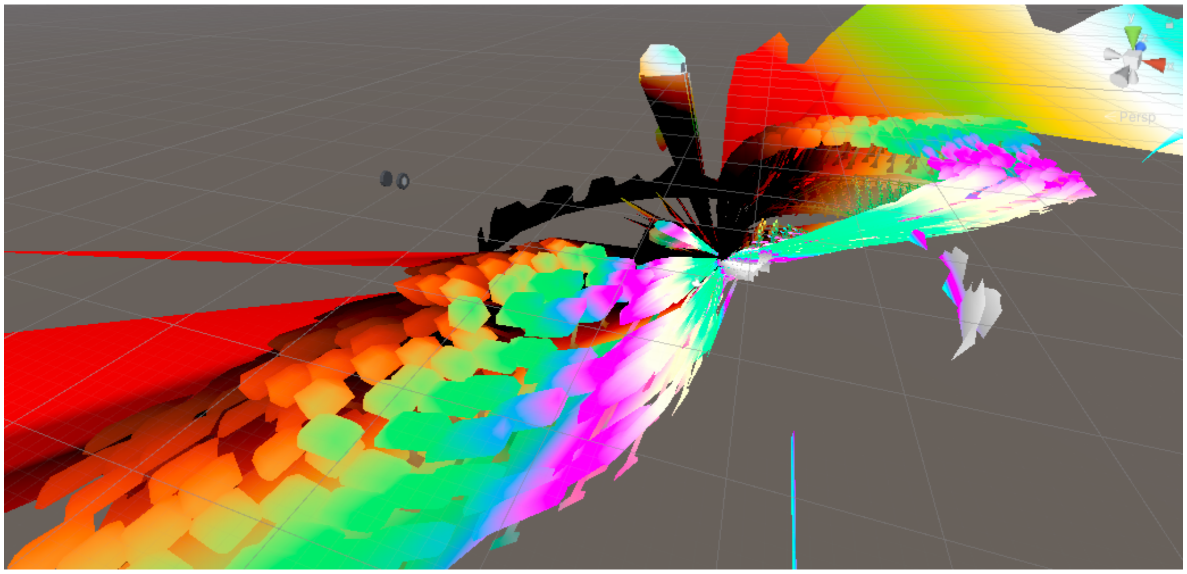
DEMONSTRATING SPECIAL RELATIVITY IN VIRTUAL REALITY



Dilation of the game world while moving near the speed of light.

Freya Heim

Callum Hepworth

Davis Johnson

Jed Yeo

Project Sponsors:

Engineering Physics Project Laboratory

Dylan Gunn

Miti Isbasescu

Bernhard Zender

Project 2253

Engineering Physics 479

Engineering Physics Project Laboratory

The University of British Columbia

April 16, 2023

Executive Summary

While equations relating to the theory of special relativity are proven and well understood, relativistic effects have yet to be directly observed by humans, as we are unable to travel at high enough speeds to perceive them. The goal of this project is to utilize the power of virtual reality (VR) to create an experience that allows users to visualize and experience these complex physical phenomena when traveling at speeds near the speed of light.

We developed a proof-of-concept VR experience for use on the Oculus Quest line of headsets. The user is able to observe four relativistic effects by simply moving around in the VR world. These effects are the Doppler effect, spotlight effect, space distortion, and time dilation. To convey these effects in an intuitive manner, we immersed the user in a VR farm, such that the optical effects were applied to traditional farm objects such as fences, barns, and tractors. Additionally, the effects of time dilation are apparent in the growing of plants and crops. To account for varying degrees of comfort with these effects, and to also provide the option to view each effect individually, we also developed a pop-up menu in the experience such that the user could toggle effects and adjust the intensity of each individual effect. We also enabled further exploration of the farm through the use of teleportation via joystick controls.

To acquaint new users with VR and as a way to introduce them to the experience, we developed a tutorial scene in which users could view individual effects applied to separate objects, as well as view basic explanations of each effect.

Our game engine of choice is Unity. Unity is extensively documented, free, and compatible with the main source of inspiration of the project: MIT Game Lab's OpenRelativity, which was developed as an open-source toolkit for video games simulating the effects of special relativity by artificially lowering the speed of light. Despite OpenRelativity not being compatible with VR, some of its codebase was adapted for our project.

Several areas for further development were identified. The depiction of relativistic phenomena could be extended to include general relativistic effects or relativistic shadows. The game could be made more interesting beyond physics demonstrations with goal-driven gameplay or more interaction possibilities within scenes. The fidelity of the currently implemented effects could also be improved if relativistic effects were transferred to a ray-tracing paradigm, both mathematically and in the shading engine by moving to a supporting engine such as Unreal 5.

Table of Contents

List of Figures

Chapter 1

Introduction

Although equations related to the theory of special relativity are well established and understood, direct observation of relativistic effects by humans has yet to be achieved since we cannot travel at relativistic speeds. This project aims to utilize the potential of virtual reality to artificially lower the speed of light in-game, so users can experience these otherwise imperceptible physical phenomena.

1.1 Sponsor

Our sponsor is the UBC Engineering Physics Project Laboratory. The Project Lab is interested in creating a VR experience that allows users to get an intuitive feel for complex physical phenomena. The Slower Speed of Light game developed by MIT Game Lab [3] demonstrated special relativity in a first-person perspective, and the Project Lab wishes to see this expand to virtual reality and with the potential for more expansive depictions of relativistic phenomena.

1.2 Background

Virtual reality has been around since the late 20th century [4], but has only become available commercially in the last decade. In 2014, Facebook invested nearly $2B USD on Oculus [5], demonstrating that the industry saw the massive potential of VR as a transformative technology. Indeed, immersive VR allows developers to provide users with sensory experiences and simulate otherwise impossible scenarios. To create an immersive, high-fidelity demonstration of special relativity, it was important for us to utilize the strengths of virtual reality.

1.2.1 VR Accommodations

We identified numerous areas of concern with using virtual reality. We recognized the potential for VR-induced motion sickness, given that the already disorienting nature of VR would be compounded by the optical effects in our experience. We also recognized that VR control is not intuitive, so it was important for us to think of ways to keep the experience simplistic control-wise. These concerns are addressed in the discussion of our implementation.

1.2.2 OpenRelativity

The main source of inspiration was MIT GameLab's OpenRelativity [3], an open-source library that provides a framework for building relativistic games and simulations in the Unity game engine. However, OpenRelativity is not compatible with VR. OpenRelativity was used in the creation of *A Slower Speed of Light (2012)* [3], a game where the player navigates an environment while experiencing a gradually decreasing speed of light, such that the effects of special relativity become more apparent to the player.

1.3 Project Objectives

In sum, an intuitive understanding of special relativity can be difficult to develop. Unlike other physics concepts (such as Newtonian mechanics), comprehending the concepts relating to relativity is challenging as experiments are far removed from the classroom. Understanding is gained through traditional studying and thought experiments. Our goal is to leverage the power of virtual reality and translate these formulae into an immersive experience so that users may gain a better understanding and appreciation of these physical phenomena.

We aim to build a VR experience that matches OpenRelativity's features. The experience should showcase the Doppler effect, spotlight effect, length contraction, and time dilation, and allow the user to explore a virtual world whilst experiencing these effects.

1.4 Scope

The experience is intended as a medium to convey the math and theory behind the relativistic effects. Our minimum viable product is feature parity with OpenRelativity - that is, showcase the Doppler effect and length contraction. Effects must be adjustable to diminish nausea and other health risks. A captivating scene for the user to explore is also a goal.

Chapter 2

Discussion

2.1 Theory

2.1.1 Special Relativity

Prior to 1905, solving inconsistencies that existed between Newtonian mechanics and Maxwell's equations of electromagnetism was an open problem in theoretical physics. Newtonian mechanics assumes that there are no absolute frames of reference (Galilean relativity), but Maxwell's equations assume that the speed of light is constant irrespective of the reference frame, seemingly requiring an absolute frame of reference. Previous attempts at reconciling this conflict were experimentally disproved in the decades leading up to special relativity, setting the stage for Einstein's breakthrough theory.

Einstein's insight was that Galilean relativity and an invariant speed of light can be unified by the blending of space and time into a space-time continuum. This result has non-intuitive effects for observers traveling close to the speed of light who witness events that occur in slow-moving frames of reference. For an exploration of these effects, see E

The inconsistency between events witnessed by two observers can be bridged using a mathematical tool called the Lorentz transform. The spacetime coordinates of an event (t, x, y, z) in a stationary frame of reference S, and coordinates (t', x', y', z') in a frame of reference S' moving with velocity v' relative to S in the x-dimension are linked according to the set of equations,

$$\begin{aligned} x' &= \gamma(x - vt) \\ y' &= y & \gamma &= \frac{1}{\sqrt{1-\beta^2}} \\ z' &= z & \beta &= \frac{v}{c} \\ t' &= \gamma\left(t - \frac{vx}{c^2}\right) \end{aligned} \tag{2.1}$$

where γ is called the Lorentz factor, and c is the speed of light in a vacuum. Armed with this mathematical framework we can move forward with understanding the four relativistic effects we chose to include in our project: time dilation, the Doppler effect, spatial distortion, and the spotlight effect.

Time Dilation

Time dilation is an observed difference in the elapsed time measured by two clocks in different inertial frames of reference. An object moving relative to an observer in an inertial frame of reference will be measured to tick slower than one in its own frame of reference. As an extreme example, a spacecraft traveling with a constant acceleration of $9.81\,\mathrm{m\,s^{-1}}$ (1 g) could theoretically cross the known universe within the lifetimes of the astronauts onboard.

The Lorentz transform (2.1) can be used to derive the degree of slowing in the moving clock seen by the stationary observer:

$$\delta t' = \gamma \delta t \tag{2.2}$$

where $\delta t'$ is the time interval of the moving object measured by the stationary observer and δt is the time interval experienced by the stationary observer. It is clear that as $v \to c$, the period of time of the moving object increases, as described qualitatively above.

Spatial Distortion

Imagine moving towards an object at relativistic speed. In the time it takes for the light emitted from the front of the object to reach us the object will have moved appreciably toward us, and in fact will reach us at the same time as light emitted from points farther away on the object at some point in the past. This is called the Terrell-Penrose effect. For example, a square of side length 1 m moving at $0.99c$ from left to right would appear as the shape shown in blue in Fig. 2.1, with the original, length contracted shape shown in red.

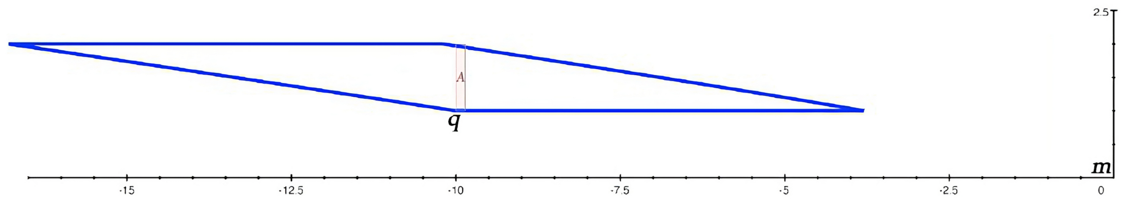


Figure 2.1: Terrell-Penrose Effect

Terrell-Penrose effect and length contraction applied to a unit square moving with $v = 0.99c$ to the right relative to a stationary observer [6].

The contracted length of the object can be calculated easily using the Lorentz transforms defined in 2.3. At a velocity of $0.99c$, the observed side length of the object would be:

$$\begin{aligned} L &= L_0\sqrt{1-\frac{v^2}{c^2}} \\ &= 1\cdot\sqrt{1-0.99^2}m \\ &= 0.14m \end{aligned} \tag{2.3}$$

The Terrell-Penrose Effect can be substantial, especially considering the degree of length contraction observed in the object by the stationary observer.

Spotlight Effect

The spotlight effect, often referred to as relativistic beaming, occurs when photons are emitted from an object moving at relativistic speeds relative to a stationary observer. Despite being released uniformly in all directions in the reference frame of the object, to the observer these photons appear to be concentrated at the front of the object. This effect can be visualized for a spaceship moving at various c in Fig. 2.2.

The exact release angle of an individual photon can be computed using Lorentz transformations and ultimately gives the result shown in Eq. 2.4:

$$\cos(\theta') = \frac{\cos(\theta)+\beta}{1+\beta\cos(\theta)}, \tag{2.4}$$

where θ is the angle from the x-axis of a photon emitted from an object moving along the x-axis at speed v as measured in the moving object's reference frame, and θ' is the same angled

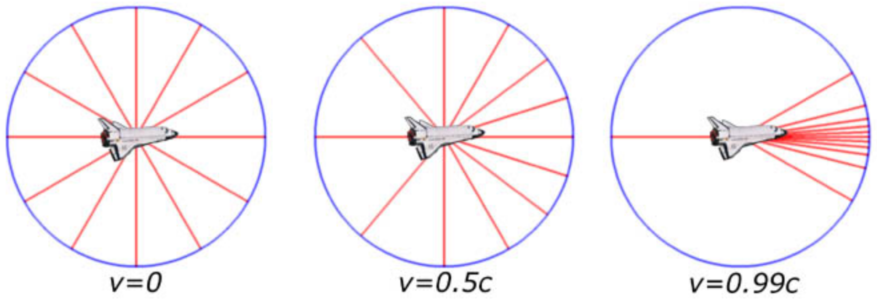


Figure 2.2: Spotlight Effect

Impact of spotlight effect (relativistic beaming) on photons emitted from a spaceship moving at various v to the right relative to a stationary observer [7].

measured by a stationary observer. Plotting θ' vs. θ for various values of $\beta = v/c$ as shown in Fig. 2.3 shows the impact of this effect at high v.

Indeed, as $v \to c$, all photons emitted from the object except those emitted at π radians from the direction of motion are grouped at the front of the object.

Doppler Effect

The relativistic Doppler effect is a change in the frequency of emitted photons from an object in a reference frame moving at relativistic speeds with respect to a stationary observer. The effect is analogous to the traditional Doppler effect, but accounts for the effects of time dilation and special relativity instead of the medium of propagation of the emitted photons.

The Doppler Effect differs due to the angle of wave compression or expansion. This phenomenon is unified through a single equation that takes into account the relative angle in the trajectory of the source relative to the receiver, dubbed the oblique Doppler effect (Eq. 2.5).

$$f_r = \frac{f_s}{\gamma\left(1 + \beta \cos\theta_r\right)}, \tag{2.5}$$

where θ_r is the angle between the velocity of the source and the axis connecting the source and receiver, with the receiver centered at the origin. We discuss the individual transverse and longitudinal effects in Appendix F.

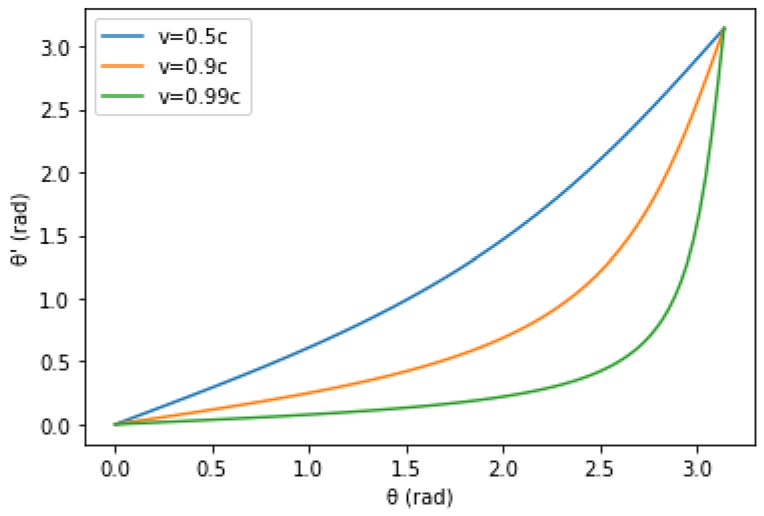


Figure 2.3: Relativistic Beaming

Effect of relativistic beaming on observed θ' of emitted photon from relativistic object.

2.1.2 Design in Unity

Objects and Materials

Each scene in Unity is composed of an arrangement of objects. Each object consists of a 3D model, which Unity represents as a wire-frame made up of a set of vertices and their connections. Then, a material is applied to the object, which instructs Unity how to assign colors and patterns to the structure. See 2.4 for an example of how a set of trees is decomposed into its wire-frame and material.

Attaching Scripts to Objects

In order to add custom behavior to objects, Unity allows adding custom components to an object. A component consists of a custom script and references to any objects the script needs access to. This allows other events in the scene to influence an object's behavior, like for example the position or motion of the player. These scripts are critical to Unity development, as they completely determine all non-cosmetic aspects of the game.

In particular, every Unity script includes two core functions:

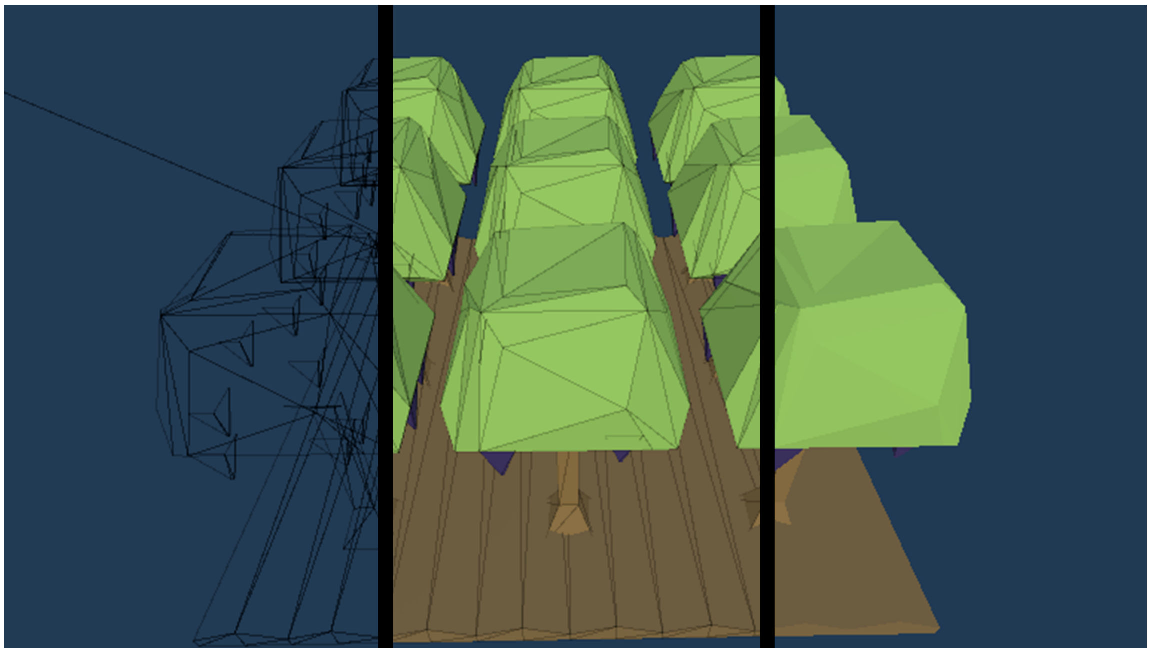


Figure 2.4: A Unity object decomposed into its wire-frame and material.
From left to right, the object's wire-frame, the wire-frame and the material applied, and only the material applied.

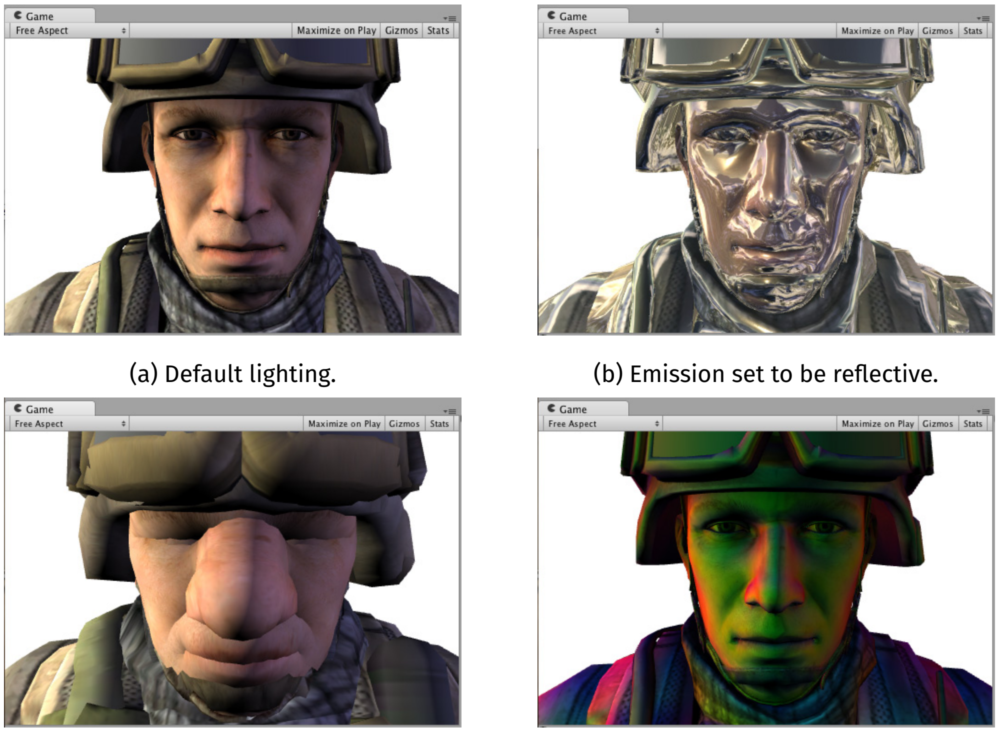
- **Start()** - Called when the game loads to perform any setup for that object.
- **Update()** - Called every frame the game is running to continuously update an object's state.

Using these functions, we can completely customize the behavior of each object.

Shaders

Shaders are programs that apply to materials to customize their appearance. A default shader program would display a material's color and patterns, but shaders can also apply lighting, shadows, and albedo. In addition, scripts in the scene can broadcast information for shaders to use, such as the player's position, or a parameter that determines how bright the scene should be.

In order to customize a material's appearance, shaders typically consist of two steps: the vertex step, and the fragment step. For the vertex step, the shader performs computations on each vertex for each object in the scene with that corresponding material. It is also possible to



(a) Default lighting.

(b) Emission set to be reflective.

(c) Translated vertices to give a swollen appearance.

(d) Custom coloring applied to fragments.

Figure 2.5: Examples of different shaders applied to a soldier courtesy of [1]

edit the vertex in this step, such as by changing its position. In the fragment step, the shader uses the vertex computations to alter each fragment, which refers to each pixel on the surface of a material. Refer to 2.5 for examples of how a shader can alter a material's appearance.

Because a shader is performing many calculations for each vertex in the scene, shaders run on the GPU rather than the CPU to maintain performance.

2.2 Design

2.2.1 Scene Development

The main goal of the project is to create an immersive and enjoyable experience. By creating an environment that incorporates more than just an empty plane, we're able to project relativistic shaders onto scene objects, and also further immerse the user in the experience.

We decided that a farm setting would be the most interesting to explore and experience relativistic effects. A potential farm scene has many vertices in the form of fences and buildings to serve as fragment points. Further, the effects of time dilation can be easily conveyed through growing crops on the farm - that is, the faster the player moves, the slower the player experiences time relative to the growing crop. We made use of the expansive Unity Store and utilized polygonal farm assets from [8]. While somewhat simplistic, these were more than sufficient for our shader program and provided an interesting environment for the user to explore.

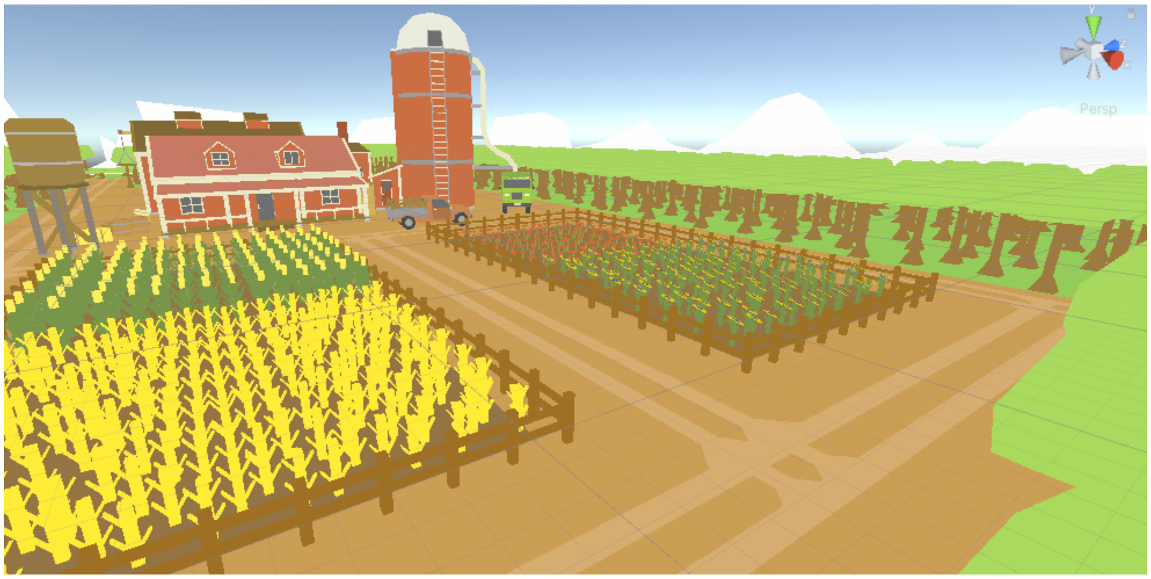


Figure 2.6: Farm scene.

With respect to the discussion in §1.2, it was also necessary for us to implement a tutorial scene before the user accesses the farm scene to serve as an introduction to virtual reality and the relativistic effects presented throughout the experience. The tutorial scene uses non-relativistic shading, except in the instance of four example objects. The effects were applied individually to separate objects, allowing the user to view each effect in isolation. For all other assets in the scene, no relativistic effects were applied. Once comfortable with VR/the effects, the user is invited to start the main experience by pressing a start button.

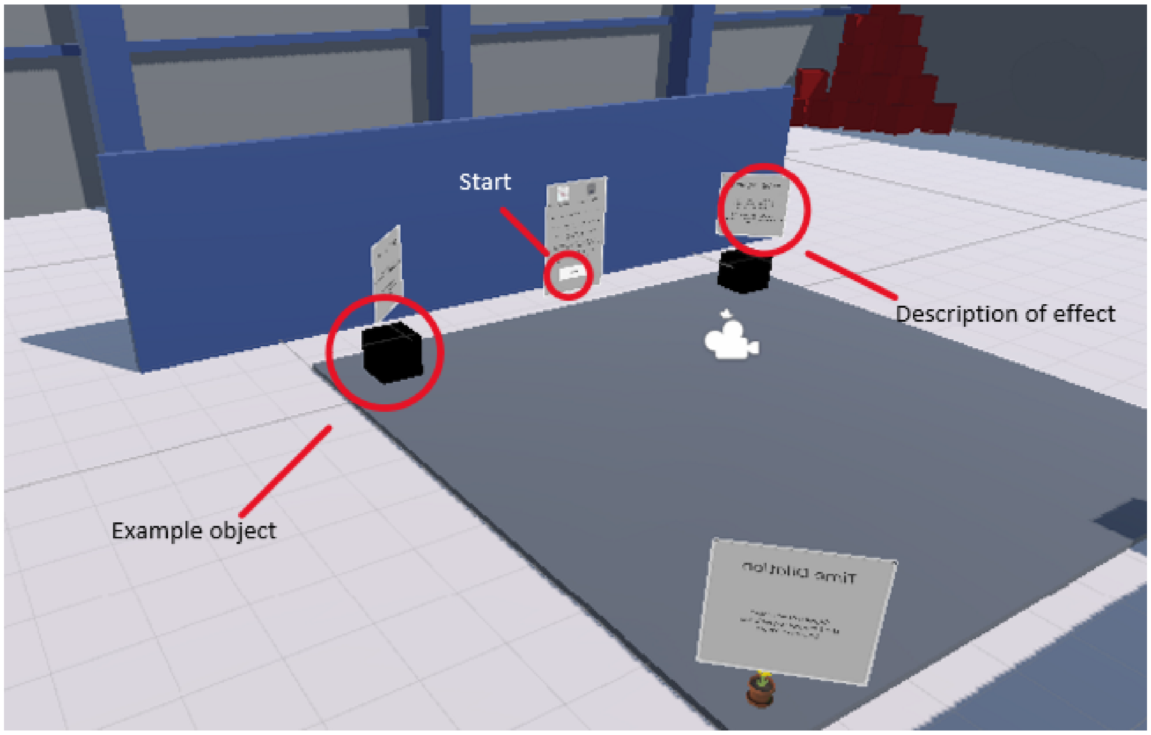


Figure 2.7: Tutorial scene, with example objects, short descriptions of each effect, and start button prompt.

2.2.2 In-game Configuration

Having a seamless user experience within the VR environment was crucial to mitigate the accommodation concerns mentioned in 1.2.1. With this in mind, our team decided to limit the interaction that the user had via the Oculus controllers to 3 inputs: the left menu button, the left trigger, and the right joystick. Consult C for more details on the Oculus controls. These inputs can be divided into two input categories:

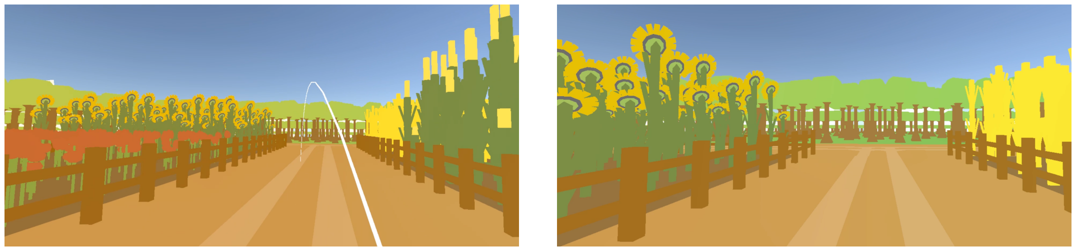


Figure 2.8: Using the joystick to teleport around the farm.

1. **user positioning** - right joystick
2. **menu interaction** - left menu button, left trigger

Input to the right joystick controls teleportation, which is enabled by moving the joystick forward. A parabola will appear indicating the location to which the user will be transported, and on the release of the joystick, teleportation occurs. This arc and the ensuing teleportation can be seen in 2.8. This feature allows a much larger scene area to be accessible than if the movement was restricted to the user's physical position while wearing the headset.

The left menu button and trigger allow the user to access an in-game menu which allows traversal between scenes and toggling and scaling of the intensities of individual in-game effects. This aspect of the game configuration gave users heightened autonomy over their experience and allowed the scaling down of effects for those who found the experience too intimidating, and up for those eager to experience exaggerated relativistic effects. The in-game menu can be seen in 2.9.

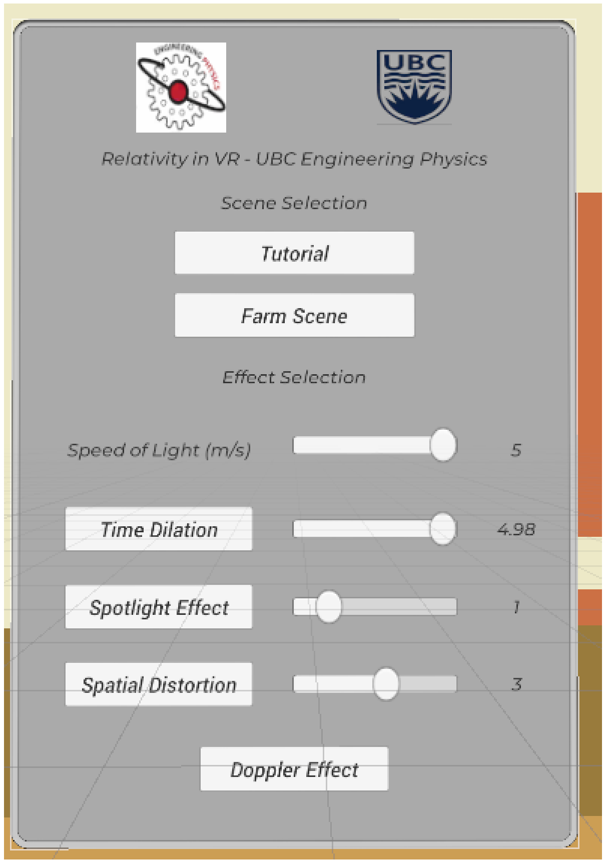


Figure 2.9: In-Game Menu

In-game menu with default values set and all effects enabled.

We determined the default effect values through consultation internally and with new users through a series of interactive demos.

2.2.3 Relativistic Effect Representation

Time Dilation

We discuss the physics governing time dilation in §2.1.1 - the implementation is straightforward. Relative velocity $\vec{v}_{\text{rel}} = \vec{v}_{\text{player}} - \vec{v}_{\text{object}}$ is derived from player velocity $\vec{v}_{\text{player}}$ and object velocity $\vec{v}_{\text{object}}$, and the Lorentz transform for time dilation (2.1) is applied to find the Lorentz factor and the time dilation for each frame,

$$\delta t' = \frac{\Delta t}{\sqrt{1 - \left(\frac{C_{ti}||\vec{v}_{\text{rel}}||}{c}\right)^2}} \tag{2.6}$$

where C_{ti} is a system constant to allow the severity of time dilation to be increased.

As in the farm scene described in §2.2.1 we apply this result to scale plants to simulate growth. Plants are expected to grow in an amount of time T, and Unity operates at a finite frame rate with an interval δt_i for each frame. So, the plant grows

$$\frac{\delta t_i}{T} = n_i\% \text{ growth in frame } i \tag{2.7}$$

which is scaled by the Time dilation formula (2.6) since it is proportional to δt_i. If the player moves quickly, the plant will scale more quickly, and hence time will appear to move more quickly.

To view a GIF of time dilation, as well as all the other effects, consult [9].

Spatial Distortion

Spatial distortion is applied to objects in the scene by a custom applied to their materials. Taking the player's position and the current speed of light as inputs, for each vertex, the shader computes the relative velocity between the player and that vertex, as in 2.1.

We modify these equations slightly, including a scaling parameter C_{sd} that scales the relative velocity so that the relative intensity of each effect can be independently controlled:

$$\beta = C_{sd}\frac{v_{rel}}{c} \qquad \gamma = \frac{1}{\sqrt{1-\beta^2}} \tag{2.8}$$

Then, given the player's position $\vec{x}_{\text{player}}$, the vertex's position $\vec{x}_{\text{vertex}}$, and $\vec{l}$, the vector from the player to the vertex, we transform each vertex to its new position $\vec{x}'_{\text{vertex}}$ according to

$$\vec{x}'_{\text{vertex}} = \vec{x}_{\text{player}} + \gamma\vec{l}. \tag{2.9}$$

This results in the distance between the player and an object contracting as the player moves toward it.

To view a GIF of spatial distortion, as well as all the other effects, consult [9].

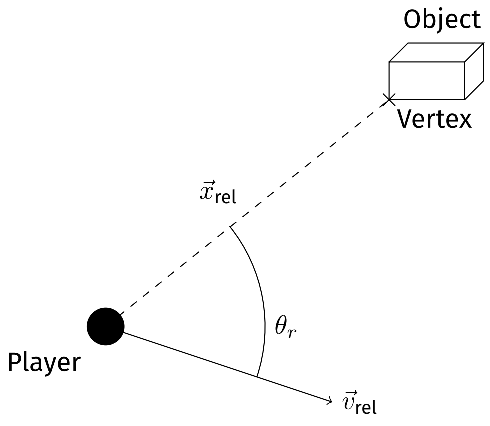


Figure 2.10: Diagram of in-engine Doppler effect geometry.

Doppler and Spotlight Effect

We discuss the physics of the Doppler effect in §2.1.1 - this gives the general equation for the relativistic Doppler effect in arbitrary motion,

$$D = \frac{f_r}{f_s} = \frac{1}{\gamma(1 + \beta \cos \theta_r)}$$

which we apply to the vertices of each object. The relative velocity $\vec{v}_{\text{rel}}$ and position $\vec{v}_{\text{rel}}$ between the player and vertex are found so that

$$\cos \theta_r = \frac{\vec{v}_{\text{rel}} \cdot \vec{x}_{\text{rel}}}{||\vec{v}_{\text{rel}}|| \cdot ||\vec{x}_{\text{rel}}||} \tag{2.10}$$

as per Figure 2.10. The resulting Lorentz factor is as expected from (2.1),

$$\beta = \frac{||\vec{v}_{\text{rel}}||}{c}, \qquad \gamma = \frac{1}{\sqrt{1-\beta^2}} \tag{2.11}$$

so that the Doppler factor can be applied.

The application of the Doppler factor is challenging as computer displays use an sRGB ("Standard Red Green Blue") color spectrum which lacks a direct correspondence to wavelengths and therefore cannot easily be shifted. Extra detail on this is provided in Appendix D. We use the process of OpenRelativity [3] for the sake of time and to avoid the significant cost of assets from color standards organizations. The resulting effect can be seen in Figure 2.11a.

The spotlight effect implementation is not dissimilar from that of the Doppler effect, although the physics are quite different. The physical process we describe in §2.1.1 can not be performed as they involve the manipulation of light intensity in space, which vertex shading does not permit. The physically accurate way to do this would be to use a raytracing method and allow the rays to be pulled to group around the user's vision so they appeared more intense as a result, but this is highly out of scope.

The spotlight implementation we used was based upon the observation that the spotlight effect depends on a factor of the Doppler scalar D [10]. The incoming light intensity is scaled with an angular dependence of $\cos\theta_r$, and a parametrically variable intensity curve so that $L \in [-1, 1]$ with a variable mapping. Hence, we calculate the luminance factor as

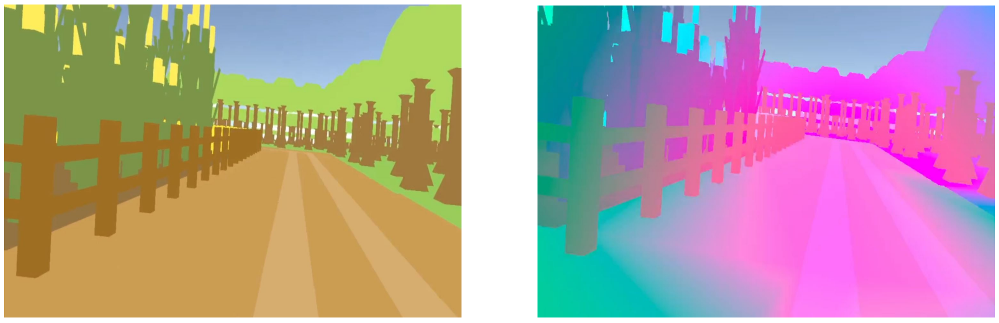
$$L = \frac{\text{Transformed luminance}}{\text{Base luminance}} = -\text{sign}\left(\cos\theta_r\right)\left|\beta^2 \cos\theta_r\right|^{1/C_{sl}} \tag{2.12}$$

where C_{sl} is a controllable factor governing the intensity of the spotlight effect so that it is relatively slow to "accelerate" with c for smaller (< 1) C_{sl}, and faster with larger C_{sl}.

Typically, as discussed in Appendix D, changes in luminance should be easy to depict in XYZ color spaces. However, we determined experimentally that the XYZ color space approximated by MIT does not have its Y parameter in direct correlation with luminance. We instead handled the increase in luminance in the sRGB space,

$$\langle R', G', B' \rangle = \langle R, G, B \rangle + L\left(1 - \langle R, G, B \rangle\right) \tag{2.13}$$

so that $\langle R', G', B' \rangle$ would approach white $\langle 1, 1, 1 \rangle$ as $L \to 1$ and black $\langle 0, 0, 0 \rangle$ as $L \to -1$ without significantly altering the chromaticity in interim values. The results of these empirically derived approximations are demonstrated in Figure 2.11b.


(a) Doppler Effect.

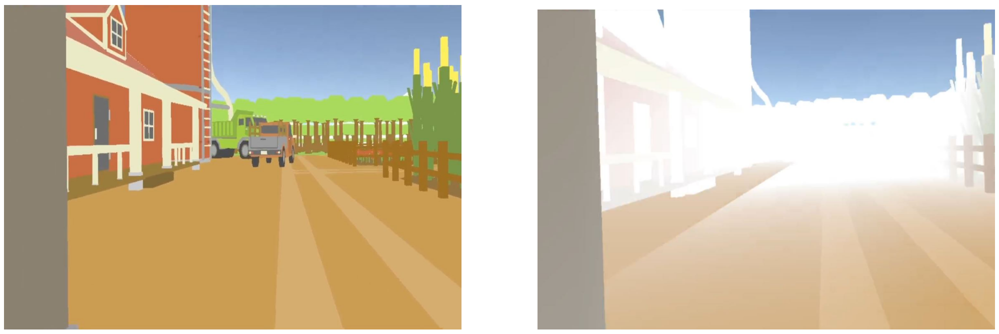

(b) Spotlight Effect.

Figure 2.11: Screenshots demonstrating the Doppler and spotlight effects within the game. *Note the drastic shift towards blue and purple coloration in (a) indicative of the Doppler effect for an observer moving forward near c. Note also the large increase in brightness, more drastically near the center of the frame in (b) as is typical of the spotlight effect for an observer moving forward near c.*

2.3 Tests

2.3.1 Feedback and User Testing

In order to get feedback on the game, we sought out the perspectives of UBC Physics faculty, the Engineering Physics Project Lab, and Engineering Physics peers. In these tests, users explored the farm scene as the tutorial was still in development.

Figure 2.12: Director of the UBC Engineering Physics Project Lab, Dylan Gunn trying out the game.

Unexpectedly, no users reported any nausea or headache from using VR or the game, which indicated that we succeeded in minimizing these risks in our planning and development.

However, many users did report that the Doppler effect was too intense, and was distracting from the other effects when all were enabled. In addition, many users had difficulty understanding the controls for interacting with the menu, although moving around the scene was intuitive.

Feedback integration

In response to feedback that the Doppler effect was too intense with our default settings, we adjusted our default parameters. Since we did not include a scaling parameter for the Doppler effect, we increased the default speed of light, in order to decrease the intensity of all the effects. Then, we individually scaled up the intensity of the spatial distortion and spotlight effects, so they were the same as previously.

To address user difficulties with learning the controls, we discussed teaching the user the controls as a part of the tutorial, where on-screen prompts would introduce each control with a diagram of which buttons to press. However, we concluded that this would excessively widen the scope of the project, so we leave it as a future consideration.

2.3.2 Project Fair Demonstration

At the Engineering Physics project fair, we showcased our game to students, faculty, and industry members. In this demonstration, users were able to explore both the tutorial and the farm scenes. As in our earlier demos, users continued to report no issues with nausea or headaches. In addition, the Doppler effect appeared to be less overwhelming, and the new default settings were better suited for first-time users. Some users with prior VR experience felt comfortable operating the menus and teleporting, although many users required more guidance to get familiar with the controls. We expect that the addition of teaching the controls as part of the tutorial, as discussed in 2.3.1, would be much more intuitive than verbal instructions.

2.3.3 Comparison with OpenRelativity

We achieved feature parity with OpenRelativity - MIT Game Lab implemented the Doppler effect and space dilation in their shaders, both of which we also implemented. MIT implemented a relativistic shader on the skybox which we did not replicate, and had a more well-featured demonstration of the effects than the basic "sandbox" of our farm.

Novel features of our project include time dilation and the spotlight effect. Virtual relativity is also a new medium and allows more interactivity for the user, as was hoped for. We believe the virtual reality experience to be a worthy competitor to that of MIT Game Lab in demonstrating special relativity.

Chapter 3

Conclusions

In this project, we aimed to condense the theories and equations behind special relativity into an immersive virtual reality experience. Our aim was to allow users an opportunity to develop a deeper understanding and intuitive sense of special relativity.

The tutorial scene serves as a means for users to get acclimated to virtual reality. The four included relativistic effects are introduced by applying them individually to example objects. A more in-depth tutorial experience that guides the user more seamlessly through each of these effects is a potential avenue for improvement.

In the main farm scene, Doppler, spotlight, and space dilation effects are applied to all vertices, and time dilation is conveyed through the use of farm crops. Further improvements might include exploring more relativistic effects, using higher resolution textures or assets to amplify the effects of the shader, or moving to a more sophisticated game engine to implement ray-tracing.

The in-game menu provides increased user autonomy over their demo experience and also takes strides to address the accessibility concerns associated with VR. The menu can be improved to provide a standardized scale for each of the individual relativistic effects, and a unique scale for the Doppler effect which is currently not implemented.

In summary, the project serves as a proof-of-concept VR experience that allows users to get an intuitive feel for these relativistic effects. While it is not perfect in terms of completely immersing the user in a relativistic environment, it allows one to gain a greater intuition for some of the theory behind special relativity.

Chapter 4

Recommendations

A few areas for further development remain, which we recommend exploring further.

1. **Explore further relativistic effects.** Further physics phenomena that would be interesting and relevant to demonstrate, but were outside of the scope of our project, are relativistic mass, time-dependent objects, as well as relativistic hands and shadows. One obstacle to time-dependent objects given the current implementation is that a shader shares parameters among all objects that use the same material. This means that we would be unable to share materials between two objects that would move with different motions in the scene.

2. **Teach the user the controls as part of the tutorial.** Although the controls are very standard compared to other VR games, for many users, this will be their first experience with VR. As such, we recommend introducing the controls one at a time in the tutorial, with an on-screen overlay that indicates the function of each button.

3. **Limit the bounds of the tutorial and farm scenes.** Given the scope of our project, users can move outside of the desired regions of the scenes using teleportation. This is acceptable when a member of our team is on hand to reset the scene for a confused user but should be prevented if the game were to release to a wider audience.

4. **Add meaningful game-play in addition to physics demo.** Although the focus of our project was to produce a VR experience that would demonstrate the effects of special relativity, it could be more engaging to provide objectives or tasks for users to accomplish in the world. Adding details like harvesting crops, or driving a tractor would provide opportunities to gamify the experience.

5. **Launch the game to the Oculus Store.** This would require going through a certification process to get the application published to the store and would significantly improve accessibility to the project. Currently, a curious user would need to clone our repository and load the game to their Oculus via Unity following the instructions in B. Enabling any user with an Oculus headset to download the application from the store would greatly increase the reach of our project.

6. **Bring relativity to a state-of-the-art engine.** Bringing the currently implemented effects into a ray-tracing engine such as Unreal 5 could improve the fidelity and impact of the effects, but would require significant migration to the Unreal platform.

Chapter 5

Deliverables

- **RelativisticFarm** - The main repository for the game that implements the tutorial and farm scenes.
- **LeekRelativity** - The sub-repository that implements the relativity shaders and time dilation scripts.

References

[1] Unity. Unity - manual: Surface shader examples.

[2] Vladimír Nepor. What xr controllers exist & why we need input manager, August 2020.

[3] Gerd Jortemeyer, Phillip Tan, Zach Sherin, Ryan Cheu, Steven Schirra, and Sonny Sidhu. Mit game lab: Open relativity.

[4] Wikipedia the Free Encyclopedia. Virtual reality, 2023.

[5] CBC. Facebook to buy oculus virtual reality firm for 2b, 2014.

[6] Andrew York. Terrell-penrose effect for objects approaching relativistic velocities, 2020.

[7] Alexis Brandeker, 2002.

[8] Unity. Polygonal farm assets by a.r.s|t on the unity store.

[9] ENPH 479 2253 VR Relativity Team. Relativistic effect demo gifs, April 2023.

[10] Wikipedia the Free Encyclopedia. Relativistic beaming, 2022.

[11] Oculus. Device setup.

[12] Oculus. Build and configuration overview.

[13] IEC. Multimedia systems and equipment - colour measurement and management - part 2-1: Colour management - default rgb colour space - srgb. *International Electrotechnical Commission*, IEC 61966-2-1:1999, 1999.

[14] dotPDN LLC. Paint.net, 2023.

[15] CIE. Colorimetry — part 2: Cie standard illuminants. *International Commission on Illumination*, ISO/CIE 11664-2.2022, 2022.

[16] Wikipedia the Free Encyclopedia. Color vision, 2023.

[17] Wikipedia the Free Encyclopedia. Cie 1931 color space, 2023.

[18] Relativity of simultaneity, Jan 2023.

Appendix A

Application Architecture

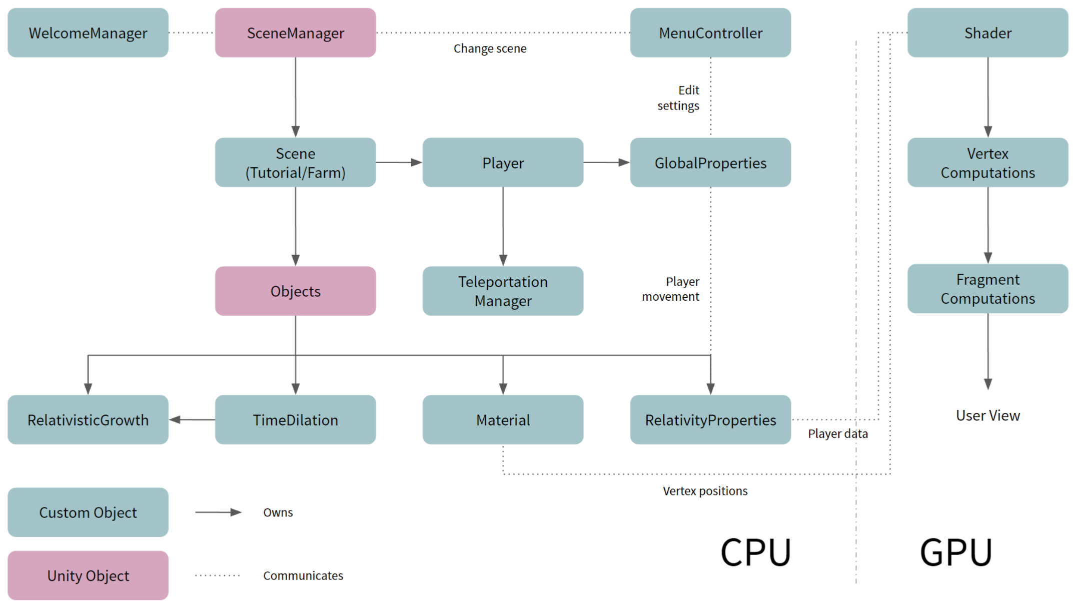


Figure A.1: The architecture of the game, including both built-in Unity components and custom developments.

Appendix B

Building Projects on Meta Quest Headsets

1. Follow the guide in [11] to set up a developer organization and enable developer mode on the headset.
2. In addition to the above procedure, you must also enable developer mode on your Oculus by downloading the Meta Quest app, logging into your developer account, and enabling developer mode under Headset Settings. Developer mode must be enabled in **both of these locations** in order to build to your headset.
3. Connect the Oculus to your computer via a USB-C to USB 3.0 cable, and connect to QuestLink using this wired connection. Building the application **does not work** using AirLink.
4. Build the project from Unity by following the guide in [12]. Select your device from the device drop-down, and select **Build and Run**.

Appendix C

Control Scheme

Button	Function
Thumbstick (Right)	Teleport in either tutorial or farm scene
Menu (Left)	Make menu appear in farm scene
Trigger (Left)	Interact with buttons or sliders in the menu or in the tutorial scene
Oculus (Right)	Bring up Oculus functions, or close game and return to Oculus home

Table C.1: In-game controls using the Oculus controllers

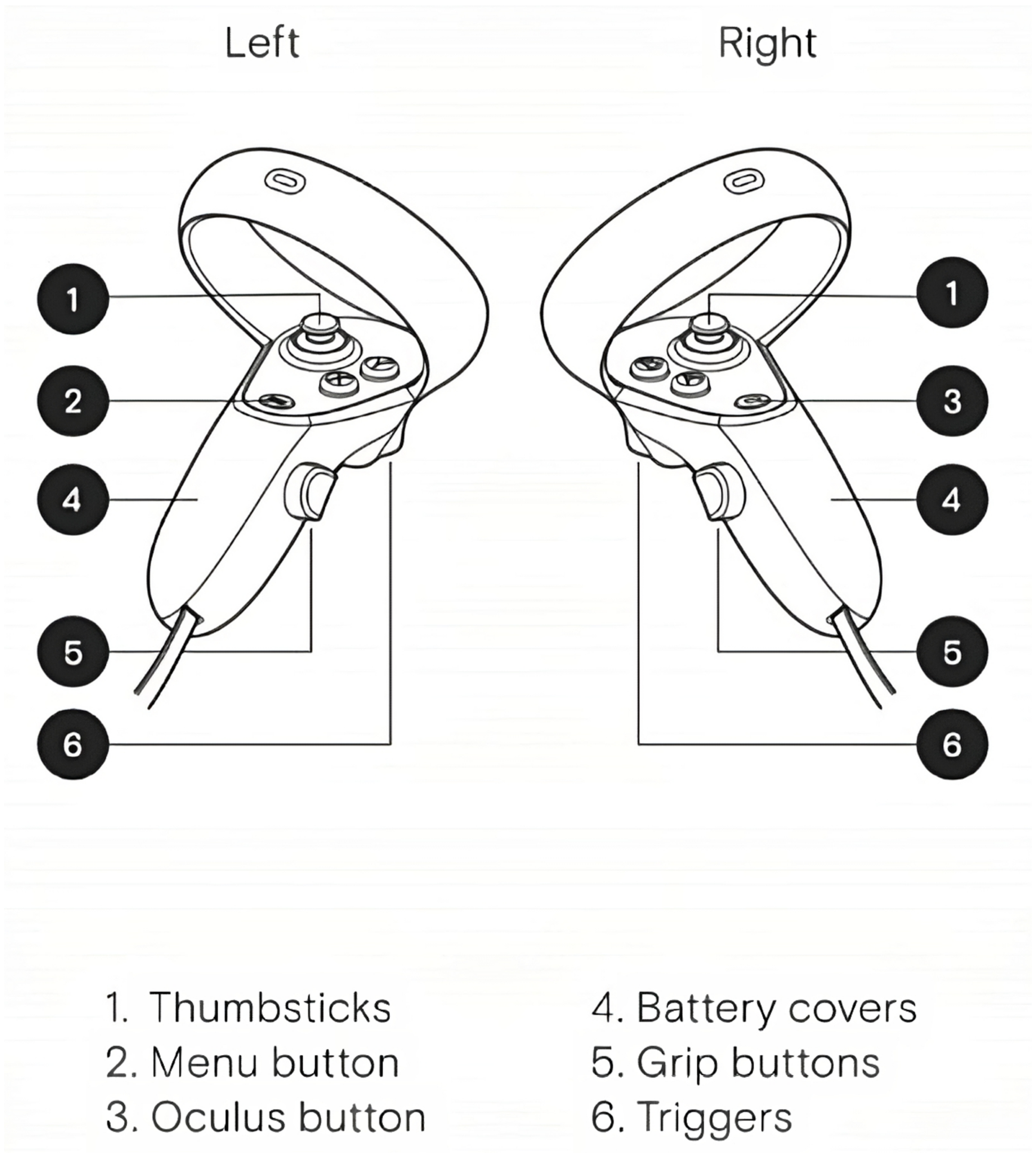


Figure C.2: The layout of buttons on the Oculus controllers from [2].

Appendix D

Colour Theory

As discussed in §2.1.1, the effect on colour due to the Doppler effect is best characterized by scaling the wavelength of monochromatic light (light that contains only one wavelength). The near-universal method of rendering light in consumer products is using the Standard Red-Green-Blue (sRGB) colour space [13].

The sRGB colour space is characterized by its three parameters: red, green, and blue which combine to form a particular output colour. Each parameter refers to the saturation of its source colour over a pitch-black background, so that smaller values appear darker. A minimal value of 0 for red, green, and blue creates black, whereas a maximal value of 1 creates white. A typical radial display of the sRGB colour space is shown in Figure D.1.

The sRGB colour space is based upon colour standards for white, red, green, and blue. The standard for white is so that a maximum luminance can be chosen - luminance refers to the

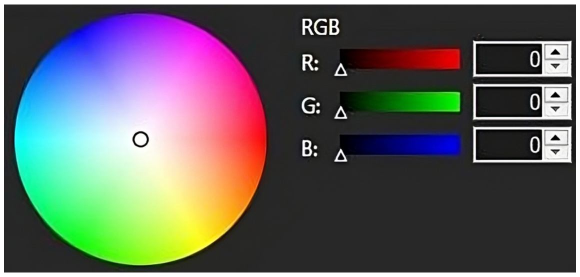


Figure D.1: The RGB color space.
A typical display of the RGB colour space. The wheel (left) is a transposition of the three colour axes (right). Image courtesy of the free program Paint.net [14].

intensity of visible light, and the standard used by sRGB is D65 "average daylight" ([15], [13]). The red, green, and blue standards do not refer to particular wavelengths of monochromatic light but rather to wavelength spectra that create light of each colour. The red-green-blue standard is a subset of the rods and cones present within the human eye [16] and can within the scope of this report be assumed to well-approximate the human perception of the sRGB space's expected colours.

To transpose sRGB such that it does not depend on a non-absolute colour standard, as with the scientific concept of wavelengths, the CIE 1931 XYZ colour space is useful. This spectrum serves as an absolute reference for other colour spaces [17] and is defined such that its X and Z parameters cover all chromaticities for a particular luminance Y (this luminance is still relative to D65 daylight). This space is related to the CIE 1931 RGB colour space by a linear transform,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490\,00 & 0.310\,00 & 0.200\,00 \\ 0.176\,97 & 0.812\,40 & 0.010\,63 \\ 0.000\,00 & 0.010\,00 & 0.990\,00 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{D.1}$$

The CIE 1931 RGB space is similar to the sRGB space although it is defined upon the "absolute" perception of red, green, and blue within the human eye rather than the defined red, green, and blue standards. For demonstrative purposes, it is reasonable to approximate sRGB and CIE 1931 RGB as the same so that sRGB can be linearly transformed to CIE 1931 XYZ by (D.1).

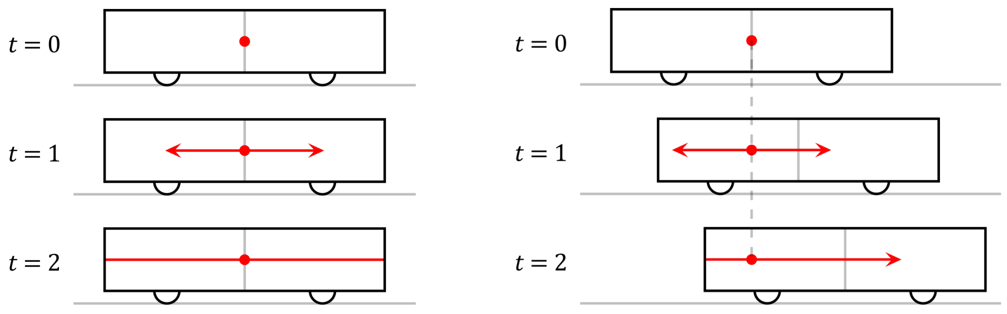
One can approximate the CIE 1931 XYZ spectrum using Gaussian curves for the wavelength spectra of each parameter. By shifting the centre of each Gaussian by the desired wavelength shift, the effect of a Doppler shift is approximately applied, after which the XYZ colour can be transformed back to RGB and bound such that the RGB spectrum has been shifted.

The details of approximating the shift have been omitted as those are a question of implementation. The method we used is identical to that of OpenRelativity [3] and differs even from the explanation given above in its numerics.

Appendix E

Train-and-Platform Example

A pedagogical example of the effects of relativity is Einstein's train-and-platform example. Consider two observers, one stationary on a train platform and another on a railcar moving parallel to the train platform at close to the speed of light. Imagine that there is a lightbulb in the middle of the railcar that emits a pulse of light just as the observers pass one another. The observer in the railcar will see the light pulse reach either end of the railcar simultaneously, as is expected. Keeping in mind that the speed of light is constant, however, the observer on the platform will see the pulse be emitted, then the railcar move forward, causing the emitted pulse to first hit the back of the railcar, followed by the front of the railcar sometime later.



(a) Observer inside railcar.

(b) Stationary observer.

Figure E.1: Relativity of Simultaneity

Light pulse on railcar as seen by observer inside railcar (a) and on the train platform (b) [18].

Appendix F

Transverse and Longitudinal Doppler Effects

The impact of the relativistic Doppler effect is twofold, the first of which occurs when the source of the emitted photons and the receiver are moving directly towards or away from one another. In this case, dubbed the longitudinal Doppler effect, the net Doppler shift is a combination of the traditional Doppler effect and an additional Lorentz factor to reflect the dilated time experienced by the source and receiver. The net equation is given in Eq. F.1 in the case where the source and receiver are moving towards one another.

$$\begin{aligned} f_r &= \gamma \cdot (1-\beta)\, f_s \\ &= f_s \sqrt{\frac{1-\beta}{1+\beta}}, \end{aligned} \tag{F.1}$$

where γ is the Lorentz factor of the two-frame system, $\beta = v/c$, v is the relative velocity of the two frames, c is the speed of light in vacuum, f_r is the frequency of the photons as measured by the receiver, and f_s is the frequency of the photons as measured by the source.

The transverse Doppler effect is most noticeable when the source and receiver are moving relative to one another at the instant of closest approach. In this case the emitted photons are blue-shifted when the emitter and receiver are at their actual points of closest approach, and are red-shifted then the source and receiver are at what the receiver perceives as the point of closest approach. These two cases are outlined in Eq. F.2 and Eq. F.3, respectively.

$$f_r = \gamma f_s \tag{F.2}$$

$$f_r = \frac{f_s}{\gamma} \tag{F.3}$$