

Computer Games Development CW208

Software Functional Specification

Year IV

Sasa Kuzmanovic

C00249246

|  |  |
| --- | --- |
|  |  |
| [14/04/2023] | |

[Declaration form to be attached]

Contents

[Acknowledgements 3](#_Toc133441897)

[Purpose 3](#_Toc133441898)

[Functional Specification 3](#_Toc133441899)

[User requirements 3](#_Toc133441900)

[Hardware requirements 3](#_Toc133441901)

[Software requirements 3](#_Toc133441902)

[Communication requirements 3](#_Toc133441903)

[Major functionalities 4](#_Toc133441904)

[1. Authenticating the viewer: 4](#_Toc133441905)

[2. Establishing a connection with the host 5](#_Toc133441906)

[3. Capturing joystick input 5](#_Toc133441907)

[4. Accepting the packet and converting it to input 5](#_Toc133441908)

[Navigation Sequence 6](#_Toc133441909)

[Host: 6](#_Toc133441910)

[Client: 6](#_Toc133441911)

[References 6](#_Toc133441912)

# Acknowledgements

I would like to thank the following people who assisted in completing this project including:

Dr. Martin Harrigan who kindly provided guidelines on how my project should be progressing and what should my next step be. Being supportive in my decisions throughout the project and help me achieve my goals.

I would also like to thank my fellow class mates for testing my software and providing input on what should be improved to help increase the quality and user experience in this project.

I would also like to thank AwesomeBlade for allowing me to user their game (8BitBoy) for testing my software.

# Purpose

This project will allow players all around the world share their gameplay with their friends and share controls of a game using Discord. It is supposed to provide a controlled way of watching a stream on discord and overtaking someone’s controls.

# Functional Specification

## User requirements

Creating a simple UI that can be used by any one will be key in the software becoming widespread and increasing its usability. Instructions that are clear and legible should be present to explain how the software works.

## Hardware requirements

Both parties participating in using the software will need a computer that is able to stream video footage in real time to discord. Viewer should have a controller that can be connected to a computer.

## Software requirements

Both parties will need to have a copy of the program, they should have discord desktop application installed, Windows operating system and a single copy of the game on the Hosts side.

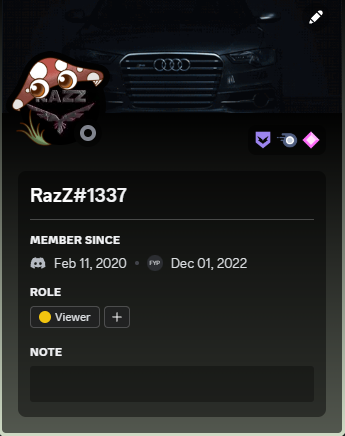
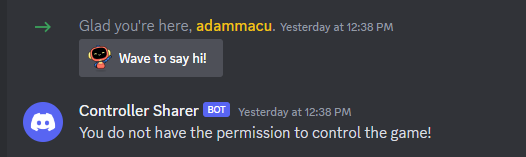
## Communication requirements

Stable internet connection is going to be key in having a good experience. Video streaming heavily relies on a stable internet connection, while sharing inputs requires very low amounts of networking to where it is nearly not noticeable. Having a specific role on discord is required for sharing controls.

# Major functionalities

## 1. Authenticating the viewer:

The program is supposed to provide a way of checking does the person have the required role on discord. This is going to decide if the person can establish a connection with the host to start sending over input.



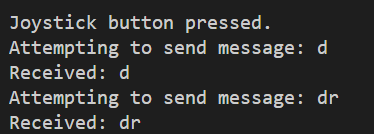
## 2. Establishing a connection with the host

Establishing a TCP connection with the Host will be necessary to transfer input from the clients joystick. This will happen using sockets to dictate which port should the Host be listening on for input from the client. Joystick input will be sent as a string that needs to be decoded on the host side. When the discord bot reads and checks the role on discord it will allow a TCP connection to be established.



## 3. Capturing joystick input

This is one of the most important aspects of the project. Reading the controller that is connected to the computer will be key to enable sending inputs. When each button is pressed, this is captured and converted into a string that will be sent over the connection to the host. Each button on the joystick corresponds to a button or a combination of buttons on the keyboard. Encoding it into utf-8 would be necessary. Clicking the button will send one packet that will say the button is being pressed. Releasing the button instructs that the button has been released by sending another packet.



## 4. Accepting the packet and converting it to input

When a button is pressed on the clients side of the program, it tries to send a packet to the host. Host is accepting the connection and stores the latest control until it has been converted into input. The host decodes the string that was sent and checks it against a set of binds previously set up. When a match is found a control is pressed. The same is with releasing the button, when a match is found the control is released, which would result in something happening in the game.



# Navigation Sequence

## Host:

When the software is first launched, the user needs to select a button preset they will use for their game. When the user selected the preset, they have to select if they are a streamer or a viewer. When the option streamer is selected you have nothing else to do besides sharing your screen on discord and assigning the person the role they are required to have to start the connection. Launching the game is the last step. To turn off the program, simply open the program window and the program will exit, disconnecting the host and the client.

## Client:

When the client selects the option to be a viewer, they will have to type Hello into the discord chat, to be authorised by the Discord bot. When the player is authorised to establish the connection, they will connect to the Host. The client can now connect their controller and join the stream that is being shared by the Host. When the client is watching the stream they can start playing their game. To turn off the program, simply open the program window and the program will exit, disconnecting the host and the client.

# References

**Report**

[1] Brown M.(2018, April 24). How to use Xbox One Mixer controller sharing [Online]

(URL: <https://www.windowscentral.com/how-use-xbox-one-mixer-controller-sharing>) (Accessed 2022 December 2)

[2] Shinners P.(2000, January 1). Pygame documentation [Online]

(URL: <https://www.pygame.org/docs/>) (Accessed 2022 December 14)

**Website**

[3] ChaosTricks, (2021, July 11). Chaos Tricks [Online]

(URL: <https://chaostricks.com>) (Accessed 2022 December 2)

[4] Danny (2015 July 7). Discord.py API Reference [Online]

(URL: <https://discordpy.readthedocs.io/en/stable/api.html>) (Accessed 2022 December 2)

[5] Unknown (2022 January 1). Pynput Package Documentation [Online]

(URL: <https://pynput.readthedocs.io/en/latest/>) (Accessed 2023 January 10)

[6] Unknown (2011 May 2) socket – Low level networking interface [Online]

(URL: <https://docs.python.org/3/library/socket.html>) (Accessed 2023 April 10)