

GESTURE BASED UI DEVELOPMENT

INTRODUCTION

This design document is for a 2D video a thrilling suspense and nerve testing game. In this document the reasons for our choosing of this sort of game will be outlined, software that we chose to go with and the hardware that was decided on as the best for this game, the mechanics, menus and overall structure of the game. How the player will interact with the game and the options/settings the player will have access to in the game's menus.

We have chosen to design a 2D game it's called "Bomb Defuse". This game makes the player think and act under pressure, it tests the players never, The player receives instructions and steps on how to go about defusing the bomb that he gets displayed with. The bomb consists of few steps that are needed to be accomplished before the bomb is defused.

1. Player needs to cut the right wire to unlock the next part of the bomb but every time that the wrong wire is cut the time on the timer gets decreased by half. User can only make one mistake with the wires.
2. Once the right wire is cut the code that needs decrypting will appear to the user. User needs to follow the bomb manual.
3. When the code is decipher and user has the correct code he enters it and wins the game. If the wrong code is entered the bomb will explode and ending the game on a defeat

The overall design of the game will be designed by the developers the assets e.g. wires, timers. We will try not to use any assets from the asset store and make it more difficult for ourselves and give ourselves a bit more work to make it original and make the game ourself.

RESEARCH

Before we decided on what hardware device to use in this game we needed to research and decide on which would suit the game the best we had three options to chose from Myo Connect, Wii Controller or speech recognition.

MYO CONNECT

The Myo Armband was announced by Thalmic Labs in February 25th 2013. It was announced by a video on YouTube where it was linked to a pre-order page.

Just in two days they had 10 thousand units pre ordered.

The Myo armband lets you use the electrical activity in your muscles to wirelessly control your computer, phone, and other favourite digital technologies.

The armband itself contains 8 pods with a proprietary EMG sensor on each, in order to detect muscle movements around your forearm and hand gestures. It also comes with a gyroscope, an accelerometer and magnetometer for the tracking of arm movement e.g. spinning.

What games are used with the armband/Where is it used ?



Productivity

Adobe Reader Connector
Android Studio Connector
Atom Connector
Connect Myo Windows Phone
Deckset Connector
Duolingo Connector
Global Mouse Control Connector
Gmail connector

Multimedia

10 ft Media Experience
Ableton Live Connector
BS Player Connector
Cinema 4D Connector
Deezer Connector
Elmedia Player Connector

Games

Agario Connector

Air-Guitar
Armanita Design Adventures Connector
Audiosurf Connector
Back to the Future Connector
Broken Age Connector
Civilization Beyond Earth Connector
Civilization V Connector

Conclusion

Myo is a device that me and Sammar had great interest in how it worked how it felt it was a hardware that we had not seen before. We wanted to use this hardware in our project but we felt that it was not the right hardware for this game. It would not be a good experience for the user and it was unnecessary extra work if we would have to continue with this hardware for this application.



WII CONTROLLER

Wii Controller similar to the myo armband requires hand movement. The controller is designed for the Wii console which is an interactive game console which requires players to move the controller while pressing buttons at the same time to accomplish tasks in the selected game.

The reason that we decided to add this device to the selected devices that we would try out in the project was because one of us has the console therefore we had the controller and knew how to use it. Due to the corona Virus this was a very valid option since the hardware was hard to acquire over this difficult time.



Conclusion

The reason that we decided not to go further then installing it on our PCs was because like with the Myo connect this device did not suit our project it was the right device but not for this project that we were using. Also we were not able to find an sdk for this device all that we had found on the web was API's and we did not know if it was valid to have it running with an API rather than the SDK.

VOICE RECOGNITION

This is a computer software program with the ability to decode the human voice. The first automatic speech recognition was made in 1952 to recognize numbers spoken by the user. Many of the ASR (automatic speech recognition) programs require the user to train the program to predict the text more accurately. For voice recognition to work you need a computer with a sound card and microphone.

Examples of where you can find voice recognition is Digital assistants and Car Bluetooth. Siri and Amazon echo use voice recognition to interact with these assistants. This voice recognition can also be used in cars to make simple commands such as "answer call" so the driver does not have to look at his/her phone while driving.



Conclusion

At the end we had to use voice recognition at the end because we were able to use a wii controller or the myo armband. This was the easiest one to implement but the game it strange to use with voice since the original idea of the game wasn't designed for this. There were loads of examples for voice recognition games on steam and the google play store. Unity documentation for voice recognition was easy to follow and YouTube had many tutorials for us to follow whenever we ran into issues.

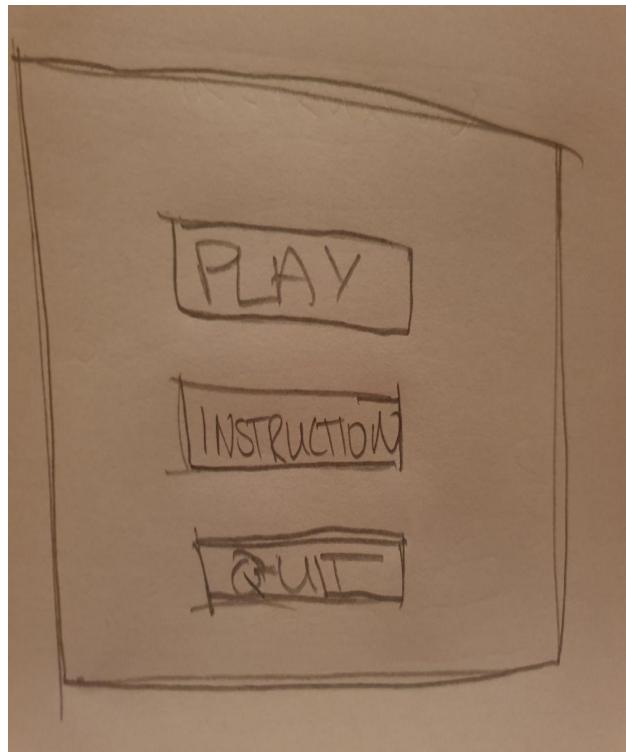
GAME DESIGN

FRONT END

When a player runs the game he/she should see the home page pop up with a various amount of options that they can chose from depending on what they want to do.

1. "Start Game" option to start the game and go straight into the game.
2. Instructions for the game such as
 - a. Phrases that the user is allowed to use
 - b. How to lose/How to win
 - c. And what the game is about
3. Quit option for the user to end the game.

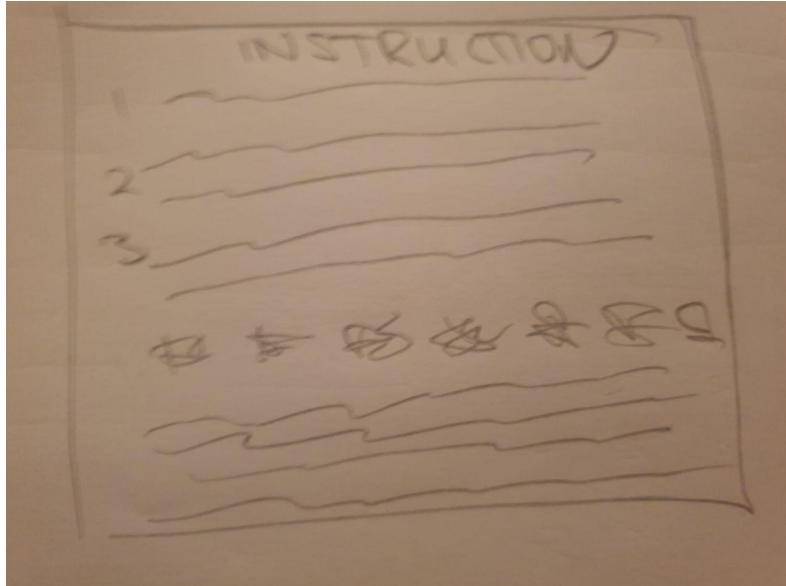
This sketch is a rough idea of what the menu should look like:



IN-GAME MENU

1. Our in game menu will consist of the instructions of the bomb defuse although its a in game menu it will not pause the game the timer down will keep decreasing as the user is looking on how to defuse the bomb therefore it counts as bomb diffusion.
2. User will use the Word "Instructions" to bring up the instructions menu
3. User will say "Game" to head back to the bomb

The below sketch is a rough idea of what the in-game menu should look like:



VOICE RECOGNITION

Everything will be controlled by the users Voice and phrases set by us in the code. to enable user to play the game.

Movement between scenes with voice recognition.

1. "Play Game" => To play the game
2. "Instructions" => Instructions
3. "Quit" => Quit the game
4. "Home" => to go back to the home page

In Game Mode

1. "Cut (colour of the wire) wire" => cuts wire
2. "Manual" => bomb manual (in game menu)
3. "Game" => from the manual back to game
4. "Enter (number)" => enters the digit to the code

IN CASE VOICE RECOGNITION IS UNAVAILABLE

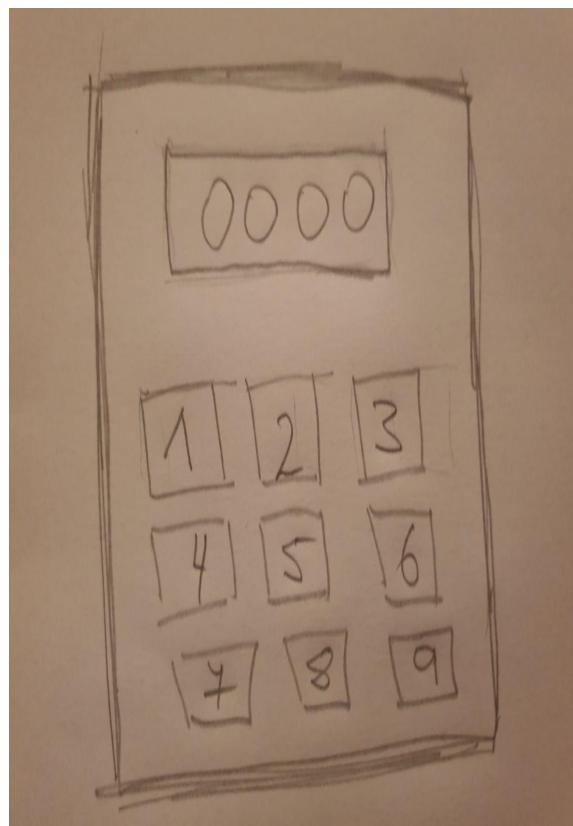
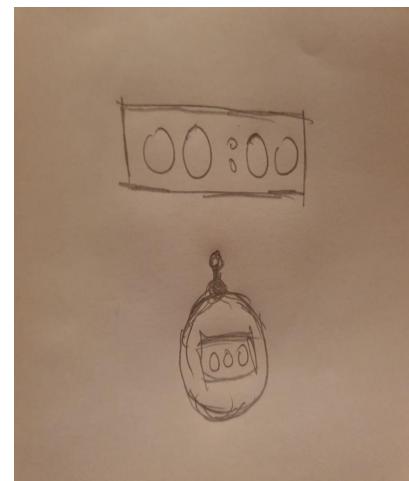
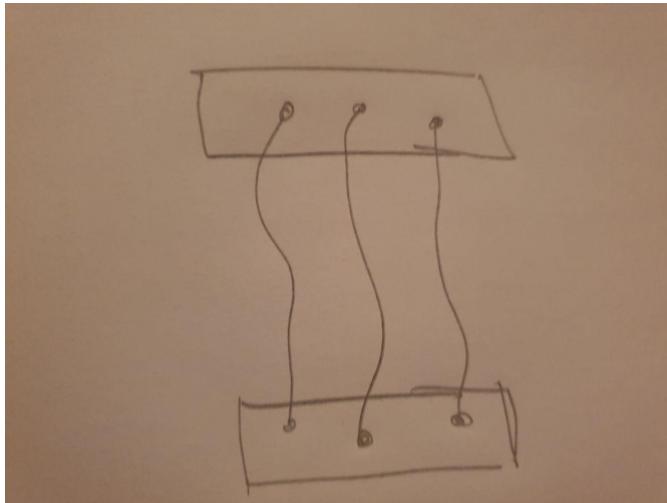
If voice recognition is not available on a persons device there will be buttons and keys that the user is able to press to complete the game without voice recognition.

1. Buttons to navigate in Home page
2. R => red wire
3. B => blue wire
4. y => yellow wire
5. W => white wire
6. keypad for the digits

THE GAMEPLAY

This game that we got the idea for this application from is “name of that game”. The challenge of the game should be in how the player overcomes the different parts of the bomb and to work against the time limit on the bomb, parts such as different wires (which to cut), code to decrypt for the right combination to defuse a bomb and come out safe at the end of the game. As the player progresses through these parts different parts come up visible such as cutting the right wire unlocks the code that user need to decipher.

Here are the design sketches of the different parts and extra sketches will be at the end of this document explaining how the game will operate.



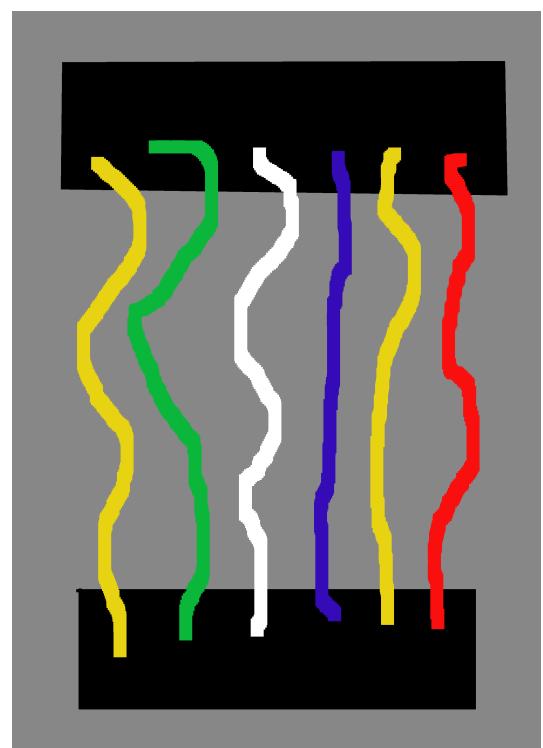
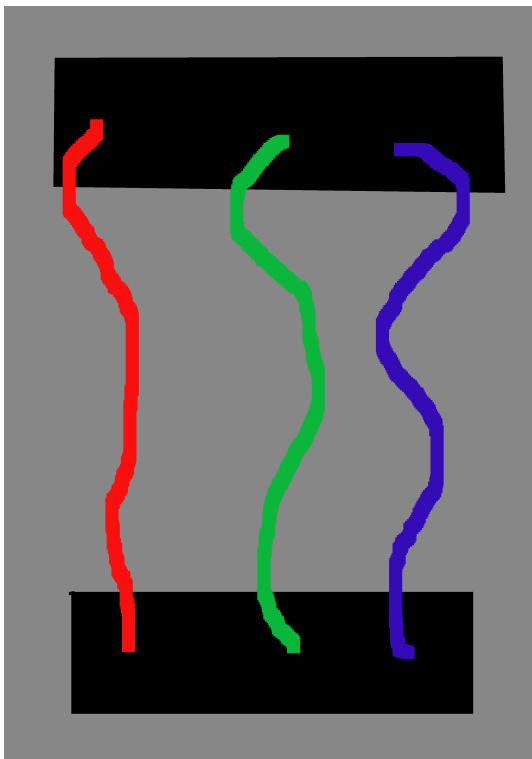
THE WIRES

The wires will be randomly generated for the user to decide which one to cut. To unlock the next stage and to defuse the rest of the bomb the right wire has to be cut .

These wires will be randomly generated in the groups of threes, fours, fives or sixes. The user HAS to follow the manual or randomly guess the wire and risk the chance of loosing half the time on the clock per wrong wire that's cut.

Wrong wire cut is the time = time - halfTheTime

Here are images of what the wires will look like in the game and more sketches at the end of the document.



THE ENCRYPTED CODE

The encrypted message is made up of four randomly generated characters. Each character is equal a number and four characters together displayed create a 4 digit code which is used to defuse the last part of the bomb.

Characters used

- "诶", "比", "西", "迪", "伊", "艾", "吉", "艾", "尺", "杰", "开", "勒", "马", "娜", "哦", "屁", "丝", "儿", "伊", "吾", "提", "吉", "吾", "豆", "贝", "尔", "维"

To figure out what number corresponds with each digit user has to use the annual and work out what they are. One wrong digit and its game over;



TEST CASES

1. All the controls should be responsive, the voice recognition phrases.
2. The code should be always hidden at the start of the game
3. Only the player cutting the right wire will display the code wrong wire will take away time .
4. When the game is started up the player should see the three options displayed to him the play, instructions and quit options.
5. when the player clicks on the play button it brings him to the game and the timer begins to decrease with the wires not cut and code not shown.
6. Instructions should be clear and easy to understand for the user they should go through each part of the bomb and cover all aspects.
7. if the time runs out or the wrong code is entered in the bomb should explode and display the end of the game to the user.
8. If the player completes the game and goes through all the parts of the bomb e.g. entering the right combination the player should be notified he won.
9. if the user asks for the in game menu it should be displayed to the user mid game .
10. If the player decides they had enough and presses the quit button or says quit the game should finish and it should quit.

DEVELOPER DIARY

STAGE 1

First thing we did was make a 3D template for the game and added some instructions to the readme. We got the application up and running in the labs. At this point we were planning on using the Myo armband for our project.

STAGE 2

We decided to change the project to 2D project for easy of development. We started doing some research on how the game should run and added some images for the wires

STAGE 3

This is where we got the basic mechanics of the game working. So far we had a pin pad working and a bomb scene made. We then added the wires to the main panel. Found out that merging the files was going to be a big issue for us. We were unable to use the myo armband because of the poor support on Linux devices. We both were doing working on Linux devices at the time. In the end we had to give back the armband and think of another way for the user to interact with the game without using a keyboard or mouse. We wanted the user to still be able to do hand movements.

STAGE 4

In this stage of development we made a homepage and updated the bomb scene. At this stage we encountered issues when merging our files on GitHub. We also made a win/lose page for the user and few more updates and buttons. We wanted to use the Wii controller for the game so the user would have a natural feel when pressing buttons or cutting wires. At the end of this stage we were unable to get our hands on a Wii controller due to the shops being closed for the pandemic.

STAGE 5

At this point we decided to finish the game without any gestures and worry about the issue later. There was no other way to get the game done without the use of hand gestures. We were thinking of using a Xbox Kinect but had no idea where get our hands on one.

STAGE 6

We had then decided to use voice controls for the game. We changed our basic Wii designed game over to something we can use with our keyboard or voice. Once changing everything around we found out that voice input does not work with Linux and would be easier for us to just find a old windows computer we can just use for testing at home. At this point the college and country was on a lock down.

STAGE 7

Got the game working with voice commands on windows computers and made a game manual. The manual explains how the game works with a keyboard and voice commands you need to do. There were some small errors to fix, which we got done. When game was completed we started updating the readme and design doc.

REFERENCES

MYO CONNECT REFERENCES

-
- <https://support.getmyo.com/hc/en-us/articles/203398347-Getting-started-with-your-Myo-armband>
 - <https://support.getmyo.com/hc/en-us/articles/360018409792-Myo-Connect-SDK-and-firmware-downloads>
 - <https://medium.com/thalmic/15-things-you-can-do-with-myo-f3f05bcc6f9c>
-

WII CONTROLLERS

-
- https://en.wikipedia.org/wiki/Wii_Remote
 - <https://www.instructables.com/id/How-to-Use-Your-Wii-Remote-on-Your-PC-As-a-Game-Co/>
 - <https://www.youtube.com/watch?v=co7xggFfE94>
-

SPEECH RECOGNITION

- <https://www.youtube.com/watch?v=29vyEOgsW8s>
 - <https://lightbuzz.com/speech-recognition-unity/>
 - <https://docs.microsoft.com/en-us/windows/mixed-reality/voice-input-in-unity>
-

ADDITIONAL REFERENCES USED

- <https://keertalkinggame.com/>
- <https://www.kongregate.com/games/bares94/bomb-defuse>
- <https://www.theverge.com/2019/7/25/20708711/keep-talking-and-nobody-explodes-bomb-defusal-cooperative-game-vr-mobile-launch-date>
- <https://docs.unity3d.com/Manual/system-requirements.html>
- <https://forum.unity.com/threads/definitive-unity-hardware-specs.471088/>
- <https://www.youtube.com/watch?v=3VLoGSVORjY>

Rules

On the Subject of Wires

Wires are the lifeblood of electronics! Wait, no, electricity is the lifeblood.

Wires are more like the arteries. The veins? No matter...

A wire module can have 3-6 wires on it.

Only the one correct wire needs to be cut to disarm themodule.

Wire ordering begins with the first on the top.

3 wires:

- If there are no red wires, cut the second wire.
- Otherwise, if the last wire is white, cut the last wire.
- Otherwise, if there is more than one blue wire, cut the last blue wire.
- Otherwise, cut the last wire.

4 wires:

- If there is more than one red wire and the last digit of the serial number is odd, cut the last red wire.
- Otherwise, if the last wire is yellow and there are no red wires, cut the first wire.
- Otherwise, if there is exactly one blue wire, cut the first wire.
- Otherwise, if there is more than one yellow wire, cut the last wire.
- Otherwise, cut the second wire.

5 wires:

- If the last wire is black and the last digit of the serial number is odd, cut the fourth wire.
- Otherwise, if there is exactly one red wire and there is more than one yellow wire, cut the first wire.
- Otherwise, if there are no black wires, cut the second wire.
- Otherwise, cut the first wire.

6 wires:

- If there are no yellow wires and the last digit of the serial number is odd, cut the third wire.
- Otherwise, if there is exactly one yellow wire and there is more than one white wire, cut the fourth wire.
- Otherwise, if there are no red wires, cut the last wire.
- Otherwise, cut the fourth wire.

Symbols - Solve these equations and figure out the code

The symbols and their equations are highlighted ion the same colour

"杰""伊","诶"

"迪","马","吾"

"艾","丝","维"

87367-87366

$10a-9=2a+31$

$4=a-5$

"比","尺","儿"

"伊","娜","豆"

$2c-3=1$

$4n+3=2n+15$

"比","开","吾"

"艾","哦","贝"

$3m+2=11$

$9m+3=2m+52$

"西","勒","提"

"吉","屁","尔"

$2c+3=11$

$5x-22=2x+2$

