



TECH 1711 - Mixed Reality Studio

Augmented Reality



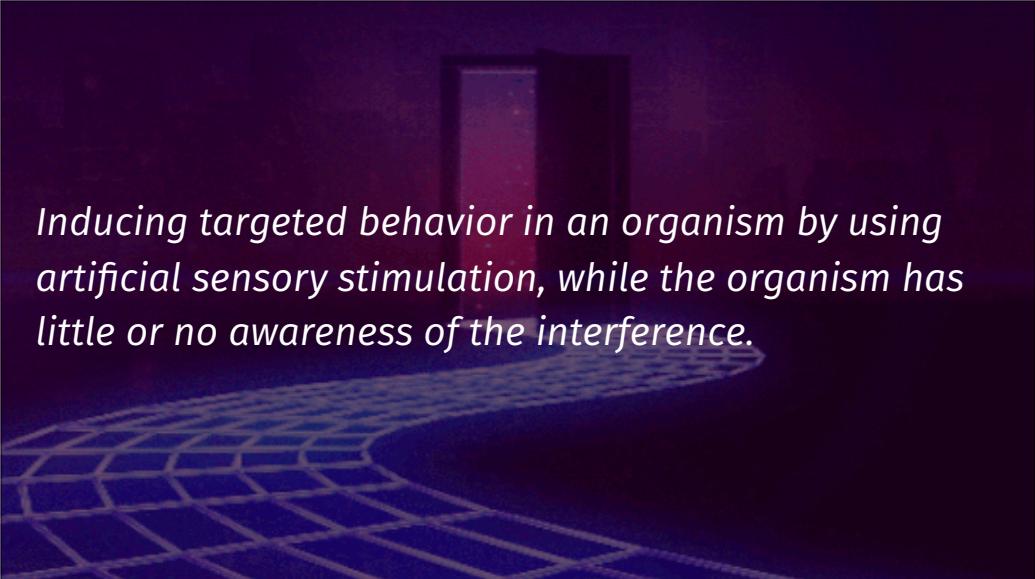
With VR we could create everything!

In AR, we have to (get to?) work as a layer on top of the real world (this is actually *harder!*).

What is AR/VR/MR?

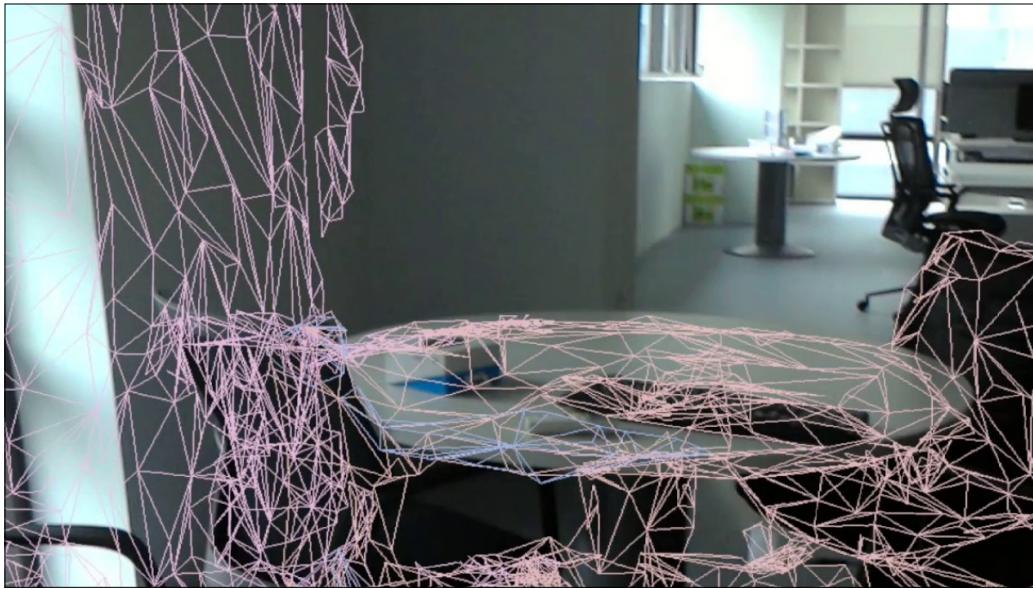


Let's take a look at our first definition again...



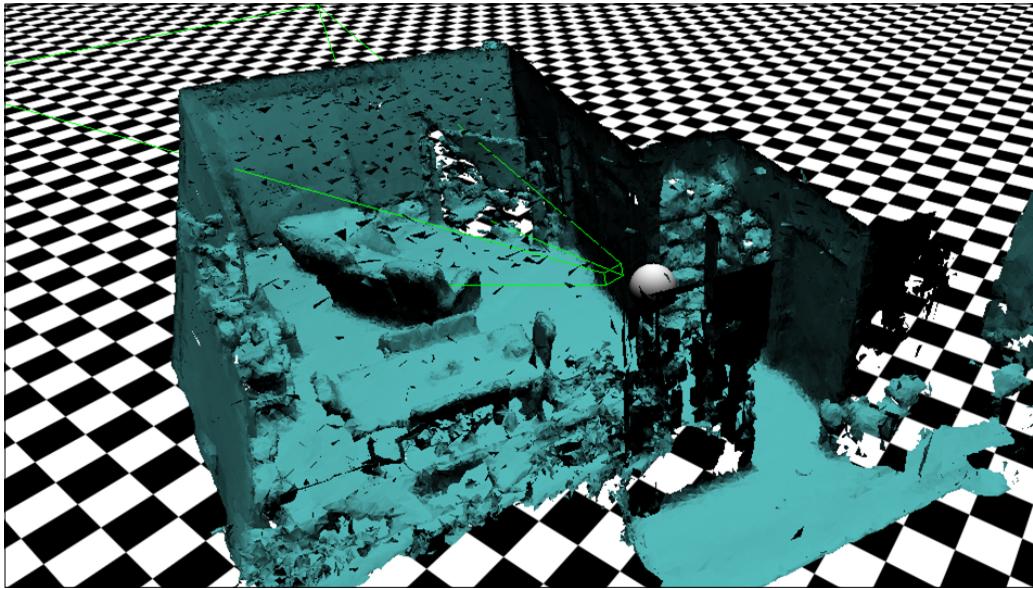
Inducing targeted behavior in an organism by using artificial sensory stimulation, while the organism has little or no awareness of the interference.

This may need to be revised a bit. AR is generally more actively engaged in that “interference”



Awareness of the world: 3d Sensors (yay occlusion!)

Similar to what a Kinect does - gets a 3D model of what it can see - but the hololens continues to build that model as you move around.



The hololens creates an increasingly detailed model of your space as you move and look around.



Awareness of the world: camera + feature-points.

ARKit/ARCore devices that don't have depth sensors try to lock on to details (features) in the images that they see, and then do complex math in-between each frame to see how those feature points have changed.



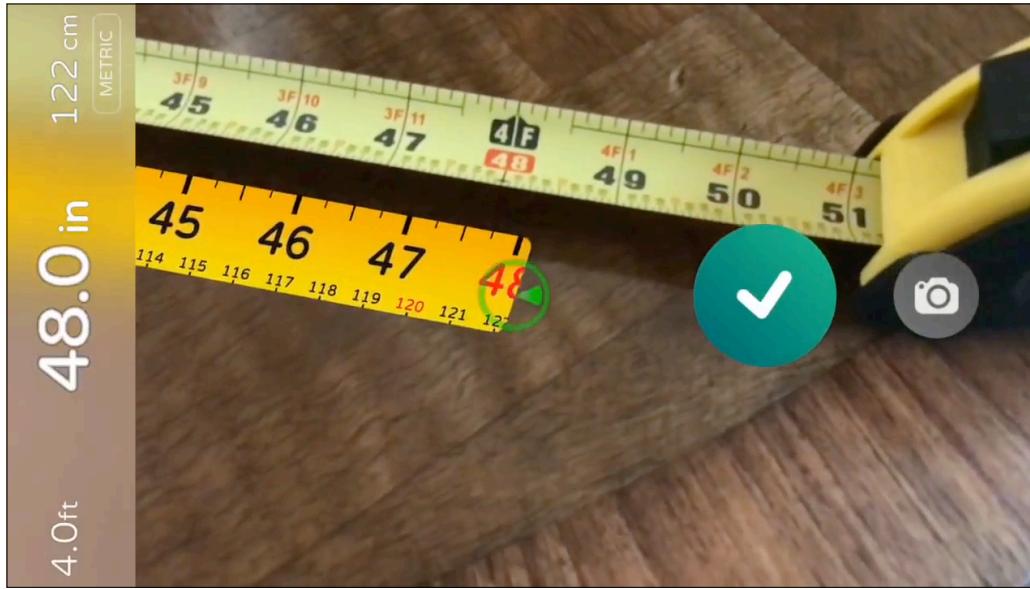
This lets us do some exciting stuff in/to the world.



Games

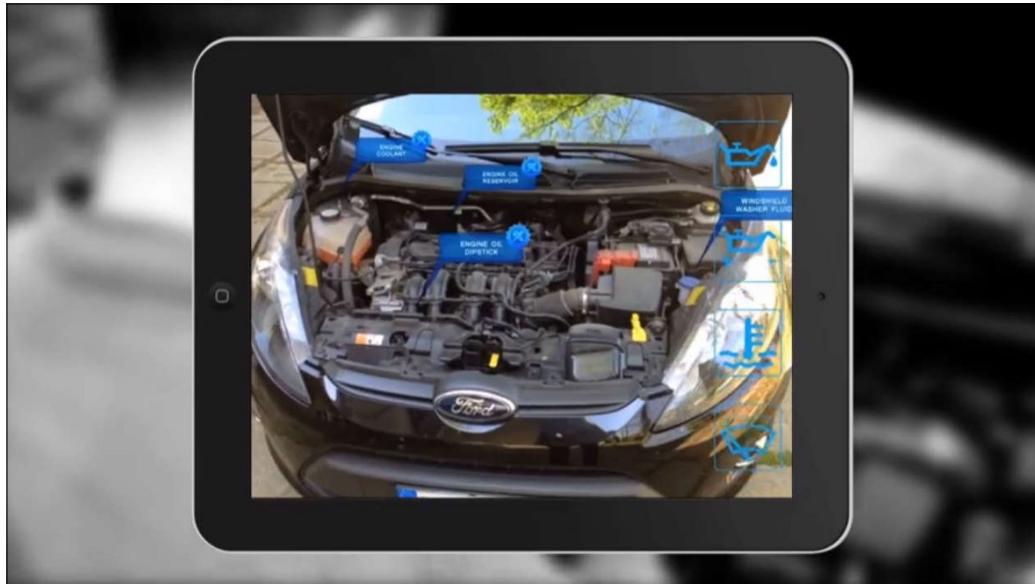


Interactive storytelling.

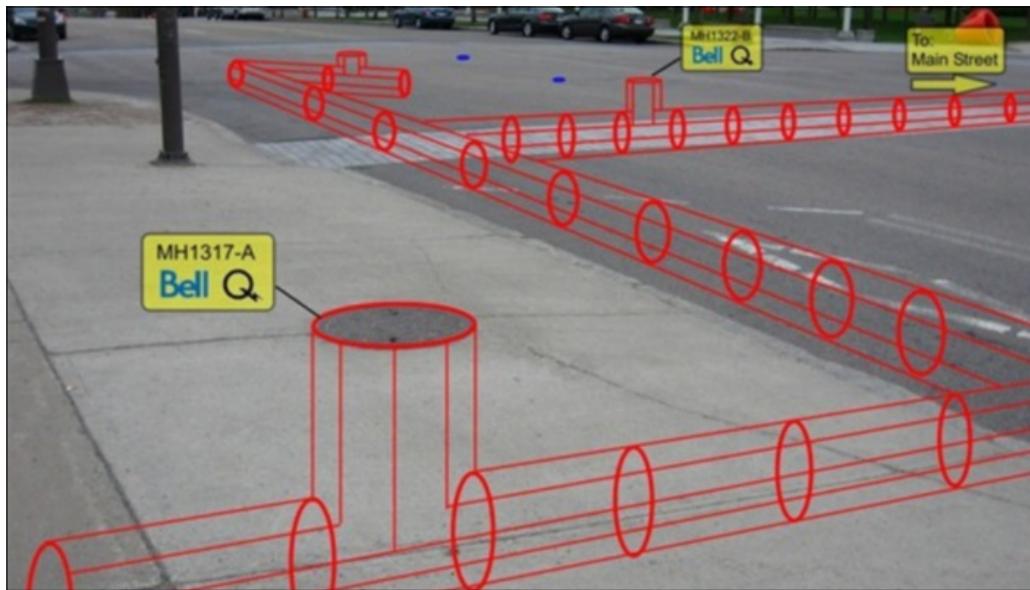


Tools

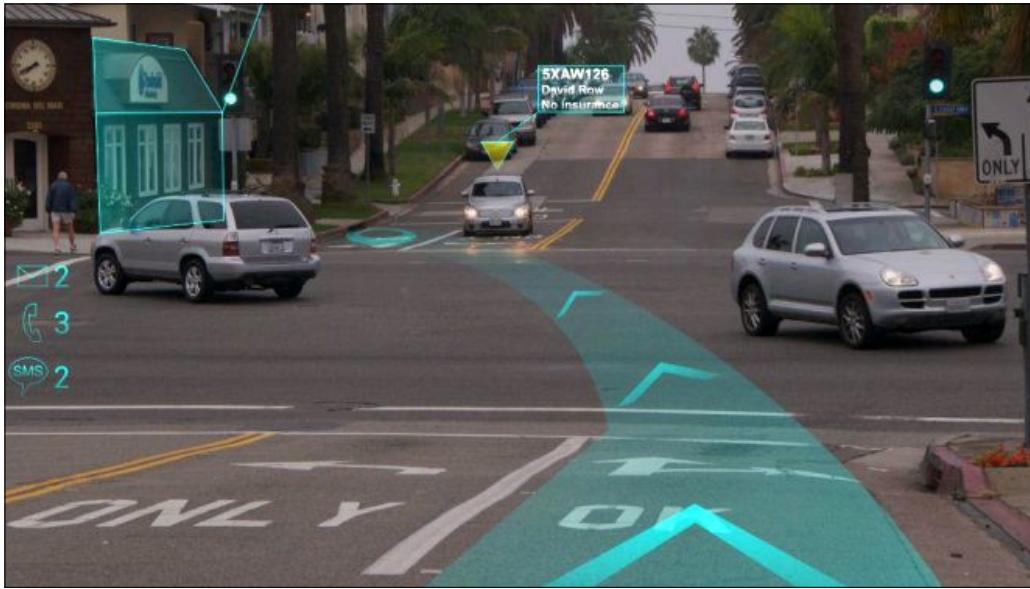
1:1 relationship with real-world scale means AR apps that can measure the environment



Instruction/teaching



Safety/Survey



Directions



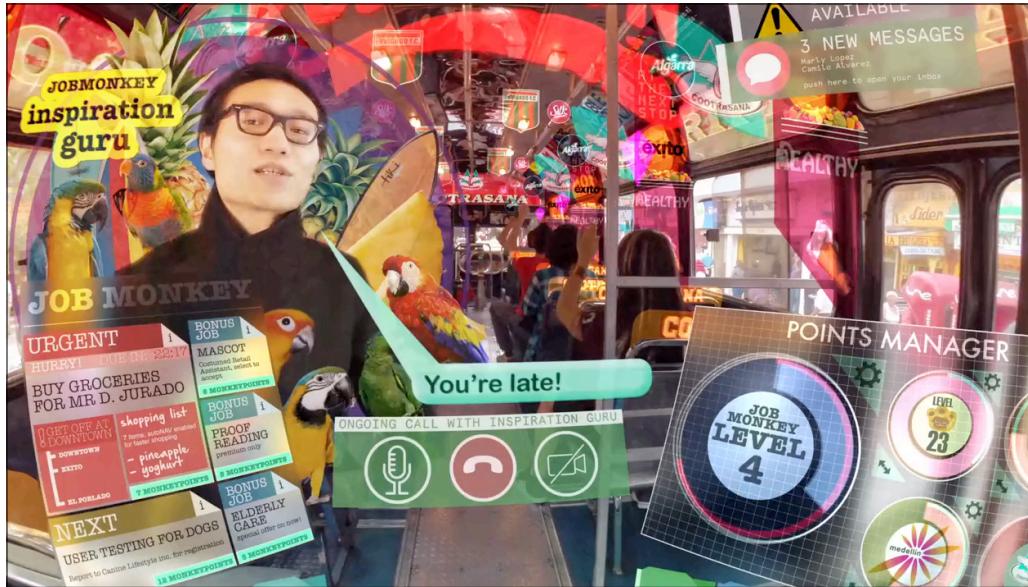
JPL uses Hololens to visualize 1:1 models of rovers, allowing designers to discover relationships and problems before anything is actually built.



You can use Google's Translate app to translate this sign in realtime!



Looking at a slightly darker application. What happens when we go from this...

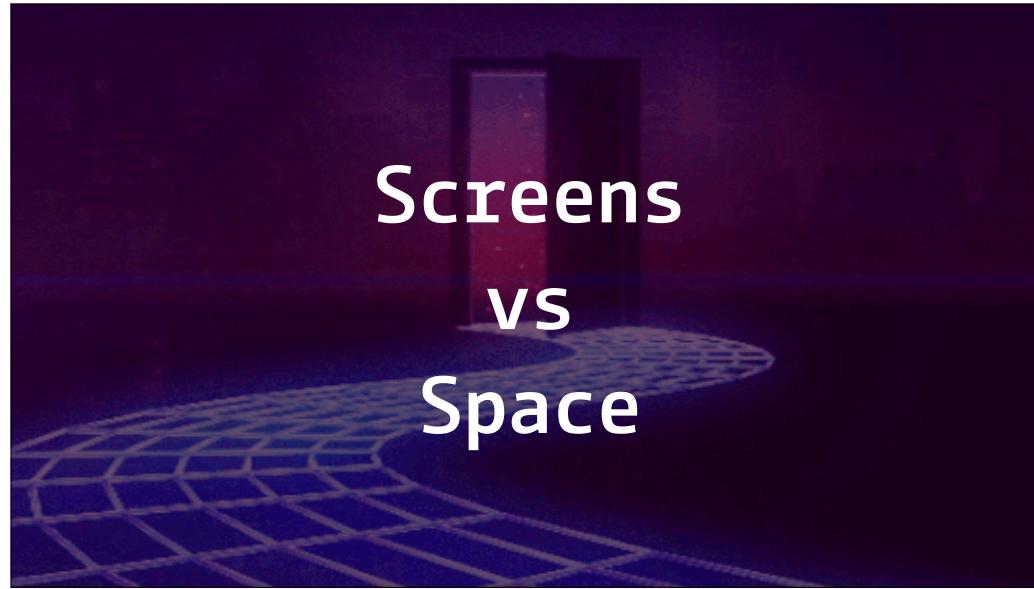


... to this? (i.e. when Times Square follows a user everywhere they go)

— — —
Keiichi Matsuda

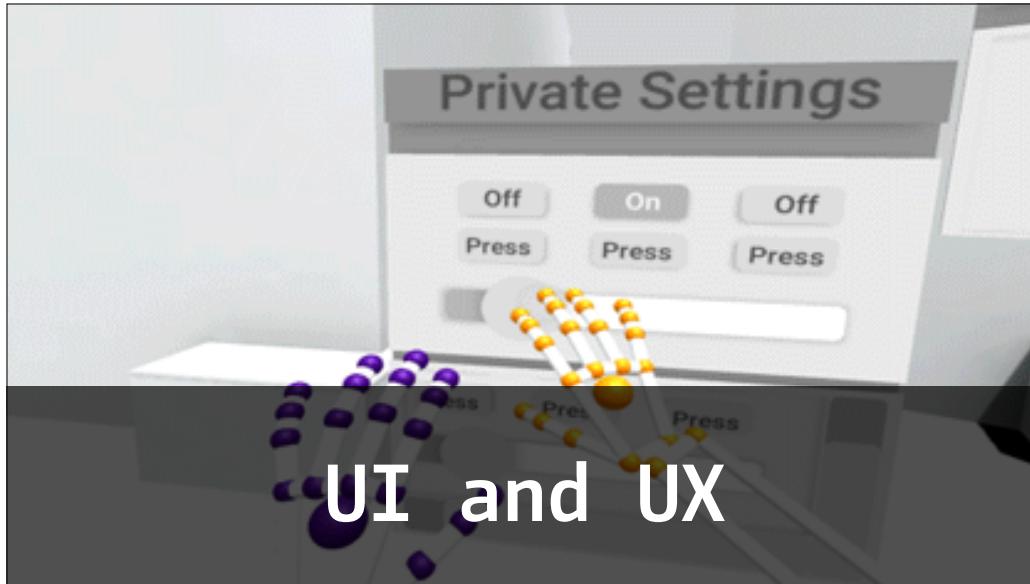
<http://km.cx/projects/hyper-reality>

<https://twitter.com/keiichiban>



Revisiting some concepts...

Our space just got a lot bigger (and smaller)



Revisiting some concepts.

What tools do we have open to us? What *don't* we have?



Diegetic vs Non-Diegetic

Diegetic elements are part of the fictional world ("part of the story"), as opposed to non-diegetic elements which are stylistic elements of how the narrator tells the story ("part of the storytelling").

In movies, subtitles and voiceover are non diegetic. The music coming out of John Cusack's boom box in *Say Anything* is **diegetic...**



Non-Diegetic - HEADS UP DISPLAY (Terminator)

Not attached to or part of anything in the scene - serves as a “narrator” describing things in the scene.



Smarter Objects by Valentin Heun, Shunichi Kasahara, Pattie Maes · MIT Media Lab

Diegetic - Same as VR menu example, but using trackable objects in the real world.



Manipulation

What is the interface?



Moving through an environment to discover content - no direct manipulation.



HoloLens gives us a new tool to play with:

Voice recognition is a very powerful (if socially awkward) way to interact with interface-less experiences.



So what tech will we be using?

Microsoft Hololens...



...Apple ARKit...





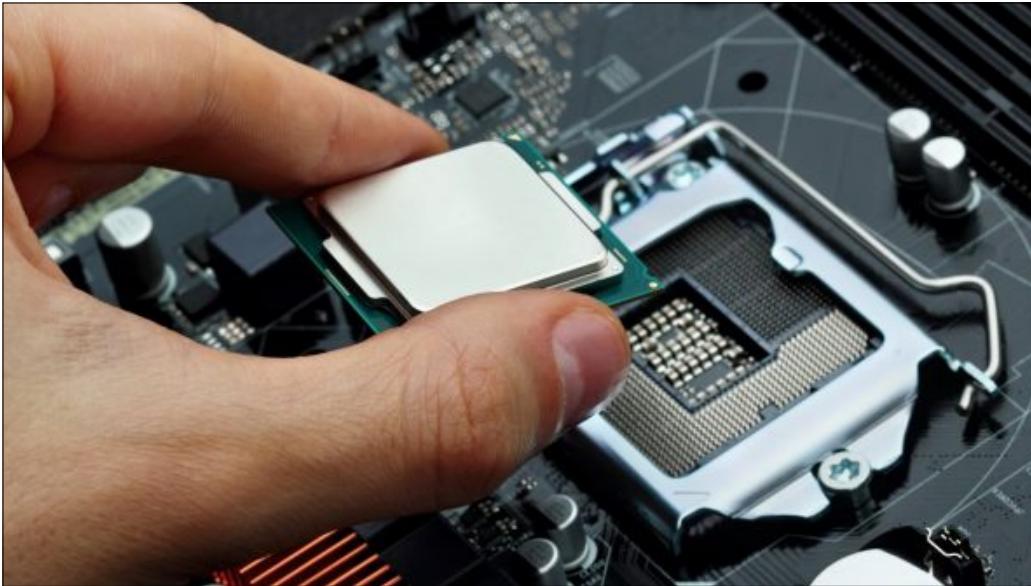
...Vuforia (Built into Unity!)

abstraction

Let's talk (again) about **Abstraction**.

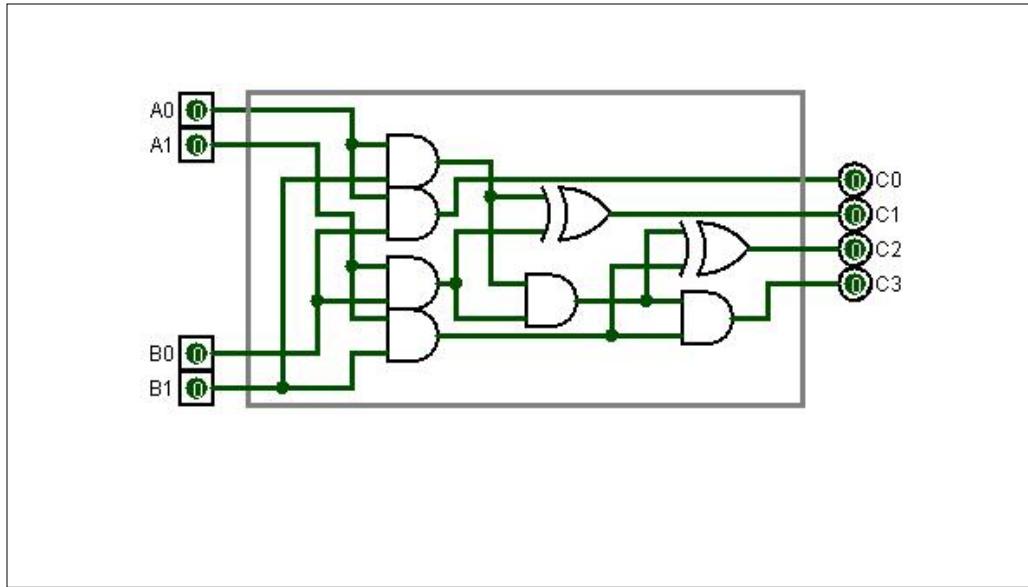
Things are about to get a bit more complex.

Allow me a bit of a tangent...



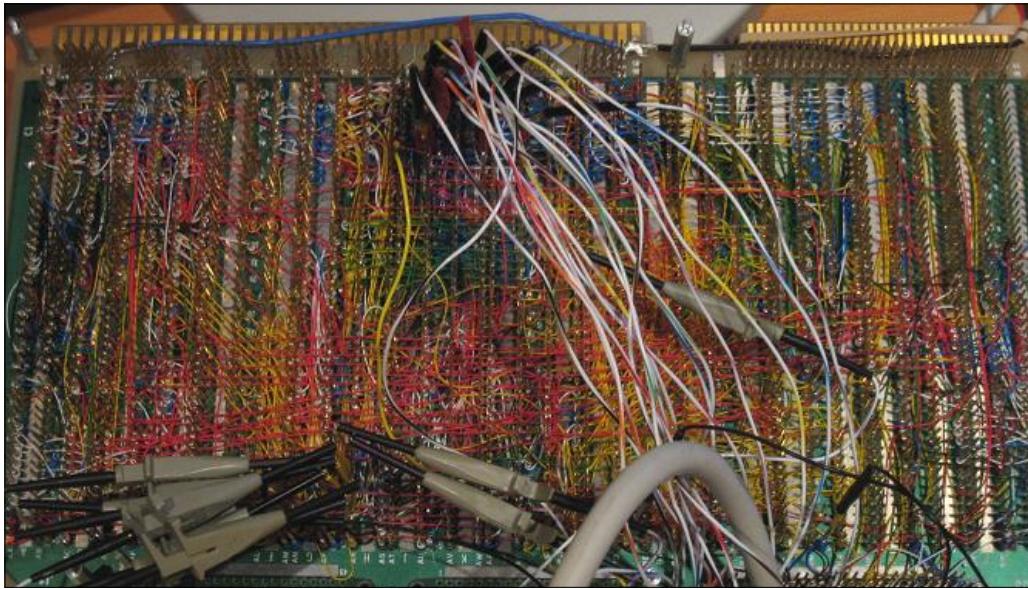
Let's talk about how computers actually work.

What do we really mean by "Computers only understand 0's and 1's"?



Multiply two numbers using **hardware**!

Bunch of AND gates and a couple of XOR (exclusive OR) gates



This is a home-built computer called the BMOW (Big Mess of Wires)

<http://www.bigmessowires.com/bmow1/>

Have you ever thought about how this chunk of metal and wires knows what to do?

```
// UdpPacket provides packetIO over UDP
public class UDPPacketIO {

    private UdpClient Sender;
    private UdpClient Receiver;
    private bool socketsOpen;
    private string remoteHostName;
    private int remotePort;
    private int localPort;

    private string multicastAddress;
    private bool enableMulticast;

    public UDPPacketIO(string hostIP, int remotePort, int localPort, bool enableMulticast=false, string multicastAddress=null) {
        RemoteHostName = hostIP;
        RemotePort = remotePort;
        LocalPort = localPort;

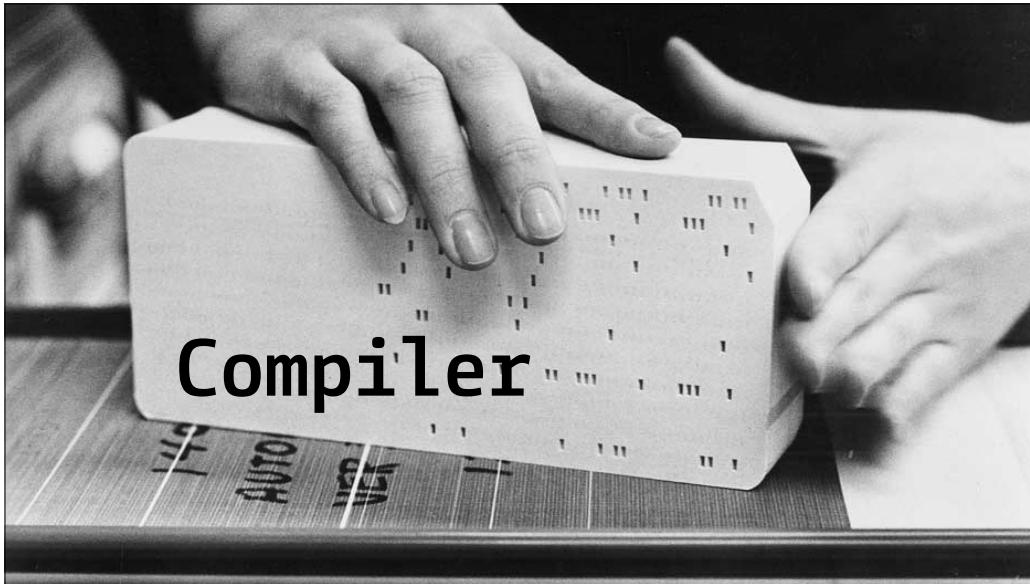
        EnableMulticast = enableMulticast;
        MulticastAddress = multicastAddress;
        socketsOpen = false;
    }

    ~UDPPacketIO() {
        // latest time for this socket to be closed
        if (IsOpen()) {
            BobRemoteControl.BobDebug.Log("closing udpclient listener on port " + localPort);
            Close();
        }
    }
}
```

A programming language is just a text file.

Written according to a standard.

It is meant to be HUMAN READABLE. Computer doesn't care if you use C# or C++ or Java or BrainFuck.



Compiler

```
b8 00 b8 8e c0 8d 36 20 03 e8 fd 01 bf a2 00 b9
02 00 eb 2b b4 06 b2 ff cd 21 3c 71 0f 84 e5 01
3c 50 b9 a0 00 74 18 3c 48 b9 a0 00 0f 84 d9 00
b9 02 00 3c 4d 74 08 3c 4b 0f 84 cc 00 eb d5 89
3e b5 09 01 cf 89 3e b3 09 e8 87 01 8b 3e b5 09
b0 20 26 88 05 26 88 45 fe 26 88 85 62 ff 26 88
85 60 ff 26 88 85 5e ff 26 88 85 9e 00 b0 07 26
88 45 01 8b 3e b3 09 89 fb 83 eb 02 d1 fb 8a 00
26 88 45 fe 89 fb 81 eb a2 00 d1 fb 8a 00 26 88
85 5e ff 89 fb 81 eb a0 00 d1 fb 8a 00 26 88 85
60 ff 89 fb 81 eb 9e 00 d1 fb 8a 00 26 88 85 62
ff 89 fb 81 eb a2 00 d1 fb 8a 00 26 88 85 5e ff
89 fb 83 c3 02 d1 fb 8a 00 26 88 45 02 89 fb 81
c3 9e 00 d1 fb 8a 00 26 88 85 9e 00 89 fb 81 c3
a0 00 d1 fb 8a 00 26 88 85 a0 00 89 fb 81 c3 a2
00 d1 fb 8a 00 26 88 85 a2 00 b0 03 26 88 05 a0
b7 09 26 88 45 01 e9 0b ff 89 3e b5 09 29 cf 89
3e b3 09 e8 bd 00 8b 3e b5 09 b0 20 26 88 05 26
88 45 02 26 88 85 9e 00 26 88 85 a0 00 26 88 85
a2 00 26 88 85 62 ff b0 07 26 88 45 01 8b 3e b3
09 89 fb 83 eb 02 d1 fb 8a 00 26 88 45 fe 89 fb
81 eb a2 00 d1 fb 8a 00 26 88 85 5e ff 89 fb 81
eb a0 00 d1 fb 8a 00 26 88 85 60 ff 89 fb 81 eb
9e 00 d1 fb 8a 00 26 88 85 62 ff 89 fb 81 eb a2
00 d1 fb 8a 00 26 88 85 5e ff 89 fb 83 c3 02 d1
```

Machine-code does not "communicate with the processor".

Rather, the processor "knows how to evaluate" machine-code.

The CPU "looks" at the current instruction. When the instructions are executed side-effects occur such as setting a control flag, putting a value in a register, or jumping to a different index (changing the PC) in the program, etc.

It is the evaluation of each instruction -- as it is encountered -- and the interaction of side-effects that results in the operation of a traditional processor.



Visual Studio and XCode

We can't just finish in Unity anymore! Because these AR devices are running on different processors than our PC, we have to go through another program that will build the final executable for the device.

Some extra nerdiness:

PC/Mac Standalone (and VR):

C# -> IL -> Bytecode -> JIT Compiler -> Machine Code

Using IL2CPP bridge:

C# -> IL -> IL2CPP -> C++ -> Anywhere!

C++ -> XCode -> iOS/Apple A9 processor Machine Code



TECH 1711 - Mixed Reality Studio

Thank you!