Virtual Rebirth- Visualization of Lost Monuments Using VR

Shreenandan Sahu | CL24M013

K S Suraksha Jain | CL24M015

AM5011: Virtual Reality Engineering

Under the guidance of

Dr. M. Manivannan

Department of Applied Mechanics & Biomedical Engineering

Indian Institute of Technology Madras

Contents

List Of Figures

Introduction

Cultural heritage is a vital link to our past, providing insights into the architectural marvels and traditions of earlier civilizations. Over time, several historic monuments have been partially or completely lost due to natural calamities, human intervention, and the passage of time. The Konark Sun Temple in Odisha, India, is one such example. This UNESCO World Heritage site, renowned for its intricate carvings and majestic design, has lost significant portions of its original structure. Preserving and presenting such heritage is a challenge that demands innovative solutions.

In this project, we aim to leverage Virtual Reality (VR) technology to digitally reconstruct and present the lost parts of the Konark Sun Temple. By creating an immersive VR experience, we not only preserve the temple's architectural legacy but also provide a platform for education and tourism. This approach is cost-effective and feasible, serving as a modern method to protect and promote cultural heritage. The resulting VR model will allow users to explore and interact with the temple's reconstructed elements, gaining a deeper understanding of its historical and cultural significance.

Figure 1 Comparison of Old (left) and Current (right) structure of Konark Temple.

Objective

- Reconstruct the lost and damaged parts of the Konark Sun Temple as a detailed and accurate 3D model.
- Develop a custom hand-held device to enable intuitive interaction and navigation around the 3D model in the virtual reality environment.
- Enhance the virtual experience by adding realistic textures to the temple reconstruction, creating a dynamic skybox to replicate natural surroundings, and integrating spatial audio to simulate the authentic ambiance, ensuring a visually and acoustically immersive environment.
- Provide an immersive, educational experience that promotes cultural preservation and attracts tourism, allowing users to explore the architectural and historical significance of the monument.

Block diagram & description

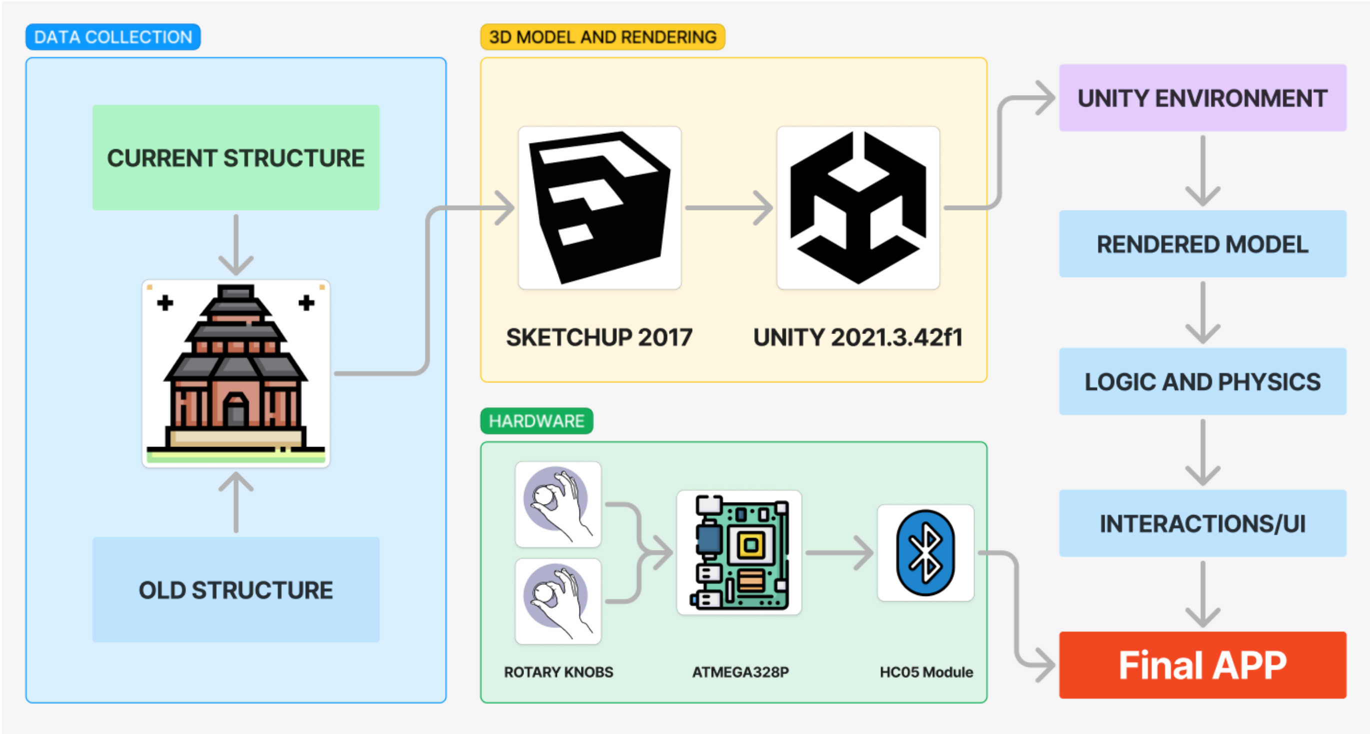

Figure 2 Flowchart describing the rough idea of making the Project.

Data Collection and 3D Modelling

- **Research and Analysis**: Historical references, photographs, and dimensional data from Google Maps were used to reconstruct the temple's architecture and proportions.
- **3D Modelling**: Using SketchUp 2017, the intricate details and proportions of the temple were meticulously modelled to ensure accuracy.
- **Rendering in Unity**: The model was imported into Unity 2021.3.42f1, where textures, lighting effects (including a static skybox and directional/point lighting), and interactive elements were applied to enhance realism and immersion.

VR Environment Development

- **XR Integration**: An XR Rig was added in Unity to facilitate interaction within the VR environment.
- **Physics and Interaction**: Logic and collision detection algorithms were implemented to provide realistic navigation and interaction within the VR space.

Hardware Integration

- A custom handheld hardware device was developed to enhance user interaction in the VR space:
  - **Components**: Rotary encoder, ATMEGA328P microcontroller, and HC05 Bluetooth module.
  - **Control Commands**: Encoded directional commands allow intuitive navigation:

- **Bluetooth Integration**: A custom plugin enables seamless connection with the VR application.

Application Development

- A functional Android application (.apk) was created to deliver the VR experience. This app integrates the 3D model, hardware controls, and immersive VR features to allow users to explore the temple in an engaging way.

This VR application bridges the gap between historical preservation and modern technology, offering an immersive and interactive experience of one of India's architectural marvels.

Implementation/ Methodology

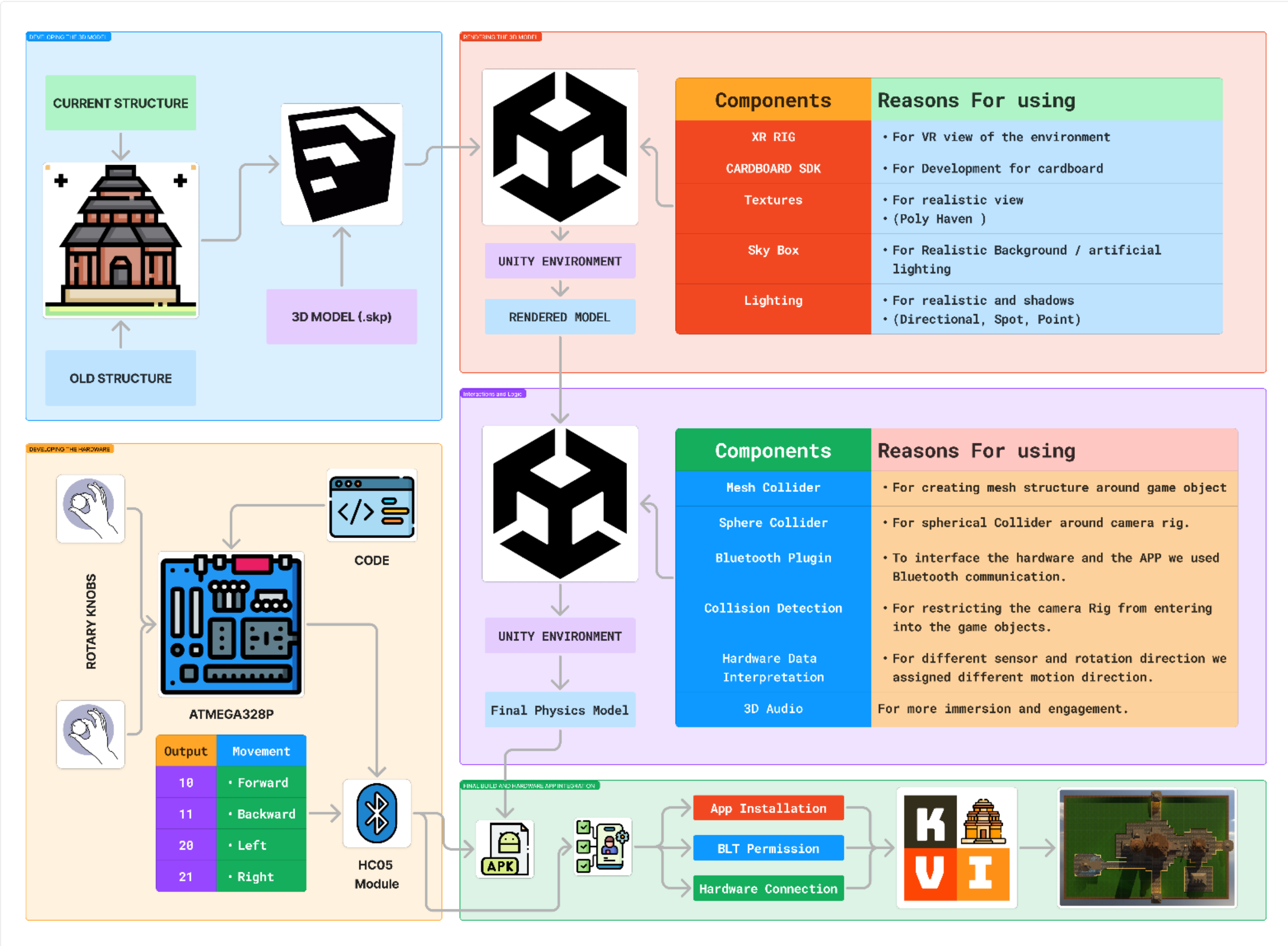

Figure 3 Flowchart Explains how the project is divided into different parts and how they are achieved.

3D Model Design

We have used SketchUp 2017 version to design the entire monument by making different components and tagging them differently. We took the help of different archive sites, old scriptures, local temple structures to get the dimensions and the ratios of the temple. The same dimension and ratios are maintained in the 3D model also.

Rendering the Model in Unity

To render the model in Unity, we followed these steps:

1. Importing the 3D Model:
   - We imported the. skp model into Unity and adjusted its scale and position to fit the scene.
2. Adding Textures:
   - Textures were sourced from Poly Haven and applied to the model's surfaces for a realistic look.
   - Different materials were assigned to specific components for accurate visual representation.
3. Skybox:
   - A skybox was added to provide a realistic background and improve the overall ambiance of the scene.
4. Lighting:
   - Directional lighting was used for global illumination across the entire scene.
   - Spotlights were added to focus on specific areas where the directional light was insufficient.
   - Shadows were enabled to enhance realism.
5. Colliders:
   - Mesh colliders were applied to match the model's geometry, enabling interaction and preventing overlap.
   - Sphere colliders were added to define interaction zones for dynamic elements.
6. XR Rig Integration:
   - The XR Rig feature was used to manage the camera's position and angles for VR viewing.

Interaction and Logic

For interaction and logic in the Unity environment, we implemented the following components:

1. Mesh Collider:
   - Used to define the mesh structure around game objects, allowing precise collision detection based on their shape.
2. Sphere Collider:
   - Added around the camera rig to create a spherical interaction zone for detecting proximity-based collisions.
3. Bluetooth Plugin:
   - Integrated to enable communication between the hardware and the Unity application, allowing real-time input and interaction.
4. Collision Detection:
   - Implemented to prevent the camera rig from passing through game objects, maintaining realistic movement and interaction boundaries.
5. Hardware Data Interpretation:
   - Mapped input data from sensors to specific motion directions (e.g., forward, backward, left, right) for smooth navigation and interaction within the scene.
6. Final Physics Model:
   - Applied physics principles to ensure realistic movements, interactions, and collisions within the environment.
7. 3D Audio:
   - Incorporated spatial sound effects to enhance immersion, aligning audio cues with in-game events and interactions.

Hardware Development

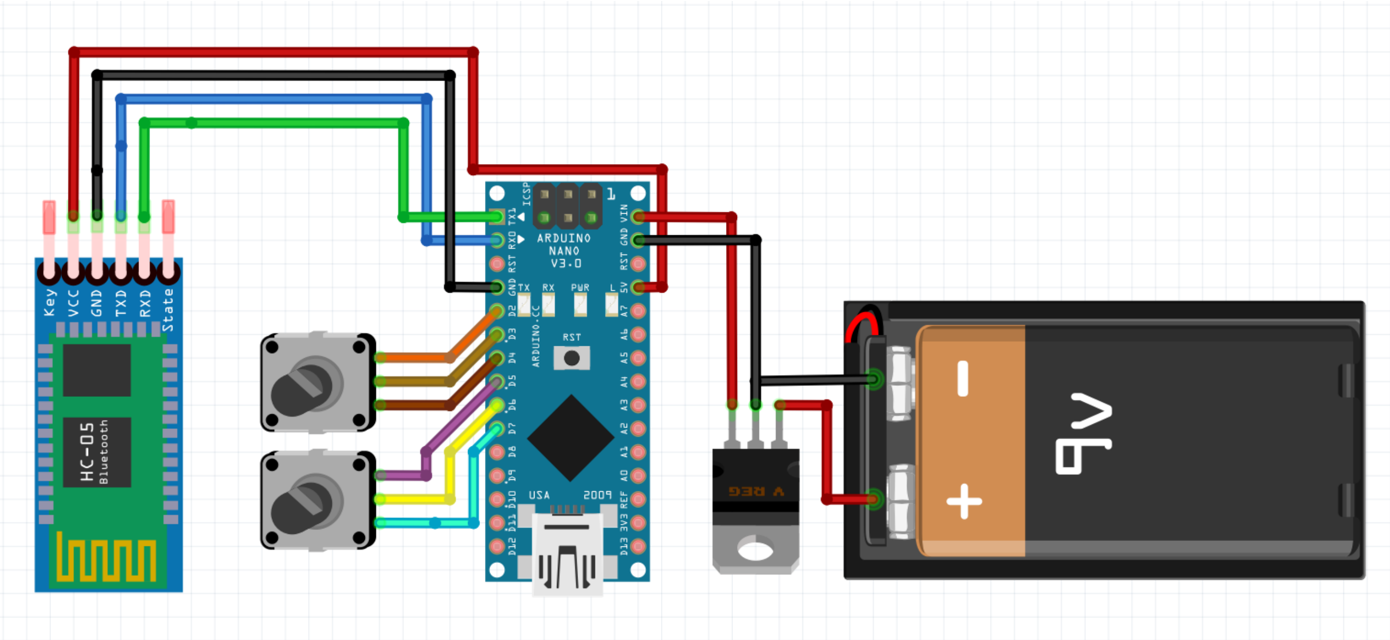

Figure 4 Schematics of the Hardware describing the connections with components.

1. Power Supply: A 9V battery is connected to the Arduino Nano via a voltage regulator to provide a stable 5V supply.
2. Arduino Nano: Acts as the microcontroller for processing inputs and controlling outputs.
3. Rotary Encoders: Two rotary knobs are connected to the Arduino's digital pins to detect rotation and direction.
4. HC-05 Bluetooth Module:
   - VCC and GND are connected to the Arduino's 5V and GND.
   - TX and RX are connected to the Arduino's RX and TX (cross-connected for communication).
5. Voltage Regulator: Ensures a stable 5V output from the 9V battery for the Arduino and other components.

This setup enables communication via Bluetooth, with rotary encoders providing input for directional control.

Building the APP

- We Optimized the Unity scene for mobile devices.
- We Exported the project as an APK for Android, ensuring compatibility with VR features and Bluetooth communication.

Outcome & Results

As the APP and the Hardware was successfully built, we have attached some pictures which shows the entire build process.

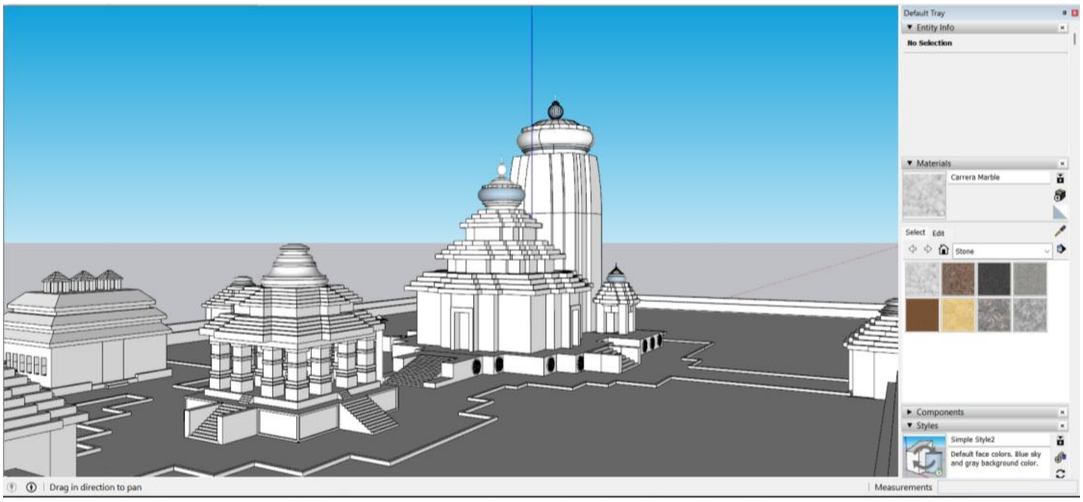

Figure 5 Model Build in SketchUp 2017

Figure 7 Model Rendered in Unity

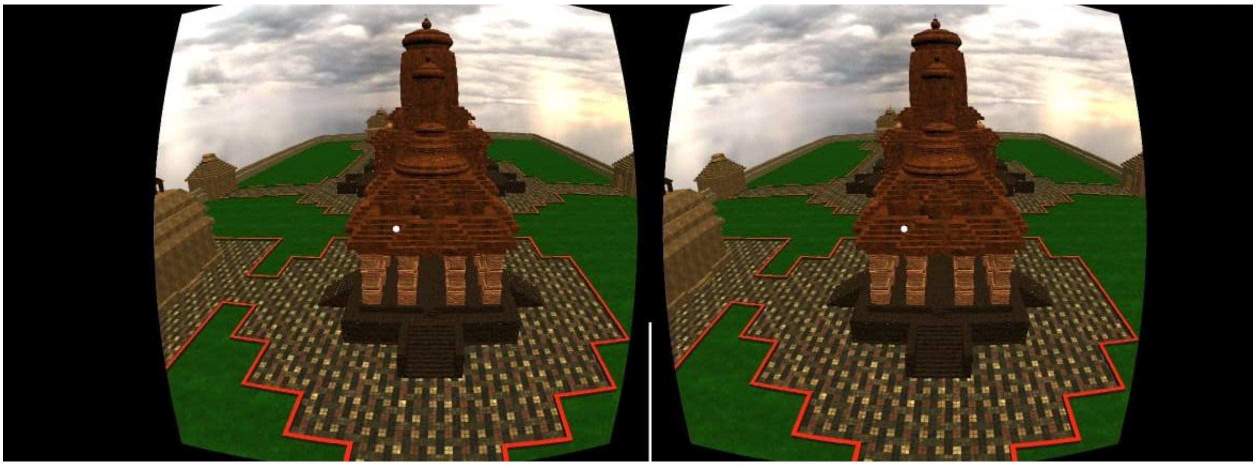

Figure 6 The Final Look of the app in VR Headset

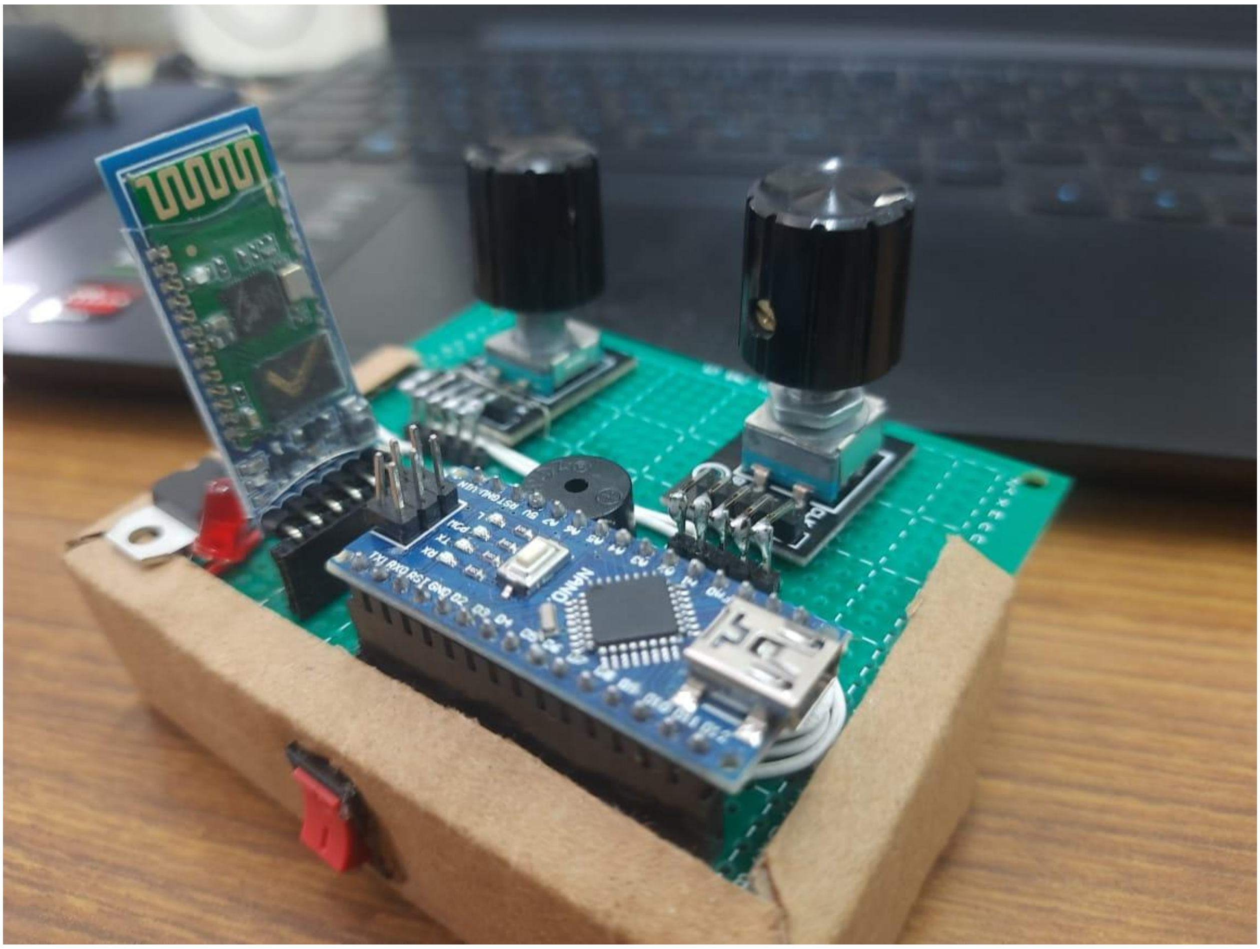

Figure 8 Hardware for navigating in the VR APP

We have added the hardware and software functional Videos to YouTube pleas feel free to check them out for better understanding of the project.

Software Video

https://youtu.be/iIKvO78teYw?si=5MBh0kfYr24ZFF8U

Hardware Video

https://youtube.com/shorts/w_pJv9OohF4?si=zEXy_RPL8XO6nMsD

We gave our friends and colleagues for reviewing the application and most of them had positive reviews to the project with major points listed out below.

- **The scenes and the navigation look good.**
- **Could have been more photorealistic.**
- **Better collision detection and handling.**
- **Better handling**

Conclusion and Future Work

This project successfully demonstrates the potential of Virtual Reality (VR) in preserving and presenting lost cultural heritage. By reconstructing the Konark Sun Temple, the team has created an immersive and interactive platform that merges historical accuracy with cutting-edge technology. The integration of hardware for navigation further enhances the user experience, making the exploration of the temple's intricate architecture intuitive and engaging.

The project not only achieves its primary goal of promoting cultural heritage but also serves as a scalable solution for other monuments worldwide. While the outcomes are promising, there is room for improvement in photorealism, collision handling, and interaction dynamics. The feedback received from users has laid a solid foundation for future iterations, which could include features like multi-user VR, animated storytelling, and AR integration.

- . **Enhanced Photorealism**: Explore tools like Unity HDRP (High-Definition Render Pipeline) to achieve superior graphics quality.
- **Educational Features**: Integrate **interactive guides** or **voiceover narratives** to explain architectural details.
- **Integration of AR**: Consider adding an AR module for mobile devices to view reconstructed elements overlaid on the real-world ruins.
- **Real-World Application**: Collaborate with tourism boards or cultural organizations to deploy this project in museums or as part of a virtual tourism app.
- **Analytics Integration**: Include data analytics to track user interaction within the VR environment to improve future iterations.
- **Use in Architecture industry**: The developed hardware can be used to navigate through VR scenes, the developed hardware along with any other rendered model can be used.

In conclusion, "Virtual Rebirth: Visualization of Lost Monuments Using VR" is a meaningful step toward preserving history through technology, offering an innovative blend of education, tourism, and cultural preservation. This work can inspire further advancements in the application of VR for historical restoration and heritage promotion.

Appendix

Microcontroller Code

```
// Rotary Encoder 1 Inputs
#define CLK1 4
#define DT1 3

// Rotary Encoder 2 Inputs
#define CLK2 7
#define DT2 6

// Buzzer Pin
#define BUZZER 13

// Variables for Encoder 1
int lastStateCLK1;

// Variables for Encoder 2
int lastStateCLK2;

// Variable to track Bluetooth connection state
bool bluetoothConnected = false;

void setup() {
    // Set encoder pins as inputs
    pinMode(CLK1, INPUT);
    pinMode(DT1, INPUT);

    pinMode(CLK2, INPUT);
    pinMode(DT2, INPUT);

    // Set buzzer pin as output
    pinMode(BUZZER, OUTPUT);

    // Setup Serial Monitor
    Serial.begin(9600);

    // Initialize last states
    lastStateCLK1 = digitalRead(CLK1);
    lastStateCLK2 = digitalRead(CLK2);

    // Power-on tone
    tone(BUZZER, 1000, 200); // Play tone at 1000 Hz for 200 ms
    delay(200);
    noTone(BUZZER);
}

void loop() {
    // Simulate Bluetooth connection status (replace with actual check)
    bool isCurrentlyConnected = isBluetoothConnected();

    // Play tone on Bluetooth connection
    if (isCurrentlyConnected && !bluetoothConnected) {
```

```
        tone(BUZZER, 2000, 200); // Play tone at 2000 Hz for 200 ms
        delay(200);
        noTone(BUZZER);
    }

    bluetoothConnected = isCurrentlyConnected;

    // Handle Encoder 2
    checkEncoder(CLK2, DT2, lastStateCLK2, 2);
    // Handle Encoder 1
    checkEncoder(CLK1, DT1, lastStateCLK1, 1);

    // Small delay for stability
    delay(1);
}

void checkEncoder(int clkPin, int dtPin, int &lastStateCLK, int encoderID) {
    // Read the current state of CLK
    int currentStateCLK = digitalRead(clkPin);

    // If last and current state of CLK are different, then pulse occurred
    if (currentStateCLK != lastStateCLK && currentStateCLK == 1) {
        // Check the direction
        if (digitalRead(dtPin) != currentStateCLK) {
            // Counterclockwise rotation
            Serial.print(encoderID);
            Serial.println("0");
        } else {
            // Clockwise rotation
            Serial.print(encoderID);
            Serial.println("1");
        }
    }

    // Update the last state
    lastStateCLK = currentStateCLK;
}

// Simulated Bluetooth connection check
bool isBluetoothConnected() {
    // Replace this with actual code to check Bluetooth connection
    // For now, we return true for testing purposes
    return digitalRead(8); // Example: Using pin 8 as a simulated Bluetooth status pin
}
```

ALL THE DOCUMENTS AND THE PROJECT FOLDER CAN BE FOUND HERE

https://drive.google.com/drive/folders/1K2IMAjl7MP34w0jAX769rz5ZgIITWrQs?usp=sharing