University of Reading

Department of Computer Science

Vision-Based Emotion Detection of Gameplayers

Thomas Shavin Croos

Supervisor: Dr Hong Wei

A report submitted in partial fulfilment of the requirements of the University of Reading for the degree of Bachelor of Science in Computer Science.

Date: 21st April 2022

Declaration

I, Thomas Shavin Croos, of the Department of Computer Science, University of Reading, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of UoR and public with interest in teaching, learning and research.

Thomas Shavin Croos

Date 21st April 2022

Abstract

Gaming has been a vital part of our lives ever since its invention and introduction back in the 1970s, especially during the lockdown period as people were trying to some form of entertainment. However, whilst people were playing games during this time, they don't really get to see and grasp the impact the gameplay has on their emotions. This has made developing games difficult for the developers and designers as they must manage the emotional effects that the new games will have impacts on the players. Therefore, this project will aim to investigate how the length of the gameplay affects the player's emotions. To answer this aim, a program that is used to detect emotions will be built in Python. Volunteers will then take part in 3 sessions of 20 minutes, 40 minutes and 60 minutes in a 2-hour sitting to record the emotions whilst they play a chosen game. The findings returned from the experiment show that was an overall increase in the emotions produced by the volunteers from 20 minutes to 60 minutes. The project concluded that the longer sessions of gameplay did have a significant impact on the player's emotions when compared to a shorter gameplay session. Based on the findings, it was suggested that the game developers could consider designing and developing more shorter games that are exciting to play to reduce the possible emotional impact caused by playing computer and video games in the future.

Keywords: emotion detection, DeepFace, gameplay, variation, OpenCV

Acknowledgements

I would like to thank all the volunteers for participating in the experiments that were carried out. In addition to this, I would also like to thank Dr Hong Wei for guiding me during this project and my parents for giving me support during this project.

Contents

Introduction

Background

Over the last 50 years, video gaming has ultimately changed the entertainment world and has played quite a significant part in our lives. As a result of the video gaming invention and creation since the 1970s [1], the demand for games and gaming has significantly increased. This has been especially quite prevalent during the lockdown period as this saw lots of people isolate themselves and thus look for some form of entertainment. According to an article from Statistica [2], the number of games sold saw a total increase of 63% during the pandemic from March 16th to March 22nd at the beginning of 2020, which is a total of 4.3 million games being sold worldwide. This is further backed up by the fact that there was also a total increase in the like-for-like sales, which topped 44%. Video games are essentially electronic games that are designed to provide entertainment to players. They are interactive and can be used to access worlds that would not exist in real life, whether in 2D or 3D. These worlds are known as virtual environments and will have certain rules and guidelines depending on the genre and media format of the game. [3] Video games can come in different forms of media such as console games, personal computer (PC) games and mobile phone games. Some of the characteristics of these forms of video games may be more desirable to some than others, based on the player's interaction with the game. There are a variety of genres of games ranging from fighting games and survival games to party games and trivia. [4]

Problem Statement

Whilst the consumers play the games of their choosing, they don't often get to truly see how the gameplay affect their own emotions. As a result of this issue, these consumers don't consider reporting their emotions when sending reviews to the companies which create and design the games. As a result, the game creators and designers have a harder time in developing exciting, enjoyable video games whilst managing the emotional effects they will have on the gamers, as well as the demand and expectations of the consumers.

Aims and Objectives

This project will investigate out how the length of the gameplay affects the player's emotions. To meet the aim, the following objectives are set, which are to find out the emotion recognition of players during gameplay and the variations of such emotions during the gameplay. Finding out what emotions are present during the gameplay will help to understand what kinds of emotion are experienced whilst the player plays the chosen game and if the duration of the playtime will have such an impact on the emotions produced from the players. By finding out the variations of the emotions detected will help to understand how the player feels overall about the game chosen and if the time effects the way the player interacts with the game emotionally.

Solution Approach

To meet the aim of this project, a solution was drafted out to obtain the relevant data. This approach was drawn and designed initially on a flowchart to get the understanding of how the program will work and what conditions are needed to meet the objectives (Figure 1).

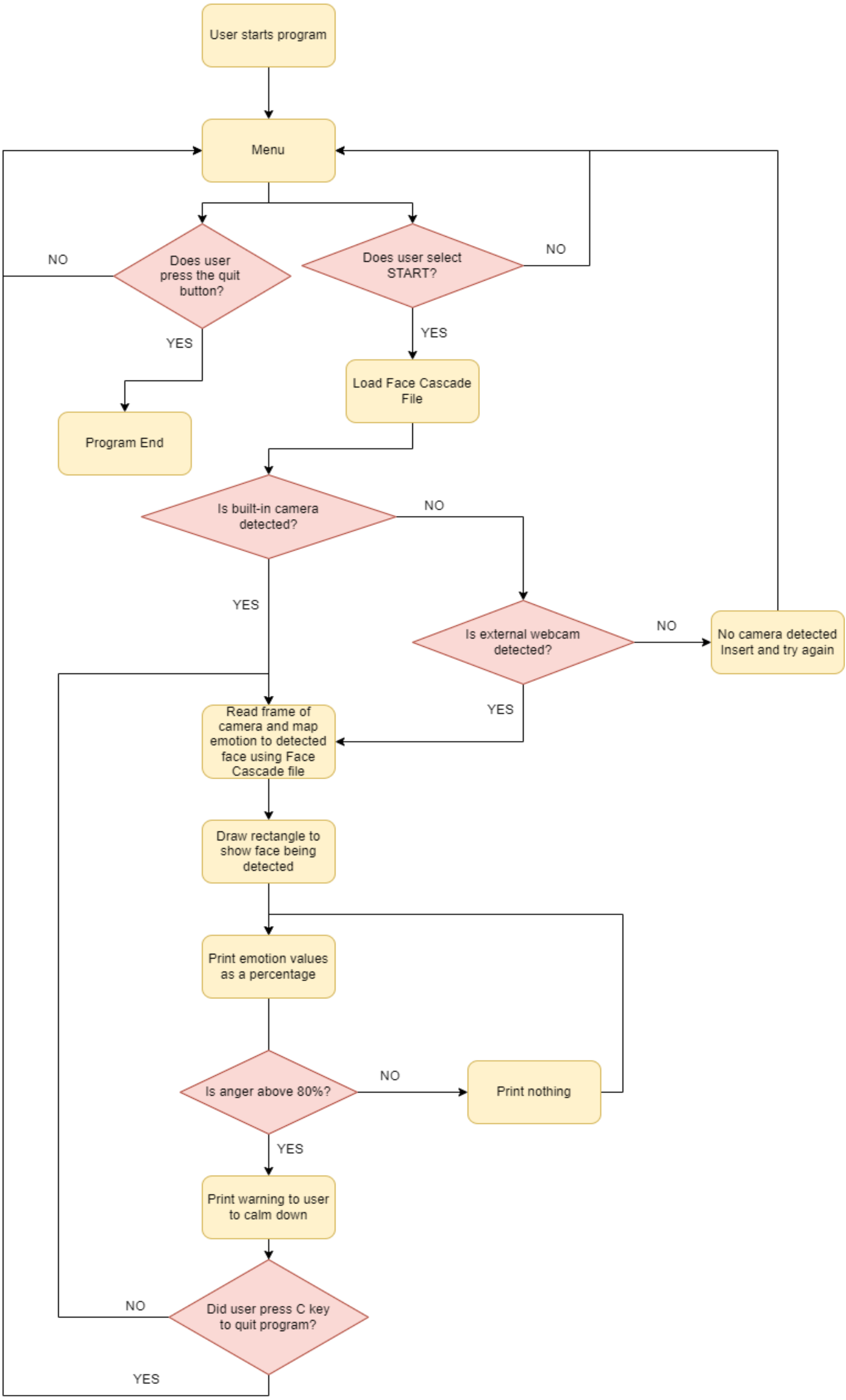


Figure 1*. This flowchart shows the flow of the program and how each of the components for this program are going to be designed.*

As the approach shows, the user will start up the emotion detection program by clicking on the START button. When this happens, the program will load up a file that contains the specific data relating to faces obtained from the internet. Once that happens, the program will try to detect for a web camera anywhere on the computer system. If the system cannot find the web camera, either internally or externally, it will tell the program to return to the main menu and tell the user to find a camera and try again. However, if one of the cameras are detected, the program will start recording the frames retrieved from the chosen camera and will map emotions to a face detected on the frames by utilising the file mentioned previously. When that happens, a rectangle is drawn around the face being detected.

When the emotions are detected, the program will write down the emotions as a percentage value to show the extent of the emotions that are being detected by the program. If the program detected anger above 80%, it would print a message to the user saying that they need to calm down a little and try and enjoy the game a bit better. Finally, the program will keep running until the "c" key is pressed, which closes the emotion detection program and return to the main menu. When the user presses the quit button on the main menu, the entire program will end.

Literature Review

Video gaming has already been shown to trigger emotional responses in people, especially around violent type games. A paper published by Hollingdale and Greitemeyer [5] explains that the chili sauce paradigm was used as a method of measurement to check their participants' emotions after playing both violent and non-violent games in offline and online modes. After this, questionnaires were conducted to obtain extra information about the emotions they experienced whilst playing the games. The results of this study suggested that there was an increase in the anger emotions when the participants were playing the violent game compared the neutral game, since the number of grams of chili sauce dispensed by the participants was heightened. The method that was used in collecting the emotional data for this paper was quite interesting as the extent of emotion could be deduced by the number of grams of chili sauce that the participants dispensed. However, the games chosen for the experiment, especially the ones for the violent game, were that of ones that most people may have played before the experiment, such as Call of Duty. In fact, some people may have played prequels of the games chosen. Because of this, the results may have been a bit more emotionally exaggerated and biased than they would have been had they played the game for the first time.

The experiment that was carried out by Hollingdale and Greitemeyer wasn't the only one that achieved similar results. A study by Hallema [6] was conducted to find out about the emotions experienced by participants, by using an eye-tracker to track the eye movement, whilst playing both violent and non-violent games. After this was done, a questionnaire was carried out to find out further information about the emotions that they had experienced during the gameplay. The study concluded that players who play violent games tended to blink less than those who played non-violent games. Furthermore, the study also gave indications that the anger and frustration emotions were triggered highly because of violent content and poor performance experienced by those who played the violent games. This suggested that there was a relationship between tracking the eye-movement of the participant and emotional response the participant produced towards the violent game. Whilst the study did use tracking the eye-movement as a factor in deciding what kind of emotions are experienced, in an era of new software and technologies coming out, this kind of method of deduction is only partially effective for understanding what kind of emotions are being detected and to what extent they are being detected. This suggests that emotion extent wasn't investigated into as much as the experiment that was conducted by Hollingdale and Greitemeyer. In similar fashion to the Hollingdale and Greitemeyer experiment, the games chosen were of ones most people have played before.

Another study done by Schrader and Nett [7] used a tower defence game, which taught about human liver functionalities, created by the researchers to find out about perception of control and the achievement emotions experienced by the participants based on the versions of the game that had either moderate, full or reduced control. The results of the study were obtained by carrying out a questionnaire after each of the three rounds of gameplay. This study had concluded that participants who had got the version of the game with either moderate or high control reported having a higher perception of control and experienced more happier emotions, compared to those who had received the version of the game with reduced control. Despite this finding however, it was also mentioned that the effects were not constant over the time of gameplay. This suggests that the length of time that one plays for, can also influence the emotions produced.

The method of carrying out a questionnaire after each of the rounds of gameplay was very crucial, as it allowed for a deep insight of what emotions were experienced by the participants. Furthermore, this data obtained was clean since it came directly from the participant. However, unlike with the previous 2 studies that had been mentioned, this study did not use any other method of collecting emotional data from the participants besides the questionnaire, such as using an emotion detection program. This means that no in-depth comparisons and validations could be made since no other methods were used during the study. This could lead to the results not being as accurate as they could have been otherwise. Moreover, there is a possibility that the some of the participants may have failed to mention certain emotions that they experienced during the gameplay.

In this literature review, there were some excellent methods that were used to detect emotions experienced by the participants during the gameplays. The chili sauce paradigm was an effective way of measuring the extent of the emotions produced during the gameplay and a factor that could be taken into careful consideration when it comes to building and designing the program. A key limitation of this literature is that there are many other ways of detecting emotions besides the methods that had been mentioned earlier, such as detecting facial emotions. Furthermore, none of the studies properly consider gameplay time as a key factor in the experiments that were carried out. Using this crucial point and some of the methods used in these studies, this will help to formulate an inclusive method into finding out how the emotions change over different time periods.

Methodology

To achieve the objectives that had been set out to meet the purpose of this project, the first task was to design and build the Emotion Recognition program in the chosen programming language. It was decided to build the program using the Python programming language because the programming language was quite simple and versatile to design and build with. Furthermore, this programming language could handle statistical data much better than other programming languages, which came in use for the experimentation stage later. Once the first task had been achieved, the next task was to carry out the main experiment, which was to record and collect data from the participants. It was decided to use modules to help me record and store the data during the periods of the experiment.

To begin with, the program was initially coded at a foundational level to check if it could simply access a given camera, be it an inbuilt webcam or an external USB webcam. This was done by importing the OpenCV module in the line `import cv2`. After importing this module, a variable called `cam` is called and is a assigned a function called `VideoCapture()` under the OpenCV module, where the value of 0 is placed inside this function as the index value (Figure 2). This line of code essentially looked for any default backend video capturing devices that are within the entire system, which in this case is the inbuilt webcam of the laptop.

```
cam = cv2.VideoCapture(0)
```

***Figure 2**. Accessing a default video capturing device, with 0 denoting that the function is set to default settings.*

If the program cannot find a default backend video capturing device to access and hence open it, the program will head on to the next step, which is to reassign the `cam` variable to a new `VideoCapture()` function call. In this instance, the value of 1 is placed inside this function as the new index value (Figure 3). This line of code will this time look for any external video capturing devices that have been connected and assigned to the number 1 slot of external video capturing devices, which in this case is the external USB webcam.

```
if not cam.isOpened():
    cam = cv2.VideoCapture(1)
```

***Figure 3**. Accessing an external video capturing device if a default backend video capturing device could not be found and accessed.*

However, if no video capturing device is detected on the system, whether it be from an internal or external source, a raised exception error and printed error will be executed (Figure 4). This will tell the user that the program cannot find a video capturing device to connect to and that the user should try to find one, connect it and try the program again.

```
if not cam.isOpened():
    raise IOError("Error! Camera not detected or cannot access Camera! Please fix the issue and try again!")
    print("Error! Inbuilt camera not detected or cannot access inbuilt camera! Please fix the issue and try again!")
```

***Figure 4**. Program will execute this error if a video capturing device, internal or external, cannot be found and accessed.*

If a video capturing device is detected on the system, the program will enter a while loop and read the captured data from the chosen video capturing device (either from an inbuilt webcam or an external webcam) through the `read()` function (Figure 5). This function essentially combines two other functions in the OpenCV module in one call, which are `grab()` and `retrieve()`. `grab()` grabs a frame recorded from the video capturing device, returning true if frame grabbed successfully, and `retrieve()` retrieves the grabbed frame, decodes the frame and then returns the frame. After the `read()` function has been executed, the returned frame is saved to the variable called `frame`. Another variable called `ret` is also called as a Boolean check to see if the video capturing device is on or not.

```
while True:
    ret, frame = cam.read()
```

***Figure 5**. This function will grab and retrieve the frame from the video capturing device before returning it again.*

Once this is done, the variable `frame` is resized to an appropriate size for viewing using the `resize()` function and assigned to the new variable called `output`. After this, the variable `output`, which contains the returned frame obtained by the `read()` function prior, is displayed as a window to the user by the function `imshow()` (Figures 6a & 6b).

```
output = cv2.resize(frame, (1920,1080))
cv2.imshow("ERS Visual V2.0 - Emotion Recognition System", output)
```

Figure 6a. *Code that resizes the retrieved frame, which is set to the screen size of a HP Omen 15 laptop.*

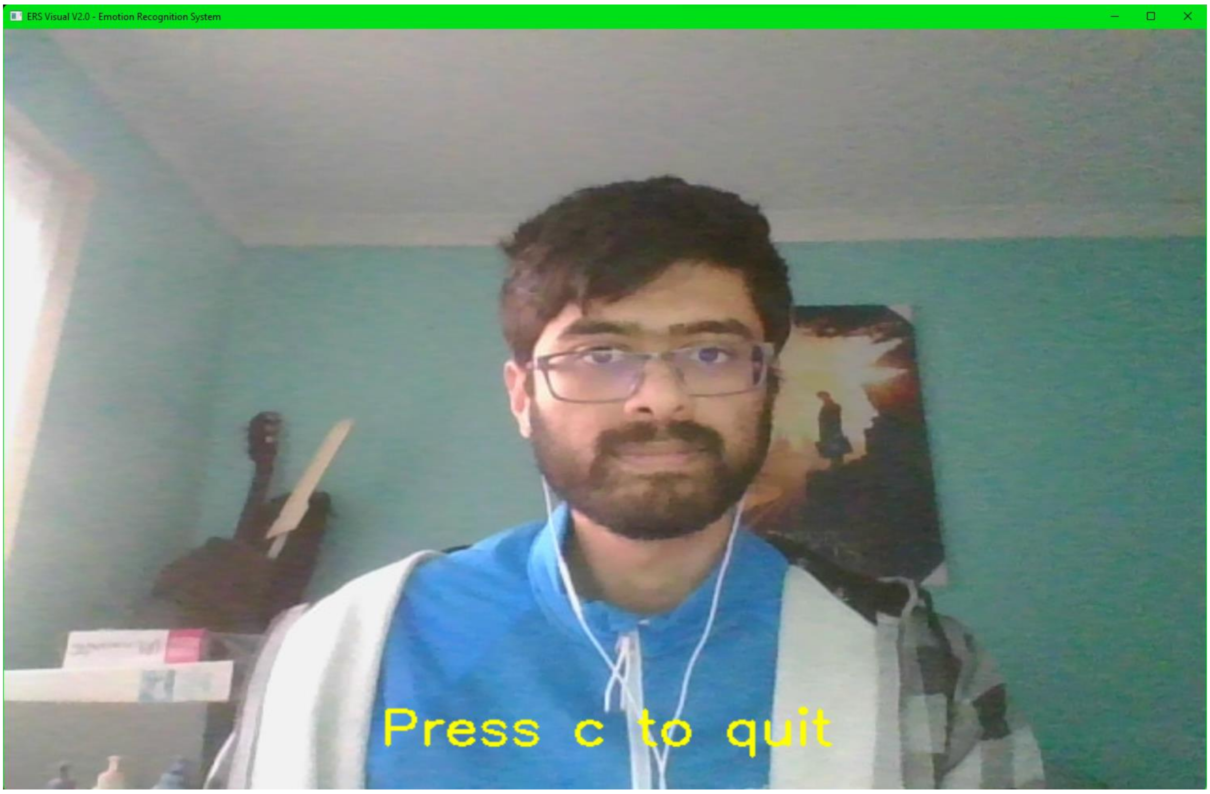


Figure 6b. *The result of that code being executed. Please note the "Press c to quit" in the window will provide a visual cue of how to exit out the program, which the mechanics of how that works will be explained next.*

To ensure that the user could exit the program safely after using it, an if statement is implemented which contains the `waitKey()` function and the bit mask 0xFF (Figure 7). This will check to see if the user presses the 'c' key on their keyboard and if so, the `waitKey()` function will return the corresponding binary value of the 'c' key. This is expressed as 0b01100011. The bit mask is expressed as 0b11111111 in binary and is there to turn the left 24 bits to 0, as the `ord()` function can only return a value in the range 0-255. This is due to the limited character set that the user's keyboard has. By following the rules of logic operators, the AND operator is applied to the binary values obtained from the `waitKey()` function and the bit mask. The resulting binary value returned from this logic operator is 0b01100011, which is the same as the binary value of the 'c' key. Once the logic operation had been completed, the resulting binary value is compared with the `ord()` function as an expression using the == operator. The `ord()` function contains 'c' in the brackets and returns the Unicode value of lowercase c, which is U+0063. Since 0b01100011 and U+0063 are both the same value of 99 in decimal format, overall expression is therefore true and thus the program breaks out of the while loop. The connection to the video capturing device is closed off through the `release()` function and the windows are destroyed thereafter through the function called `destroyAllWindows()`.

```
if cv2.waitKey(1) & 0xFF == ord('c'):
    break

    cam.release()
    cv2.destroyAllWindows()
```

Figure 7. *If statement checking to see if the 'c' key was pressed by the user, which will lead to the termination of the program upon detection. The termination occurs by releasing the webcam and destroying the windows that were open during the operation of the program.*

Once the test of accessing a given webcam had been completed, the next step was to make sure that the program could pick up on a face upon the user entering the viewing range of the camera. To achieve this, the program needed what is known as a Haar Cascade classifier. This is essentially a powerful method of object detection that was suggested by Paul Viola and Michael Jones in their research study. [8] It is a machine learning method by which a cascade function is trained from loads of images obtained from public domain sources, in both positive and negative forms. [9] The positive images are the images that the classifier will use to identify the objects relevant to the use case. The negative images are the images of other things which doesn't contain the objects relevant to the use case that the classifier will try to detect for. By utilising the training, the classifier is then able to identify the objects in other images. [9] The classifiers are essentially massive .xml files which contains lots of feature sets. These .xml files coincide to a unique type of use case, which in this case is the frontal face.

Since the program is going to focus on detecting emotions of individuals, a Haar Cascade classifier that will pick up feature sets that are associated with the font of the face (e.g. eyes, nose, mouth etc.) is needed. Therefore, the classifier `haarcascade_frontalface_default.xml` is used for this program. To call this .xml file into the program (Figure 8), the `CascadeClassifier()` function of the OpenCV module is called, along with a special variable with the folder name containing the .xml file called `cv2.data.haarcascades` and the string (name) of the .xml file. This result of this function is then assigned to a variable called `faceCascade`.

```
faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
```

Figure 8. `CascadeClassifier()` *function called to utilise the front face classifier to help detect the faces.*

Once the classifier is called into the program, the frame obtained from the `read()` function is passed into the `cvtColor()` function to convert it to grayscale (Figure 9). The grayscale conversion occurs with the colour conversion code `cv2.COLOR_BGR2GRAY`. Normally, the images that people normally see are seen in the RGB (Red, Green, Blue) channel format. When an RGB image is read by OpenCV, the image ends up being kept in the BGR (Blue, Green, Red) channel format. In this scenario, the face is the identifying aspect to be detected and so the BGR channel is converted to grayscale. This is because processing the gray channel is much simpler than the BGR channel and is therefore less intensive computationally as only 1 channel of black-white is contained. [9] Once the grayscale conversion is complete, the resulting frame is then stored in the variable gray.

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

Figure 9. *Grayscale conversion of frame carried out for simpler detection.*

Next, the process of trying to identify the faces in the frame is carried out. The `faceCascade` variable, which contains the `haarcascade_frontalface_default.xml` file, is called again. This time, a function is called after calling `faceCascade`, which is called `detectMultiScale()` (Figure 10). The function will utilise features obtained from the `faceCascade` variable to identify the features of the frame retrieved. In this function, several parameters are passed into it. The first parameter to be passed will be the variable that contains the grayscaled frame, which is gray in this scenario. Next, the `scaleFactor` parameter is determined, which dictates the reduction of the image size based on every image scale. This means that this parameter allows one to create their scale ratio to decide what size of images are returned. To further elaborate, the model obtained earlier has a set size outlined during training, which is shown in the .xml file. This means that only that set size of the face will be picked up by the program if it is present on the image (frame) provided.

However, if a scale factor to the image being inputted is applied, bigger faces can be scaled down to smaller ones. This will increase the chances of algorithm detecting a wider range of different sized faces on the provided image. This parameter has a drawback as changing the scale factor can also change the speed and effectiveness of the algorithm. For this reason, a compromise between the scaling of images and speed of the algorithm had to be reached. Therefore, the `scaleFactor` parameter was set to 1.1 as this was found to reduce the input image size by 10% and increase the chance of detection. The decided value also keeps the algorithm working at an appropriate speed, allowing for a smoother detection of the face.

Lastly, the `minNeighbors` parameter is passed, which determines the number of neighbours each candidate rectangle should have to retain it. This means that the quality of the faces picked up by the program will be determined by this parameter. If this parameter is set at a high level, only a few high-quality detections will be made by the program. The recommended values, according to the OpenCV documentation, fall in the range of 3-6. For this program, the decided value for the `minNeighbors` parameter was 4 as most of the detections made with this value were of high quality.

```
faces = faceCascade.detectMultiScale(gray, 1.1, 4)
```

***Figure 10**. `detectMultiScale()` function called to appropriately scale the images for detection.*

Upon execution of the `detectMultiScale()` function, 4 values are returned. These values correspond to the detected feature of the face, which are the x-coordinate, the y-coordinate, the width(w) and the height(h). Using these values obtained, a rectangle is drawn around the face that appears in the frame (Figures 11a and 11b). This is done through passing the 4 values in the for loop and using the function `cv2.rectangle`.

```
for(x, y, w, h) in faces:
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

***Figure 11a**. The 4 values obtained from `detectMultiScale()` are being passed through the for loop to draw the rectangle on the frame when face is detected.*

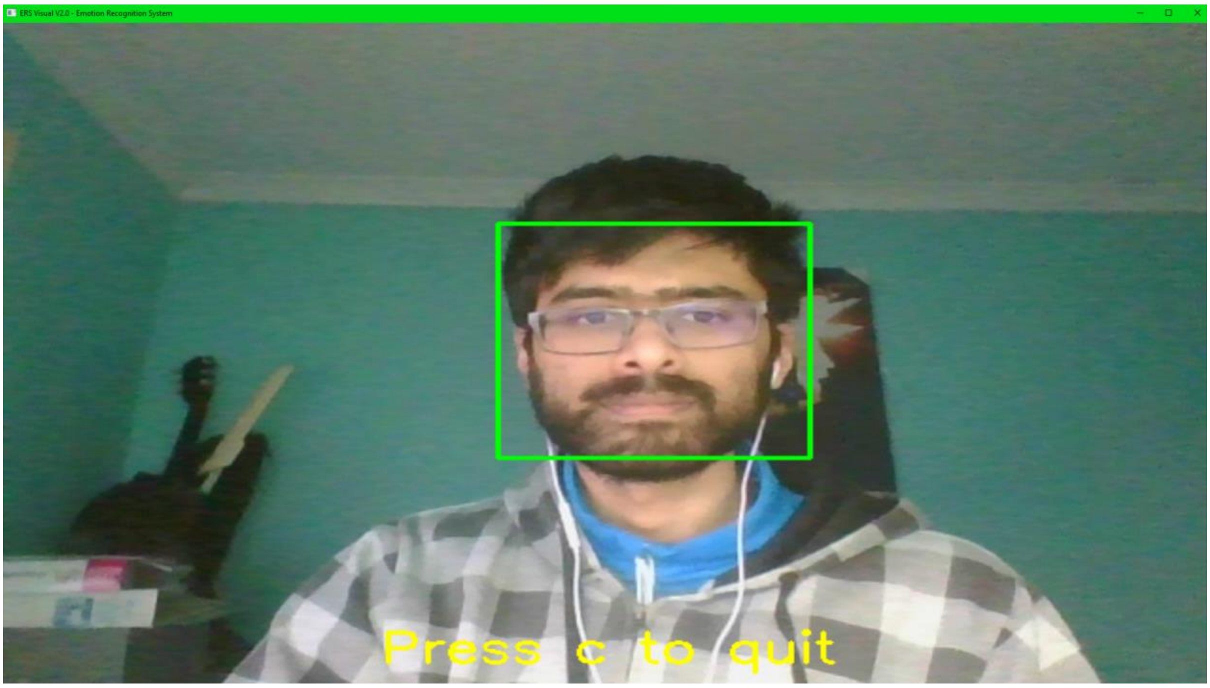


***Figure 11b**. The result of the detection of the face.*

After getting the webcam to turn on and getting the face detection to work, the principal step of building the emotion detection aspect of the program now comes into play. For this program, the module called DeepFace [10] is used to get the emotions based on reading a face. DeepFace is a "hybrid face recognition framework that uses state-of-the-art models for analysis such as VGG-Face, Google Facenet, Facebook Deepface, all wrapped together in one" [11]. The reason for choosing DeepFace as the main framework for the emotion detection is because, according to an article on DeepFace [11], the "accuracy of identification goes up to 97% and has proven to be more efficient in detecting faces than other average face recognition frameworks". On top of this, the framework can handle most external factors that may affect the effectiveness of the emotion detection, such as the amount of light that is detected in the room the emotion detection is being carried out in.

The DeepFace module is called into the program through the line `from deepface import DeepFace`. After this is done, a function called `analyze()` is called from the DeepFace module (Figure 12) into the while loop. The function works by taking the frame parameter and applying emotion action parameter to the frame. Another parameter is `enforce_detection`, which is set to False to avoid returning exception errors for valid faces in the frame. Applying the emotion action parameter will return all 7 emotions of the face being detected on the frame, which are angry, fear, neutral, sad, disgust, happy and surprise. These emotions are then stored in a variable called `emotRes`.

```
emotRes = DeepFace.analyze(frame, actions = ['emotion'], enforce_detection = False)
```

***Figure 12**. `DeepFace.analyze()` used to retrieve the information on the emotions detected on faces that come into view of the camera.*

All that needs to happen now is to write the emotions as text on the frame. In order for this to happen, a function is called from OpenCV, called `putText()`. For this program, it was decided that the emotions would be displayed as percentage to show the extent of the type of emotion the user was giving. In the `putText()` function, several parameters are called, which are the frame, `emotRes` (as a percentage rounded to 1.d.p) and other parameters to do with the formatting and positioning of the text on the frame. This `putText()` function is repeated several times for each emotion, following the same template for passing the parameters as mentioned previously (Figures 13a and 13b).

```
cv2.putText(frame, 'Angry: ' + str('%.1f' % (emotRes['emotion'].get('angry'))) + '%', (10, 20), font, 0.5, (255, 0, 0), 2, cv2.LINE_4)
cv2.putText(frame, 'Fear: ' + str('%.1f' % (emotRes['emotion'].get('fear'))) + '%', (10, 40), font, 0.5, (255, 0, 0), 2, cv2.LINE_4)
cv2.putText(frame, 'Neutral: ' + str('%.1f' % (emotRes['emotion'].get('neutral'))) + '%', (10, 60), font, 0.5, (255, 0, 0), 2, cv2.LINE_4)
cv2.putText(frame, 'Sad: ' + str('%.1f' % (emotRes['emotion'].get('sad'))) + '%', (10, 80), font, 0.5, (255, 0, 0), 2, cv2.LINE_4)
cv2.putText(frame, 'Disgust: ' + str('%.1f' % (emotRes['emotion'].get('disgust'))) + '%', (10, 100), font, 0.5, (255, 0, 0), 2, cv2.LINE_4)
cv2.putText(frame, 'Happy: ' + str('%.1f' % (emotRes['emotion'].get('happy'))) + '%', (10, 120), font, 0.5, (255, 0, 0), 2, cv2.LINE_4)
cv2.putText(frame, 'Surprise: ' + str('%.1f' % (emotRes['emotion'].get('surprise'))) + '%', (10, 140), font, 0.5, (255, 0, 0), 2, cv2.LINE_4)
```

***Figure 13a**. The function cv2.putText() is applied 7 times to get the percentage value of each emotion individually.*

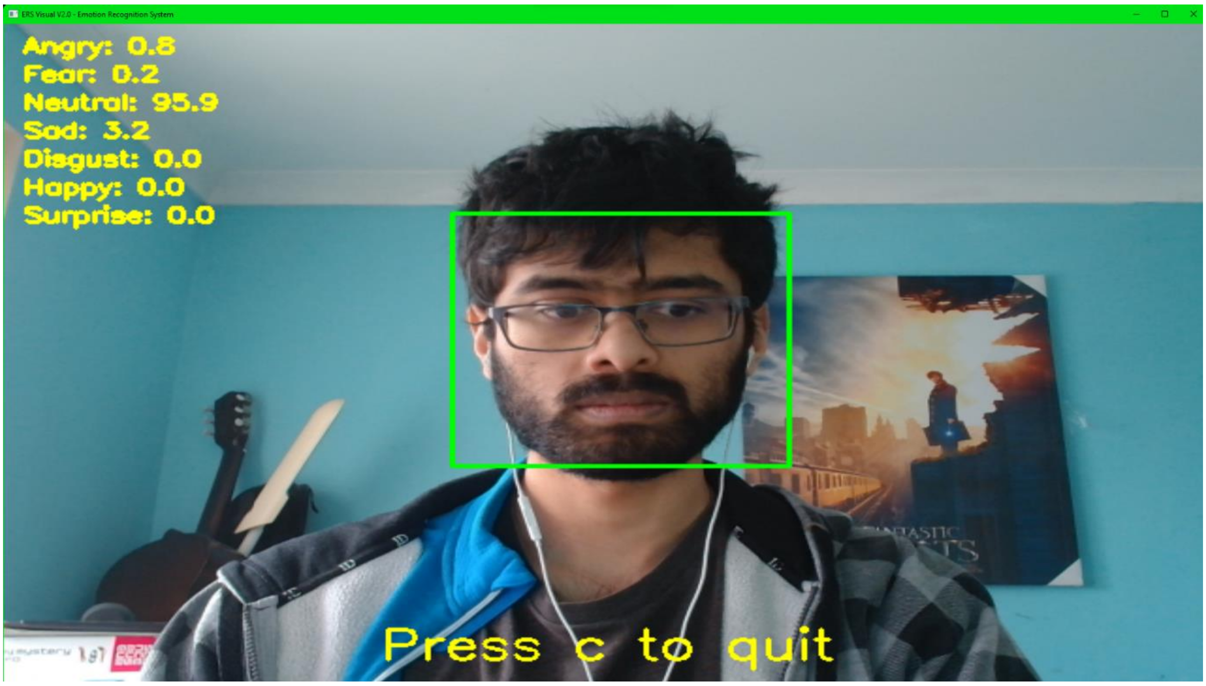


***Figure 13b**. The result of cv2.putText() being applied to the emotions.*

As the program tended to detect other objects which it saw as other faces in the background, another module was imported to fix this issue, called `cvzone`. Through `cvzone`, a class was imported into the program called `SelfiSegmentation`, which is designed to carry out the basic background subtraction of an image. This then was assigned to a variable called `segmentor`. The background subtraction process was then carried out by calling a function of `SelfiSegmention` called `removeBG()` (Figure 14a), which called 3 parameters into itself. These were the `frame` variable, the colour of the background to be replaced after background subtraction had been completed and the threshold, which determines the amount of the cut around an object that is in view of the camera in the foreground. If this value is set to 1, then everything is cut out from the frame, so this value is set to 0.825 to ensure that the user stays in frame when using the program (Figure 14b). The `frame` variable is then reassigned to this function to apply the background subtraction.

```
frame = segmentor.removeBG(frame, (218, 197, 99), threshold=0.825)
```

***Figure 14a**. The function removeBG() will apply the background subtraction to the frame.*

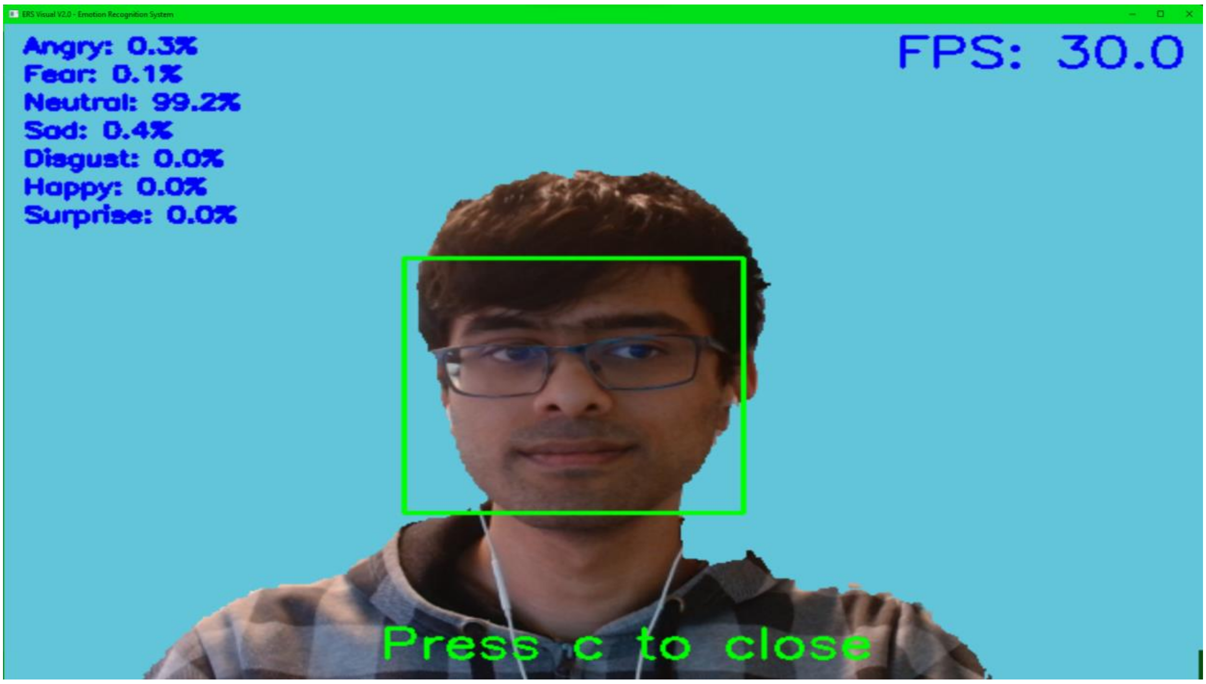


***Figure 14b**. The output of the function being executed.*

An extra feature in the program included telling the user when they are getting too angry, so that they should try to enjoy the game they were playing a bit better (Figure 15a). This reminder only triggered when the percentage of the angry emotion surpassed 80%. Another feature included telling the user when they are being happy. This printed a message saying that they are doing well, and that they should continue to enjoy the game (Figure 15b). The message only triggered when the percentage of the happy emotion surpassed 70%.

```
if(emotRes['emotion'].get('angry') > 80):
    cv2.putText(frame, 'Angry: ' + str('%.1f' % (emotRes['emotion'].get('angry'))) + '%', (10, 20), font, 0.5, (0, 0, 255), 2, cv2.LINE_4)
    cv2.putText(frame, 'You are getting a bit too angry, please try to enjoy the game!', (75, 400), font, 0.5, (0, 0, 255), 2, cv2.LINE_4)
    print("WARNING! Anger levels detected above set threshold of 80%!\n")
```

***Figure 15a**. A message alerts the user when they are getting too angry.*

```
if(emotRes['emotion'].get('happy') > 70):
    cv2.putText(frame, 'Happy: ' + str('%.1f' % (emotRes['emotion'].get('happy'))) + '%', (10, 120), font, 0.5, (0, 255, 0), 2, cv2.LINE_4)
    cv2.putText(frame, 'Great job, keep enjoying the game!', (185, 400), font, 0.5, (0, 255, 0), 2, cv2.LINE_4)
    print("NOTICE! Happy levels deteced above set threshold of 70%! Great job, please keep it up!\n")
```

***Figure 15b**. A message tells the player when they are happy, encouraging the player to carry on playing.*

With that, the main emotion detection side of the program has been completed. After this, the menu for the program was designed to give the program a bit of an aesthetic look. This was simply done by importing the Tkinter module into the program and then using functions to create separate windows for the menu itself, the main emotion detection program, the about page and the credits page (Figure 16). To see how the menu operates, you can view the entire code at the end of this report by following the link to the Git repository for this project (see Appendix 1).

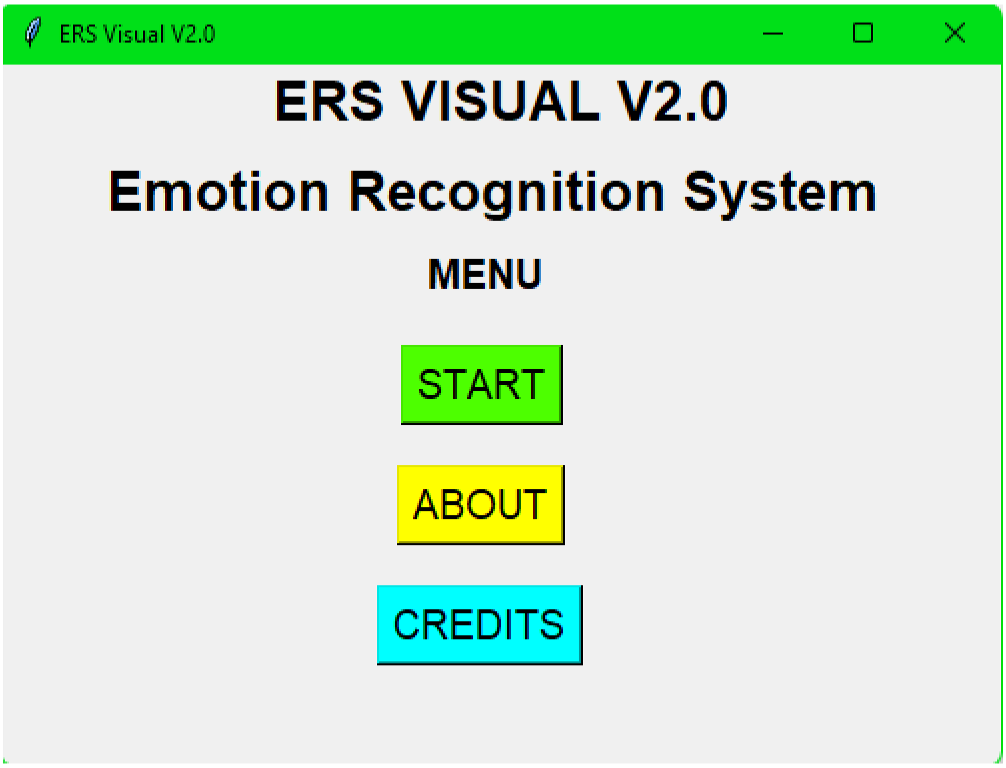


Figure 16. *The result of using Tkinter to design a generic menu for the emotion detection program.*

The next part was to modify the program to be ready for the implementation of the experimentation stage. Since the project aims to find out how the length of the gameplay affects the players' emotions, the modifications had to be designed in a manner that it would try to achieve these aims. The objectives of understanding about the emotions detected from players during the gameplay and the variations of such emotions during the gameplay would further help in considering the modifications that need to be made. With all the aspects considered, it was decided that modifications would be made in accordance with the objectives and the aim.

The modification of the program began with importing new modules that would be required for the recording and storing of the data collected. These modules were `datetime` and `OS`. The `OS` module was imported to create and access the folders and text files in a specific location, called a directory, during the experiments. The datetime module was imported since time would be recorded and measured during the experiments.

With the `OS` module imported, the next step of the modification to was create the location of where the text files of the data recorded and collected were going to be written and saved to. This part of the process happens before the while loop begins (the emotion detection algorithm). Firstly, a variable called `name` is initialised, where the name of the folder that is to be created is inputted using the `input()` function. When this happens, the inputted result is stored in the `name` variable and is then converted to a string object using the `str()` function. The returned string of this function is the reassigned to the new variable called `child_dir`. This new variable is simply the name of the folder where the text files containing the data collected and recorded are going to be stored in.

To properly find and pinpoint the location of where that folder is going to be placed, it is essential to create what is known as a directory to the folder. What this means is that a path of folders needs to be created that will lead to the data collection folder (`child_dir`), so that the program knows where exactly to write and save the text files to. This will be created as a new variable called `parent_dir` and will contain the string value of the directory where the data collection folder will be saved to. After this is done, the actual directory needs to be created. This happens by calling the `mkdir()` function of the OS module. In this function, a method called `os.path.join()` is passed, which itself calls 2 parameters. These parameters are variables that were initialised earlier, `parent_dir` and `child_dir` respectively. A print statement is also added to confirm that the data collection folder had been successfully created.

Now that the proper location for the data recording and collection had been established, a method of opening, writing and saving the text files needs to be built next. This is done by initialising a new variable, called `txt`, to open a text file in the data collection folder that had been created previously. This happens by a function called `open()`, where 2 parameters are passed. The first parameter passes the full path of where the text file is going to be opened, including the name of the text file and the .txt extension after to ensure that the file is saved as a text file. The second parameter defines the mode that the file being opened is in, which in this scenario will be opened for writing in. As this parameter only accepts this mode in string form, this will be written as "`w`".

Following this, a method to write all the data headings and the data itself needs to be called. This is achieved through the use of the `write()` method. The headers are written in the form of column headers, which include the frame number, time of when the frame was recorded and the 7 emotions that are to be recorded during the experiment. To evenly space out the headers between themselves, `\t` was added appropriately to create tabs in the text. Since frame number is going to be a variable that needs to be recorded, a new variable called `f` is initialised and is set to 0. All the folder and file setup happened before the while loop is initialised.

As the program works by retrieving lots of frames during runtime, the time is measured by retrieving the local time that the user device is set to, which in this case is the UK time. This retrieving process is done through calling the `now()` function of the datetime constructor, which is in the `datetime` module. The function is called within the while loop to ensure that the time only keeps recording if the while loop is not terminated. The result of this function returns the year, month, day, hour, minute, second, and microsecond, which is stored in the new variable `ct`. However, since only the time is the key factor here in this experiment, only the hour, minute and second values should be kept. This is done by calling the `strftime()` method, which returns a string of the hour, minute and second values and stores them in the new variable called `current_time`.

At the same time, another `write()` method is called, which passes the `f` variable and `current_time` variable as strings, as well as all 7 emotional values retrieved from the `emotRes` variable. Unlike the emotional values printed as percentages on the screen, the emotional values collected in the text file will be given in 8.d.p. This `write()` method will be executed for each frame for as long as the while loop is running. The `f` variable will also be incremented each time the `write()` method is executed to increase the frame number. When the while loop is broken out of, the `close()` function is called to save and close the text file safely after the experiments have finished. The coding behind the opening, writing, closing and saving of the text files generated during the experiment can once again be seen in full code (Appendix 1).

After the implementing the modifications that were required for the experimentation, the next step was to design and setup the experiments to be carried out. The designing of the experiment began with considering the variables that could possibly have an effect during the process of undertaking the experiments. There are 3 types of variables had to be taken into consideration, which are called the dependant, independent and extraneous variables. The dependant variables are the effects of the experiment, in which the value of it is influenced by any alterations made in the independent variable. In this scenario, the dependent variables were looking for the emotions that are detected by the program and the variation of the emotions being detected.

The independent variable is the variable that is manipulated, which in turn is supposed to create an impact on the dependant variables. The value of this variable does not depend on other variables used in the experiment. In this instance the independent variable was the length of the gameplay. The final type of variable that had to be taken into consideration was what is known as extraneous variables. These types of variables are ones that are not investigated into to but could have an impact on the results collected in the experiment. In this case, the extraneous variables included the following:

- The game chosen for the volunteers to play during the experiment, as some of the volunteers may have played the game before the experiment.
- The location of experiment as other people may be present that may accidentally impact the emotions detected from the volunteer whilst they're playing the game.
- The location of the experiment also may also vary the lighting that is projected in the room, which can further influence the emotions detected from the volunteer. This was proven by a study done by Mario Fritz and Bernt Schiele [12] as they mention that face detection often fails if the background light is too much.

Now that the possible extraneous variables had been listed out for this experiment, solutions are considered as to how these variables can be controlled during the experiment. Firstly, to ensure that the game chosen was not one that they had played before, possible solutions were made through either by searching on the Internet to find any games that came out recently and people have yet to get or by looking at people's wishlists on gaming stores to find out what kind of games they would like to try out. By doing this, any emotional bias was eliminated since they didn't play the games that they had played before. Secondly, to eliminate the chance of people outside the experiment accidently entering the location of the experiment, the experiment was carried out at a private location. This significantly reduced any chances for rogue readings to be recorded during the experiment.

Lastly, to limit the effect of the lighting influencing the emotions detected from the volunteer, the room chosen for the experiment to be carried out was done in evenly lit place with little to no objects in the background. By doing this, the chances of any false detections were significantly reduced and thus the emotion detection returned more accurate readings. Considering all the possible variables that will come into play for this experiment, two hypotheses were made. The null hypothesis (H_0) was that as the gaming session time increases, the emotions of the players did not change at all. The alternative hypothesis (H_A) was that as the gaming session time increases, the emotions of the players reduced to the point of being neutral for majority of the time (getting bored after playing for a while or being focused on the game).

For the experiment, random volunteers who were into gaming on a regular basis were chosen. Since the numerical data on faces not from the public domain was going to be collected, the ethics approval had to be obtained so that the experiment could proceed. The setup of the experiment began with choosing the game for the experiment. The game chosen was called SplitGate, which was an action-adventure first-person shooter (FPS) game that had come out recently. For each volunteer, they played the game over 3 sessions in one sitting of 2 hours. During the sitting, the sessions they played for lasted for 20 minutes, 40 minutes and 60 minutes respectively.

The volunteers were asked to download the game prior to the sitting and to bring their own laptops to play the game. When the sessions were being undertaken, the emotion detection program was used to record their emotions (anger, happiness, sadness, surprise, fear, disgust and neutral) and the strength of the emotions produced, as numerical data. The data was stored in folders that were anonymised and encrypted to ensure that this data is not accessed by anyone else. At the same time, the values were presented as percentages on the screen to the volunteer. The volunteers were entitled to 5 – 10 minutes of break time between the sessions. This helped the volunteers to prepare for the next sessions mentally and physically so that they could play to the best of their ability.

At the end of the sitting, the volunteers were asked to fill out a form on Google Forms to describe and explain in more detail about how they felt during the gameplay over the different time sessions. This helped to obtain further data that was more in-depth about the emotions they experienced and make interesting comparisons between the data recorded during the sittings and the data obtained from the forms completed. Like with the data recorded during the sitting, the data obtained from the forms was anonymised.

Results

The emotions were recorded continuously throughout each period and were plotted on graph. However, since having all 7 emotions plotted on one graph for each participant made it difficult to read the data, it was ultimately decided that the 7 emotions would be divided into 3 graphs for all the time periods of 20 minutes, 40 minutes and 60 minutes (see Appendices 2.1 – 16.3). For these graphs, the colours chosen were appropriate to the emotions that were being plotted. The variation of the emotions detected in each graph are recorded between 0 and 100%, where 0% denotes that the emotion is not detected by the program and 100% denotes that the emotion is being strongly detected by the program.

With the first volunteer, during the 20-minute session, the volunteer experienced more fear at the beginning of the session (Appendix 2.1) mostly in the ranges of 60% - 100%. This can be observed in the intensity of the colour purple on the graph. After 10 minutes of the 20 minute session had passed, the anger emotion started to kick in a little more, with the range not exceeding more than 80% most of the time. The volunteer experienced a lot of sadness, which was denoted by the intensity of the colour dark blue on the graph, more than disgust, reaching 100% at times, whilst playing the 20 minute session (Appendix 2.2). At the same time, the volunteer experienced happiness, once again reaching 100% at times (Appendix 2.3). On top of this, the volunteer experienced a strong neutral emotion, though reaching 100% less of the time when compared to happiness and sadness, and a very weak surprise emotion. In fact, the neutral emotion drops significantly to as low as 5% in the 2 minute section between 13:03:58 and 13:05:58.

In the 40-minute session, the first volunteer experienced even more fear emotion at near 100% for around 35 minutes of the session, before experiencing anger for the last 5 minutes (Appendix 3.1). This was seen as a wide spread intensity of the purple colour. The anger emotion experienced in this session is a lot more than that of the 20-minute session. Like with the 20 minute session, the first volunteer experiences more sadness than disgust, where the levels of sadness reach 100% majority of the time (Appendix 3.2). This time however, the disgust emotion is experienced more towards the end of the 40 minute session, with levels ranging from 0% to roughly 72%. Simultaneously, both the happy and neutral emotions are experienced for most of the 40 minute session, mostly at 100% (Appendix 3.3). These emotions can be observed in the intensity of the colours yellow and grey respectively on the graph. At the very beginning of the 40 minute session, there are strong 100%

spikes in the surprise emotion. Towards the end of the 40 minute session, the neutral emotion drops to within the range of 0 – 75%.

Next in the 60 minute session, it can be observed that the first volunteer experiences slightly more anger than fear, which is denoted by the colour red on the graph, despite the roughly even levels of both anger and fear being detected (Appendix 4.1). Interestingly, despite the sad emotion being experienced by the volunteer at mostly 100% for most of the session, the disgust emotion has been detected a bit more strongly than in the 20 minute and 40 minute sessions (Appendix 4.2). The sad emotion dips below 60% towards the end of the session. The happy emotion was experienced a lot in the first half of the session at 100% most time, before decreasing the frequency in the second half and then eventually decreasing to 0% towards the end of the session (Appendix 4.3). At the same time, the volunteer experiences less of the neutral emotion than with the 20 minute and 40 minute sessions. The neutral emotion is only strongly experienced towards the end of the session at 100%. The surprise emotion is slightly more strongly detected in comparison to the 20 minute and 40 minute session.

Now with the second volunteer, the fear and anger emotions were experienced evenly for the full 20 minute session, hardly reaching 100% at times, but dropping off to roughly 45% towards the end of the session (Appendix 5.1). The volunteer experienced more disgust, despite the level of sadness being 100% most of the time, with some peaks reaching 40% or more (Appendix 5.2). Between 11:35:58 and 11:48:04, the sad emotion experienced by the volunteer looks a lot sparser when compared to that of the first volunteer. The neutral emotion is experienced very strongly during the 20 minute session, reaching levels of 100% majority of the time (Appendix 5.3). In addition to this, the surprise emotion had been experienced more than the happy emotion, with most of the peaks reaching 40% or more. The happy emotion is more strongly experienced in sections between 11:35:58 – 11:37:06 and 11:42:20 – 11:46:22.

The 40 minute session follows and, as with the 20 minute session, it is evident that the fear and anger emotions were experienced evenly (Appendix 6.1). However, these emotions were detected more strongly than in the 20 minute session, with more peaks falling in the 80 – 100% level range. The disgust emotion was experienced by the volunteer even more than in the 20 minute session, with most of the peaks reaching levels of 30% or more (Appendix 6.2). In roughly the first half of the session, the sad emotion was detected very strongly, with most of the peaks reaching 100%. However, in the second half, the frequency of the sad emotion peaks reaching 100% decreased, with hardly any sad emotions being experienced near the end of the session. The neutral emotion was strongly experienced by the volunteer for the entire session, reaching 100% most of the time (Appendix 6.3). This time the happy emotion was being detected less, with occasional peaks reaching 100%. Initially, the surprise emotion wasn’t being detected strongly, but after 7 minutes had passed, the emotion was being strongly detected. The peaks reached anywhere between 60% and 100% and lasted for the remainder of the session.

Afterwards, the 60 minute session was carried out and as before, the fear and anger emotions were experienced evenly (Appendix 7.1). This time, the anger emotion was more prominent than the fear emotion. The sad emotion was experienced more than in the 20 minute session, with peaks reaching 100% most of the time (Appendix 7.2). During the first half of the session, the disgust emotion was being strongly detected with peaks reaching 60% or more. Then, in the second half, the disgust emotion detected reduced to peaks less than or equal to 60%. Like in the 40 minute session, the neutral emotion was being strongly detected, with peaks reaching 100% majority of the time (Appendix 7.3). Once again, the happy emotion was being detected less, with occasional peaks reaching 100%. In the first almost three quarters of the session, the surprise emotion was being strongly detected, with peaks reaching 100% majority of the time. During the last quarter of the session, the surprise emotion became weakly experienced by the volunteer.

The third volunteer came after and in the 20 minute session, the fear emotion was more strongly detected than the anger emotion, with most of the peaks reaching between 80 – 97% (Appendix 8.1). In this session, the disgust emotion was not detected at all, and the sad emotion was only moderately detected, with peaks ranging from 30 – 90% (Appendix 8.2). Both the happy and neutral emotions were detected strongly, with peaks reaching 100% majority of the time (Appendix 8.3). Also, during this session, the surprise emotion was weakly detected.

During the 40 minute session, the fear emotion was detected more strongly than the anger emotion, with peaks reaching between 80 and 98% (Appendix 9.1). The anger emotion was detected moderately towards the end of the session and had peaks which never exceeded 60%. During this time, the sad emotion was again detected moderately, with only a few peaks ever reaching 100% (Appendix 9.2). The disgust emotion was only strongly detected near the 16:10:59 timestamp, with that peak reaching just over 30%. Just like with the 20 minute session, both the happy and neutral emotions were strongly detected, with peaks reaching 100% majority of the time (Appendix 9.3). This time, the surprise emotion was also more strongly detected, with more peaks reaching 100% when compared to the 20 minute session.

In the 60 minute session, the fear emotion was even more strongly detected than the anger emotion, with most of the peaks reaching between 90 – 100% (Appendix 10.1). In fact, the anger emotion was barely registered by the program in this session. The sad emotion was moderately detected in the first half of the session, with most peaks reaching 100% (Appendix 10.2). Towards the second half, the sad emotion started to reduce to levels where most peaks don't surpass 30%. The disgust emotion was not detected at all during this session. The neutral emotion was detected very strongly, where peaks reached 100% majority of the time (Appendix 10.3). In the beginning, the happy emotion was strongly detected, with most of the peaks reaching 100%. However, towards the end of the session, the happy emotion was being detected less, where fewer peaks reached 100%. A similar pattern can also be observed with the surprise emotion, but in reverse almost.

Now with the fourth volunteer's 20 minute session, the fear emotion was being strongly detected like before (Appendix 11.1). Interestingly, the fear emotion has been split almost 3 clean sections, with dips in-between them at the time sections of 19:35:08 - 19:36:15 and 19:44:18 – 19:45:27. The anger emotion was also being strongly detected more than before, but most of the peaks never exceed 80%. The sad emotion was moderately being detected, with only the strongest spikes reaching 100% (Appendix 11.2). The notable spikes are detected in the time sections of 19:35:41 – 19:36:15 and 19:43:44 – 19:44:52. The disgust emotion was weakly being detected, with only a few spikes slightly exceeding 20%, especially in the first half of the session. The neutral emotion was being strongly detected, with most of the peaks staying above 50% for the entire session (Appendix 11.3). On the other hand, the happy emotion was being moderately detected, with most peaks reaching 100%. This is quite prevalent at the beginning of the session but reduces towards the end of the session. The surprise emotion was being detected moderately throughout the session.

In the 40 minute session, the fear emotion was being strongly detected and the anger emotion was being moderately detected (Appendix 12.1). The fear peaks reached between 80-100% most of the time and the anger peaks don't reach 80%. The sad emotion was being detected moderately, with some of the peaks exceeding 80% (Appendix 12.2). The disgust emotion was being detected very weakly, with most of the peaks not exceeding 20%. At the same time, the neutral emotion was being detected strongly, with majority of the peaks being above 90% (Appendix 12.3). The happy emotion was being detected moderately, with most peaks reaching 100%. On top of this, the surprise emotion was being detected, not as strongly as the happy emotion, but more frequently than the happy emotion. Most of the peaks of this emotion reached above 60%.

In the 60 minute session, the fear emotion was quite strongly detected, with most peaks reaching 100% (Appendix 13.1). Simultaneously, the anger emotion was moderately being detected, with most peaks exceeding 50%. Moreover, the sad emotion was moderately being detected, with some peaks reaching 100% (Appendix 13.2). At the same time, the disgust emotion was being weakly detected, with most peaks not surpassing 40%. The neutral emotion was strongly detected in the session, with majority of the peaks reaching 100% (Appendix 13.3). Whilst this was happening, the happy emotion was being detected strongly, but very infrequently, with stronger peaks reaching 100%. In addition to this, the surprise emotion was being strongly detected, but with more frequent peaks. Most of the peaks reached levels of between 70% - 100%.

Finally, with the fifth volunteer's 20 minute session, both the fear and anger emotions are much more weakly detected when compared to the other volunteers' 20 minute session (Appendix 14.1). Only a few of the peaks detected ever reach 100% in this session. The sad emotion was moderately detected, but once again not as strong when compared to the other 20 minute sessions that were conducted (Appendix 14.2). Even more so, the disgust emotion was not detected at all, except for a small spike reaching 42% at 15:23:52. The neutral emotion was detected at 100% for most of the time (Appendix 14.3). Also, during the session, the happy emotion was picked up upon a lot more frequently, with most of the peaks reaching 100% at times. What is more is that the surprise emotion detected was less than what the second and fourth volunteers had produced in their 20 minutes.

In the 40 minute session, both the fear and anger emotions were detected quite evenly (Appendix 15.1). Some of the anger spikes reached 100% at the beginning of the session and some of the fear spikes reached 100% towards the end of the session. At the same time, the sad emotion was being moderately detected, with many of the spikes not exceeding 60% (Appendix 15.2). The disgust emotion was only strongly detected towards the end of the session. As with the 20 minute session, the neutral emotion was detected very strongly, with most of the peaks reaching 100% (Appendix 15.3). Also, the happy emotion was strongly detected, with most of the peaks reaching 100% again. However, there is a massive gap where the happy emotion isn't detected as strongly as the rest of the session was. This section lies between the timestamps of 15:39:21 – 15:54:23. At the same time, the surprise emotion was being moderately detected, with only a few peaks surpassing 80% near the end of the session.

Onto the 60 minute session and once again, both the fear and anger emotions were detected quite evenly (Appendix 16.1). Majority of the anger and fear spikes reached 100%. In addition to this, the sad emotion was quite strongly detected, with most of the spikes reaching 100% at times (Appendix 16.2). There are few detections of the disgust emotion, as signified by the spikes at 16:31:01, 17:04:14, 17:07:39, 17:12:42, 17:22:36 and 17:25:56. Half of these spikes reached levels above 80%, whilst the other half reached levels of barely 60%. The neutral emotion was detected very strongly during the session, with majority of the spikes reaching 100% (Appendix 16.3). Furthermore, the happy emotion was strongly detected at 100% at times. However once again, there were sections where the emotion wasn't being strongly detected compared to the rest of the session. These sections happened between 16:34:28 – 16:46:57, 16:55:36 – 16:59:00, 17:12:42 – 17:19:13 and 17:25:56 – 17:29:18. Lastly, the surprise emotion was quite strongly detected as well, the few of the peaks reaching near 100%, especially the key ones located at the timestamps of 17:02:30 and 17:12:42.

The post-review forms (Appendix 17) also provided some interesting information and comments about what the volunteers felt after playing the chosen game. During the 20 minute session, all the volunteers mentioned that they felt really excited about the game and that they were able to stay focused on the game for that time period. In the 40 minute session, all of the volunteers mentioned that they still felt the excitement for the game and were able to concentrate on the gameplay. However, one of the volunteers mentioned that they felt angry and annoyed at times whenever the game didn't go their way and mentioned that wanted to improve their skills. When it came to the 60 minute session, all of the volunteers had mentioned that they started to feel tired and thus lost an interest as they felt the game was getting a bit repetitive. Overall, the volunteers found the gameplay to be very interesting and exciting for the most part. However, some of the volunteers commented on the longer sessions being either frustrating or quite boring.

Analysis & Discussion

In real life, the consumers decide to play the game of their choosing, but they don't truly get to see the effects of the gameplay have on their emotions. Because of this, these consumers don't consider reporting their emotions when sending reviews to the companies which create and design the games. This leads to an increase in difficulty for companies who create and design these games in developing more better and exciting games to keep up with the demand.

Because of this, it was decided that the aim was to find out how the length of the gameplay affects the player's emotions. Based on the experiment that was carried out, the results indicate that as the length of the gameplay increased from 20 minutes to 60 minutes, the emotions of the volunteers mostly became increasingly more neutral and less happy. Furthermore, the levels of sad and disgust emotions significantly increased as the length of the gameplay increased. In addition to this, the levels of fear and anger emotions increased.

In the 20 minute session, 3 volunteers showed higher levels of fear for the entirety of the session than the other 2 volunteers did. This suggested that these volunteers may have been more fearful because of the fast-paced environment that the game produced and the fact that they were worried that they would be shot from any direction by the opponents in the game. As a result, this would have heightened their senses since they would be focusing on keeping an eye on their surroundings during the gameplay. This is like a study done by Shu-Hua Yeh [13] as she found action video gameplay elicited higher stress than non-action video gameplay. Out of all the volunteers, the second volunteer experienced the most amount of anger, disgust, neutral and surprise emotions. This could imply that the volunteer took the game more seriously than the other volunteers and may have had a difficult time trying to stay alive in the game for the most part of the session. This is like the study done by Hollingdale and Greitemeyer [5] as they found that those who played the violent game online were angrier and more aggressive compared to those who played the same game offline. On the other hand, the analysis only partially correlates with the comments written by volunteer as he mentioned that he felt rather happy and excited about the game. This indicated that what the volunteer felt did not always equate to the emotions detected by the program.

The fourth volunteer expressed similar levels of the neutral and surprise emotions like the second volunteer did, which suggests that this volunteer was focused on the game as well. However, the amount of the anger and disgust emotion that was experienced by this volunteer was a lot less than what the second volunteer experienced. This could suggest that the fourth volunteer didn't particularly care as much about dying a lot during the gameplay as the second volunteer did. This is like a study by Bonus, Peebles and Riddle [14] as they found that initially frustrated participants purged their negative emotions after playing a video game and rated the gameplay as more enjoyable, regardless of the content. The fourth volunteer could have already been used to the emotional feel of how the FPS games are, which could also be a reason as to why the anger and disgust emotion levels are low. In addition to this, the similarly lower levels of the disgust and anger emotions, coupled with the higher levels of the happy emotion, obtained from the third and fifth volunteers prove that they rather enjoyed the gameplay a lot better than the other volunteers did. This point is further backed up the fact that the level of the sadness from these volunteers were the lowest of all the volunteers. These results support the findings obtained in a study by Hollingdale and Greitemeyer [5] as they also found that the participants enjoyed the violent game, despite being the game being more challenging.

With the 40 minute session, all the volunteers showed more heightened emotions than they did during the 20 minute session. This could be due to the concentration of volunteers during the gameplay being more intense since they were facing harder challenges in the game. Like before with the 20 minute session, the second volunteer experienced more of the anger, disgust, neutral and surprise emotions. This correlated with the information received from the form as the volunteer mentioned that he felt angrier and more annoyed because of being killed or missing the target in the game. This agrees with how the anger and frustration emotions were trigged in the study done by Hallema [6], as that report found that the cause of this trigger was down to poor coordination and violent content. The results contributed to the findings as it showed how the anger levels changed between 3 different time sessions. The third, fourth and fifth volunteers still showed low levels of the sad and disgust emotions, which suggests that they still enjoyed playing the game during this session. This is proved by the forms (Appendix 17) as they also mentioned that they still felt good about the game during the gameplay.

At the same time, the fourth volunteer showed less of the surprise emotion than the second volunteer did. This suggests that the volunteer felt less enthusiastic about the game as he was used to the kind of difficulty that the FPS games gave. In addition to this, levels of disgust and sadness were still low, despite being higher than levels of the sadness emotion produced in the 20 minute session. This correlates with the fourth volunteers comments in the form as he mentions that he lost concentration a bit during the gameplay, but was still happy. This is like a study done by Mihan, Anisimowicz and Nicki [15] as they found that multiplayer gameplay conditions significantly lessen the anger felt by the participants. The first volunteer’s 40 minute session produced an anomaly in the data towards the end of the session, where the anger emotion levels were higher than that of the fear emotion levels. This anomaly could possibly be due to the program not detecting a proper emotion correctly because of a lag in the data collection from the frames recorded.

During the 60 minute session, the data retrieved from the volunteers showed that the emotions were heightened even more. This could be due to the result of the players being more emotionally ready for the game since they have already played the previous two sessions. The second volunteer once again gives off high levels of fear, anger, surprise and neutral emotions as with the previous sessions. This is due to the volunteer being still interested in playing the game and thus still concentrating on the game at hand. Furthermore, the anger emotion levels being high are the result of violent content produced by the game. A study by Hollingdale and Greitemeyer [5] agrees with the results as it also showed the extent of the anger emotions exhibited after playing a chosen game through the amount of chili sauce dispensed by the participants. The results contribute to the findings as the extent of the emotions recorded are done over several time periods as opposed to one.

Compared to the 20 minute and 40 minute sessions, the first, third and fifth volunteers showed less of the happy emotion. The volunteers could have experienced this because of quickly adapting to the atmosphere the game produced and thus taking the game a bit more seriously than before. This is similar to how the second volunteer emotionally reacted to the game during his 20 minute and 40 minute session, except that he was motivated to do well from the beginning of those sessions. These results don't correlate very well with the findings made in the study done by Schrader and Nett [7] as they found that the participants who played the moderate or full control version of their game had experienced more happier emotions compared those who had the reduced control version of the game. This implies that not all games give the same euphoric feelings once a person gets used to the controls of the game. In addition to this, the comments made in the forms (Appendix 17) support this analysis as these volunteers also mention finding the game being a bit repetitive after a while and thus lost a little interest in the game.

The limitations of this experiment were the number of volunteers that took part. As this study required participants to spend over 2 hours of their time to complete the experiment, some of the volunteers ended up pulling out of the experiment as they were unable to make that commitment. In addition to this, some of the volunteers that were chosen decided that this game was not for them and withdrew.

Conclusions & Future Work

Experiments that were carried out on previous work confirm that there was a relationship between the gameplay and the emotions experienced from the gameplay. This project aimed to find out how the length of the gameplay affects the player's emotions. As seen in the results obtained, there was a clear increase in the intensity of the emotions experienced from the 20 minute session to the 60 minute session. Based on these outcomes and the analysis that was carried out after, it can be concluded that the longer sessions of gameplay did have a significant impact on the player's emotions when compared to shorter gameplay sessions. While an effect on all the emotions were seen with longer gameplay, a significant increase was seen in the negative types of emotions, such as anger and sadness.

Using these findings, the game developers could consider designing and developing more games that are exciting to play but can be completed in a shorter time frame to reduce the possible emotional impact caused by playing computer and video games in the future. Furthermore, more research could be done in finding the optimal duration of time that the gamers could engage with, so that they can enjoy the game, but the effects on their emotions could be kept to a minimum. Incentivising participants through the involvement of the gaming industry could be looked into for recruiting larger sample sizes for future studies.

Reflection

Looking back on this project, I picked up some skills that were useful and faced challenges. For example, my programming skills have improved over time and are better than when I first started this project. Furthermore, I have come to learn that games have a unique way of generating feelings and emotions in people based on the content that was created. Originally, I had wanted to turn my program into a working application that could have been used by other people. However, I had faced issues in converting my program into an application because of errors that I could not fix. If I could do this project again, I would test the experiment with other games that people haven't tried before and compare those results with the ones I had obtained in this project. In this instance, I would try to fix the application conversion issue before trying the experiment again.

References

[1] Wikipedia. (2022, April 14). *Video game*. Wikipedia. Retrieved 18 April 2022, from https://en.wikipedia.org/wiki/Video_game

[2] Statista, & Clement, J. (2021, January 29). *COVID-19: global video game sales increase as of March 2020*. Statista. Retrieved 18 April 2022, from https://www.statista.com/statistics/1109977/video-game-sales-covid/#professional

[3] Quwaider, M., Alabed, A., & Duwairi, R. (2019). The impact of video games on the players behaviors: A survey. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2019.04.077

[4] Wirtz, B. (2022, March 31). *34 Popular Types of Video Games, Explained (With Examples)*. Video Game Design and Development. Retrieved 18 April 2022, from https://www.gamedesigning.org/gaming/video-game-genres/#:%7E:text=How%20many%20different%20types%20of,are%20over%2030%20different%20types.

[5] Hollingdale, J., & Greitemeyer, T. (2014). The effect of online violent video games on levels of aggression. *PLoS ONE*. https://doi.org/10.1371/journal.pone.0111790

[6] Hallema, G. (2013, January 31). *Exploring the relations between video games, emotional responses and eye-tracking*. ResearchGate. Retrieved 19 April 2022, from https://www.researchgate.net/publication/275332994_Exploring_the_relations_between_video_games_emotional_responses_and_eye-tracking

[7] Schrader, C., & Nett, U. (2018). The perception of control as a predictor of emotional trends during gameplay. *Learning and Instruction*. https://doi.org/10.1016/j.learninstruc.2017.08.002

[8] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2001.990517

[9] Khan, T. (2022, March 30). *Computer Vision — Detecting objects using Haar Cascade Classifier*. Medium. Retrieved 10 April 2022, from https://towardsdatascience.com/computer-vision-detecting-objects-using-haar-cascade-classifier-4585472829a9

[10] Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2014.220

[11] Dey, V. (2021, October 13). *Face Recognition System using DEEPFACE (With Python Codes)*. Analytics India Magazine. Retrieved 12 April 2022, from https://analyticsindiamag.com/face-recognition-system-using-deepfacewith-python-codes/

[12] Fritz, M., Schiele, B., & Piater, J. H. (2009, September). Computer Vision Systems. In D. S. Bolme, J. R. Beveridge, & B. A. Draper (Eds.), *FaceL: Facile Face Labeling* (pp. 28–29). Springer Publishing. https://doi.org/10.1007/978-3-642-04667-4

[13] Yeh, C. S. H. (2015). Exploring the effects of videogame play on creativity performance and emotional responses. *Computers in Human Behavior*. https://doi.org/10.1016/j.chb.2015.07.024

[14] Bonus, J. A., Peebles, A., & Riddle, K. (2015). The influence of violent video game enjoyment on hostile attributions. *Computers in Human Behavior*. https://doi.org/10.1016/j.chb.2015.05.044

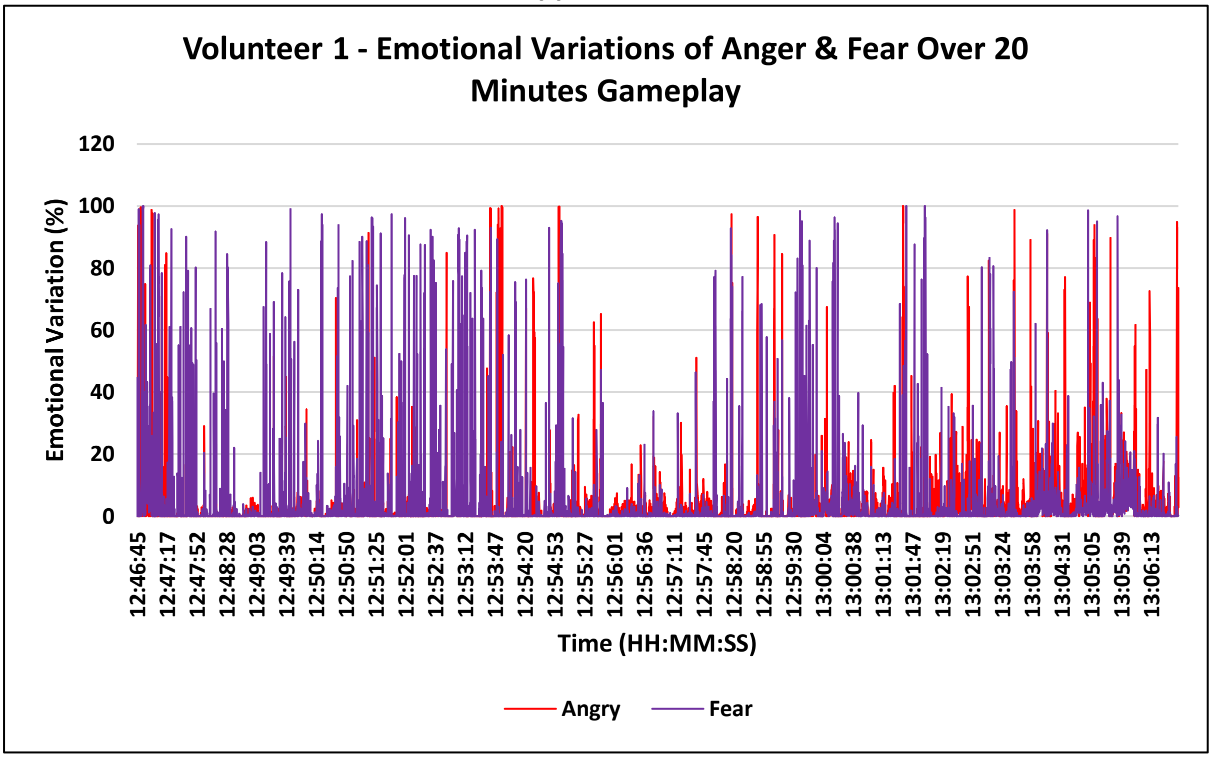
[15] Mihan, R., Anisimowicz, Y., & Nicki, R. (2015). Safer with a partner: Exploring the emotional consequences of multiplayer video gaming. *Computers in Human Behavior*. https://doi.org/10.1016/j.chb.2014.11.053

Appendices

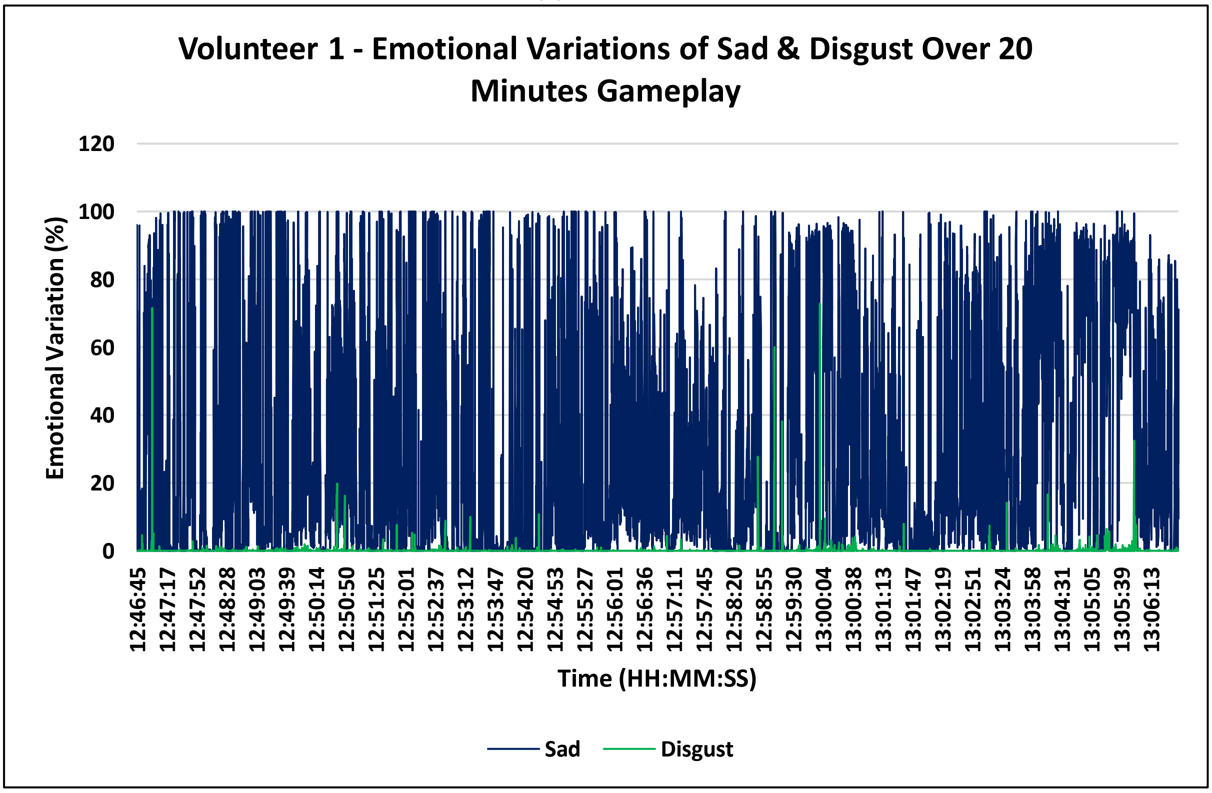
Appendix 1 – Link to the Project Git Repository for the Code

https://csgitlab.reading.ac.uk/at015244/individual-project

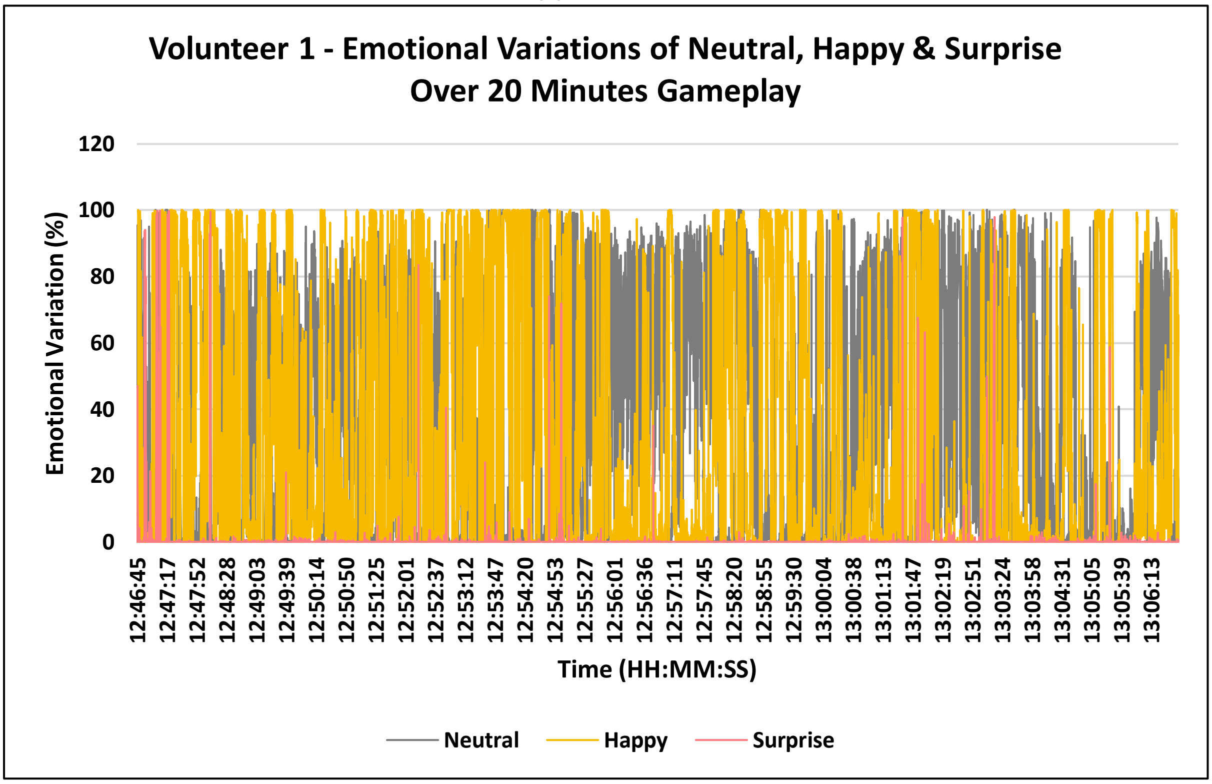
Appendix 2.1



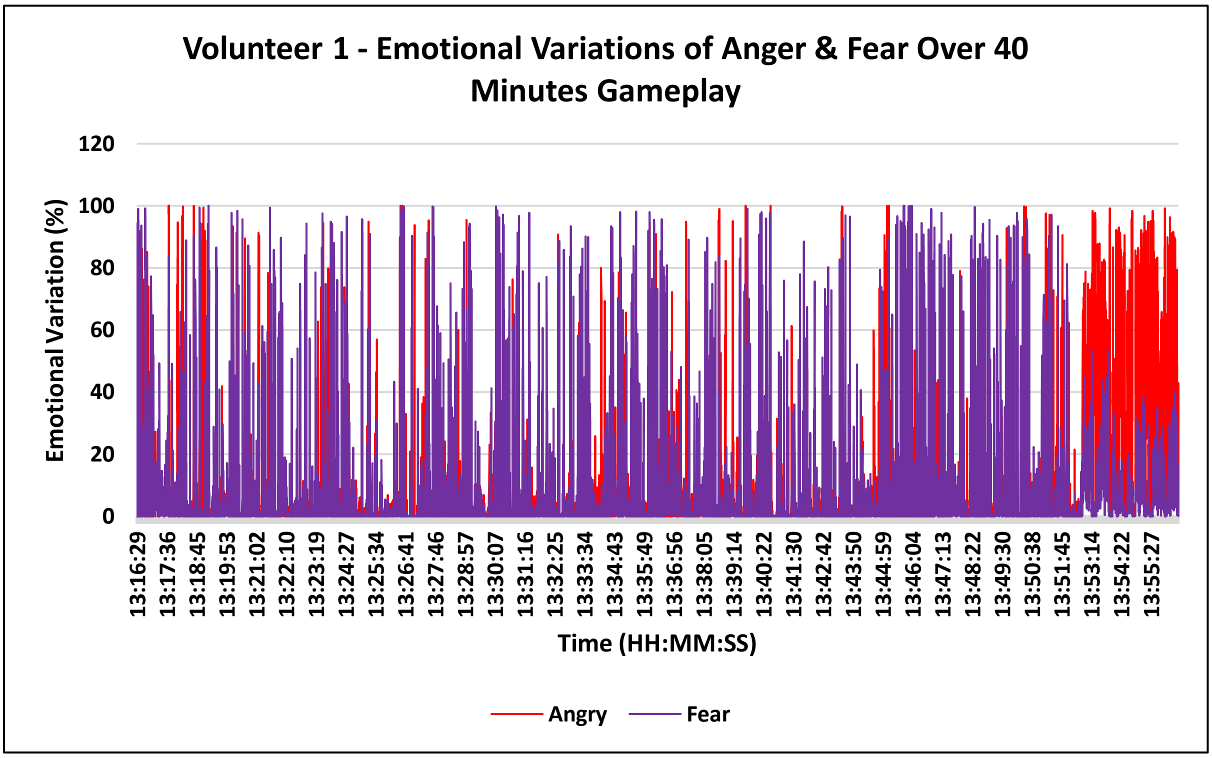
Appendix 2.2



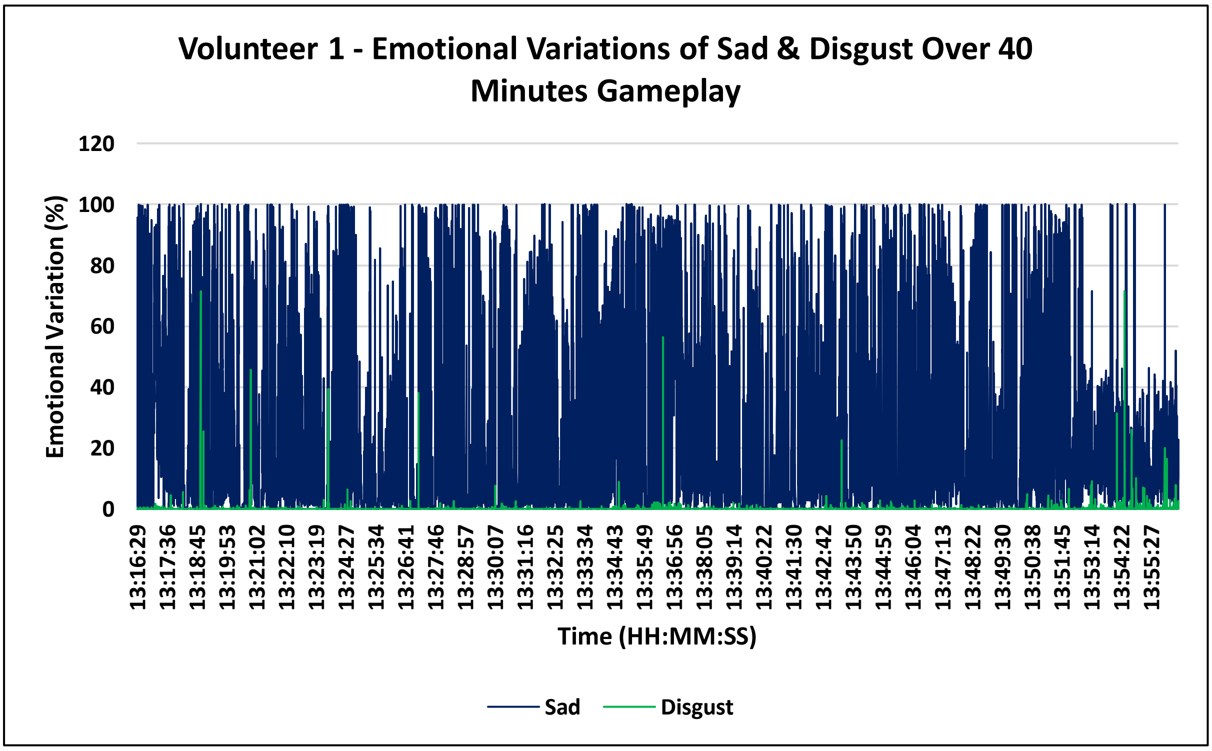
Appendix 2.3


Volunteer 1 - Emotional Variations of Neutral, Happy & Surprise Over 20 Minutes Gameplay
Emotional Variation (%)
120
100
80
60
40
20
0
12:46:45
12:47:17
12:47:52
12:48:28
12:49:03
12:49:39
12:50:14
12:50:50
12:51:25
12:52:01
12:52:37
12:53:12
12:53:47
12:54:20
12:54:53
12:55:27
12:56:01
12:56:36
12:57:11
12:57:45
12:58:20
12:58:55
12:59:30
13:00:04
13:00:38
13:01:13
13:01:47
13:02:19
13:02:51
13:03:24
13:03:58
13:04:31
13:05:05
13:05:39
13:06:13
Time (HH:MM:SS)
Neutral
Happy
Surprise

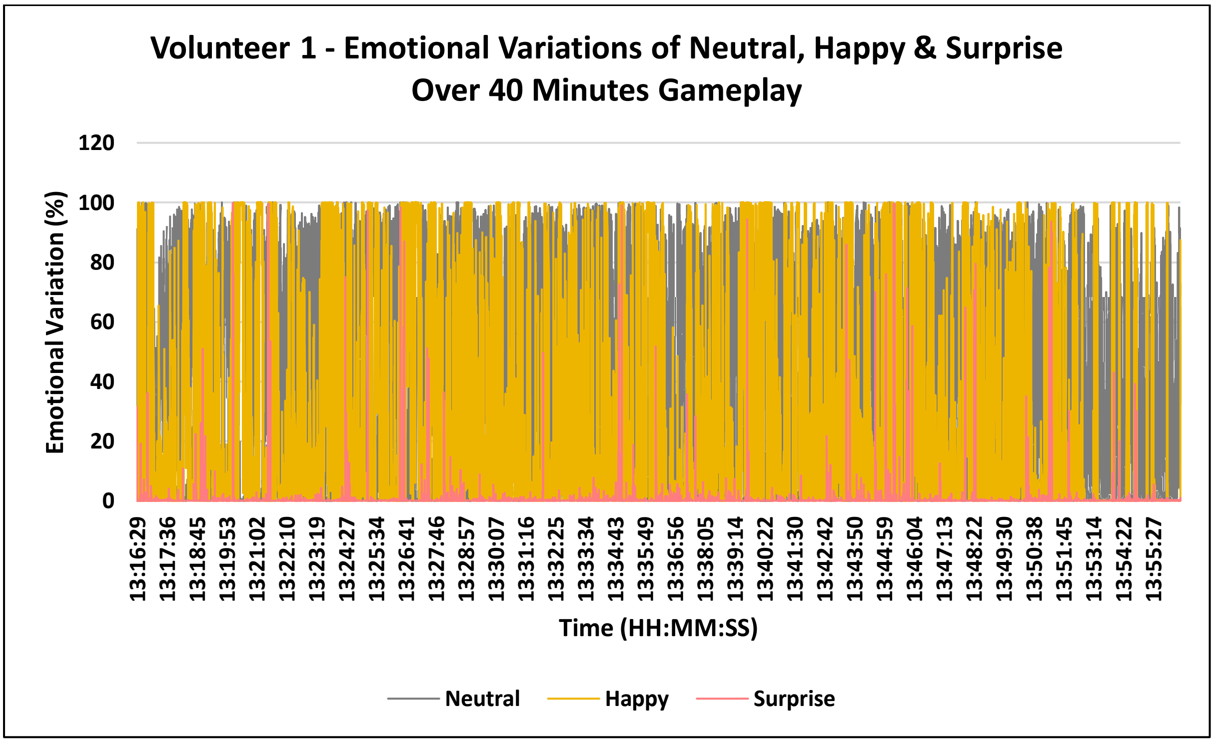
Appendix 3.1



Appendix 3.2

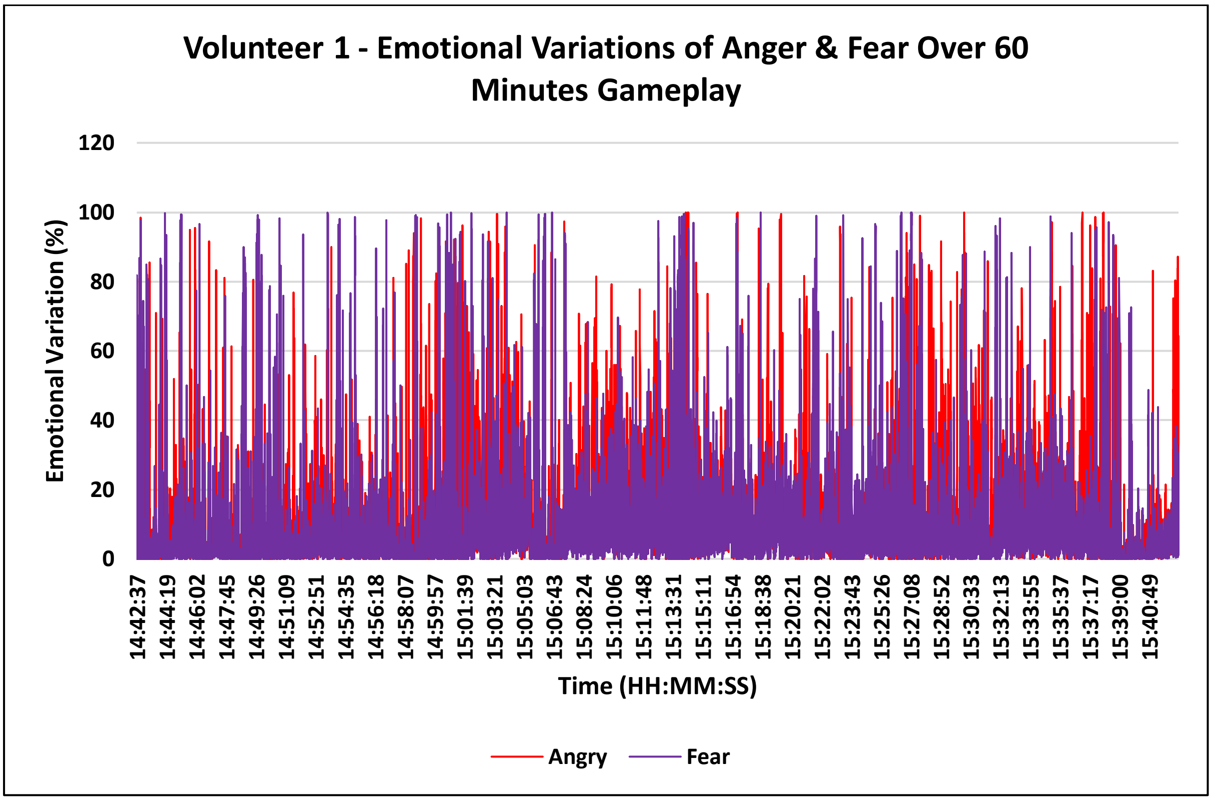


Appendix 3.3

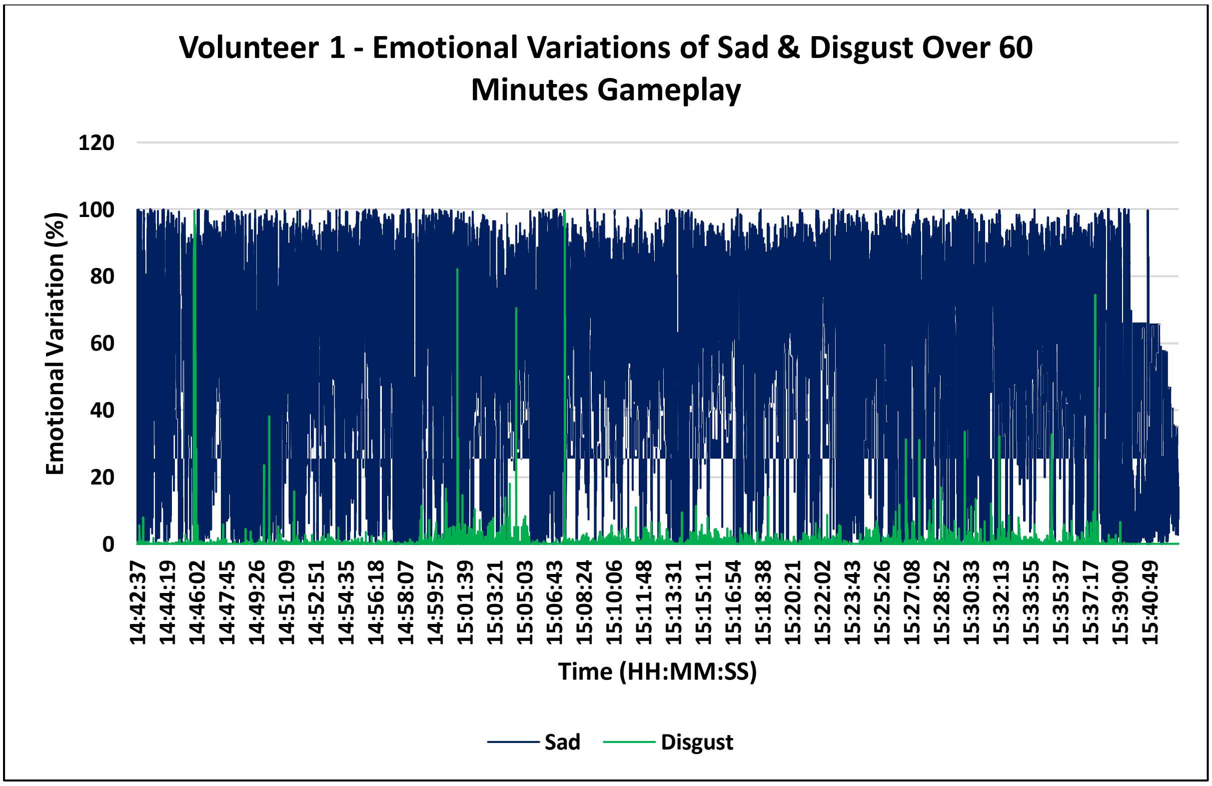


Volunteer 1 - Emotional Variations of Neutral, Happy & Surprise Over 40 Minutes Gameplay
Emotional Variation (%)
120
100
80
60
40
20
0
13:16:29
13:17:36
13:18:45
13:19:53
13:21:02
13:22:10
13:23:19
13:24:27
13:25:34
13:26:41
13:27:46
13:28:57
13:30:07
13:31:16
13:32:25
13:33:34
13:34:43
13:35:49
13:36:56
13:38:05
13:39:14
13:40:22
13:41:30
13:42:42
13:43:50
13:44:59
13:46:04
13:47:13
13:48:22
13:49:30
13:50:38
13:51:45
13:53:14
13:54:22
13:55:27
Time (HH:MM:SS)
Neutral
Happy
Surprise

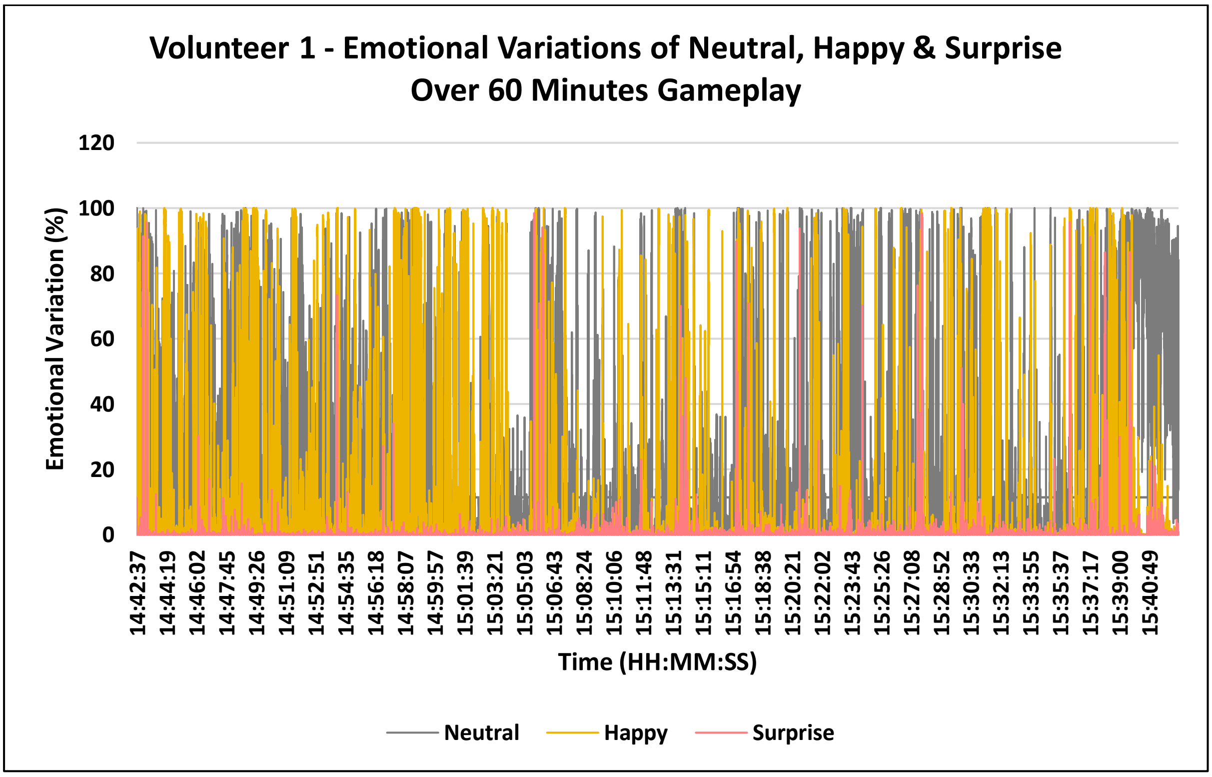
Appendix 4.1



Appendix 4.2

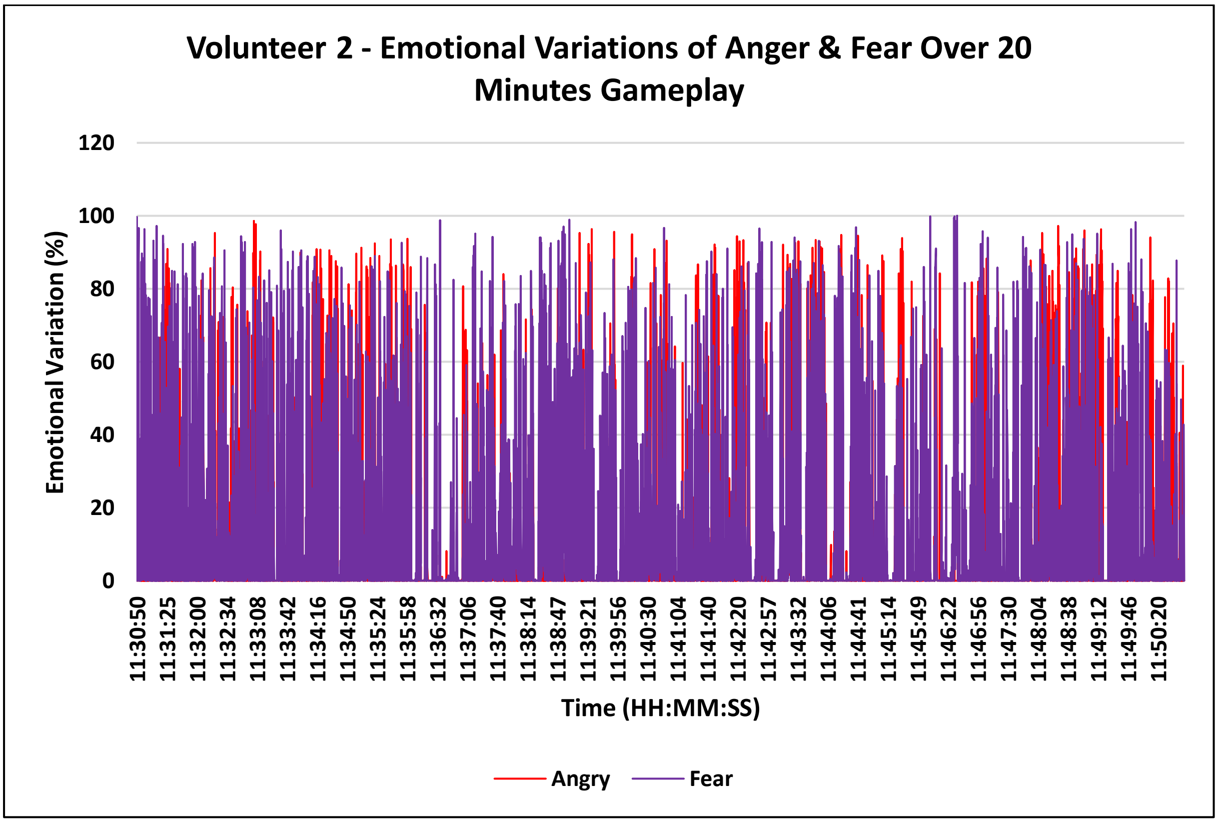


Appendix 4.3

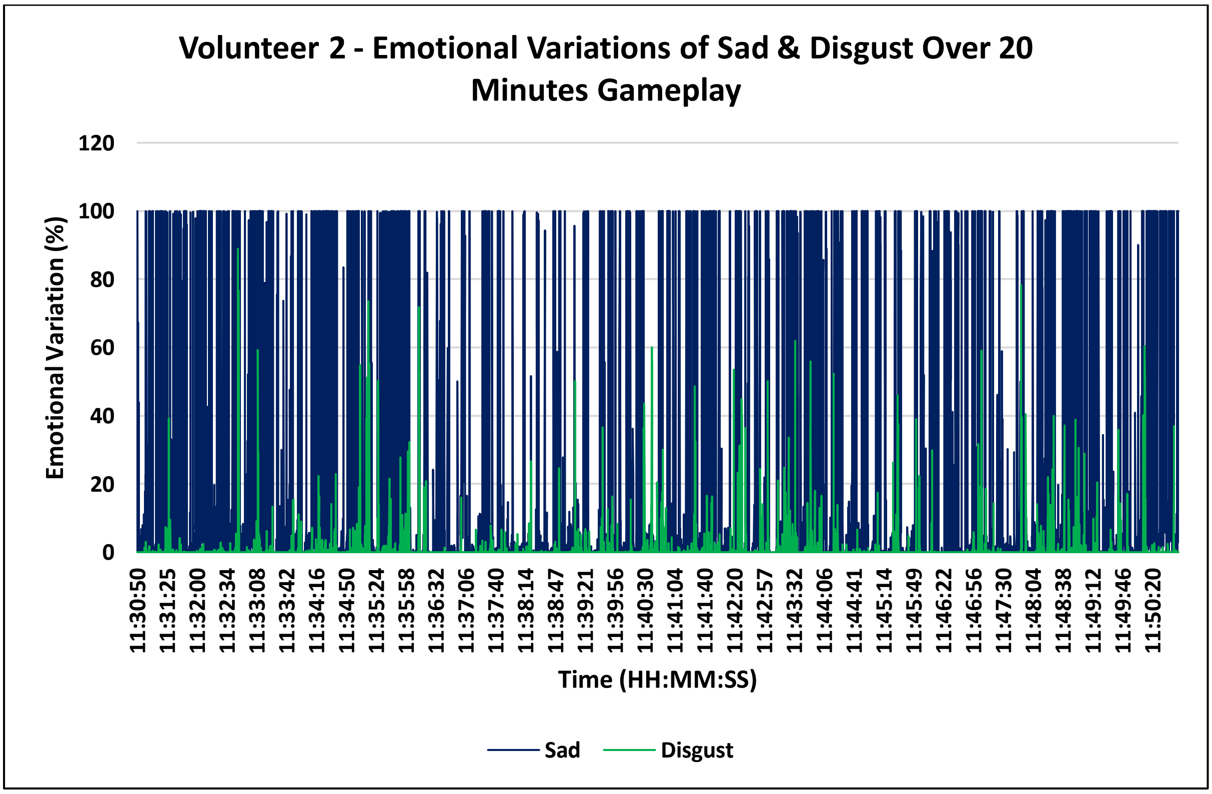


Volunteer 1 - Emotional Variations of Neutral, Happy & Surprise Over 60 Minutes Gameplay
Emotional Variation (%)
120
100
80
60
40
20
0
14:42:37
14:44:19
14:46:02
14:47:45
14:49:26
14:51:09
14:52:51
14:54:35
14:56:18
14:58:07
14:59:57
15:01:39
15:03:21
15:05:03
15:06:43
15:08:24
15:10:06
15:11:48
15:13:31
15:15:11
15:16:54
15:18:38
15:20:21
15:22:02
15:23:43
15:25:26
15:27:08
15:28:52
15:30:33
15:32:13
15:33:55
15:35:37
15:37:17
15:39:00
15:40:49
Time (HH:MM:SS)
Neutral
Happy
Surprise

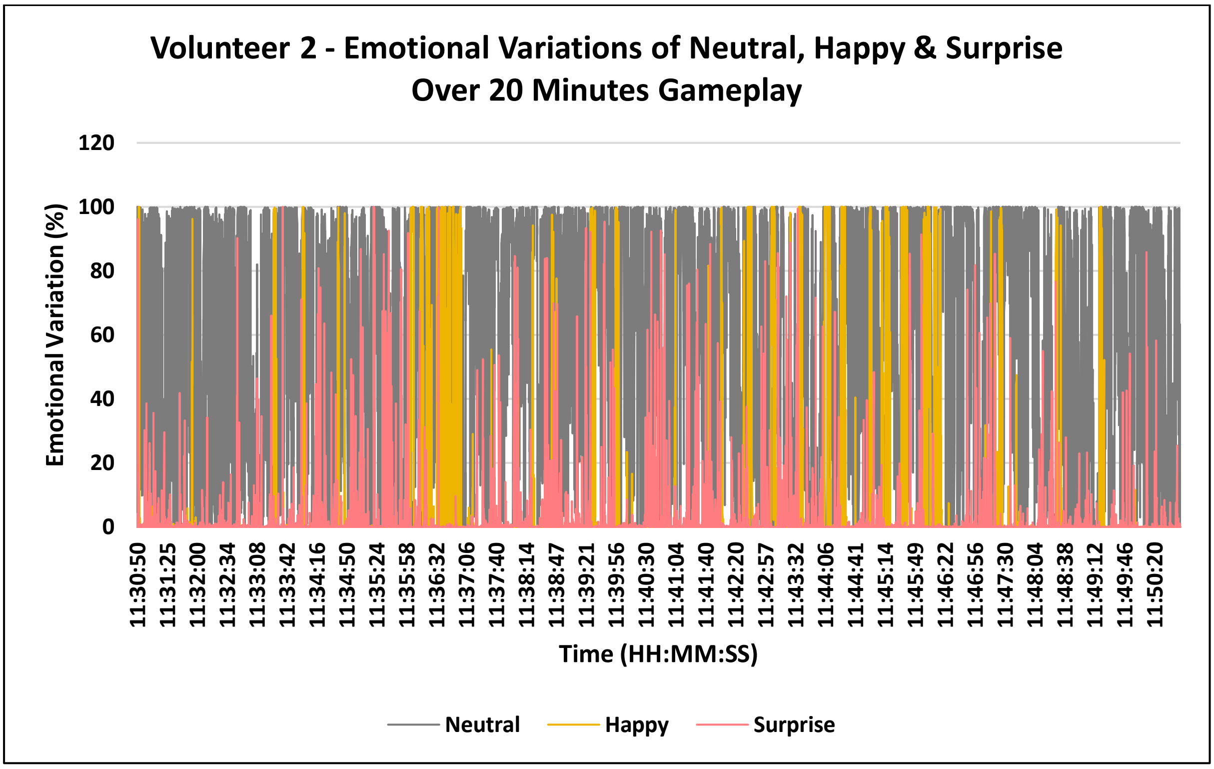
Appendix 5.1



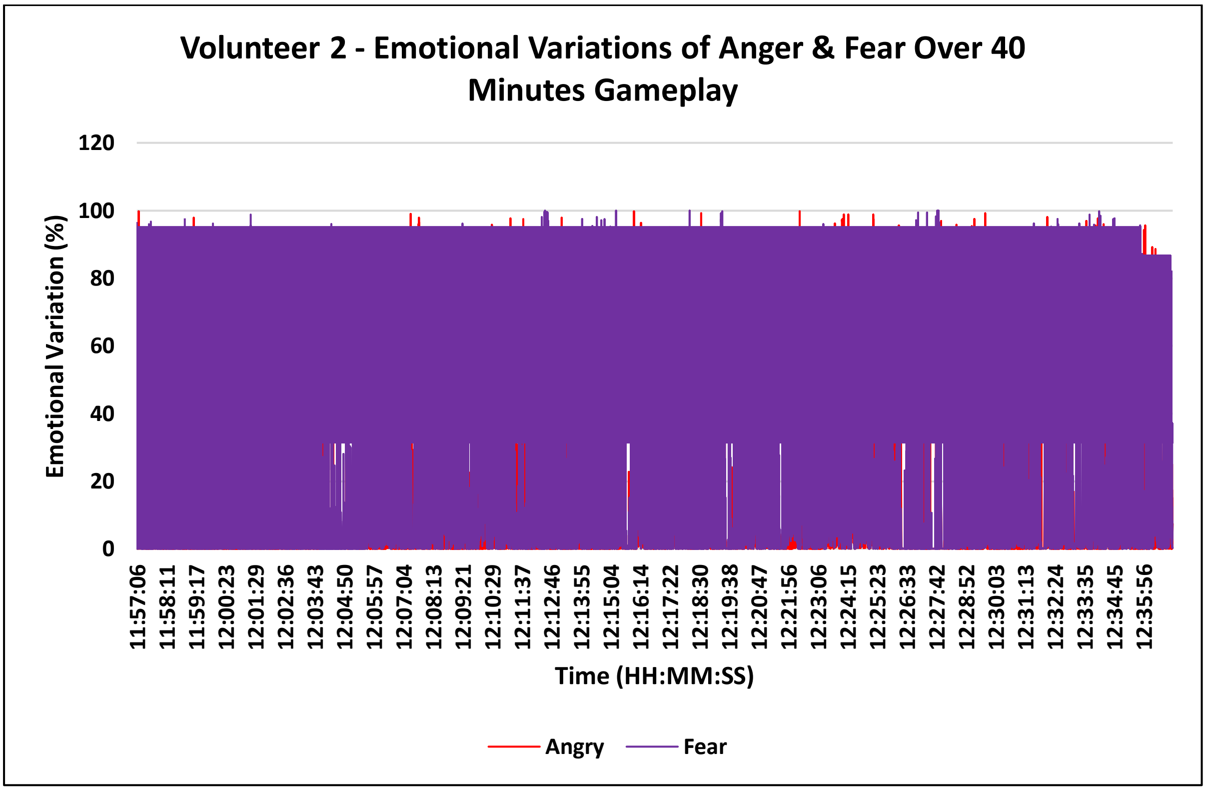
Appendix 5.2



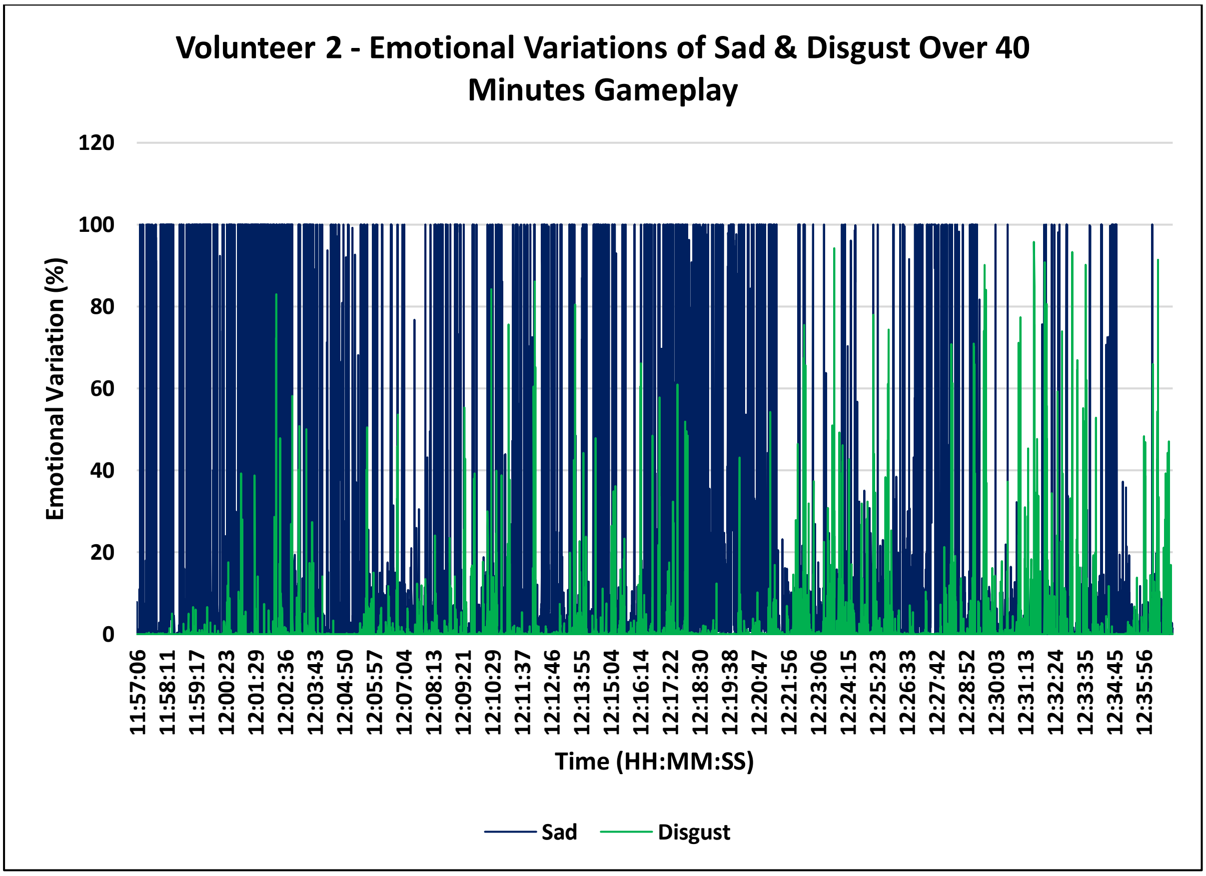
Appendix 5.3


Volunteer 2 - Emotional Variations of Neutral, Happy & Surprise Over 20 Minutes Gameplay
Emotional Variation (%)
120
100
80
60
40
20
0
11:30:50
11:31:25
11:32:00
11:32:34
11:33:08
11:33:42
11:34:16
11:34:50
11:35:24
11:35:58
11:36:32
11:37:06
11:37:40
11:38:14
11:38:47
11:39:21
11:39:56
11:40:30
11:41:04
11:41:40
11:42:20
11:42:57
11:43:32
11:44:06
11:44:41
11:45:14
11:45:49
11:46:22
11:46:56
11:47:30
11:48:04
11:48:38
11:49:12
11:49:46
11:50:20
Time (HH:MM:SS)
Neutral
Happy
Surprise

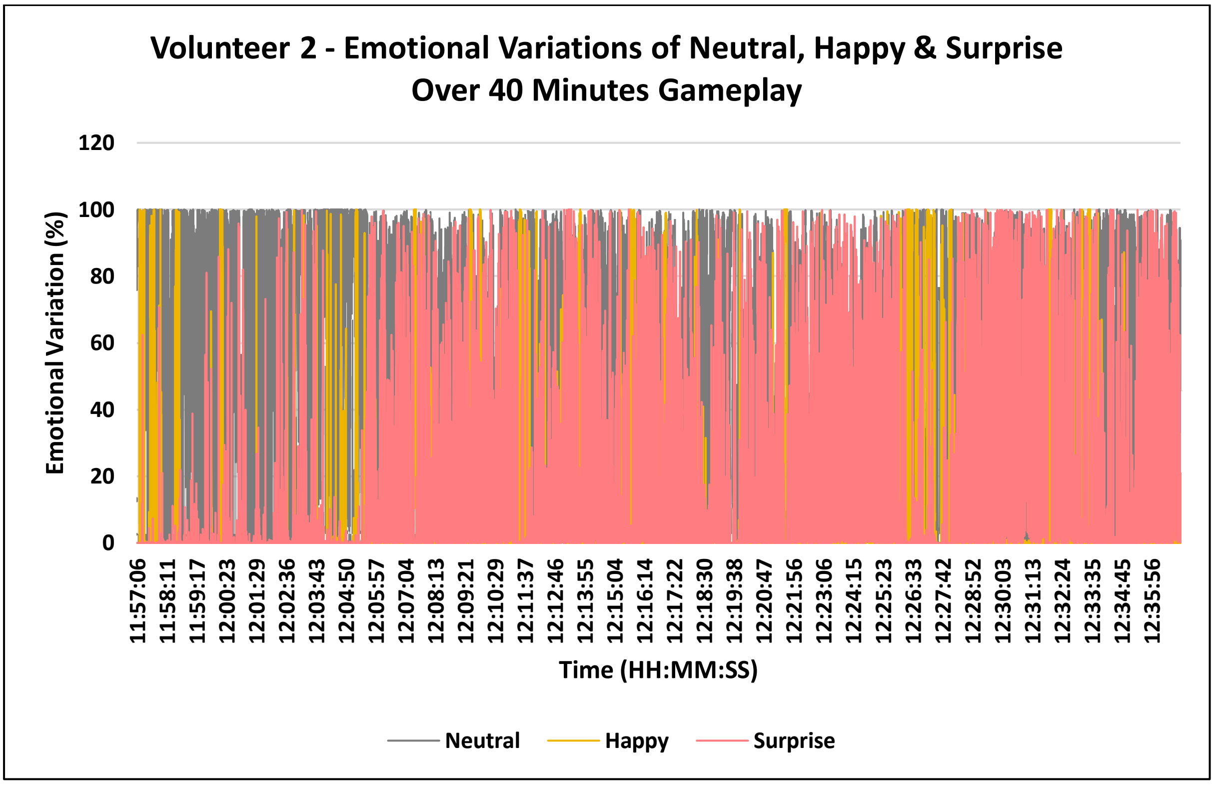
Appendix 6.1



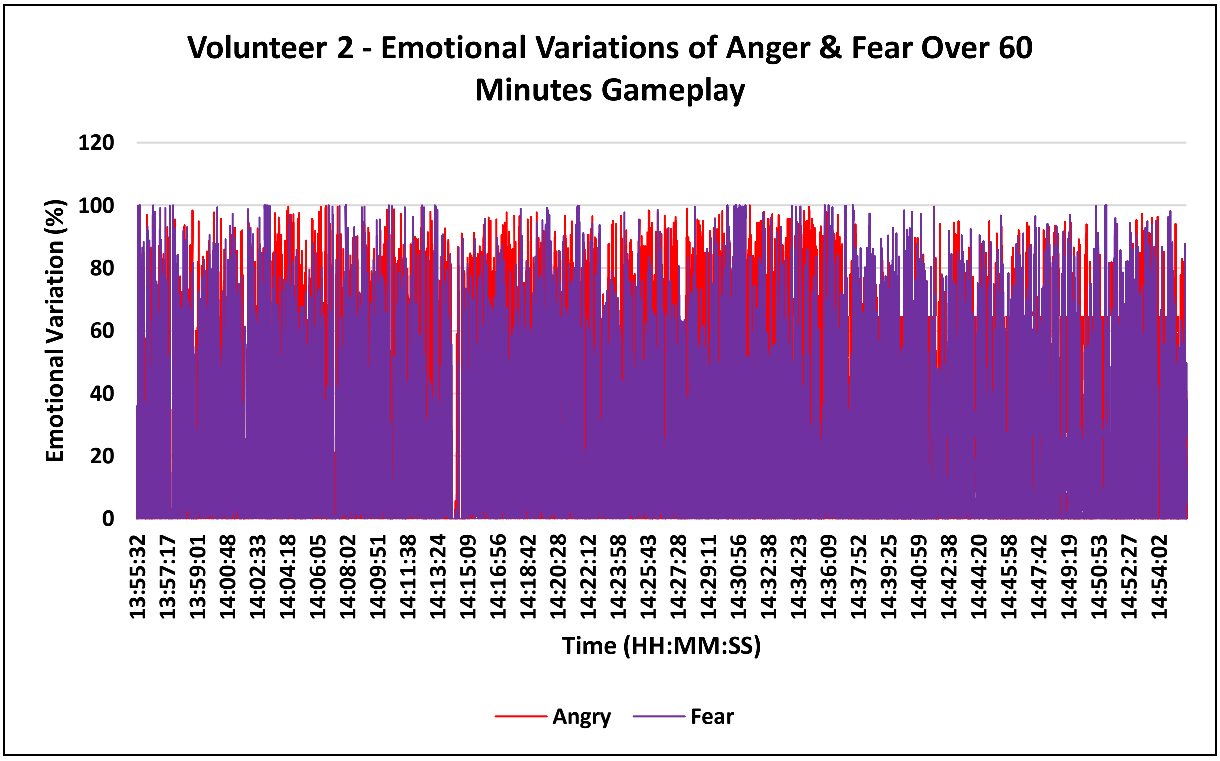
Appendix 6.2



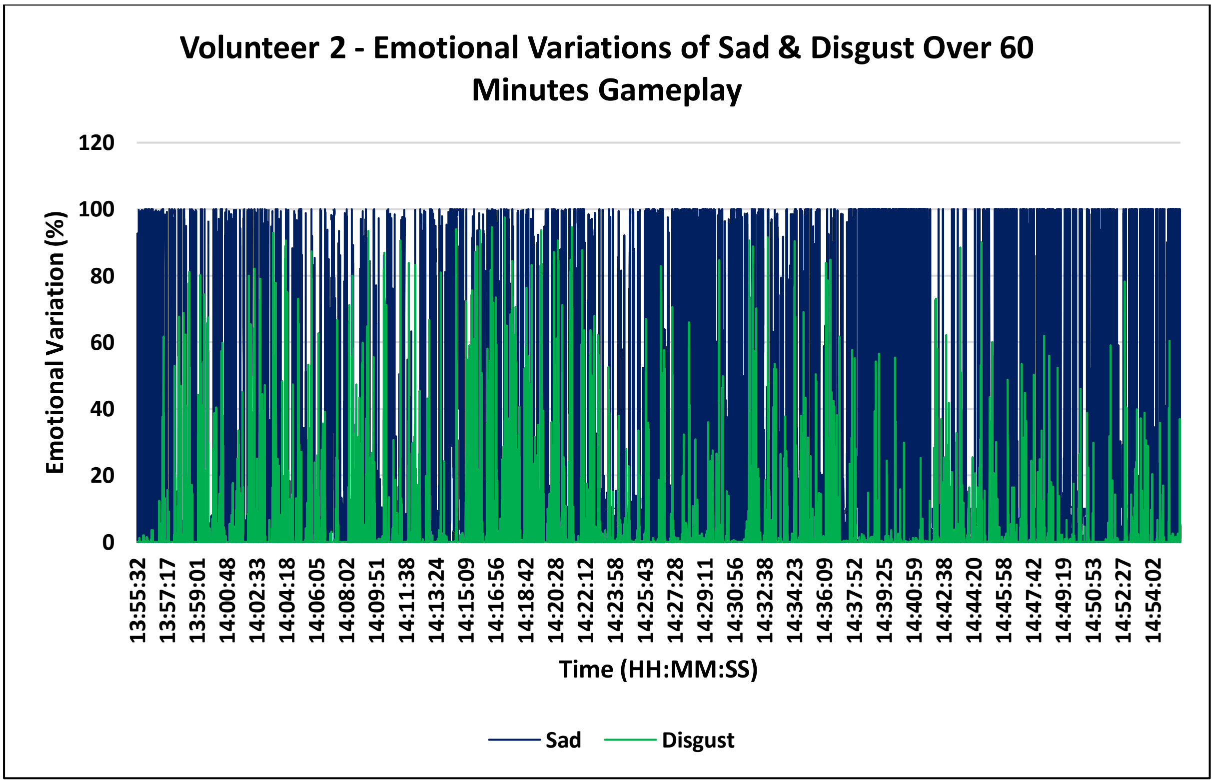
Appendix 6.3


Volunteer 2 - Emotional Variations of Neutral, Happy & Surprise Over 40 Minutes Gameplay
Emotional Variation (%)
120
100
80
60
40
20
0
11:57:06
11:58:11
11:59:17
12:00:23
12:01:29
12:02:36
12:03:43
12:04:50
12:05:57
12:07:04
12:08:13
12:09:21
12:10:29
12:11:37
12:12:46
12:13:55
12:15:04
12:16:14
12:17:22
12:18:30
12:19:38
12:20:47
12:21:56
12:23:06
12:24:15
12:25:23
12:26:33
12:27:42
12:28:52
12:30:03
12:31:13
12:32:24
12:33:35
12:34:45
12:35:56
Time (HH:MM:SS)
Neutral
Happy
Surprise

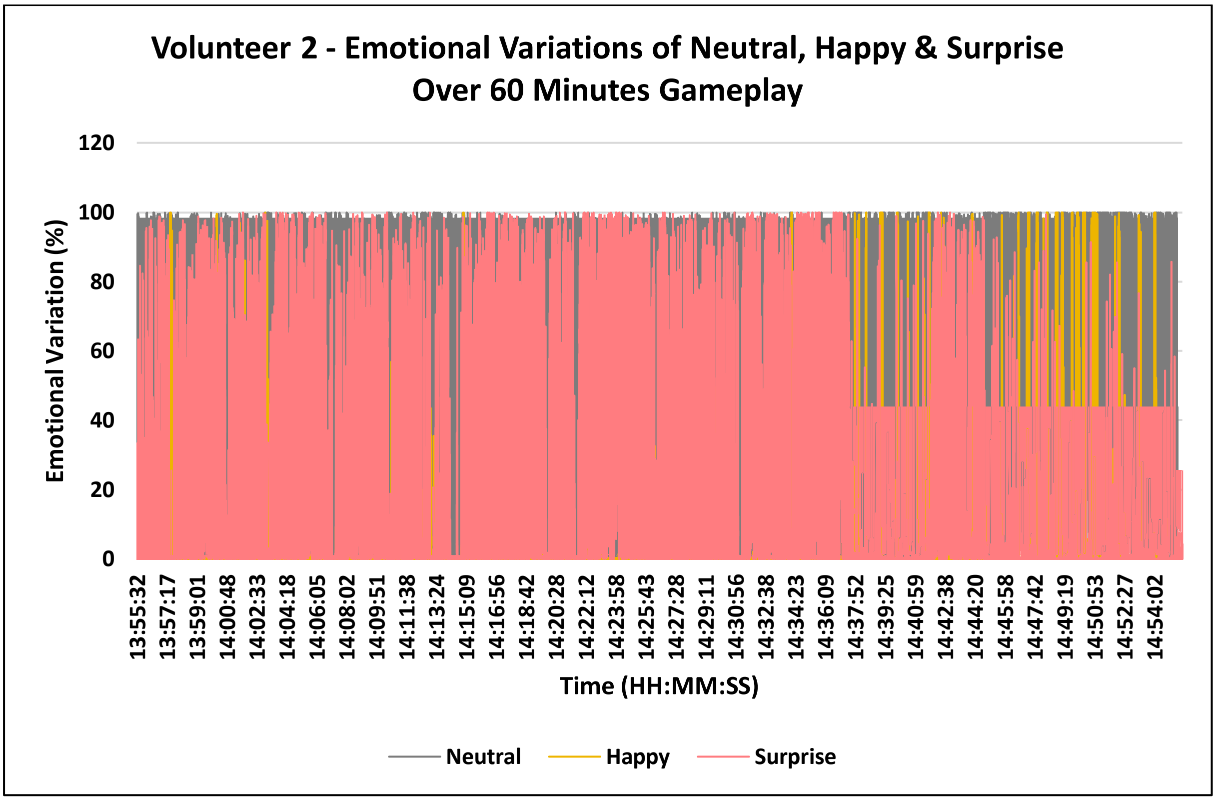
Appendix 7.1



Appendix 7.2

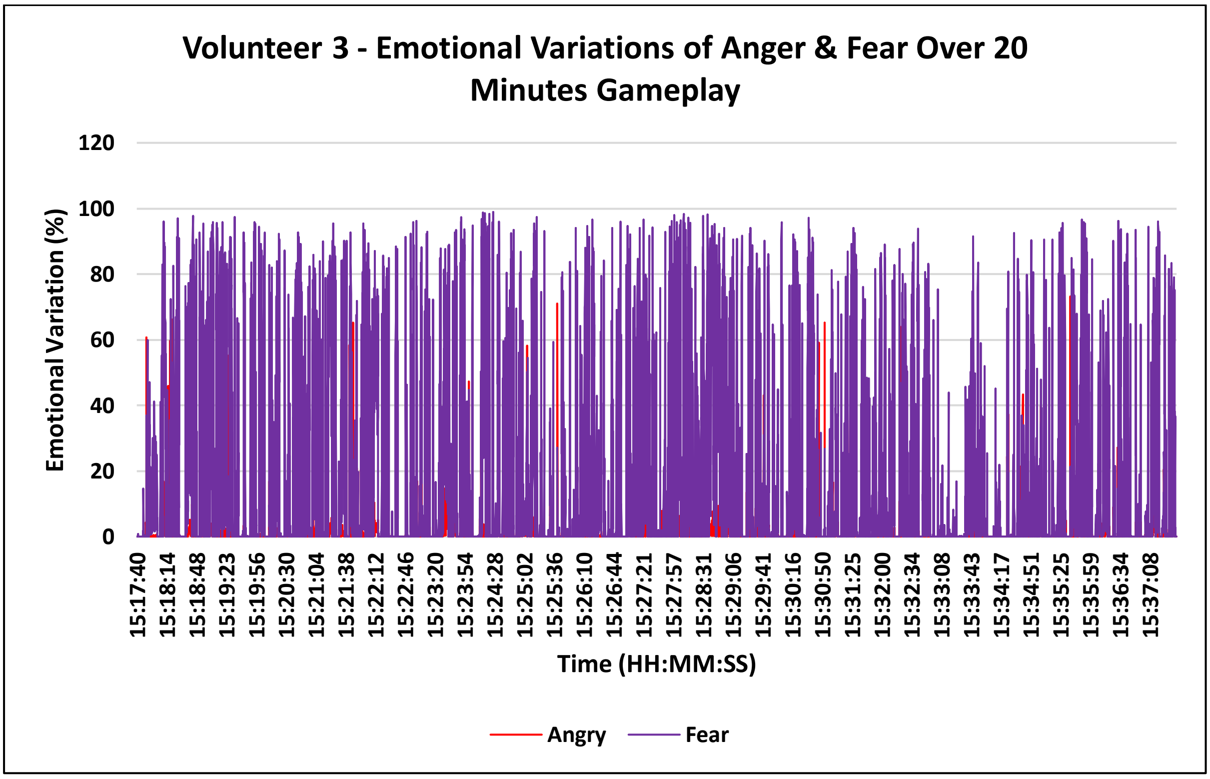


Appendix 7.3

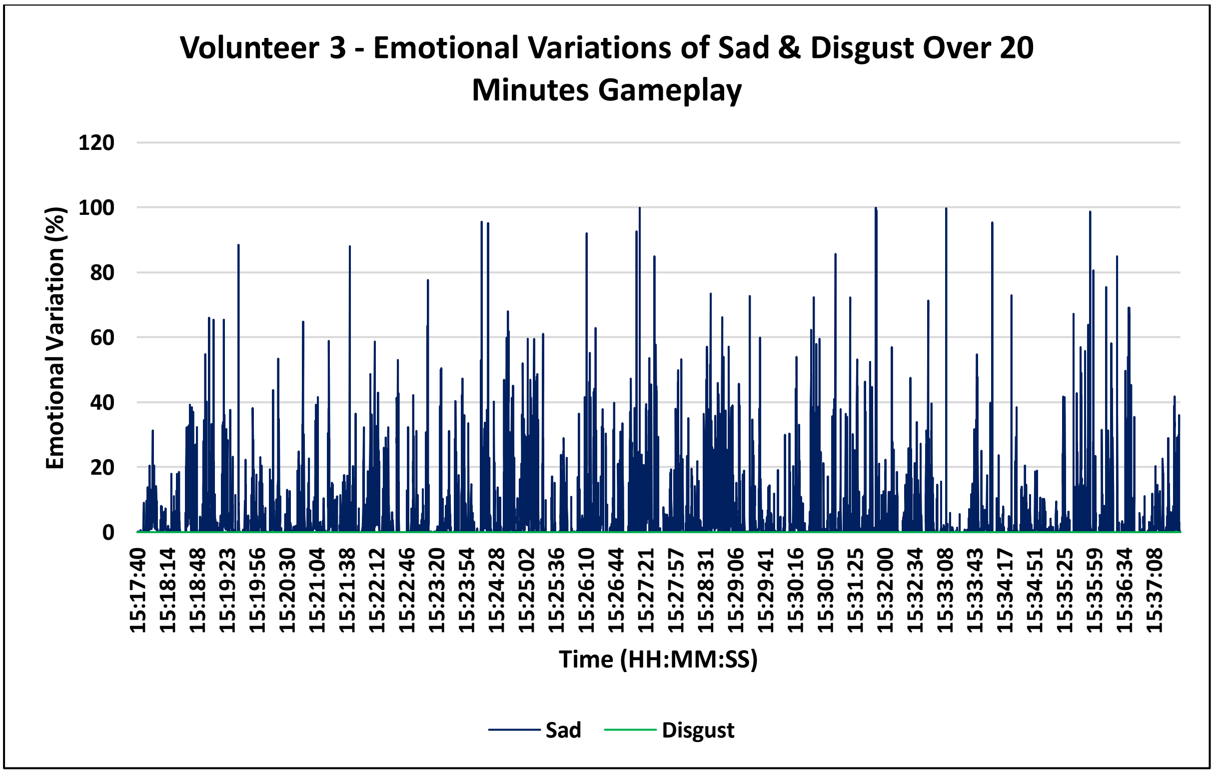


Volunteer 2 - Emotional Variations of Neutral, Happy & Surprise Over 60 Minutes Gameplay
Emotional Variation (%)
120
100
80
60
40
20
0
13:55:32
13:57:17
13:59:01
14:00:48
14:02:33
14:04:18
14:06:05
14:08:02
14:09:51
14:11:38
14:13:24
14:15:09
14:16:56
14:18:42
14:20:28
14:22:12
14:23:58
14:25:43
14:27:28
14:29:11
14:30:56
14:32:38
14:34:23
14:36:09
14:37:52
14:39:25
14:40:59
14:42:38
14:44:20
14:45:58
14:47:42
14:49:19
14:50:53
14:52:27
14:54:02
Time (HH:MM:SS)
Neutral
Happy
Surprise

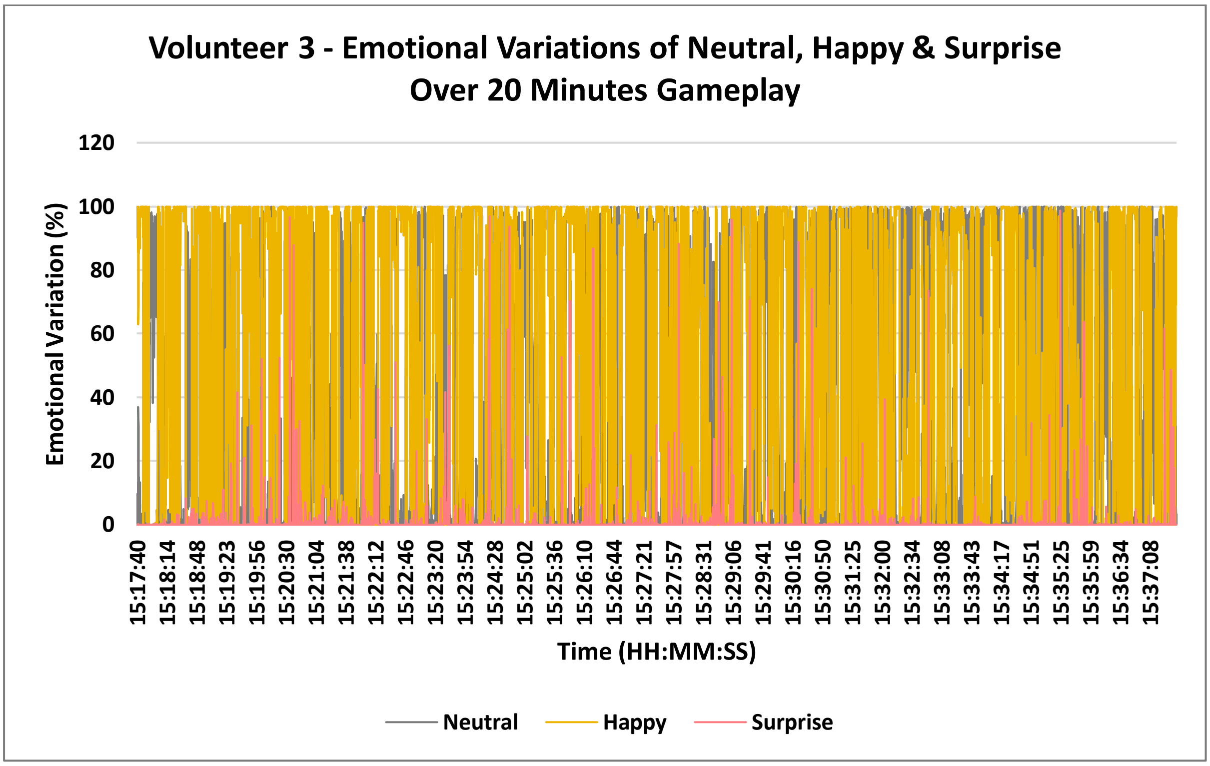
Appendix 8.1



Appendix 8.2

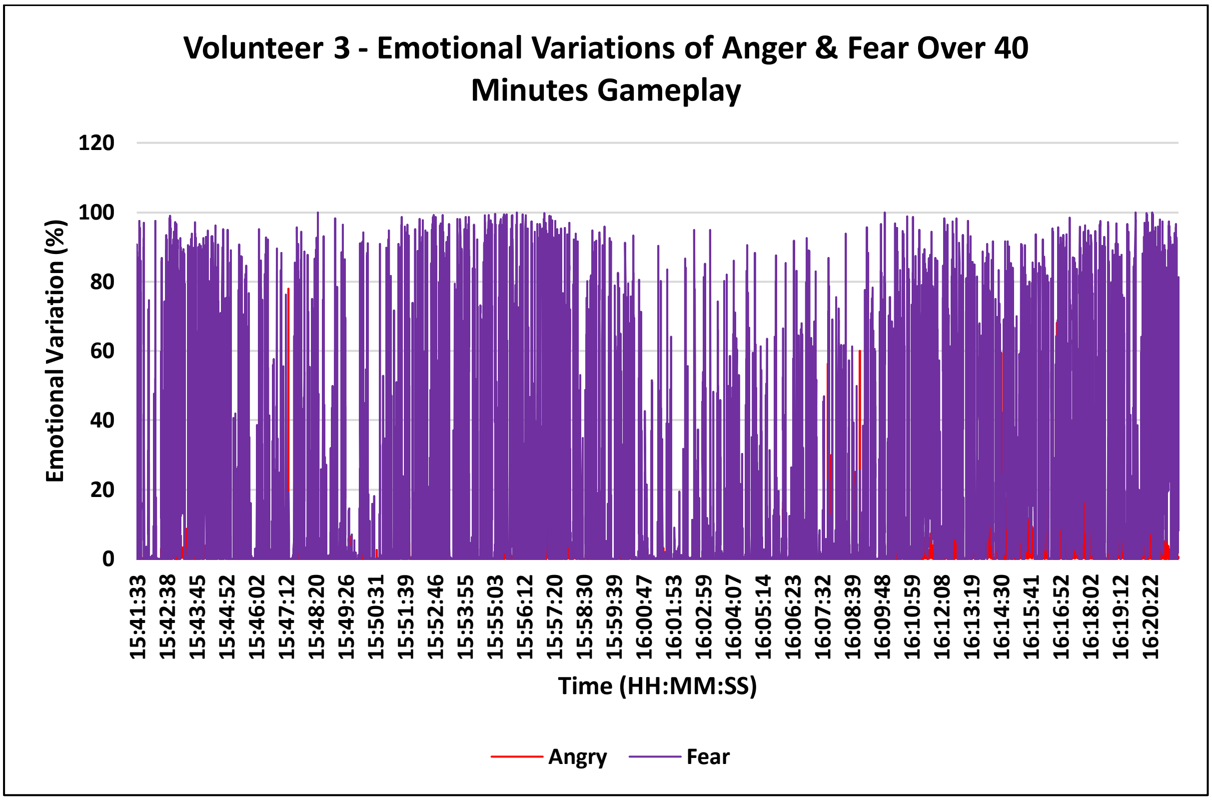


Appendix 8.3

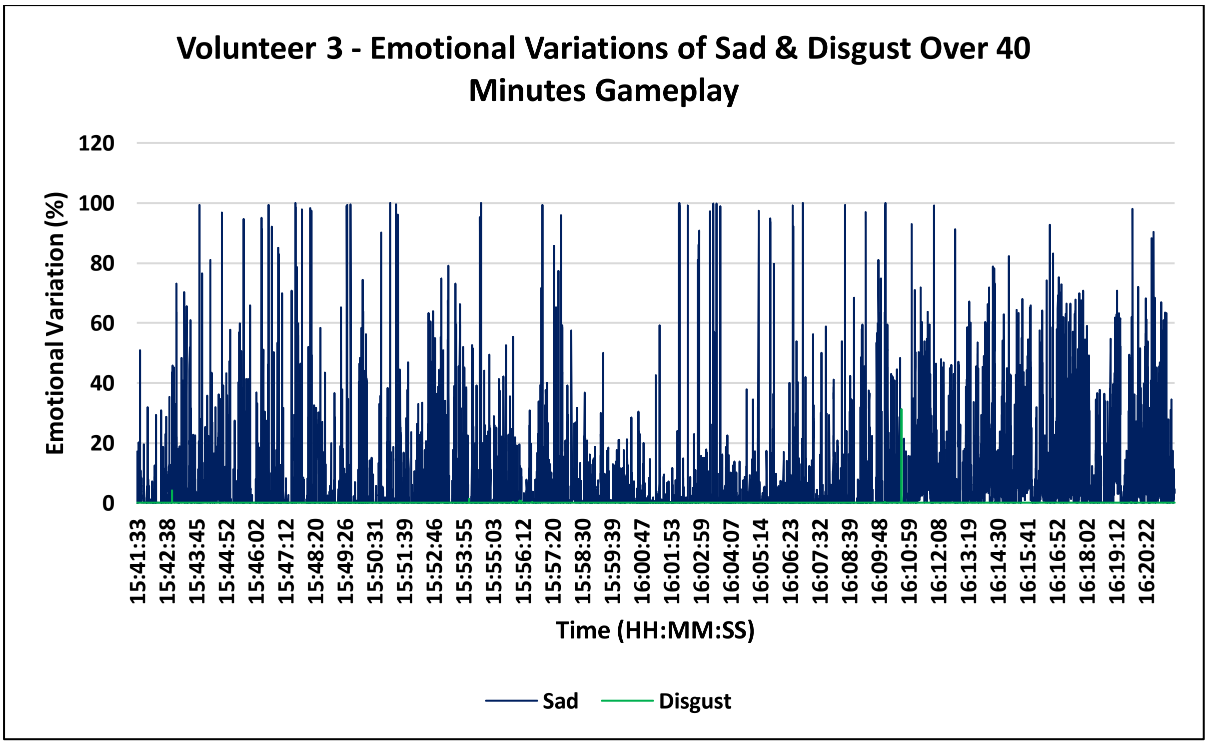


Volunteer 3 - Emotional Variations of Neutral, Happy & Surprise Over 20 Minutes Gameplay
Emotional Variation (%)
120
100
80
60
40
20
0
15:17:40
15:18:14
15:18:48
15:19:23
15:19:56
15:20:30
15:21:04
15:21:38
15:22:12
15:22:46
15:23:20
15:23:54
15:24:28
15:25:02
15:25:36
15:26:10
15:26:44
15:27:21
15:27:57
15:28:31
15:29:06
15:29:41
15:30:16
15:30:50
15:31:25
15:32:00
15:32:34
15:33:08
15:33:43
15:34:17
15:34:51
15:35:25
15:35:59
15:36:34
15:37:08
Time (HH:MM:SS)
Neutral
Happy
Surprise

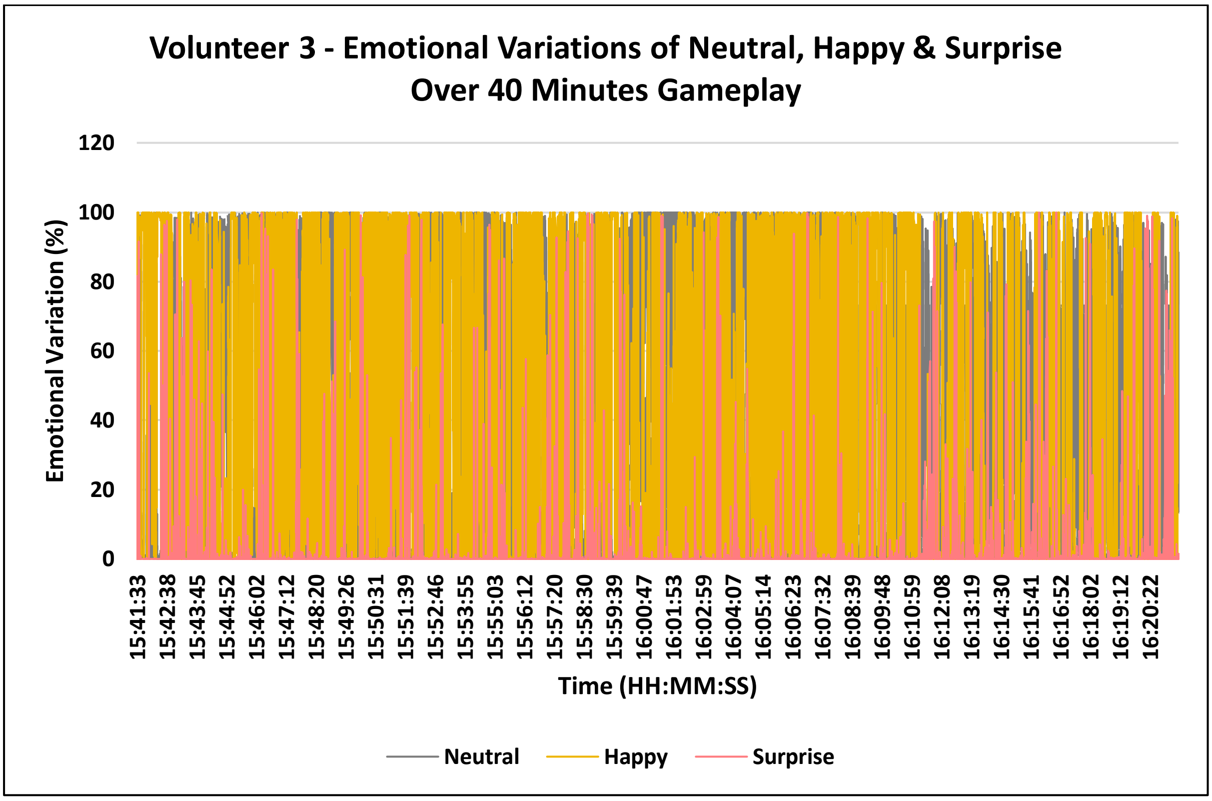
Appendix 9.1



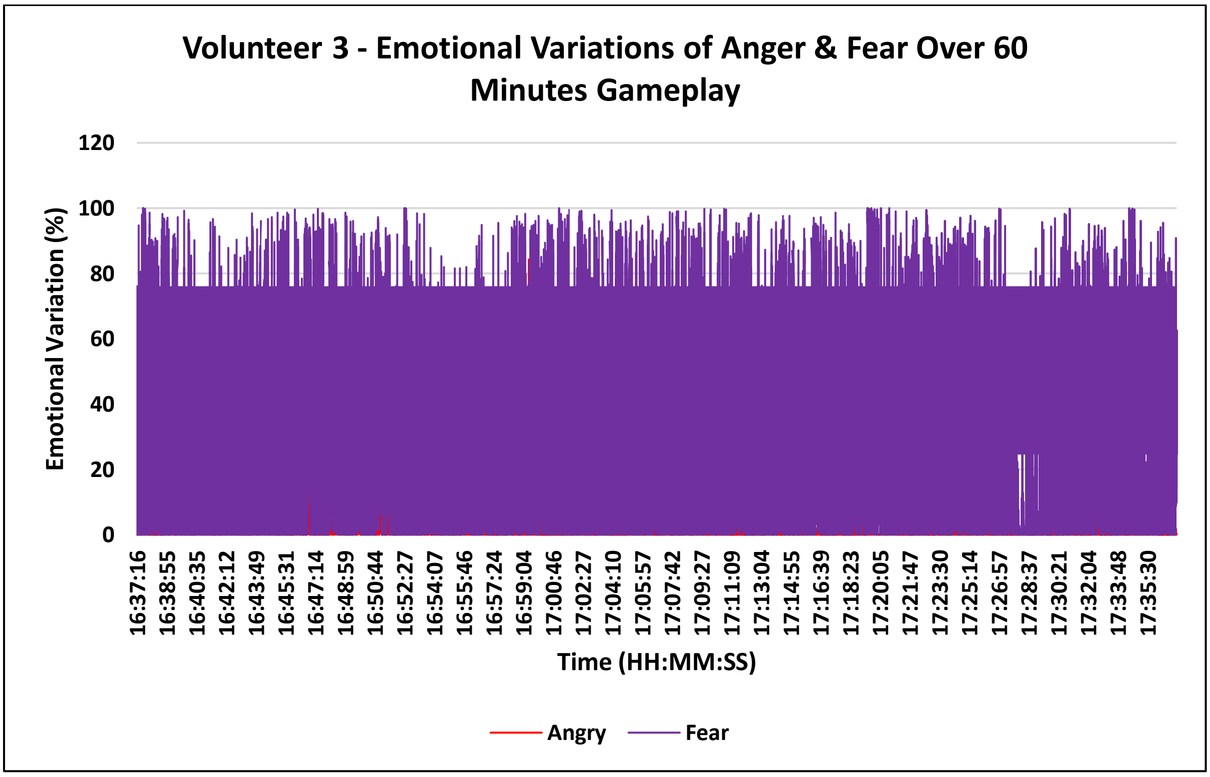
Appendix 9.2



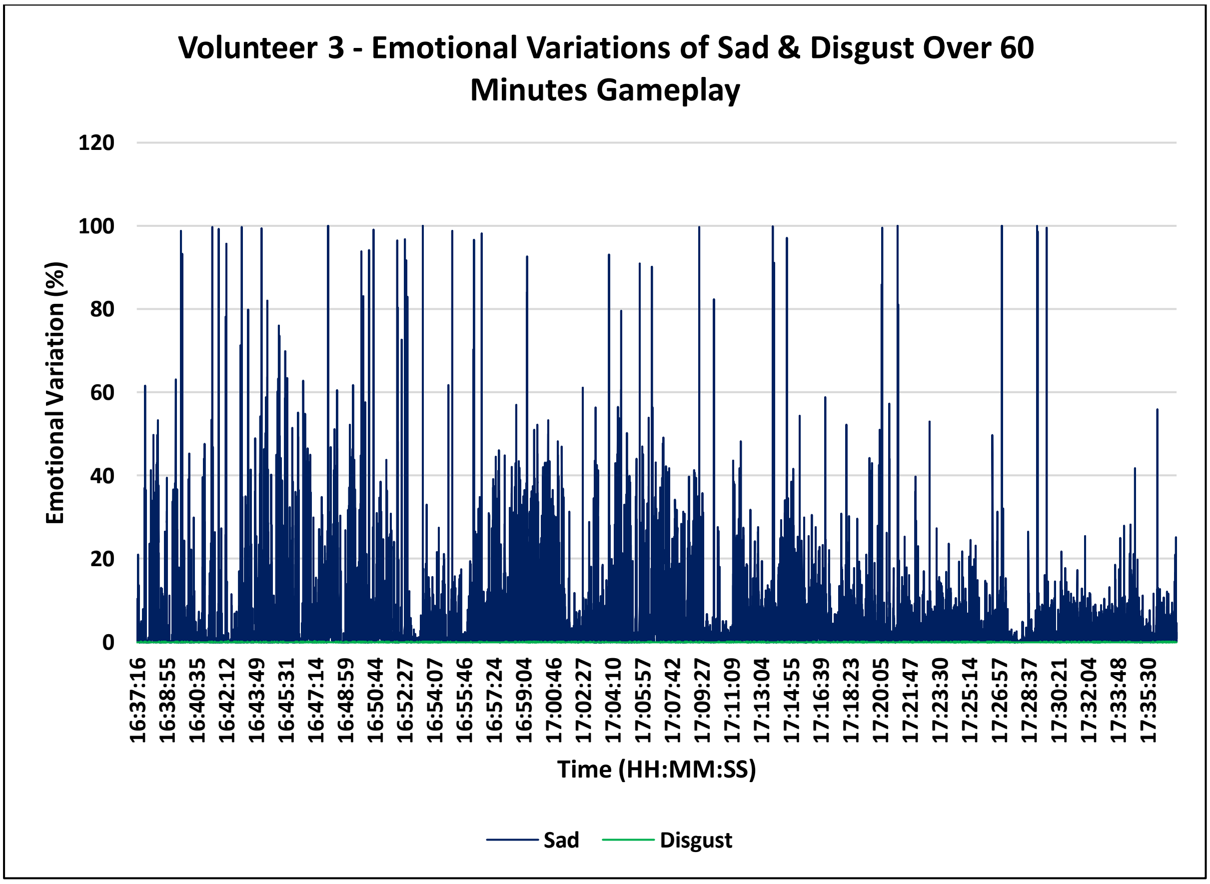
Appendix 9.3


Volunteer 3 - Emotional Variations of Neutral, Happy & Surprise Over 40 Minutes Gameplay
Emotional Variation (%)
120
100
80
60
40
20
0
15:41:33
15:42:38
15:43:45
15:44:52
15:46:02
15:47:12
15:48:20
15:49:26
15:50:31
15:51:39
15:52:46
15:53:55
15:55:03
15:56:12
15:57:20
15:58:30
15:59:39
16:00:47
16:01:53
16:02:59
16:04:07
16:05:14
16:06:23
16:07:32
16:08:39
16:09:48
16:10:59
16:12:08
16:13:19
16:14:30
16:15:41
16:16:52
16:18:02
16:19:12
16:20:22
Time (HH:MM:SS)
Neutral
Happy
Surprise

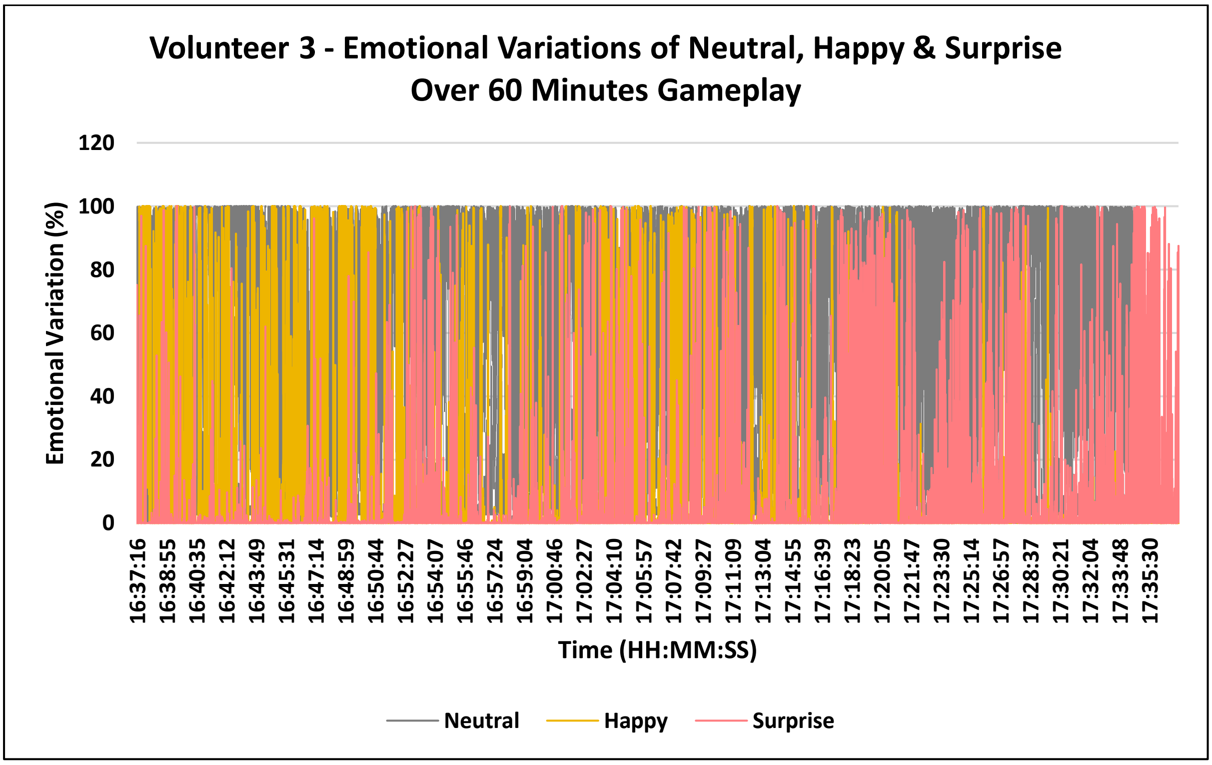
Appendix 10.1



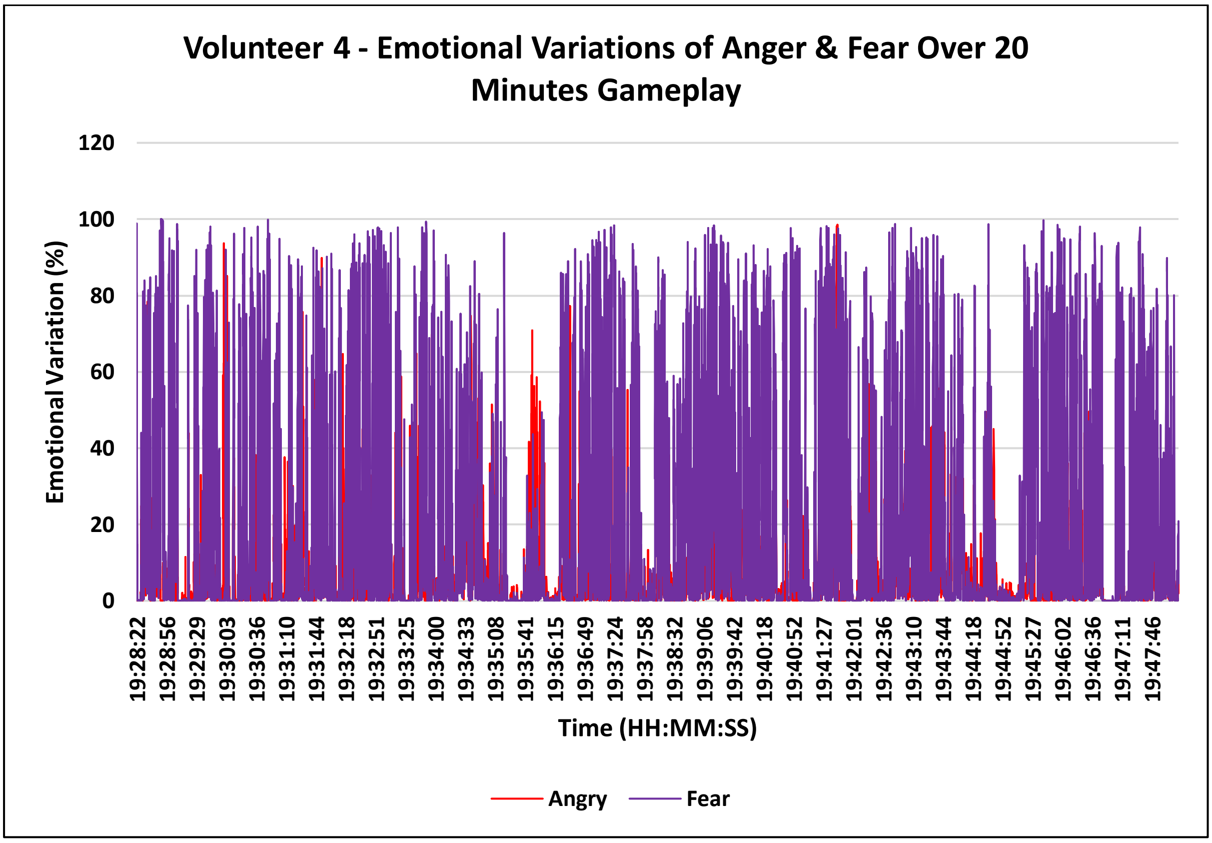
Appendix 10.2



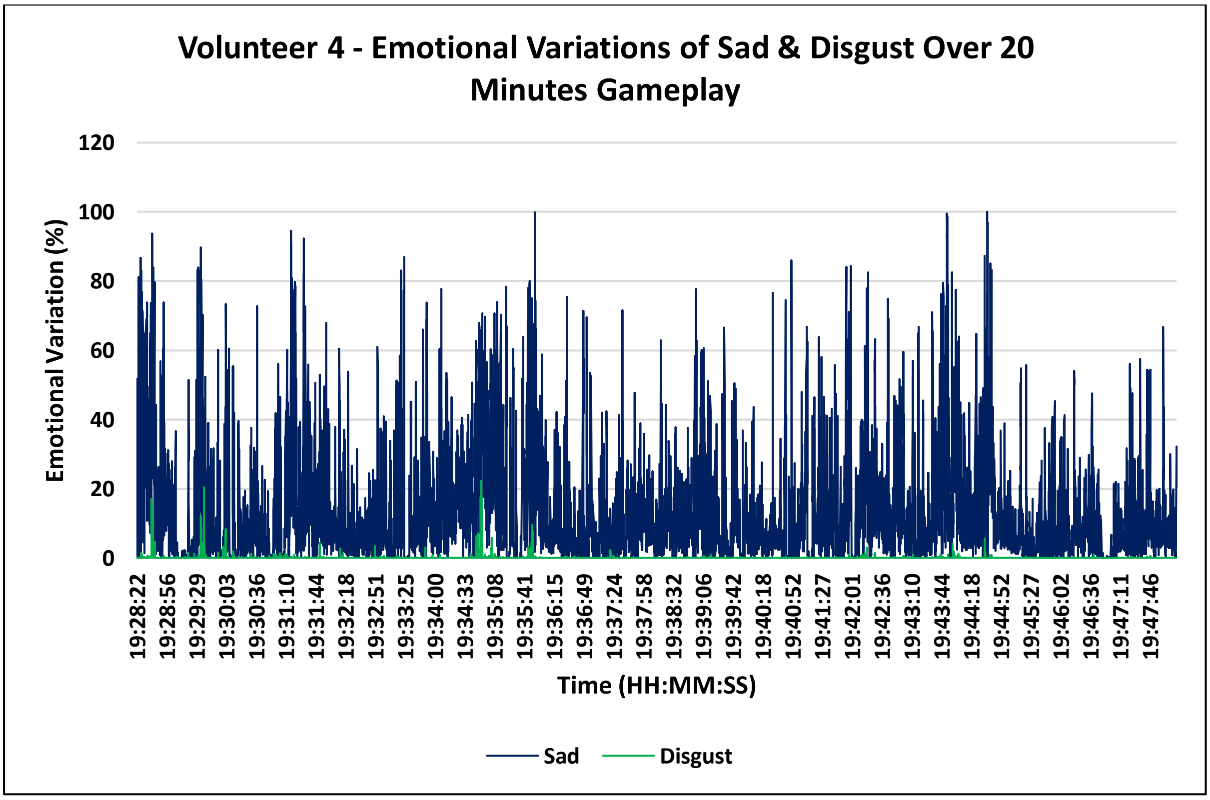
Appendix 10.3



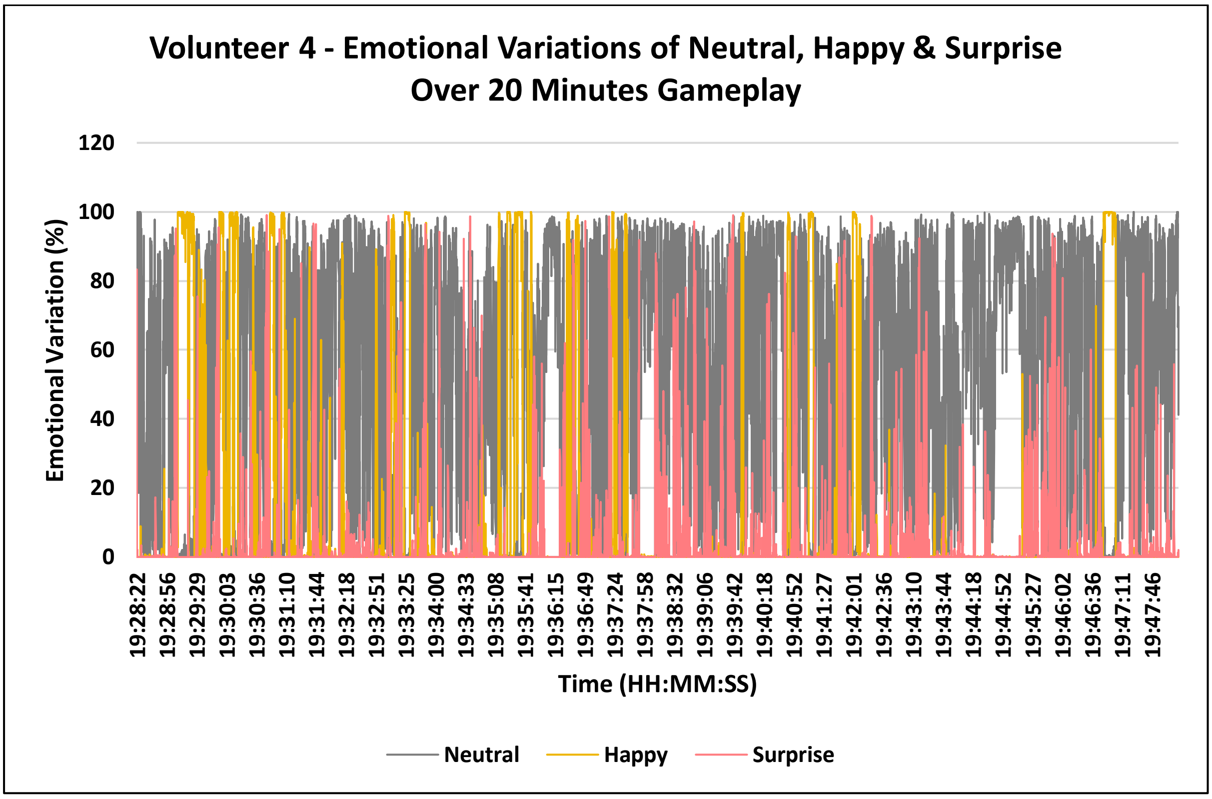
Appendix 11.1



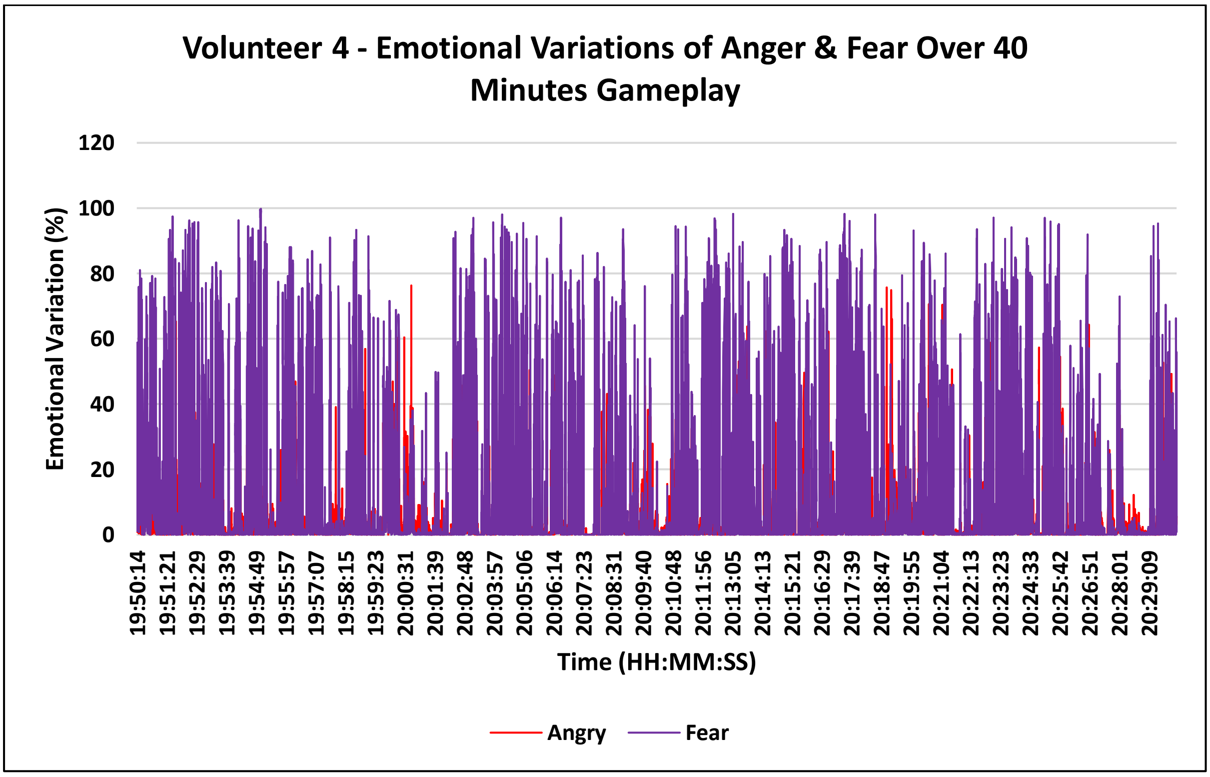
Appendix 11.2



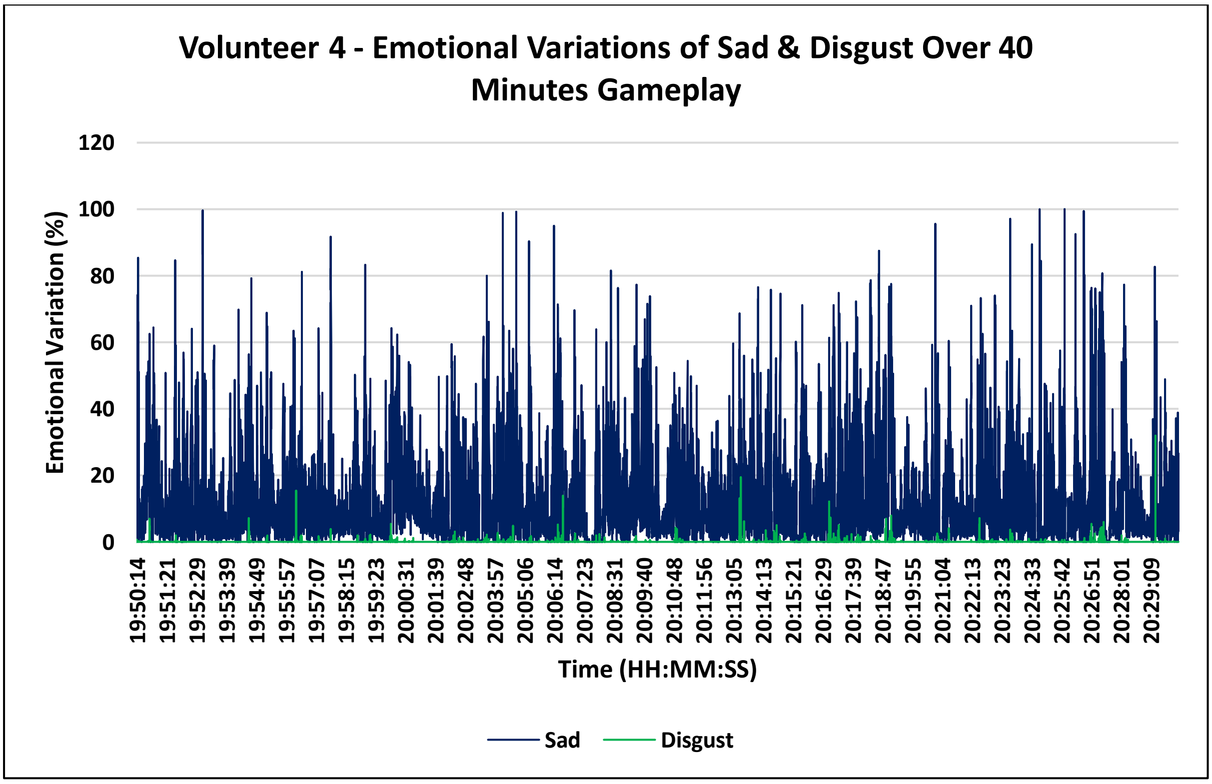
Appendix 11.3



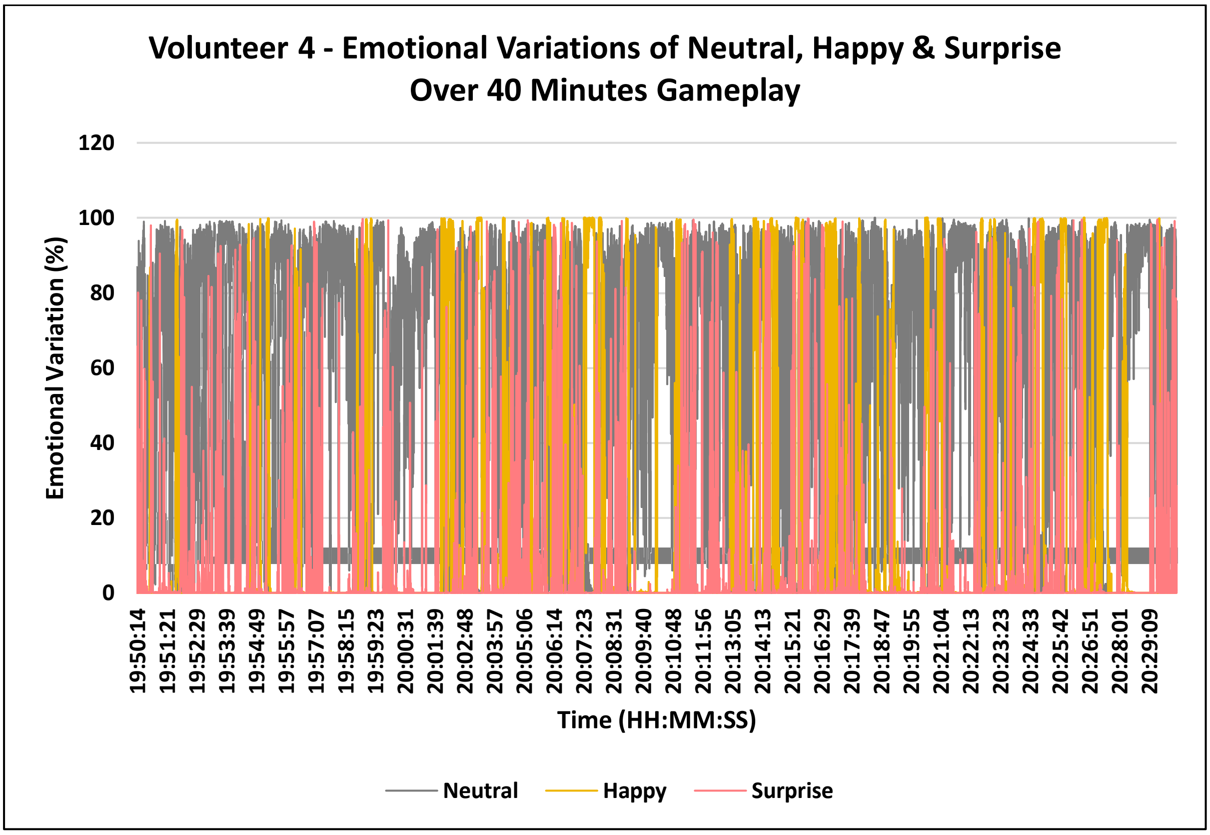
Appendix 12.1



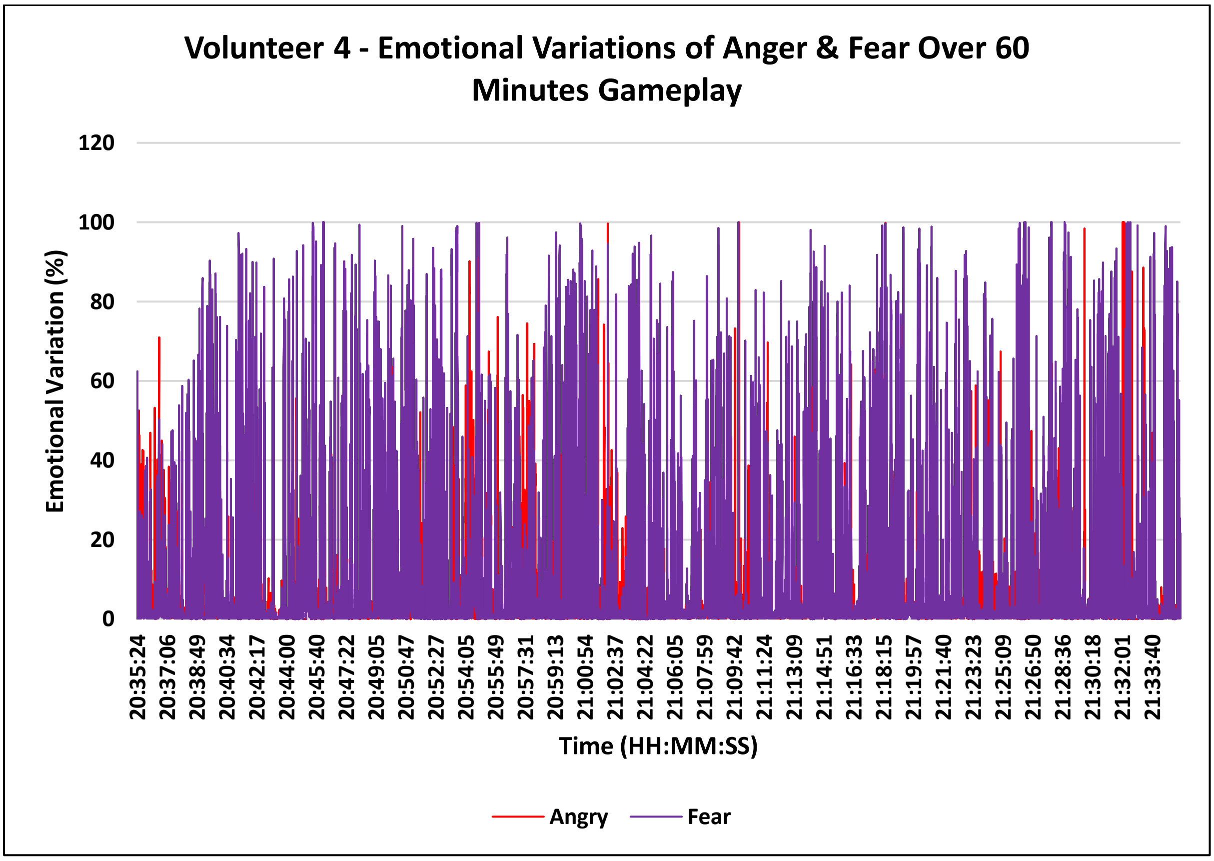
Appendix 12.2



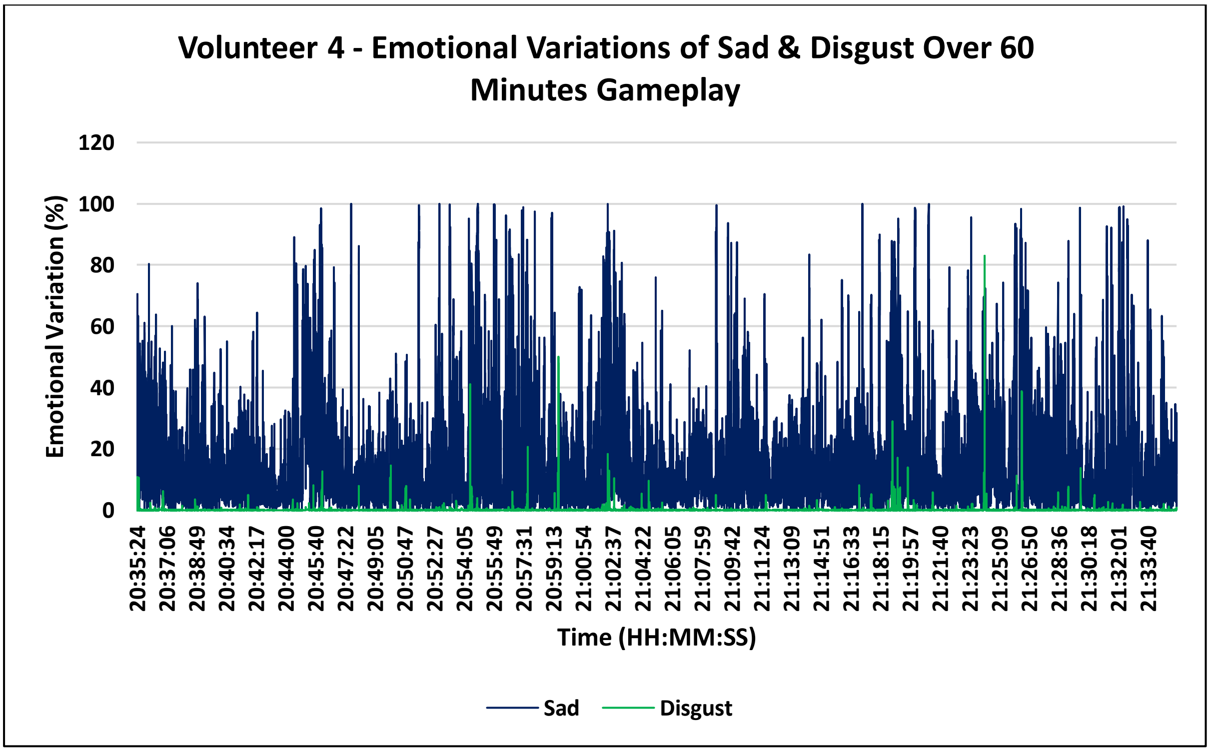
Appendix 12.3



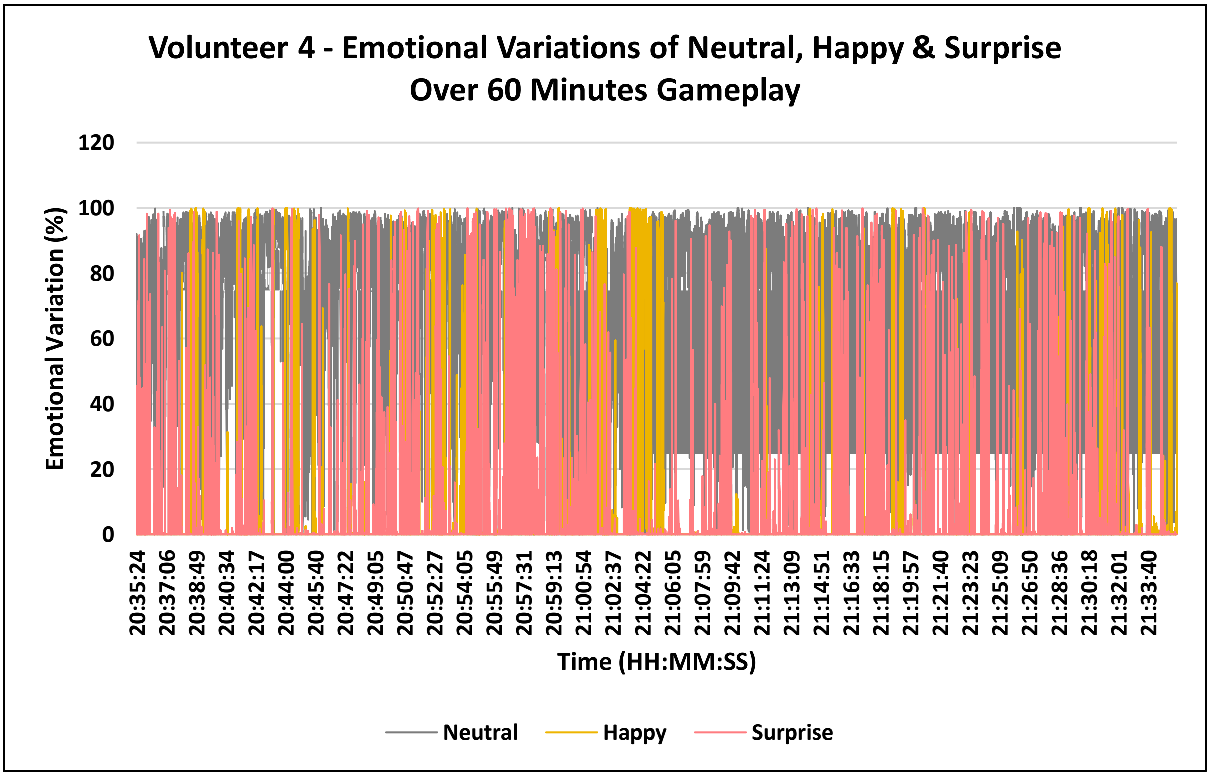
Appendix 13.1



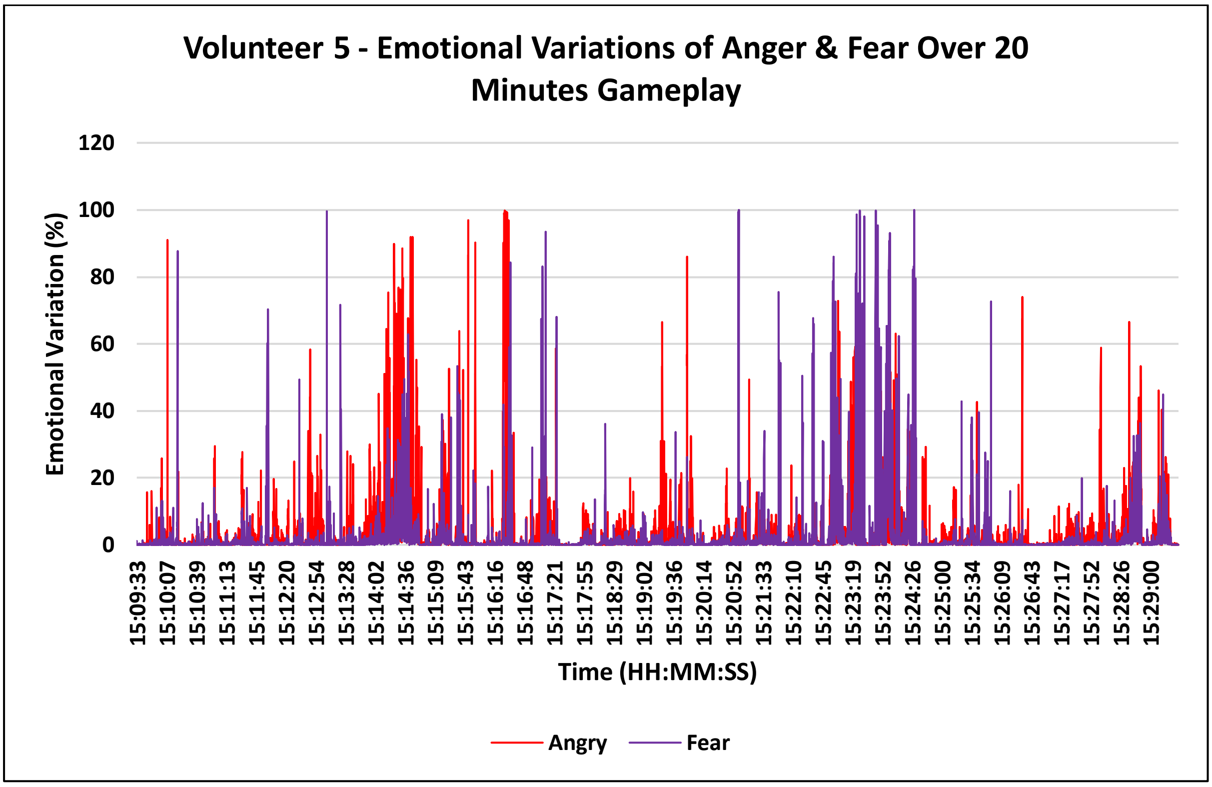
Appendix 13.2



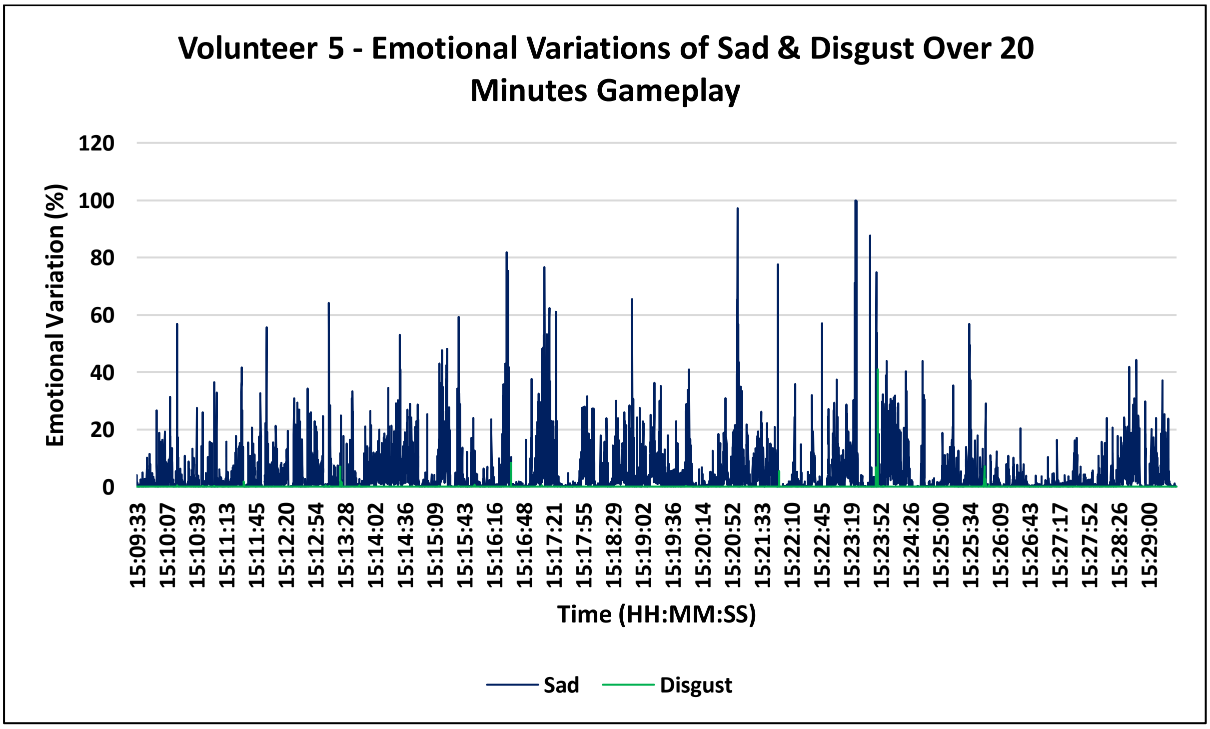
Appendix 13.3



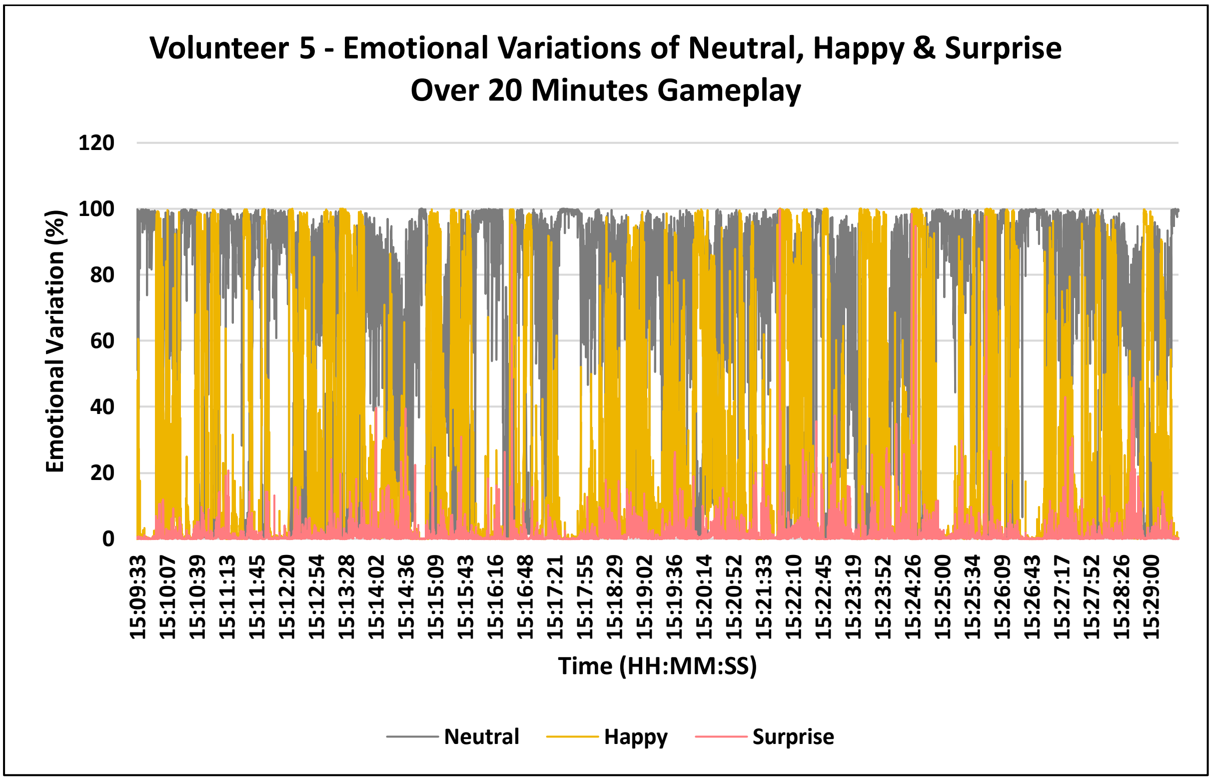
Appendix 14.1



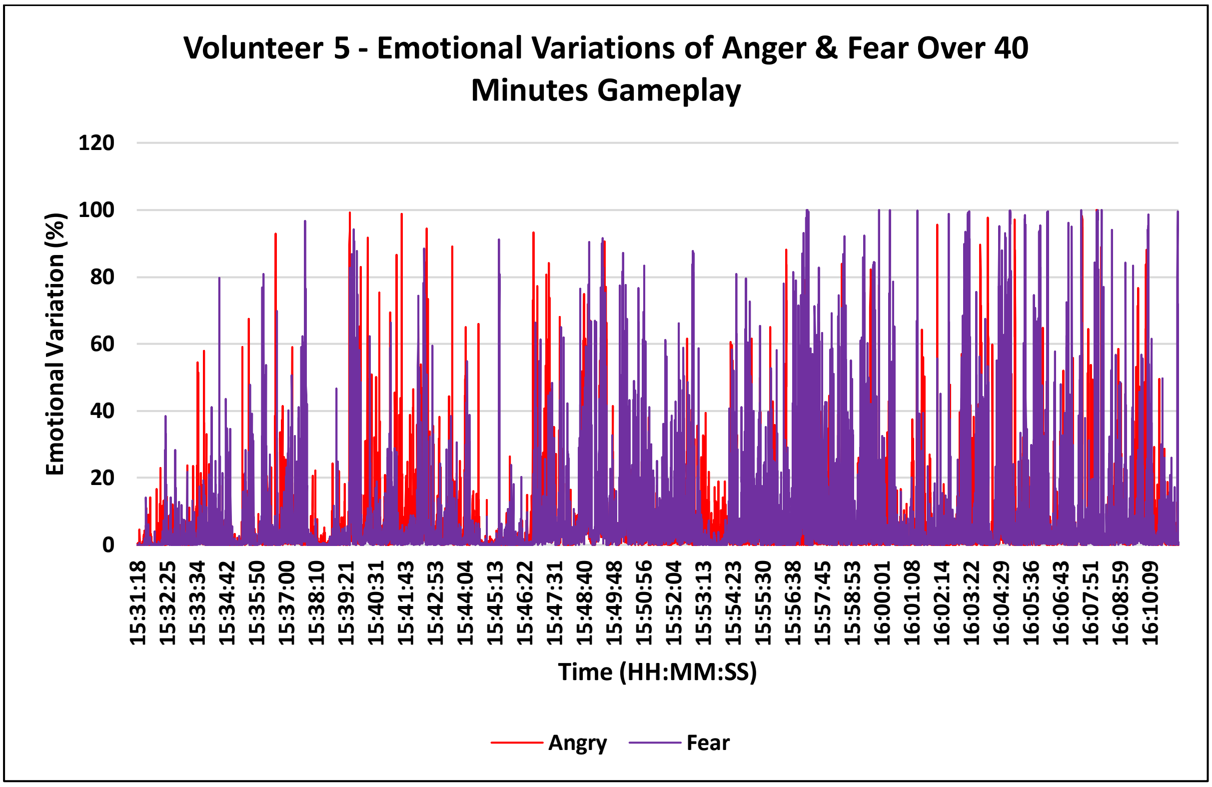
Appendix 14.2



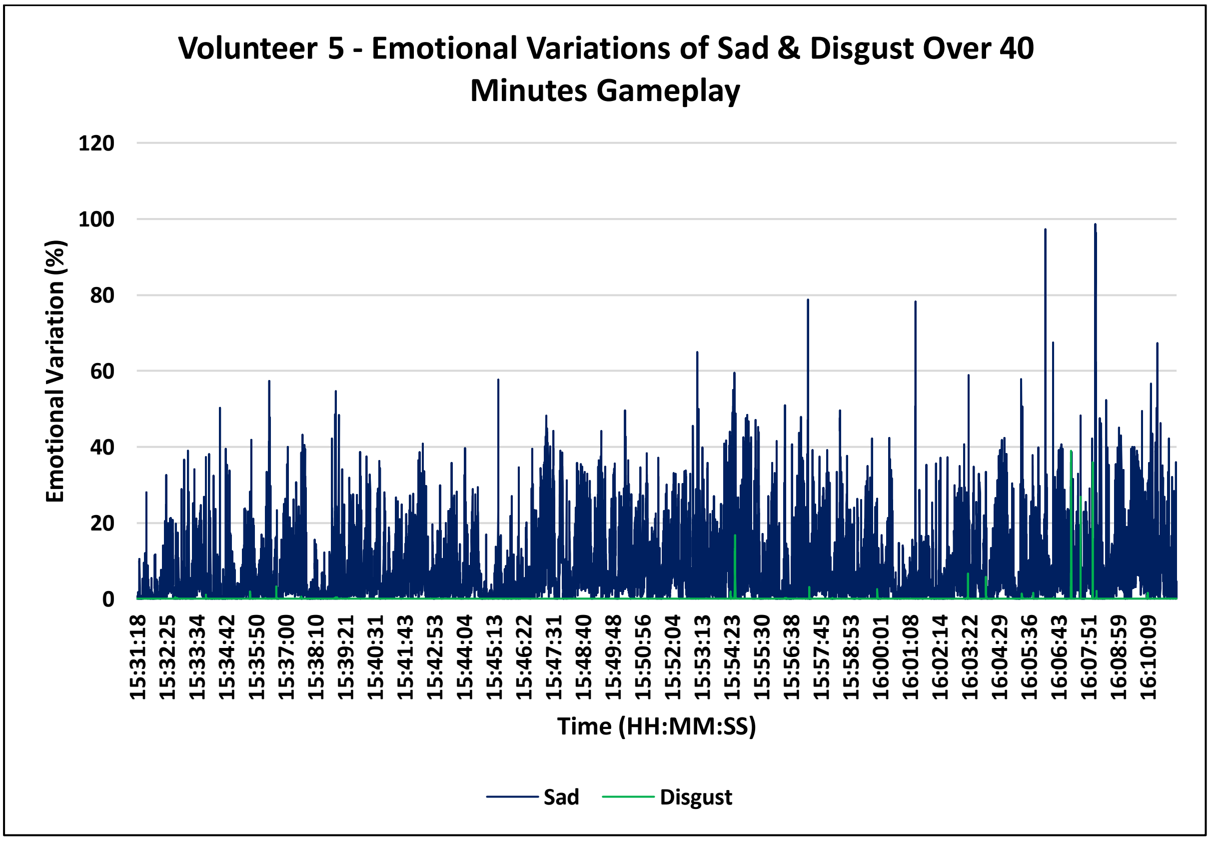
Appendix 14.3



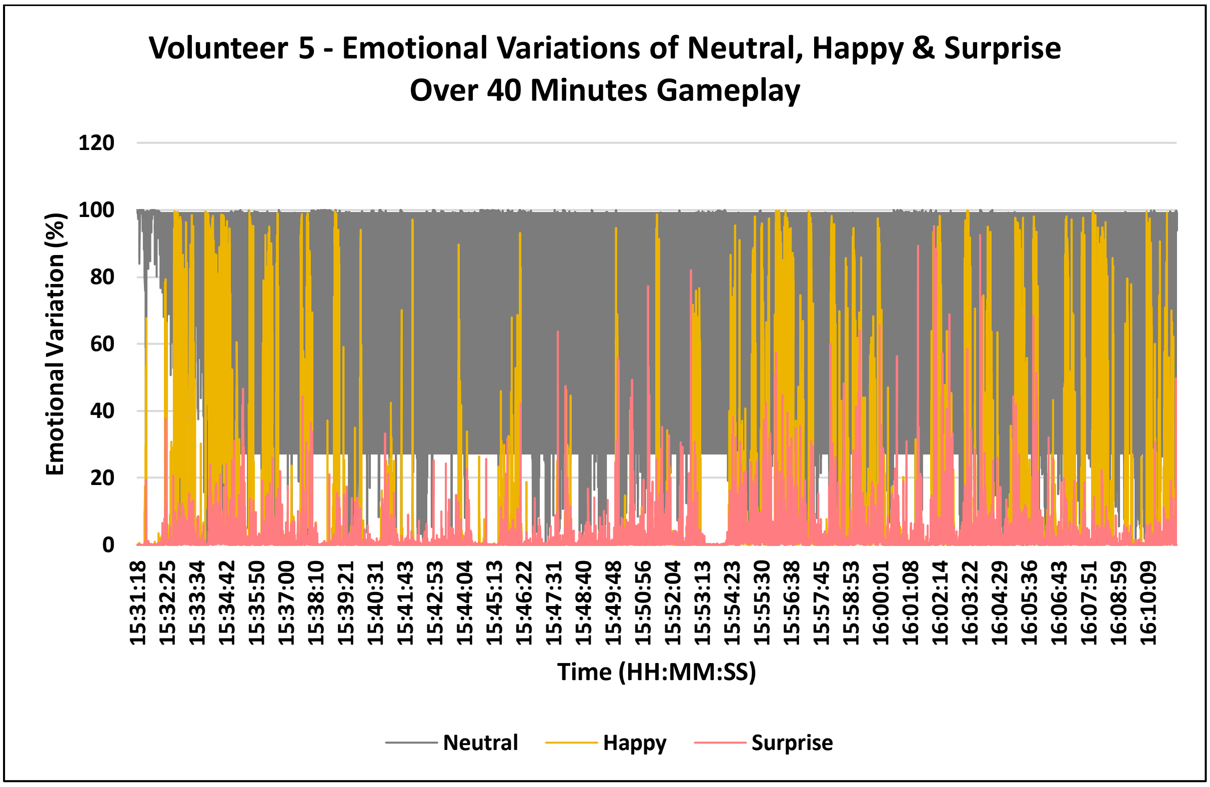
Appendix 15.1



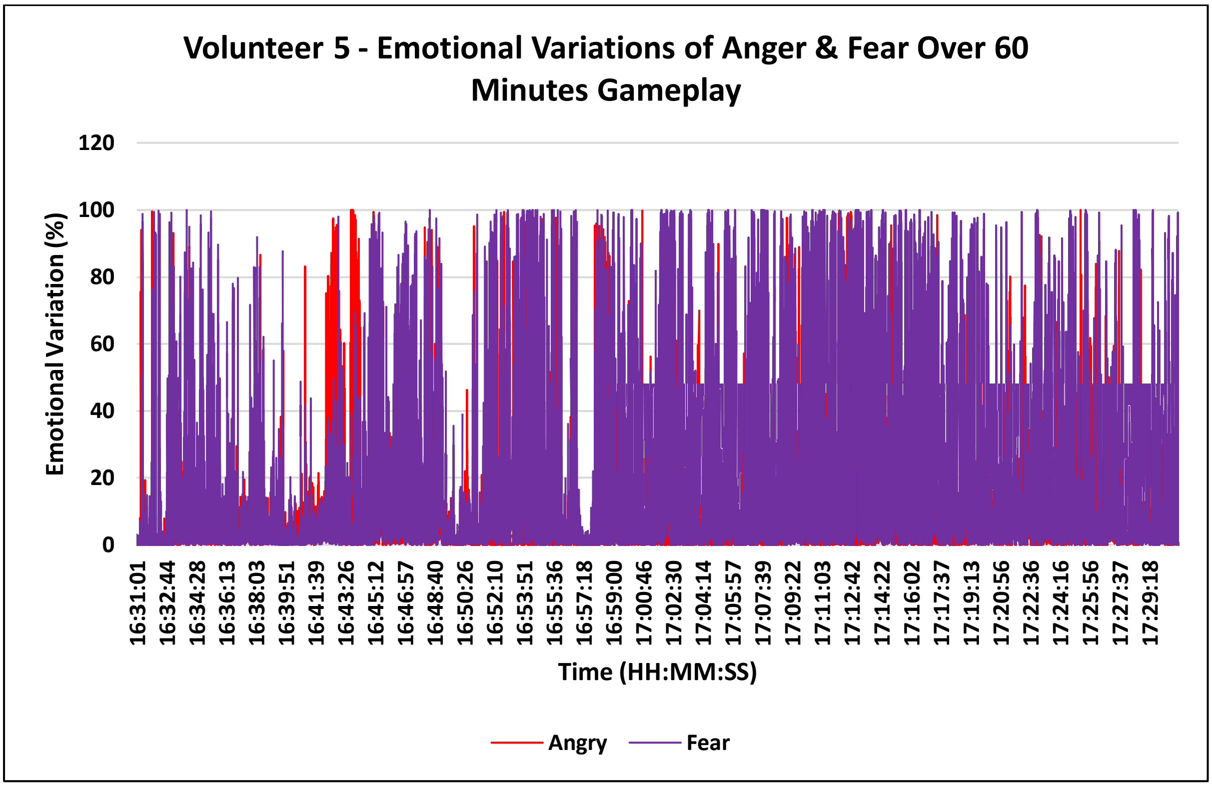
Appendix 15.2



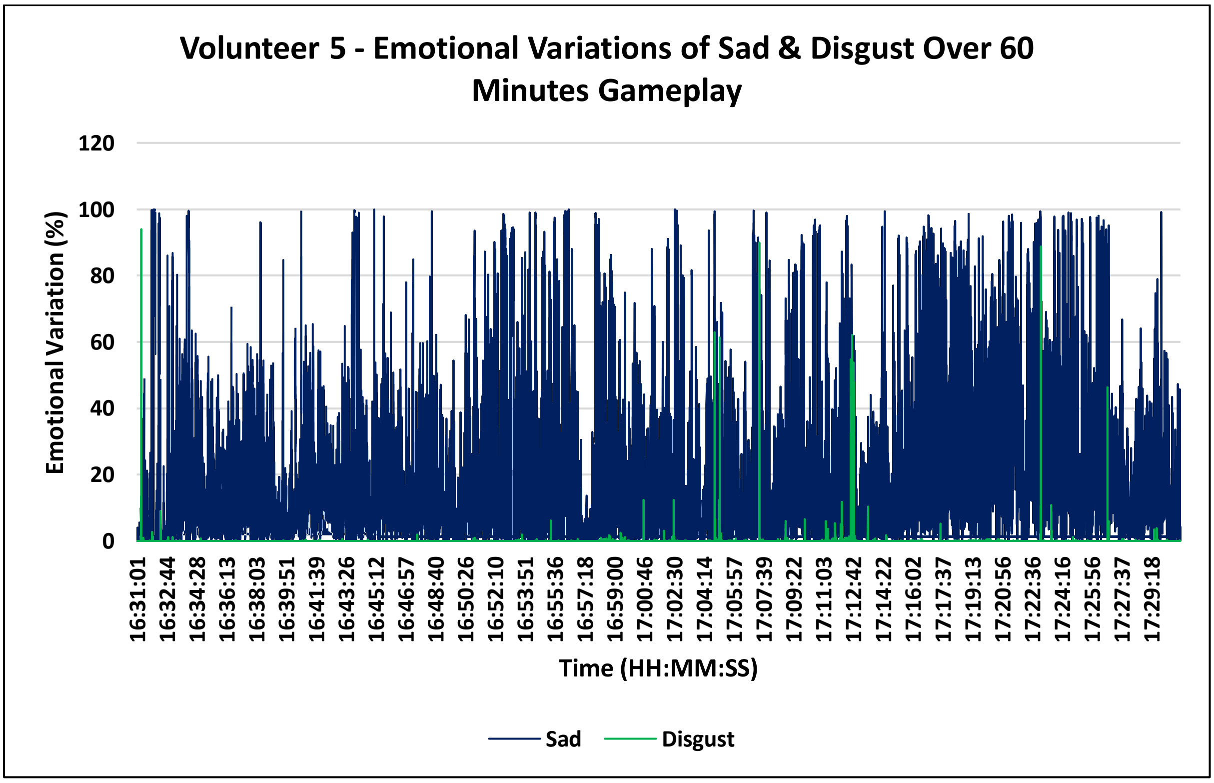
Appendix 15.3



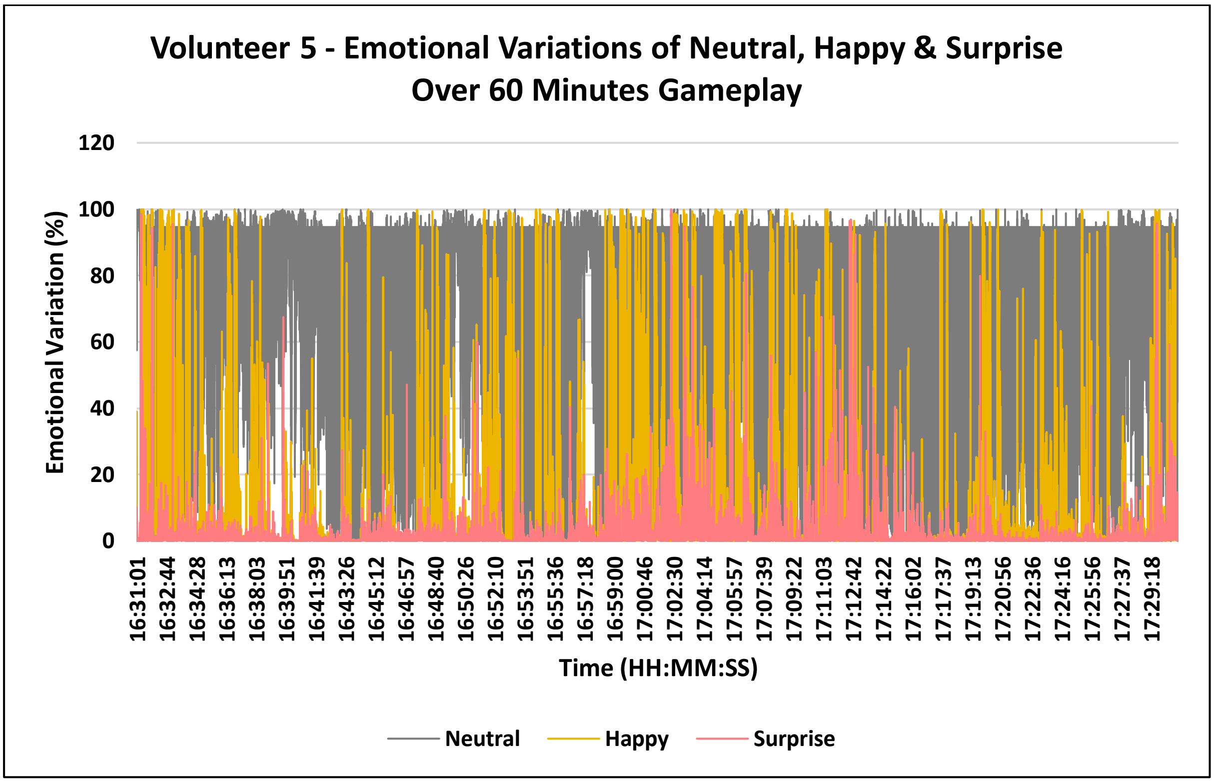
Appendix 16.1



Appendix 16.2



Appendix 16.3



Appendix 17

Name:

5 responses

Volunteer 5
Volunteer 4
Volunteer 3
Volunteer 2
Volunteer 1

How did you feel about the game during the 20 minute session?

5 responses

The game was fun and engaging during the session.

Very good as I was able to fully focus on the game.

Relaxed and good to go. Feeling great that I chose team choices as I could work with others playing the game.

The 20 minute session was the first time I played the game ever, it was exciting and interesting to see how the game played and I felt quite interested and excited to get into a game straight away. I was quite happy with the game in this session and I was just casually getting to grips with how it played.

Pretty Good, was enjoyable to play the game and had most of the emotions shown during the 20 minutes.

How did you feel about the game during the 40 minute session?

5 responses

Similar to the 20 minutes, 40 minutes felt good too. Was able to get a lot more of my emotions out as it was a longer period of time

Felt like a groove was kicking in as I started to get used to the game in action. Still felt good during the gameplay.

Lost concentration a bit but still carried on well. So I was happy.

This session was where I played as well as I could, I was concentrating a lot and whenever I had died or missed a shot I felt angry and annoyed as I wanted to improve my skills in the game. When I got a good kill or an unexpected kill I was quite excited and happy and laughing a lot to celebrate the kill I just got.

When playing the game it was still exciting, which I'm used to gaming for that amount of time.

How did you feel about the game during the 60 minute session?

5 responses

As the time went on, things got a bit tiring and so my emotions got a bit underwhelmed and not accurate. But still had fun during the hour session

Felt a little bit tired heading towards the end but didn't stop me from producing the gameplay that I did. Overall a solid 60 minutes.

I felt as though my focus was going and felt frustrated and tired.

The beginning of this session was similar to the end of the 40 min session, quite engaging and I was happy at times and angry/sad at times depending on if I had gotten a kill or I had died. Towards the end of this session however, the game started getting repetitive and boring, so I started feeling more neutral/uninterested the majority of the time with moments of sadness and happiness.

for the 1 hour session, my hands was a bit tired and i had a bit of eye strain due to how close i was to the screen. note that it was a competitve game as well

Overall, how did you feel about the game over the 3 sessions?

5 responses

The game was perfect for the test as it was able to capture all of my emotions within the time span.

I felt like the game was a really clever concept with the given choices of what you could do. Whilst being challenging in some cases, it was also fun to play.

They went well but the longer sessions brought me a lot of frustration. However the idea is very innovative.

The game was quite fun, made me happy and laugh most of the time with a few angry/sad moments. Towards the end again it was quite monotonous and repetitive but still quite fun!

The overall session was really interesting and i have a sense of how emotion recognition works durinng the game test.

Any other comments?

4 responses

Hope you get a good grade ;)

It is good to try something you never done before, for all you know you could be really good at it like I had shown in the gameplay.

I'm not that good at fps but I liked the experience.

I would've like it if the gameplay test overall was less than 2 hours due to the fatigue of the eyes and hands, Thank you.