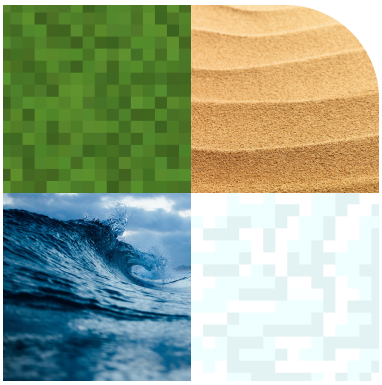
Mixed Reality Design Tool

User Guide



Preface

Team:

Brian Khieu
Chen Chen
Matthew Martin
Yuri Sun

In Collaboration with:

Professor Nelson Max
Professor Xiaoguang "Leo" Liu

Intended Audience:

This document is intended for future developers and application users.

Product Overview

This application is a landscape design tool that offers a mixed reality (MR) experience by integrating augmented reality (AR) and virtual reality (VR) technologies. The user is able to interact with a virtual landscape, rendered in Oculus Rift, using Oculus Touch controllers. The application receives a real-time video stream from a gimbal-mounted camera that moves together with the Oculus headset, generating an immersive MR experience. The hardware system utilizes a serial XBee wireless connection with an embedded system (Arduino-Uno), a gimbal-mounted camera and an onboard GPS, offering real-time video streaming and gimbal control. The software was developed and tested on a Windows PC in Unity3D development environment. This landscape design application allows users to create, edit and remove virtual objects in the real world and creates a seamless mixed reality scenario.

This system is intended for further integration with a drone and a GPS with better accuracy. All the hardware modules can be carried by a drone whose real-time physical location will determine the virtual location in the program.

Installation Process for Windows PC

Hardware Requirements:

1. An VR-ready desktop or laptop with an HDMI port, 3 USB-3 ports, and 2 USB-2 ports.

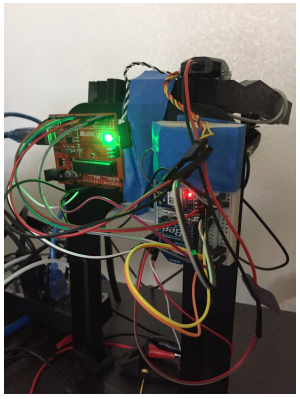
   Here is a good resource to determine whether a machine is VR-ready: http://www.roadtovr.com/how-to-tell-pc-virtual-reality-vr-oculus-rift-htc-vive-steam-vr-compatibility-tool/

   If a machine does not have enough USB ports, it is also possible to use a powered USB port.

2. Oculus Rift Headset

3. Oculus Touch Controllers

4. Customized Gimbal Set with 12V Power Supply and XBee Wireless Module Pair



5. Gimbal Mounted USB Webcam

Hardware Setup:

1. Oculus Rift and Touch Controller Setup

   Install Oculus software and follow the setup instruction via the following link: https://www3.oculus.com/en-us/setup/

   User will be instructed to connect two sensors and one headset taking up 3 USB-3 ports and one HDMI port.

2. Connect an Xbee module to PC with USB-2 connector.

3. Connect a gimbal-mounted camera to PC with USB-2 connector.

Software Setup:

Unity program can be executed from a standalone folder which is copied directly onto a VR-ready Windows PC.

For developer:

The source code for this application can be found via this link: https://github.com/cche80/ecs193_arvrdrone

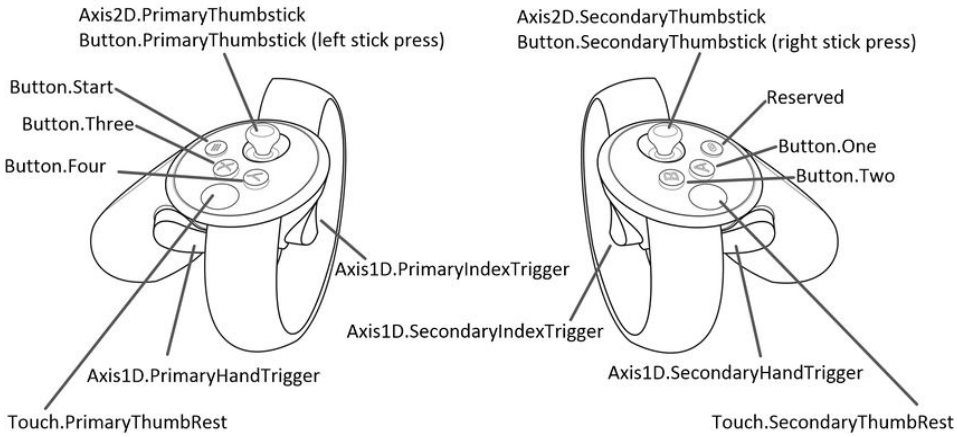
After the directory has been clone to a local repository, simply treat the folder “ARVR” as a Unity3D project. Download Unity development environment from https://unity3d.com/. The file hierarchy can be seen from the Unity environment.

Functionalities

Initialization and User Interface:

When the program starts, user is initialized in a 3D space with a randomly generated landscape. The background of the viewable area in the program will be provided by the gimbal-mounted camera, generating a background layer displaying the real world.

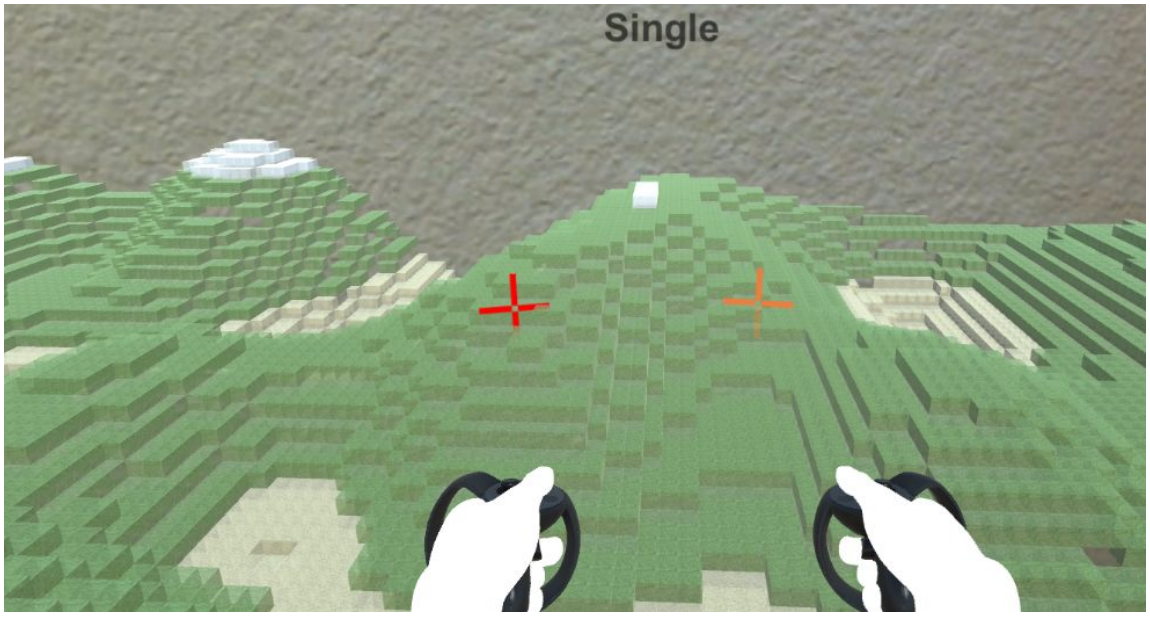
The camera will turn together with the Oculus headset if the gimbal is installed and communicates correctly with the program. The GPS will give the program the current coordinates of the camera setup.



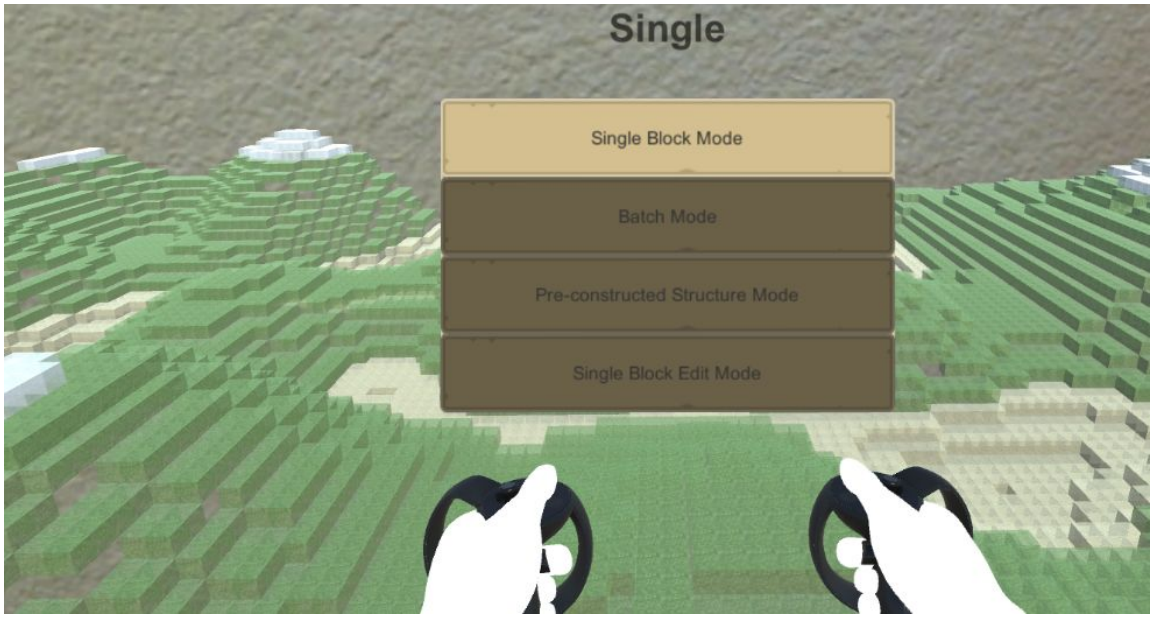
The virtual camera position can be controlled by using left (primary) and right (secondary) thumbsticks. Left thumbstick controls horizontal movements and right thumbstick controls vertical movement. Rotation is controlled by the Oculus headset.

Users can interact with the virtual world from a first-person perspective. The touch controllers can be held to point at a certain location of the virtual world to interact with the landscape. A pair of reticles would appear on the screen to inform the user which block is currently being referred to. Red reticle comes from the left (primary) controller and orange reticle from the right (secondary). The functionalities of left and right controller should be identical. However, due to a hardware flaw in the left controller, all the functionalities are demonstrated on the right controller.

This application has four modes: Single Block Mode, Batch Mode, Pre-constructed Structure Mode and Single Block Edit Mode. The current mode is display at the center top of the UI screen as “Single”, “Batch”, “Pre” and “Edit”. The default mode is “Single” when the program first starts.

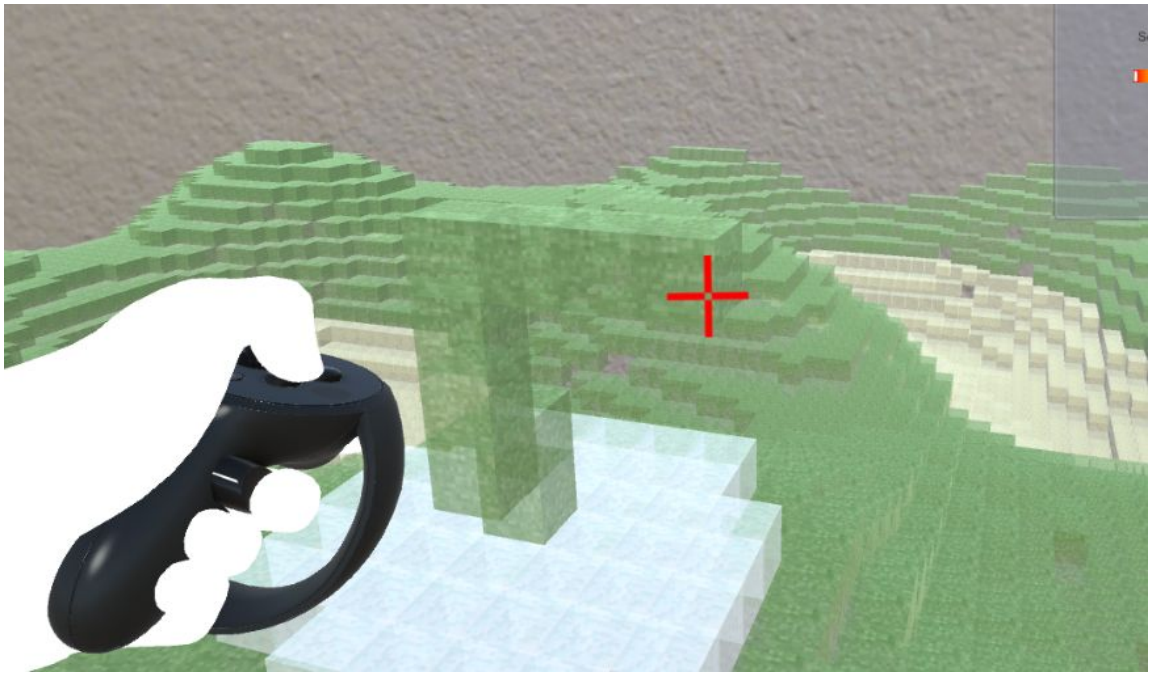


To change mode, press ‘X’ and ‘Y’ to cycle through the 4 modes sequentially. The Mode Menu can be toggled by pressing ‘B’.



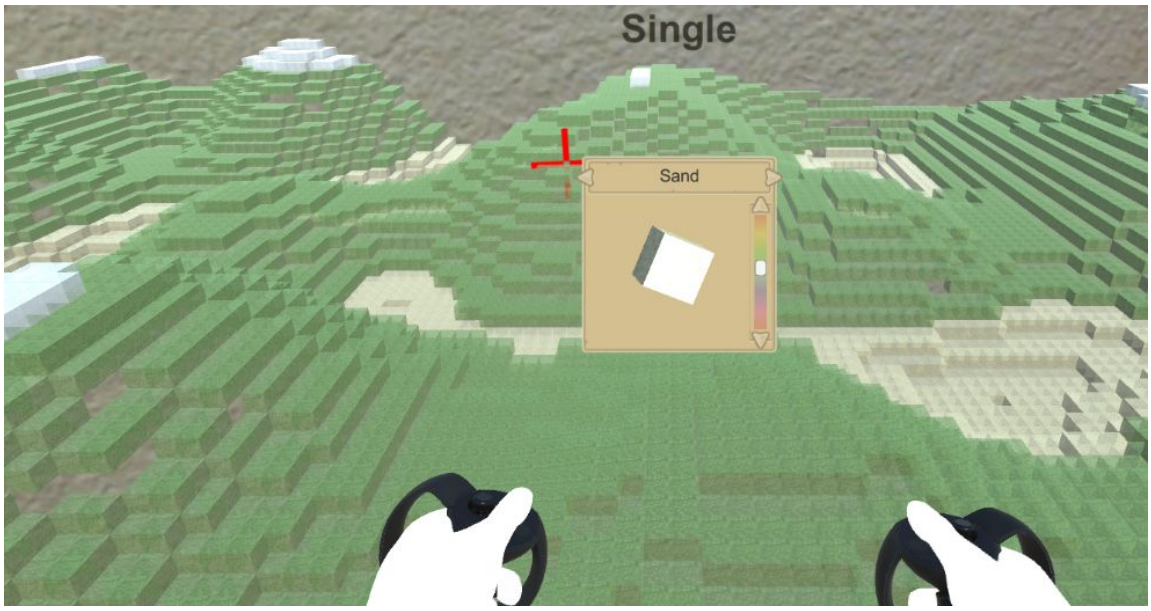
Single Block Mode:

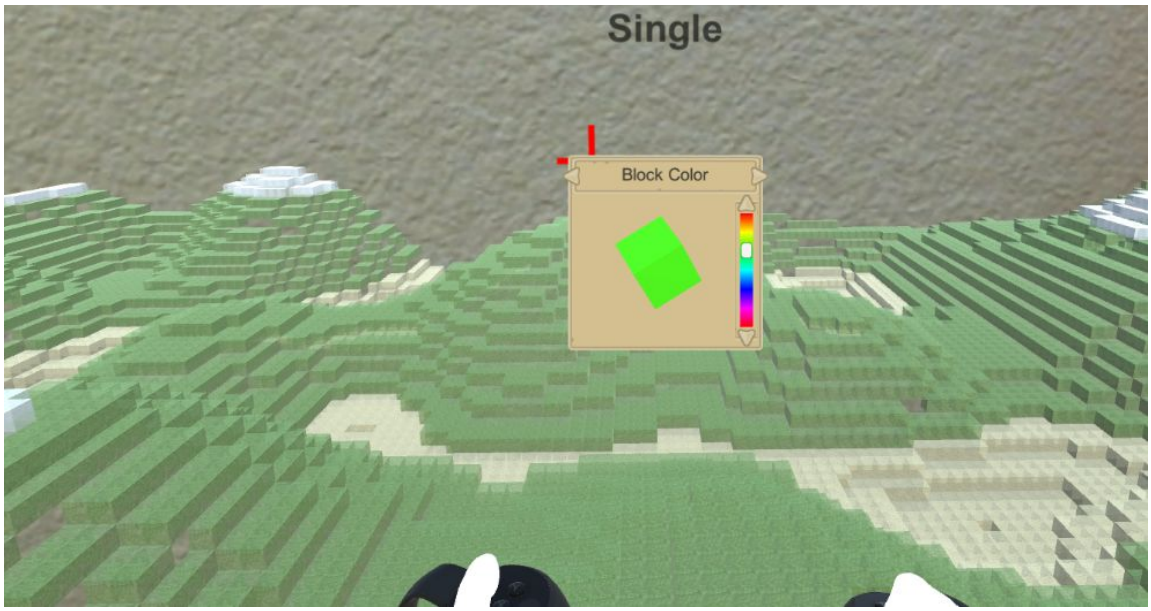
In Single Block Mode, Right (secondary) index trigger is responsible for creating a single block. Right (secondary) hand trigger is responsible for deleting a single block. When right thumb is not rested on the touch zone, each trigger will create/delete a single block.



When right thumb is rested on the touch zone, the creation and deletion of single blocks will be consecutive.

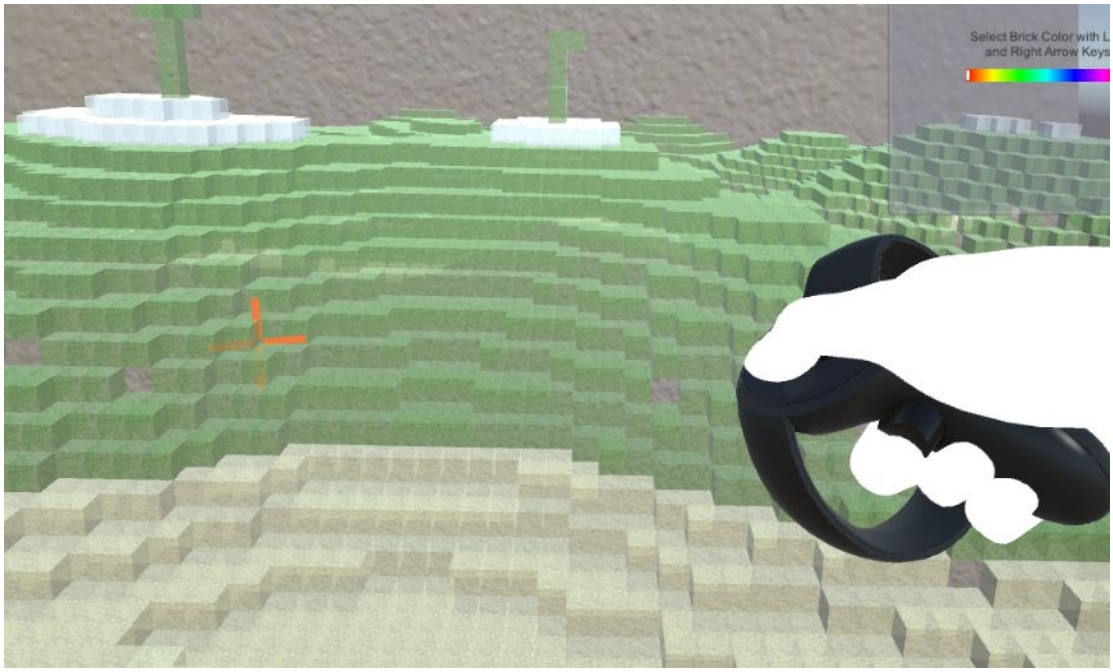
The texture and color of the block can be adjusted in the Color Selection Menu toggled by pressing button ‘A’. Use the ‘left’ and ‘right’ button of the right thumbstick to cycle through texture, and use ‘up’ and ‘down’ to select color.



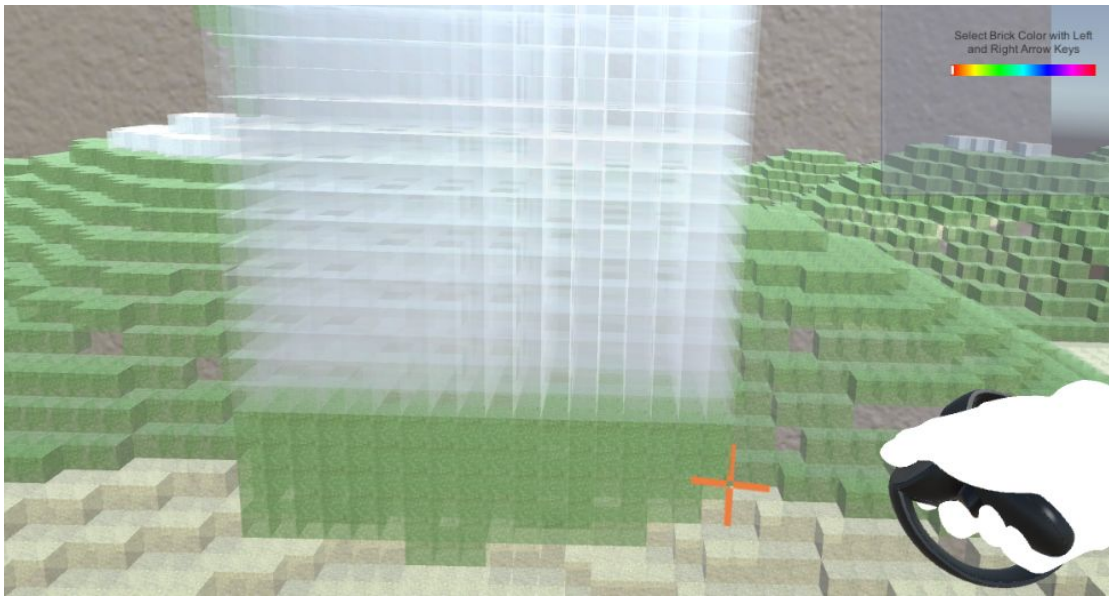


Batch Mode:

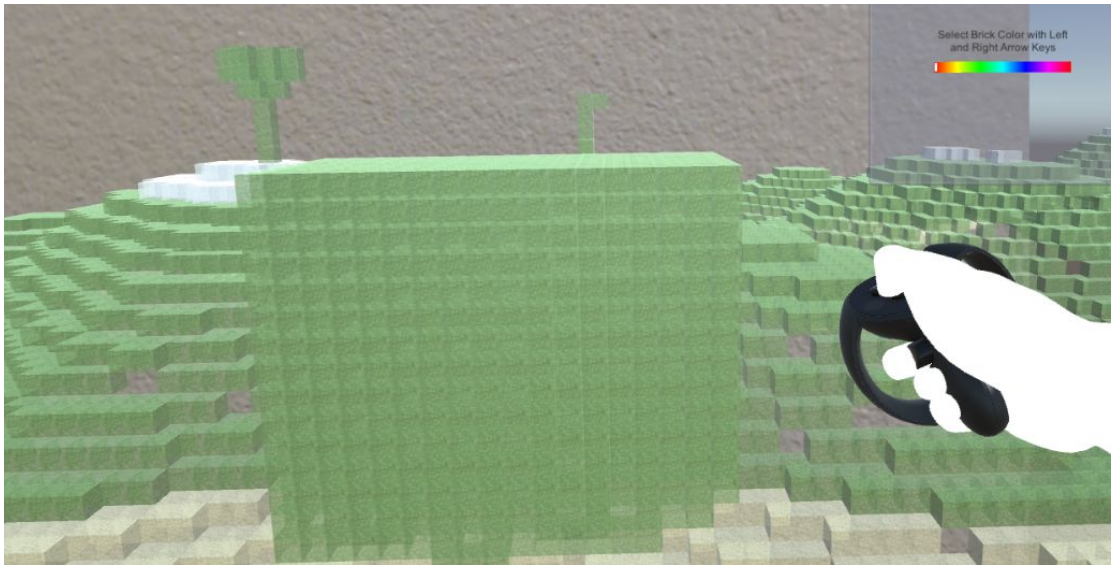
Another user-friendly way of creating a block structure is to use Batch Creation Mode, where a user could easily construct a cuboid structure. The cuboid creation is initiated by pressing the ‘Index Trigger’ on the right touch controller. The block when the right ‘Index Trigger’ is first pressed will serve as the first vertex of the base rectangle.



Now the user can drag across the virtual space, with the right ‘Index Trigger’ still pressed, and point towards the diagonal vertex of the base of the cuboid. When ‘Index Trigger’ is released, the block pointed by the reticle will serve as the diagonal vertex of the base. Now the program will enter Preview Mode where the height of the cuboid can be determined at the next move.



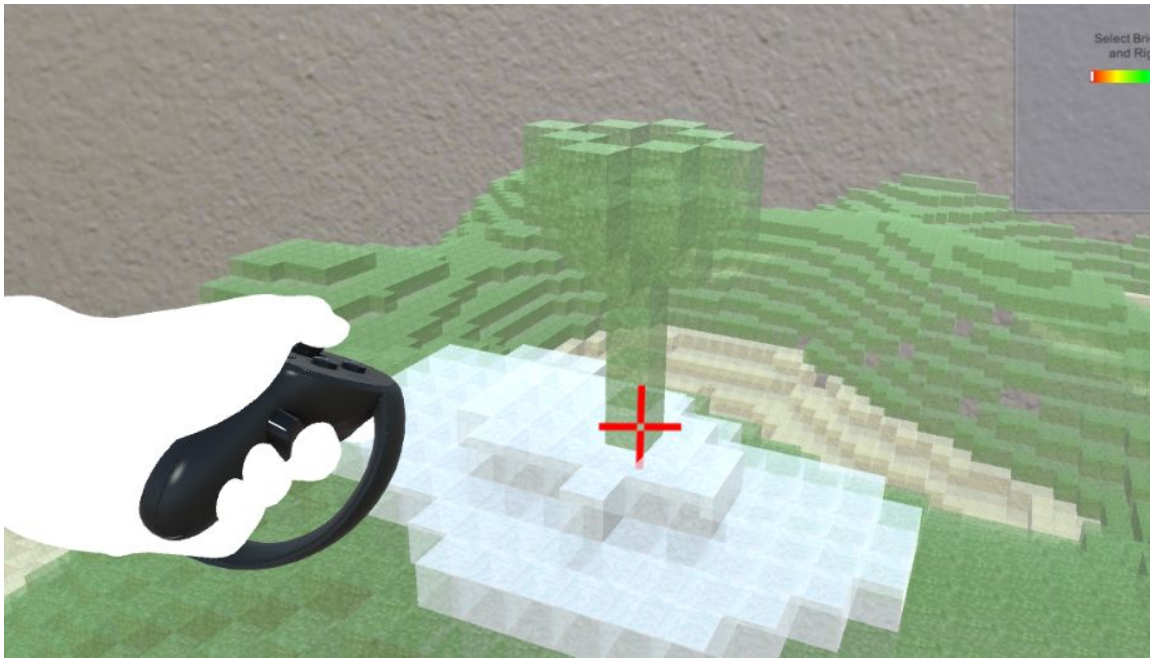
Now the user can point to a desire height of the structure and press the right ‘Index Trigger’ again in order to finalize the build.



The color selection process is the same as Single Mode.

Pre-constructed Structure Mode:

Again, the right ‘Index Trigger’ is responsible for creating a pre-constructed object which is imported from file. A pre-defined structure is placed in the virtual world as shown in the following figure.



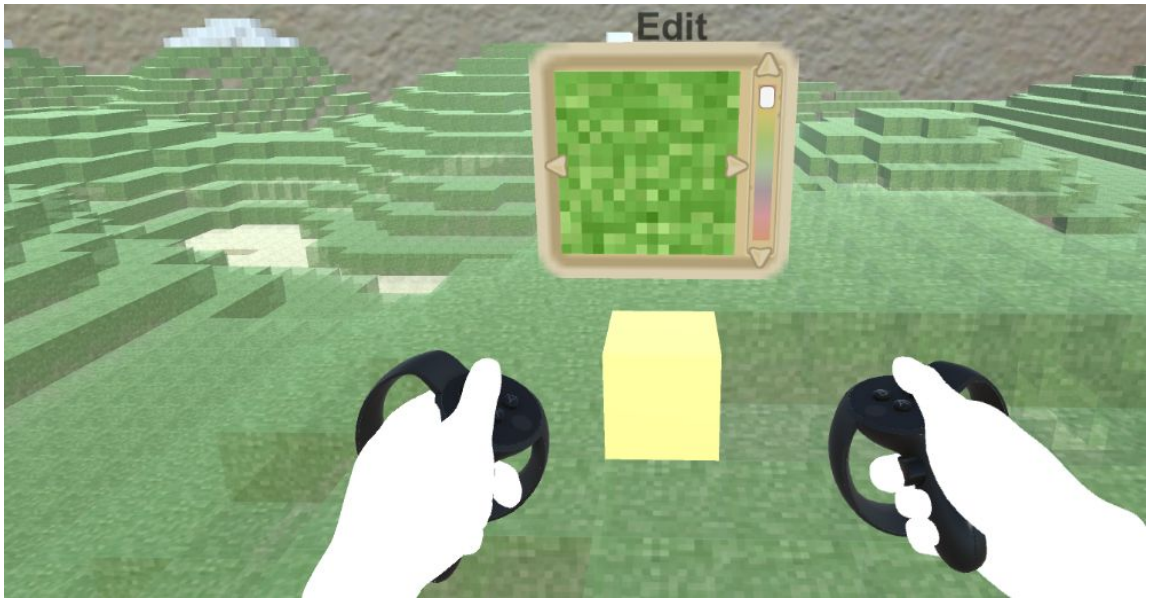
The following image shows a file that defines a pre-constructed tree. The first three lines define the dimension of the structure (5 x 8 x 5). The next 3 lines define the base point of the structure with regard to the space defined by the first three lines. The next 25 lines completely defines each block type in the 5 x 8 x 5 space. ‘0’ means no block and ‘g’ represents a grass block. Users can define their own structures with the same format.

tree.data

```
5
8
5
2
0
2
00000000
00000000
000000gg
00000000
00000000
00000000
000000gg
00000gg0
000000gg
00000000
000000gg
00000gg0
0ggggg00
00000gg0
000000gg
00000000
000000gg
00000gg0
000000gg
00000000
00000000
00000000
000000gg
00000000
00000000
```

Edit Mode:

In this mode, the user can change the color and texture of a single block. Use right ‘Index Trigger’ to select a block for edit. Change its color using the same controls as other modes. Press right “thumb button” to finalize the change, or press right “hand trigger” to close the edit window and discard the change.


Edit

Button Functionalities

Single Block Mode: create a single cube.

LeftThumbstick: move forward, backward, left and right
LeftStickPress: N/A
X: switch to previous mode
Y: switch to next mode
LeftIndexTrigger: N/A
LeftHandTrigger: N/A
LeftTouch: N/A

RightThumbstick: up: move down, down: move up
RightStickPress: N/A
B: call main menu
A: call preview panel
RightIndexTrigger: create one cube
RightHandTrigger: delete one cube
RightTouch: hold to create/delete cubes continuously

Batch Mode: create cubes, decide the width, length and height, in cube unit.

LeftThumbstick: move forward, backward, left and right
LeftStickPress: N/A
X: switch to previous mode
Y: switch to next mode
LeftIndexTrigger: N/A
LeftHandTrigger: N/A
LeftTouch: N/A

RightThumbstick: up: move down, down: move up
RightStickPress: N/A
B: call main menu
A: call preview panel
RightIndexTrigger: First press and drag: create a base with length and width;
Second press on the created blocks: to determine the height
RightHandTrigger: delete one cube
RightTouch: N/A

Pre-constructed Structure Mode: create pre-designed models.

LeftThumbstick: move forward, backward, left and right
LeftStickPress: N/A
X: switch to previous mode
Y: switch to next mode
LeftIndexTrigger: N/A
LeftHandTrigger: N/A
LeftTouch: N/A

RightThumbstick: up: move down, down: move up
RightStickPress: N/A
B: call main menu
A: call preview panel
RightIndexTrigger: Create a designed model - Tree.
RightHandTrigger: delete one cube
RightTouch: hold to create/delete cubes continuously

Single Block Edit Mode: edit each cube.

LeftThumbstick: move forward, backward, left and right; when the edit panel is on, function deactivated.
LeftStickPress: N/A
X: switch to previous mode; when the edit panel is on, function deactivated.
Y: switch to next modewhen the edit panel is on, function deactivated.
LeftIndexTrigger: N/A
LeftHandTrigger: N/A
LeftTouch: N/A

RightThumbstick: up: move down, down: move up; when the edit panel is on, switch the texture(left and right), switch the color(up and down)
RightStickPress: confirm to apply the current style
B: call main menu; when the edit panel is on, function deactivated.
A: call preview panel; when the edit panel is on, function deactivated.
RightIndexTrigger: Call the edit panel
RightHandTrigger: Close and cancel the edit panel
RightTouch: N/A

Troubleshooting

1. If the program freezes at startup, please check serial port connection with Xbee. The program will not start properly if the serial connection is not established.

2. If the touch controllers fail to control the camera perspective, please check the hardware connection of Oculus Touch controllers. Please make sure both controllers are clicked before they can be recognized by the program.

3. GPS may not function reliably indoors. If realtime GPS coordinate is needed, please use the application outdoors.

4. The gimbal might stop retaining its intended position and stop moving altogether. This is because the load of the gimbal has been exceeded by the movement of the camera. If this happens, switch off the 12V power supply for all embedded systems and turn on again after the indicator lights are off. The gimbal will reset itself and start working again. This problem could be solved by using a better gimbal with a higher load capacity.

Frequently Asked Questions

1. How can the application solution be used in real life?

   This system is intended for use in any application where object-planning in relation to the real world is needed. Applications include event planning and civil design.This application solution is intended for further integration with drones and a more accurate GPS system. The entire embedded system could be carried by a drone, and the camera stream needs to be wirelessly transmitted to a PC running the software.

2. The landscape is randomly generated. What if I want to start fresh and design my own landscape from scratch.

   This is not implemented in the current version of the application. This could be a feature in future releases.

3. Can I save/load my current design?

   This is not implemented in the current version of the application. However we have implemented pre-constructed structures to allow for easier building.

4. Why does the program resemble Minecraft?

   As a broadly recognized platform, modeling the user experience around Minecraft allows us to deliver a program that is easy for many users to understand, without the need for explanation.

5. Why is the computer connected with the gimbal with a wired connection instead of a wireless connection?

   This is for us to debug and troubleshoot faster. The arduino needs wired connection for flashing and reprogramming.

Technical Difficulties

1. Our team initially wanted to adopt an RTK Kinematic GPS for accurate location control. It turned out that the base station connection was intermittent and we decided against GPS control.

2. The wireless streaming of video to our system turned out to be difficult to implement. A balance needs to be achieved between power consumption and camera performance, which was not possible with our resources.

3. There is a delay between the update of the virtual object and the streaming of the real world video feed. The update of the virtual objects happens before the reality background does. This is a major issue in many mixed reality projects, which can potentially be solved by computer-vision-based techniques.

Contact Information

Brian Khieu:
btkhieu@ucdavis.edu

Chen Chen:
ccccche@ucdavis.edu

Matthew Martin:
mdmartin@ucdavis.edu

Yiru Sun:
yrsun@ucdavis.edu

Glossary of Terms

Arduino:

an open-source electronics platform or board and the software used to program it. Arduino is designed to make electronics more accessible to artists, designers, hobbyists and anyone interested in creating interactive objects or environments.

Augmented Reality (AR):

A live, direct or indirect view of a physical, real-world environment whose elements are augmented (or supplemented) by computer-generated sensory input such as sound, video, graphics or GPS data.

Differential Global Positioning System (DGPS):

An enhancement to Global Positioning System that provides improved location accuracy, from the 15-meter nominal GPS accuracy to about 10 cm in case of the best implementations.

Embedded System:

a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts.Embedded systems control many devices in common use today.

Gimbal:

A pivoted support that allows the rotation of an object about a single axis.

Inertial Measurement Unit (IMU):

An electronic device that measures and reports a body's specific force, angular rate, and sometimes the magnetic field surrounding the body, using a combination of accelerometers and gyroscopes, sometimes also magnetometers.

Mixed reality (MR):

The merging of real and virtual worlds to produce new environments and visualisations where physical and digital objects co-exist and interact in real time.

Oculus Rift:

A virtual reality headset developed and manufactured by Oculus VR.

Oculus Touch:

A virtual reality controller developed and manufactured by Oculus VR. It is used together with a Oculus Rift headset to allow users to use a virtual hand to interact with the program.

Unity3D:

A cross-platform game engine developed by Unity Technologies and used to develop video games for PC, consoles, mobile devices and websites.

Virtual reality (VR):

Immersive multimedia or computer-simulated reality, replicates an environment that simulates a physical presence in places in the real world or an imagined world, allowing the user to interact in that world.

XBee:

A brand of radio communication modules (made by Digi) that can support a number of protocols, including ZigBee, 802.15.4, WiFi, etc. Its range is 10 to 30 meters.

Appendix: Required Document Version 3.0

Mixed Reality Landscape Design Application

Team MR

Team Members: Brian Tuan Khieu, Chen Chen, Matthew David Martin, Yiru Sun

Intro

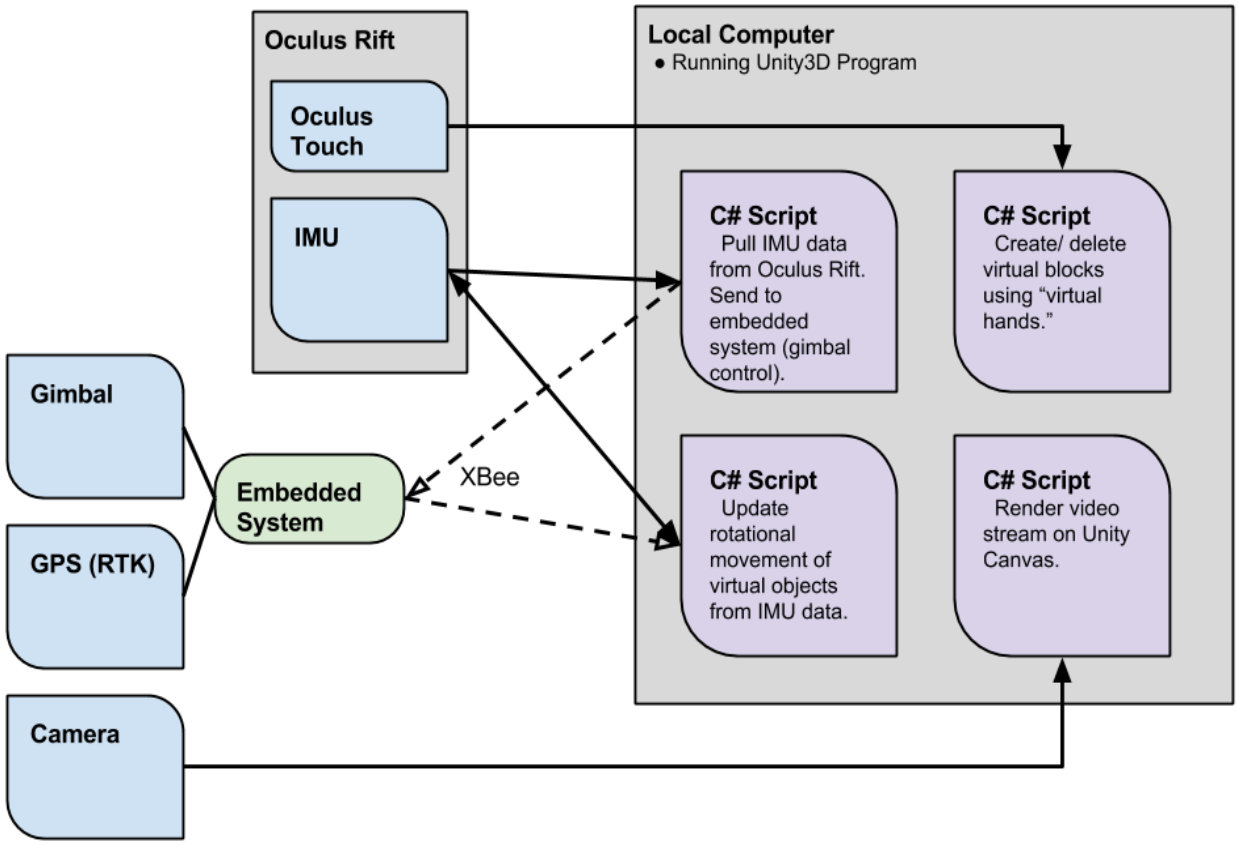
Our team's goal is to create mixed reality experience by integrating AR and VR technology. By meshing the virtual world accurately with a feed of the real world, users will be able to create, make changes, and interact with virtual objects, the locations of which could be precisely realized in the real world. The application of our program covers gaming, architecture, event planning, architecture, military applications, etc. We transpose the video stream from a web camera of the real world accurately onto the virtual space and have both worlds tightly synchronized.

Technology Employed

Unity3D is our main platform for integrating different sensory inputs and creating a seamless mixed reality scenario. The program consists of two major parts. The first part focuses on delivering a real world experience from the perspective of a gimbal-mounted web camera to the inside of the Oculus Rift virtual reality headset. We used a USB webcam that are suitable for video transmission. The video stream from the camera will be transmitted back to Unity3D through USB and rendered on the VR headset screen. The second part of the program takes the headset angle as inputs and reflect those changes accurately in the virtual space. The hardware and sensory technologies that are crucial to our project are gimbal, IMU, GPS, embedded systems, Unity3D, virtual reality headset, touch controllers and their respective APIs. The user interface of the application allows users to create, edit and delete virtual objects.

This application solution is meant for further integration with drones and a more accurate GPS system. The entire embedded system will be carried by a drone and the camera stream needs to be wirelessly transmitted to a PC running the software.

System Architecture Overview: High Level



Use Cases / User Stories

- As a user, I will see a main menu when donning the oculus rift. I will be able to start the program to start streaming from the web camera.
- As a user, I will be able to see the reflected movement of the headset by use of the camera on the gimbal through the displayed feed of the oculus rift.
    - Acceptance test: Gimbal mimics headset position and movement and Oculus streams the video taken from the camera.
- As a user, I will be able to select an object by use of the oculus rift and use in the the rendered virtual space.
- As a user, I will be able to create/edit/remove virtual objects so that the virtual object will reflect more complicated structures and designs.
- As a user, I will be able to specify an area as the base of a certain structure and than the height so that a 3D structure can be constructed in an intuitive and fast manner.
- As a user, I will be able to see my current GPS coordinate.
- As a user, I will be able to drop pre-constructed structures in the virtual space.