The Moon Illusion in Virtual Reality

Author: **Elef Schellen**

Supervisor: **Jacob Jolij**

Secondary Supervisor: **Mark Nieuwenstein**

22-08-2016

Abstract

Despite the moon illusion being one of the oldest and most studied visual illusions, new insights into its workings can yet be obtained using novel virtual reality technology. Using a virtual reality head mounted display to allow subjects to view immersive virtual environments, the present study aims to elucidate on the interaction between vestibular and visual mechanisms which together give rise to the moon illusion. Results show a contribution of depth cue salience similar to that described in the literature, but an effect of head position that runs counter to the one found by Wood, Zinkus & Mountjoy (1968). Additionally, an effect of stimulus size was found on illusion magnitude, indicating a possible mnemonic component of the moon illusion.

Introduction

Given the tremendous amount of studies conducted concerning the moon illusion over the past 2000 or so years, this thesis will not provide anywhere near a comprehensive report of the research done on the moon illusion throughout history. More comprehensive reviews of moon illusion literature can be found in the book by Ross and Plug (2002) or the more concise article by Nanavati (2009).

The underlying mechanisms that give rise to the moon illusion can be broadly categorized as either structural or strategic (Coren & Girgus, 1978). Structural mechanisms refer to causes that are physiological such as Holway and Boring's (1940) angle of regard hypothesis or Wood, Zinkus and Mountjoy's (1968) vestibular hypothesis. Strategy mechanisms, on the other hand, relate to the way sensory information is processed and, despite what he name may suggest, include both conscious and unconscious cognitive factors. Despite the fact that many of these mechanisms have been shown to indeed underlie the moon illusion, they are usually studied in isolation. An illustrative example of this comes from an experiment in Kaufman & Rock (1962). Here, an experiment was conducted in a completely blacked out room, with subjects reporting the size of projected circles of light at differing angles. An illusion with a small magnitude was found i.e., when asked to point out a reference stimulus that was of equal size to the projected circle of light, subjects picked larger stimuli when the circle was projected near the horizon than when it was projected above the subjects head. Regardless of this effect's statistical significance, Kaufman & Rock dismissed this finding as being due to some artefact, reasoning this effect is too small to account for the moon illusion on its own and is therefore not of any importance.

This lack of studies investigating the relative contributions of different causes of the moon illusion is in part due to methodological constraints. In investigations of the role of depth cues, for instance, different scenes can be presented on paper or on a computer screen (Coren & Aks, 1990; Jones & Wilson, 2009). Using these 2D methods allows for sequential presentation of scenes with widely differencing depth cue salience, but it has its drawbacks in that structural factors like head position cannot be easily studied. Conversely, methods used to study structural factors often don't allow for investigation of strategy mechanisms. Usually, studies into structural factors use elaborate contraptions such as a set of tubes through which a stimuli can be seen at different angles (Carter, 1977) or a large swing on which a subject sits and reports the size of a stimulus seen at the moment when the swing reaches its apex (van Eyl, 1972).

Alternatively, it has been shown that looking at the moon by bending over forwards and looking at it through one's legs reduces the moon illusion. This was first hailed as support for vestibular contributions, but as was later pointed out, looking at a scene upside down also greatly reduces depth cue salience (Coren, 1992). Again, this paradigm suffers from the fact that both depth cue salience and vestibular factors are involved in the effect, and there is no way to accurately and reliably disentangle these factors.

The goal of this project is to perform a single experiment which quantitatively investigates the main contributing factors of the moon illusion in a within-subject design. To this end, a virtual reality device will be used to present subjects with a range of virtual environments. A virtual reality (VR) head mounted display (HMD) allows for the presentation of visual environments that greatly vary in depth cue salience, while still allowing the subject to move their head freely. Additionally, because stimulus presentation and response submission will both take place in this virtual environment, data is accurately quantified. By having subjects perform

in conditions that are aimed at investigating just structural mechanisms, just strategy mechanisms and finally both, the relative contributions of each mechanism to the overall moon illusion can be delineated.

This experiment will consider two factors that are thought to contribute to the moon illusion. Namely: depth cue salience and vestibular input.

Depth cue salience refers to the degree in which visual indications of depth are present in a visual scene. Depth cue salience has been shown to influence moon illusion magnitude in pictures, a method that excludes structural effects (Coren & Aks, 1990; Jones & Wilson, 2009; Redding, 2002). In scenes with higher depth cue salience, the overestimation of the horizon moon size is generally more pronounced. The zenith moon is generally not as affected by depth cue salience as the horizon moon. Although disputed, the most common explanation for this phenomenon involves size-distance constancy (Ross & Plug, 2002). High depth cue salience in a scene emphasises the distance between the observer and the moon, making it appear further away. Conversely, these cues are not as visible when looking up to view the zenith moon, leading to a distance estimation unaided by any sort of depth cue. This estimated distance is shorter than the one to the horizon moon, and given that the actual angular size of the moon at any position in the sky is practically equal, the principle of size-constancy causes the horizon moon to look larger. After all, size-constancy dictates that, given two items of equal angular size, the item further away from the observer will appear to be larger. The notion that perceived distance to the zenith moon is smaller than the perceived distance to the horizon moon has been first proposed in the 11th century *Book of Optics* by Alhazen. In his works, Alhazen describes how the sky is generally perceived as a flattened dome, with the distance to the horizon being larger than the distance “to the sky” directly above the observer.

The size-constancy explanation is challenged by the fact that people don't actually report the zenith moon to be further away than the horizon moon. In fact, the opposite is true. This leads to the so called further-larger-nearer hypothesis. This seemingly paradoxical phenomenon can be explained by assuming that there are two distance judgements; one conscious and one unconscious (Claparède, 1906). The distance judgement that leads the size-constancy principle to activate is unconscious, leading to an enlarged percept. This enlarged percept is then consciously, and rather logically, reported as being closer. Recent neuropsychological findings support the plausibility of such a mechanism. Weidner et al. (2014) found the left V3v area to be involved in combining angular size and distance cues in the context of the moon illusion. Since the area V3v combines information from the ventral stream of visual processing, and given that visual illusions have been shown to affect the earliest neuronal part of this stream; the primary visual cortex (Murray, Boyaci & Kersten, 2006), it is possible that the conscious percept on which a subject's distance judgement is based arises after the unconscious combination of information that is already affected by the illusion.

The second contributing factor that will be investigated here is vestibular input. Wood, Zinkus & Mountjoy (1968) were the first to knowingly perform an experiment into the vestibular components of the moon illusion, and found the apparent increase of perceived size for horizontally viewed stimuli compared to those viewed at an upward angle. Wood et al. (1968) only vaguely alluded to the physiological underpinnings of this effect, but did not suggest any actual mechanisms in which vestibular information could affect visual perception.

More recent studies have looked into the interaction between vestibular input and visual perception. For instance, direct manipulation of the vestibular organs through galvanic vestibular stimulation (GVS) in healthy subjects influences responses on a line bisection task (Ferrè,

Longo, Fiori and Haggard, 2013). Depending on the direction of the current used in GVS, subjects either showed a leftward or rightward bias in their perception of the middle of a line.

To further determine the effects of the vestibular organs on visual perception, it is worthwhile to differentiate between two parts of the vestibular organs: the semicircular canals and the otolith organs. These two components have differentiated functions, with the semicircular canals providing information about angular motion and the otolith organs providing information about linear motion (Goldberg et al., 2012). The otolith organs also provide information on the direction of gravity. This makes it so that even without any other information, one can determine (with varying levels of accuracy) their position relative to the direction of gravity.

This information has been shown to influence perception of certain visual illusion. Patients with damaged otolith organs, for instance, are less susceptible to the illusory effects of certain visual illusions than healthy controls (Clément, Fraysse & Dequine, 2009). The illusions that were affected were geometric, such as the Ponzo illusion and the inverted T illusion. The influence of otolithic information on the perception of these illusions has been confirmed using healthy participants in different bodily positions (Clément & Eckardt, 2005) as well as astronauts in long-duration spaceflight (Clément, Skinner, Richard & Lathan, 2012). The astronauts, similar to the patient group in Clément et al. (2009), showed a decreased susceptibility to certain geometric illusory effects.

Another finding that illustrates the role of otolithic input regarding perception comes from subjects estimating the dimensions of shapes. In healthy subjects, the vertical sides of a square are perceived as longer than their horizontal counterparts. This effect was absent and reversed in astronauts living in 0g and patients with otolithic dysfunction, respectively (Clément

et al, 2012, Clément et al., 2009). These findings suggest that information from the otolithic organs influences size and/or distance judgements.

Harris and Mander (2014) studied the effect of body position or perceived body position on depth perception. For example, certain conditions had subjects lying in a supine position as well as standing up in a tilted room, resulting in a situation where visual information suggests a standing position, but vestibular and proprioceptive information suggests a supine position. Subjects that either were, or believed they were in a supine position, perceived a longer line, projected on the far wall of a room, to be equal to a reference line, compared to subjects who were in an upright position. This suggests that the target stimulus was perceived as closer due to the fact that an object of a constant physical size (note that the task was to match the size of a stimulus line to a reference line of constant physical size) taking up a larger visual angle must be closer than this object when taking up a smaller visual angle. Although these findings might appear to run counter to the traditional moon illusion effects, this is not necessarily the case. These findings are in accordance with the “flattened dome” explanation of the moon illusion, since vertical distances are perceived as shorter.

Another interesting finding made by Harris and Mander (2014) concerned the relationship between binocular vision and depth perception. At shorter distances (122cm), binocular viewing of the stimulus negated any effect of real or perceived bodily orientation, whereas effects were present under monocular viewing conditions. At a longer distance (366 cm), the distinction between monocular and binocular viewing conditions is still present, but less pronounced. It therefore appears that binocular depth cues can override information from the vestibular senses up to a certain point. At longer distances, and especially at optical infinity as occurs when viewing the moon, binocular cues will only provide a negligible amount of

information. This supports the idea that otolith input is partly responsible for the effect seen in the moon illusion.

Methods

Participants

20 participants, of whom 11 were female, took part in the study. All participants had normal vision. Participants were recruited online, and given a monetary compensation for their participation.

Materials

The task, as experienced by the participant, took place in a virtual environment. The stimuli consisted of a series of globes of different sizes and colours. Some stimuli also contained a bump map, meaning that the surface of the object had a certain 3D texture. Sizes of these stimuli ranged from 0.67 degrees visual angle to 2.92 degrees visual angle.

There were three virtual environments in which participants performed their tasks. One was an empty, grey space. It contained no depth cues at all. The second environment consisted of a scene with abstract depth cues. The ground had a grid texture and had a different colour than the sky, providing a horizon line, linear perspective and texture gradient. The environment contained multiple rectangular objects, located throughout the scene, providing the participant with depth cues such as interposition, shadows and, to the extent that subjects moved their heads, motion parallax. The third environment consisted of a realistic scene, with the same number of different depth cues as the abstract scene. The main difference between the abstract and the realistic scenes is that in the realistic scene, the stimulus was textured so as to resemble the moon. As a reference stimulus, all response screens contained an untextured, light gray globe which could be adjusted to match the size of the previously seen stimulus. The abstract and realistic environments were presented twice each; once with stimuli appearing from 0° to 90°

elevation and allowing subjects to look directly at the stimuli (movement conditions), and once with stimuli appearing between 0° and 35°elevation and having subjects keep their head level with the horizon. With one's head kept horizontal, stimuli at elevations of up to 35° can be seen by just moving one's eyes, hence this was chosen as the upper elevation limit in the no movement condition.

The environments were created and presented using the Unity engine (Unity®, 2016). The scripts (in C#) needed to run the experiment and store data into an external file are included in appendix A. The device used to present the environments was an Oculus Rift, Development Kit 2. The Oculus Rift contains a monitor for each eye, each with a resolution of 960 by 1080 pixels, and offers a field of view of 100º.

To separate the effects of head orientation from those of depth cue salience, both the abstract scene and the realistic scene were used in two conditions. In one condition the participants were instructed to keep their head level, only looking up at the moon by moving their eyes (no movement condition). The other condition had participants move their heads to directly look at the stimulus (movement condition). Head tilt was measured using the HMD's internal gyroscope.

Conditions were made up of 10 blocks consisting of 20 trials each, resulting in 200 trials per condition, and 1000 trials in the entire experiment.

Procedure

Participants read and signed an informed consent form, and were verbally instructed on the nature of the task. Participants were seated at a desk and wore the virtual reality head mounted display (HMD) while performing the task. In all conditions, the basic structure of a trial

was identical. A stimulus of a size randomly selected from a predetermined range of sizes would appear in front of the participant at different elevations. There was no time limit; subjects could look at the stimulus as long as was necessary for them to form a judgment concerning its size. After pressing a button, the reference stimulus, also of random size, would appear at an elevation of 0º and subjects would be prompted to adjust the reference stimulus so that it would be equal in size to the previously seen target stimulus. The method of stimulus reproduction was chosen because it has been shown to produce the largest illusory effects (Coren & Girgus, 1972). In conditions with any amount of depth cues, the reference stimulus would appear on an overlay which obscured vision of the scene, so as to not have its perceived size be influenced by the scene's depth cues. Participants could take the HMD off during breaks between the blocks.

Results

Data from all 20 participants was used in the analysis. One participant did not manage to complete the task in time, and just the completed conditions of this participant were included in analysis. After removing outliers on illusion magnitude using the 1,5 times interquartile distance criterion, 18.252 trials remained on which analysis was performed. In the no movement condition, multiple comparisons of means showed significant differences in average size estimation for stimuli presented at different elevations. Figures 1 and 2 show illusion magnitudes in the no movement condition for the abstract and realistic environments respectively.

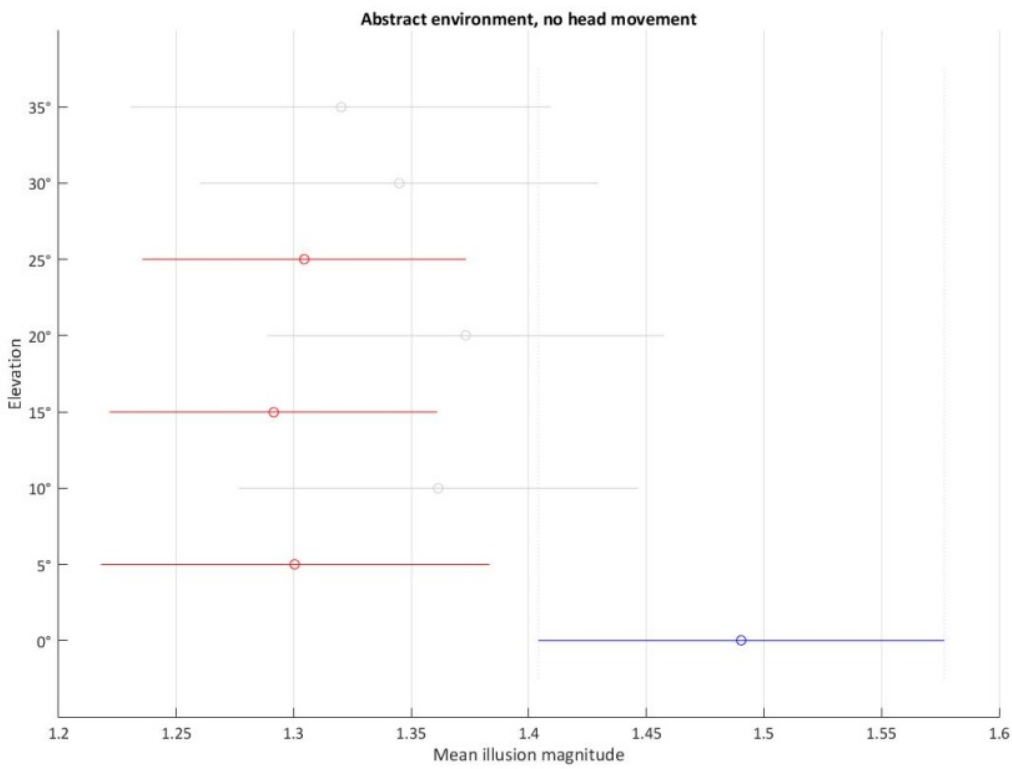

Figure 1: illusion magnitude means and 95% confidence intervals (x-axis) for the no movement condition in the abstract cue environment per target stimulus elevation (y-axis).Bars in red differ significantly from the bar in blue.

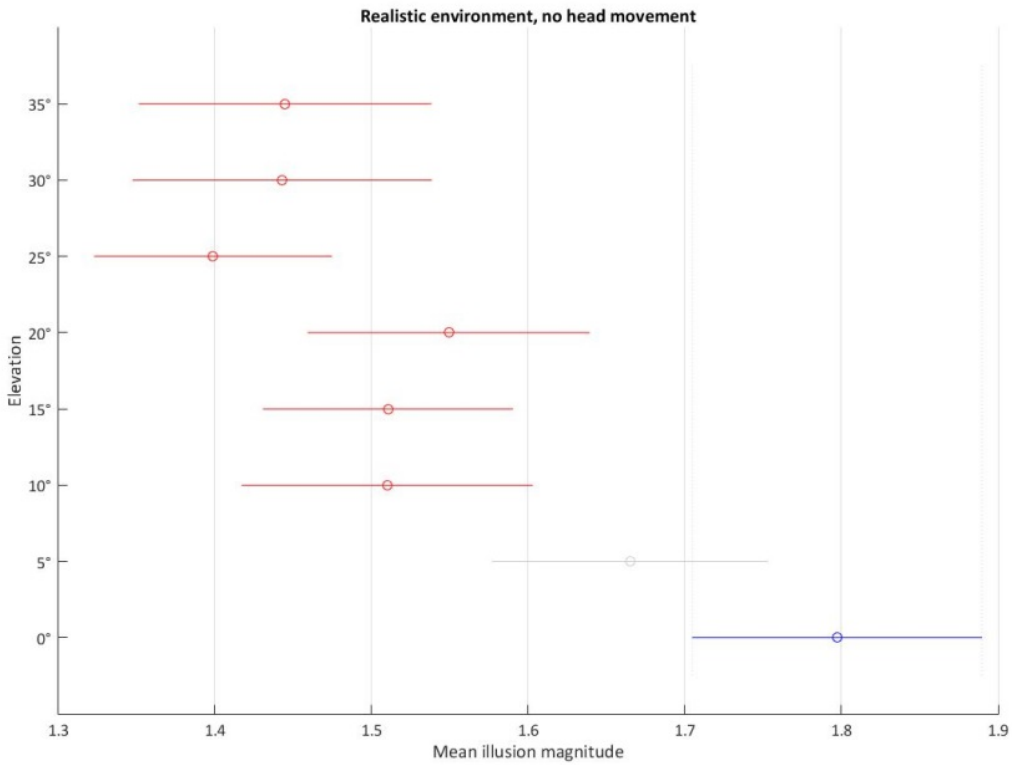

Figure 2: illusion magnitude means and 95% confidence intervals (x-axis) for the no movement condition in the realistic cue environment per target stimulus elevation (y-axis).Bars in red differ significantly from the bar in blue.

In the condition where the environment was devoid of any depth cues (blank condition), multiple regression analysis with head rotation as linear and squared terms revealed significant predictive value of headrotation (*p=0.023)*. Figure 3 shows a scatterplot with illusion magnitude and head rotation when subjects were looking at the stimulus on its axes. The cubic fit line is closest to 1 near an elevation of 0°, and rises at high and low head rotations.

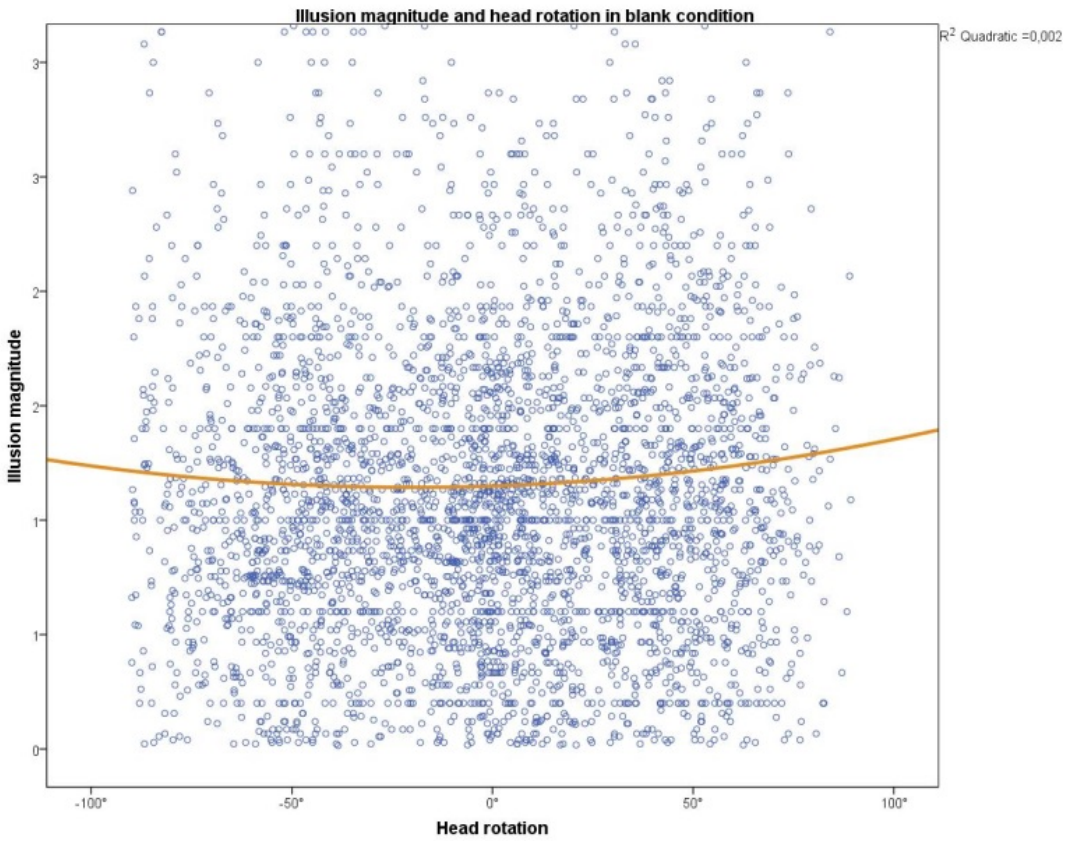

Figure 3: head rotation in degrees is shown on the x-axis, and illusion magnitude is shown on the y-axis. Each dot represents an individual trial.

For the movement condition data, linear mixed effects models were fitted in order to investigate the relative contributions of head rotation and stimulus elevation. Tables 1 and 2 show the estimates and significance levels for stimulus elevation, head rotation and the interaction between these two factors, as well as the starting size of the reference stimulus, and the target stimulus size for the abstract and realistic environment, respectively.

Name	Estimate	P Value
Fixed effects		
Intercept	0.65879	P<0.001
Stimulus elevation	-3.73 e-03	P<0.001
Reference stimulus starting size	0.27	P<0.001
Head rotation (quadratic term)	1.18e-04	0.035
Target stimulus size	-.045	P<0.001
Rotation*Elevation	-8.83 e-07	0.11
Random effects		
Subject	.3147	

Table 1. Coefficients and p values of the linear mixed effects model for the abstract environment. $R^2_{adj}=0.418$.

Name	Estimate	P Value
Fixed effects		
Intercept	0.74719	P<0.001
Stimulus elevation	-3.96 e-03	P<0.001
Reference stimulus starting size	0.25	P<0.001
Head rotation (quadratic term)	6.82e-05	0.15
Target stimulus size	-.070	P<0.001
Rotation*Elevation	9.92e-07	0.04
Random effects		
Subject	.2086	

Table 2. Coefficients and p values of the linear mixed effects model for the realistic environment. $R^2_{adj}=0.404$.

In both the abstract and realistic environments, stimulus elevation, reference stimulus starting size and target stimulus size are significant contributors to illusion magnitude. The interaction effect between head rotation and stimulus elevation is only significant in the realistic

environment, but only has a minute estimated contribution. Head rotation, a factor seen to be a significant predictor in the blank environment, is only significant in the abstract environment.

An unexpected contributor to illusion magnitude was target stimulus size. Closer analysis of the data reveals that the effect is mainly caused by the smallest target stimulus (figure 4).

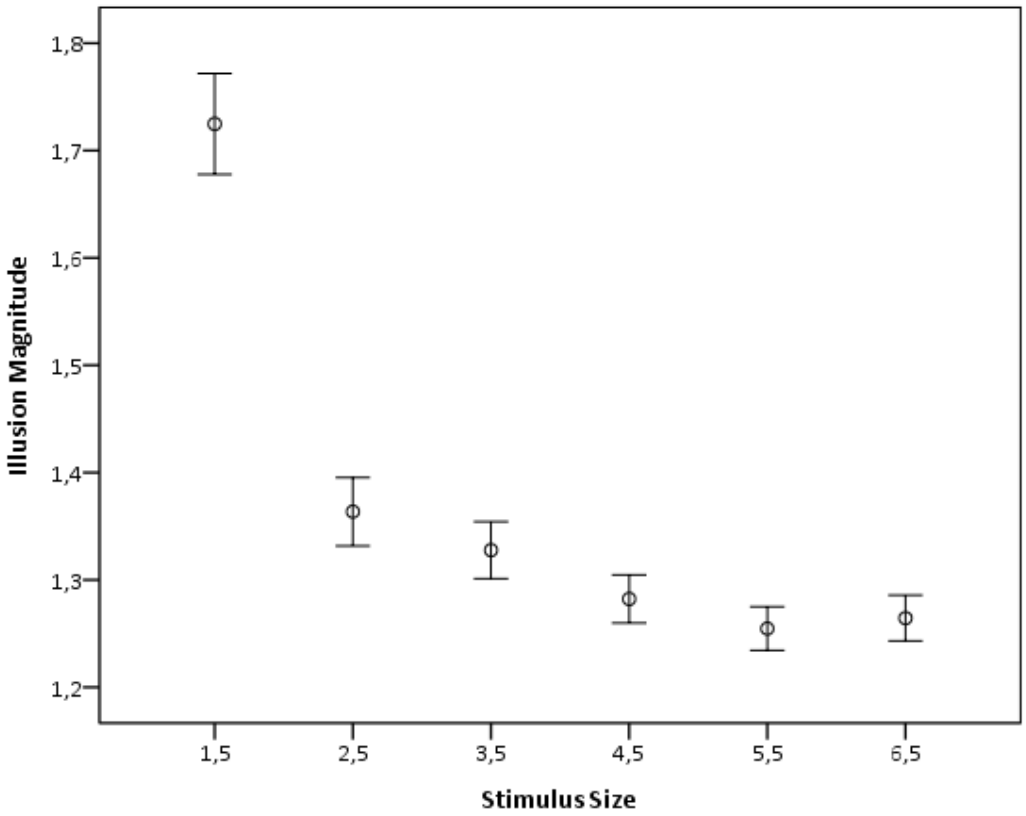

Figure 4: Illusion magnitude mean (on the y-axis) versus target stimulus size (on the x-axis). Stimulus size is indicated in Unity units. Here, 1 unity unit translates to 0.449° subtended visual angle.

This effect is not a result of relatively constant overestimation. Rather, the absolute difference between reported size and actual size follows Weber's law for all stimuli sizes but the smallest one (figure 5).

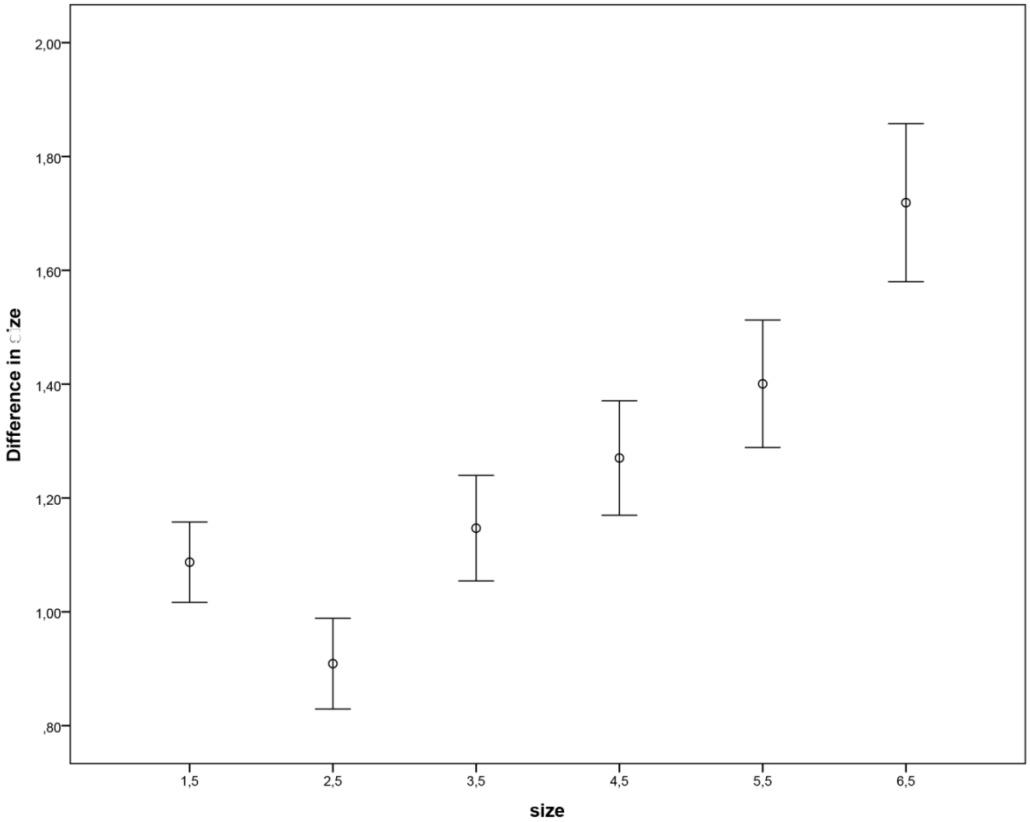

Figure 5: The y-axis shows the average difference between estimated size and actual size in unity units, flanked by 95% confidence intervals. The x-axis lists target stimulus size.

Discussion

This project aimed to investigate the roles of depth cue salience and vestibular input as contributors to the moon illusion. When studied in isolation, both of these factors proved to be significant predictors of illusion magnitude. In the no movement condition, the moon illusion was found in both environments, i.e. there was a systematic overestimation of stimulus size when it was seen at low elevations (figures 1 and 2). The effect was more pronounced in the realistic environment, presumably as a result of higher depth cue salience in that environment. Note that the illusion magnitude in the realistic environment is higher than most magnitudes reported in studies where 2D stimuli are used, which range roughly from 1.01 to 1.4 (Jones & Wilson, 2009; Weidner, Plewan, Chen, Buchner, Weis & Fink, 2014; Coren & Aks, 1990). Instead, the magnitude found here is closer to that of naturalistic observations of the moon, which can range from 1.5 to 2 (Ross & Plug, 2002). This shows the capabilities of VR HMDs to have subjects experience immersive environments which closely resemble natural settings, while still allowing for accurate measurements in a controlled environment.

In the ‘blank’ condition, data show that participants are most accurate in their size judgement when looking at objects that are straight ahead of them. When participants tilt their head up or down, slight overestimation of object size occurs. These results seem to imply a direct link between head tilt and size perception. Interestingly, this effect runs counter to that found by Wood, Zinkus & Mountjoy (1968), who reported an underestimation of size for stimuli directly above the participant. It cannot be said with certainty that this phenomenon is caused by the otolith organs signalling to the visual system. However, given the fact that head rotation is the only variable, and given the previously cited literature linking otolith organ input to size perception, it is a plausible explanation.

In the movement condition, stimulus elevation is a significant predictor of illusion magnitude in both environments. Head rotation, on the other hand, is only significant in the abstract environment, presumably due to its small effect size and confounded effect with stimulus elevation. The interaction between head rotation and stimulus elevation is only significant in the realistic environment, where it has a very small coefficient. The effect implies that for stimuli of higher elevations, when a subject's head is at a more extreme angle, the resultant overestimation would be larger. However, with its minute coefficient and significance in only one of the environments, both the reliability and possible relevance of this effect are suspicious. Most likely, the relative contributions of head rotation and depth cue salience in the moon illusion are additive.

A relatively strong effect in all conditions was the starting size of the reference stimulus. This effect is positive, indicating that larger reference stimuli lead to larger overall responses and thus larger illusion magnitudes on average. This is not very surprising, as it indicates an anchoring effect; the final size judgement is not only based on the initial size judgement held in working memory, but also on the presently seen globe that the subject is adjusting. Because the reference stimulus size was randomized, it should not have any systematic confounding effect on the size judgements for any given target stimulus size.

Another strong, yet unexpected, effect is that of target stimulus size. Specifically, the smallest stimulus shows a much greater illusion magnitude (figure 4), which cannot be explained by the effects of Weber's law. Absolute differences in size judgements show a steadily increasing difference between actual and estimated size as predicted by Weber's law, with the overestimation for large stimuli being larger than the overestimation for small stimuli (figure 5). However, the overestimation of the target stimulus of size 1,5 is larger than the 2,5 one, and on

par with the 3,5 one. Notably, the size that shows this effect has a subtended visual angle of 0.67°, and is closest in size of all stimulus sizes to the actual moon as seen from the earth, which subtends around 0.5° visual angle. This could be indicative of a mnemonic component to the moon illusion. More specific explanations concerning the mechanics of this component would be tenuous at this point, and further investigation into this effect may be worthwhile.

To conclude, a VR HMD was successfully used to show the roles of depth cue salience and head rotation in producing the moon illusion. Effect sizes were larger than those reported in studies using 2D stimuli and closer to those found in naturalistic observations of the moon, implying an extra level of immersion added by VR. Also, the effect found for head rotation runs counter to the one reported by Wood, Zinkus & Mountjoy (1968), in that increased head rotation seems to lead to overestimation of stimulus size. Additionally, the actual size of the stimulus used to measure the moon illusion seems to play a large role in the illusion, as stimuli that are close to the actual angular size of the moon elicit a larger illusion magnitude. Finally, this project showcases the potential of VR HMDs as research tools in perceptual science. VR HMDs allow for presentation of highly immersive environments, full control over the subject’s visual experience and accurate quantification of a subject’s visual experience, head position and head movement.

References

Carter, D. S. (1977). The moon illusion: A test of the vestibular hypothesis under monocular viewing conditions. *Perceptual and Motor Skills, 45*(3), 1127-1130.

Claparède, É. (1906). L'agrandissement et la proximité apparents de la lune à L'horizon. [The enlargement and Apparent Proximity of the horizon moon] *Journal of Philosophy, 3*(2), 49-52.

Clément, G., Fraysse, M., & Deguine, O. (2009). Mental representation of space in vestibular patients with otolithic or rotatory vertigo. *NeuroReport: For Rapid Communication of Neuroscience Research, 20*(5), 457-461.

Clément, G., Skinner, A., Richard, G., & Lathan, C. (2012). Geometric illusions in astronauts during long-duration spaceflight. *NeuroReport: For Rapid Communication of Neuroscience Research, 23*(15), 894-899.

Clément, G., & Eckardt, J. (2005). Influence of the gravitational vertical on geometric visual illusions. *Acta Astronautica, 56*(9–12), 911-917.

Coren, S. (1992). The moon illusion: A different view through the legs. *Perceptual and Motor Skills, 75*(3), 827-831.

Coren, S., & Aks, D. J. (1990). Moon illusion in pictures: A multimechanism approach. *Journal of Experimental Psychology: Human Perception and Performance, 16*(2), 365-380.

Coren, S., & Girgus, J. S. (1972). A comparison of five methods of illusion measurement. *Behavior Research Methods & Instrumentation, 4*(5), 240-244.

Coren, S., & Girgus, J. S. (1978). *Seeing is deceiving: The psychology of visual illusions*. Oxford, England: Lawrence Erlbaum.

Ferrè, E.,R., Longo, M. R., Fiori, F., & Haggard, P. (2013). Vestibular modulation of spatial perception. *Frontiers in Human Neuroscience, 7*, 660.

Goldberg, J. M., Wilson, V. J., Cullen, K. E., Angelaki, D. E., Broussard, D. M., Büttner-Ennever, J. A., . . . Minor, L. B. (2012). *The vestibular system A sixth sense* (1st ed.). New York: Oxford university press, Inc.

Harris, L. R., & Mander, C. (2014). Perceived distance depends on the orientation of both the body and the visual environment. *Journal of Vision, 14*(12), 17-17.

Holway, A. H., & Boring, E. G. (1940). The moon illusion and the angle of regard. *The American Journal of Psychology, 53*(1), 109-116.

Jones, S. A. H., & Wilson, A. E. (2009). The horizon line, linear perspective, interposition, and background brightness as determinants of the magnitude of the pictorial moon illusion. *Attention, Perception, & Psychophysics, 71*(1), 131-142.

Kaufman, L., & Rock, I. (1962). The moon illusion, I. *Science, 136*(3520), 953-961.

Murray, S. O., Boyaci, H., & Kersten, D. (2006). The representation of perceived angular size in human primary visual cortex. *Nature Neuroscience, 9*(3), 429-434.

Nanavati, S. (2009). A history and analysis of the moon illusion. *The New School Psychology Bulletin, 6*(1), 15-25.

Redding, G. M. (2002). A test of size-scaling and relative-size hypotheses for the moon illusion. *Perception & Psychophysics, 64*(8), 1281-1289.

Ross, H., & Plug, C. (2002). *The mystery of the moon illusion* (1st ed.). Oxford: Oxford university press.

Van Eyl, F. P. (1972). Induced vestibular stimulation and the moon illusion. *Journal of Experimental Psychology, 94*(3), 326-328. doi:10.1037/h0032755

Weidner, R., Plewan, T., Chen, Q., Buchner, A., Weiss, P. H., & Fink, G. R. (2014). The moon illusion and size–distance scaling—Evidence for shared neural patterns. *Journal of Cognitive Neuroscience, 26*(8), 1871-1882.

Wood, R. J., Zinkus, P. W., & Mountjoy, P. T. (1968). The vestibular hypothesis of the moon illusion. *Psychonomic Science, 11*(10), 356-356.

Appendix A

The following script initializes the experiment and presents the introduction screen.

Script i

```
using UnityEngine;
using System.Collections;

public class Introt : MonoBehaviour {
    public GameObject introText;
    public GameObject Background;
    public static Introt intro;
    //testing
    private bool active;
    void Awake()
    {
        if (intro == null) {
            DontDestroyOnLoad(gameObject);
            intro = this;//assign static varable for anyone to use
        } else {
            if (intro != this) Destroy(gameObject);  //make sure there is only one intro
        }
    }
    // Use this for initialization
    void Start()
    {
        active = false; //don't respond to input
        introText.SetActive(false);
        Background.SetActive(false);
        setActive(); //for now: activate this
    }
    public void setActive() {
        introText.SetActive(true);//show intro text
        Background.SetActive(true);
        active = true;//respond to input
        DataLogger.dataLog.logStart();
    }

    // Update is called once per frame
    void Update()
    {
        //  "Fire1", "Fire2" "Fire3" are mapped to Ctrl, Alt, Cmd keys and three mouse or
joystick buttons
        if (active && Input.GetButtonDown("Fire1")) {
            introText.SetActive(false);
            Background.SetActive(false);
            active = false;
            Experimentt.experimentt.setActive();
        }
    }
}
```

The following script creates a text file in which gyroscope data from the rift as well as data concerning the task are stored

Script ii

```
using UnityEngine;
using System.Collections;
//will not work on "Web" - could not save local files
using System;
using System.Runtime.Serialization.Formatters.Binary;
using System.IO;
using UnityEngine.VR;

public class DataLogger : MonoBehaviour
{
    public static DataLogger dataLog;

    private string filePathAndName;
    private string dataBuffer;


    void Awake()
    {
        if (dataLog == null)
        {
            DontDestroyOnLoad(gameObject);
            dataLog = this;
        }
        else
        {
            if (dataLog != this)
            { //make sure there is only one datalog
                Destroy(gameObject);
            }
        }
    }
    //find a new filename to write to
    void SetFilename()
    {
        try
        {
            filePathAndName =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments) + "\\" +
Application.productName;
            if (!File.Exists(filePathAndName))
            {
                Directory.CreateDirectory(filePathAndName);
            }
        }
        //catch (System.PlatformNotSupportedException)
        catch
        {
            filePathAndName = Application.persistentDataPath;
        }
        filePathAndName += "\\";
```

```
        bool useIndex = false;
        bool useDateTime = true;
        if (useDateTime)
        {
            DateTime date1 = DateTime.Now;
            string fileName = date1.ToString("s");//sortable time format: 2008-04-
10T06:30:00
            fileName = fileName.Replace(':', '-');
//            Debug.Log(string.Format("MESSAGE -{0}-",fileName));
            if (!File.Exists(filePathAndName + fileName + ".txt"))
            {
                filePathAndName += fileName + ".txt";
            }
            else
            {//Never use an existing file!
                filePathAndName += fileName;
                useIndex = true;
            }
        }
        if (useIndex)
        {
            int index = 0;
            while (index < 999999)
            {
                string fileIndex = String.Format("{0,6:D6}", index);
                if (!File.Exists(filePathAndName + fileIndex + ".txt"))
                {//Never use an existing file!
                    filePathAndName += fileIndex + ".txt";
                    break;
                }
            }
        }
    }
    public void logStart()
    {
        SetFilename();
        dataBuffer += "Log start\r\n";
        Save();
        InvokeRepeating("logHeadOrientation", 0.2f, 0.2f);//repeat call every 0.2 seconds
    }
    public void logHeadOrientation()
    {
        var quaternion = InputTracking.GetLocalRotation(VRNode.Head);
        var euler = quaternion.eulerAngles;
        //        Debug.Log(string.Format("{0} {1}, ({2}) Quaternion {3} Euler {4}",
logPrefix, id, m_VRNode, quaternion.ToString("F2"), euler.ToString("F2")));
        //string text = string.Format("({0}) Quaternion {1} Euler {2}\r\n", VRNode.Head,
quaternion.ToString("F2"), euler.ToString("F2"));
        string text = string.Format("{0} Headrot {1}\r\n", DateTime.Now.ToString("s")
,euler.ToString("F2"));
        dataBuffer += text;
    }
    public void logTextLine(string text)
    {
        string fullText = string.Format("{0} {1}\r\n", DateTime.Now.ToString("s"), text);
        dataBuffer += fullText;
    }
    public void Save()
```

```
    {
        // WriteAllText creates a file, writes the specified string to the file,
        // and then closes the file.    You do NOT need to call Flush() or Close().
        System.IO.File.AppendAllText(@filePathAndName, dataBuffer);
        dataBuffer = "";

    }
    //Write out data and start a new file
    public void logStop()
    {
        CancelInvoke();
        Save();
        SetFilename();//make sure we have a new filename, even if we will never use it
    }
    //    public void Restart()
    //    {
    //        Stop();
    //        logStart();
    //    }
    void OnDestroy()
    {
        logStop();
    }

                                                    }
```

The following script controls the presentation of trials, randomizing size and elevation of stimuli within set boundaries as well as presenting break screens after a set amount of trials and switching between conditions.

Script iii

```
using UnityEngine;
using UnityEngine.SceneManagement;



public class Experimentt : MonoBehaviour
{


    public GameObject MoonOrbit;
    public GameObject MoonScale;
    public GameObject IndicateSizeText;
    public GameObject RefMoonScale;
    public GameObject GoodbyeText;
    public GameObject BreakText;
    public GameObject Background;
    public GameObject Moon;
    //public GameObject FixationCross;
    public GameObject NewSceneScreen;
    public Material MoonMat;
    public Material DefaultMaterial;

    public static Experimentt experimentt;



    private int active;//respond to input
    private int stage;
    //private int numScenes;
    private int sceneNum;
    private int sceneNumTot;
    private bool showSlider;
    private int numTotalTrials;
    private int numTrialperBlock;
    private int numBlocks;
    //private int numElevAndSizes;
    private int[] Elevations;
    private float[] r;
    private float[] ri;
    private float[] Sizes;
    private float[] sz;
    private float[] s;
    private float[] ms;
    private int[] rps;
    private float[] szps;
    private int[] ssbe;
    private int[] ssee;
    private int[] ssbs;
```

```
    private int[] sses;
    private int sf;
    private float[] refSizes;
    private Vector2 scrollDelta;
    private static System.Random rnd = new System.Random();



void Awake()
    {
        if (experimentt == null)
        {
            DontDestroyOnLoad(gameObject);
            experimentt = this;//assign static varable for anyone to use
        }
        else {
            if (experimentt != this) Destroy(gameObject); //make sure there is only one
experiment
        }
    }
    public void setActive()
    {
        stage = 0;
        placeMoon(stage);
        MoonOrbit.SetActive(true);
        active = 1;//respond to input
    }
    // Use this for initialization
    void Start()
    {
        Transform[] Moons = MoonScale.GetComponentsInChildren<Transform>(true);
        foreach (Transform moon in Moons) DontDestroyOnLoad(moon.root);

        Transform[] RefMoons = RefMoonScale.GetComponentsInChildren<Transform>(true);
        foreach (Transform moon in RefMoons) DontDestroyOnLoad(moon.root);
        GameObject light = GameObject.Find("Directional Light");
        DontDestroyOnLoad(light);
        //DontDestroyOnLoad(IndicateSizeText);//will not destroy because
"CameraPostition" will not be destroyed
        //DontDestroyOnLoad(GoodbyeText);
        //DontDestroyOnLoad(BreakText);


        active = 0;//don't respond to input
        MoonOrbit.SetActive(false);
        IndicateSizeText.SetActive(false);
        RefMoonScale.SetActive(false);
        GoodbyeText.SetActive(false);
        BreakText.SetActive(false);
        Background.SetActive(false);
        NewSceneScreen.SetActive(false);
        //subset the following commands per scene
        stage = 4; // trialCounter
        sceneNum = 5; //scene counter
        sceneNumTot = 5; // total number of scenes
        showSlider = false;
        // Can easily change numTrailperBlock & numBlocks
        numTrialperBlock = 20; // trialCounter within Block
```

```
        numBlocks = 10; //Block counter indicator;
        numTotalTrials = numBlocks*numTrialperBlock; //Total number of trials per block
        //numScenes = 5;//number of scenes, including a final empty scene
        //numElevAndSizes = 6;

        Elevations = new int[23] {5, 15, 25, 35, 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, -
10, -20, -30, -40, -50, -60, -70, -80, -90 };
        Sizes = new float[6] { 1.5f, 2.5f, 3.5f, 4.5f, 5.5f, 6.5f };
        refSizes = new float[6] { 1.5f, 2.5f, 3.5f, 4.5f, 5.5f, 6.5f };
        ssbe = new int[5] { 4, 0, 0, 4, 4 }; // Starting point of range for Elevations
per scene
        ssee = new int[5] { (Elevations.Length), (8), (8), (14), (14) }; // End of range
for Elevations per scene
        ssbs = new int[5] { 0, 0, 0, 0, 0 }; // Starting point of range for Sizes per
scene
        sses = new int[5] { (Sizes.Length), (Sizes.Length), (Sizes.Length),
(Sizes.Length), (Sizes.Length) }; // End of range for Sizes per scene
    }


    bool placeMoon(int stage)
    {

        if (stage >= numTrialperBlock*numBlocks) return false;
        /* if (sceneNum==0)
        {
            int rps = rnd.Next(ssb[sceneNum+1], sse[sceneNum+1]);
            int szps = rnd.Next(ssb[sceneNum+1], sse[sceneNum+1]);
        }
        int r = rnd.Next(0, (Elevations.Length-1));
        int sz = rnd.Next(0, (Sizes.Length-1));
        */
        int rps = rnd.Next(ssbe[((sceneNum) * (-1)) + sceneNumTot], ssee[((sceneNum) * (-
1)) + sceneNumTot]);
        int szps = rnd.Next(ssbs[((sceneNum) * (-1)) + sceneNumTot], sses[((sceneNum) *
(-1)) + sceneNumTot]);


        //place the (next) moon:
        // MoonOrbit.transform.Rotate(-40, 0, 0); //add rotation relative to current
        int ri = Elevations[rps]; // (int)Elevations[r], random element for Elevations,
or Elevations[rps] for Elevation per scene
        float ms = Sizes[szps]; // Test if it actually keeps the float.
        Vector3 angles = new Vector3(360.0f - ri, 0.0f, 0.0f);//set to 20 deg above
horizon (add 360.0f - ri)
        string text = string.Format("Moon Elevation {0} Size {1}\r\n", ri, ms);
        DataLogger.dataLog.logTextLine(text);
        MoonOrbit.transform.localEulerAngles = angles;
        MoonScale.transform.localScale = new Vector3(ms, ms, ms);
        return true;
    }
    void LoadScene(int index)
    {
        int sceneIndex = index;
        if (sceneIndex == 0) sceneIndex = SceneManager.sceneCountInBuildSettings - 1;
//last scene should be empty
        Debug.Log(string.Format("SceneManager.LoadScene({0})", sceneIndex));
        SceneManager.LoadScene(sceneIndex);
```

```
        string text = string.Format("Start of scene {0}", sceneIndex);
        DataLogger.dataLog.logTextLine(text);
        GameObject cam = GameObject.Find("Main Camera");
        Camera camera = cam ? cam.GetComponent<Camera>() : null;
        if (camera)
        {
            if (index == (2) || index == (4))
            {
                camera.clearFlags = CameraClearFlags.Skybox; //turn on skybox for
realistic scene
            }
            else{
                camera.clearFlags = CameraClearFlags.SolidColor;
            }
         if (index == (2) || index == (4))
         {
                Moon.GetComponent<Renderer>().material = MoonMat;//give the moon a
realistic texture for a given scene
         }
            else { Moon.GetComponent<Renderer>().material = DefaultMaterial; //give the
moon a bland texture for other scenes
            }
         /*if (index == (1) || index == (2))
            {
                FixationCross.SetActive(true);
            }
         else
            {
                FixationCross.SetActive(false);
            }*/

        }
    }

    // Update is called once per frame
    void Update()
    {
        if (active > 0) ++active;
        if (active > 5)
        {
            //  "Fire1", "Fire2" "Fire3" are mapped to Ctrl, Alt, Cmd keys and three
mouse or joystick buttons


            if (Input.GetButtonDown("Fire1"))

            {
                if (showSlider)

                {
                    if (BreakText.activeSelf | NewSceneScreen.activeSelf)
                    {
                        BreakText.SetActive(false);
                        NewSceneScreen.SetActive(false);
                        Background.SetActive(false);
                        MoonOrbit.SetActive(true);
                        placeMoon(stage);
                        showSlider = false;
```

```
                    }
                    else {
                        sf = rnd.Next(0, (refSizes.Length)); //randomize reference moon
size
                        float s = refSizes[sf] + .1f * scrollDelta.y;
                        string text = string.Format("Moon Size estimate {0} (start at)
{1}\r\n", s, refSizes[sf]);
                        DataLogger.dataLog.logTextLine(text);
                        showSlider = false;//stop showing question
                        IndicateSizeText.SetActive(false);
                        RefMoonScale.SetActive(false);
                        Background.SetActive(false);
                        ++stage;
                        if ((stage % numTrialperBlock) != 0)
                        {
                            MoonOrbit.SetActive(true);
                            placeMoon(stage);           // move back to trial start
                        }

                        else
                        {
                            if (stage >= numTotalTrials) //if the last trial in a
condition is reached
                            {
                                --sceneNum;
                                Background.SetActive(true);
                                RefMoonScale.SetActive(false);
                                MoonOrbit.SetActive(false);
                                NewSceneScreen.SetActive(true);
                                showSlider = true;

                                if (sceneNum != 0) //check to see if there is still a
next scene
                                {
                                    LoadScene(sceneNum); //load next scene
                                    stage = 0; //reset trial counter

                                }
                                else
                                {
                                    // exit exp.
                                    LoadScene(0);
                                    NewSceneScreen.SetActive(false);
                                    GoodbyeText.SetActive(true);
                                    active = 0; //stop responding to input
                                    DataLogger.dataLog.logTextLine("Finished
experiment.\r\n");
                                    stage = 0;
                                    //we're done. thank you and goodbye}
                                }
                            }

                            else
                            {
                                if (sceneNum != 0)
                                {
                                    NewSceneScreen.SetActive(false);
                                    RefMoonScale.SetActive(false);
```

```
                                    MoonOrbit.SetActive(false);
                                    BreakText.SetActive(true); // end of block, show
break screen
                                    Background.SetActive(true);
                                    showSlider = true;
                                    DataLogger.dataLog.logTextLine("Finished current
Block.\r\n");
                                }

                            }
                        }
                    }
                    //LoadScene(0);
                    //SceneManager.LoadScene(0);
                }
                else { // response screen func
                    MoonOrbit.SetActive(false);//hide the moon
                    BreakText.SetActive(false);
                    Background.SetActive(false);
                    IndicateSizeText.SetActive(true);
                    NewSceneScreen.SetActive(false);
                    Background.SetActive(true);
                    float s = refSizes[sf];
                    RefMoonScale.transform.localScale = new Vector3(s, s, s);
                    RefMoonScale.SetActive(true);
                    showSlider = true;
                    scrollDelta.Set(0.0f, 0.0f);
                    //LoadScene(1);
                }
            }
            else{
                if (showSlider)
                {
                    scrollDelta += (2*Input.mouseScrollDelta);
                    float s = refSizes[sf] + 0.1f * scrollDelta.y;
                    //Debug.Log(string.Format("scrollDelta {0} {1}", scrollDelta.x,
scrollDelta.y));
                    RefMoonScale.transform.localScale = new Vector3(s, s, s);
                }
            }
        }
    }
       }
```