

Institiúid Teicneolaíochta Cheatharlach



At the Heart of South Leinster

# Computer Games Development CW208

## Software Functional Specification

### Year IV

Allexis Alvarico

C00242855

28/04/2022

# Contents:

1. Introduction	2
1.1 Purpose	2
1.2 Intended Audience and Reading Suggestions	2
1.3 Project Scope	2
1.4 Document Conventions	2
2. Overall Description	
2.1 Product Perspective	3
2.2 Product Features	3
2.3 User Classes and Characteristics	4
2.4 Operating Environment	4
2.5 Design and Implementation Constraints	4
3. System Features	
3.1 Functional Requirements	5
4. External Interface Requirements	
4.1 User Interfaces	6
4.2 Hardware Interfaces	6
4.3 Software Interface	6
4.4 Communications Interfaces	6
5. Non-functional Requirements	
5.1 Performance Requirements	7
5.2 Safety Requirements	7
5.3 Security Requirements	7
5.4 Software Quality Requirements	7
6. Acknowledgments	8
7. References	8

# Introduction

## 1.1 Purpose

The purpose of this document is to build a telepresence remote-controlled drone with a raspberry pi that has facial/object recognition. The drone provides the user to explore the surrounding area and gather information from another physical place.

## 1.2 Intended Audience and Reading Suggestions

This project is a prototype for a telepresence RC drone and it is restricted within the college premises. This has been implemented under the guidance of a college professor. This project is useful for security/military and as well as for civilians.

## 1.3 Project Scope

The purpose of the telepresence RC drone is that it allows the user to be in a safe remote location and be able to send a drone into a small hole or a dangerous place for a human to navigate or gather information. The Drone streams the camera to the user to allow visual confirmation of object/facial recognition of the area.

## 1.4 Document Conventions

This document uses the following conventions below.

Acronym	
ERM	Entity-Relationship Model
VR	Virtual Reality
RC	Remote Controlled
SSH	Secure Shell

# Overall Description

## 2.1 Product Perspective

The Telepresence RC drone systems:

### VR Camera movement:

This allows the user to use a VR headset to view the surrounding area from the drone and safely gather information from a remote physical place.

### Object/Facial recognition:

This allows the user to have information displayed on the screen to if the drone identifies an object or facial with the view of the camera to gather information.

### Software and hardware communication:

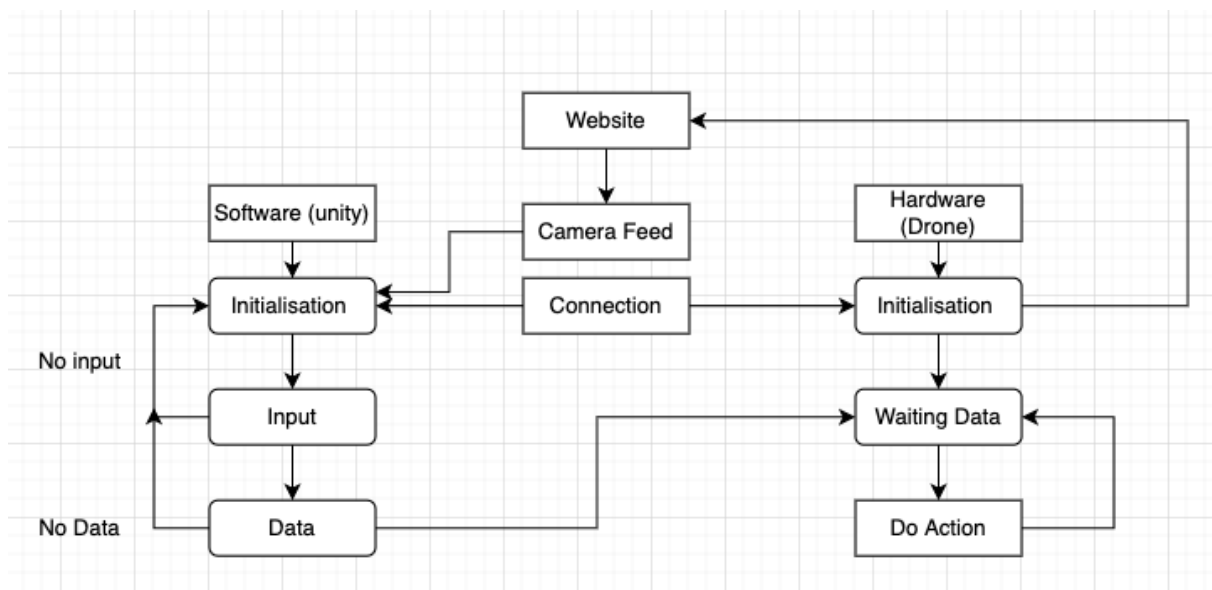
This allows the software (desktop) to communicate with the drone (hardware) by sending commands to the drone via the internet from the desktop allowing the drone to move or give information to the user.

### Streaming:

The drone includes the ability to stream the camera to the internet for the user to receive the live feed from the camera to be viewed from a safe remote place.

## 2.2 Product Feature

The major features of the drone and its systems is shown below ERM.



## **2.3 User Classes and Characteristics**

The systems that are involved in this project are the client system and the server system. The two systems connect to each other from the internet, once connected via the internet the systems can feed each other with information e.g live camera feed and movement commands for the drone to move.

The client system should be able to do the following functions:

- Connect to the server.
- Get a live camera feed from a website.
- Sending and receiving data.
- Commanding the server via controller
- Object/Facial recognition view.

The server system should have the following functions:

- Waiting for connection from the client.
- Stream a live feed and create an HTTP site.
- Sending the camera feed, back to the client.
- Receiving commands from the client.
- Moving the RC drone.

## **2.4 Operating Environment**

The operating environment for the telepresence VR is listed below:

- Client/server system.
- Operating system: macOS, Windows, and Linux(Raspberry pi).
- HTTP
- Platform: python, C#, and C.

## **2.5 Design and Implementation Constraints:**

1. SSH connection to the Raspberry Pi.
2. Live feed format.
3. Data lost due to distance.
4. Wifi connection.

# System Features

## 3.1 Functional Requirements

system features include:

- Client and Server system:

The connection between the client and the server must be strong and continuous. If the client has lost its connection to the server, the drone can be lost since the user is from another location and the drone is far from retrieval. This function is a high priority because without this communication between the user and the drone there's no way for the drone to receive commands and the user can't interact with the drone.



- Live streaming and creation of a website:

The drone is able to stream its camera for the user to see what's in front of the drone to be able to change course if there is an obstacle. Once the camera is online, the Raspberry Pi will create a site for it to stream and be retrieved by the user's software. This function is also a high priority since if the user can't receive visual information from the drone, the user is blind to control it.

- Command Data:

This function allows the user to give commands to the RC drone to move or control the camera view direction. This allows the telepresence to work since the drone is in a different location and the user is at their desktop. This is a medium priority since it only sends data from the client to the server.

- Facial/Object recognition:

The facial/object recognition function allows the drone to identify the facial/object from the camera feed. This allows the user to gather information from the surroundings of the drone. The priority of this function is low-medium.

# External Interface Requirements

## 4.1 User Interfaces

- Unity for GUI
- Terminal/command line/anaconda

## 4.2 Hardware Interfaces

The program (Access) uses the hard disk. Access to the hard drive and other hardware is managed by the operating system and Access.

## 4.3 Software Interface

The following is the software used for the Telepresence VR drone.

Software	Description
Unity	I have chosen unity to use its VR plugin and used for the connection between the user and drone.
Operation System	Picked macOS and Linux due to their similarity to the terminal.

## 4.4 Communications Interfaces

The communication interface used for this project is an internet connection for the software and the drone to communicate e.g Allows the drone to receive the data from the software.

# Non-functional Requirements

## 5.1 Performance Requirement

The steps involved to work on the project pc specification listed below.

Minimum requirements	Windows	MacOS
Operating system version	Windows 10 or newer	High Sierra 10.13+ (Intel editor)
CPU	X64 architecture with SSE2 instruction set support.	X64 architecture with SSE2 instruction set supports Apple M1 or above.
Graphics API	DX10 or DX11	Metal-capable Intel and AMD GPUs

## 5.2 Safety Requirements

The safety requirements will be making sure the software and hardware (drone) is set up before connecting to each other. If not done correctly, the pc/drone it will take long for it to realise that the other is not on causing it to freeze for a bit.

## 5.3 Security Requirements

The security system shall be designed with a level of security appropriate for the IP address. The IP must be protected from any malicious attack/hack.

## 5.4 Software Quality Requirements

Receiving/Sending Data: The software is able to send data to the drone and can receive the camera feed.

maintainability: The software and drone must maintain connection in order to be used.



## 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)