ARFIT
(Augment Reality for IT)

12/4/2018

Presented by: Edward Chrzanowski
CS

WatITis

Let me break down the Title

AR - Augmented Reality

for

IT - Information Technology

Definitions - IT

IT is More than Computers and Networks

While IT is often used to describe computers and computer networks, it actually includes all layers of all systems within an organization — from the physical hardware to the operating systems, applications, databases, storage, servers and more. It encompasses all forms of technology used to create, store, exchange, and use **information** in its various forms.

Augmented Reality

- So, AR, of course, stands for augmented reality (and not pirate talk).
- The most widely accepted definition is that AR must have these characteristics:
  - combines real and virtual
  - interactive in real time
  - registered in 3D
- So, this means that AR has to deal with things in the real world.
  - They happen in the room around you, in the environment you're in, or in your place on the map.
- And the other part of augmented reality, is augmented.
- Augmented can mean what you know about the world.
  - It can mean getting information that the device can understand about the world, and giving that to you.
  - It can mean placing virtual things out into the world, giving the virtual this physical context.
- Taking something that is real, and enhancing it, augmenting it.

Differences between AR and VR

Virtual Reality blocks out the real world

The idea is to block out the sensory input from the outside and use visual and auditory cues to make the virtual world seem more real.

Augmented Reality enhances the real world

Augmented Reality has the capability of superimposing information on the world we see. Hence, all what this technology does is to add to the reality you ordinarily see rather than replacing it.

Virtual Reality limits your mobility

With Virtual Reality you psychologically escape the physical world and enter a virtual one.

Augmented Reality is mobile

Augmented Reality is also mobile in the sense that it can live and breathe on our current wearable devices.

Where is AR used today

As an example,

- AR “heads-up” displays
  - navigation, collision warning, and other information directly in drivers’ line of sight are now available in dozens of car models.
- AR wearable devices for factory workers
  - that superimpose production-assembly or service instructions are being piloted at thousands of companies.

AR is supplementing or replacing traditional manuals and training methods at an ever-faster pace.

So how can we in the Information Technology sector utilize this technology to our advantage.

This is where this presentation takes us. But just a bit of history.

history

Sword of Damocles (1968)

The Sword of Damocles was the nickname of the world’s first head-mounted display, built in 1968.

Worn by University of Utah student Donald Vickers

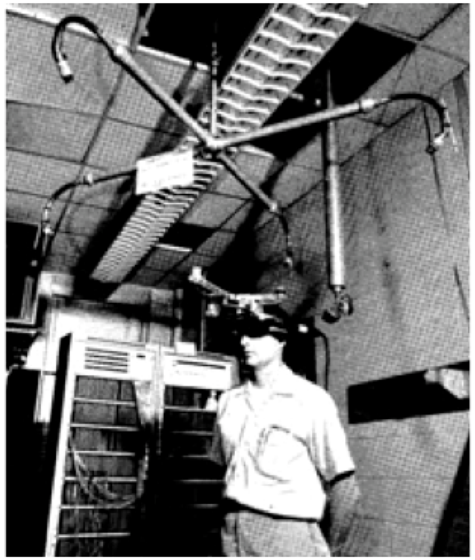


Image: Ivan Sutherland

history

Boing Wire Harness Assembly (1992)

Researchers at Boing used a see-through HMD to guide the assembly of wire bundles for aircraft.

Image: David Mizell

history

KARMA (1993)

KARMA was the first knowledge-driven AR application.
A user with an HMD could see instructions on printer maintenance

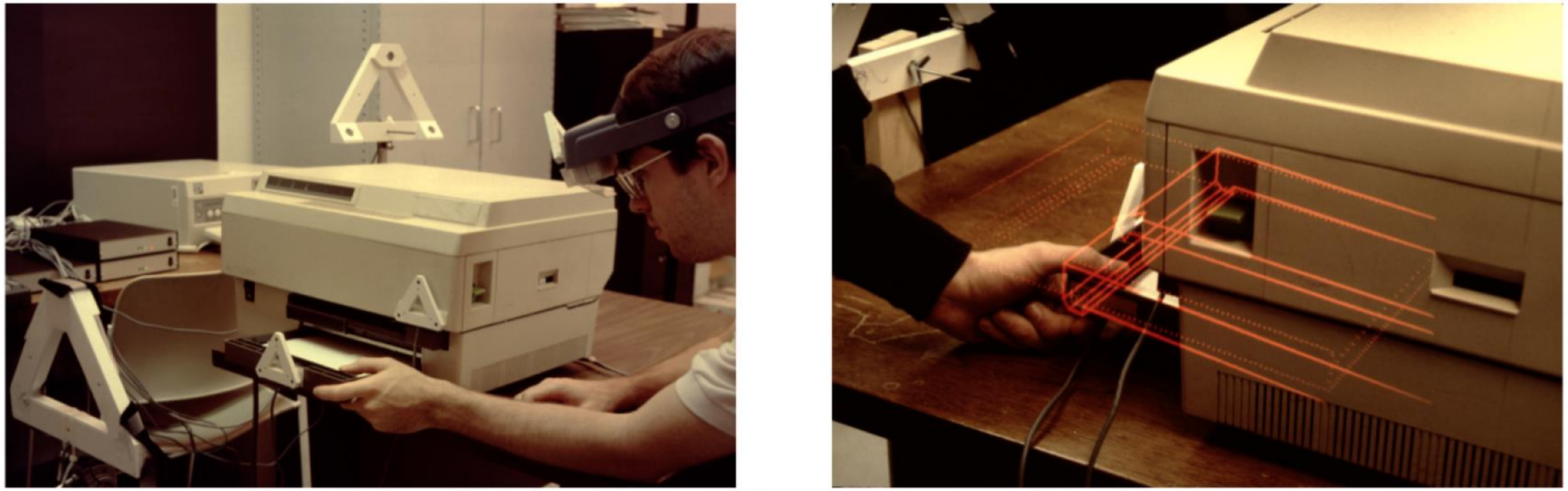
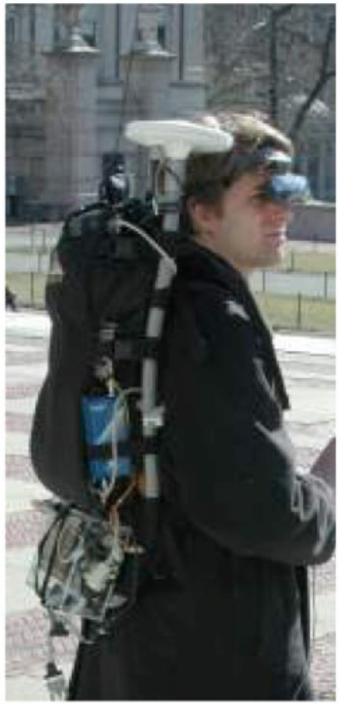


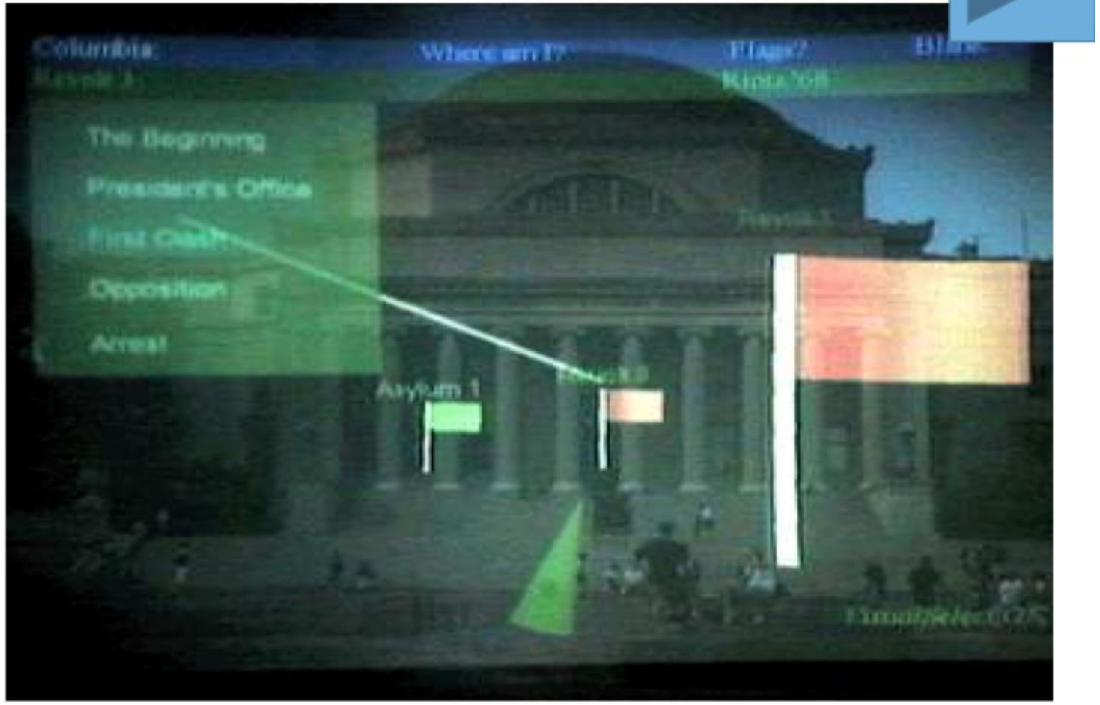
Image: Steve Feiner, Blair MacIntyre and Doreé Seligmann, Columbia University

history

Touring Machine (1997)



Touring Machine, the 1st outdoor AR system

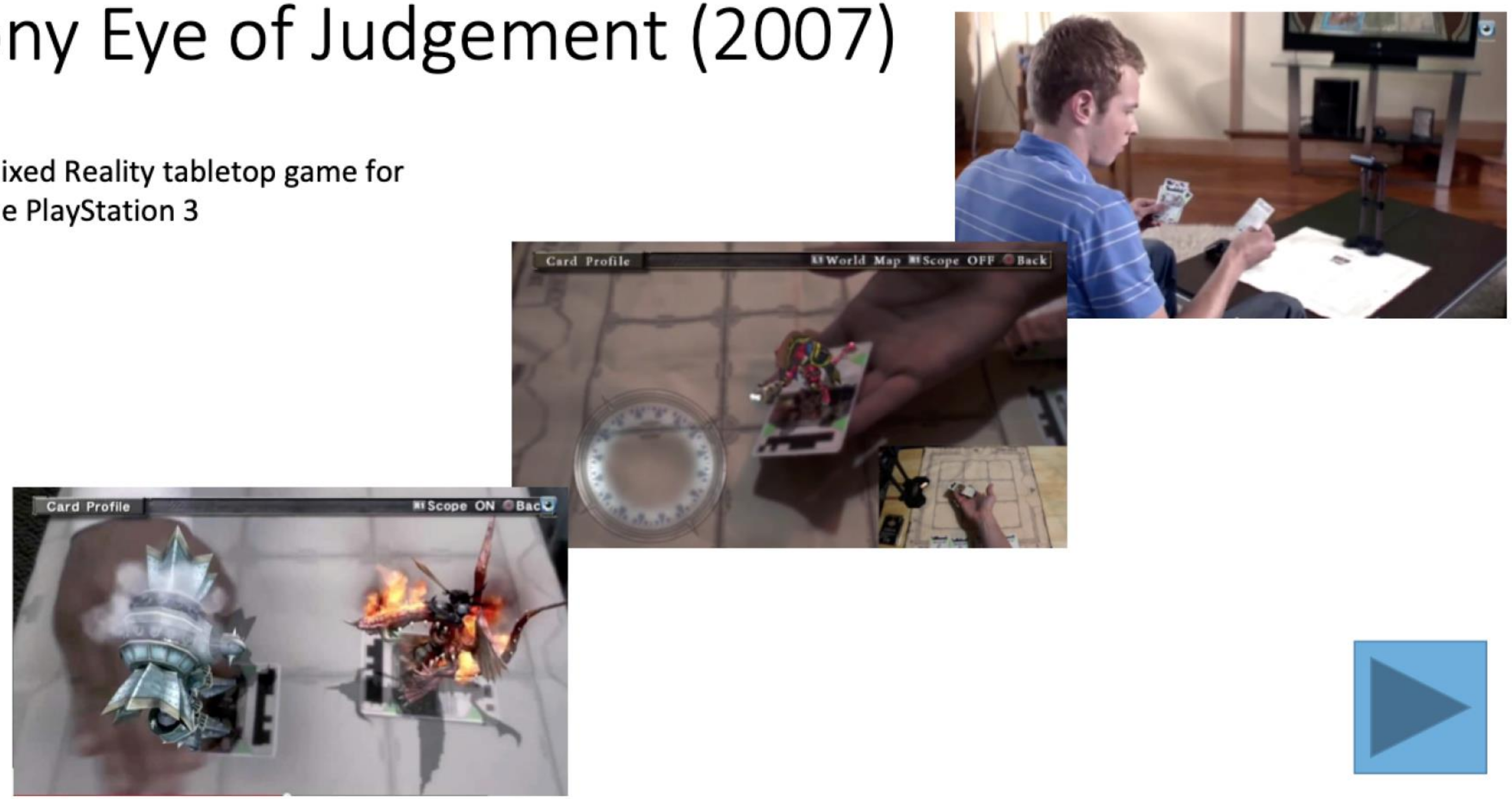


Situated Documentaries , a campus tour guide running on the Touring Machine

example

Sony Eye of Judgement (2007)

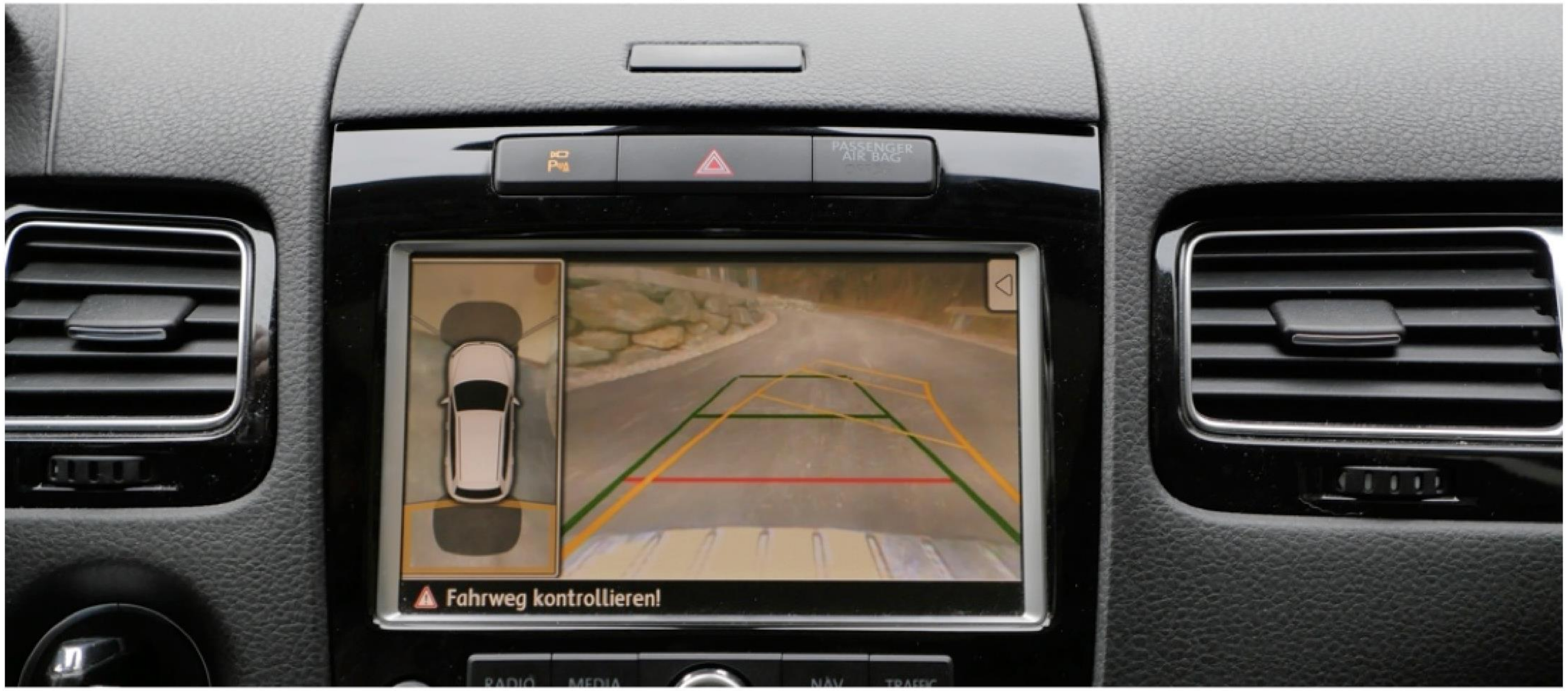
Mixed Reality tabletop game for the PlayStation 3



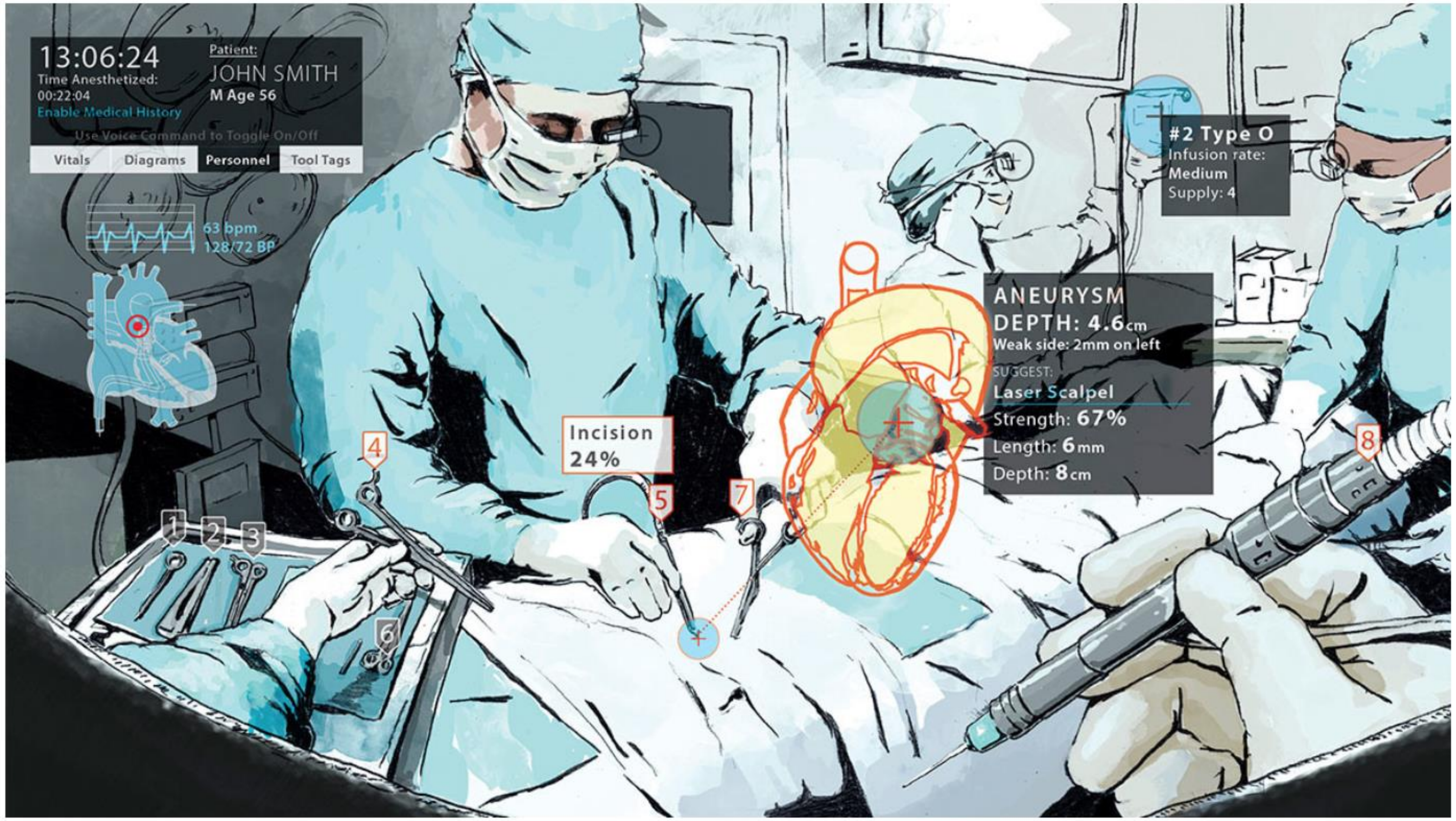
example

Parking Assistant

The parking assistant is a commercially available AR feature in many contemporary cars



medical example



Just Visualization?

You walk down the carpeted hallway and into the movie theater… you find your seat…
The lights dim… you clutch your bucket of popcorn to your chest… open your eyes…
and… surprise! The smell of **freshly cut grass** sucks you right into the film.
No one will forget a movie they've seen with **Olorama Technology**TM!

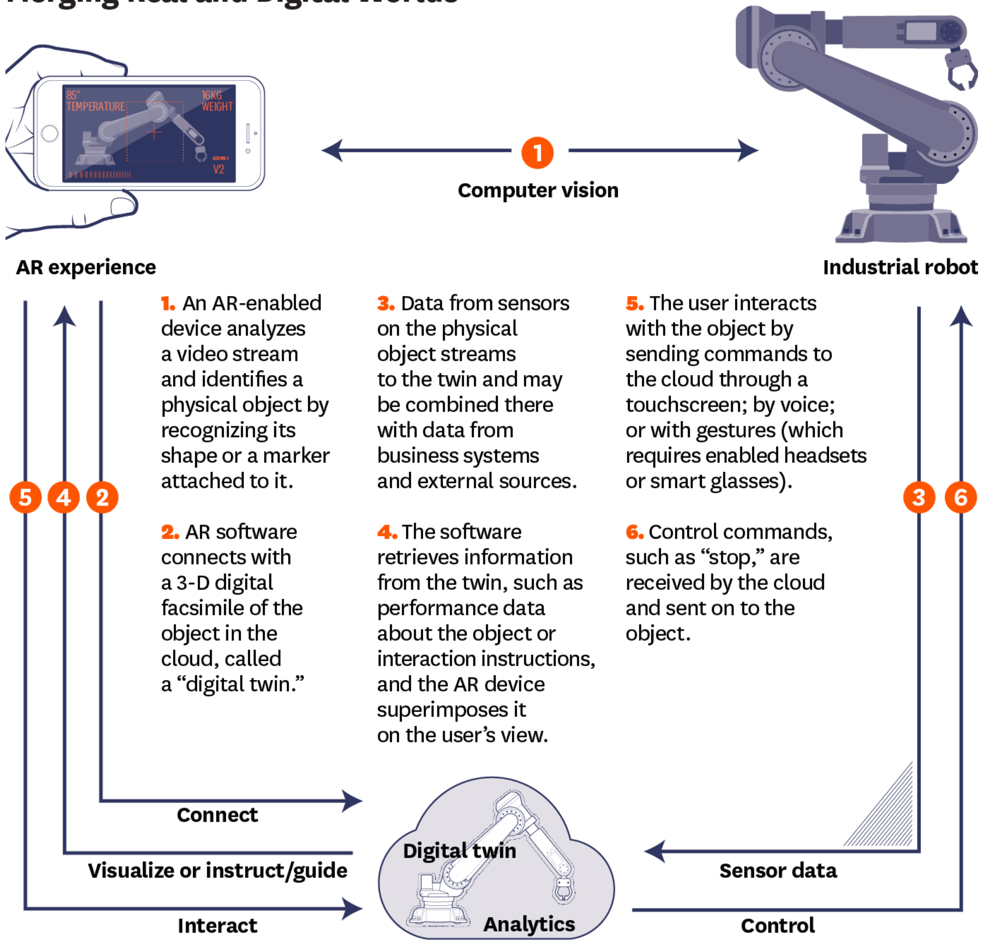
WatITis

This Code Stinks

- paper I wrote a few years ago

How does AR work?

Merging Real and Digital Worlds



1. An AR-enabled device analyzes a video stream and identifies a physical object by recognizing its shape or a marker attached to it.

2. AR software connects with a 3-D digital facsimile of the object in the cloud, called a “digital twin.”

3. Data from sensors on the physical object streams to the twin and may be combined there with data from business systems and external sources.

4. The software retrieves information from the twin, such as performance data about the object or interaction instructions, and the AR device superimposes it on the user’s view.

5. The user interacts with the object by sending commands to the cloud through a touchscreen; by voice; or with gestures (which requires enabled headsets or smart glasses).

6. Control commands, such as “stop,” are received by the cloud and sent on to the object.

ILLUSTRATION BY CLINT FORD
FROM “HOW DOES AUGMENTED REALITY WORK?”
BY MICHAEL E. PORTER AND JAMES E. HEPPELMANN, NOVEMBER–DECEMBER 2017

Type to enter a caption.

Now we have Thalmic (North)

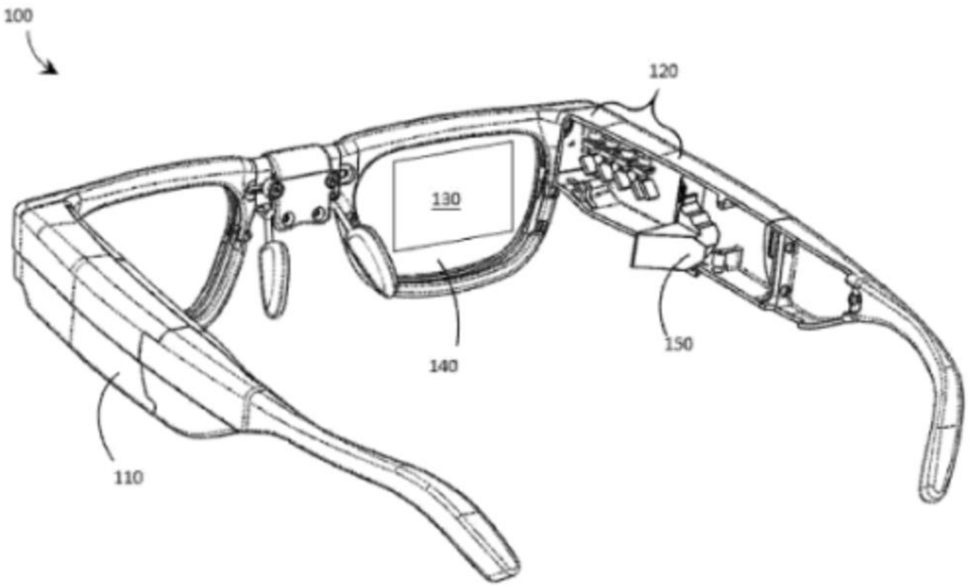


FIGURE 1

An image from a Thalmic patent indicating how the company will cram all that tech into the smart glasses' temple.

How to get started?

- Toolkits
    - ARKit - Apple

“Build unparalleled augmented reality experiences for hundreds of millions of users on iOS — the biggest AR platform in the world.”

    - ARcore - Google

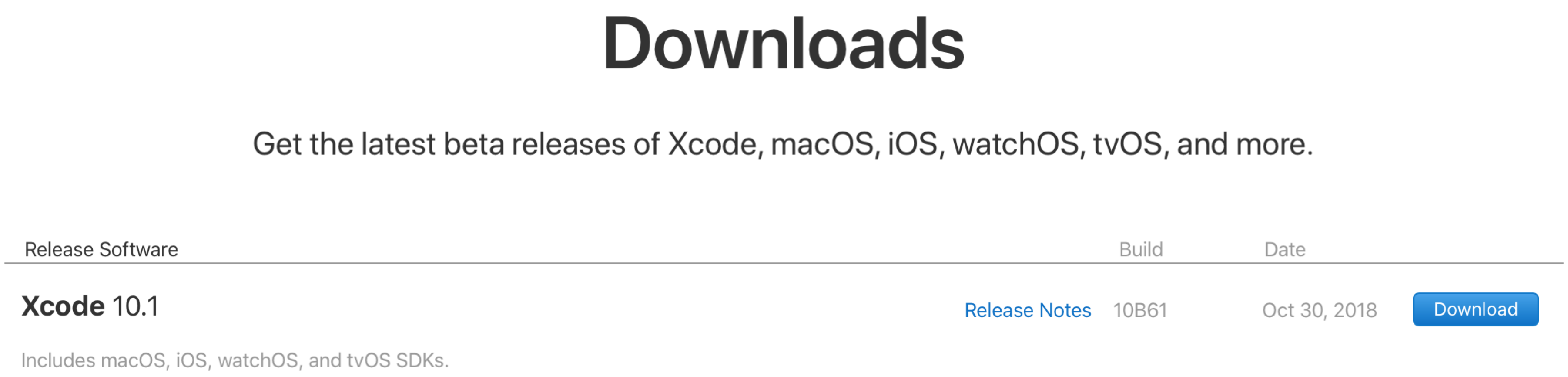
“With ARCore, build new augment reality experiences that seamlessly blend the digital and physical worlds. Transform the way people play, shop, learn, create, and experience the world together - at Google scale.”

- Real world

ARKit

- The basic requirement for any AR experience—and the defining feature of ARKit—is the ability to create and track a correspondence between the real-world space the user inhabits and a virtual space where you can model visual content.
- When your app displays that content together with a live camera image, the user experiences augmented reality: the illusion that your virtual content is part of the real world.
- Real world tracking uses visual-inertial odometry (or SLAM - Simultaneous localization and Mapping) - depth image integration (i.e. truncated signed distance function)
    - real world tracking is an inexact science. Some caveats are:
        - image analysis - tracking quality is reduced when the camera cannot see details
        - device motion - excessive motion results in blurred images - ARCamera provides an UI to help resolve this.
        - anchor point - choose wisely

ARKit SDK



Downloads

Get the latest beta releases of Xcode, macOS, iOS, watchOS, tvOS, and more.

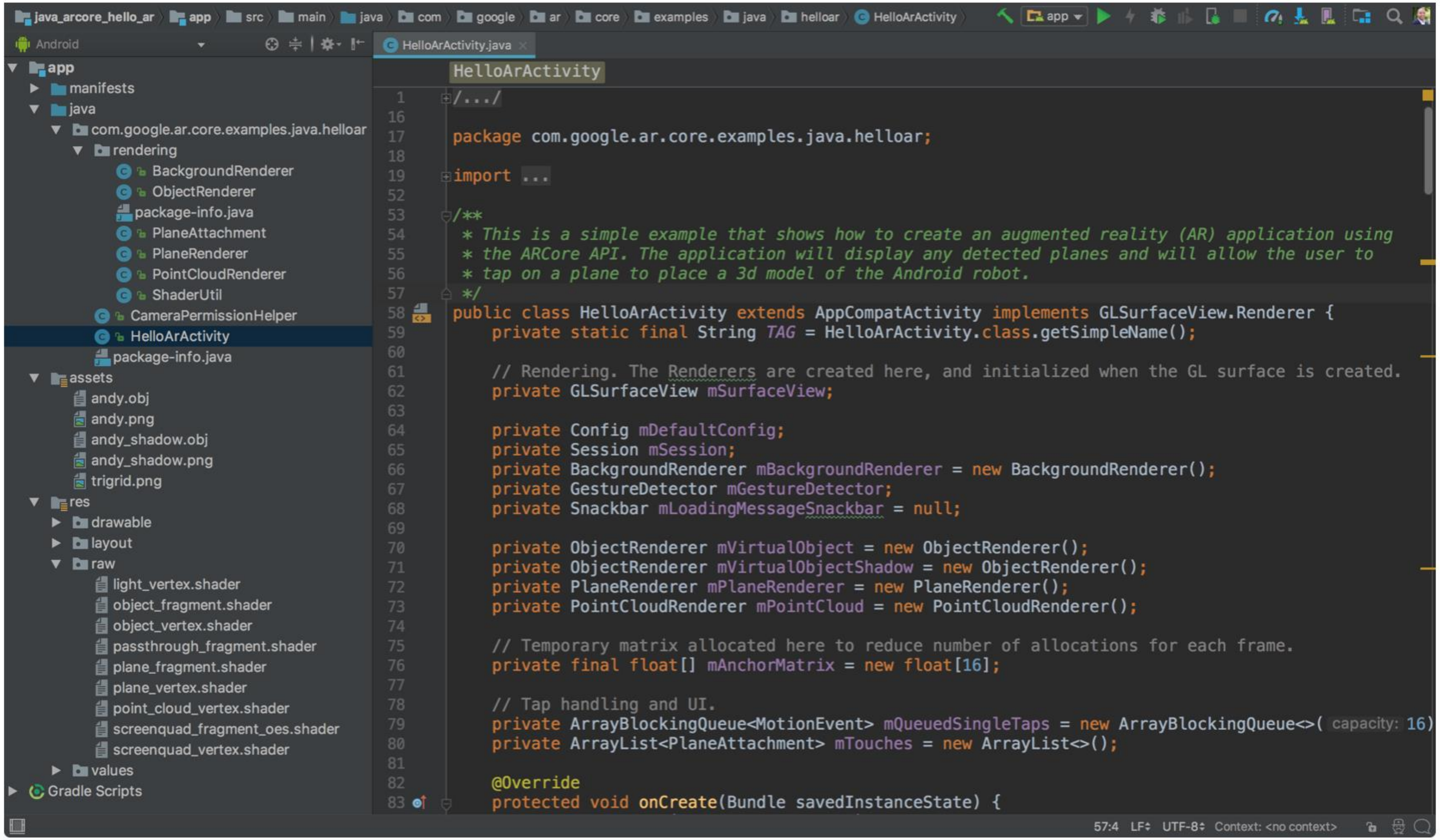
Release Software		Build	Date	
Xcode 10.1 Includes macOS, iOS, watchOS, and tvOS SDKs.	Release Notes	10B61	Oct 30, 2018	Download

ARCore

- ARCore is Android’s version of Apple ARKit. It's a baked-in augmented reality platform that developers can leverage.
- As of February 2018, Google launched the 1.0 software developer kit (SDK), allowing third-party developers to more easily create AR apps for Android phones.
- With the ARCore SDK for iOS (August 2018), you can build ARKit apps targeting iOS devices that are also supported by ARCore.

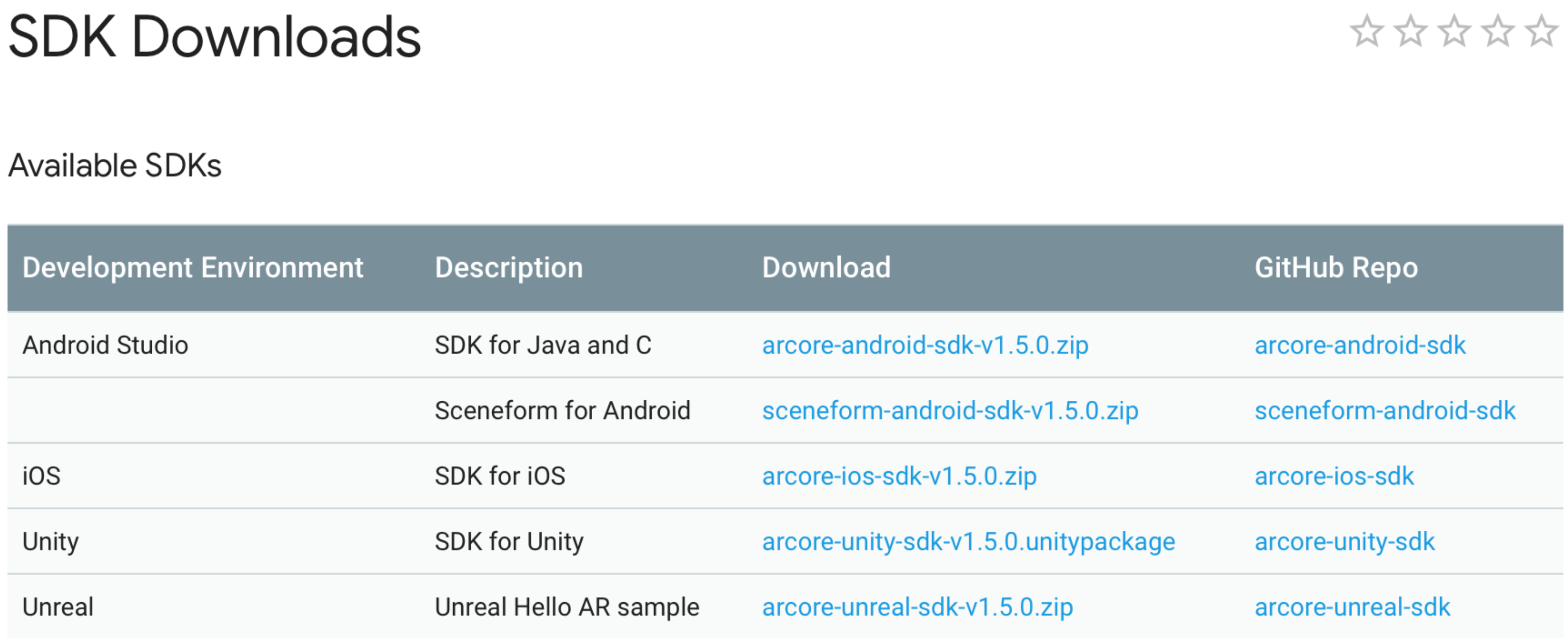
- Quickstart: Android
- Quickstart: Android NDK
- Quickstart: Unity for Android
- Quickstart: Unity for iOS
- Quickstart: iOS
- Quickstart: Unreal

A Java example using ARCore



```java
/.../

package com.google.ar.core.examples.java.helloar;

import ...

/**
 * This is a simple example that shows how to create an augmented reality (AR) application using
 * the ARCore API. The application will display any detected planes and will allow the user to
 * tap on a plane to place a 3d model of the Android robot.
 */
public class HelloArActivity extends AppCompatActivity implements GLSurfaceView.Renderer {
    private static final String TAG = HelloArActivity.class.getSimpleName();

    // Rendering. The Renderers are created here, and initialized when the GL surface is created.
    private GLSurfaceView mSurfaceView;

    private Config mDefaultConfig;
    private Session mSession;
    private BackgroundRenderer mBackgroundRenderer = new BackgroundRenderer();
    private GestureDetector mGestureDetector;
    private Snackbar mLoadingMessageSnackbar = null;

    private ObjectRenderer mVirtualObject = new ObjectRenderer();
    private ObjectRenderer mVirtualObjectShadow = new ObjectRenderer();
    private PlaneRenderer mPlaneRenderer = new PlaneRenderer();
    private PointCloudRenderer mPointCloud = new PointCloudRenderer();

    // Temporary matrix allocated here to reduce number of allocations for each frame.
    private final float[] mAnchorMatrix = new float[16];

    // Tap handling and UI.
    private ArrayBlockingQueue<MotionEvent> mQueuedSingleTaps = new ArrayBlockingQueue<>( capacity: 16)
    private ArrayList<PlaneAttachment> mTouches = new ArrayList<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

Arcore SDK's



SDK Downloads

☆☆☆☆☆

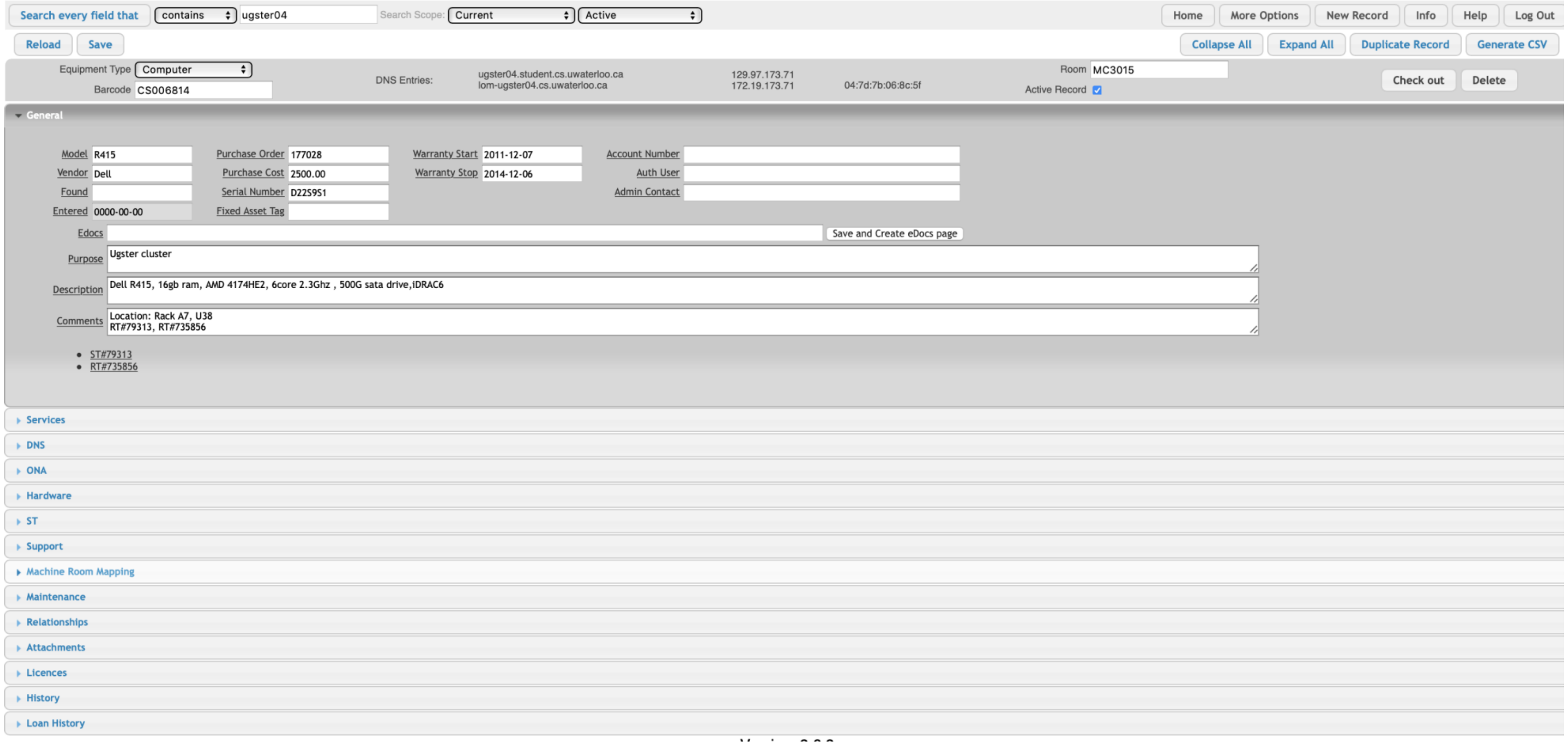
Available SDKs

Development Environment	Description	Download	GitHub Repo
Android Studio	SDK for Java and C	arcore-android-sdk-v1.5.0.zip	arcore-android-sdk
	Sceneform for Android	sceneform-android-sdk-v1.5.0.zip	sceneform-android-sdk
iOS	SDK for iOS	arcore-ios-sdk-v1.5.0.zip	arcore-ios-sdk
Unity	SDK for Unity	arcore-unity-sdk-v1.5.0.unitypackage	arcore-unity-sdk
Unreal	Unreal Hello AR sample	arcore-unreal-sdk-v1.5.0.zip	arcore-unreal-sdk

And we have the Real world

- The Information Technology world covers all layers of all systems within an organization — from the physical hardware to the operating systems, applications, databases, storage, servers and more. It encompasses all forms of technology used to create, store, exchange, and use **information.**
- Now to an example I was working on …

inventory page



Search every field that: contains ugster04 Search Scope: Current Active

Equipment Type: Computer

Barcode: CS006814

DNS Entries: ugster04.student.cs.uwaterloo.ca 129.97.173.71
lom-ugster04.cs.uwaterloo.ca 172.19.173.71 04:7d:7b:06:8c:5f

Room: MC3015

Active Record: ☑

Model	R415	Purchase Order	177028	Warranty Start	2011-12-07	Account Number	
Vendor	Dell	Purchase Cost	2500.00	Warranty Stop	2014-12-06	Auth User	
Found		Serial Number	D22S9S1			Admin Contact	
Entered	0000-00-00	Fixed Asset Tag					

Edocs:

Purpose: Ugster cluster

Description: Dell R415, 16gb ram, AMD 4174HE2, 6core 2.3Ghz , 500G sata drive,iDRAC6

Comments: Location: Rack A7, U38
RT#79313, RT#735856

- ST#79313
- RT#735856

CVS output from inventory

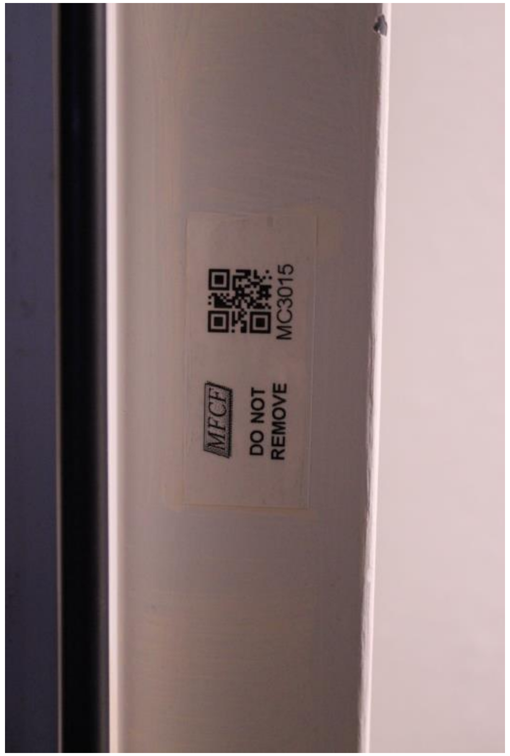
```
elegans1,172.19.154.51,,,,MC3015,
chardonnay-login,129.97.152.130,,https://cs.uwaterloo.ca/twiki/view/CFPrivate/chardonnay-login,,MC3015,
chardonnay-login-ilom,10.0.154.32,,,,MC3015,
chardonnay-ilom,10.0.154.33,,https://cs.uwaterloo.ca/twiki/view/CFPrivate/chardonnay,,MC3015,
"ugster15, lom-ugster15","129.97.173.82, 172.19.173.82",CS008826,,"Dell R415",MC3015,
"ugster17, lom-ugster17","129.97.173.84, 172.19.173.84",CS008813,,"Dell R415",MC3015,
"ugster14, lom-ugster14","129.97.173.81, 172.19.173.81",CS008839,,"Dell R415",MC3015,
"ugster13, lom-ugster13","129.97.173.80, 172.19.173.80",CS008807,,"Dell R415",MC3015,
"ugster12, lom-ugster12","129.97.173.79, 172.19.173.79",CS008805,,"Dell R415",MC3015,
chardonnay,129.97.152.131,,https://cs.uwaterloo.ca/twiki/view/CFPrivate/chardonnay,,MC3015,
mc-3015-h7-ups-2,10.15.134.6,CS009013,,,MC3015,
mc-3015-e7-ups-1,10.15.134.9,CS008930,,,MC3015,
,,CS008908,,,MC3015,
,,CS006724,,,MC3015,
cs-teaching,10.10.154.5,,,,MC3015,
ceph-mon-210,10.10.154.119,,,,MC3015,
ceph-mon-211,10.10.154.118,,,,MC3015,
ceph-osd-210,10.10.154.120,,,,MC3015,
ceph-osd-212,10.10.154.122,,,,MC3015,
ceph-osd-213,10.10.154.123,,,,MC3015,
```

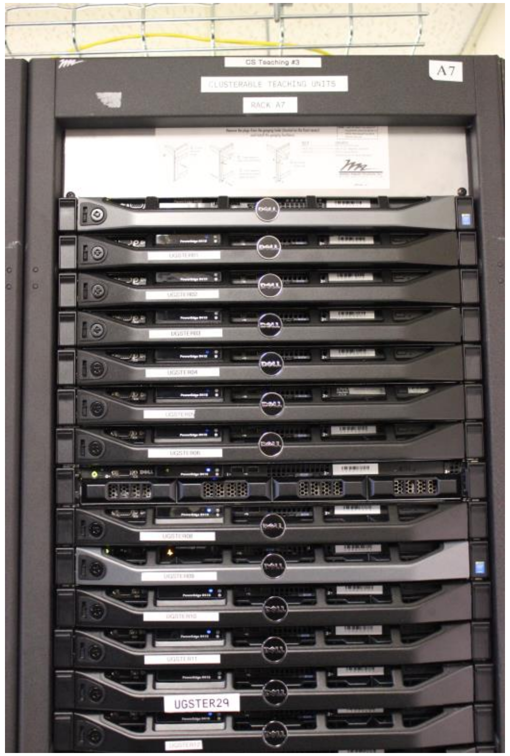
Vendor Information

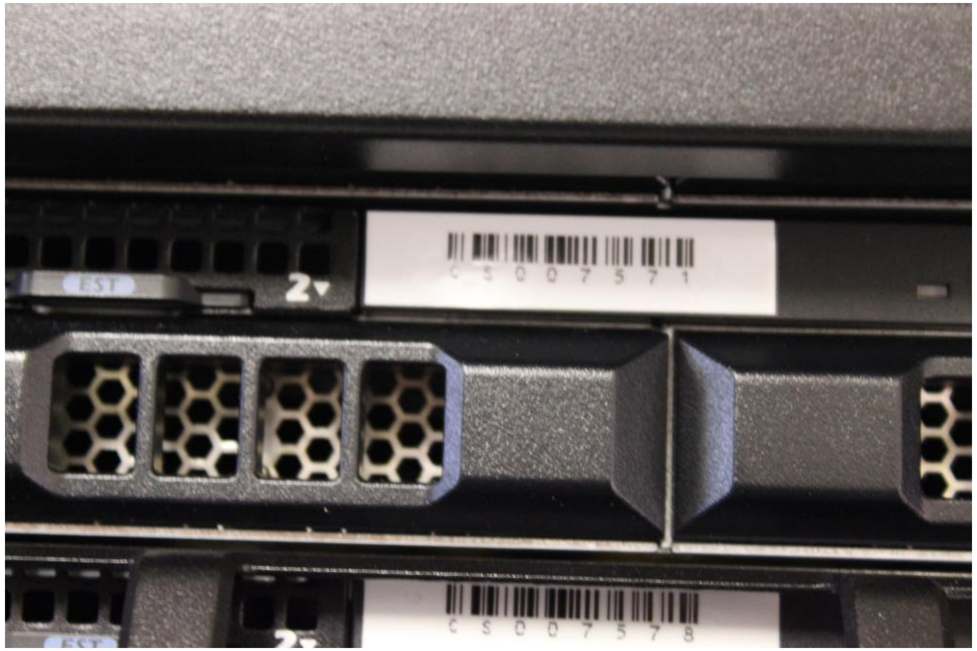
- Technical guides and specifications from vendor or experience.
- example:
  - https://www.dell.com/downloads/global/products/pedge/en/Poweredge-r415-technicalguide.pdf

anchors

- triggers in the environment - room and rack mapping







Combining

- we have toolkits
- we have information (cvs or other inventory database)
  - we have inventory data base
  - we have machine twikis
  - we have manufactures manuals

USDZ - the glue

- USDZ, which stands for Universal Scene Description is a container file format for storing, sharing and working on 3D files for mobile devices.
    - Universal Scene Description (USD) is the first publicly available software that addresses the need to robustly and scalably interchange and augment arbitrary 3D scenes that may be composed from many elemental assets.
    - Because USD's core scenegraph and "composition engine" are agnostic of 3D, USD can be extended in a maintainable way to encode and compose data in other domains.
    - remember the inventory page - we saw a heading for “machine room mapping”
- The USDZ format is open sourced developed by Pixar.
- https://graphics.pixar.com/usd/docs/Usdz-File-Format-Specification.html
- Apple announce they would adapt USDZ file format for 3D content across its AR-enabled iPhones and iPads, which can display illusory, animated images placed in real surroundings.
- In July 2018 you can use this format for Android devices.
- In iOS 12, the system provides an AR view for 3D objects when you use QLPreviewController with USDZ files in an app, or use Safari or WebKit with USDZ files in web content.
- So knowing that you can use USDZ files in web content we can write some code like this:

USDZ file format

Looks similar to HTML

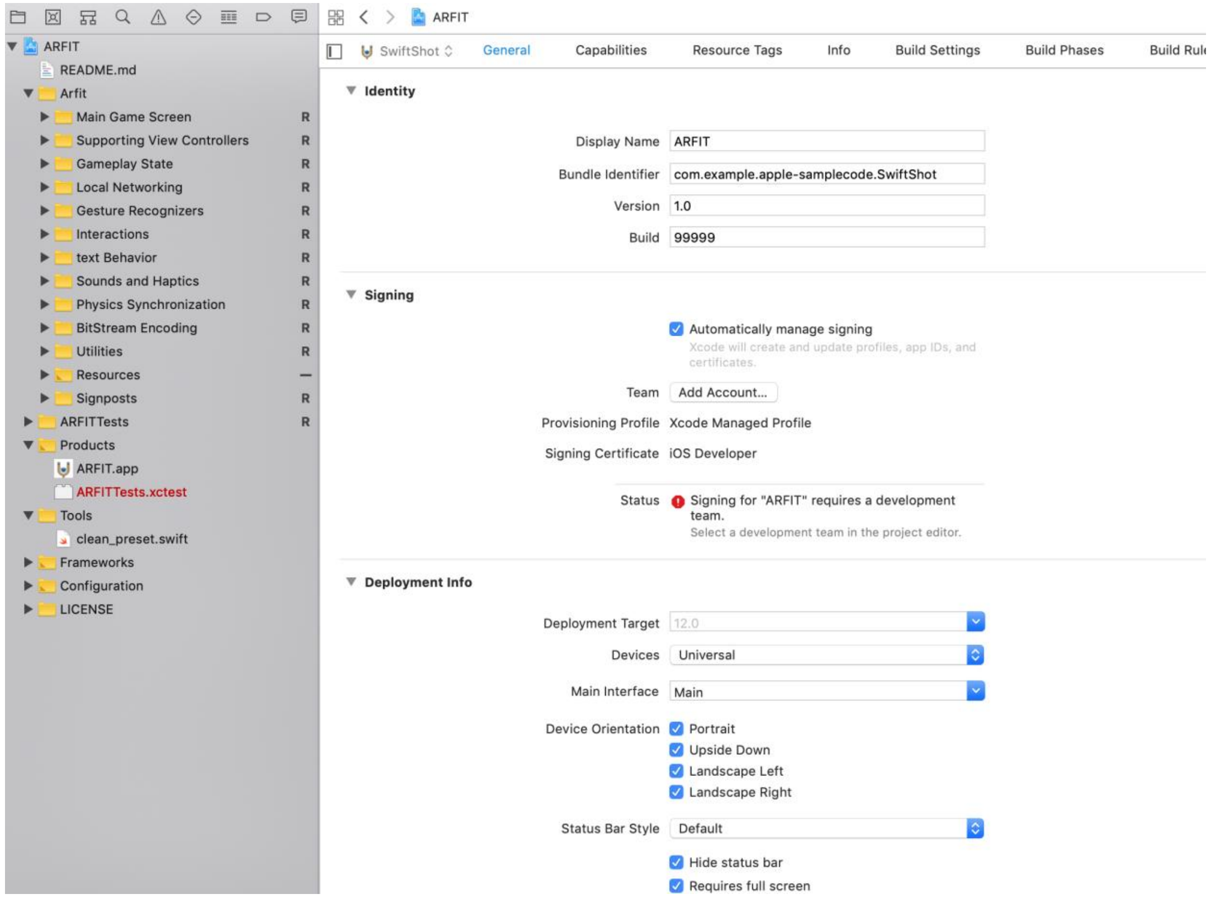
Which means we can incorporate this into web pages.

```
<a rel="ar" href="model.usdz">
<picture>
<source srcset=“wide-image.png”
media="(min-width: 600px)"> <img src="narrow-image.png">
</picture>
</a>
```

Anchor the real world

```
/*

Abstract:
Custom anchor for saving camera screenshots in an ARWorldMap.
*/

import ARKit

class KeyPositionAnchor: ARAnchor {
    let image: UIImage
    let mappingStatus: ARFrame.WorldMappingStatus

    init(image: UIImage, transform: float4x4, mappingStatus: ARFrame.WorldMappingStatus) {
        self.image = image
        self.mappingStatus = mappingStatus
        super.init(name: "KeyPosition", transform: transform)
    }

    override class var supportsSecureCoding: Bool {
        return true
    }

    required init?(coder aDecoder: NSCoder) {
        if let image = aDecoder.decodeObject(of: UIImage.self, forKey: "image") {
            self.image = image
            let mappingValue = aDecoder.decodeInteger(forKey: "mappingStatus")
            self.mappingStatus = ARFrame.WorldMappingStatus(rawValue: mappingValue) ?? .notAvailable
        } else {
            return nil
        }
        super.init(coder: aDecoder)
    }

    // this is guaranteed to be called with something of the same class
    required init(anchor: ARAnchor) {
        let other = anchor as! KeyPositionAnchor
        self.image = other.image
        self.mappingStatus = other.mappingStatus
        super.init(anchor: other)
    }

    override func encode(with aCoder: NSCoder) {
        super.encode(with: aCoder)
        aCoder.encode(image, forKey: "image")
        aCoder.encode(mappingStatus.rawValue, forKey: "mappingStatus")
    }
}
```

Xcode project



Identity

Display Name: ARFIT
Bundle Identifier: com.example.apple-samplecode.SwiftShot
Version: 1.0
Build: 99999

Signing

☑ Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.

Team: Add Account...
Provisioning Profile: Xcode Managed Profile
Signing Certificate: iOS Developer
Status: Signing for "ARFIT" requires a development team.
Select a development team in the project editor.

Deployment Info

Deployment Target: 12.0
Devices: Universal
Main Interface: Main
Device Orientation: ☑ Portrait
☑ Upside Down
☑ Landscape Left
☑ Landscape Right
Status Bar Style: Default
☑ Hide status bar
☑ Requires full screen

Conclusion and Future Directions

- I have briefly touched on applying AR to IT using SDK's from Apple and Google.
  - next I will have to purchase equipment (you cannot really simulate this)
    - a pair of North glasses (possible) or a newer tablet
- I would like to produce a rich environment (using USDZ) for monitoring, LOM, maintenance and repair.
  - imagine (using UW portal maps) entering a building and then using a machine room anchor know the health within and drill down on critical zones and if necessary, remotely repair them (software) or follow guides (physically).

Future Direction

- VICAAR (Visuality In Connected Algorithmic Augmented Reality)
  - obtaining code from the natural environment
  - augment your reality with the algorithms describing what you are sensing.

BIBLIOGRAPHY

- https://developers.google.com/ar/ ARCore
- https://developer.apple.com/arkit/ ARKit 2
- https://developer.apple.com/videos/play/wwdc2018/603/
- https://graphics.pixar.com/usd/docs/
- Augmented reality: Principles and Practice, Addison Wesley, 2016

Thank you

WatITis