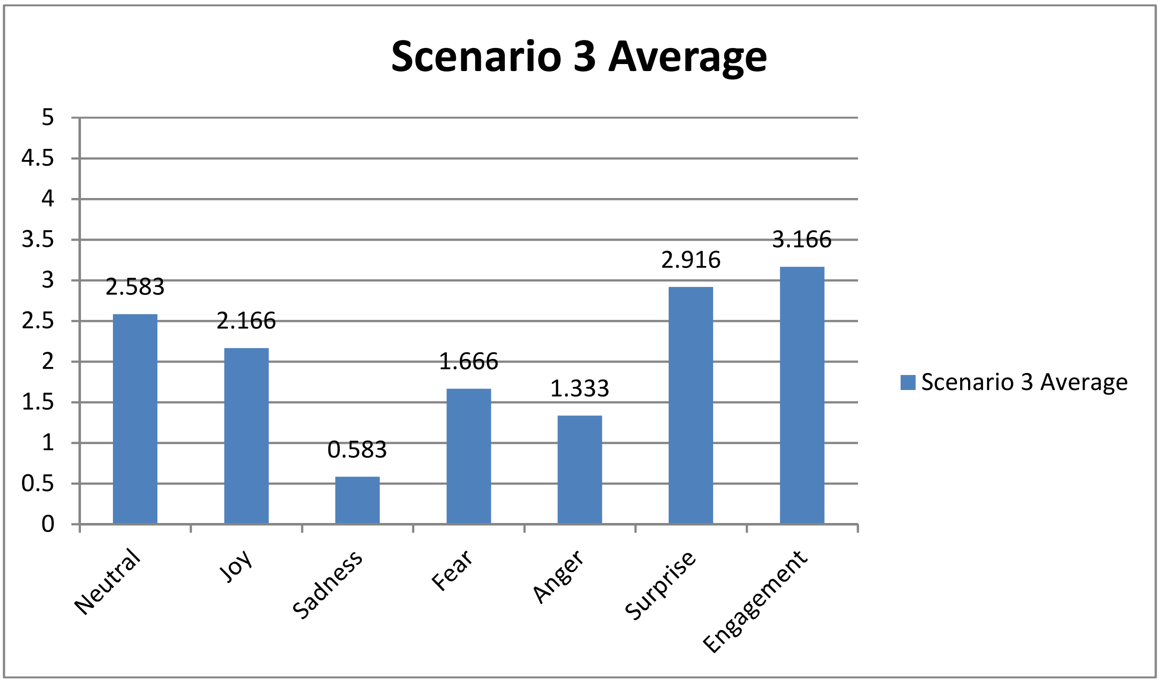


Figure 23 – Question 5

6. Discussions

6.1 Discussion of Quantitative Results

Figures 18 to 20 show the facial expression and heart rate results from a single participant across the 3 scenarios at points where that expression was at its highest for an extended amount of time. This is in order to see how the heart rate changes over that same period of time. When the facial recognition system detected joy and anger in figures 18 and 19 for a few seconds, the heart rate during that same time rose by a few beats per minute as can be seen in figures 18 and 19. While there was only a slight change in the heart rate during this time, this does suggest that the player was becoming engaged, excited or frustrated at this time and it was having an effect on their heart rate. However, figure 20 did not reflect the same when the player became surprised during scenario 3. The heart rate seemed to dip slightly rather than rise which is what is expected when someone is feeling surprised or frightened. This may be caused by and issue with the heart rate sensor or the participant of the test wasn't surprised or scared enough to show the expected reaction.

The participants of the test were asked what emotions they felt during each scenario on a scale of 0 to 5 with 0 meaning they did not feel that emotion at all and 5 meaning they felt that emotion very strongly. The above figures show the mean average taken from each emotion for each scenario from the 12 participants of the test.

Figure 21 shows the average scores from the first scenario where the player was given some simple Minecraft tasks to complete. The aim of this scenario was to make the player feel relaxed, happy and ease them into how Minecraft works. As the results show, the strongest emotions being felt were joy and engagement with averages of 3.833 and 3.666 respectfully. This shows that the scenario managed to achieve its objective and the correct emotional data was collected. The data also showed that the players felt slightly surprised, this may be the case if they are new to Minecraft or do not play games as much.

The second scenario's average values can be seen in figure 22. This scenario's aim was to be more exciting and elicit emotions such as anger and surprise as the players are tasked with fighting some of the monsters found in Minecraft. As the data shows, the two strongest emotions, like scenario 1, were joy and engagement with averages of 3.083 and 4.583 respectfully. Out of the 3 scenarios, the second scenario proved to be the most engaging which could be a sign that the testers found the map to be exciting or immersive. Anger and surprise came in as the next strongest emotions which are what the map was aiming to elicit in the players. While it is good that the map was able to elicit these emotions, they were perhaps not strong enough in order to show a significant change on the input systems such as a change in heart rate.

The average values for the final scenario can be found in figure 23. The third scenario was a horror map that was downloaded and designed to trigger emotions like fear, surprise and sadness. As the data shows sadness had an average of 0.583, fear had an average of 1.666 and surprise had an average of 2.916. While the fear and surprise averages are higher than that of the scenarios, the participants did not find the map scary enough to make a significant difference to the data collected.

6.2 Discussion of Open Question Results

As part of the questionnaire, the participants of the test were asked open questions about if they experienced any other emotions during the scenarios. The testers are also asked if they have any positive or negative comments about the test scenarios or the test environment. A discussion about these results can be found below;

Question 2: Did you experience any other emotions while playing scenario 1?

“Amusement”

“Confusion, stupid”

“Relaxed”

“Unfamiliar”

“Deep guilt at punching a sheep”

“Relaxation”

“No, I mainly experienced engagement”

The intention of scenario 1 was to be relaxing, sedate and to ease the user into how Minecraft works and get used to the controls and the rest of the test set up. As the results of the question shows, this scenario was able to make some of the testers feel relaxed which is exactly the purpose of this map. However, it also caused some of the testers to become confused as some were unfamiliar with Minecraft or not used to using a controller. This could be prevented in the future by perhaps creating a tutorial level so users could get used to the set up.

Question 4: Did you experience any other emotions while playing scenario 2?

“Stressful”

“Stress”

“Busy”

“Baby zombies were terrifying”

"No, the main difference was more things took me by surprise"

The aim of the second scenario was to be more exciting or frustrating as the users are tasked with defending themselves and other characters from monsters. As the answers show, the map was stressful which hopefully meant it was more exciting and invigorating for the participants to play.

Question 6: Did you experience any other emotions while playing scenario 3?

"Bewilderment"

"Shock"

"Confused"

"Anxious"

"Good jump scare at the start"

"Not spooky enough, could be longer"

Scenario 3 was meant to try and elicit fear, surprise and tension, however it unfortunately it wasn't very effective at accomplishing this. Participants thought that map was more boring than scary. The map doesn't have much of a story which caused the participants to become confused as they played as they did not really know what was happening. The map features a couple of jump scares which some of the testers thought were okay but it could have had more.

Question 7: Do you have any positive comments about the game scenarios? (Intensity, length of scenario, etc.)

"All of them were good; however, scenario 3 was rather short and was more funny than scary"

"No major use of sounds"

"The horror map was lame, find a better horror map"

“Liked the books in first map explaining everything. Preferred first 2 maps. The sheep were cute! Zombies are scary! Enjoyed planting things.”

“Less villagers!”

“Every scenario was good and enjoyable”

“Add creepers and baby zombies”

As the comments for the scenarios show, the participants found the first two of the scenarios enjoyable, relaxing and engaging which is good because that was the purpose of the first two scenarios. However they found the third scenario to be too short and not scary enough. For future testing, it would be more beneficial to find a better map that is longer and can more effectively elicit emotions such as fear or surprise. If more time were available, it may be worthwhile to simply create the map so that it could be customised to exactly to what is needed.

One of the comments also mentioned that there is no major use of sounds which is an excellent point. Sounds and music in any media of entertainment can have a massive impact on the tone and feeling of the media being consumed, especially in games. Games developers have been making use of sounds in games for years to affect how someone who is playing a game feels. For example, in games such as The Legend of Zelda (Nintendo, 1986) and Sonic the Hedgehog (Sega, 1991), the game uses the major pentatonic scale as a reward cue to let the players know they are on the right path. This can also have the effect of making them feel happy about what they’ve accomplished (Amplifon, no date). For any future testing that is carried out for the project, it would be worth looking into how sounds can be used to affect emotions and how it could possibly be incorporated into testing for different emotions.

A few comments also related to the content found within the testing scenarios in Minecraft. At the start of each scenario there was a book to guide the player through the map. It was written to be light hearted, funny but also a guide so it was good to see that it guided the players smoothly but also provided a joyful response which was good for data collection. Villagers were also placed in the maps to add a bit more to the atmosphere of the level, however they became more of a distraction and got in the way of the testers then add anything to the map. One participant also suggested adding creepers and baby zombies to the levels. This would be a good suggestion as creepers are a monster in Minecraft that sneaks up behind the player and explodes. This would be useful to try and frighten the player or surprise them. Baby zombies on the other hand run toward the player very quickly and could have been used to cause the player to panic or make them frustrated.

Question 8: Do you have any positive or negative comments about the test? (Equipment, settings of test, etc.)

"Getting to see how I react facially was very interesting. Seeing my heart rate was also interesting"

"Fair spread"

"Watch was comfy – interesting to see heart rate. Controller was sexy."

"All positive. Facial recognition was interesting."

"The test was good. Felt comfortable during the test. No issues with the equipment being used."

It is positive to see that the participants of the test were comfortable with all the equipment being used during the test. When deciding on which smart watch to use with the project, one of the key goals was that it had to be comfortable and not disrupt the user while they played the game. Fortunately the smart watch selected for use in this test functioned as hoped and did not distract the tester as they played. Having the heart rate displayed to the user was also a benefit as the testers as it helped them be more engaged with the game. It may be beneficial to have the heart rate displayed on the screen to the user instead of just on the watch as the user as to look away from the screen to see the watch which could disrupt the facial recognition system.

Much like the heart rate on the smart watch, the testers also found it interesting to see how their face was reacting while they played the game. During the test, the values were being displayed on Unity at the side of the screen so that the participant can see how the facial recognition system was tracking their face. Although it did make the user more engaged with the game, it was observed that it did slightly distract them as for some participants; it did make them giggle or react differently then what was intended which gives incorrect data to the recognition system. It may be worth looking into a way to indicate which facial expression the user is showing but making it less disruptive.

6.3 Open Discussion

An unfortunate issue that was discovered during the testing had a major impact on the data collected was the fact the data could not be collected from the controller. The reason for this is to do with the fact that Unity is used to track the controller and facial recognition data. Unity will only track controller input while the window is in focus which was an issue as Minecraft was being used to test the applications which had to be played in a separate window. This issue could easily be fixed by simply doing the testing within Unity instead of using a separate game; however there was not enough time to create scenarios in Unity that would be able to elicit all of the emotions that are wanted to test the application.

One observation about the third scenario during the testing and by some of the comments made was that it was not scary or surprising enough. This may be down to the participants who were targeted for this test being used to playing games on a regular basis so are simply not as affected by horror games as much as someone who does not play games regularly. There was trouble in trying to find a suitable map as not many maps met the requirements for testing. The map needed to only take around 5 to 10 minutes to complete as explained earlier, but unfortunately many of the horror maps shared online would take 20+ minutes to complete which limited the choices available.

For the test, the targeted participant was ideally someone who was used to playing computer games and used to using a controller. This was so that the tester spent less time focusing on the controller and focused on the test scenario instead. During testing, some of the testers had previous experience with games but were more used to using a keyboard and mouse rather than a controller. These individuals found using the controller somewhat confusing and commented on how they were looking for the right button to press rather than focusing on the in game tasks. These participants also took slightly longer to complete the scenarios and required a little more guiding through the maps than some of the other participants. This did reinforce the point of having testers who are more familiar with controllers. In an actual computer game scenario where a controller would be used to track the players emotional state, someone who is less familiar with a controller could give off incorrect data and cause the system to behave incorrectly to what the player is actually feeling.

7. Conclusions and Future Work

7.1 Research Question

At the start of the project, the research question was asked to “develop and evaluate new techniques to improve computer games using affective computing”

To begin with, research was carried out to find previous work looking at if it would be possible to track a player’s emotional state using standard equipment such as controllers, heart rate sensors in smart watches and facial recognition in webcams. The research found that while separately, these devices could be used to some degree to track affective data, it had never been used together at the same time. The research also looked at the accuracy of each input and found that facial expressions were the most accurate way to tell someone’s emotional state with heart rate and controller input being less accurate depending on the smart watch and controller being used.

Developing the applications to be used in this project was interesting, enlightening and at times also frustrating. Developing a technique to track affective data using the controller was somewhat experimental as little research had been carried out in this area. For this project it was theorised that the length of time for a button to be fully pressed could be used to track affective data. However for the final testing unfortunately this feature could not be properly tested which is disappointing. Hopefully for future work, a test can be created to see if the controller can be used to collect some form of affective data.

Creating the applications for tracking a user’s heart rate was the most frustrating part of this project. At first, The Fitbit Ionic and the Fitbit software development kit was used to try and develop an application for tracking the user’s heart rate, however, issues were encountered with the development kit so an alternative watch was used. Development for the Android Wear watch was also challenging as the online documentation for developing an application was complicated and difficult to interpret which took up valuable time. While suitable applications for both the smart watch and connected mobile device were developed and worked as intended, another major issue came when trying to move the data into Unity. No way was unfortunately found to move the data from the smart watch to Unity, so the data was instead saved onto a mobile device and manually analysed. In the future when smart watches become more common, an easier method will be developed to connect a smart watch to Unity.

The easiest of the applications to be developed was the facial recognition system as a readymade system was selected for use in this project. The Affdex system also had a Unity plug-in already available as well which made integration into the project even easier. Scripts had to be created in order to use the system but fortunately there was very helpful documentation online to show how this was possible.

The results of the test were on the whole were positive and showed that the facial expression and heart rate could be used to some degree to collect affective data. The facial recognition was most effective way of detecting a user's affective state, while the heart rate was less effective and did not change much during this testing. This may be down to the testing scenarios not being effective enough at eliciting certain emotions that may raise someone's heart rate such as fear or anger.

The comments made on the questionnaire did show however that having an affective element did make them more engaged and interested in the game being played. Some of the testers found that seeing their affective data such as the heart rate and the facial recognition values was interesting and increased their engagement.

7.2 Future Work

Moving forward with this project, the next steps would be to continue testing each of the three inputs to see how the body reacts and changes depending on the player's emotions. While this project was able to collect some good data, the testing scenarios were unable to elicit some of the necessary emotions strongly enough to make a real noticeable difference that could be picked up by both the facial recognition and heart rate monitor. As suggested by some of the test participants, this could be solved by testing the system with more effective scenarios such as horror games that will be able to trigger emotions such as fear and anger more intensely and for longer durations. This would give the affective tracking system a much better change to notice differences in the player's biometric data.

Testing a much larger and wider sample should also improve the results and collect much more valuable data that can be used to further improve the system. For this project, the targeted test participants were mainly people who were used to playing game. This may have meant that they were less likely to be affected by surprises or challenges that would trigger an emotional response. For future testing, participants should come from varying background experiences with playing computer games in order to collect a wider set of emotional responses.

More research and testing should also be carried out on using the controller as a way to track affective data. Unfortunately it was unable to be fully tested as part of this project due to the testing scenario used. If the same tracking application could be created outside of Unity or if the testing scenario was made in the Unity engine, this would allow for the controller to be tested on and be developed further.

From a technical point of view, the implementation of the watch could also be carried out differently to allow the heart rate data to be passed straight into Unity instead of being saved on the mobile device. This may be made easier in the future as smart watches become more popular and updates are created for both game engines and smart watches themselves.

After further testing has been carried out, the next step would be to see how collected affective data would actually affect a game and how the game could potentially react given a person's emotional state. The use of affective data can be used to change many aspects of a game such as a non-playable character, the in-game environment and even the music. As the project as shown, there is great potential for affective computing to improve computer games and the number of different techniques available to developers shall only grow as time goes on.

8. References

See You (Short Horror) [Adventure]. (2017) Available at: http://mcpedl.com/see-you-short-horror-map/ (Accessed: Apr 26, 2018).

Doing Thumbstick Dead Zones Right. (2013) Available at: http://www.third-helix.com/2013/04/12/doing-thumbstick-dead-zones-right.html (Accessed: 27 Apr 2018).

Analyze the camera stream. Available at: https://knowledge.affectiva.com/v3.2/docs/analyze-the-camera-stream-4 (Accessed: Apr 27, 2018).

Android Wear. Available at: https://www.android.com/intl/en_uk/wear/ (Accessed: Oct 23, 2017).

Emotion AI Overview. Available at: https://www.affectiva.com/emotion-ai-overview/ (Accessed: Mar 17, 2018).

Fitbit Official Site for Activity Trackers & More. Available at: https://www.fitbit.com/uk/home (Accessed: 23 Oct 2017).

'The Global Games Market 2017 | Per Region & Segment', Newzoo. Available at: https://newzoo.com/insights/articles/the-global-games-market-will-reach-108-9-billion-in-2017-with-mobile-taking-42/ (Accessed: Mar 17, 2018).

HUAWEI WATCH 2 | Wearables | HUAWEI UK. Available at: https://consumer.huawei.com/uk/wearables/watch2/ (Accessed: .

Microsoft - Official Home Page. Available at: https://www.microsoft.com/en-gb (Accessed: Oct 23, 2017).

Mojang (2011) Minecraft. [Video game]. Mojang.

Nintendo 3DS Family. Available at: https://www.nintendo.co.uk/Nintendo-3DS-Family/Nintendo-3DS-Family-94560.html (Accessed: Apr 29, 2018).

Playing with your mind: The psychology of sound in video games. Available at: http://www.amplifon.ie/resources/playing-with-your-mind/ (Accessed: 18 April 2018).

Unity - Game Engine. Available at: https://unity3d.com (Accessed: Apr 29, 2018).

Watch. Available at: https://www.apple.com/uk/watch/ (Accessed: Oct 23, 2017).

Xbox one controller mapping [SOLVED] - Unity Answers. Available at: https://answers.unity.com/questions/1350081/xbox-one-controller-mapping-solved.html (Accessed: Mar 1, 2018).

YouGov | Wearables sector grows as smartwatches increase in popularity. Available at: //yougov.co.uk/news/2016/03/18/wearables-sector-grows-smartwatches-increase-popul/ (Accessed: Oct 23, 2017).

Bakhtiyari, K., Taghavi, M. and Husain, H. (2015) *Hybrid Affective Computing - Keyboard, Mouse and Touch Screen: From Review to Experiment.*

Da Silva, M.P., Courboulay, V., Prigent, A. and Estraillier, P. (2008) *Real-time face tracking for attention aware adaptive games.*

Hamdy, S. and King, D.J., *"Affect and Believability in Game Characters – A Review of the Use of Affective Computing in Games",* Proceedings of GameOn'2017, 18th International Conference on Intelligent Games and Simulation, Carlow, Ireland, 6-8 September 2017, pp90-97.

Ilves, M., Gizatdinova, Y., Surakka, V. and Vankka, E. (2014) *'Head movement and facial expressions as game input'*, Entertainment Computing, 5(3), pp. 147-156. doi: 10.1016/j.entcom.2014.04.005.

Koshy, A., Sajeev, J., Zureik, M., Rajakariar, K., Street, M., Cosgrave, N., Wong, M., Hamer, A., Cooke, J., Roberts, L. and Teh, A. (2017) *Heart Rate Assessment by Smart Watch: Utility or Futility?.*

Nintendo (1986) The Legend of Zelda – Nintendo Entertainment System [Video Game]. Nintendo

Picard, R.W. (2000) *Affective computing.* 1st MIT Press pbk. edn. London: MIT Press.

Reynolds, E. (2013) *Nevermind: creating an entertaining biofeedback-enhanced game experience to train users in stress management.*

Sega (1991) Sonic the Hedgehog – Sega Genesis [Video game]. Sega

Sykes, J. and Brown, S. (2003) *Affective gaming: measuring emotion through the gamepad.*

9. Appendix

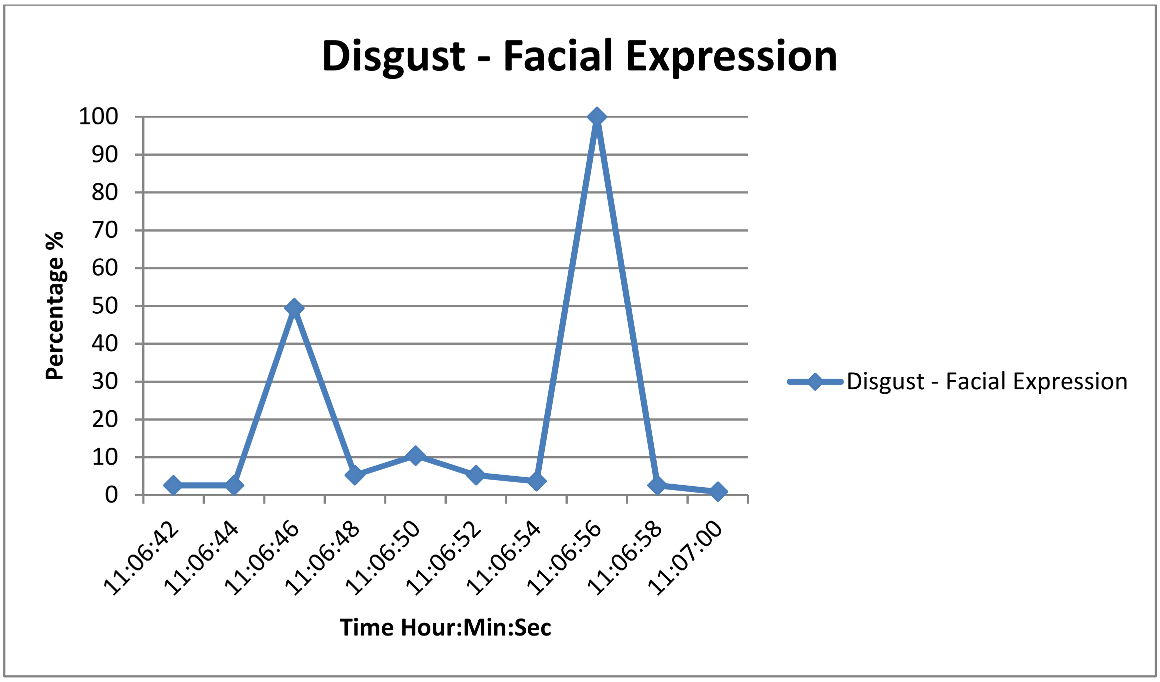


Figure 24 – Disgust facial expression value taken during Scenario 1

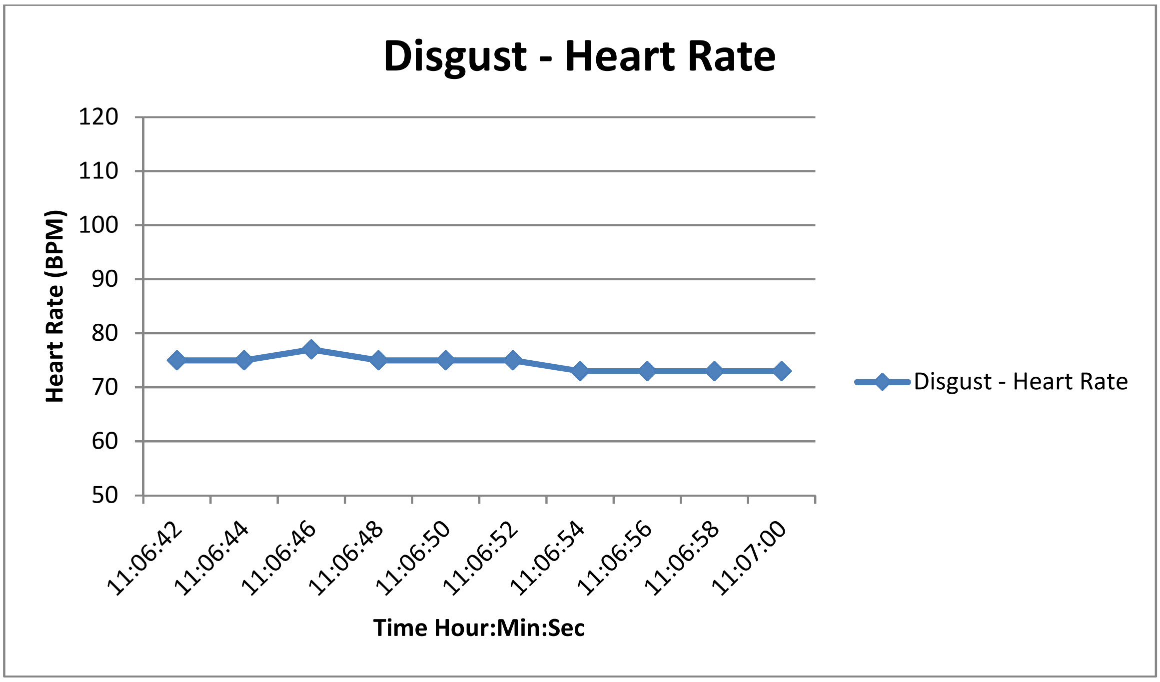


Figure 25 – Heart rate taken during Scenario 1

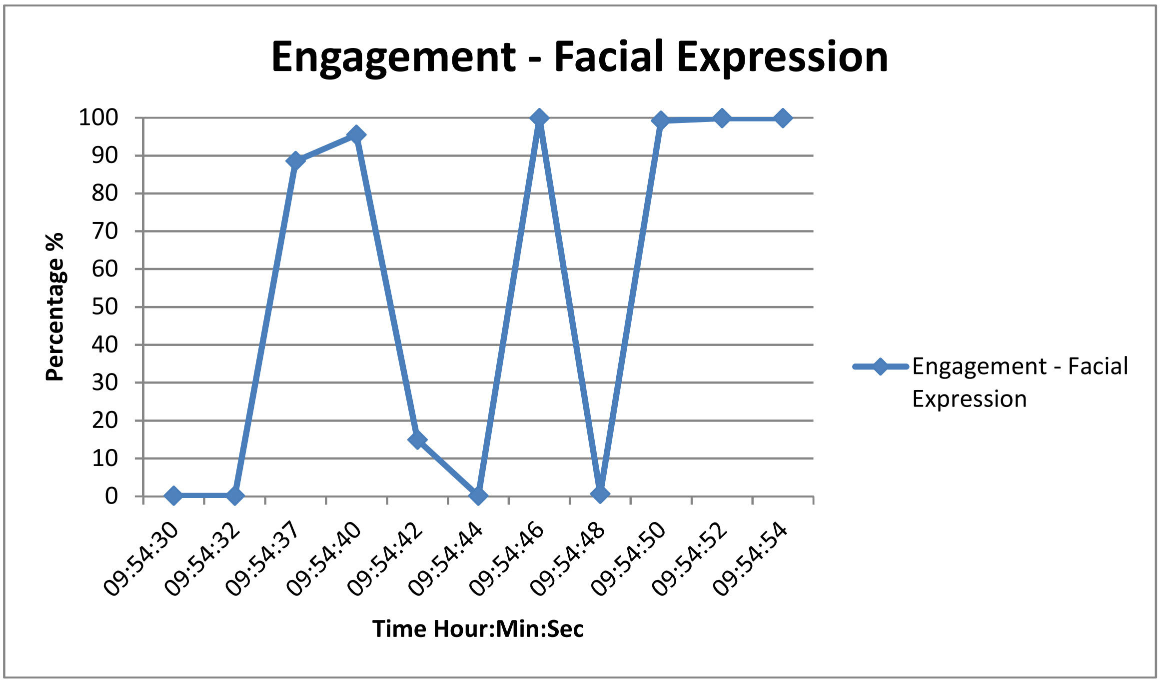


Figure 26 – Engagement facial expression value taken during Scenario 1

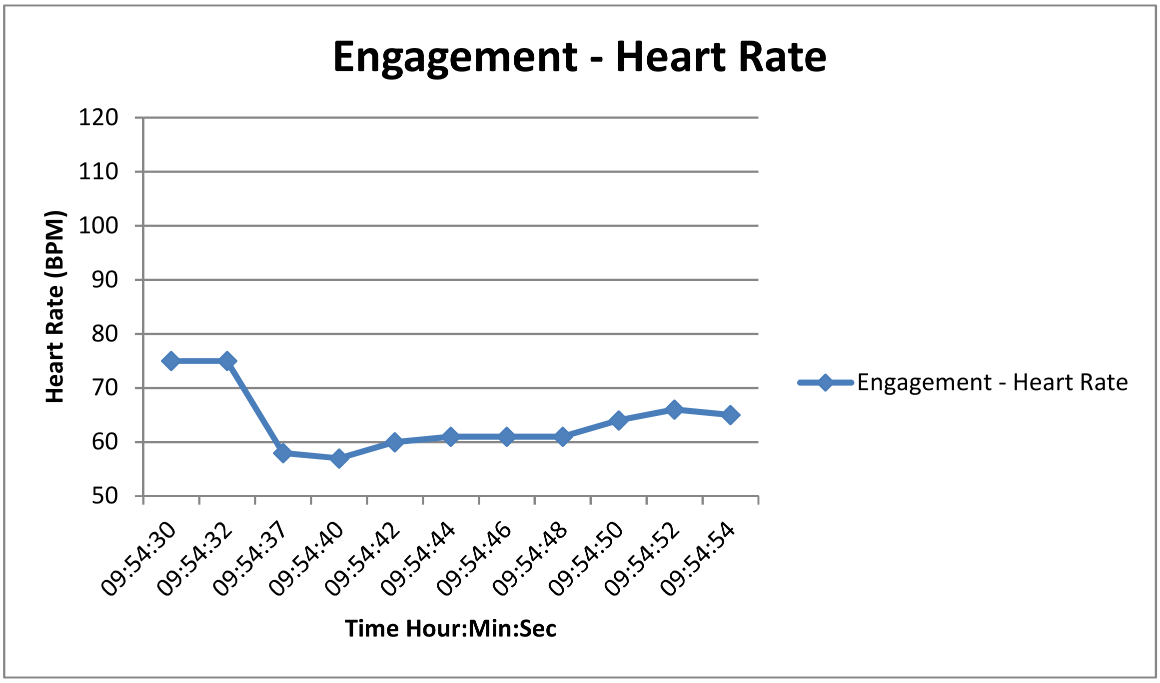


Figure 27 – Heart rate taken during Scenario 1

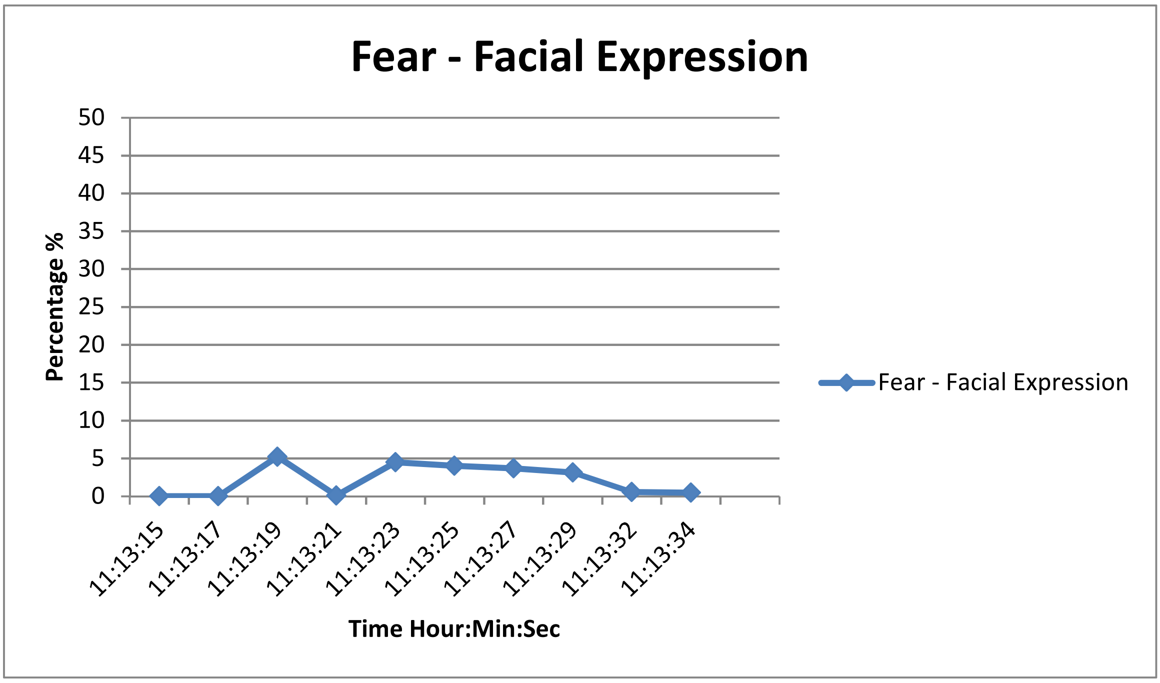


Figure 28 – Fear facial expression taken during Scenario 3

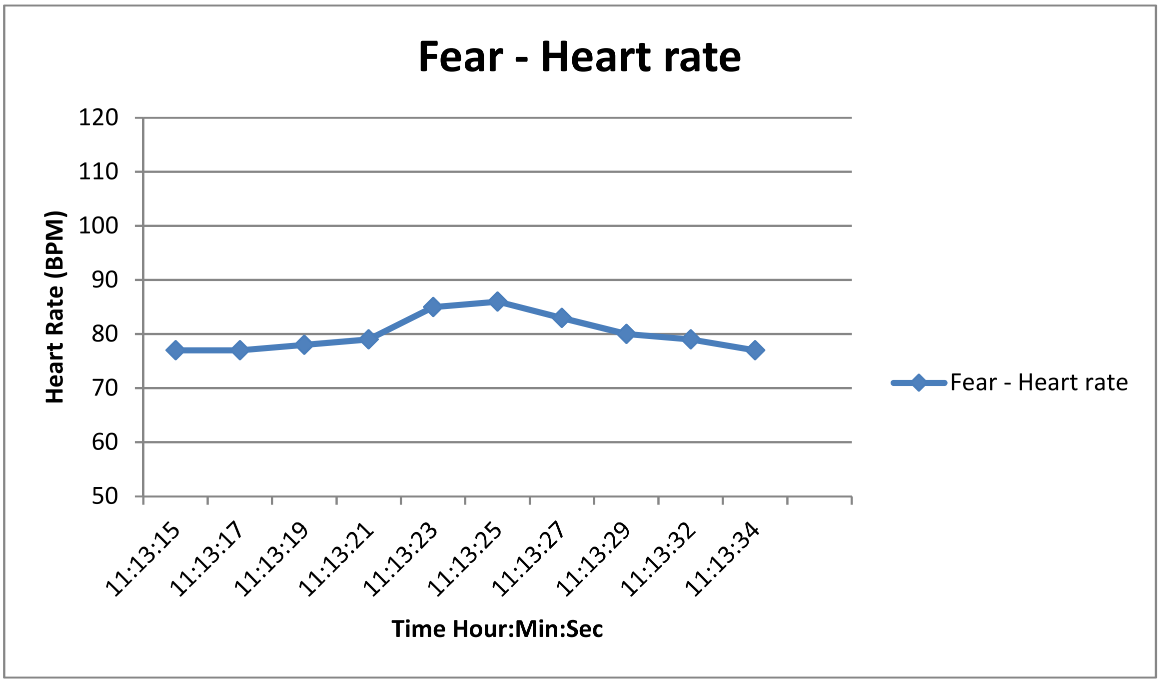


Figure 29 – Heart rate taken during Scenario 3

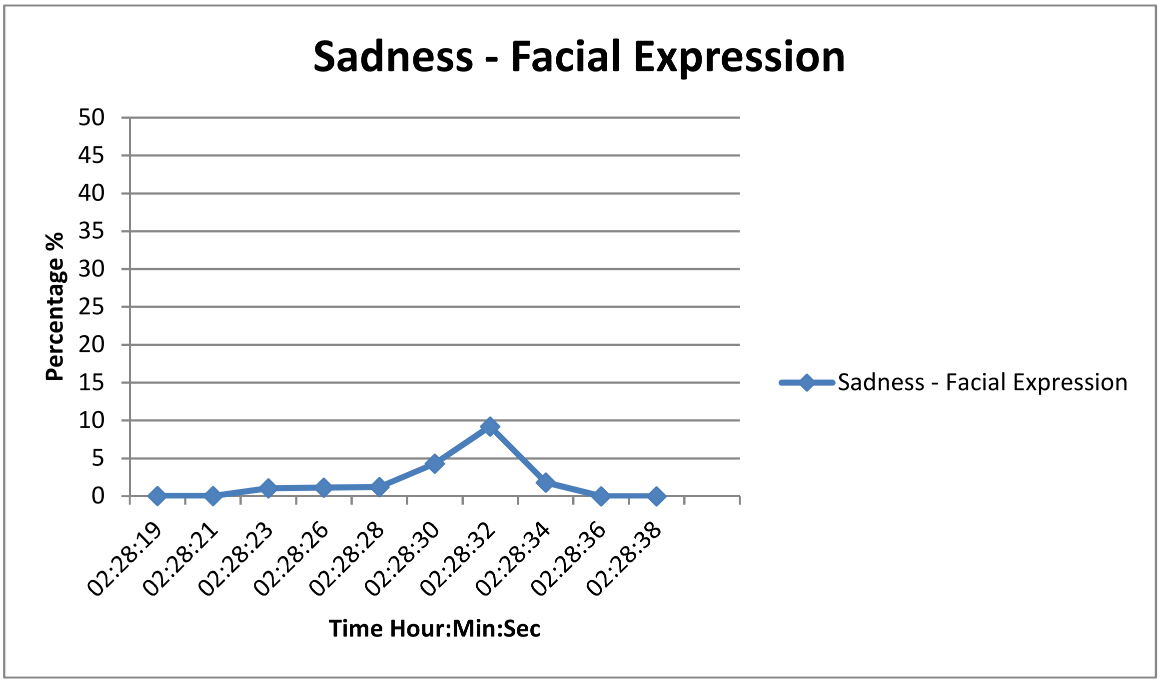


Figure 30 – Sadness facial expression value taken during Scenario 1

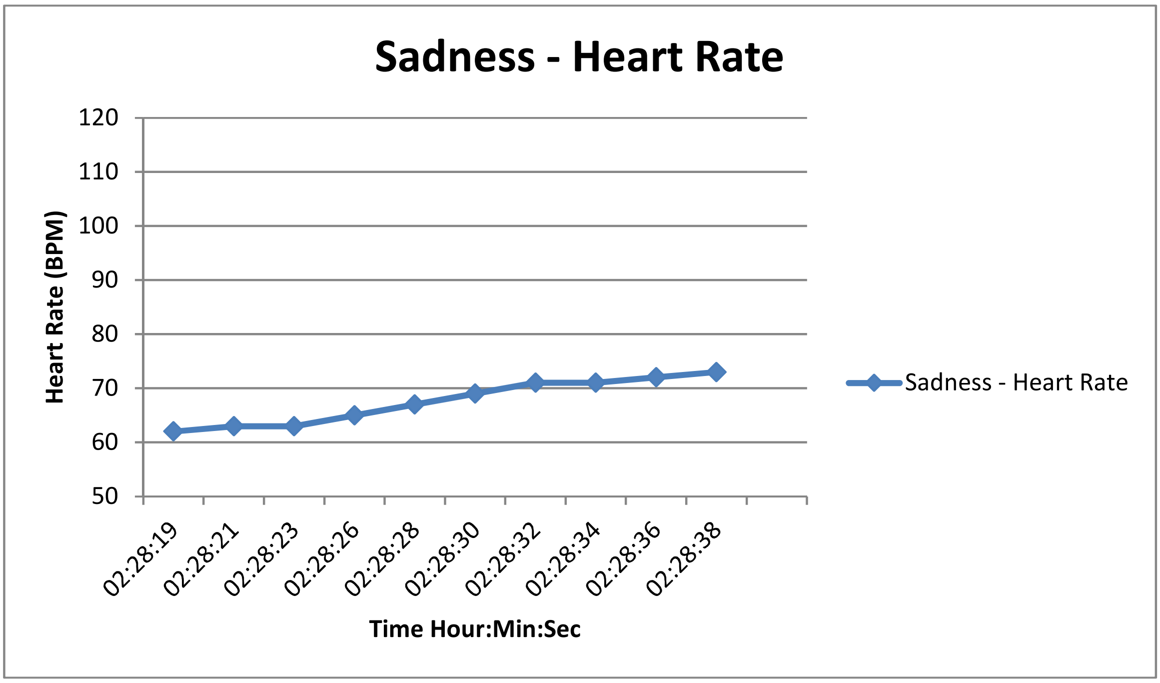


Figure 31 – Heart rate taken during Scenario 1

Consent Form

Please complete this consent form by ticking the appropriate boxes and signing below:

1.	I have had information about the project and the test explained to me by the researcher.	☐
2.	I have been given the opportunity to ask questions about the project and my participation.	☐
3.	I voluntarily agree to participate in the project.	☐
4.	I understand I can withdraw from the test at any time without giving any reason.	☐
5.	The procedures of confidentiality have been clearly explained to me (Use of names, data etc.)	☐
6.	The use of the data in research, publications, sharing and archiving has been explained to me.	☐
7.	I understand that other researchers will have access to this data only if they agree to preserve the confidentiality of the data and if they agree to the terms I have specified in this form.	☐
8.	I am comfortable with using biometric scanners (smart watch, facial recognition) and can ask to have them adjusted at any time.	☐
9.	That I have informed the researcher of any medical/mental issues that may impact the test (Epilepsy, Low mood, psychological disorders).	☐
10.	I am comfortable with being asked of my emotional state during the test and may choose to not answer the question at any given time.	☐
11.	Select only **one** of the following: I would like my details used and understand what I have said or written as part of this study will be used in reports, publications and other research outputs so that anything I have contributed to this project can be recognised. I do not want my details used in this project.	☐ ☐
12.	I, along with the researcher, agree to sign and date this informed consent form	☐

Participant:

Name of Participant ____________ Signature ____________ Date ____________

Researcher:

Name of Researcher ____________ Signature ____________ Date ____________

Figure 32 – Consent Form

Evaluating Different Techniques for

Improving Computer Games using Affective Computing

Thank you for participating in this test! The purpose of this questionnaire is to check the accuracy of the created application that you have just tested. The results of this questionnaire will be compared to the data gathered during the test and will be evaluated to see if the application is accurate or not. The questionnaire should only take around 5 minutes to complete and you can choose to answer as many or few of the questions as you would like. Please fill out the consent form and the profile section before starting.

Profile

Name: ______________________________

Age:

19-20 ☐ 21 – 29 ☐ 30+ ☐ Prefer not to say ☐

Gender:

Male ☐ Female ☐ Other ☐ Prefer not to say ☐

How often do you play computer games?

Never ☐ Once a week ☐ More than once a week ☐ Every day ☐

On average, how long do you play computer games in a single sitting?

Less than an hour ☐ 1 – 2 hours ☐ 2-6 hours ☐ 6+ hours ☐

What genre of games do you like to play? (tick as many as applies)

FPS (first person shooter) ☐ RPG (Role playing game) ☐ Action-adventure ☐

Platformer games ☐ Strategy ☐ Fighting ☐

Sports ☐ Horror ☐ Simulation ☐

MMO (Massively multiplayer online) ☐ Other: ________________________

Which Console/platforms do you play/own?

PC ☐ Xbox One ☐ Playstation 4 ☐

Nintendo Switch ☐ Nintendo 3DS ☐

Others:

Figure 33 – Profile Form

Test Questions

Please indicate which emotions you experienced during Scenario 1 (The farming map) with 0 being not at all and 5 being very strongly

Neutral ______ Joy ___ Sadness _____ Fear ___ Anger _____ Surprise ___ Engagement ___

Did you experience any other emotions while playing scenario 1?

Please indicate which emotions you experienced during Scenario 2 (The Village Attack) with 0 being not at all and 5 being very strongly

Neutral ______ Joy ___ Sadness _____ Fear ___ Anger _____ Surprise ___ Engagement ___

Did you experience any other emotions while playing scenario 2?

Please indicate which emotions you experienced during Scenario 3 (The horror map) with 0 being not at all and 5 being very strongly

Neutral ______ Joy ___ Sadness _____ Fear ___ Anger _____ Surprise ___ Engagement ___

Did you experience any other emotions while playing scenario 3?

Do you have any positive or negative comments about the game scenarios? (intensity, length of scenario, etc.)

Do you have any positive or negative comments about the test? (Equipment, settings of test, etc.)

Thank you for taking part in the test!

Figure 34 – Test Questions