VR Final Project Design Document

Betty Yang, Leo Geng, Serana Liu

Professor: Min Kwak

04/15/2024

Extended Reality: UX for AR/VR

Table of Contents

Project Summary

Overview

In this project, our main focus was to conceptualize, design, and develop an immersive Virtual Reality (VR) application aimed at uncovering scientific facts. With a strong emphasis on education and exploration, we aimed to create an engaging and interactive VR app that immerses users in the fascinating world of science.

Objective

The objective of this project is to design a VR product primarily for educational purposes, with a focus on uncovering intriguing scientific facts.

Scope

The scope of this project includes designing the VR app's user flow, visual design, and user experience. The project will involve conducting secondary research, competitive analysis, creating prototypes, and developing an engaging VR app development strategy that aligns with the needs of our target audience.

Duration

April 2024 — May 2024.

Phase 1: Discover

Research on the subject

Through our research efforts, we gained insights into how children can effectively learn physical concepts and the conditions that encourage their willingness to engage in learning.

Our findings indicate that children are naturally inclined to explore knowledge in entertainment. Specifically, we discovered that children prefer learning through interactive experiences, such as games. Furthermore, we also conducted additional research to identify the types of physical games that children are likely to enjoy and their levels of acceptance.

This information will serve as the foundation in the later stages of design, ensuring that our settings are tailored to meet the preferences and interests of our young learners.

Phase 2: Define

Key Insights:

Hands-on experimentation is important for people to learn physics, especially children.

Problem Statement:

How might we redefine the way children learn scientific concepts?

Phase 3: Ideation

Brainstorm Takeaway:

Through our brainstorming sessions, we decided to move forward with the idea of designing a VR app to uncover scientific facts.

In addition, we also gained valuable insights into the potential impact and significance of our VR app:

1. **Engaging and Interactive Learning**: Our VR application has the potential to revolutionize the way users engage with scientific content by providing an immersive and interactive learning experience. By incorporating hands-on exploration and interactive simulations, users are empowered to take control of their learning journey and deepen their understanding of complex scientific concepts.
2. **Practical Application of Knowledge**: Our VR application enables users to apply theoretical knowledge to practical situations by simulating real-world scenarios and practical experiments.

Research on the competitive products
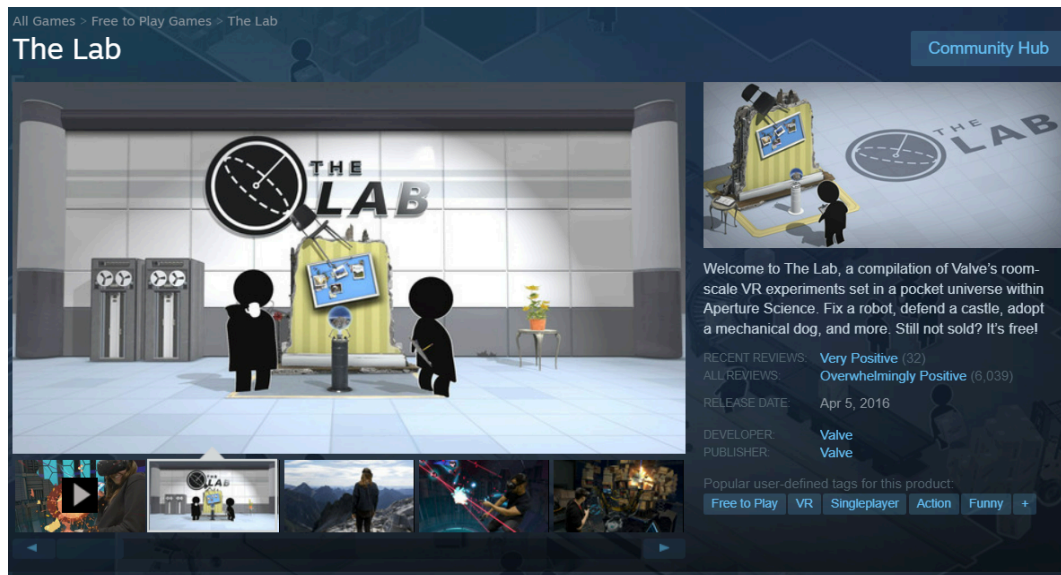
"The Lab" is a virtual reality (VR) compilation developed by Valve Corporation set within the universe of Aperture Science from the "Portal" series. It offers a diverse collection of VR experiences, each designed to showcase different aspects of VR gameplay and technology. Players can engage in activities like using a slingshot to cause havoc in Aperture's storage facility, defending a castle gate with archery skills, navigating a retro-style space shooter, exploring the human body through medical scans, touring the solar system, and even participating in robot repair tasks. The game combines humor, innovative gameplay mechanics, and immersive VR environments to provide a varied and entertaining experience.

Embibe Lab Experiments 4+

Virtual Lab Experiments App

Indiavidual Learning Limited

Designed for iPad

★★★★★ 4.9 • 50 Ratings

Free · Offers In-App Purchases

View in **Mac App Store** ↗

Screenshots iPad iPhone
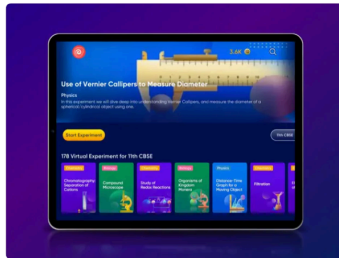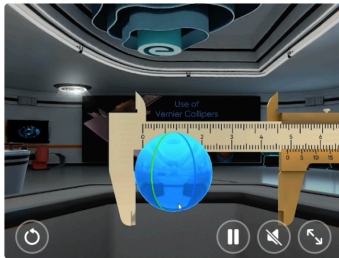
EMBIBE Experiments is an ideal VLE platform for science experiments. This app enables students of different boards (CBSE, ICSE & all state boards) to effortlessly perform Physics, Chemistry & Biology lab experiments at their own pace!

Embibe is an app that helps students get ready for tough exams in a smarter way. It uses fancy AI and analytics to understand how each student learns best. Then, it creates personalized study plans and gives recommendations based on what you need to focus on. Embibe also has fun interactive experiments, like the "Embibe Lab Experiments," to make learning more interesting. Overall, Embibe uses tech to make studying easier and helps you do your best in exams.

Phase 4: Design

Target Audience

Primary Users:

- Children and Teenagers aged 6 to 18

Secondary Users:

- Individuals who are passionate about physics

User flow

Flow Hierarchy:

1. The user launches the software.
2. Introduction: Engage with branding animation.
3. Gameplay: Enter the level and engage in experiments.
4. Object Selection: Explore available items and pick them up to experiment with.
5. Unlock the next level: By exploring the object, and finishing the setting experiment, the user can unlock the next level.
6. Settings: Explore different behavior settings or go back to the previous menu page.

User flowchart:
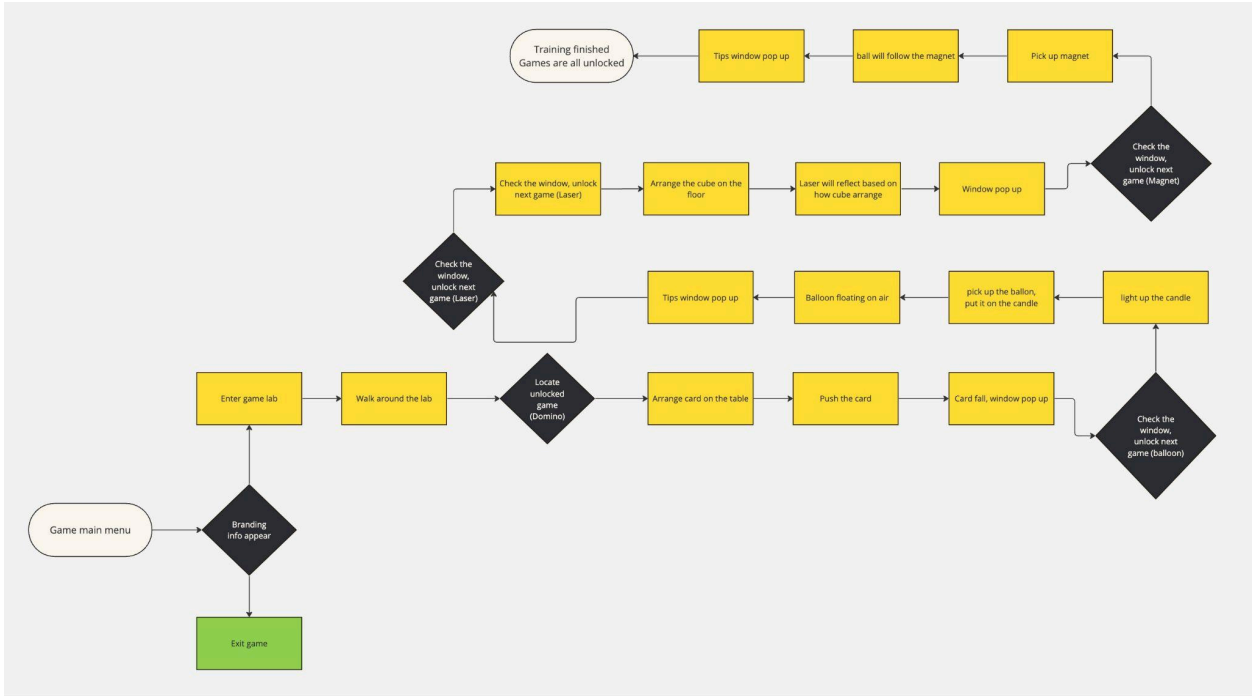
Sketches
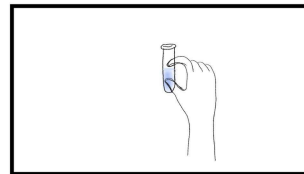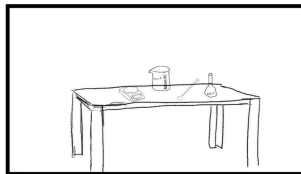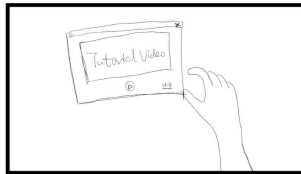
Key Features

1. **Detailed Explanations**: Users are provided with clear and concise explanations of scientific concepts, helping them understand complex topics engagingly.

2. **Interactive Simulations**: Users can engage in interactive simulations that allow them to manipulate objects, conduct experiments, and observe scientific principles in action, fostering hands-on learning and experimentation.

3. **Provide various Interactive Objects**: Allows users to learn various principles of physics, such as buoyancy, velocity, etc in a fun and interactive way by interacting with provided objects.

Design Challenge
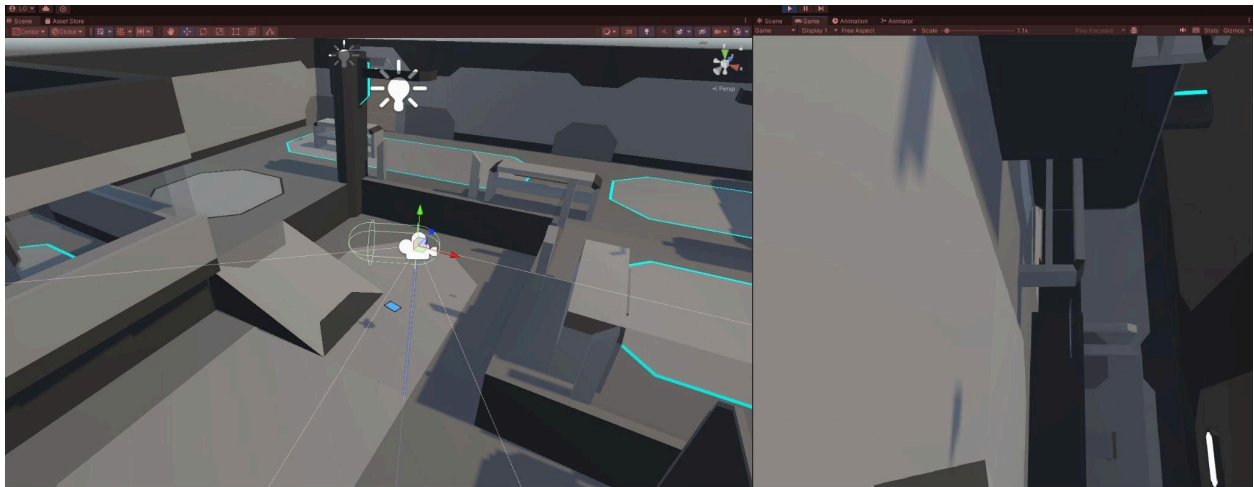
To make the physics lab interactable with the player's body, a scalable capsule was created, however, without freezing the rotation and giving the player scripted height, the "player" will simply fall because of the rigid body.
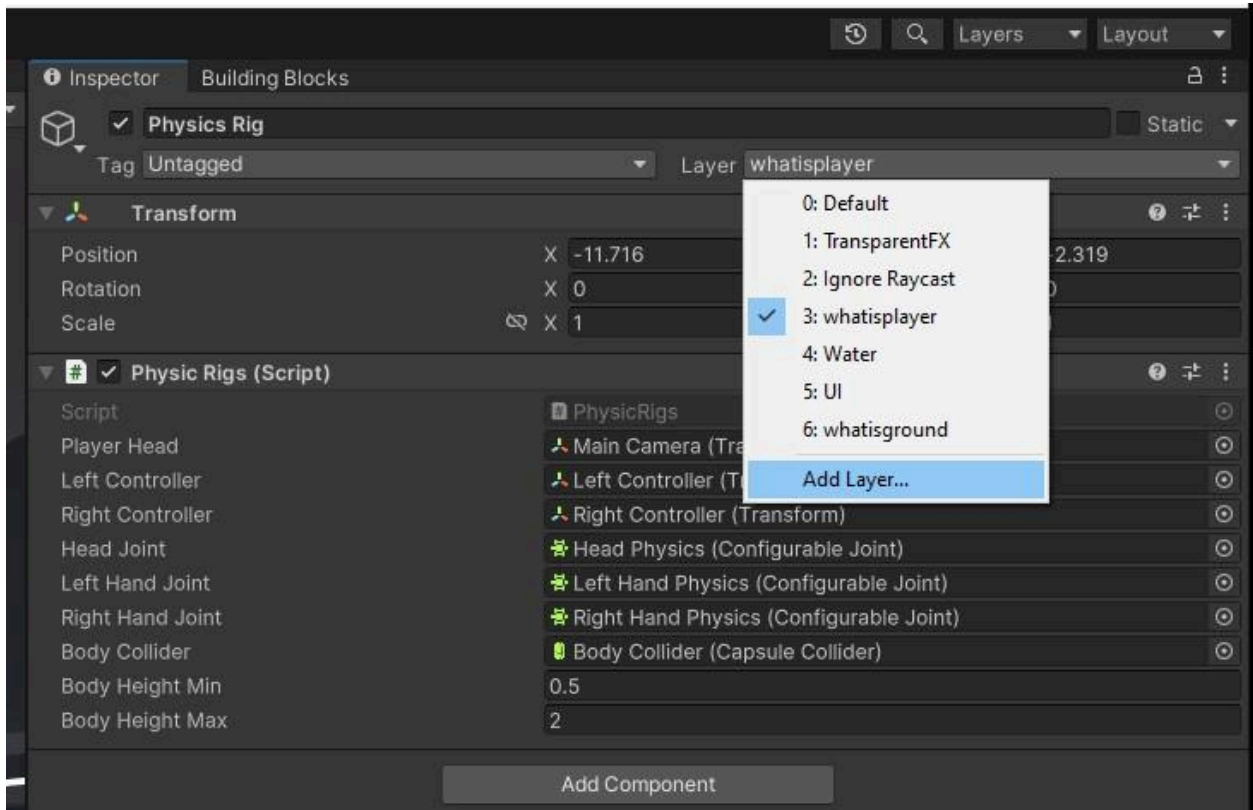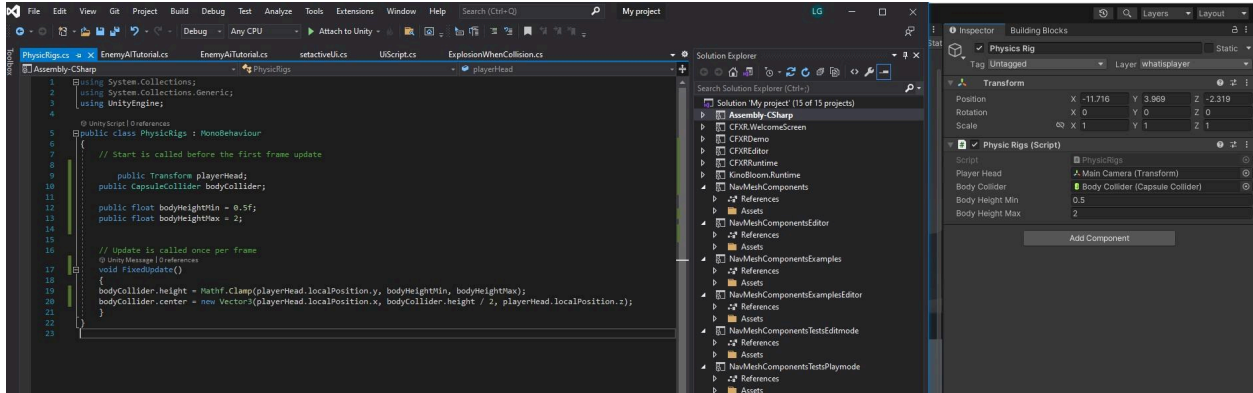
```csharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PhysicRigs : MonoBehaviour
{
    // Start is called before the first frame update

    public Transform playerHead;
    public CapsuleCollider bodyCollider;

    public float bodyHeightMin = 0.5f;
    public float bodyHeightMax = 2;

    // Update is called once per frame
    void FixedUpdate()
    {
        bodyCollider.height = Mathf.Clamp(playerHead.localPosition.y, bodyHeightMin, bodyHeightMax);
        bodyCollider.center = new Vector3(playerHead.localPosition.x, bodyCollider.height / 2, playerHead.localPosition.z);
    }
}
```

```csharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PhysicRigs : MonoBehaviour
{
    // Start is called before the first frame update

        public Transform playerHead;

    public Transform leftController;
    public Transform rightController;
    public ConfigurableJoint headJoint;
    public ConfigurableJoint leftHandJoint;
    public ConfigurableJoint rightHandJoint;



    public CapsuleCollider bodyCollider;

    public float bodyHeightMin = 0.5f;
    public float bodyHeightMax = 2;


    // Update is called once per frame
    void FixedUpdate()
    {
        bodyCollider.height = Mathf.Clamp(playerHead.localPosition.y, bodyHeightMin, bodyHeightMax);
        bodyCollider.center = new Vector3(playerHead.localPosition.x, bodyCollider.height / 2, playerHead.localPosition.z);

        leftHandJoint.targetPosition = leftController.localPosition;
        leftHandJoint.targetRotation = leftController.localRotation;
        rightHandJoint.targetPosition = rightController.localPosition;
        rightHandJoint.targetRotation = rightController.localRotation;

        headJoint.targetPosition = playerHead.localPosition;



    }

}
```

By giving the player(the capsule) a flexible height and freezing the horizontal rotation, the player is now able to "bump" into walls of different heights and climb ramps. Then Hand joints were added for more flexible interaction on rotating objects in hands.

Screenshot 1 — LaserSource.cs:

```csharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class LaserSource : MonoBehaviour
{
    [SerializeField] Transform laserStartPoint;
    Vector3 direction;
    LineRenderer lr;
    GameObject tempReflector;
    void Start()
    {
        lr = gameObject.GetComponent<LineRenderer>();
        direction = laserStartPoint.forward;
        lr.positionCount = 2;
        lr.SetPosition(0, laserStartPoint.position);
    }
    void Update()
    {
        RaycastHit hit;
        if (Physics.Raycast(laserStartPoint.position, direction, out hit, Mathf.Infinity))
        {
            if (hit.collider.CompareTag("Reflector"))
            {
                tempReflector = hit.collider.gameObject;
                Vector3 temp = Vector3.Reflect(direction, hit.normal);
                hit.collider.gameObject.GetComponent<LaserReflector>().OpenRay(hit.point, temp);
            }
            lr.SetPosition(1, hit.point);
        }
        else
        {
            if (tempReflector)
            {
                tempReflector.GetComponent<LaserReflector>().CloseRay();
                tempReflector = null;
            }
            lr.SetPosition(1, direction * 200);
        }
    }
}
```

Screenshot 2 — LaserReflector.cs:

```csharp
    void Start()
    {
        isOpen = false;
        lr = gameObject.GetComponent<LineRenderer>();
    }

    void Update()
    {
        if (isOpen)
        {
            lr.positionCount = 2;
            lr.SetPosition(0, position);
            RaycastHit hit;
            if (Physics.Raycast(position, direction, out hit, Mathf.Infinity))
            {
                if (hit.collider.CompareTag("Reflector"))
                {
                    tempReflector = hit.collider.gameObject;
                    Vector3 temp = Vector3.Reflect(direction, hit.normal);
                    hit.collider.gameObject.GetComponent<LaserReflector>().OpenRay(hit.point, temp);
                }
                lr.SetPosition(1, hit.point);
            }
            else
            {
                if (tempReflector)
                {
                    tempReflector.GetComponent<LaserReflector>().CloseRay();
                    tempReflector = null;
                }
                lr.SetPosition(1, direction * 100);
            }
        }
        else
        {
            if (tempReflector)
            {
                tempReflector.GetComponent<LaserReflector>().CloseRay();
                tempReflector = null;
            }
        }
    }

    public void OpenRay(Vector3 pos, Vector3 dir)
    {
        isOpen = true;
        position = pos;
        direction = dir;
    }
    public void CloseRay()
    {
        isOpen = false;
        lr.positionCount = 0;
    }

    void OnMouseDown()
    {
        screenPoint = Camera.main.WorldToScreenPoint(gameObject.transform.position);
        offset = gameObject.transform.position - Camera.main.ScreenToWorldPoint(new Vector3(Input.mousePosition.x, Input.mousePosition.y, screenPo
    }

    void OnMouseDrag()
    {
```

Scripting the laser and laser reflector, assigning light beam starting point and "reflector" tag.

```
29        private float inputTurnAxis;
30        // Start is called before the first frame update
31
32        // Update is called once per frame
          Unity Message | 0 references
33        void Update()
34        {
35            inputMoveAxis = moveInputSource.action.ReadValue<Vector2>();
36            inputTurnAxis = turnInputSource.action.ReadValue<Vector2>().x;
37
38            bool jumpInput = jumpInputSource.action.WasPressedThisFrame();
39        }
40
          Unity Message | 0 references
41        private void FixedUpdate()
42        {
43            bool isGrounded = CheckIfGrounded();
44            if (isGrounded)
45            {
```
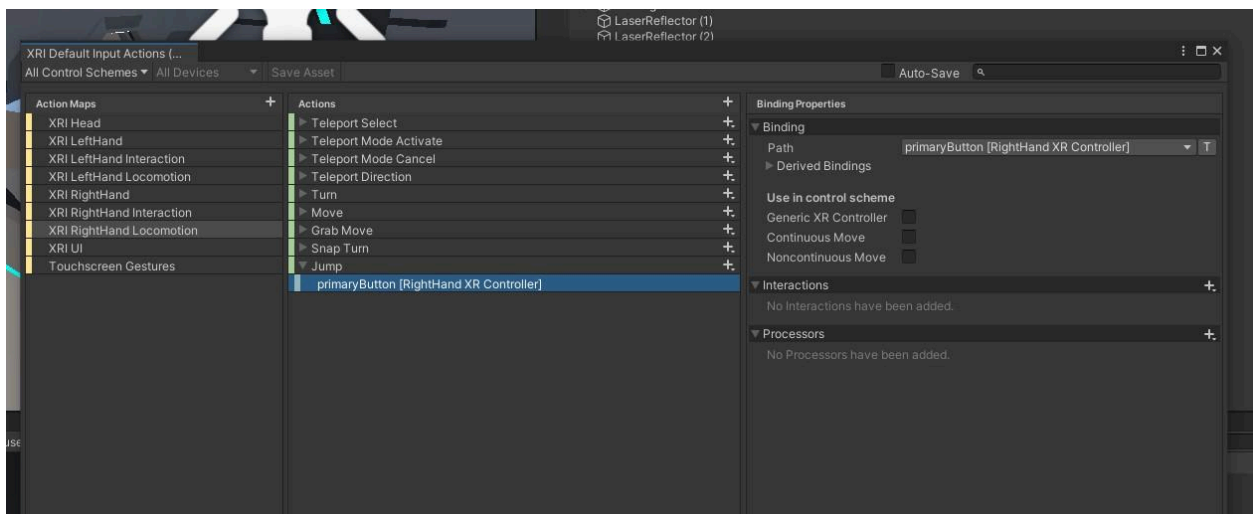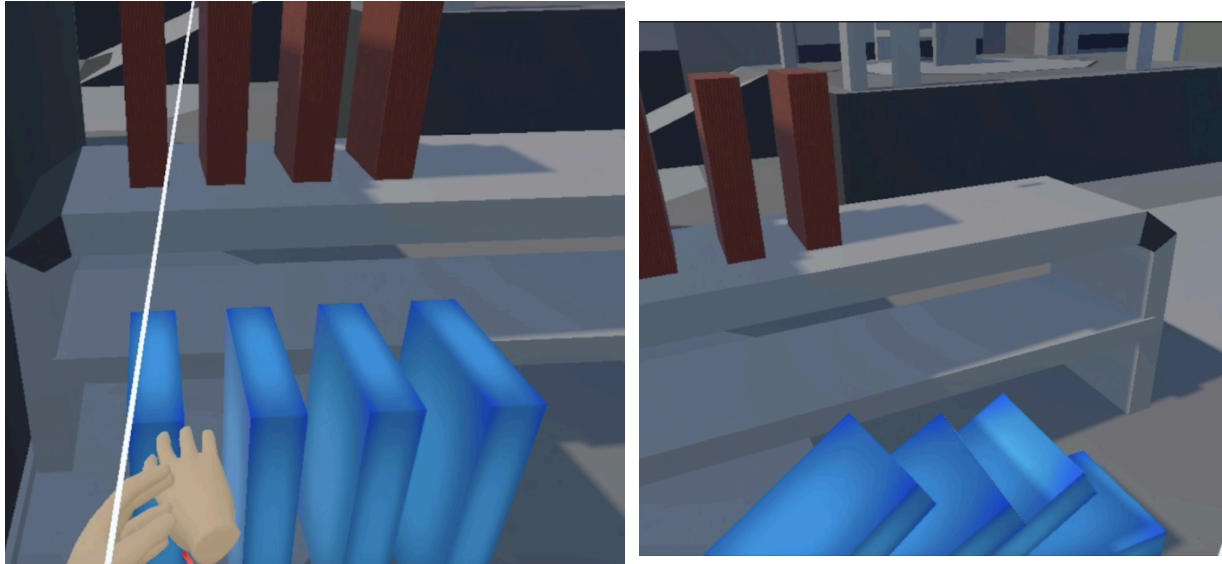

Adding Jump by scripting lefthand control primary button.

We have three levels in this lab game, from the easy to the hardest. Each level will teach the user physical knowledge via playing the game and interacting with the object in our game. From the simplest level: in a domino game, users can arrange the domino by themselves and push them to fall.
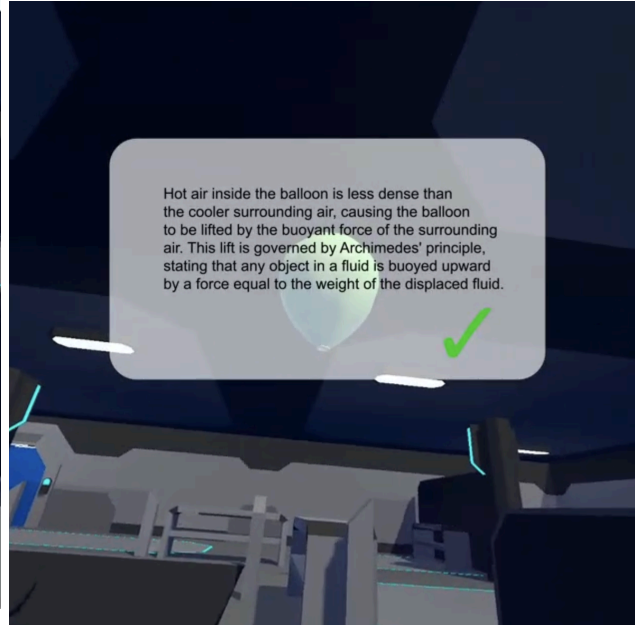

When all domino cards fall, a window will pop up and some knowledge based on this game will be shown for users to study. Users click the "checkmark" and they can be led to the next level of this study process.

As dominoes are lined up and toppled, they showcase principles such as potential energy converting to kinetic energy. The falling dominoes also illustrate concepts of momentum transfer, where the force exerted by one domino causes the next to fall, and this chain reaction exemplifies Newton's laws of motion.

Please go explore next experiment, heading to your right!

The next experiment will be: the user lights up the candle that we provide and picks up the envelope, places it on the top of the envelope, and then the balloon will rise because of the physical effect.
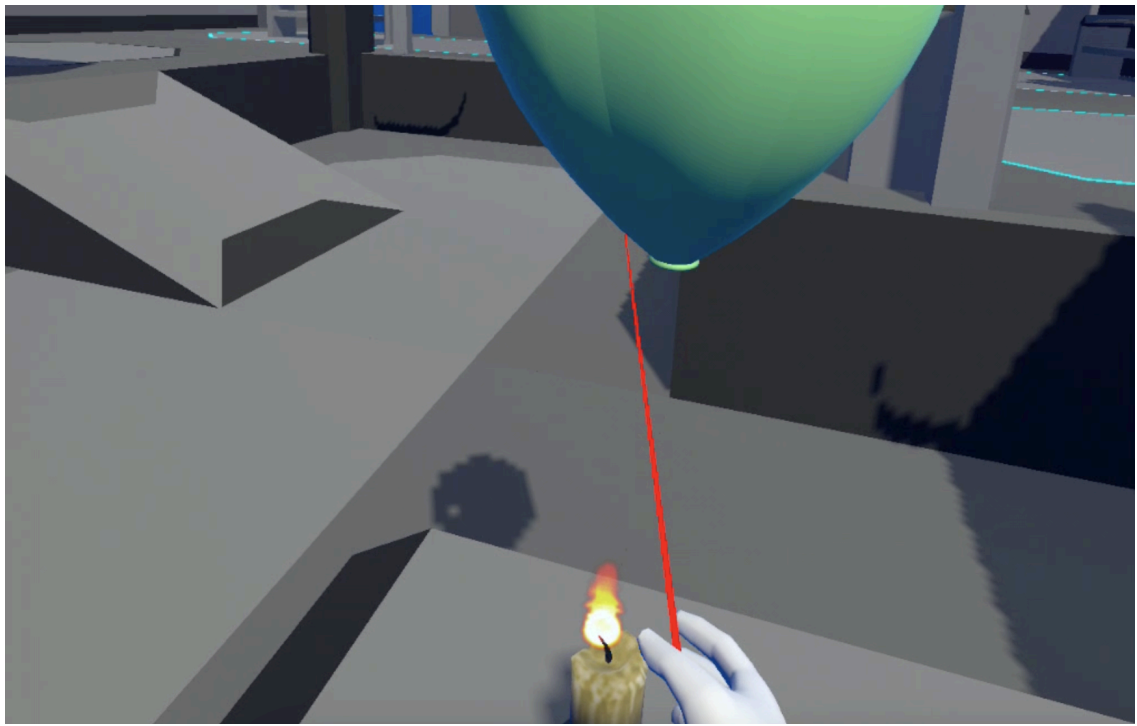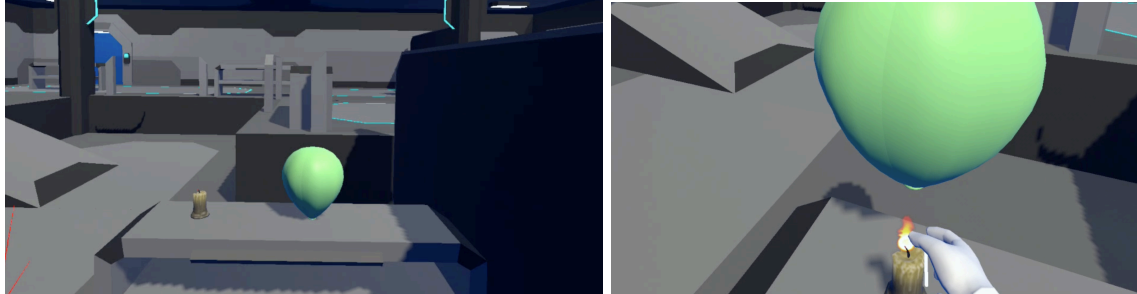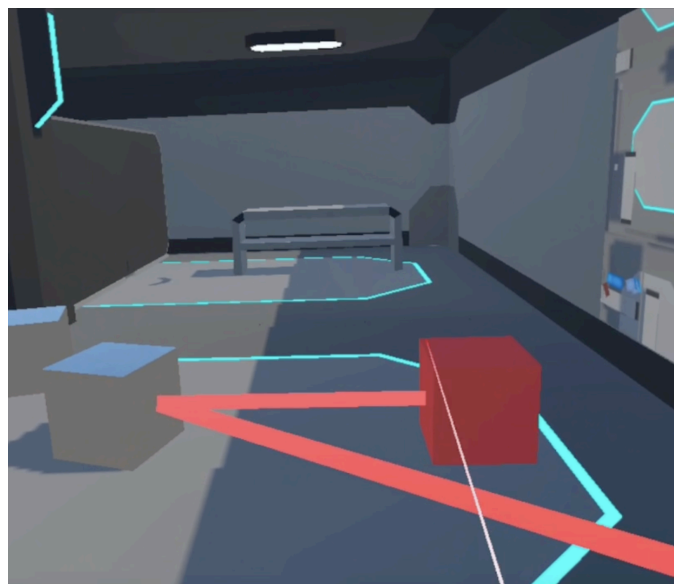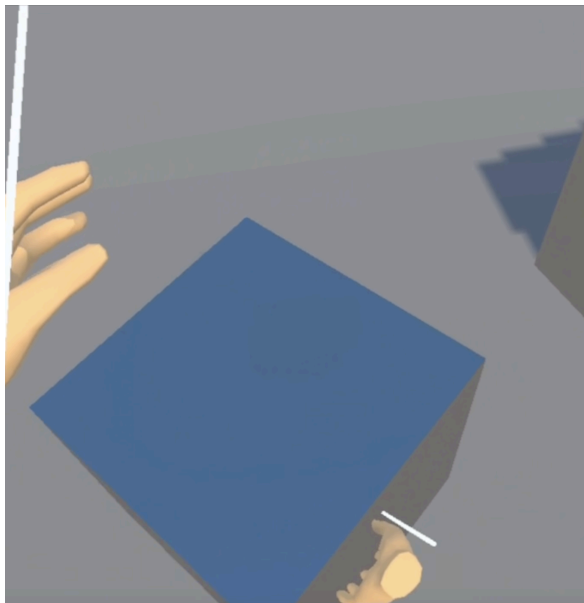

When the balloon floats in the air for a few seconds, a window will pop up and some knowledge based on this game will be shown for users to study.
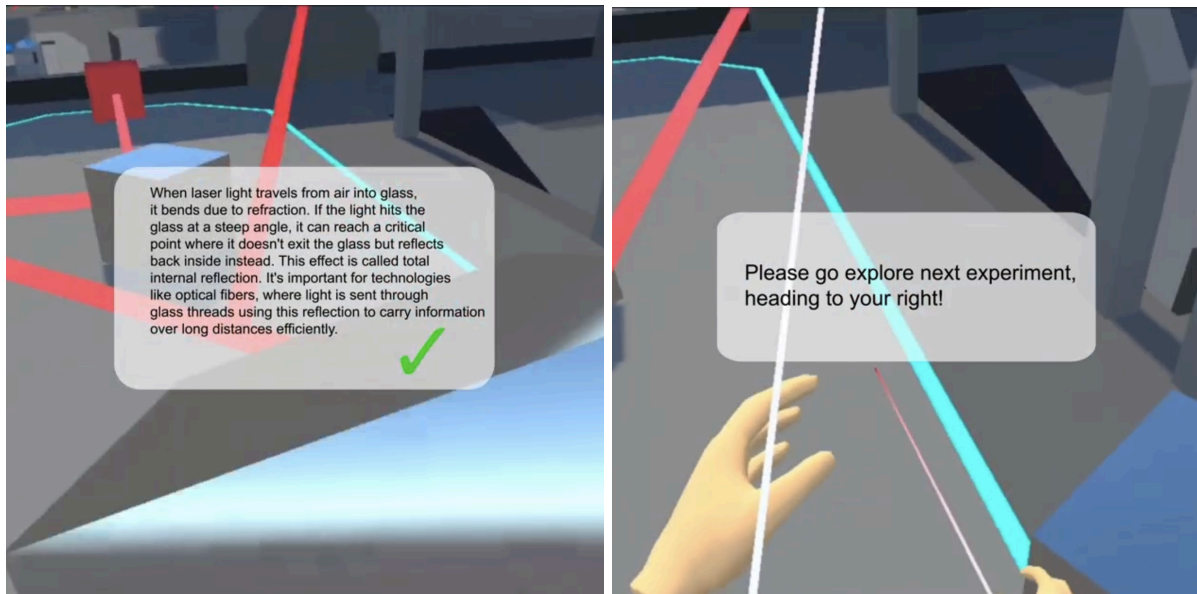
Users who click the "checkmark" can be led to the next level of this study process.

The next experiment will be: users pick up the cube (has glass texture) and arrange them on the floor by themselves. The laser beam will reflect the laser light based on how the user arranges the cube.
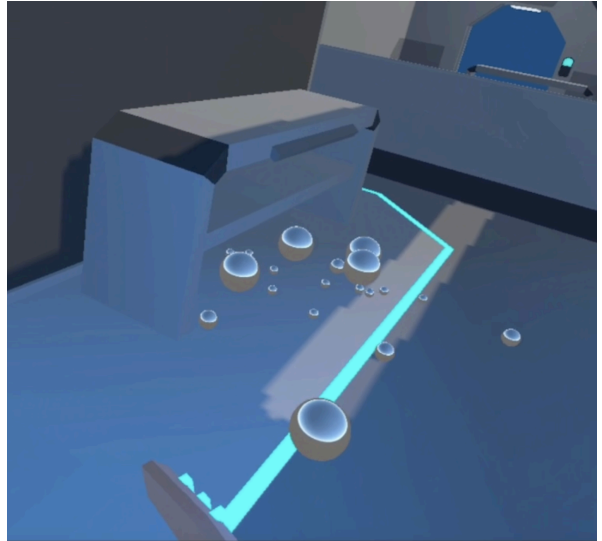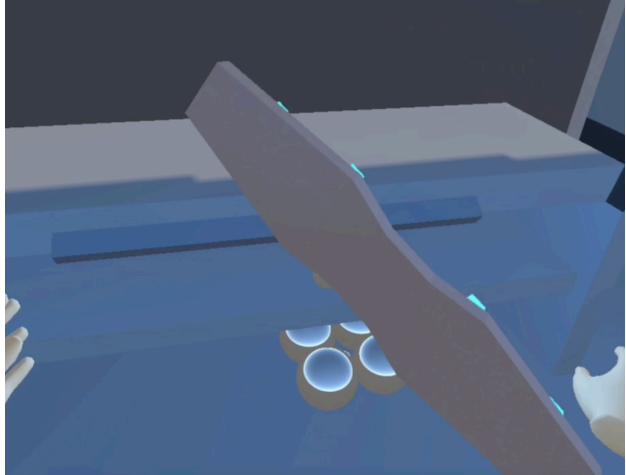
The use will be to explore how the lasers will be reflected on different angles of the glass cube. A window will pop up and some knowledge based on this game will be shown for users to study.

When laser light travels from air into glass, it bends due to refraction. If the light hits the glass at a steep angle, it can reach a critical point where it doesn't exit the glass but reflects back inside instead. This effect is called total internal reflection. It's important for technologies like optical fibers, where light is sent through glass threads using this reflection to carry information over long distances efficiently.

Please go explore next experiment, heading to your right!

Users who click the "checkmark" can be led to the next level of this study process.

The final level is a magnet experiment. The user picks up the big magnet on the floor and waves the magnet. The small iron balls on the floor will start to move in different directions because of the magnetic force.

A window will pop up and some knowledge based on this game will be shown for users to study.

Inside a magnet, there are regions called magnetic domains where groups of atoms align in the same direction, creating a magnetic field. When a magnet comes near an iron object, like a paperclip, the magnetic field of the magnet exerts a force on the iron atoms in the paperclip. This force causes the iron atoms to align temporarily with the magnetic field of the magnet, causing the paperclip (or other iron object) to be attracted to the magnet.

Congratulations! All the games are unlocked! Please keep trying out in the lab.