



Department of Biomedical Engineering

# Virtual Reality Goggles for Visual Field Diagnostics

---

MSc Biomedical Engineering Thesis

Krishna Patel

2014/2015

## **Declaration of Authenticity and Author's Rights**

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has lead to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Signed:

Date:

## **Abstract**

Current methods of perimetry testing present substantial difficulties with regards to patient fatigue and cooperation. This project aims to implement the building block for a game-like perimetry test in virtual reality, which aims to improve the patient compliance. The game elements have been implemented on virtual reality environment using the Unity game development software. The implemented building blocks for this task work correctly. Further studies need to be conducted to define the parameters of the game task as well as the clinical protocol, to implement perimetry testing.

## **Introduction:**

A visual field test, also known as perimetry testing is an examination conducted by optometrist to check for defects in a patient's field of vision and in the peripheral vision. (Flanagan, 2009) This test can help to identify if the defect is due to glaucoma or neurological defects, depending on the location and type of blind spots observed by the examiner. (Gault, 2007) Standard Perimetry testing involves massive bulky instruments, and a high degree of active participation of the patient for fixation point and their response. (Damato, 1985) It requires a patient to actively be fixated at a designated point and responds by pressing a buzzer to indicate a presence of light stimuli in their vision field, if they see it. (Choplin & Edwards, 1998) The massive sizes of the required instruments pose some serious concerns and problems, including the ability to be portable. (Wroblewski, et al., 2014) Current methods rely highly on patient compliance, which is proven to be difficult in children. (Tschopp, et al., 1995) Young patients are very impatient and do not have full control of their reflexes, especially in children under 5 years of age. At this age, children still have natural reflex tendencies to move gaze to the moving object, hence leading to a greater loss of fixation error. (Murray, et al., 2013) Errors, however, can be cause by both children and adults due to point fixation loss, the buzzer response time, the fatigue due to the tedious test and even error from the examiner in marking the point on field map.(Morales, Weitzman & Gonzalez de la Rosa, 2000; Pineles, et al., 2006)

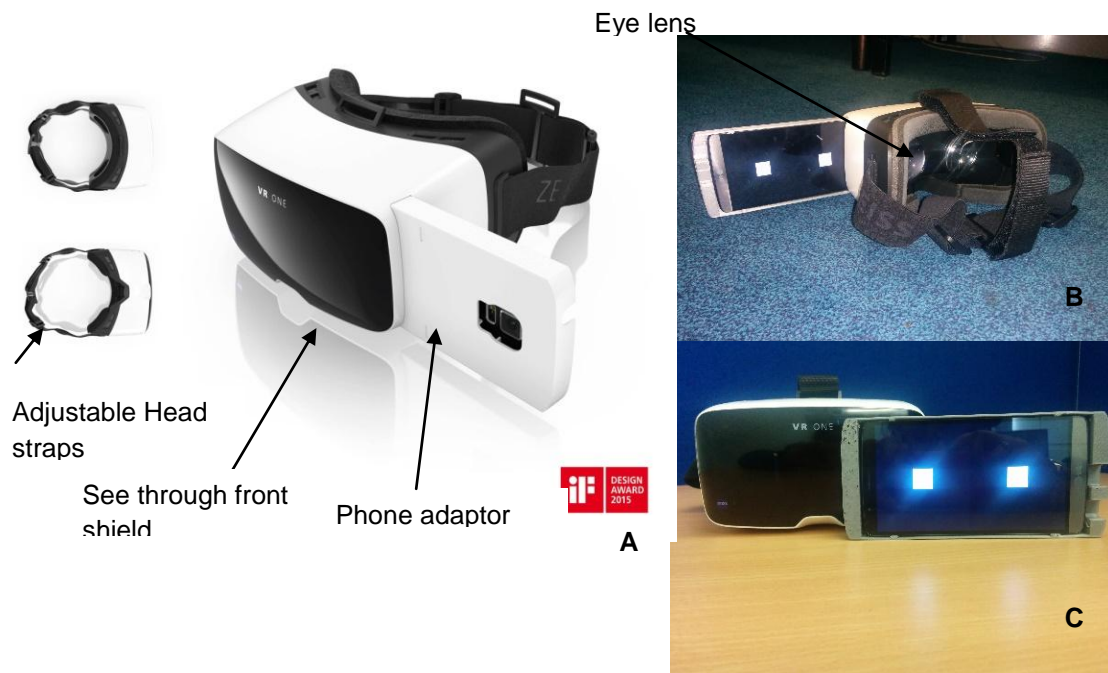
Patients are more focused on the examination if any tedious task is turned into a form of game and are more likely to perform it to the best of their abilities. (Green & Bavelier, 2012) Hence, the approach to use gamification of the standard perimetry testing with virtual reality on a simple smartphone to increase the portability and viability of perimetry testing is being examined. (Hasegwa, Koshino, & Ban, 2015) Virtual reality is widely used in the gaming world today, and is seen as an enhanced experience for the player in the game. It allows the player to experience and interact with the gaming environment in three-dimension during the game. (Burdea & Coiffet, 2003) While using gamification solves the matter of the lack of focus, virtual reality also give the benefit of a wide visual field, as compare to normal game on a phone screen. (King, et al., 2013) Since, we are trying to develop a game like environment to test for visual field, in any game development software with virtual reality ability which allows transferring the game onto multiplatform would be ideal.

The aim of this project will be to set up the first steps into building a portable game for perimetry testing, done on phone with the aid of head mounted virtual reality goggles. In specific, the initial steps to setting up the head tracking, building base elements such as the target, cross hair, and their interaction with each other, and other objects entering the game.

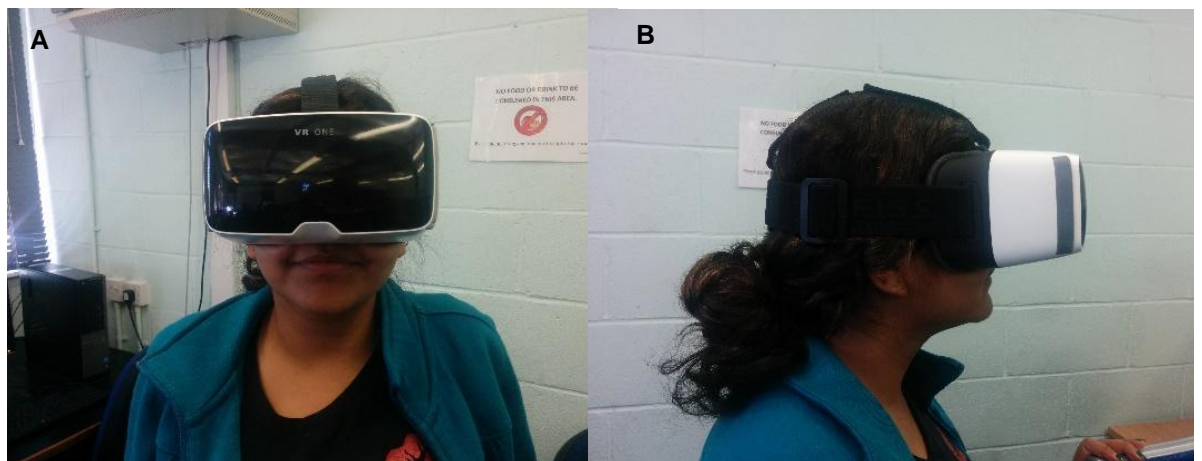
## **Methods:**

### **Hardware:**

We use a head mounted visual device, Zeiss VR One (VR1), Zeiss, Germany virtual reality goggles, see fig. 1. (Zeiss, 2014) The VR1 goggles support smartphone with screen display between 4.7 to 5.2 inches, give approximately a 100° of field of view, and can easily be worn by patient with or without glasses see fig. 2. These goggles can work with multiple different smartphone devices. Hence, we are not restricted to use any one in particular. The default phone adaptor that is currently available, which comes with the goggles is for Samsung Galaxy S5, Samsung, South Korea and iPhone 6, Apple, USA, however CAD model for 3D printing for others, like LG G3, are available to be downloaded from their site. (Samsung, 2015; Apple, 2015) The goggles rely on the smartphone device to play and project the game or scene in the goggle, so the patient can see the scene while wearing the goggles.



*Figure 1: A: Zeiss (2014) Zeiss VR ONE [Textile]. Available at: [http://40.media.tumblr.com/513a4dc22db2f76eee2dce5223050e4c/tumblr\\_nk0hg2Er0z1u0v4y2o1\\_1280.jpg](http://40.media.tumblr.com/513a4dc22db2f76eee2dce5223050e4c/tumblr_nk0hg2Er0z1u0v4y2o1_1280.jpg) (Accessed: 8 August, 2015). (Zeiss, 2014). Zeiss VR1 (A) with Phone Inserted (C)with LG G3.*



*Figure 2: Head Mounted VR1 (A) Front View (B) Side View.*

We are using the LG G3, LG Japan smartphone with 5.0 Lollipop Google, USA, android operating system, to run the virtual reality app, which is used to stream images from the phone and implement augment reality. (LG, 2015; Google, 2014) The goggles use the internal tracking of smartphone sensors to rely tracking information to allow head tracking to be done. The goggles have a see-through front shield, see in fig. 1, which allows the phone camera to augment reality app more easily. We are using this phone as it is ideal for the purpose of this project, it has a quad HD screen with resolution of 1440 x 2560 pixel resolution, which means the image will not get distorted when projected onto the VR1 goggles, or on a much larger canvas. The advanced quad core processor with 2.5GHz on the phone allows us to run complex apps, like this, with ease and without any problem.

#### **Environment/Scene/Software:**

We created the virtual environment using the Unity software by Unity, USA. (Unity, 2015) Unity can create 3D and 2D gaming environments; it allows the user to create a virtual environment with players, threats, and enemies etc, which are all necessary components for building a game. The software allows the user to create the complete virtual scenes and to build them to be used on multiple different platforms, including X-box, Nintendo, iOS, and Android, which is the platform we used. The components developed on the virtual environment in the Unity scene can be assigned movement and motion by using scripts which may be written in either Java or C# languages. We wrote the code for the virtual reality app, for this project in coding language C# in Unity 5.0, 64-bit personal edition. The scene is generated by



the software via the images captured by the virtual camera in the virtual scene, and this can be viewed while building the game. The software itself has a built in virtual reality and head tracking option, which can be enabled to turn the scene created into a virtual reality environment at the end when it is built. One great benefit of using Unity with android phones, is the program is able to access the gyroscope of Android to help it with head tracking, hence giving the game a virtual reality feel when put into the VR1 goggles.

We used Java Development Kit 1.7.0\_80 – 64-bit edition for Windows by Oracle, USA to allow us to transfer the scene we created in Unity to an Android platform. (Oracle, 2015) We were also required to use Android Development Kit, Android Studio, and Android SDK tool for Android 5.0 (Lollipop) platform by Google, USA to allow the Unity scene to be converted it into SDK file, so we can run it on the android system, on LG G3. (Google, 2015) We used a proprieter built computer from University of Strathclyde, UK, running Windows 8, to develop the virtual environment and transfer it to LG G3 phone. The virtual reality and head tracking libraries created by Dr. Mario Giardini, University of Strathclyde, UK, are used in the project rather than the built-in feature in Unity. These libraries are used because it enables the image to be split into two parts on the screen, one for each eye with virtual reality and head tracking.

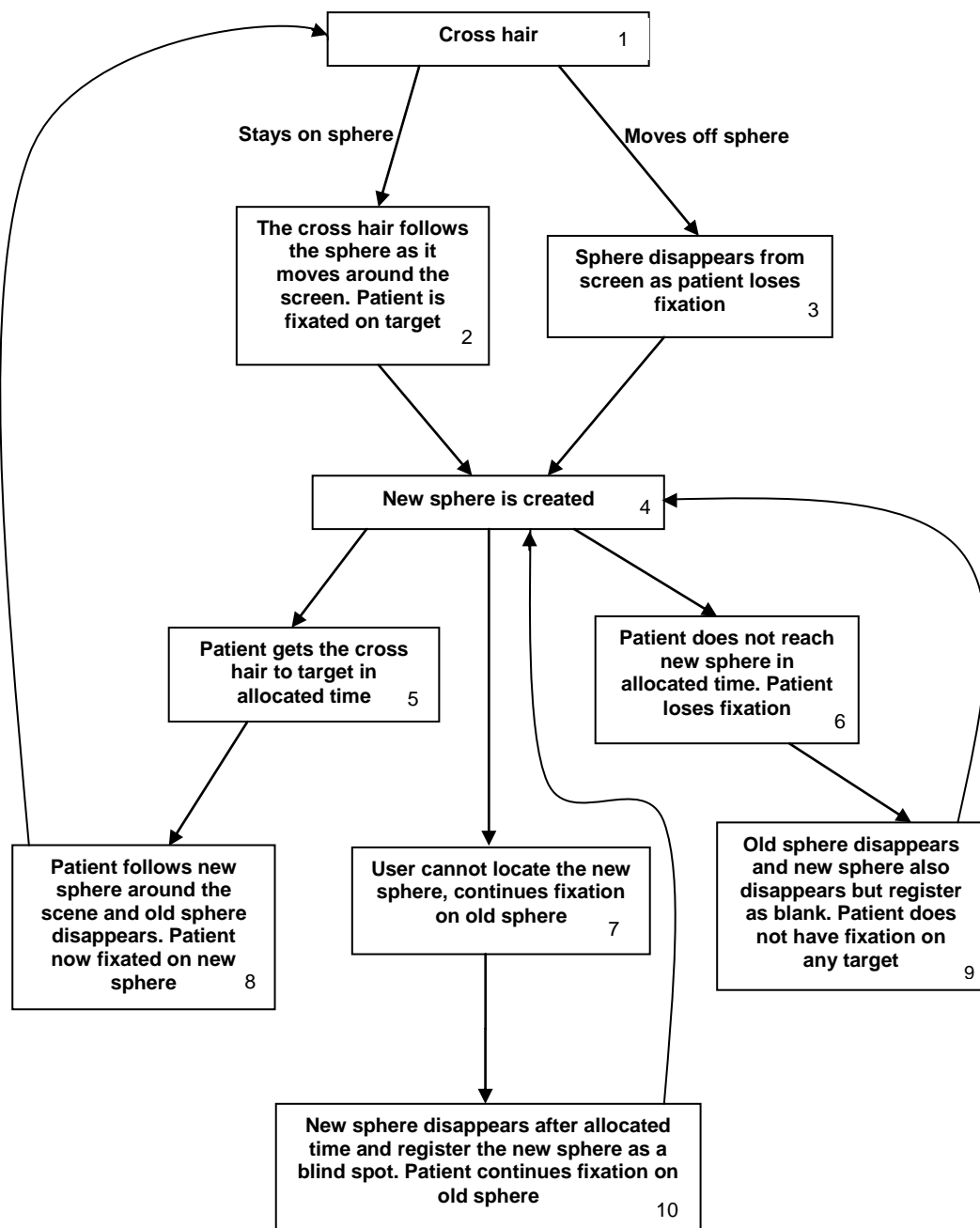
## **Game Development:**

A virtual sphere, a cross hair with a poker, and an invisible collision detection line at its centre, perpendicular to the cross hair was created in the Unity virtual scene. The virtual sphere was assigned a random rotation, so as to make the target more appealing and interesting to the patient. The random rotation script also gives the target a more lifelike feel and its speed can be controlled by the examiner, in the tumble parameter, to make it slower or faster, as per patient requirement. The script for random rotation is:

```
using UnityEngine;
using System.Collections;

public class RandomRotator : MonoBehaviour
{
    public float tumble;

    void Start ()
    {
        GetComponent<Rigidbody>().angularVelocity =
            Random.insideUnitSphere * tumble;
    }
}
```



*Figure 3: Block flow diagram outlining the logic behind the steps to developing the game to measure patient fixation. The number in each block is referred to in the methods for further detailed explanations.*

For the cross hair to remain fixated on the sphere it must follow the target sphere as it moves around the scene in random motion. The transform motion for the sphere is determined by the mover script. The motion of the sphere is determined by designating a maximum radius, movement velocity and the maximum angle for the sphere. The maximum radius controls the area in which the sphere can move around in. The movement velocity controls the maximum speed and the direction of the sphere, both of which can be random. The maximum angle is responsible for the angular deviation during a single video frame. It therefore controls how easily the sphere will be able to change its direction. The mover script controls the new movement target position of the sphere using the inside unit circle variable, with the radius is defined by the examiner. The movement velocity is also controlled in a similar manner, but instead using the movement velocity that will be defined, again by the examiner. Lastly, the rotate towards variable allows the sphere to be rotated towards the new movement target position at the new movement velocity. The importance for using a mover script on the target sphere is to ensure the fixation task does not become trivial for the patient and it forces the patient to fixate and be interactive with the game to remain fixated on the target sphere. Below is the mover script, with the explained variables.

```

using UnityEngine;
using System.Collections;

public class Mover : MonoBehaviour
{
    GameObject player;           // Reference to the player's position.

    public float movementRadius;
    public float movementVelocity;
    public float maxAngleRad;

    private Rigidbody rb;
    private float pz;

    void Start ()
    {
        // Set up the references.
        player = GameObject.FindGameObjectWithTag ("Player");
        rb = GetComponent<Rigidbody> ();
        rb.velocity = movementVelocity * Random.insideUnitCircle.normalized;

        //need to store this to avoid computational inaccuracies
        pz = player.transform.position.z;
    }

    void Update ()
    {
        Vector3 targetPosition = movementRadius * Random.insideUnitCircle;
        targetPosition.z = pz;

        Vector3 newVelocity = movementVelocity *
            (targetPosition -
             player.transform.position).normalized;
        rb.velocity = Vector3.RotateTowards
            (rb.velocity, newVelocity, maxAngleRad,
             movementVelocity);
    }
}

```

The patient is asked to maintain contact on the target sphere using the cross hair, and to continue to do so until a change is seen. This ensures that the patient is fixated on a target. The cross hair movement is controlled by the head tracking scripts. The cross hair will detect a fixation on the sphere by using collision detection between the poker and the target sphere. The poker also has a script, controller, associated with it, this script destroys the target, if the patient is no longer to fixate on it, i.e. no collision is detected between the cross hair and the target sphere. The controller script for the poker is:

```

using UnityEngine;
using System.Collections;

public class Controller : MonoBehaviour
{
    void OnTriggerExit(Collider other)
    {
        Destroy (other.gameObject);
    }
}

```

As the patient continues to be fixated on the target sphere a new sphere will appear in the scene. The new target sphere may also be generated if the patient loses fixation on the first sphere. In both cases the patient is asked to move the fixation from the old target to the new target sphere in an allocated amount of time. If however, the patient is unable to reach the new target sphere in the allocated time, it will disappear and register as an invalid data point; if the patient has moved off the first sphere to reach the new one. In both case, the first sphere will have disappeared, as the patient will no longer have been fixated on it, and no collision will have been detected. Another, possibility is when a new sphere is generated, and the patient does not see it and continues to follow the one they are currently in collision with, it will disappear after the allocated time, but register this spot as a bind spot in the patient's visual field. A game controller element has both, a script for new sphere generation and to destroy them, attached to it in the virtual environment. The script which controls the generation of new target sphere, spawning of new spheres, is the game controller script. The base script for the spawning of the sphere is a loop script with a 'for' and 'while' conditions applied to it. The spawning of new sphere is controlled by multiple variables, which are controlled by the examiner. The time between each new sphere being spawned is controlled by spawn wait time. Spawn

wait time is the break time between two waves of sphere. The stimuli count, is the total number of spheres to be spawned in a single loop, before the cycle repeats itself. In our case the number of spheres to be spawned in one loop is 1, and the spawn wait time is 5 seconds. Hence, a new sphere will span every 5 seconds. Another variable that is added into the spawning script is the start wait time. Start wait time is a set time, in our case 1 second, before the sphere being to spawn once the game starts, to allow the patient to get prepared. The loop script uses the 'for' conditions to spawn a new sphere, and it continues to do so, because of the 'while' condition. The whole loop is put in a co-routine so as to not pause the game as a whole, while there is a pause in the spheres being spawned due to the wait time. Also, each time a sphere is spawned it will be spawned from a new position within the range defined by the swan values. These values are controlled by the examiner and may be set to be outside the set movement radius or within. We used the range values of x at +/- 2, y at +/-2, and fixed z at 5. The value is fixed in the z direction so the sphere only moves in x-y plane and does not roll past the patient.

```

using UnityEngine;
using System.Collections;

public class GameController : MonoBehaviour
{
    public GameObject stimuli;
    public Vector3 spawnValues;
    public int stimuliCount;
    public float spawnWait;
    public float startWait;

    void Start ()
    {
        StartCoroutine (SpawnWaves ());
    }

    IEnumerator SpawnWaves ()
    {
        yield return new WaitForSeconds (startWait);
        while (true)
        {
            for (int i = 0; i < stimuliCount; i++)
            {
                Vector3 spawnPosition = new Vector3 (Random.Range
                    (- spawnValues.x, spawnValues.x),
                    Random.Range(- spawnValues.y,
                    spawnValues.y), spawnValues.z);
                Quaternion spawnRotation = Quaternion.identity;
                Instantiate (stimuli, spawnPosition,
spawnRotation);
                yield return new WaitForSeconds (spawnWait);
            }
        }
    }
}

```

Lastly, once the spheres have been spawned they need to also be destroyed over time, so the screen is not full of spheres. A Destroy by Time script is selected to solve this matter. The script uses 'if' condition to select which spheres will get destroyed and which will not. It also has a variable, lifetime i.e, the allocated time, which is controlled by the examiner. The examiner can select how long the sphere will be in the scene before it gets destroyed; if the first condition is not met. The condition that the sphere must meet, to not be destroyed, is that it must be in collision with the poker. If this condition is not met, and it also has been active in the scene for more than the determined lifetime, 5; the sphere will be destroyed.



```

using UnityEngine;
using System.Collections;

public class DestroyByTime : MonoBehaviour
{
    public float lifetime;
    public GameObject player;
    public GameObject poker;

    void Start()

    player = GameObject.FindGameObjectWithTag ("Player");
    poker = GameObject.FindGameObjectWithTag ("Poker");

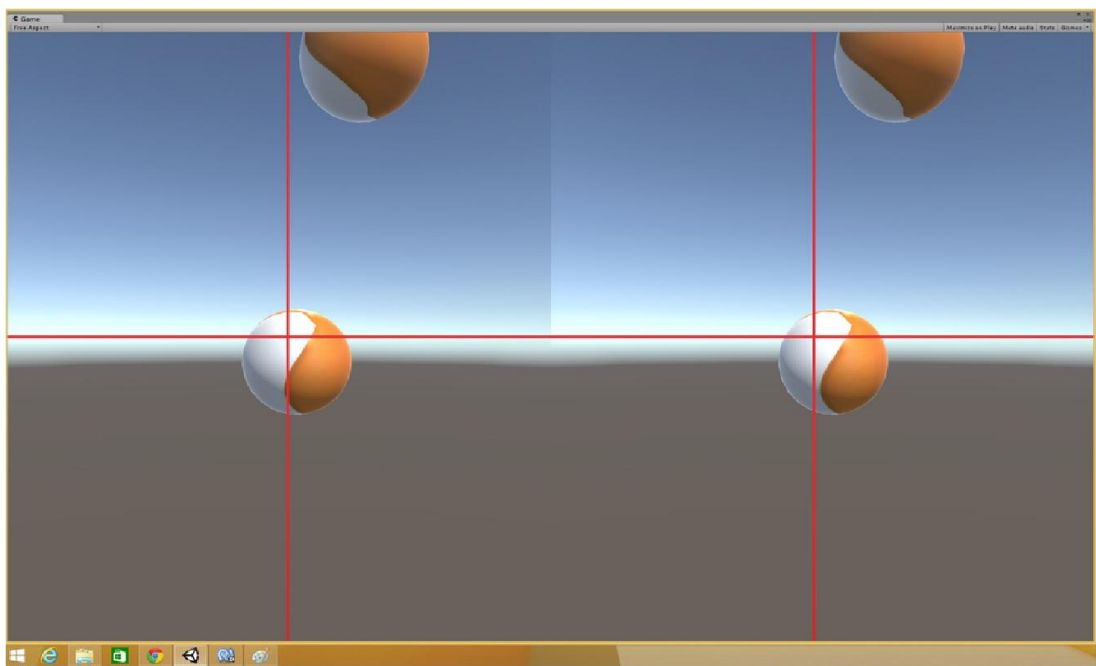
    {
        if (player.transform.position.z ==
            poker.transform.position.z)
        {
            return;
        }
        Destroy (player, lifetime);
    }
}

```

Fig 3, below, outlines and explains the logic that is used to create the game on Unity to be tested. The game was transferred to the smartphone, LG G3 via online cloud. The game was then installed onto the phone from the cloud and the phone was placed in the VR1 phone adaptor to be inserted into the goggles. The game was then tested by me to ensure all the transitions were working and the results were recorded.

## Results:

Fig 5 shows a game view scene on our game. The scene show a cross hair in collision targeting a sphere with another sphere spawned into the scene.



*Figure 5: Game View with cross hair, targeted sphere and a spawned sphere.*

The transitions between blocks are outlined in figure 3 were each tested individually and the results were recorded in table 1.

**Table 1: Transition Matrix, Outlining the Transition Outcomes:**

		From Block									
		1	2	3	4	5	6	7	8	9	10
To Block	1								Ok		
	2	Ok									
	3	Ok									
	4		Ok	Ok						Ok	Ok
	5				Ok						
	6				Ok						
	7				Ok						
	8					Ok					
	9							No			
	10								No		

*Table 1: The above table shows the results of the transitions between block that the game must accomplish. The numbers for the tasks correspond to the task block labelled in Figure 3, in the methodology. The grey boxes indicate if this transition from one task box to the other is not possible. The boxes labelled “Ok” indicate this transition worked. Impossible transitions are greyed out.*

All transitions, except two, were functional and allowed the game to proceed forward. The cross hair detected collision with the sphere in the scene to indicate fixation, seen in figure 4, and transitioned forward. However, the transitions for the spheres disappearing under no collision detection did not occur, and the sphere remained in the scene, as seen in figure 5. The spheres did disappear if a collision was initially made, but later not detected. The spheres continued to spawn at defined rate, and the game overall ran smoothly without any glitches.

Blink and close eyes test, is conducted to check if the sphere, in collision, remains in scene when the volunteer blinks. However, it is required to disappear if the volunteer's eyes are closed for an extensive period, longer than an average blink. The time used to test this condition was 2 seconds. Blink test was conducted on the working game to select the most reasonable parameters for the game to work with. The parameters outlined in table 2 below show the various values used and their outcomes. The values of the parameters were used to conduct the Blink and closed eyes test. Speeds of above 1 units/sec were too quick and yielded negative results. The results also as the angle is made smaller, showed that the sphere takes a long time turn and move to new position. Low angles also give the sphere a more shaky motion, not very smooth, in association with higher speeds. Desired results are obtained in tests 5 and 6.

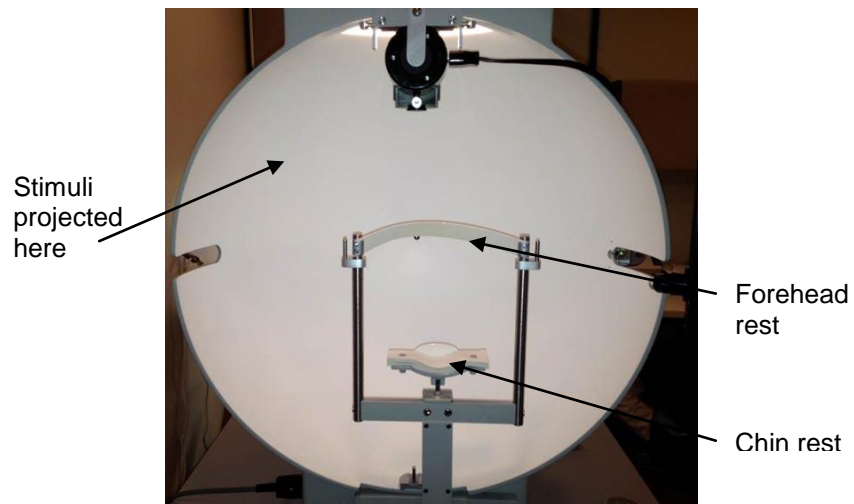
**Table 2: Blink Test Results for Various Parameters for movement Variables:**

Results	1	2	2-1	3	4	5	6	7
Tumble	5	5	5	5	5	5	5	5
Movement Radius	2	2	2	3	3	3	3	2
Movement Velocity	0.5	1	1	1	0.8	2	0.8	0.8
Maximum Angle	0.15	0.05	0.12-0.04	0.1	0.12	0.1	0.1	0.1
Blink Test	Pass	Fail	Fail	Fail	Pass	Fail	Pass	Pass
Closed Eyes Test	Fail	Fail	Fail	Pass	Pass	Pass	Pass	Pass

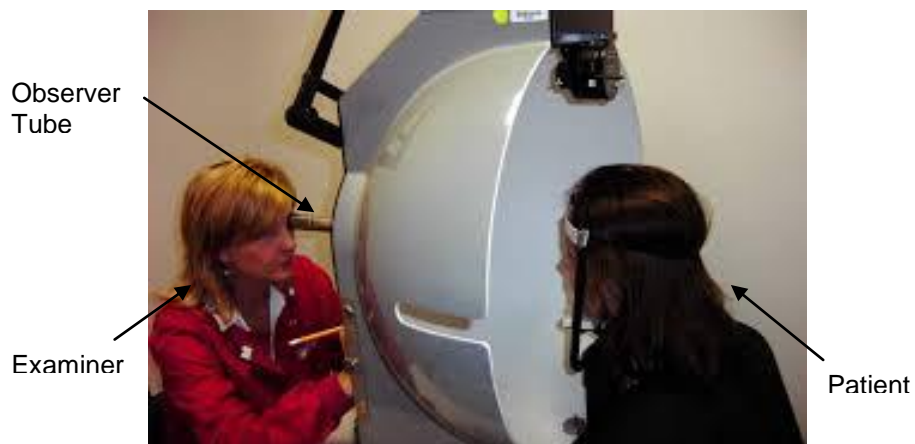
*Table 2: The above table shows the different values for the movement radius, velocity and maximum angle tested for the blink test. The blink test and the closed eyes test results are shown as well. **Pass**, indicates a positive test, and **Fail** indicates a negative test.*

## **Discussion:**

Traditional perimetry requires a great deal of input and assistance from both the examiner and patient. Perimetry testing is usually done at the hospital or at the ophthalmologist's office. The test may be performed by a computer system, known as automated perimetry or, manually by an examiner, which is the more common method. Test is conducted to one eye at a time, and requires the patient to remain fixated in the centre at all times. (Carroll & Johnson, 2013) The machine used to conduct perimetry testing is a large dome like structure, figure 5 and 7, with an eye hole, known as the observer's tube in the centre on one side of the dome. (Dersu, et al., 2006) Another, much bigger hole is on the opposite side for the patient's head. The patient sits opposite to the examiner and rests their head on the chin rest, which also has a forehead rest attached to it, so the patient does not lean into the dome. (Dersu, et al., 2006) The chin seat and the forehead can be adjusted by the examiner to ensure the eye of the patient is visible and cantered in the observer's tube. However, the resting seat is not comfortable, the head feels as though it is locked in position and can cause a great deal of fatigue to the patient.



*Figure 5:* Carroll, J.N and Johnson, C.A. (2013) Goldmann Perimeter [Textile]. Available at: <http://www.eyerounds.org/tutorials/VF-testing/index.htm> (Accessed: 12 August, 2015). (Carroll & Johnson, 2013).



*Figure 6:* Author's own image. Dersu et al. (2006) Positioning of Examiner and Patient for Goldmann Perimetry [Paper]. (Dersu, et al., 2006).

The observer's tube on the opposite side of the dome from the patient is used to check for patient fixation during the test. Hence it makes the fixation task a subjective matter, based on the examiner's interpretation. (Flanagan, 2009) However, high myopia or aphakia may be required to wear contact lenses. (Dersu, et al., 2006) The stimuli, which the patient is asked to respond to will be projected in the dome, by the examiner using a pantograph handle. Which has a light source attached to it on one end, and its movement

can be controlled by the examiner, which gets projected into the dome, who is sitting opposite to the patient. (Dersu, et al., 2006) The stimuli presented to the patient may be presented at different intensities, at different frequencies, and size and location. (National Research Council, 1994) The patient is required to acknowledge they see a stimulus by pressing a buzzer. (Choplin & Edwards, 1998) This manner of responding causes uncertainty and unreliability, especially in younger patients. This leads to a high degree of human errors and compromises the reliability of the test results. (Morales, Weitzman & Gonzalez de la Rosa, 2000) Another concern for traditional testing is the time required to complete the tests can lead to patient fatigue in head, neck and the eye(s) being examined. (Dersu, et al., 2006)

The lack of alternatives to traditional perimetry testing proves to be difficult when testing and collecting reliable and feasible data in children of a young age. (Patel, et al., 2015) The size and bulkiness of the instruments also make it difficult to be made portable to reach and test patients in isolated locations. (Wroblewski, et al., 2014) In light of all this, a new and different approach to testing perimetry is the gamification of the test. Over the recent years, video games and gaming have been studied as an alternative to traditional learning methods. (Squire, 2003) The number of hours, on average, a child spends playing video game is between 20–60 minutes per day, in children aged 11 and younger. (Christakis, 2004) This time however, increases to up to 7 hours a day in adults. (PricewaterhouseCoopers, 2012) In studies conducted by Green & Bavelier (2004, 2005) Subjects who played video games on a daily bases were better able to identify targets and objects



in the tests. (Green & Bavelier, 2004; Green & Bavelier, 2005) When presented with showing 'x' number of dots in a short time frame, the video game players were able to count more targets in the same time as the non-players, ratio being 5:3; conforming gaming increases the user's ability to engage and be attentive towards the game. (Green & Bavelier, 2003; Green & Bavelier, 2004) PwC (2012) also show that a majority of the users play games on their smartphones or online, rather than on a proper gaming console. (PricewaterhouseCoopers, 2012) The use of gamification in this project allows us to make a task which is more appealing and less tedious to the patient to conduct perimetry testing. During traditional testing an adult's ability to remain focused on the task, and not have a fatigue eye is approximately 10 minutes, effectively for children this is much less. (Dersu, et al., 2006) To perform the complete perimetry test it takes approximately 20-30 minutes. Hence, turning the test into a game will mean the patient will feel as though they are playing an interactive game, while actually undergoing the perimetry testing. Another benefit of video games is the increased spatial skills while gaming; children also have an increased awareness to multiple moving targets which is beneficial for us. (Green & Bavelier, 2005) The task of following a target around a scene, and responding when a new target is scene will challenge their ability to respond to multiple objects. Thus, gamification of the test on a mobile device or a smartphone will make this test more portable and more reachable to patients, who otherwise may not be able to. And it is also a very appealing approach to solve the engagement

issue with younger patients as it will no longer be a task for them, but more like a video games they play on their smartphone as a leisure activity.

One of the biggest breakthroughs to occur in gaming technologies is the introduction of virtual reality. Virtual reality is a technology which can be dated back to the 1962, when the first patent was issued for a virtual reality video arcade to Morton Heiling, however at the time he was unable to find investors and the technology was not developed further until the 1970s. (Burdea & Coiffet, 2003) In the 70s, the US Army invested a great deal of funding to develop this technology to be used a flight training simulation. At the time, the device was large and bulky, and only displayed simple scenes, at a rate of 1/20 of a second. It was able to display a simple scene of 200-400 polygons; however complex scenes require a greater number of polygon and this took a longer time to display the scene, effecting the animation and a decreased number of scenes. (Burdea & Coiffet, 2003) By, 1980s the military was testing with head mounted display with better software and with motion control. (Burdea & Coiffet, 2003) The first commercially available virtual reality product was the DataGloves, created by NASA, was sold in 1984, as was the first head mounted device. The price point, however, ranged from about \$70 000 - \$ 100 000 per component for a virtual reality workstation. (Burdea & Coiffet, 1994) Over the years, the advances in technology have made it possible to get an affordable, low cost computer with high performance and high resolution to work with virtual reality software. In the past 10 years the technology for virtual reality goggles has evolved from needing to be plugged into computers to goggles being able to function with

as small a screen as 4-5 inches; like that of a phone. Some of the earliest commercially available goggles to be implemented with phone screen are the Oculus Rift with its first prototype proposal in 2012, and releasing their final product in 2016. (Oculus, 2015) Google Cardboard, by Google USA, was one of the earliest and cheapest virtual reality goggles to be commercially available and to work on smartphones, in the market, in 2014. (Google, 2015) This was one of the earliest versions of cheap and affordable virtual reality goggles technology. Another improvement to enhance virtual reality goggles was a wider field of view like in the Samsung Gear VR, by Samsung, South Korea of 96° released back in 2014 and sits at £200. (Samsung, 2015) One of the more recently developed goggles, Zeiss VR One which is the one we are using for this project gives us a 100° field of view and costs only \$120 USD. (Zeiss, 2014)

Thanks to the low costs of virtual reality goggles, an alternative to traditional perimetry testing using virtual reality with head mounted goggles, VEye (VEye). It has goggles with a screen where the stimuli being projected into it via a computer, which it is attached to, running the program. The goggles use eye tracking technology to calculate fixation losses and false positives and false negatives. This is a great approach to compensate for the lack of fixation and false results incurred in testing children. However, VEye still requires bulky computer which does not solve portability issue. (Wroblewski, et al., 2014) Young children are still unlikely to give reliable results as VEye still requires the patient to be seated down and do the test in a tedious manner, simply with the head mounted goggles instead of

perimetry test machines, like the Octopus. (Haag-Streit Diagnostics, 2015)

Another concern with VEye is the cost of eye tracking device to measure the fixation of the patient. Thus, our alternative to eye tracking device to measure the fixation of the patient on the target is to use a target with some non-trivial movement with head tracking to ensure fixation. As the virtual reality goggle technology has evolved and uses such a wide field of view, an equally high quality screen should be used to yield the most optimal response. Hence, we have used LG G3, as this was one of the top phones available with the best screen resolution and processor power, when this project was started. The phone screen has a resolution has four individual HD screen of 1440 x 2560 pixels. (LG, 2015)

To create our task which will measure the fixation of the patient like in a perimetry test, but feels like a game to appeal to children. The game needs to have graphics as in any other video games. The graphics must be as well defined and move in as life like manner as possible, like the video games in the market today. To satisfy user expectation on the look and feel, these graphics and scenes must be developed in an environment which is used to build games. Hence, the game was created on Unity game development environment. The software is used mainly to create video games and gaming apps. Unity allows the user to create a wide range of games from simple tasks to complicated multi-level games. One major benefit of using Unity to create the games is that using a gaming environment, like such, allows us to ensure the final product, i.e. the game, will have a look and feel like any other game. Using a game development environment to build our game and we

must ensures us that software overhead that may be present will not interfere with the functionality of the test. Unity also allows creating a game for multiple different platforms. Meaning, the game can be created to work on phone, or gaming console system, or simply to be played on the computer. In our case, the game was created to be used on a smartphone on an Android platform to be added into a phone adaptor for the VR1 goggles. Unity also allows the user to create a game for virtual reality environment with built in virtual reality features and head tracking abilities.

### **Examining Results:**

The functionality of the transitions is an important factor in determining if the game is being hindered or affected by computer overheads, due to the graphics, or the accelerometer. We want to ensure that these overheads do not interfere with the transition flow. The current results, as seen in table 1, indicate that the transition protocol flows correctly. The game created, is merely the initial step toward this ultimate goal of testing perimetry via gamification. The current game, allows the patient to remain fixated on a target without needing massive complex machinery or an expensive device and nor an examiner presence. The fixation task is a non trivial task, of having a collision being detected between the cross hair and the target sphere, as it moves around the scene in random motion. The patient is said to be fixate if he/she is able to follow the moving target and maintain the collision and also, change to fixate on a new target, if it is seen. In doing this, the old target will effectively disappears, due to the loss of fixation on it by the patient i.e. no collision detection. Each transition on the game was tested on

a volunteer, and anecdotal evidences were collected and highlighted in tables 1. The volunteer was easily able to intersect the cross hair with the target sphere to detect a collision and follow the sphere in its random path with the cross hair to create a fixation. The positive results for collision detection and transitions of collision detection also indicate a proper functioning head tracking in real time, with no lag.

The parameters of the random motion such as the ideal velocity, maximum radius and maximum angle were determined by conducting a blink and closed eyes test (discussed later on). The sphere disappears from the screen if the subject lost fixation i.e. collision detection between sphere and cross hair. As a new sphere spawns and appears on the screen, if the subject moves away from the current target sphere to go towards the new sphere, and manages to collide with the new sphere, in the allocated time, a new fixation target is created and the old sphere disappears, due to lack of fixation. However, as seen in the results the transition between tasks 6 to 9, and 7 to 10 did not go through as expected. These transitions are suppose to make the new spawned sphere disappear, after the allocated time, if the volunteer was unable to reach the new sphere in time, or did not see it. Due to the time constrain, I did not have time to examine this code further and test other possibilities and check for bugs. These need to be re-examined, checked for errors, and alterations need to be made to it to ensure the transitions go through, and need to be further tested.

The parameters for the mover script were determined by conducting a blink and closed eyes test on the game. For the fixation to be considered,

the parameters of the game need to ensure that there is no fixation loss due to average blinking of the eyes. However, if the volunteer closed their eyes for an extended period of time a loss of fixation leading to the sphere disappearing. The game was tested for various combinations of parameter values to test this. The results for these tests are shown in table 2. The game also did not induce fatigue and neck strains, which were some of the concerns in traditional testing. The results from the blink test indicated that a speed above 1unit/second was too quick and gave a false blink test. It was also seen that the maximum angle should be decreased as the speed is increased, otherwise the sphere looks as if it is jittery and there is some glitch in the game. However, making the angle too small with a high speed also yields a negative result for the test, as the sphere takes a long time to turn, and head towards the new position. Parameters in tests 2, 2-1, 3, and 5 all gave a negative result to the blink test. The sphere disappeared when I blinked; hence the speed of the sphere was too fast. The angle values in tests 2, 2-1, and 4 were too large with the associated speed; this caused the sphere to be very jittery. The angles tested in test 2 and 2-1 showed to take a long time as the angle decreased. The ideal values for the maximum velocity, maximum radius and maximum angle were determined to be 0.8, 2, and 0.1, respectively, test 7. A maximum radius of 3, in test 6, along with the other parameters being constant, also gave positive test results however; the radius is kept small so the patient does not move around too far from one end to the next and to not fatigue the patient. These parameters were

determine to be ideal after testing on a single volunteer and it did not lead to any fatigue to the patient.

The parameters determined to be the ideal values for this project needs to be tested further. Due to the number of subjects and the lack of time the project was tested on a single subject, for the ideal parameters and the affect of the test on patient fatigue. To conclude more confident conclusion on these, further studies need to be conducted on multiple subjects to check for patient fatigue and parameter statistics. Lastly the transition logic being used needs to clearly define a clinical protocol for the gaming of perimetry testing. The protocol need to be clinically tested to examine the transitions and the varying parameters.

### **Conclusion:**

The initial task of getting the building blocks for a game in virtual reality environment with collision detection and non-trivial transitions to determine fixation have been built and tested, with further examination needed to correct the two transitions which do not currently work. A non-trivial task of target collision is implemented to detect fixation of target, build on a gaming development environment. The Unity ensures the game runs smoothly and the overheads do not affect the running of the game. The parameters determined during this project are arbitrary and needs to be examined further to ensure they are viable values or not. The results collected in this project are anecdotal evidence, and further examination of the scripts and the transition needs to be conducted to define a definite protocol to test perimetry.



## **Works Cited**

Apple, 2015. *iPhone6*. [Online]

Available at: <https://www.apple.com/uk/iphone-6/>

[Accessed 08 08 2015].

Burdea, G. & Coiffet, P., 2003. *Virtual Reality Technology*. Second Edition ed. Hoboken: John Wiley & Sons, Inc..

Carroll, J. & Johnson, C., 2013. *Visual Field Testing: From One Medical Student to Another*. [Online]

Available at: <http://www.eyerounds.org/tutorials/VF-testing/index.htm>

[Accessed 12 08 2015].

Choplin, N. & Edwards, R., 1998. Section I: Introduction to Visual Fields. In: A. Drummond, ed. *Visual Fields*. Thorofar: SLACK, pp. 6-8.

Christakis, D. E. B. R. F. & Z. F., 2004. Television, Video and Computer Game Usage in Children Under 11 Years of Age. *The Journal of Pediatrics*, 145(5), p. A2.

Cohen, J. G. C. & B. D., 2008. Training Visual Attention with Video. In: H. & P. R. O'Neil, ed. *Computer Games and Team and Individual Learning*. Oxford: Elsevier, pp. 205-228.

Damato, B., 1985. Oculokinetic Perimetry: A Simple Visual Field Test for use in The Community. *British Journal of Ophthalmology*, Volume 69, pp. 927-931.

Dersu, I. et al., 2006. Understanding Visual Fields, Part I; Goldmann Perimetry. *Journal of Ophthalmic Medical Technology*, 2(2).

Flanagan, J., 2009. Visual Field Examination. In: *Optometry: Science, Techniques and Clinical Management*. Edinburgh: Butterworth Heinemann Elsevier, pp. 317-342.

Flanagan, J., 2009. Visual Field Examination. In: M. Rosenfield, N. Logan & K. Edwards, eds. *Optometry: Science, Techniques and Clinical Management*. Second Edition ed. Edinburgh: Elsevier, pp. 314-334.

Gault, J., 2007. Visual Fields. In: *Ophthalmology: Secrets*. Philadelphia: Mosby Elsevier, pp. 59-76.

- Google, 2014. *Android 5.0 Lollipop*. [Online]  
Available at: <https://www.android.com/versions/lollipop-5-0/>  
[Accessed 08 08 2015].
- Google, 2015. *Android Studio*. [Online]  
Available at: <https://developer.android.com/sdk/index.html>  
[Accessed 09 08 2015].
- Google, 2015. *Google Cardboard*. [Online]  
Available at: <https://www.google.com/get/cardboard/>  
[Accessed 11 08 2015].
- Green, C. & Bavelier, D., 2003. Action Video Game Modifies Visual Selective Attention. *Nature*, Volume 534-537, p. 423.
- Green, C. & Bavelier, D., 2004. Does Action Video Game Play Really Enhance the Number of Items that can be Simultaneously Attended?. *Journal of Vision*, Volume 4, p. 632.
- Green, C. & Bavelier, D., 2005. Enumeration Versus Multiple Object Tracking: The Case of Action Video Game Players.. *Cognition*, Volume 101, pp. 217-245.
- Green, C. & Bavelier, D., 2012. Learning, Attentional Control, and Action Video Games : Review. *Current Biology*, 22(6), pp. 197-206.
- Greve, E., 1973. *Single and Multiple Stimulus Static Perimetry in Glaucoma; The Two Phase of Visual Field Examination*. Dordrecht: Springer Science + Business Media .
- Haag-Streit Diagnostics, 2015. *Octopus 900*. [Online]  
[Accessed 12 08 2015].
- Hasegawa, T., Koshino, M. & Ban, H., 2015. An English Vocabulary Learning Support System for the Learner's Sustainable Motivation. *Springer Plus*, 4(99).
- King, D., Greaves, F., Exeter, C. & Darzi, A., 2013. 'Gamification': Influencing Health Behaviours with Games. *Journal of the Royal Society of Medicine*, Volume 106, pp. 76-78.
- LG, 2015. *LG G3 Features*. [Online]  
Available at: <http://www.lg.com/uk/lg-g3/index>  
[Accessed 08 08 2015].

Morales, J., Weitzman, M. & Gonzalez de la Rosa, M., 2000. Comparison Between Tendency-Oriented Perimetry (TOP) and Octopus Threshold Perimetry. *American Academy of Ophthalmology*, 107(1), pp. 134-142.

Murray, I. et al., 2013. *Saccadic Vector Optokinetic Peimetry (SVOP): A Novel Technique for Automated Static Perimetry in Children Using Eye Tracking*. Osaka, IEEE, pp. 3186-3189.

National Research Council, 1994. *Measurement of Visual Field and Visual Acuity for Disability Determination*, Washington: National Academy Press.

Oculus, 2015. *Rift: Next-Generation Virtual Reality*. [Online] Available at: <https://www.oculus.com/en-us/rift/> [Accessed 11 08 2015].

Oracle, 2015. *Java SE Development Kit 8 Downloads*. [Online] Available at: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> [Accessed 09 08 2015].

Patel, D. et al., 2015. Study of Optimal Perimetric Testing In Children (OPTIC): Normative Visual Field Values in Children. *Ophthalmology*, 122(8), pp. 1711-1717.

Pel. J. J. M., B. M. C. M. T. G. & S. J., 2013. Validity and Repeatability of Saccadic Response Times Across the Visual Field in Eye Movement Perimetry. *Translational Vision Science & Technology*, 2(7), p. 3.

Pineles, S. et al., 2006. Automated Combined Kinetic and Static Perimetry. *Department of Ophthalmology Schele Eye Insitute, University of Pennsylvania School of Medicine*, Volume 124, pp. 363-369.

PricewaterhouseCoopers, 2012. *The Evolution of Video Gaming and Content Consumption*, Delaware: PwC.

Samsung, 2015. *Samsung Gear VR*. [Online] Available at: [http://www.samsung.com/global/microsite/gearvr/gearvr\\_features.html](http://www.samsung.com/global/microsite/gearvr/gearvr_features.html) [Accessed 11 08 2015].

Samsung, 2015. *Why Wait to be Ahead of the Curve: Samsung Galaxy S6 | S6 edge*. [Online] Available at: <http://www.samsung.com/uk/galaxys6/> [Accessed 08 08 2015].

Squire, K., 2003. Video Games in Education. *International Journal of Intelligent Simulations and Gaming*, 2(1), pp. 49-62.

Tschopp, C. et al., 1995. Automated Static Perimetry in the Child: Methodologic and Practical Problems. *Klinische Monatsblätter für Augenheilkunde*, 206(5), pp. 416-419.

Unity, 2015. *The Best Development Platform for Creating Games*. [Online] Available at: <https://unity3d.com/unity> [Accessed 09 08 2015].

Wroblewski, D. et al., 2014. Testing of Visual Field with Virtual Reality Goggles in Manual and Visual Grasp Modes. *BioMed Research International*.

Zeiss, 2014. *VR One*. [Online] Available at: <http://zeissvrone.tumblr.com/> [Accessed 08 08 2015].

Zeiss, 2014. *Zeiss VR ONE*. [Online] Available at: [http://40.media.tumblr.com/513a4dc22db2f76eee2dce5223050e4c/tumblr\\_nk0hg2Er0z1u0v4y2o1\\_1280.jpg](http://40.media.tumblr.com/513a4dc22db2f76eee2dce5223050e4c/tumblr_nk0hg2Er0z1u0v4y2o1_1280.jpg) [Accessed 08 08 2015].