Immersive Google Classroom with Interactive PPT and Gaze-Controlled Navigation

In the rapidly evolving world of educational technology, Virtual Reality (VR) is transforming traditional learning experiences by bringing the classroom into the virtual realm. This project exemplifies this innovation by presenting an immersive virtual classroom, built using Unity and optimized for Google Cardboard. Designed to enhance the educational experience, this VR environment integrates interactive features that allow users to engage with content in a dynamic and intuitive way.

Key Features:

- ❖ **Immersive Classroom Environment**
  The virtual classroom features realistic 3D models, providing a lifelike setting that immerses students in the learning experience, enhancing focus and engagement.
- ❖ **Interactive PowerPoint Navigation**
  With gaze-based controls, users can seamlessly navigate through PowerPoint slides without the need for physical controllers, making the learning process more intuitive and fluid.
- ❖ **Seamless VR Integration**
  Optimized for Google Cardboard, this VR experience ensures smooth, lag-free performance, making advanced educational technology accessible and affordable for a wide audience.
- ❖ **User-Friendly Interface**
  The interface is designed for simplicity, featuring intuitive navigation buttons that allow users of all experience levels to interact easily with the virtual environment.

Advantages

- ❖ **Enhanced Engagement**
  The immersive virtual environment captivates students' attention, making learning more interactive and enjoyable.
- ❖ **Accessible Learning**
  Optimized for Google Cardboard, this VR experience is both cost-effective and accessible, removing geographical barriers and providing a broad reach.
- ❖ **Future-Ready Technology**
  This VR solution is at the cutting edge of educational technology, preparing users for future advancements in virtual learning environments.
- ❖ **Intuitive Interaction**
  Gaze-based controls for PowerPoint navigation offer a hands-free, user-friendly experience that simplifies lesson delivery and student interaction.

Installation

1. Unity 2018.1.1f Personal:

- **Download and Install Unity:**
  - Visit the Unity Releases Archive.
  - Locate version 2018.1.1f Personal and download the installer for your operating system.
  - Run the installer and follow the on-screen instructions to complete the installation.

2. Google VR SDK:

- **Download and Import Google VR SDK:**
  - Go to the Google VR SDK releases page.
  - Download the Google VR SDK package (v1.130.0) for Unity.
  - Open Unity, go to Assets > Import Package > Custom Package... and select the downloaded Google VR SDK package to import it.

3. Android SDK:

- **Download and Install Android SDK:**
  - Visit the Android Studio download page.
  - Download and install Android Studio, which includes the Android SDK.
  - Open Android Studio and follow the setup wizard to install the required SDK components.

4. Android NDK:

- **Download and Setup Android NDK:**
  - Visit the Android NDK Releases.
  - Choose version r13b and download the appropriate package for your operating system.
  - Extract the downloaded NDK package to a location of your choice.

5. Java:

- **Download and Install Java (OpenJDK):**
  - Visit the OpenJDK Download.
  - Select version jdk-8.0.332.09-hotspot and download the installer suitable for your operating system and Run the installer and follow the on-screen instructions to complete the installation.

Developing a Scene in Unity

1. Importing School Free Assets:

- **Download and Import Assets:**
  - Open Unity and go to the Asset Store (Window > Asset Store).
  - Search for "school free assets" and download the desired assets.
  - Import the downloaded assets into your Unity project.

2. Building a Classroom:

- **Create the Classroom Environment:**
  - Use the imported assets to build a classroom scene. Add elements like chairs, racks, walls, floor, and a projector to the scene.
  - Position and arrange the assets to create a realistic classroom environment.

3. Creating Interactive PPT Navigation:

- **Prepare and Import PPT Images:**
  - Convert your PowerPoint slides into image format (e.g., PNG or JPG).
  - Import the images into Unity by dragging them into the Assets folder.
- **Add Navigation Buttons:**
  - Create two UI buttons (green for "Next" and red for "Previous") using the Unity game objects.
  - Position the buttons in the scene where they are easily visible to the user.

4. Setting Up Google VR SDK for Interaction:

- **Implement Gaze-Based Navigation:**
  - Use the Google VR SDK to enable gaze-based interaction. When the user looks at a button for a specified duration, it triggers a click event.
  - Create scripts to handle the navigation logic. When the user gazes at the "Next" button, it moves to the next slide, and when gazing at the "Previous" button, it goes back to the previous slide.

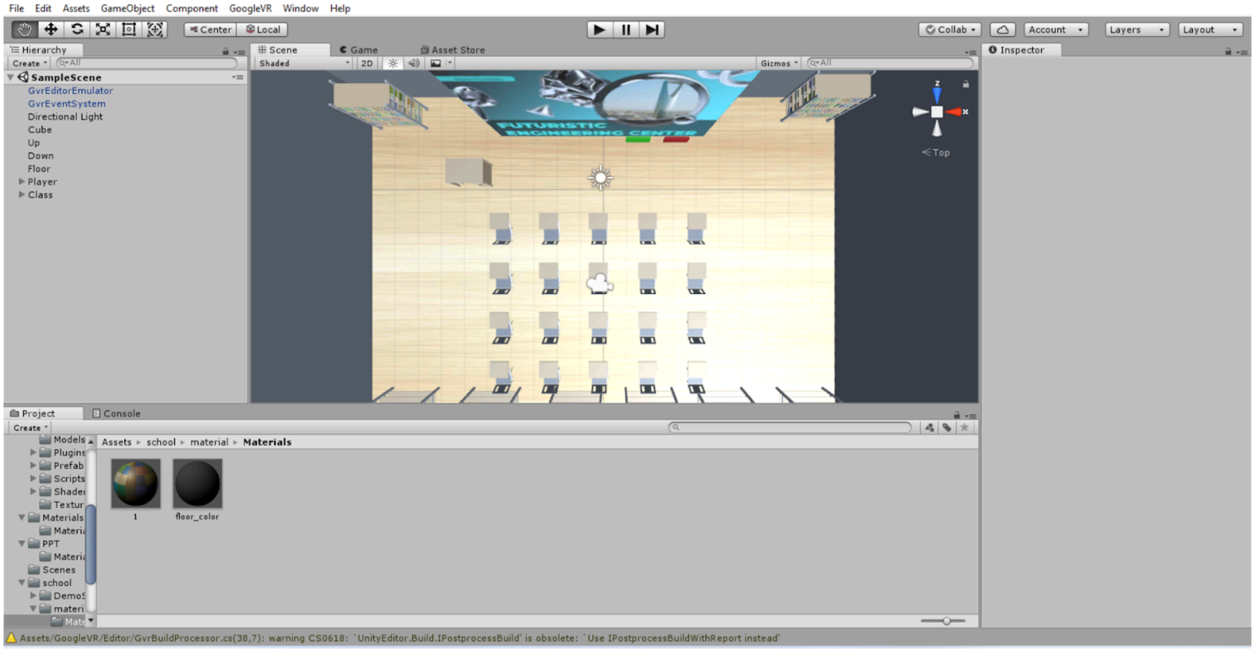
Script - SlideController

```
using UnityEngine;

public class SlideController : MonoBehaviour
{
    // Array to store references to slide materials
    public Material[] slideMaterials;

    // Reference to the screen object whose material will change
    public Renderer screenRenderer;

    // Singleton instance
    private static SlideController instance;

    // Static variable to store the current slide index
    private static int currentSlideIndex = 0;

    void Start()
    {
        // Ensure the screen renderer is assigned
        if (screenRenderer == null)
        {
            Debug.LogError("Screen Renderer is not assigned!");
            return;
        }

        // Check if there are slide materials
        if (slideMaterials.Length == 0)
        {
            Debug.LogError("No slide materials assigned!");
            return;
        }

        // Set the material of the screen to the initial slide
        screenRenderer.material = slideMaterials[currentSlideIndex];
    }

    // Method to change to the next slide
    public void NextSlide()
```

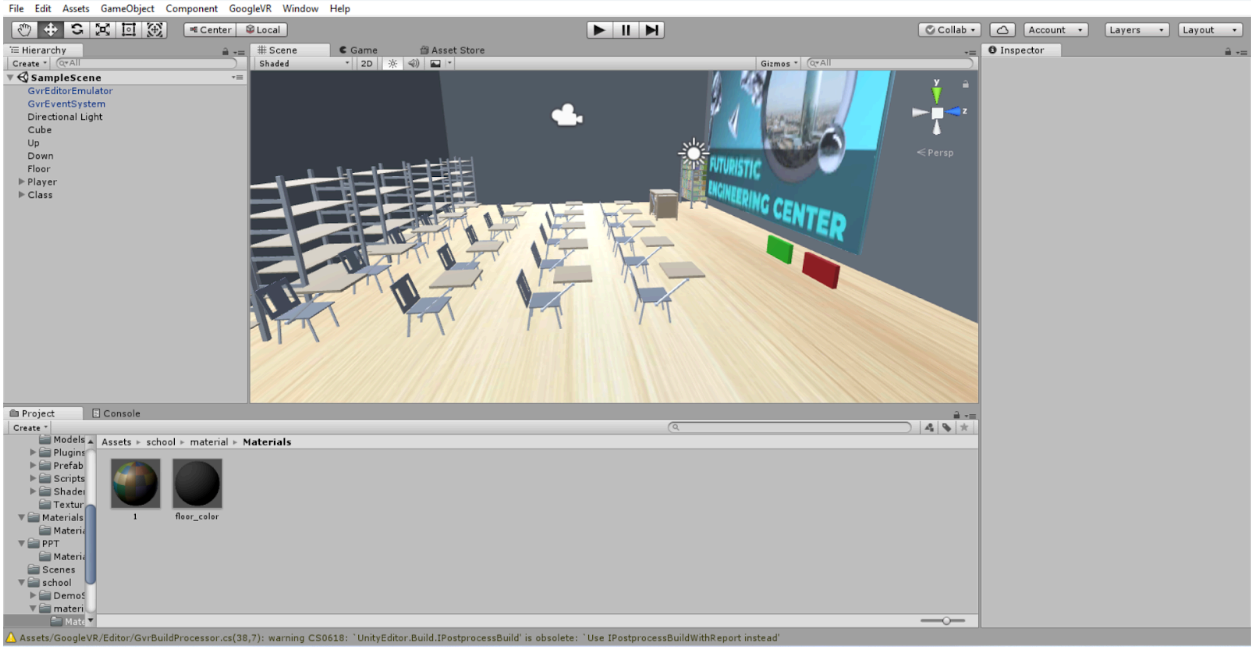
```
{
    // Increment the slide index
    currentSlideIndex++;

    // If we reach the end, loop back to the first slide
    if (currentSlideIndex >= slideMaterials.Length)
    {
        currentSlideIndex = 0;
    }

    // Update the slide for all instances
    UpdateSlideForAllInstances();
}

// Method to change to the previous slide
public void PreviousSlide()
{
    // Decrement the slide index
    currentSlideIndex--;

    // If we reach the beginning, loop to the last slide
    if (currentSlideIndex < 0)
    {
        currentSlideIndex = slideMaterials.Length - 1;
    }

    // Update the slide for all instances
    UpdateSlideForAllInstances();
}

// Method to update the slide for all instances
private void UpdateSlideForAllInstances()
{
    // Find all instances of SlideController in the scene
    SlideController[] slideControllers = FindObjectsOfType<SlideController>();

    // Update the slide index for each instance
    foreach (SlideController controller in slideControllers)
    {
        controller.screenRenderer.material = slideMaterials[currentSlideIndex];
```

```
            }
        }
    }
```

Unity Screenshots



Assets/GoogleVR/Editor/GvrBuildProcessor.cs(38,7): warning CS0618: `UnityEditor.Build.IPostprocessBuild' is obsolete: `Use IPostprocessBuildWithReport instead'



Assets/GoogleVR/Editor/GvrBuildProcessor.cs(38,7): warning CS0618: `UnityEditor.Build.IPostprocessBuild' is obsolete: `Use IPostprocessBuildWithReport instead'

Building and Testing a Unity Scene for Android with Google VR

1. Finalizing the Unity Scene:

- **Ensure Scene Completeness:**
    - Verify that all assets are correctly placed and the interactive elements (buttons, slides, etc.) are functioning as expected.
    - Check for any errors or warnings in the Console window and resolve them.

2. Setting Up the Build:

- **Configure Build Settings:**
    - Go to File > Build Settings.
    - Select Android as the platform and click Switch Platform.
    - Add the current scene to the "Scenes in Build" by clicking Add Open Scenes.
- **Player Settings:**
    - Click on Player Settings in the Build Settings window.
    - Under Other Settings, ensure the Minimum API Level is set to Android 8.0 (API level 26).
    - Set the Package Name (e.g., com.yourcompany.yourapp).
    - Enable XR Settings and check Virtual Reality Supported, then add Cardboard under Virtual Reality SDKs.

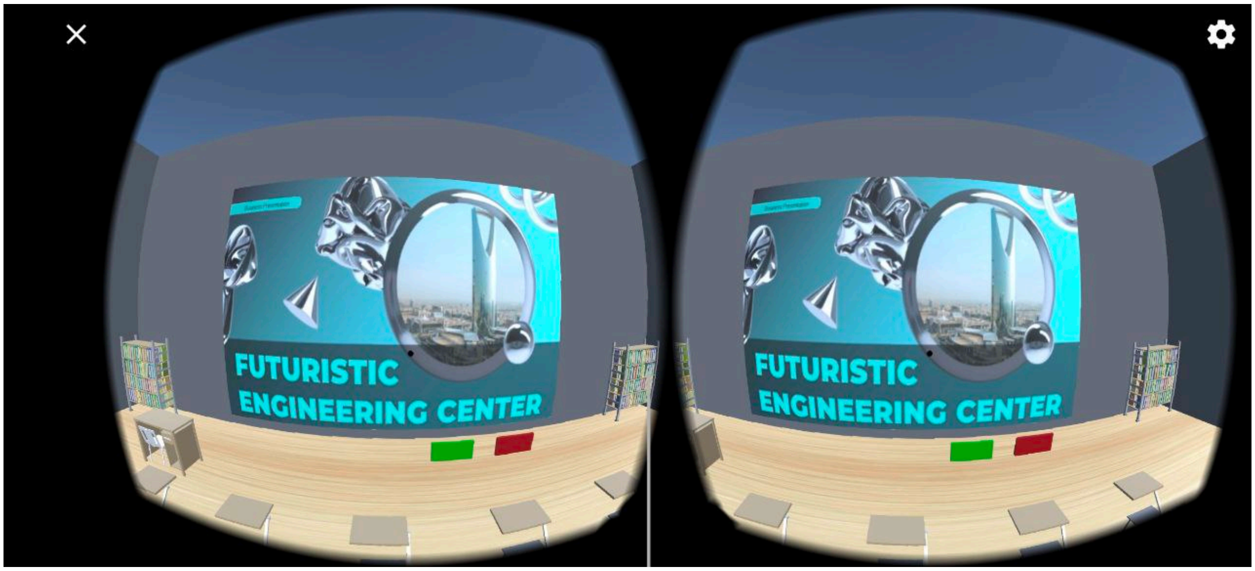
3. Building the APK:

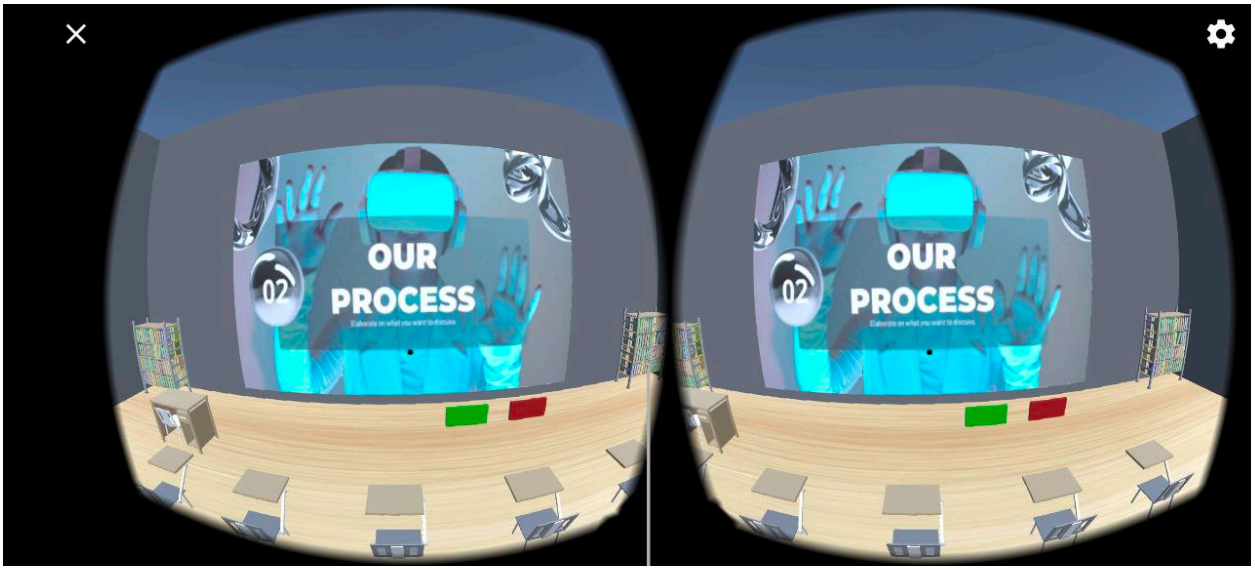
- **Build the Application:**
    - In the Build Settings window, click Build.
    - Choose a location to save the APK file and click Save.
    - Unity will compile the project and generate an APK file.

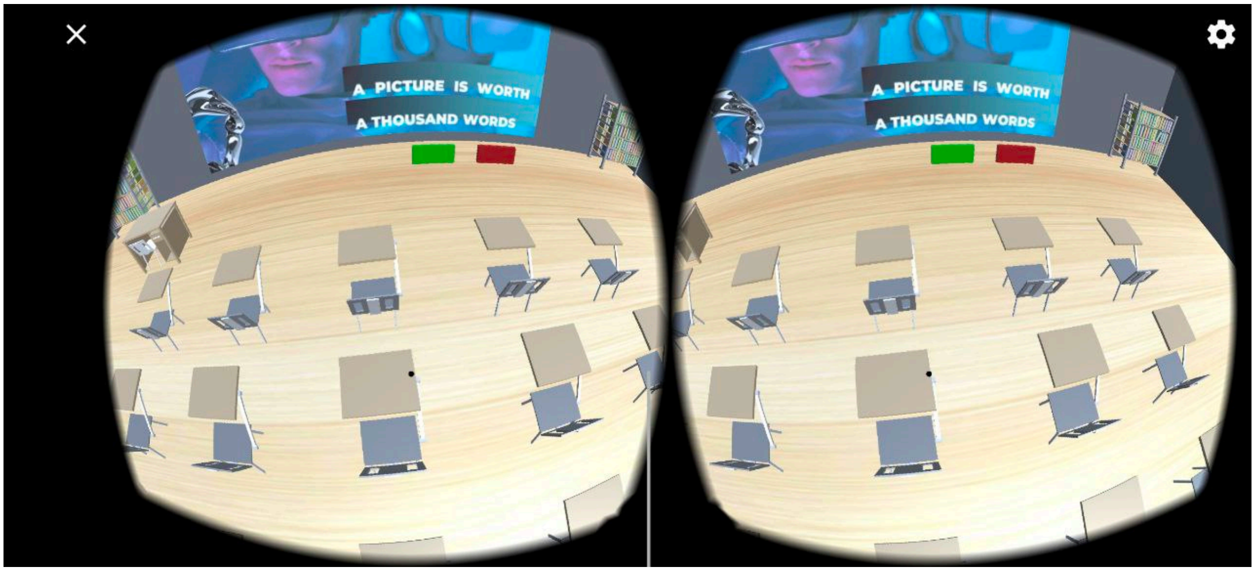
Testing the App on an Android Device:

- **Install the APK:**
    - Connect your Android device to your computer via USB.
    - Enable USB Debugging on your Android device (Settings > Developer Options > USB Debugging).
    - Transfer the APK file to your Android device.
- **Install the APK on the Device:**
    - Use a file manager on your Android device to locate the transferred APK file.
    - Tap the APK file and follow the prompts to install it.
- **Test with Google Cardboard:**
    - Launch the installed app on your Android device.
    - Place the device into the Google Cardboard headset.
    - Interact with the app to ensure the VR features, navigation, and overall functionality are working correctly.

Mobile App Screenshots




02
OUR PROCESS
Elaborate on what you want to discuss


A PICTURE IS WORTH
A THOUSAND WORDS