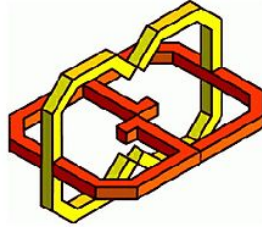
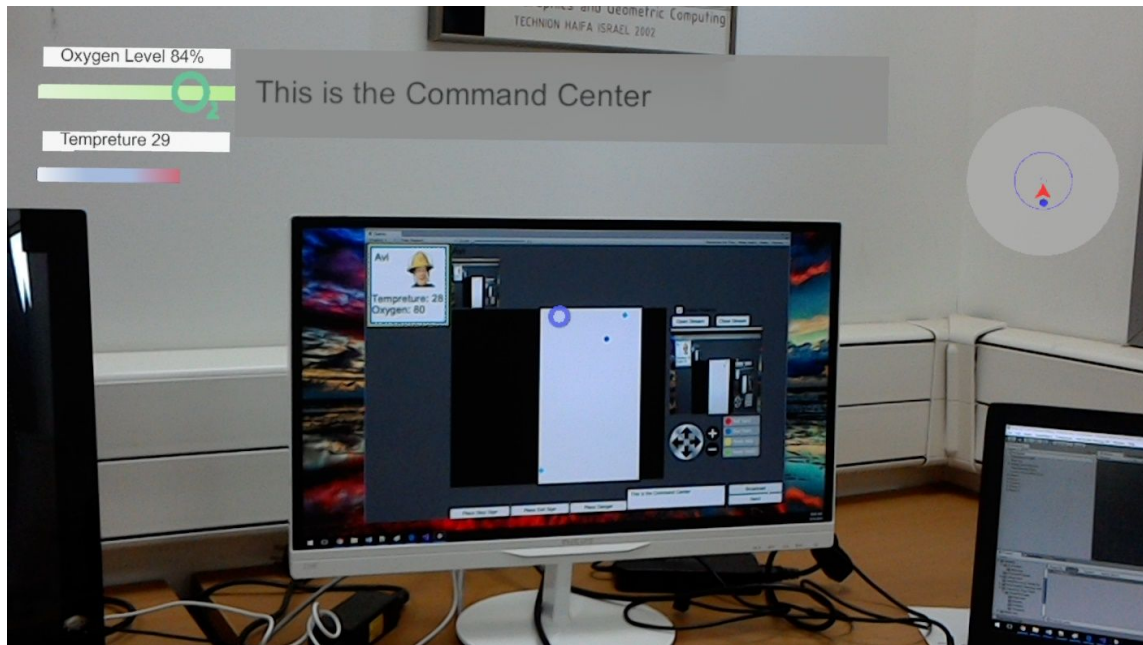


Geometric Image Processing Laboratory



Center for Graphics and Geometric Computing



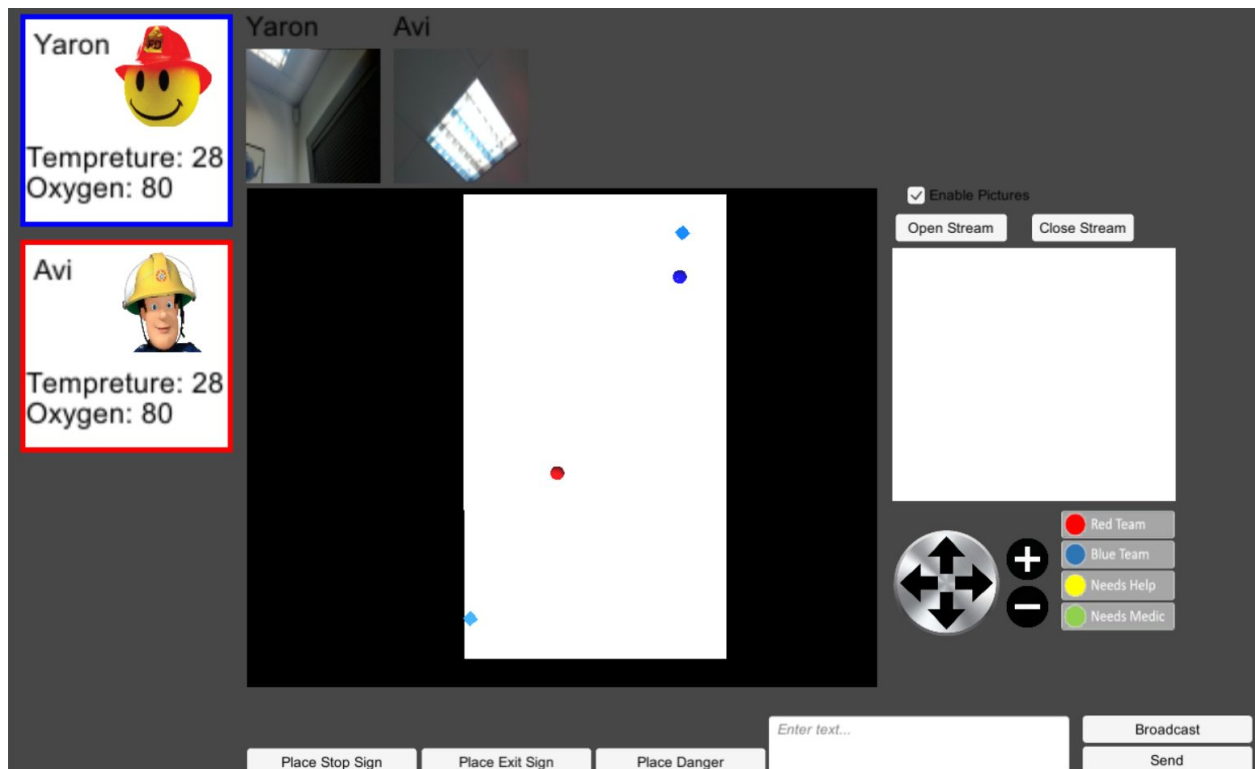
Augmented Emergency Response

Avi Kotlicky, Arik Reinberg

[GIP] | [234329] | [10.09.17]

Project idea

We developed a 3D Augmented Reality Multi-User emergency response application. Our application allows all users to simultaneously see the same scene, while a command center on a PC sees in real time users and their movements displayed on a real time map. The command center can communicate with all the users in scene, display hologram warnings in areas and manage all the users. The users may send warnings and request help from the command center and other users. The users also have a user location system that allows them to keep track of their team.



Introduction

We developed HoloLens application, in Unity 2017 - graphic engine, scripted with C# in Visual Studio 2017 and MonoDevelop environment.

Unity is a cross-platform game engine that is used to develop video games for PC, consoles, mobile devices and websites.

Microsoft HoloLens is a pair of mixed reality smart-glasses.



Hardware:

The HoloLens features an inertial measurement unit (IMU) (which includes an accelerometer, gyroscope, and a magnetometer) four "environment understanding" sensors (two on each side), an energy-efficient depth-camera with a $120^{\circ} \times 120^{\circ}$ angle of view, a 2.4-megapixel photographic video camera, a four-microphone array, and an ambient light sensor.

The HoloLens contains an internal rechargeable battery, with average life rated at 2–3 hours of active use, or 2 weeks of standby time. The HoloLens can be operated while charging.

The headset uses Bluetooth LE to pair with the included Clicker, a thumb-sized finger-operating input device that can be used for interface scrolling and selecting.



Augmented Reality Setup

Minimum Requirements for computers

- 64-bit Windows 10 (Not home Edition)-(for SDK)
- 64-bit CPU - (for SDK)
- 8 GB of Ram- (for SDK)
- Visual Studio 2017 or later -(for UWP)
- Bios Features required and enabled: - (for SDK)
 - Hardware-assisted virtualization
 - Second Level Address Translation (SLAT)
 - Hardware-based Data Execution Prevention (DEP)
- GPU**
 - DirectX 11.0 or later - (for SDK)
 - WDDM 1.2 Driver or later - (for SDK)
 - Feature level 9.3 capabilities - (for Unity)

***The software could possibly work on lower quality GPU's but the speed and performance will be impacted.*

More Details can be found here:

https://developer.microsoft.com/en-us/windows/mixed-reality/install_the_tools#system_requirements

Setup -> Build -> Deploy

- Download the required software
 - Visual studio 2017
 - Hololens Emulator
 - Unity 2017.2

Links to download locations can be found here:

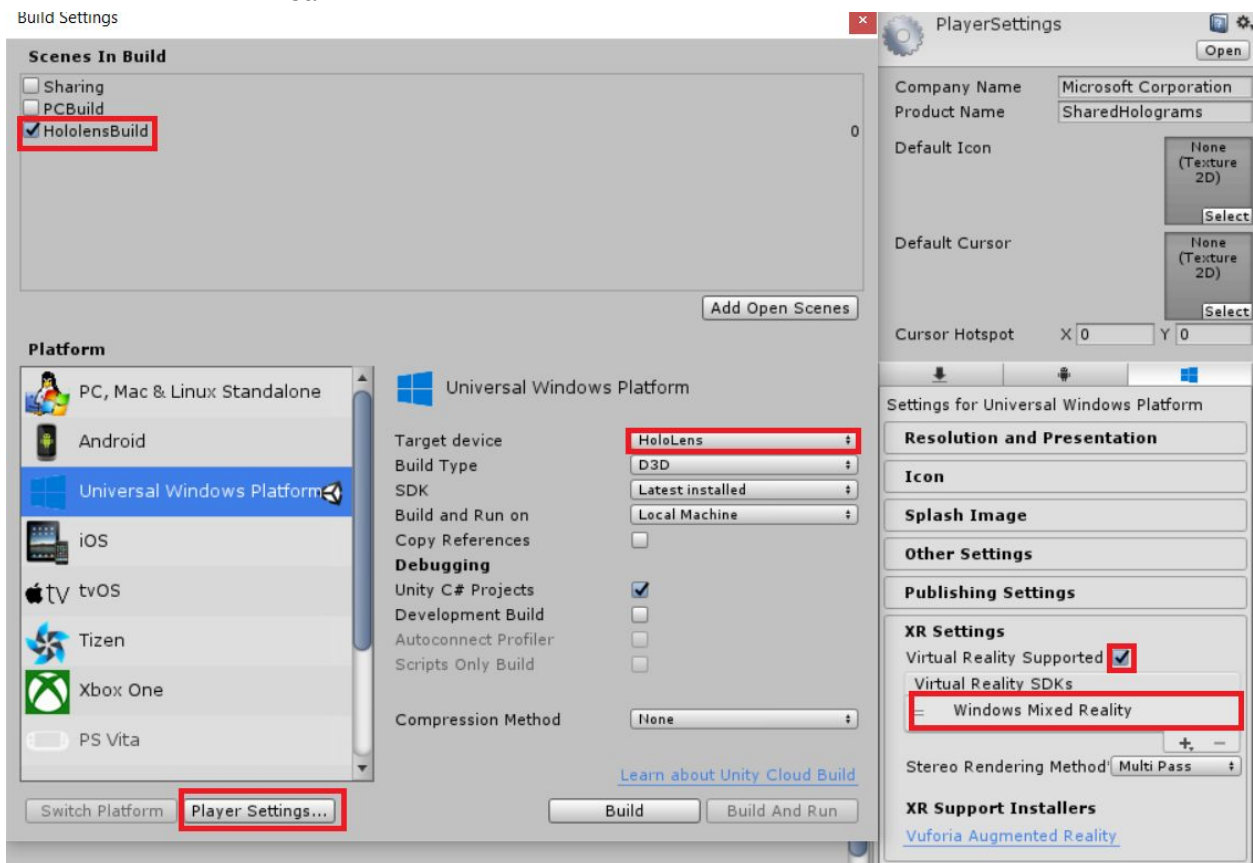
https://developer.microsoft.com/en-us/windows/mixed-reality/install_the_tools

- **Download the "AER" Unity Project from our website:**
 - <https://TechnionCGGCProject.wixsite.com/website>

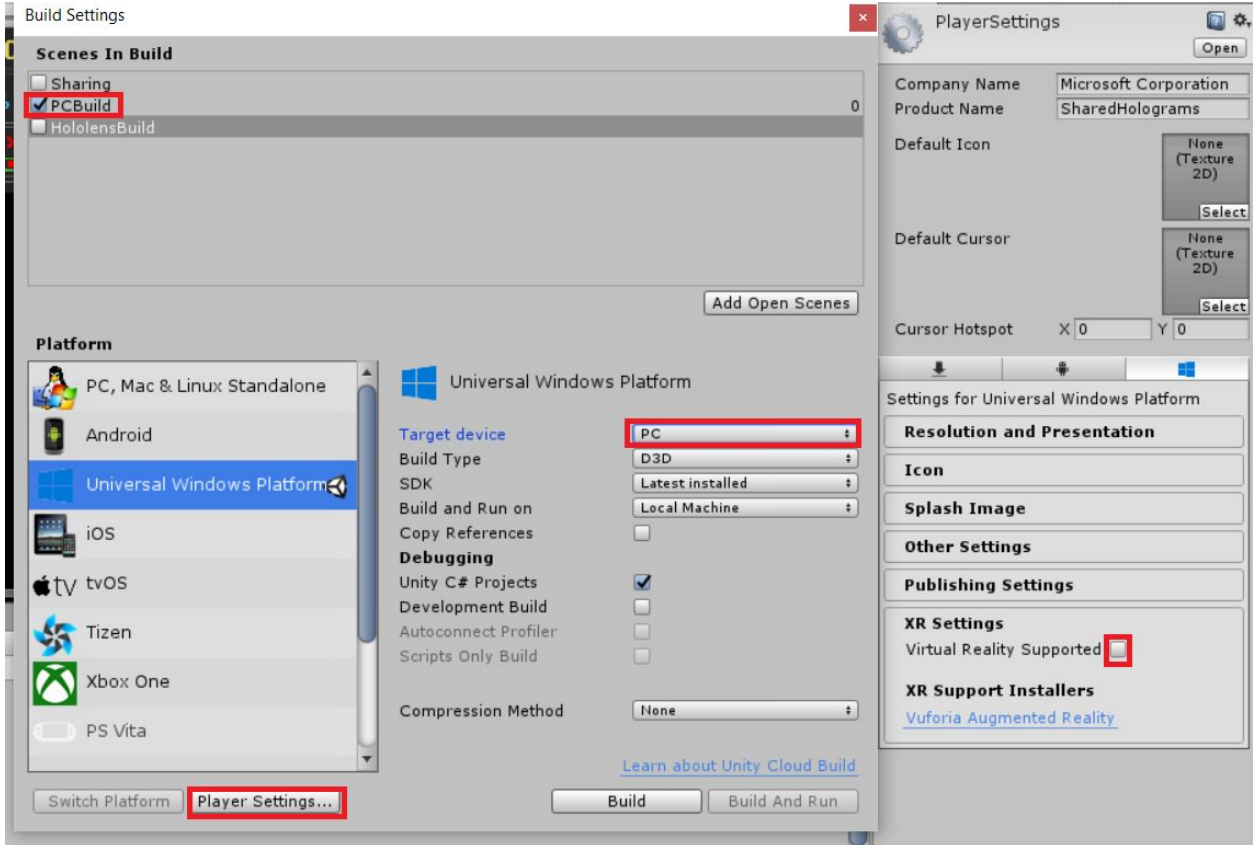
- Open the AER project in Unity 2017
- In the Hierarchy go to Managers->Sharing
In the inspector under “Server Address” put the IP address of the computer that will be running the sharing service.

***This can be found by running the command ipconfig from command prompt of the desired computer.*

- Go to File->Build Settings and select the Windows store under Platform
***Install the windows store plugin for Unity if it isn't already installed*
- Build the project for Windows with the settings listed to the right.
- Notice that you make a build with the “HololensBuild” scene and “PCBuild” scene
 - On hololens build make sure:
 - the HololensBuild scene is set
 - the target device is Hololens
 - Under player settings the VR is supported and the WMR is loaded in the sdk



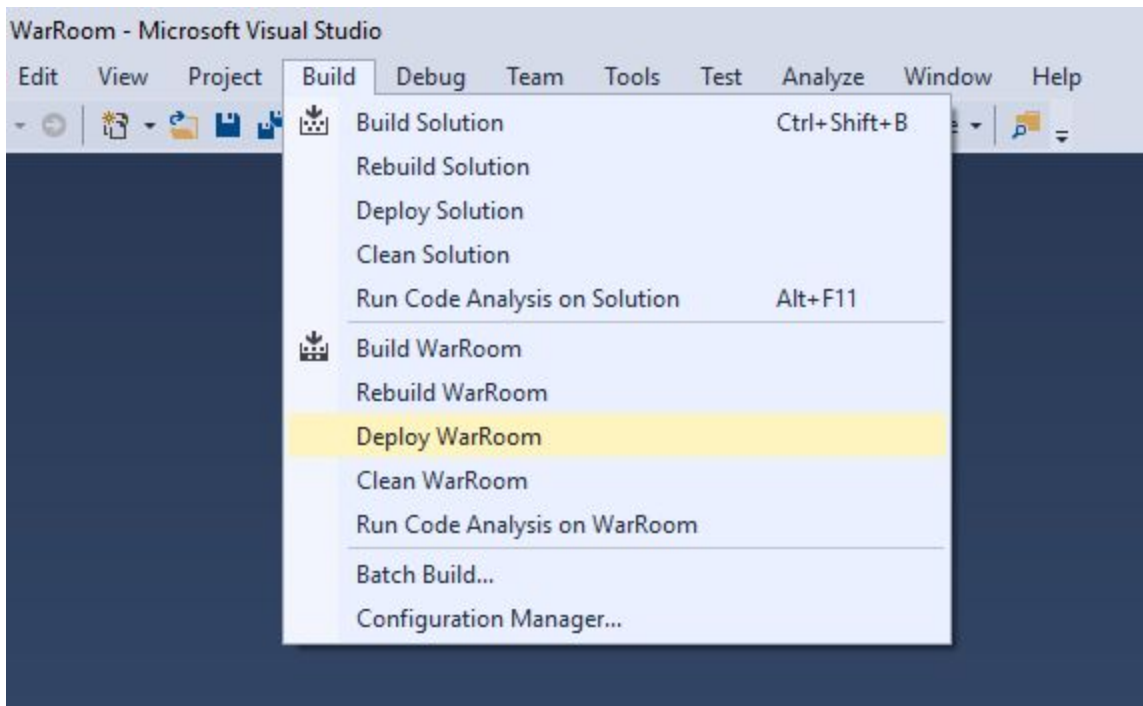
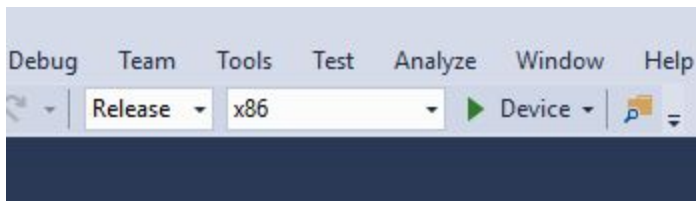
- On PC build make sure:
 - the PCBuild scene is set
 - the target device is PC
 - Under player settings the VR is not supported



- Connect the hololens and open the Visual Studio Solution
https://developer.microsoft.com/en-us/windows/mixed-reality/using_visual_studio

In the hololens:

- Setting->Update->For Developers->Developer Mode should be set to ON
 - Pair device - The Hololens will display a pin for Visual Studio
 - The Device portal for your hololens can be accessed in Settings->Update and Security->For Developers->Device Portal
 - <https://docs.microsoft.com/en-us/windows/uwp/debug-test-perf/device-portal-hololens>
- Select the “Release x86 Device” then press deploy



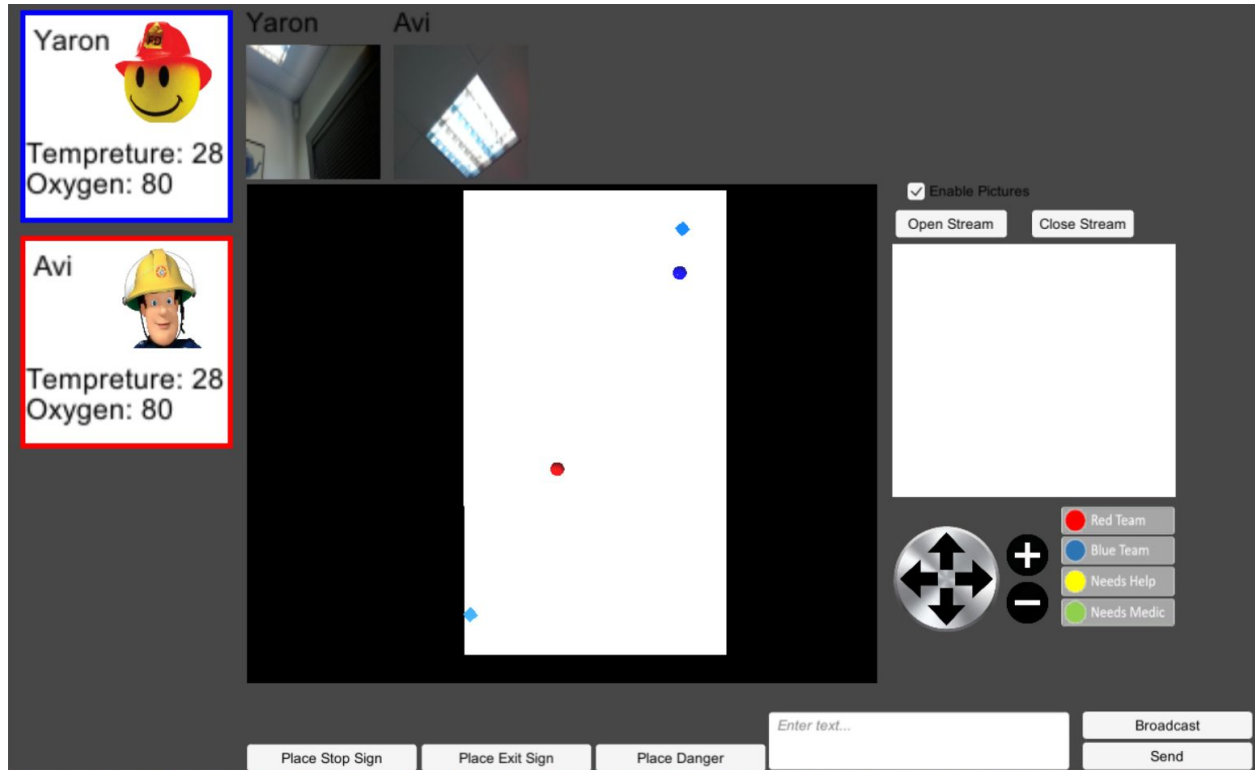
Hololens User Guide

Key Gestures



- Long tap to open and close menu
- Menu options
 - ◆ Help - Alerts everyone you need help
 - ◆ Medic - Alerts everyone you need medical evac.
 - ◆ Camera - Sends screenshot to Command center.
 - ◆ Close - Closes the menu
- When entering the scene you have an option to choose your avatar you will be seen as.
- The top left of your view are your vitals
- The top right of your view is your private radar where you will see other users.
- Messages will be shown on the top of your view from the command center.
- Voice Commands
 - ◆ Open Menu - opens the menu
 - ◆ Close Menu - closes the menu
 - ◆ Help - Alerts everyone you need help
 - ◆ Medic - Alerts everyone you need medical evac.

PC Command Center User Guide



- Sending a message
 - ◆ If you want to send a message to a specific user select their icon on the left side, enter text into the text box and click send on the bottom right.
 - ◆ When broadcasting a message you only type the message into the textbox and click broadcast.
- Sign placing for users
 - ◆ When placing a sign click the desired sign button and click a location into the scene for placement
- Streaming video from users
 - ◆ First make sure the Enables Pictures is toggled
 - To stream from a specific user select the user from the users on the left and click open stream
 - To close the stream click close stream
 - ◆ To disable streaming untoggle Enable Pictures

- The user can have their team color switched by right clicking on the avatar and changing their team.
- The map legend is as follows
 - ◆ A blue or red user moving on the map displays their team color
 - ◆ A yellow user needs help
 - ◆ A green user needs medical assistance

Application Server Multi User Guide

- ❖ Run the SharingService.exe from the desired computer
- ❖ Launch the PC build
- ❖ Launch the Sharing on all the hololenses
- ❖ Choose an avatar

Features and Implementations

- ❖ SendSMS.cs
 - Multiple functions to send complex string messages to be parsed and decoded userside
- ❖ SoftFollowObjectScript.cs
 - Allows the menu to soft follow the user
- ❖ RecieveImage.cs
 - OnImageRecieved
 - Parses the message bit array from the network and converts it into an image to display
- ❖ SpawnManager.cs
 - OnSmsRecieved
 - Checks and parses the SMS for complex commands (like spawning of objects)
- ❖ PlaceCube.cs

- Spawns the signs in the scene
- ❖ PhotoCaptureScript.cs
 - TakePicture
 - Takes a picture Asynchronously
 - OnCapturedPhotoToMemory
 - Converts the picture to a bit array for sending to the server
 - OnStoppedPhotoMode
 - Closes the camera to allow it to take another picture
- ❖ LookUpScript.cs
 - This script helps manage the radar for individual users
- ❖ MapHandlerScript.cs
 - We control all the map buttons from this
- ❖ HololensMenuGestureManager.cs
 - We add the long hold gesture here and open/close the menu accordingly
- ❖ GUIClickController.cs
 - OnPointerClick(PointerEventData eventData)
 - Allows for custom click control, left right or middle on UI
- ❖ CustomMessages.cs
 - SendSms(XString text)
 - Allows for text to be sent to other users through the server
 - SendImage(byte[] data, uint length)
 - Allows for images converted to byte array to be sent to other users through the server
 - SendSpawnObject
 - Spawns an object to every user in the same location