# ICVGoggles: Wearable Adjustable Simulations of Impaired Colour Vision

Babak Momen
AC40001 Honours Project
BSc (Hons) Applied Computing
University of Dundee, 2016
Supervisor: Dr. David Flatla

## ABSTRACT

Colour is used to tie in closely with a specific meaning, signal or message; however it is mostly used for aesthetics. People with Impaired Colour Vision (ICV) come across challenges every day to distinguish between colours, designers must consider users with ICV since information can be misinterpreted or even missed. Current software and hardware solutions provide real time simulations of various spectrums of impaired colour vision. IVCGoggles goes above and beyond current technologies to provide designers with adjustable simulations viewed with a hands-free Oculus Rift headset.

Two user studies took place to evaluate how useful the system can be for designers. ICVGoggles gave participants more insight into colour vision deficiencies as well as aiding users find poor colour choices in their own work. The system received positive feedback and has applications in areas such as teaching and within design practices.

## 1. INTRODUCTION

### 1.1 Background

Most cases of Impaired Colour Vision (ICV) are hereditary meaning it is passed on genetically from parent to offspring. They can also occasionally be acquired as a result of certain eye diseases. Failing to discriminate between red and green is the most common form of ICV (Protanopia / Deuteranopia) and the gene is X- linked recessive which explains the prevalence difference between genders (8% in males and 0.5% in females). Blue-yellow ICV or Tritanopia is rare and tritanomalous symptoms are more commonly acquired from environmental factors such as age, where the eye lens becomes increasingly yellow over time, cataracts or trauma to the front or the back of the head. Monochromacy is even rarer and affects two or more types of cone in the eye. Colour vision can be said to be an illusion created by the interactions of billions of neurons in our brain[1], we do not all perceive colours the same way. There is an information gap for designers about ICV, many do not know the problems some users will suffer because of their colour choices. Impaired colour vision varies in severity and because of this, it affects people differently; sometimes it is unnoticeable. Figure 1 was created using an online tool and demonstrates different types of ICV, however just like most other tools available, these images show the maximum severity only.

**Figure 1 - Left-to-right, top-to-bottom: Normal colour vision, Protanopia, Deuteranopia, Tritanopia.**

Designers do not normally consider ICV in their design practice; they are a small demographic of users. Without doing so, designers run the risk of creating a colour palette which will be confusing to some users. Small measures such as colour checking and ICV simulations can vastly improve the users' experience [1].

### 1.2 The Problem

The process of simulating ICV has been around for a while and there is no shortage of applications and tools which can do so. What many of these tools cannot do is simulate the surroundings of the user in real-time, this is the problem ICVGoggles aims to solve. Adjusting the severity of the type of ICV is a feature which is not present in any simulation tool currently available. ICVGoggles aims to give users a method to actively change the severity of ICV during the simulation and without any noticeable delay.

Currently most ICV simulation tools provide a solution to specific problems and have downfalls when applied in other areas. For example, Vischeck provides developers with a way to see their work through the eyes of an ICV user, however it does not take external lighting or screen brightness into consideration. Other solutions such as

1 P.Gouras, 'Colour Vision', in webvision.med.utah.edu, last update 1 July, 2009

mobile applications[2] are restricted by the hardware they are on; users must always use one hand to move the mobile device around.

## 1.3 Motivation

It is becoming more commonplace for applications to have colour blind settings, however there are very few which display ICV simulations in real time. ICVGoggles aims to take simulating ICV to the real world and will use an Oculus Rift headset to do so. This will result in a simulation tool which will give the user both hands to operate the world around them. The tool could be used for designing both digital and physical products, or it could be used to educate and teach. ICVGoggles will be open to many different routes when it is complete.

## 1.4 Available Products

When designing for users, it can be useful to view content through the eyes of ICV users. This can reveal poor colour choices which can lead to confusion and even missed information. There are a plethora of applications available on many different platforms which can detect and manipulate pixels to simulate ICV. One flaw most software applications present when simulating ICV is the exclusion of environmental factors such as room brightness.

### *1.4.1 Spectrum for Chrome*

It is possible to obtain a browser add-on for Google Chrome called "Spectrum" which simulates ICV for the current web page [2]. This extension enables the user to select a type of ICV and see what the webpage looks like; it is especially helpful when looking at data visualisations where colours could be misinterpreted. Spectrum does not allow the user to adjust the severity of the condition specified and so it cannot simulate weak/mild ICV. This add-on is also limited to web pages only.

### *1.4.2 Vischeck*

Vischeck [3] provides different toolsets for simulating ICV, some are available to use online whilst others require downloading and installing. The web version of Vischeck lets users upload images and see them through the eyes of an ICV user. There is also a Photoshop plugin which enables ICV simulations within the application. Vischeck does not provide a real-time ICV simulation tool however.

[2] 20 iPhone Apps for the Colour Blind - http://www.color-blindness.com/2010/12/13/20-iphone-apps-for-the-color-blind/

### *1.4.3 Colourblind Vision*

Colourblind Vision [4] is an application developed for Android smartphones. It was created by Bradley C. Grimm and provides real-time ICV simulations using the smartphones back camera. The application also enables users to simulate ICV on images found on the camera reel. There are similar applications available for iPhone users, showing there is an interest in simulating ICV on handheld devices.

### *1.4.4 Chromatic Vision Simulation*

This application is available on the iTunes store and android market and can simulate various types of ICV in real-time. It bares similar functionality to what ICVGoggles aims for, however Chromatic Vision Simulation [5] does not allow the user to adjust the severity. One user review on the store was from a parent who used the application to shop for their colour blind child. Being on a small device makes it easier for this simulation tool to be used in public.

### *1.4.5 Other Applications*

The market for ICV simulations tools exists and there are many applications available. Chromatic Vision Simulator provides similar functionality to what ICVGoggles will, however this is not the first application to do real-time simulations. ColourBlindness SimulateCorrect is also available on mobile devices and was created by Seewald Solutions; it uses the back camera and then simulates ICV. All products mentioned are able to simulate some form of ICV, however none of them are able to adjust the severity of the type of ICV selecting. This lack of functionality could confuse users into thinking there are very few ways ICV can affect someone, limiting their knowledge on the subject.

## 1.5 Social Context

It can be difficult to understand how ICV affects people; emulating the effects on images is useful, however this confines the condition within just the images. ICVGoggles aims to provide a simulation tool which works in real time giving users a peek at ICV 'in the wild'. Once complete, ICVGoggles can be applied to different social contexts in order to raise awareness and to educate users. For example, the tool could be used on trainee teachers to educate them in how ICV affects people. This sort of training could eventually help teachers to recognise comments made by children who have ICV; the earlier ICV is diagnosed, the better. Another application could be the use of the tool on parents who have children with ICV to give them a rich learning experience.

## 1.6 Scientific Context

By providing an interactive tool which simulates ICV in real time, this project can be used to inform and educate. Teaching ICV can be a difficult task depending on who is learning; it may be more effective to provide a hands-on tool which can show how it affects people. The severity value can be adjusted by the user; all current simulation techniques show the most severe forms of ICV only instead of incrementing or decrementing the severity. Because of this, ICVGoggles is a powerful tool to raise awareness for ICV.

## 1.7 Solution

ICVGoggles is a hands-free headset which can provide adjustable simulations in real time. This solves the current problems which exist with current tools, it will be hands-free and the only adjustable tool around. With users being exposed to a greater variety of types and severities of ICV, they can gain new knowledge which may improve their design practice.

The project timespan is around seven months and so managing time effectively was key to complete all components. User studies were ran during and after the development of the system before the individual videos were transcribed separately.

## 1.8 Results & Evaluation

The transcriptions for the user studies were compared and contrasted in order to evaluate ICVGoggles. Nine participants out of ten said their understanding of ICV had increased through the use of the system. Four participants brought in their own work and two found colours which could not be distinguished. The user studies results could suggest ICVGoggles provides an engaging learning experience which can be applied in design practices to improve the usability of products. A deeper look into the results takes place in Section 6 – Evaluation.

## 1.9 Contributions

ICVGoggles is a project which has inherited work from previous research. The colour swapping in ICVGoggles is based off of CVDSimulation, a Processing application created by Dr. David Flatla at DAPRlab [6]. CVDSimulation provides the ground work for ICVGoggles and is itself based on the paper "A Physiologically-based Model for Simulation of Color Vision Deficiency" [7] and a tutorial[3]. The tutorial contains a large 4 dimensional array which contains float values. By applying the correct formula to red, green and blue pixels with these values Protanopia, Deuteranopia and Tritanopia ICV can be simulated. Similar research was published in 1997 called "Computerized simulation of colour appearance for dichromats" by Brettel et al [8]. An algorithm is proposed in the paper which transforms an image to simulate different forms of ICV.

The student chose the ECVD (Exploring Colour Vision Deficiencies) module and was taught by Dr. David Flatla. The module greatly aided as research papers were covered on a weekly basis, reinforcing the students' knowledge of ICV. One paper by Dr. David Flatla and Carl Gutwin [9] was particularly useful as it gives a very understandable breakdown of ICV and how it can affect someone. The research in the paper also looked at personalised simulations. It was necessary for the student to do plenty of background reading to ensure all work done was accurate.

CVDSimulation gave ICVGoggles a basic framework to build upon. The application lets the user select a webcam and then begins streaming. Protanopia, Deuteranopia and Tritanopia are the three types of ICV which can be freely switched between. The severity can also be adjusted, giving the users a sense of how it can affect people differently. CVDSimulation is available to use inside the ICVGoggles GitHub repository.

# 2. SPECIFICATION

## 2.1 Feature List

When planning ICVGoggles, the student and supervisor discussed technical features the software would have. Ideas were bounced back and forth; it was decided to make sure the system was extensible for the future. With that in mind, the student programmed from the ground up and made sure more could be added at any time.

### *2.1.1 Display Camera Feed in Real Time*

The first feature to initiate the project was to display the OVRVision cameras on the Oculus Rift. This task would set the foundation for ICVGoggles's code. Once the surroundings are within the headset, the next task would be to start doing image manipulation.

[3] Tutorial where the CVDSimulation colour value matrices can be found. http://www.inf.ufrgs.br/~oliveira/pubs_files/CVD_Simulation/CVD_Simulation.html

### 2.1.2 Monochromacy Simulations

After displaying the camera feed, the next task involves simulating Monochromacy. This type of ICV was chosen first as it is relatively simple to program in OpenFrameworks. Since the framework gives access to RGB values in pixels, simulating Monochromacy is done by just invalidating two colours from red, green or blue. Each frame in the camera feed is treated as an image and the pixel adjustments are made to every frame taken in. After doing research into simulating Monochromacy, the student and supervisor came across the specific values in a blog post "Converting to Grayscale" [15] by John D. Cook. By utilising the conversion method from this source, the monochromacy simulation would be more accurate than simply invalidating two colours.

### 2.1.3 Protan, Deutan and Tritan Simulations

By referring to CVDSimulation, a Processing application developed by Dr. David Flatla, the student would go on to create simulations for different types of ICV. The colour value swap matrices mentioned in *Section 1.9 Contributions* would be the basis of how to interact with the system. The float array takes in two values from the user, type and severity. By adjusting these values in real-time, the user can actively increment and decrease severity and switch types.

### 2.1.4 Control Scheme

With the Oculus Rift on the users face, it can be quite difficult to navigate and press keys on a keyboard. The student took this into consideration when creating the controls for ICVGoggles (See Figure 2). Switching between types uses the numbers 1 - 4 on the keyboard; when deployed in the lab, sticky tack was applied to the keys to give them extra tactility. Pressing Spacebar toggles the heads up display as it is an easy to locate key with a large surface area. Finally, left and right click on the mouse controls the severity value. The severity can range from 0 - 10 and each increment affects the severity of the current ICV displayed.

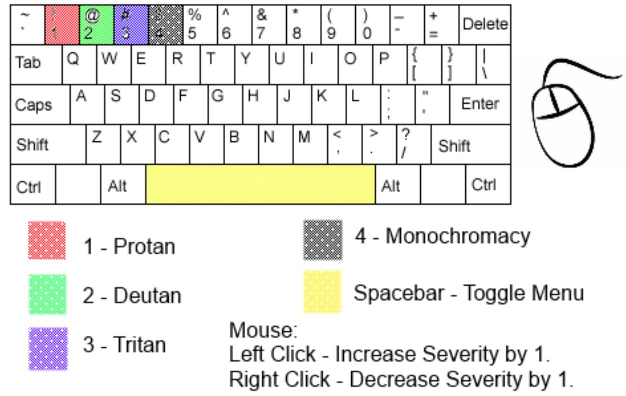


**Figure 2 - Proposed Control Scheme for ICVGoggles**

### 2.1.5 Heads up Display (HUD)

After the simulations and control scheme are completed, the next stage would be to create a HUD to display information to the user. The basic HUD for the project deadline contains information about the current settings and keyboard controls; future versions will become more complex when doing personalised simulations.

## 2.2 Project Plan

This projects lifecycle spanned over seven months and so it was necessary for the student to correctly plan out their course of action. The weekly meetings set up with the supervisor ensured progress could be monitored at regular intervals.

### 2.2.1 Work Schedule

The project began on the 5th October 2015 with a meeting with Dr. David Flatla. It was decided that the student should have weekly meetings with their supervisor to ensure all progress was being monitored. The weekly meetings also meant the student could assign smaller deadlines on a weekly basis. Notes were taken at every session and have been compiled into Appendix A. These minute meetings contain thoughts gathered with the supervisor and were used to plan work and to be reflected upon for the following week. A GitHub repository was set up at the start to store all work done whilst the commits and punch card feature proved useful to track progress. For the mid-term progress hand in, the student produced a Gantt chart which would serve as a rough schedule. The mid-term progress report can be found in Appendix J

### 2.2.2 Deliverables

The student created deliverables in the Gantt chart seen in Appendix B. These were spread out over the course of the project lifecycle and extra time was given for all tasks to give the student some leeway if any complications occurred. Initially the deliverables were centred on the ethics submission which required multiple consent documents to be created and checked by the Ethics Committee. These documents included consent forms for the pre-exposure interviews and second stage evaluations, both for ICV and Non-ICV participants.

After these were completed, the student then moved onto producing the code for the project. As seen in the Gantt chart, this production was done over the course of several months whilst both user studies were taking place. The plan for the programming was created before any coding had begun, therefore the student made sure the tasks were split into manageable, realistic deadlines. The progression of coding is also seen on the chart and each deliverable is a logical increment towards the final goal.

### *2.2.3 Resources*

The student took advantage of resources available. Labs within the Queen Mother Building at Dundee University were used for both the pre-exposure interviews and the second stage evaluations. Within the labs is a PC which contains the Oculus Runtime and this enables the use of ICVGoggles.

Whilst studying in their final semester, the Student chose the ECVD Research Frontiers module to further aid in their understanding of colour vision deficiencies. This module covered many research papers to do with ICV and discussions were held on a weekly basis with the class to cover them in detail.

# 3. DESIGN

## 3.1 Decisions and Trade-offs

The student required a platform to develop ICVGoggles on. The Oculus SDK is available for Visual Studio and for Unity; both platforms could be used for the project. It was decided by both the student and their supervisor to write the software in C++ on Visual Studio as opposed to within Unity. This is because Unity was designed to create games; even though a simulation tool is entirely possible, it would have still been created in a platform centred on a different technical area. This means there would have been a large amount of unused overhead which can cause system slowdown if the application was not optimised correctly. On the other hand, projects created with C++ in Visual Studio contain just the essentials in order to get up and running. Since the system will be doing image manipulation at around twenty times per second, it is necessary to ensure there is as little overhead as possible to maintain a respectable frame rate.

### *3.1.1 Personalised Simulations Delay*

Due to the timescale of the project, the student and supervisor felt it would be best to postpone the production of personalised simulations until the adjustable ones were complete. By doing so, it gave the student enough time to finish the adjustable simulations for user testing. The personalised simulations have been added to future work and will be tackled once adjustable simulations work and have been tested by users.

## 3.2 Software & Hardware

### *3.2.1 C++ in Visual Studio*

The coding language of choice for ICVGoggles is C++ since it is what the Oculus SDK is coded in. C++ is a middle level language and is extremely versatile; it allows for low-level memory manipulation [10] which is ideal for changing RGB values efficiently. The student had not touched upon this programming language for a few years and so it was important to solidify the fundamentals early through revision. Thankfully OpenFrameworks contains a project generator which aided in starting up greatly. Visual Studio was chosen as it is the development platform the student is most confident with.

### *3.2.2 OpenFrameworks*

Coding ICVGoggles with just C++ and the Oculus SDK is a difficult task without the help of a framework. OpenFrameworks is an intuitive open-source C++ framework designed to assist creative processes [11]. The toolkits design philosophy is DIWO (Do it with Others), where creating content is done as a community. This proved helpful during the design process as members of the online forum actively aided in progressing with some small issues for ICVGoggles. OpenFrameworks is very similar to the programming language Processing [12]; a language the student had touched upon in previous years of study. Because of this, it was easier for the student to learn the framework structure and documentation.

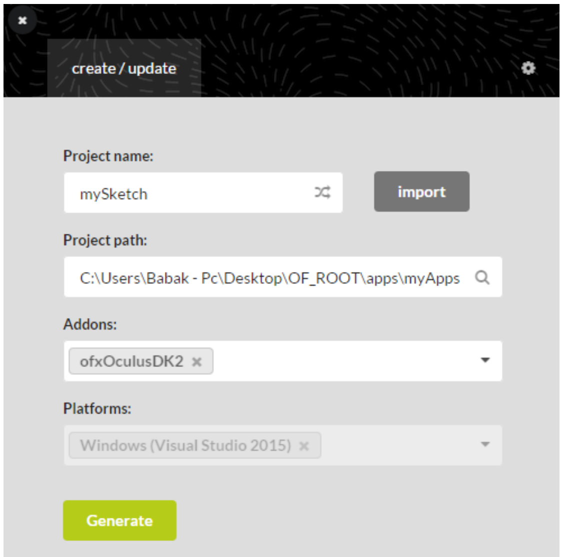


**Figure 3 - OpenFrameworks Project Generation**

OpenFrameworks comes with a project generation tool. After searching and finding which IDE's are installed on the machine, the tool allows the user to select additional add-ons to include in the project. For ICVGoggles, the only add-on used was ofxOculusDK2.

### *3.2.3 Oculus SDK v0.6*

In order to utilise the Oculus Rift, the software development kit (SDK) must be installed. This large package contains all code required to use the headset. The Oculus Rift is mostly used for Virtual Reality (VR) and so many code examples and tutorials cater for this. ICVGoggles only requires feeding a camera stream to the headset and so the student decided to use a community add-on which facilitated this functionality. It was important for the student to use the SDK version 0.6 as the community addon required this specific version.

### *3.2.4 ofxOculusDK2 Add-on*

Since OpenFrameworks is open-source, there is a vibrant and active community which revolves around it. It is commonplace for developers to create helpful packages and share them amongst the community. ofxOculusDK2 (OpenFrameworks Oculus Developer Kit 2) was created by developer Andreas Muller and refactored by James George [13]. This add-on accesses certain parts of the Oculus SDK to enable basic rift rendering functionality, giving the student a lightweight gateway to simulate ICV. The fish eye distortion is applied automatically to anything drawn on the left or right eye, meaning the student had one less task to worry about.

### *3.2.5 Oculus Rift*

The Oculus Rift was chosen as the headset for ICVGoggles to use; specifically the Dev Kit 2 which was released on March 25th 2014. The headset has an OLED display and uses a positional tracking system which enables the movement within a 3D space. For ICVGoggles, both eyes will be drawing and manipulating a camera feed to simulate the real world.

### *3.2.6 OVRVision Cameras*

In order to simulate the world in real-time, cameras are required for the Oculus Rift. OVRVision was created by a small tech start-up in Japan and provides two cameras to be mounted on the Oculus Rift. OpenFrameworks was able to detect both mounted cameras and so it was then possible to continue coding.

**Figure 4 - The Oculus DK2 with OVRVision Cameras attached.**

### *3.2.7 Oculus Runtime v0.7*

The Oculus Runtime is required to detect the Oculus HMD. This piece of software is essential to run ICVGoggles as it allows the Oculus Rift HMD to become a secondary monitor. It is important for users to have the Oculus Runtime <0.7 as the secondary monitor feature was removed from future versions.

## 3.3 Design Process

### *3.3.1 OpenFrameworks Application Template*

OpenFrameworks projects have a default template which contains the basic structure of an application. Before the student could start designing the software for ICVGoggles, they had to learn the structure OpenFrameworks applications follow. Figure 5 shows the basic templates for a new project. When an application is built and ran, the main method is the first to be initiated. From here, an instance of the application begins and the code flow moves to ofApp.cpp.

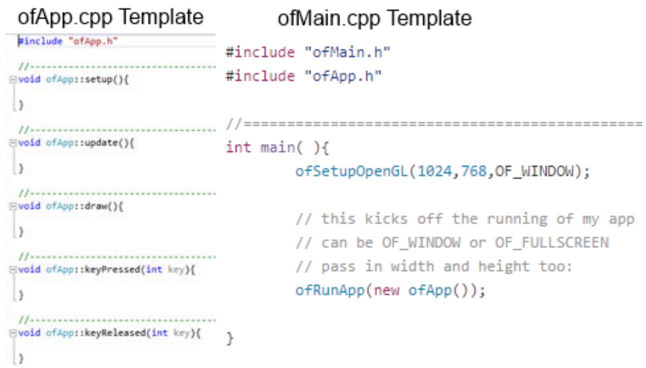


**Figure 5 - OpenFrameworks ofApp.cpp & ofMain.cpp Templates**

The setup method initialises properties and is ran only once. From here, the update and draw methods loop endlessly, creating a running application. After understanding how applications operate in OpenFrameworks, the student then applied their past knowledge of Processing and C++ to design each feature from the specification.

### *3.3.2 Feature Driven Development Methodology*

The feature driven development (FDD) methodology [14] was the most suitable for ICVGoggles as the template for OpenFrameworks projects develops the overall model automatically. The feature list (See 2.3 Feature List) was designed by the supervisor and student. Since this stage of the development process was already completed, the project could progress to planning each features design. The features were ordered sequentially since they depend on previous features to operate correctly. For example, the camera feed must be functional in order for the student to implement ICV simulations. From here on, the student developed features and frequently committed work to the GitHub repository (See Appendix G - GitHub Commit Statistics). The students log book consists of a mixture between the weekly meetings and the GitHub repository. Commits can all be traced to the date and link directly with goals set during the weekly meetings with supervisor.

### *3.3.3 ICVGoggles Workflow*

The workflow for ICVGoggles is seen in Appendix C and shows how the code works in real time. Once initialised, the application runs the setup code and some variables are set. Type and severity are both integers which can be used in to locate variables in the 4D float array. The first dimension has a size of three and this represents the three simulations available: protanopia, deuteranopia and tritanopia. The second dimension has a size of ten and this represents the severity which ranges from one to ten. By having access to changing the severity and type while the application is running, the user can change simulations in real-time.

## 3.4 Ethics & User Testing

The student and supervisor agreed that user testing should be done with this project. Initially the student was going to do quantitative tests with and without the headset to gauge its accuracy and effectiveness. However the focus changed to qualitative data from participants, as both the student and supervisor felt there would be more rich information given this way.

Eventually the student finalised their user testing plans. Whilst the application is in development, the student would begin doing Pre-exposure Interviews with designers and programmers. In these short interviews, data would be gathered about the participants' current understanding of ICV and to invite them to the second stage of testing.

When the application is able to simulate adjustable ICV, the participants will be asked to return to try out ICVGoggles. The second stage evaluations are more in depth than the pre-exposure interviews and participants gain the opportunity to play about with the settings and view their surroundings. For both parts of user testing, the student will be recording the session with a video camera. The participants must read the ethics form and consent must be gained before recordings can take place.

The ethics documentation can be found in Appendix H and all of these forms have been reviewed and accepted by the Ethics committee at the University of Dundee. Explicit consent is required and a copy of all forms signed must be given to participants. When referring to their data, the student guaranteed anonymity.

# 4. IMPLEMENTATION & TESTING

## 4.1 Production

Soon after the ethics documentation had been completed and sent off, production of the code began. Code was only committed to the repository if the project could be built successfully and if it worked as intended. The student spent the first few weeks setting up the development environment. This was an unnecessarily long process as it took a while for the student to discover the ofxOculusDK2 addon only worked on Windows 8.1 or below. Since the student was developing on Windows 10, they had to partition their hard drive for a previous version. It also took a while to link the Oculus SDK correctly to the Visual Studio solution. When importing the ofxOculusDK2 addon, the C++ library links remained the way there were on the developers IDE. This caused numerous build errors; the solution was to include the addon through the project generator instead of copying it into the addon folder. Eventually the project was set up and all of the header files from addons were linking correctly.

### *4.1.1 Global Variables & Setup Function*

The global variables and objects for ICVGoggles are declared in both Main.cpp and within ofApp.cpp. These include severity, type, ofVideoGrabber (a webcam device), the Oculus HMD, overlay bool and size dimensions. Some of these need to be actively changed whilst the application is running, hence why they have been made globally accessible. The most important object set is the 4D float array, mult. This is how the application can determine which colour values to display depending on the type and severity set.

The setup function is the start of the application and is where the environment is initialised. Firstly, the code forces itself into fullscreen since the Oculus Rift has difficulty rendering windowed applications. Then the system sets the default variables such as zero for severity and type and overlay as false. Lastly, the application creates a char * array to store each individual pixel on a texture with dimensions set earlier.

### *4.1.2 Camera Feed on Oculus Rift*

The oculusRift object is the end result of the ofxOculusDK2 addon; it provides several basic functions to interact with the Oculus Rift in OpenFrameworks. This object has its base camera set to an ofCam (OpenFrameworks Cam object), which initiates the canvas for the Oculus to render. Once the setup method has finished, the update and draw methods loop multiple times a second.

```
void ofApp::draw(){
 ofSetColor(255);
 glEnable(GL_DEPTH_TEST);
 oculusRift.beginLeftEye();
 drawSceneLeftEye();
 oculusRift.endLeftEye();
 oculusRift.beginRightEye();
 drawSceneLeftEye();
 oculusRift.endRightEye();
 oculusRift.draw();
 glDisable(GL_DEPTH_TEST);
```

**Figure 6 - ofxOculusDK2 methods used**

Functions from the addon ofxOculusDK2 are used in the draw method of the application as seen in Figure 6. Each eye is begun individually, with a drawSceneLeftEye method called before they end. This method simply draws whatever is loaded into the mirror texture, in this case, the camera feed. Mirror texture is simply a texture object which will store the newly horizontally flipped pixels. The mirror texture is mapped to the Oculus Rift automatically by ofxOculusDK2 and moves with the head.

The camera feed is updated multiple times per second in the update method. Before simulations were included, the update method began with just the allocation of camera feed frames to a texture.

### *4.1.3 Pixel Colour Splitting*

The camera feed was now displaying on the Oculus Rift, however the output was horizontally flipped. After browsing the framework forums, the student found a solution to this and a way to progress with simulations. A user pointed out the frames can be split into their pixels and these pixels can be split into their red, green and blue values. The application checks if there is a new frame, if so the entire image is stripped down and the pixels are reordered to horizontally flip itself.

### *4.1.4 Monochromacy Simulation*

With the applications now accessing each pixels RGB values, the student added a keyPressed option to switch between the types. If the type value became 3, the update method would specifically change the pixels to become monochrome using a luminosity method found from "Converting to Greyscale" by John D. Cook [15], a blog post on the internet. The value range for RGB is 0 - 255 and all colours on screens are comprised of three values between that range. By doing certain multiplications, accurate monochromacy can be simulated. Initially the student simply removed two of the three values being returned to simulate monochromacy, however the values from John D. Cook produce a more accurate representation. The code to change the values to monochromacy can be found starting at line 174 in ofApp.cpp.

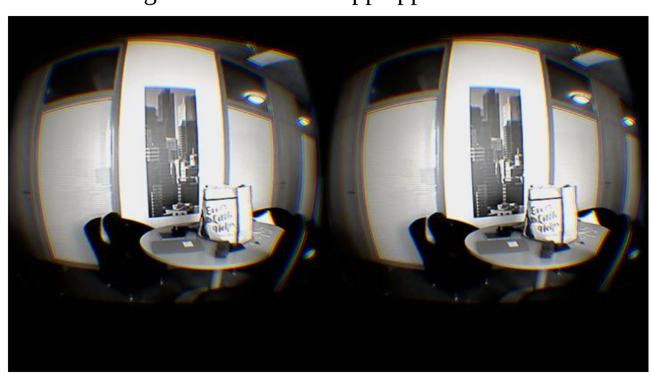

**Figure 7 - Luminosity method values applied to RGB values of each pixel with example monochromacy simulation.**

### *4.1.5 Protan, Deutan and Tritan Simulations*

Once the Monochromacy simulations were working correctly, the student moved onto simulating other types of ICV. The calculations for each colour starts at line 178 in ofApp.cpp and these were heavily influenced from those done in the CVDSimulation Processing application. The correctValue function which wraps around each pixel calculation is to ensure the char value does not exceed 255 or go below zero. This prevents integer overflow, which causes certain areas of the image to change into a bright colour, mostly colours which that type of ICV should not be able to see.

The RGB values are calculated by multiplying the pixels' RGB to values in the mult 4D array. The first two dimensions of mult were explained in 3.3.3 ICVGoggles Workflow and depend on the type and severity values. The third dimension depends on which pixel to change, be it Red, Green or Blue. The fourth dimension depends on the position of where the multiplication is taking place.

With all of these multiplications combined, the application now simulates ICV depending on what type and severity are set. Each pixel is split into its RGB components and the values are swapped depending on the settings; surprisingly the application runs efficiently and can reach thirty frames per second.

### *4.1.6 Control Scheme Development*

Deciding upon a control scheme was a difficult task for the student, as it was the first time they had considered designing for a user with a VR headset on. The control scheme proposed in 2.3.4 Control Scheme seemed the most logical. Left and right click were made to increase and decrease severity as the student felt users would change this value quite often in comparison to type. Changing the type is mapped to 1-4 on the keyboard; when the application is deployed, the keys will have a tactile object such as blue tack attached to aid the users in finding the keys.

```
void ofApp::keyPressed(int key){

    if (key == 'f') {
        ofToggleFullscreen(); //Toggle Fullscreen
    }
    if(key == 'r'){
        oculusRift.reset();  //Reset OculusRift position & values
        type=0;
        severity=0;
    }
    if(key == '1'){
        type=0;   //Protan
    }
    if(key == '2'){
        type=1;   //Deutan
    }
    if(key == '3'){
        type=2;   //Tritan
    }
    if(key == '4'){
        type=3;   //Monochromacy
    }
    if(key == '='){
        if(severity<10) //Increase Severity: Max value 10
            severity=severity+1;
    }
    if(key == '-'){
        if(severity>0) //Decrease Severity: Min value 0
            severity=severity-1;
    }
    if(key == ' '){
        if(overlay==false) //Show Overlay (Spacebar key)
            overlay=true;
        else
            overlay=false;
    }
}
```

**Figure 8 - ICVGoggles code for keyboard controls.**

The reset functionality was added after user testing was done for the second stage evaluations. This is because Participant 6 felt it would be helpful if there was a key which would reset the values so they could see the difference instantly.

### *4.1.7 HUD Development*

Before starting user testing, the student decided to complete the HUD, since it would contain keyboard controls and helpful information. The bool named overlay is initially set to false, however when the spacebar is pressed, the state of overlay toggles. When overlay is true, the oculusRift.beginOverlay method starts the drawing process. Between here and oculusRift.endOverlay, all draw methods apply to the overlay, a special area designated to move with the head positioning. This is why the overlay stays in a fixed position.

## 4.2 Testing & Debugging

Since Visual Studio was being used as the development environment, ICVGoggles could be rapidly tested thanks to the debugger tool included. Once ofxOculusDK2 was included in the project successfully, the student could make minor adjustments to the draw methods and rebuild the solution. This meant progress could be constantly tested; only working code which could be built was ever committed to the GitHub repository.

During the testing process, the student discovered multiple errors, some of which were corrected. For example, without the correctValue() wrapper function, ICVGoggles returned values greater than 255 for some pixel colours. This resulted in colours being displayed erroneously. Another bug found was to do with the OVRVision cameras. Due to a USB power issue, both cameras could not be instantiated at the same time. This resulted in the student using just one camera and blending them into both eyes.

Debugging was slightly difficult as the source code could not be loaded onto any machine other than the development one in use at the student's home. This is because C++ uses linking to other libraries, and these paths were all set for Babak – PC. When downloaded and unpackaged on another machine, the project does not load correctly since the IDE cannot find the linked libraries. This meant the student had to describe his errors with his supervisor and peers when asking for help, as opposed to showing them. Thankfully this did not hinder the student too much as the OpenFrameworks forum user base is extremely active.

Two previous students at the University of Dundee had used the Oculus Rift and mounted cameras in the past and the student liaised multiple times to aid in debugging ICVGoggles. One reason was because the documentation for OVRVision is in Japanese and so the student could not understand how to use them correctly.

## 4.3 Finished Features & Release Package

With adjustable simulations working and the controls set up, the student felt the application was ready to be used in the second stage evaluations. Figure 9 compares an image captured with normal colour vision with Protanopia set on severity 10. Both eyes are shown in the images, however when the headset is on, these two images are blended together to form one image. This functionality is provided in the ofxOculusDK2 add-on. The most noticeable difference is the lack of red in the bottom image. The release version of ICVGoggles has the following types of ICV: Protanopia, Deuteranopia, Tritanopia and Monochromacy. The severity of each type can be set from zero to ten, zero being normal colour vision and ten being the highest severity the headset can simulate. There are a few extra keyboard shortcuts such as "R" which resets type and severity and "F" which toggles fullscreen (to view the console or resize window).

**Figure 9 - Normal colour vision compared to Protanopia on severity 10.**

The control scheme can be seen if the user presses spacebar and toggles the HUD. This can be seen in Figure 10, where the severity and type are also displayed. The images used here can be found in the GitHub repository under the folder "Images" in much higher quality. Figure 10 also displays the Tritan simulation on severity ten; the colours turn pinkish.

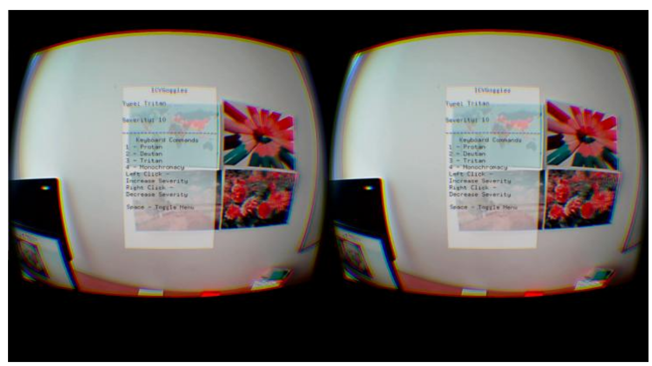

**Figure 10 - HUD in ICVGoggles.**

The release package was built in Visual Studio and contains all of the necessary files to run ICVGoggles apart from the Oculus Runtime. There is also an instructions manual seen in Appendix F which contains an install guide, controls and system requirements. The instructions manual also has the project guide attached.

## 4.4 Project Plan Deviations

Due to certain limitations and time constraints, the student had to deviate slightly from the project plan in order to get a working copy of ICVGoggles for the user testing. Firstly, personalised simulations are not in the release package, however it will most likely be the first feature to complete in the future. The student and supervisor felt the adjustable simulations would be an ideal feature to complete for user testing, as changing the severity would give users a sense of scale. The OVRVision cameras held the project back as it took a while for the student to find out the two cameras would not stream at the same time in OpenFrameworks. One possible solution is for the student to invest in a new set of front facing cameras, the functionality already exists and so it would be quick to code ICVGoggles to use two cameras.

It took the student a while to realise the ofxOculusDK2 add on did not work with the any Oculus Runtime above 0.6. The developers at Oculus decided to remove the "Extended Desktop" feature of the runtime; it enabled the headset to act as a PC monitor. With the Oculus 0.6 runtime installed, the user could successfully test ICVGoggles on the headset by opening the application from within.

During the second stage evaluations, Participant 6 asked if the student could add a reset button to the control scheme. This would enable the participant to switch to default settings and quickly see the colour differences. This functionality was added and so pressing the R key now resets type to Protanopia and severity to zero.

# 5. DESCRIPTION OF FINAL PRODUCT

ICVGoggles provides adjustable simulations of ICV in real-time through the use of the Oculus Rift. Users can select from four different types of ICV to simulate where their severities can be changed; watch as the world around you changes colour. The controls are simple to learn and the application provides an engaging and informative look at how some people with ICV see the world. The only equipment required to run ICVGoggles is a PC with mouse and keyboard, Oculus Rift and front mounted cameras. With the Oculus Rift headset on, users have their hands free to interact with objects, something not possible with other real-time ICV simulation applications. Colours change as the user adjusts the severity setting with the left and right mouse buttons. Attempting colour blindness tests with and without the headset on can be an enlightening experience.

The application was developed over the course of the student's fourth and final year of University. The source code is written in C++ and the IDE used was Visual Studio. This project is heavily influenced off of previous work and is the next step in ICV simulations. ICVGoggles can run on any system which has the Oculus Runtime <0.6.

The source code and application can be found on the students' GitHub page, specifically within the ICVGoggles repository. This repository also contains documents such as the instruction manual and several images of participants using the system in the lab.

User testing took place over two sessions with the help of designers from DJCAD and the University of Dundee. All participants invited had created a visual graphic or website for users in the past and so were ideal to test the system on.

# 6. EVALUATION

## 6.1 Participant Demographics

### *6.1.1 Overview*

All participants were between the ages 22 and 26 with the average age being 24. This meant the whole demographic was limited to a relatively young audience, however all participants were designers which was all that mattered. Seven of the participants were male, with the other three being female. Gender did not show to have any effect on results and comments made. Nine of the participants are currently students and the tenth is a graduate, six study Applied Computing at the University of Dundee. Three participants are currently studying Digital Interaction Design at DJCAD (Duncan of Jordanstone College of Art & Design). Although only ten participants were used, the demographics split seems balanced and representative of web and graphics designers.

### *6.1.2 Previous Design Experience*

All participants have designed for a project in the past which uses colours of their choosing being prime candidates for the user study. Designers were chosen as they work with colour all of the time; it was valuable obtaining feedback from users who potential don't take ICV into consideration but should. Participant 2 had designed a kinetic story telling adventure for a project and so was familiar with taking other users into consideration. Four participants brought in their own work so that they could view their colour choices with various simulations of ICV. If any designers who participated did not consider ICV in their design practice, they probably will now they have tried ICVGoggles.

## 6.2 Pre-exposure Interviews

During the development of ICVGoggles, the student sent out emails to DJCAD designers and members of the Computing Society at the University of Dundee. The emails contained an invite to the pre-exposure interviews and second stage evaluations in order to gather qualitative data about ICVGoggles. Ten people responded and became participants for the study and individual interviews were set up. The student gained explicit consent from each participant using documentation accepted by the Ethics Committee. This consent gave the student permission to record each interview; any use of data gathered from these videos has been anonymised by the student.

### 6.2.1 Pre-exposure Procedure

Each participant was invited to the Queen Mother Building to do the initial pre-exposure interview. The student used empty lab space so the video recording would be clear. After welcoming the participant, the student handed over two copies of the information sheet and consent form combined together. One copy was for the student to keep and the other was for the participant. Once all forms were completed, the student turned on the video camera and begun asking questions. The student tried to continue the questions on to create a conversation in order to gain as much rich information as possible.

After the questions, the student informed the participant about the second stage evaluation. This would take place when ICVGoggles could simulate multiple types of ICV. The student also asked the participant if they would like to bring in their own work for the second stage evaluation. This would allow the participants with a background in design to see their own work through the eyes of an ICV viewer. Most participants welcomed this idea and left the lab with simulating their own work in mind.

### 6.2.2 Pre-Exposure Questions

The questions were created for both user testing processes early on in the project lifecycle since they had to be approved by the Ethics Committee first, which can take some time.

**Question 1**: Do you know anyone who has ICV? If so, have they mentioned any obstacles or difficulties when viewing graphics or interfaces?

**Rationale**: The initial question is to see if the participant knows anyone who has ICV and if they have an understanding of how it can affect people.

**Question 2**: Have you had to design for users with a sensory impairment in the past? If so, what did you design?

**Rationale**: This question asks if the participant has had to consider different types of users in the past for sensory impairments.

**Question 3**: Do you consider ICV users in your design practice? If so, how do you do so?

**Rationale**: If the participant does not consider ICV users when designing content with colour, they may be at risk of producing unsuitable colour palettes.

**Question 4**: Have you had to design for users with ICV in the past? If so, what did you design?

**Rationale**: This question aims to gain an understanding on how designers have created content for ICV users in the past.

**Question 5**: How would you approach the task of creating a graphic or interface for a user with ICV?

**Rationale**: This is an open question which aims to see how the participant would design for an ICV user at that current moment, exploring techniques they would employ.

**Question 6**: Would you consider using external tools to aid in your design practice for users with ICV?

**Rationale**: If not mentioned previously, this question aims to prod the participant into talking about current ICV simulation tool they know about, if they know any at all.

**Question 7**: Would an ICV simulation tool on a tablet/smartphone suffice for this? Or do you believe a hands free option would be better?

**Rationale**: This question aims to gauge the need for ICVGoggles given that there are currently other simulation techniques available.

## 6.3 Second Stage Evaluation

The second stage evaluation took place once ICVGoggles was able to do adjustable simulations. The participants were asked to come back in at a time of their choice to engage in the practical part of the study.

### 6.3.1 Lab Setup

The study required a quiet, isolated area to take place. The student spoke with their supervisor and was advised to ask about the SiDE Lab in the Queen Mother Building. This lab is used by the members of staff who are actively working on the Social Inclusion through the Digital Economy (SiDE) project. After gaining administrator access to one of the lab machines, the student was able to install all of the necessary software to get ICVGoggles working. The lab was prepared thirty minutes before the participant arrived to ensure the software was working correctly. The student brought in several photographs on A4 paper which were all extremely colourful. These were stuck on the wall using white tack in front of the participant's desk. The excess white tack was stuck on the keyboard keys 1-4 so the participants could feel the controls with the headset on. Finally, the student laid out ten yarn balls which were all different colours. These would be used to test the participant’s perception of colour with and without the headset on.

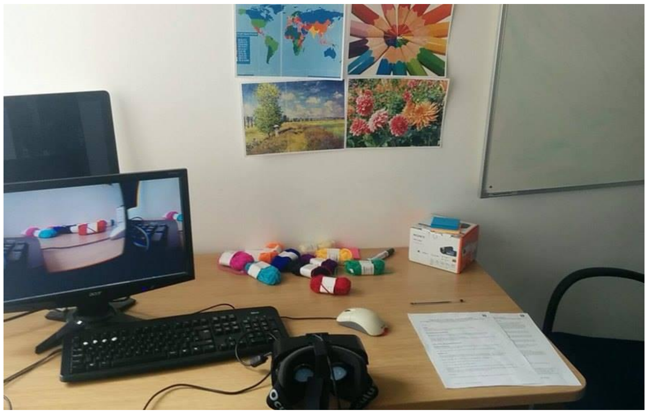

**Figure 11 - ICVGoggles set up in the lab with coloured yarn balls and posters.**

Figure 11 shows the equipment all set up in the SiDE lab. The coloured pictures on the wall include a demographic map from the internet. These images were all chosen since they have vibrant colours in; it was important for the participants to be exposed to the ‘narrowness’ of colour that occurs with ICV. The lab was tidied up and the computer was removed before locking the room up.

### *6.3.2 Second Stage Evaluation Procedure*

With the lab set up, the participant was welcomed back to two more information and consent forms. After reading them thoroughly and signing the date, the participant was ready to begin. This stage of user testing was also recorded with a video camera. Before trying on the headset, the student first asked the participant to group the yarn balls however they like, in terms of colour. Once completed, the student then set ICVGoggles to Protanopia on severity 10 before giving it to the participant. While the participant was putting on the headset and adjusting the straps, the student shuffled the yarn balls. With the headset on, the participant was then asked to order the yarn balls again. Doing this would hopefully provoke an experience of understanding.

**Figure 12 - Participant viewing their own designs with ICVGoggles.**

After looking at the newly organised yarn balls, the participant was asked to put the headset back on. The student then explained the controls for ICVGoggles before giving the participant several minutes to explore the room and their surroundings with different simulations. Some participants brought in their own work to view with the headset on as seen in Figure 12.

### *6.3.3 Second Stage Evaluation Questions*

**Question 1**: Has your understanding of ICV increased, decreased or stayed the same through the use of ICVGoggles? Please explain.

**Rationale**: This question aims to see if ICVGoggles can improve the user's understanding of ICV.

**Question 2**: How did you personally find the experience of using ICVGoggles? How did you feel when you first saw a simulation?

**Rationale**: This question aims to see if the participant had any particular feelings when they first tried the headset simulations.

**Question 3**: How did you find the yarn ball colour organising exercise after putting the headset on?

**Rationale**: The student made the participants order the yarn balls with and without ICVGoggles on. This was to see if the task changes in difficulty when ICV is simulated.

**Question 4**: If ICVGoggles were available, would you consider using them in your design practice? If so, how?

**Rationale**: With so many ICV simulation tool available, this question gauges the need for a hands-free solution.

**Question 5**: If ICVGoggles were available, would you recommend them to a graphics or web designers? If so, please explain why.

**Rationale**: This question leads on from the previous in seeing if ICVGoggles is usable and needed by designers.

**Question 6**: Are there any additional comments you would like to make regarding the experiment or equipment used?

**Rationale**: This final question is open ended and gives the participant some free space into injecting their own thoughts and opinions.

## 6.4 User Testing Results

### *6.4.1 Pre-Exposure Results*

The student uploaded the video recordings from each participant and transcribed their answers to each question. These answers were then analysed to group opinions and experiences. Nine out of ten participants know someone personally who has ICV. Participant 3 mentioned their friend with ICV had trouble distinguishing features in maps and graphs. Three of the participants had designed for users with sensory impairments; Participant 2 created a kinetic story telling project for their dissertation. Two participants had designed a website which took ICV users into consideration. Interestingly, 40% of participants consider ICV in their design practice, however 80% have not had to design for ICV users specifically in the past.

When asked how they would approach creating a graphic or interface for a user with ICV, five out of ten participants said they would search for a tool to aid in their design. Three participants said they would research ICV and move on from there. Participant 8 said they would ask a person with ICV to check their work. All participants said they would consider using external tools to aid in their design practice for users with ICV.

Participants 4 and 8 felt a hands free solution would make designing for ICV users easier. Participant 6 said they would need to test the system before making a decision. Five out of ten participants felt a hands free solution would be better than a tablet/phone application. Four participants thought tablet and phone simulations would be good enough for their design. The pre-exposure transcripts can be found in Appendix D.

### *6.4.2 Second Stage Evaluation Results*

When asked to organise the coloured yarn balls, the participants had varying speeds of completion. Six completed the task quickly with not much hesitation while two worked in silence, concentrating on their colour ordering.

The participants then put on the headset while the student shuffled the yarn balls. Figure 13 shows a participant doing the yarn ball organising task. Three participants expressed shock when first looking at Protanopia on severity 10 and this time only two completed the task quickly. The other eight participants spent more time on this task than the previous. Four participants expressed hesitation when selecting and ordering the yarn balls. Two participants were very surprised at the ordering they did with the headset on, Participant 1 said "It immediately made me feel very disabled, I was missing a lot!"


**Figure 13 - Participant organising yarn balls with ICVGoggles on.**

Eight participants picked up the controls quickly, with six commenting how they liked them. Two participants said the sticky tack helped them identify which keys to press with the headset on. Participants 6 and 9 had suggestions to improve the control scheme and these were taken into consideration before the final submission. Participants 7 and 8 mentioned how they liked the menu and how they found it helpful.

Once the participants had spent some time learning the controls, they were then asked to explore their surroundings and adjust the severity and type freely. Seven participants expressed wonder and amazement when playing about with the settings with six specifically mentioning Tritanopia. Participant 7 said "All the colour are incredibly vivid with Tritanopia" and Participant 9 said "Life would be really vibrant if you had Tritanopia severely". Seven participants spent some time examining the images printed out by the student. Out of these seven, four expressed difficulty when viewing the coloured map. Six participants enjoyed adjusting the severity for long periods of time to see how differently ICV can affect people. Three participants did not know there were different severities of ICV. Participant 5 said "It is weird for me to image the world as being just these colours".

Four of the ten participants brought in their own work to view with ICVGoggles on and all of them found the experience to be helpful. Two participants found colours in their work which could not be distinguished, Participant 1 said "My chef app recipes look a lot less appetising now".

The participants were asked some final questions after they had finished exploring the room with ICVGoggles on.

**Question 1 Responses:** Nine out of ten participants felt their understanding of ICV had increased through the use of ICVGoggles. Participants 1, 6 and 10 said the tool helped them think how people with ICV may feel. Five participants were shocked and didn't know how badly ICV could affect someone. Participant 4 said it was very interesting to see through the eyes as opposed to a screen.

**Question 2 Reponses:** Three participants found the experience different and interesting. Five participants wear glasses and so it was a strange experience having long distance emulated. Three participants said quite a few colours looked the same.

**Question 3 Responses:** Five participants expressed difficulty in distinguishing colours with the yarn balls after putting the headset. Nine out of ten participants found the task more challenging with the headset on. Three participants used shades and brightness to order the yarn balls. Participant 8 said they felt frustrated doing the task with the headset on.

**Question 4 Reponses:** Eight participants said they would consider using ICVGoggles in their design practice if they were available. One said they would only consider them if the hardware is not too expensive. Three participants said ICVGoggles is more suitable for designing physical products instead of on screen. This is because screens can be quite hard to focus on with the headset due to the quality of the OVRVision cameras.

**Question 5 Reponses:** Eight participants said they would recommend ICVGoggles to graphic or web designers. Participant 5 said "There are tools online but they're not the same as this".

**Question 6 Reponses:** For the additional comments, three participants said the camera quality was a bit too low and could be improved upon with better hardware. Seven participants really liked the project and felt there were applications in many different areas; Participant 9 said "This could have applications in teaching to help teachers understand what some of their pupils could be living with". Two participants said they would prefer if they could wear their glasses underneath the headset. Participant 1 wished the wires were longer to allow more freedom of movement.

## 6.5 Main Findings

### *6.5.1 Pre-Exposure Evaluation*

It took the student around three weeks to complete the Pre-exposure interviews. ICVGoggles was in development during this period of time and almost complete, meaning participants were able to complete the study sooner. All participants said they would use an external tool if they had to design for someone with ICV and half of them thought a hands-free option would be better than a phone/tablet application. This shows there is an interest in the use of simulation tools when designing for users with ICV.

Nine participants said they knew someone with ICV, however only four of them actually consider ICV in their design practice. This could suggest that there is an information gap for some of the designers. Four participants said they would bring in their own work to view with ICVGoggles and showed a vested interest in doing so. One participant spoke of a friend who had trouble seeing graphs and maps. The student took this into account and decided to print out colourful images to put on the wall in the lab during the second stage evaluation. One of the images used was a population density map with various bright colours.

Surprisingly, two participants had specifically designed for ICV users in the past. Both of these participants felt a hands-free solution could be more beneficial than a tablet/phone application. Every participant gave off the impression they were excited to try out the headset. This made the student feel there was already an interest in the project.

### *6.5.2 Second Stage Evaluation*

The second stage evaluations took longer than the pre-exposure interviews and required a dedicated lab to increase reproducibility. Participants organised a time to meet with the student at the Queen Mother Building and were invited to the lab. After doing the necessary ethics paperwork, the participants were ready to try ICVGoggles.

It was interesting to notice six participants complete the initial yarn ball organising in a quick manner and six complete the second exercise slower than the first. This shows participants spent more time to investigate the colours before making their decisions. It was evident the task was more difficult with Protanopia on severity 10, with nine participants mentioning this. Four participants hesitated frequently during the task with the headset on. This could indicate more cognitive effort is needed, especially since the simulation is a new experience altogether.

The student was worried about the control scheme initially as they found it difficult to create controls for users with limited vision with the headset on. Reassuringly, eight participants learned the controls quickly with six mentioning how they liked them. The menu was particularly useful as all participants used it to view the settings. In an ideal environment, the student would use hardware with very few buttons to control ICVGoggles.

After finding out graphs were particularly difficult to see with ICV, the participant used a demographics map with colours indicating areas of interest. Four participants mentioned this difficulty in the exploration phase of the study and shows how poor colour choices can affect user experience. Six participants spent part of their exploration time viewing the images printed out, including the graph. Participant 7 said "I can't tell the leaves apart from the petals" when looking at the picture of the colourful field. Comments about viewing images showed a close resemblance to those made by people with ICV[4].

When the project began, the student envisioned ICVGoggles aiding designers; four participants brought in their own work to view with ICV simulations. Half of these participants found colours which could not be distinguished and caused a negative effect on the user experience. All of the participants felt testing with ICVGoggles was a helpful experience. This feedback shows how ICVGoggles can be applied in a participants design process and be beneficial for them. There are still some improvements to make to the system; three participants were slightly disoriented by the depth perception. Also, one participant requested the wires be longer to give more freedom of movement. Both of these changes have be logged in the future work and would not take the student very long to implement.

The questions at the end of the study were to provoke thoughts from the participants. Most questions also asked for a rationale to understand how the participant came to their answer. Nine participants said their understanding of ICV had increased through the use of ICVGoggles. Out of these nine, five said it was because they "...didn't know how bad it could be". Participants 1, 6 and 10 said their understanding increased because it "Made me think how they feel". Participant 4 said it was "Helpful and understanding to see through the eyes and made more of a difference than just screen". With nine participants having an increased understanding after the use of ICVGoggles, it is clear to see the system has a positive impact on the user. If ICVGoggles were available, seven participants would recommend the system to graphics or web designers they know and six would use the system themselves. Participant 5 said "There are tools online but not the same as this." which suggests there is an interest in the system due to the way it delivers the simulations; Participants liked using the Oculus Rift and it was the first VR headset some had used. The camera quality from OVRVision was brought up by three participants and this limited interactions with small objects and some screens.

Overall, the system was well received with participants as seven said they really liked it at the end of the study. It is evident there is potential for ICVGoggles to be used in design practices and the project even has educational applications. There are also clear improvements which must be made such as improving the camera quality and using longer wires. Ideally, more user testing should take place to gain further feedback.

[4] Article about having Colour Blindness: http://wearecolorblind.com/article/guest-article-a-mothers-journey-into-colorblindness/

## 6.6 Usability

Precautions were taken to ensure ICVGoggles was as efficient to use as possible. By spacing the "type" shortcuts as numbers close to each other, the user could navigate through them by touch. This was made easier with white tack applied to the keys, two participants specifically mentioned how it aided them. The controls were picked up quickly by most participants showing the system is easy to learn which improves its usability. Participants were satisfied with the final product and four stated the system helped them with their own designs. The ISO definition of usability is a framework of system acceptability and contains several components: Learnability, Efficiency, Memorability, Errors and Satisfaction. ICVGoggles has demonstrated it is learnable and satisfying to users; the system is fit for purpose and efficient at simulating ICV.

The pre-exposure interviews aided in the production of ICVGoggles as recommendations and comments were taken into account. It was not necessary for the student to prototype, instead all efforts were focused on developing the final application.

# 7. DISCUSSION

This honours project was a learning experience for the student and was guided by weekly meetings with the supervisor. Initially the scale of the project seemed overwhelming, however after creating a Gantt chart for the work schedule the student felt more confident with completing the system. OpenFrameworks aided greatly in the creation of ICVGoggles as the framework supported most of the functionality required. The addon ofxOculusDK2 provided core functionality to the project, enabling the student to easily draw onto the Oculus Rift HMD. Simulating ICV is the systems main goal and this was made possible with the values being provided by previous research [5]. The 4D float array was transferred to C++ and is accessed frequently by ICVGoggles to simulate ICV.

The pre-exposure interviews gave insight into the designers' current perceptions of ICV and how much they take it into consideration in their design practice. The data suggests designers do not normally consider ICV users in their design practice and knowledge of the subject could be improved. It is interesting to note nine out of ten participants knew someone who had ICV, yet so few actually considered it in their design practice. Participants seemed keen to try out the headset in the second stage evaluation which could be due to the novelty of trying an Oculus Rift.

After using ICVGoggles, nine participants said their understanding of ICV had increased. Two participants did not know there were different severities of ICV and were shocked when they tried adjustable simulations. This suggests there is an information gap for some designers, however ICVGoggles combats this by providing a learning experience. Evidence for this information gap is also shown when four participants mentioned difficulty in distinguishing features on a map. By informing users about how ICV can affect people differently with various severities, future work could potentially be accessible to ICV users if taken into consideration. The control scheme was generally liked, however in an ideal environment the system would be set up with a few buttons only. This would make navigating the buttons far easier than small keyboard keys. The extra tactility from the white tack ensured users could find the correct controls to utilise ICVGoggles.

Overall, the project was well received and the design process ensured it was completed for user testing. In hindsight, the student would have spent more time on development to create personalised simulations. However in terms of the aims of the project, adjustable simulations were good enough for the user testing.

## 7.1 Strengths & Weaknesses

ICVGoggles displayed both strengths and weaknesses during the user testing process; there is still more work to be done. The system succeeded in achieving its aim to inform users about ICV and it did so in an impactful manner. Most participants liked the system and took something valuable from the session. The simulations caused multiple participants to realise how ICV may affect someone, showing how impactful ICVGoggles can be. Four participants brought in their own work and two found colours they could not distinguish with certain types of ICV. This demonstrates ICVGoggles can be applied in a design practice to prevent poor colour choices which may confuse users.

The system could be improved by upgrading the cameras on the headset, changing the system to binocular vision and improving the wire lengths. Due to power issues with the hub and OpenFramworks having trouble accessing the cameras, only one was used. This causes monocular vision in the headset, however no participants mentioned this as a problem. An early improvement the student can make is to use two new cameras mounted on the front. Longer wires would allow the user to have more freedom to move and explore the room. Currently the short cable limits the user to sitting down at the computer and increasing the length would improve user experience.

## 7.2 Applications

Potential applications of ICVGoggles were mentioned in sections 1.3 and 1.4 and the user testing process reinforced some of these ideas. Participant 9 said they felt ICVGoggles could be used in a teaching environment to inform teachers about how ICV can affect some of their pupils. Five participants said the headset made them feel how people with ICV feel, giving an indication the system can provide a unique experience. Parents who have children with ICV may benefit from this and could understand what they are going through. Instead of viewing images on a screen or printed out, parents could see the world around them change colour. The student feels this is an impactful way to show ICV and could change opinions and educate.

# 8. APPRAISAL

The design and production processes went to plan and the student completed the software before user testing. Some Gantt chart deadlines were delayed due to version problems with the software and for some visual errors, however it was only important for a working copy to be ready for users. All delays were logged in the minute meetings notes and were reported to the supervisor at the weekly meetings.

## 8.1 Rationale for Design

The functionality for ICVGoggles was already envisioned before the student began the project. When designing, the student took into consideration all of the features necessary using the Feature Driven Design methodology. OpenFrameworks was recommended by the supervisor as a framework to use. Due to its similarity to Processing, the frameworks architecture was understandable for the student. The framework was perfect for ICVGoggles as it fit the design of the application. By repeating draw and update methods, the framework creates an endlessly running 3D space where drawing can take place. Here, the camera feed has its colours converted and displayed on the Oculus Rift.

The features were ordered so that the functionality improves logically towards the final goal of simulating ICV. The decision to delay personalised simulations was made when development slowed down due to coding issues. Instead, the student implemented adjustable simulations first and presented these for the user studies. Three participants specifically mentioned the adjustable option being a good feature. The next step for the student is to plan on implementing personalised simulations.

## 8.2 Rationale for Implementation Decisions

The student decided to include coloured yarn balls and printed images to be placed on the wall for the second stage evaluations. These would serve as colourful objects around the lab which users could view with various types of ICV at different severities. When the sessions were taking place, all participants interacted with the coloured yarn balls with and without the headset on. Seven participants studied the images closely, it is evident the colourful objects aided in helping users understand ICV.

The control scheme was designed to be as easy to learn as possible. This is because using keys with the Oculus Rift headset is quite difficult due to a change in depth perception. The white tack added tactility and enabled users to locate the correct controls. By implementing small features and considering the users vision with the headset, the student felt the controls were well received by participants.

The dedicated lab borrowed from the SiDE team helped the student maintain reproducibility with the study and aided the evaluation process. Each participant completed the study differently, however the process followed some guidelines to ensure the results could be compared. Firstly, the student recorded how the participant completed the yarn organising without the headset on. Then, the student recorded comments made when the headset was used to organise the yarn balls. By comparing and contrasting comments made at each section, the student could come to rationale conclusions based on them.

The poster for ICVGoggles was based on a template provided by the University of Dundee and can be seen in Appendix I. It is laid out so a viewer can gain all the information they need in around thirty seconds of looking. The student placed three images on the poster, one of the headset, one of an example simulation and finally an image of a participant using ICVGoggles. Explicit consent was gained to use the image of the participant.

## 8.3 Lessons Learnt

The student learned they should spend more time reading into the documentation of the SDK's they use in development. Many solutions to the student's problems could have been solved quicker if they had known the source code more in depth. There were also limitations with software available to aid the student and they spent a lot of time trying alternate solutions instead of sticking with one. Since the Oculus Rift is a relatively expensive piece of hardware, not many developers have the chance to develop for one. This means the community online is still small and so finding answers to basic debugging questions was a long and arduous process. Eventually the student became adept at programming for the Oculus Rift as simulations were developed quickly. It was the initial start-up which slowed the student down the most, as well as getting to grips with the development environment and the hardware. After simulations were complete, the student was confident with the second stage evaluations and feels the process went smoothly.

# 9. SUMMARY

## 9.1 Summary

The project was chosen since the student was interested in creating an application which could aid users somehow. The ECVD course was chosen and helped the student understand the problem more, that ICV is not as noticed as it should be. Working with the Oculus Rift was a fun experience since it is a new piece of technology, useful functions were packaged with OpenFrameworks making development much easier. As with all software development, there were blocks the student came across which greatly slowed progress. Such examples include using the wrong operating system, not linking the libraries correctly, being unable to use two cameras and integer overflow which caused images to appear the wrong colours. After tackling these hurdles, the students' progress picked up and they second stage evaluations went smoothly. Organising the participants was a relatively stressful process as the student had to juggle University deadlines and a part time job. ICVGoggles is a functional ICV simulator which has shown to improve understanding of ICV and can aid in designing content. The headset is hands-free and provides adjustable simulations, the aims of the project have been fulfilled and now the student should move onto future work.

## 9.2 Personal Input

I thoroughly enjoyed creating ICVGoggles as the project plunged me into new and exciting fields of computing. The Oculus Rift is an attraction in itself and I witnessed most participants commenting on them being excited to use it. I feel this is really beneficial for the project, as it gauges interest in a project which can educate users. By using a new technology, it seems like people were more eager to try out ICVGoggles.

I feel like the project went well and found the weekly meetings to be critical to the project's success. By being in close contact with the supervisor, ideas could be bounced back and forth more often and allowed the project to evolve over time. This fluidity helped guide the project to where it is now, with features being delayed to ensure users could get a working adjustable ICV experience. The start of the project was probably the most difficult, as I had to pick up C++ again which took a while. Thankfully OpenFrameworks has a very helpful community who helped me if I had a question about the framework. After a couple of months, I feel I became more adept at using OpenFrameworks and drawing on the Oculus Rift.

Hopefully I can continue working on ICVGoggles as there is definitely more work which can be done to improve the project. I feel personalised can be tackled now the adjustable simulations are working correctly and perhaps throughout the summer of 2016 I can continue to refine the system.

### 9.3 Future Work

There are several planned feature for the next iteration of ICVGoggles. Firstly, new cameras will be used and binocular vision will take place inside the headset. This will vastly improve the resolution of the image displayed to users and thus would increase user experience. Users could then look at screens and view small text with simulations on.

The student would then go on to create personalised simulations. Through the HUD, the user should be able to take colour vision tests and somehow the system can simulate their specific type and degree of ICV. With this, the system would have a wider appeal for both educational and scientific purposes.

The system could potentially be ported to a different device such as the Gear VR. This headset has no wires and would make ICVGoggles more accessible since it is a relatively cheap piece of hardware. The system could even be updated to function on the latest Oculus Rift released recently. By increasing the amount of devices the software is available on, there will be more people learning about ICV.

## 10. ACKNOWLEDGEMENTS

The student would like to thank Dr. David Flatla from DAPRLab for being an engaging lecturer for the ECVD (Exploring Colour Vision Deficiencies) module and for guiding them throughout the project lifespan. The student developed a passion for the project after doing the ECVD module and this pushed the student to ensure the project succeeded. The lab was kindly temporarily donated by the SiDE project and gave the student a quiet, isolated area to do the study. The student would like to acknowledge the previous work on which ICVGoggles was built upon. Gustavo M. Machado, Manuel M. Oliviera and Learndro A. F did research which gave functionality to the project. Also the CVDSimulation application created by Dr. David Flatla gave the student insight into how to design the application for the Oculus Rift in C++.

## 11. REFERENCES

[1] Bernhard Jenny, Nathaniel Vaugn Kelso 2007. *Color Design for the Color Vision Impaired.*

[2] Spectrum, offered by Yehor Lvivski for Google Chrome https://chrome.google.com/webstore/detail/spectrum/ofclemegkcmilinpcimpjkfhjfgmhieb?hl=en

[3] Vischeck, Simulation tools for web and photoshop. http://www.vischeck.com/vischeck/

[4] Colourblind Vision by Bradley C. Grimm. https://play.google.com/store/apps/details?id=com.givewaygames.colorblind_ads&hl=en

[5] Chromatic Colour Simulation, CVD simulation app for apple and android systems. http://asada.tukusi.ne.jp/cvsimulator/e/

[6] DAPRLab (Digitally-Augmented Perception Research Lab) http://daprlab.com/

[7] Gustavo M. Machado, Manuel M. Oliveira & Leandro A. F. Fernandes. Previous research used for CVDSimulation. *A Physiologically-based Model for Simulation of Color Vision Deficiency*

[8] Brettel et al 1997. Previous research on ICV simulations. *Computerized simulation of color appearance for dichromats.*

[9] David R. Flatla, Carl Cutwin 2012. Paper covered in ECVD. *"So That's What You See!" Building Understanding with Personalized Simulations of Colour Vision Deficiency.*

[10] Graham M. Seed 1996. C++ and its low level memory management. *An Introduction to Object-Oriented Programming in C++*.

[11] OpenFrameworks C++ Toolkit. www.openframeworks.cc/about/

[12] Processing, the flexible software sketchbook. https://www.processing.org/

[13] ofxOculusDK2 OpenFrameworks Add-on. https://github.com/obviousjim/ofxOculusDK2

[14] Steve R. Palmer, Mac Felsing 2001. Feature Driven Development. *A Practical Guide to Feature-Driven Development*.

[15] John D. Cook 2009. Converting color to Grayscale. *Converting to Grayscale.* http://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/

## 12. APPENDICES

Appendix A - Minute Meetings
Appendix B - Gantt Chart
Appendix C – ICVGoggles Flow Chart
Appendix D – Pre-exposure Transcripts
Appendix E – Second Stage Evaluation Transcripts
Appendix F - Instructions Manual & Project Guide
Appendix G - GitHub Commit Statistics
Appendix H – Ethics Documentation
Appendix I – Demonstration Poster
Appendix J – Mid-Term Progress Report