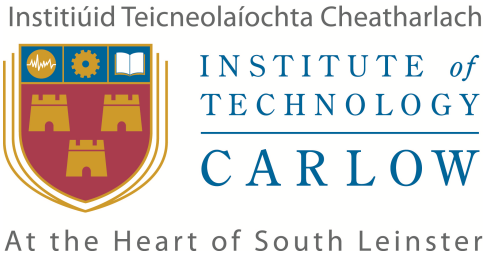


Computer Games Development CW208
Technical Design Document
Year IV

Allexis Alvarico
C00242855
28/04/2022

Content

Technical Design

The purpose of this document is to communicate effectively the technical details and design decisions of the system/algorithm to the readers.

It could include software architecture, algorithm design, class specifications, pseudo code, etc. with tools such as UML, Class Diagram, and CRC Cards.

List of Features

2.1 Receiving live camera feed

This feature allows unity to stream in the game by passing the IP address into an input field and streams the live camera onto the screen if it has a successful connection.

This is the processing of the stream

```
public void ParseStream(Uri uri)
   {
        WebRequest = URI
        Get Response from the website
   }

//Checks if the website is responding.
private void OnGetResponse(IAsyncResult asyncResult)
{
        Wait to confirm that the website is working
}

//If the website is streaming the live feed
 loop if the stream is working
        {
                Check if the frame is not -1
                Get the frames and store them
                Check if it's not the last frame
                break from it

        }
```

if the stream or response is down, break away from it.

This is the setup of the texture video into the object and updates the frame of the live camera feed.

```
void update()
increase the deltatime
      Update the frame
      {
         assign the texture with the new frame from the stream.
         assign the material to the object
         set deltaTime back to zero
      }
```

2.2 Object/Facial Recognition

This is the feature where the raspberry pi's drone AI recognizes objects or facials when moving around the area. The library that I'll be using is OpenCV since there are tutorials on how to make it work with a webcam and train the AI to recognize these objects or facials.

2.3 Unity and Raspberry Pi communication.

Unity connects to the raspberry by wifi and Raspberry Pi drone waits for unity to send data to it and process the command. I'm also using unity's system.io and network stream class for retrieving and sending data.

This sets up the communication between unity and the raspberry pi

```
void setupSocket()
         get the IP address and port of the raspberry pi
         Connect to the IP endpoint
         set the sockets for a writer and reader
         if all goes to plan socket is ready
         if not throw an exception.
```

This sends data from unity to the raspberry pi

```
void writeSocket(string str)
         check if there is a connection
         if not return

         send the str
```

```
string readSocket()
         check if there is a connection
         if not return nothing

         check if there's data available
                  return socket reader;
```

2.4 VR camera movement

This feature is to allow the user to use VR to move the camera and to achieve the goal of telepresence. I want to use the ZED Package since it has the necessary code for head tracking with the VR

2.5 Controller/Commands

This feature allows unity to send commands to the drone. To send a command to the drone is to create a game object and assign it to the raspberry connection script and use the get component function.

E.g

```
void toggle()
string command = “on”;
TCPListener.GetComponent<RaspberryCon>().writeSocket(command);
```

once the game object writes it, the command will be on the raspberry side.
The data will be sent to a python script and be read by the raspberry and take the command and execute.

```
def execute command(self,cmd):
        if str(cmd[2] == ‘on’:
                do action here and so forth.
```

This is how unity and the raspberry communicate with each other. To add a new command is to just give a command string and on the write socket and go to the function execute commands and place an if statement of the string if its equals to it then give its command if data match below.

Software tools

3.1 Unity

I chose Unity for my main interface for the user and drone communication because I wanted to make a small AR/mix reality for the project. The other reason is that I can use the plugins for VR since it's already in the engine.

3.2 OpenCV

This library will allow me to use facial/object recognition AI for my RC drone to identify the surrounding area.

Hardware tools

4.1 VR HTC Vive

The use of the VR HTC Vive is for the telepresence aspect of the project.

4.2 RC Drone

The RC Drone from sunFounder allows me to move the RC Drone.

4.3 Raspberry Pi

The Raspberry Pi handles the drone’s movement, the camera live feed, and the server aspect of the project.

CRC Cards

Client
Responsibility - Handles and sends data to the raspberry pi
Collaborator - SocketConnection scene - Server

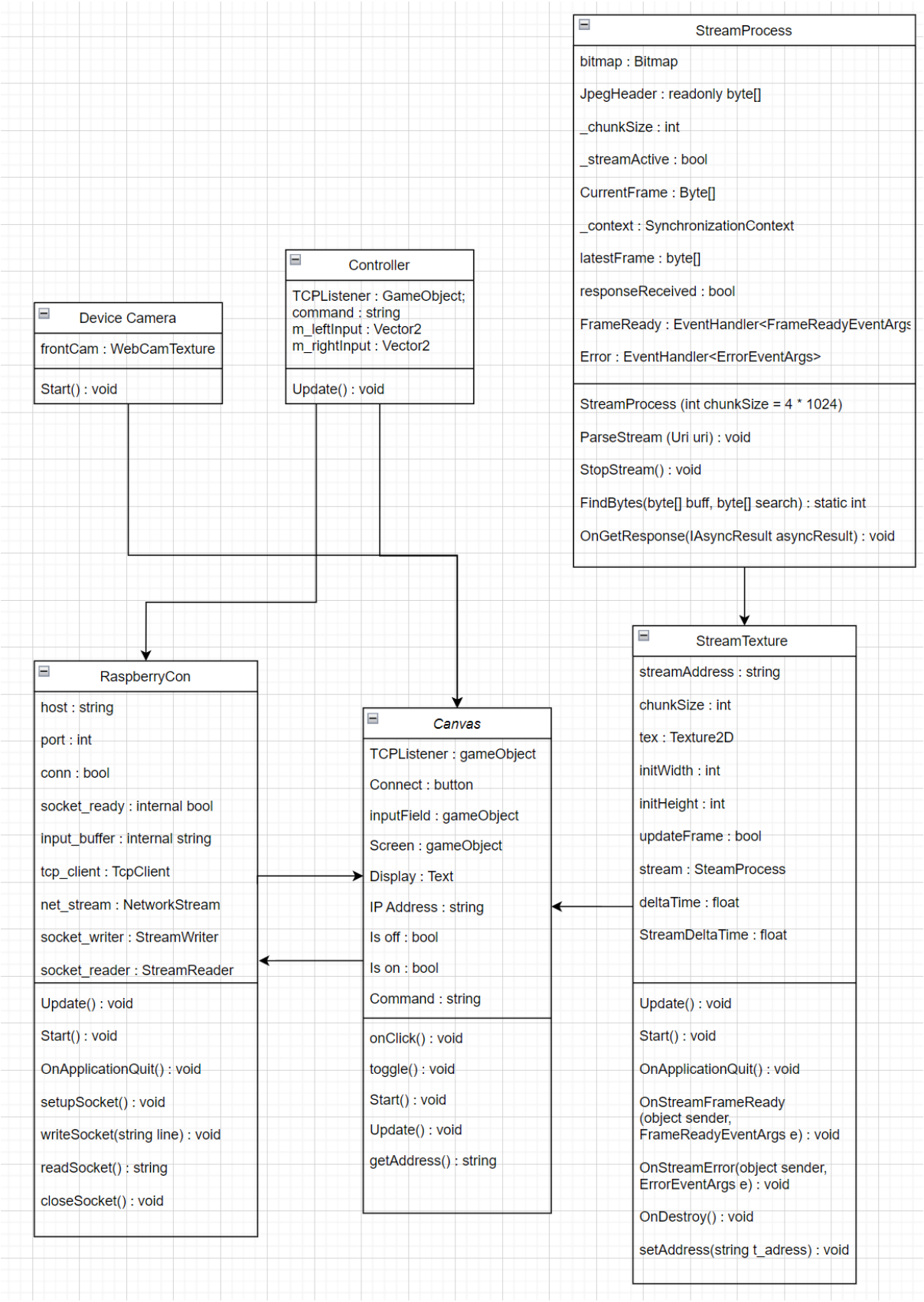
Server
Responsibility - Retrieval of data from the client and reads it.
Collaborator - SocketConnection scene - Client

Controller/Command
Responsibility - Handles Commands from the unity and sends commands to the drone
Collaborator - SocketConnection scene - Drone RC - Camera of the Drone

Streaming Texture
Responsibility - Handles the live feed from the drone - Retrieving the IP address of the live feed and assigning it to a game object.
Collaborator - SocketConnection scene - Raspberry Pi

Streaming Processing
Responsibility - Handles the camera feed and processes the frames.
Collaborator - SocketConnection scene - Drone RC - Camera of the Drone

Class diagram



Acknowledgments:

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

Oisin Cawley for being my supervisor and giving me a drone to work with.

Lei Shi for giving the necessary information for what needs to be done for the project.

Misperry on how to make a camera stream web for the raspberry pi.

References

Author(s) - family name, initials. (Year). *Title of book.* Edition. Place of publication: Publisher.

Book

Paulsen, Kris (2017). Here/There: Telepresence, Touch, and Art at the Interface. The MIT Press

Report

Jin, ZheKai and Shao, YiFei. 2017 Fall. Telepresence Realization with Virtual Reality. Github.

Web-site

Zara, Moheeb (25 MAR 2020), Building a Raspberry Pi telepresence robot using serverless: Part 1. Link. (Accessed 10 October 2021)