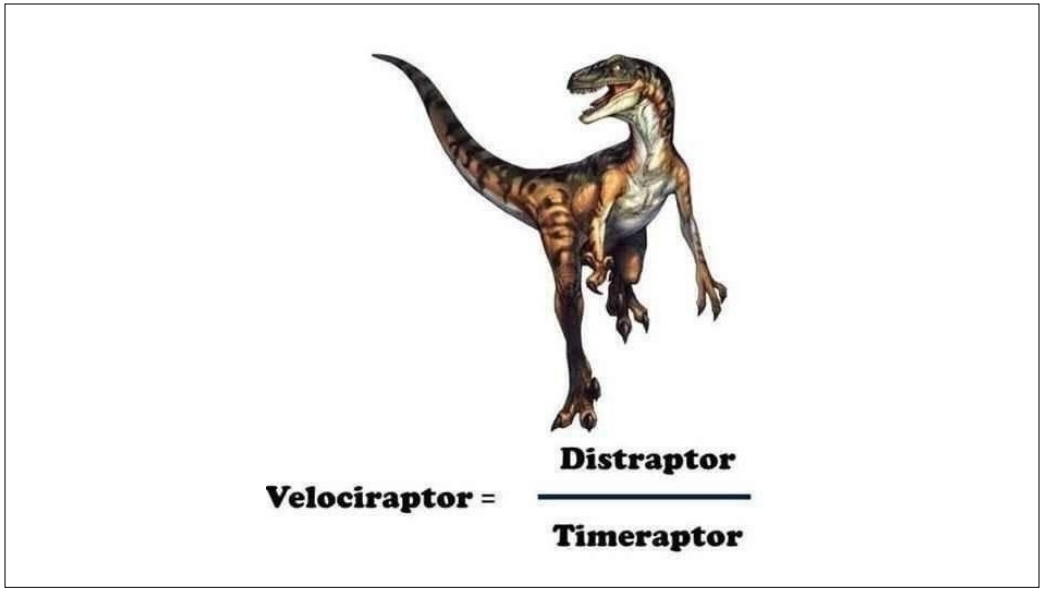

TECH 1711 - Mixed Reality Studio

Before we get started…

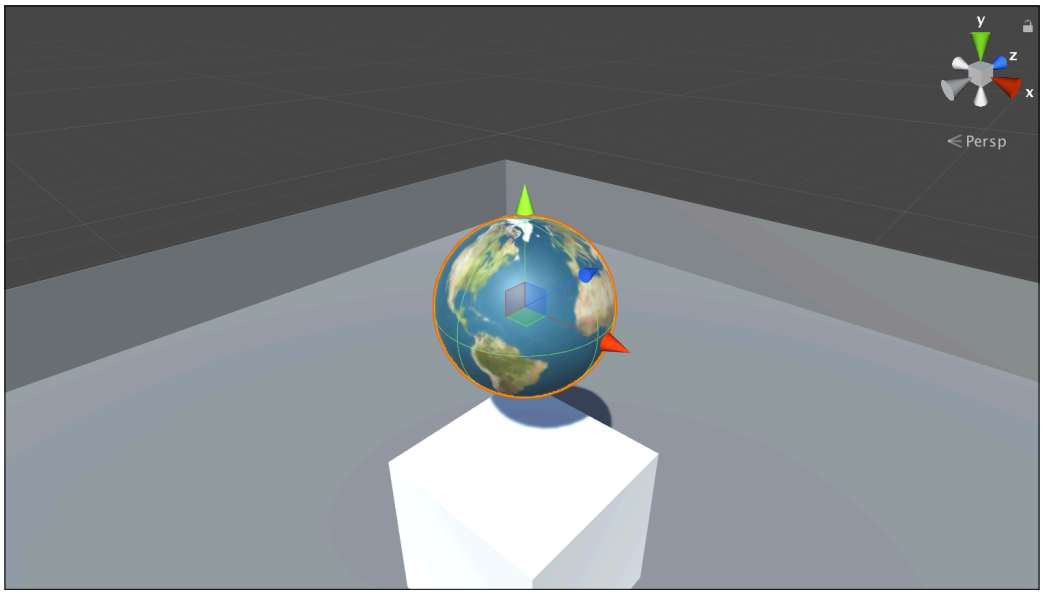
Last week we talked a bit about Velocity.

Velocity = Distance ÷ Time

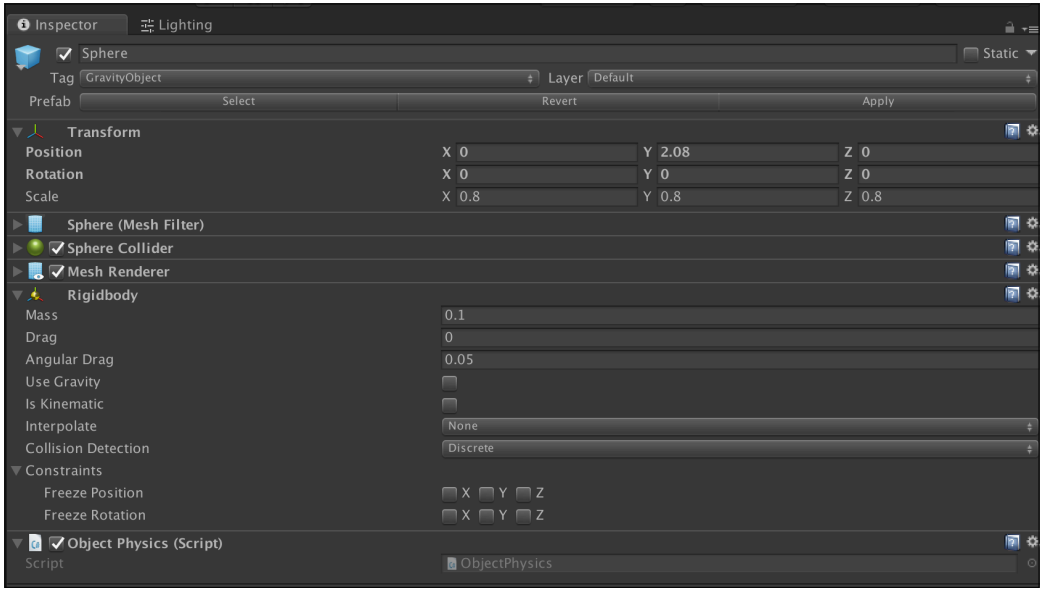


I can’t believe I missed the perfect chance last week to make this joke 🤦

Quick topic recap…

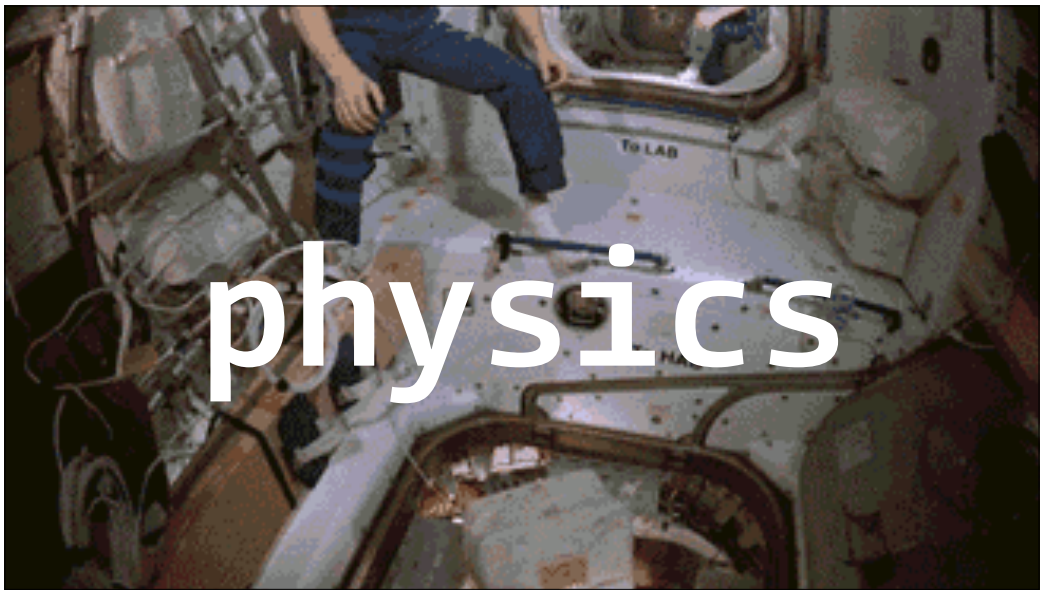


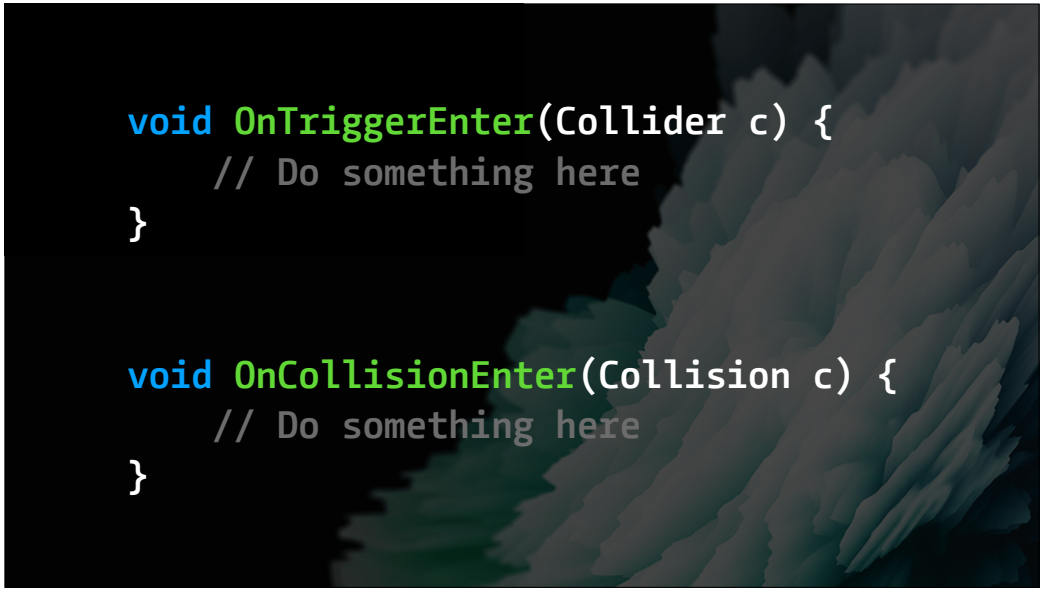
Organization:
Game Objects



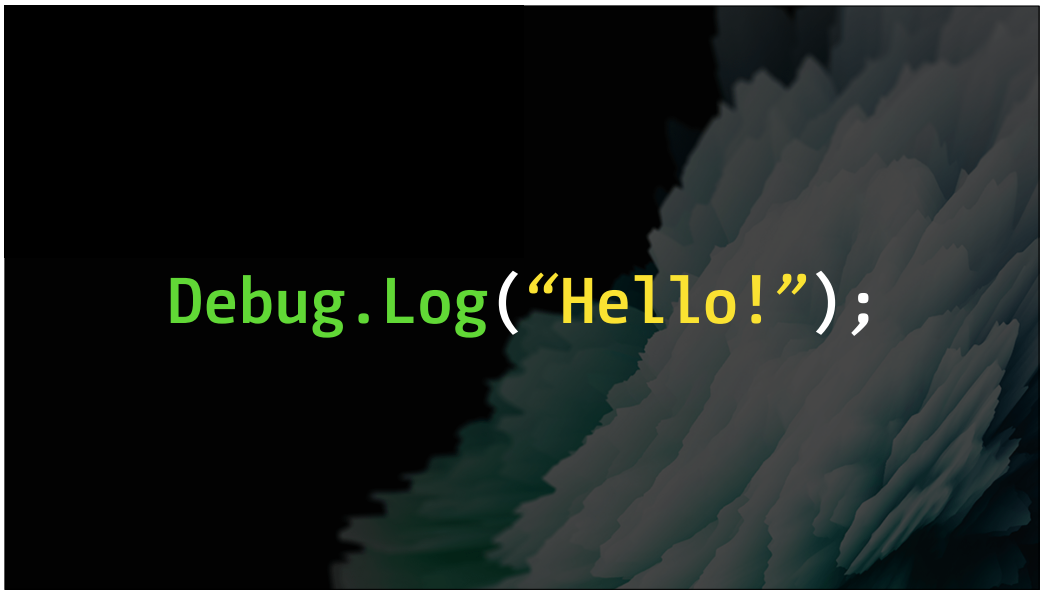
Organization:

Game Object **Components**


physics



Collisions (hard contact) vs Triggers (passes through)



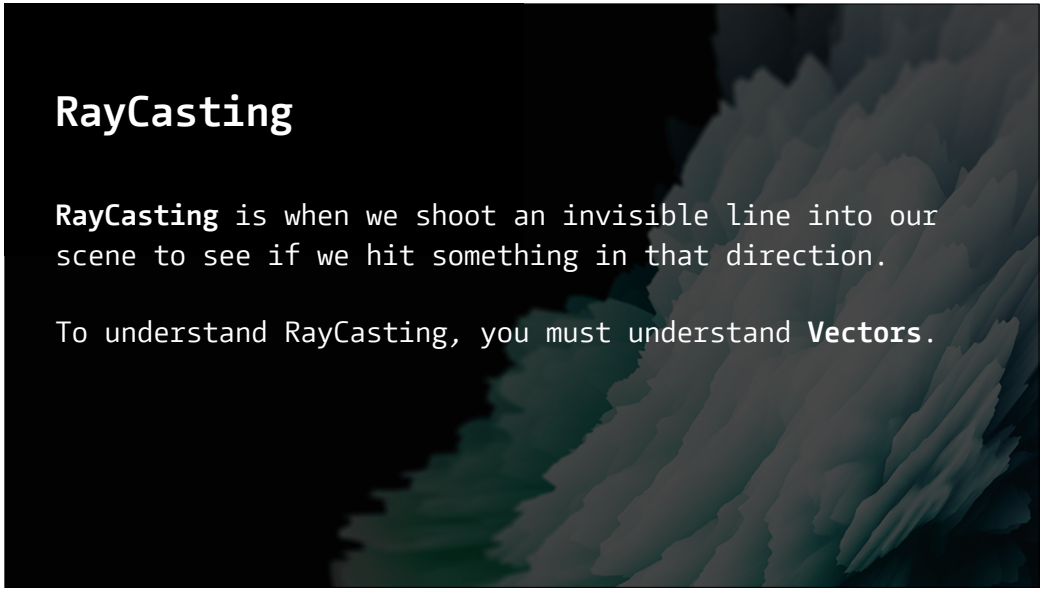
If you learned no other coding from this class, remember that Debug.Log is your friend!

It's not always your fault that something isn't working.

Vectors

In programming terms, you can think of Vectors as a way to store 2, 3, or 4 values in one easy-to-use package:

```
Vector2 someNumbers = new Vector2(1.0, 2.2);
Vector3 someOtherNumbers = new Vector3(5.3, 2.6, 12.0);
Vector4 evenMoreNumbers = new Vector4(7.4, 2.1, 12.0, 9.8);
```


RayCasting
RayCasting is when we shoot an invisible line into our scene to see if we hit something in that direction.
To understand RayCasting, you must understand **Vectors**.

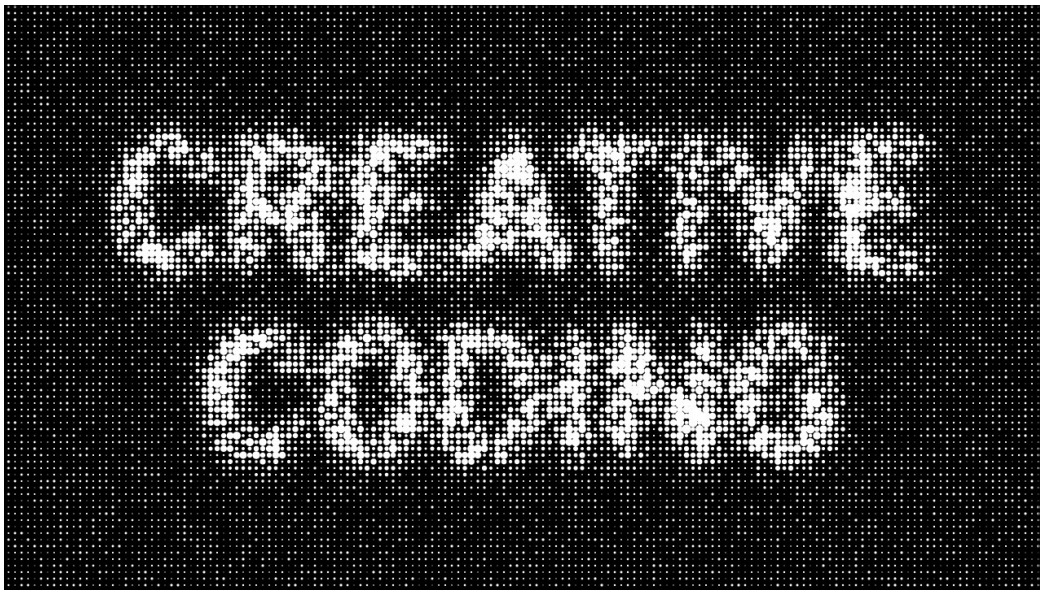
http://docs.unity3d.com

Don’t expect yourselves to be Unity experts right away.

Point is: you know enough to know **what** you can do, and hopefully where to look to get to **how** something is done.

I look stuff up almost every time I use Unity.

Concept Recap..



What is creative coding?


Screens
vs
Space

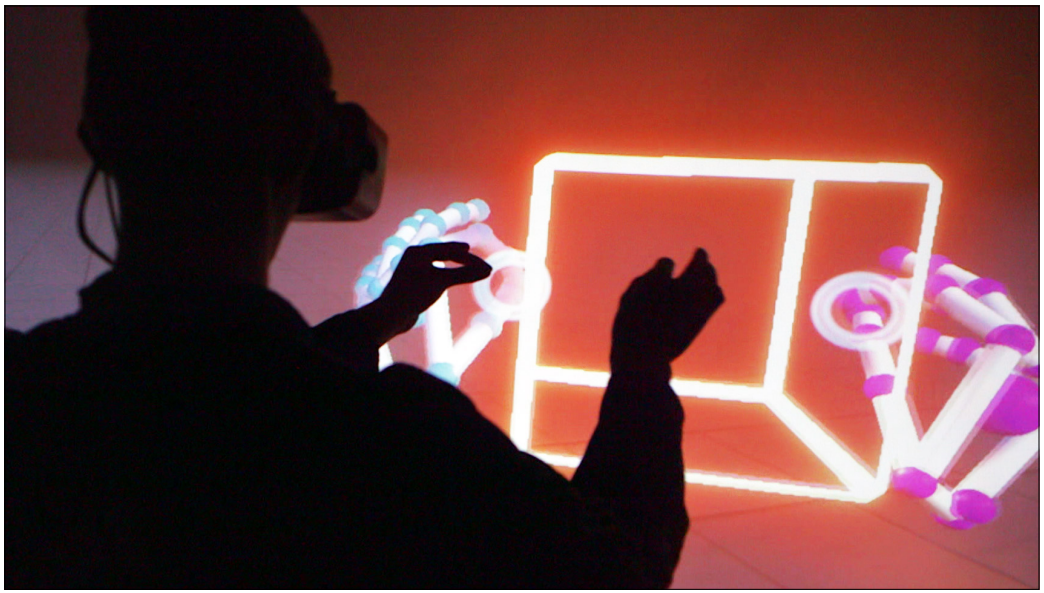
What do we mean by *interaction*?

- How do you act **upon** the HARDWARE of the computer/device?
- What does that **action** DO?

Two main types we focused on: **Manipulation** and **Exploration**

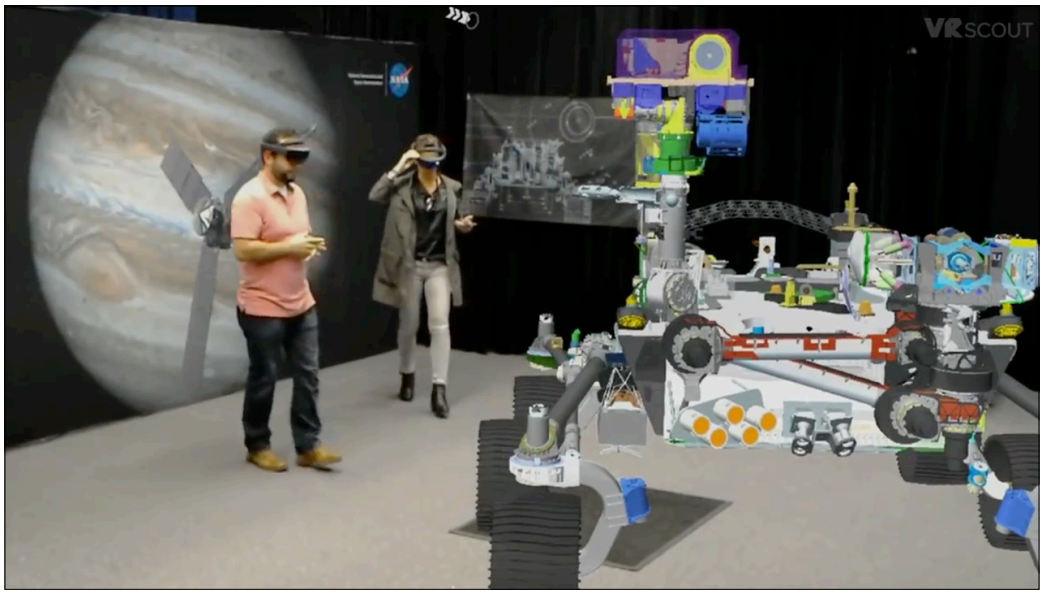
What is the interface?

Attempts at 3D interfaces go way back.



Gestures can be used to manipulate

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

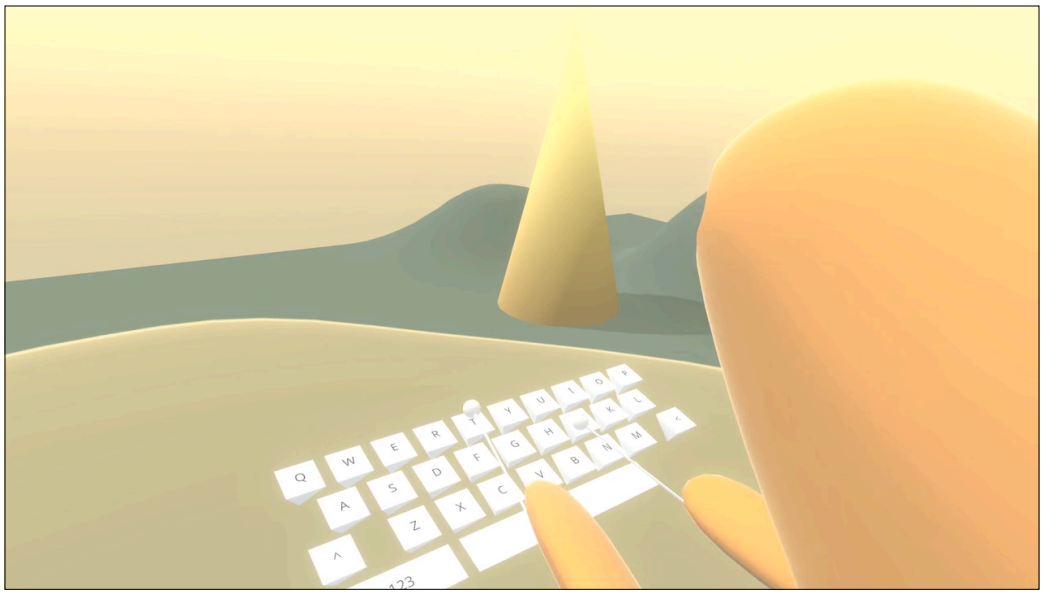

VR SCOUT

Designing for Interaction

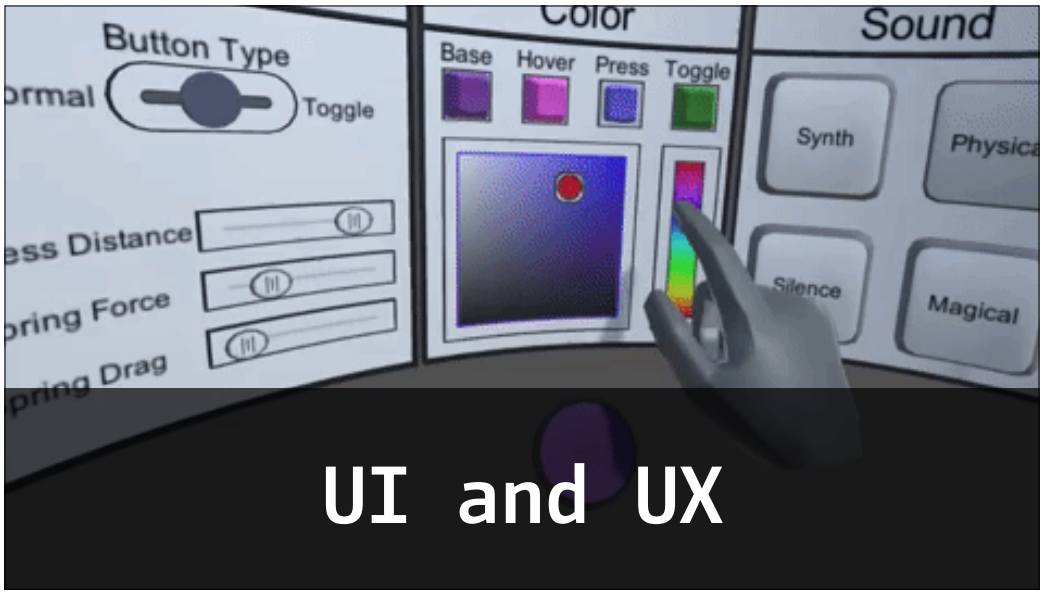
Why

To Reveal
To Change
To Visualize

Bad/frustrating example of bringing keyboard into space: using gaze direction + time.



Good example: uses space and natural interaction for the tools you have available (handheld remotes)



User Interfaces and User Experience

UX is the design of *how* the user interacts with your program and UI is the *visual manifestation* of the design

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.

Diegetic - Menu attached to an object you can manipulate in the scene - feels like it some thing **in** the scene.

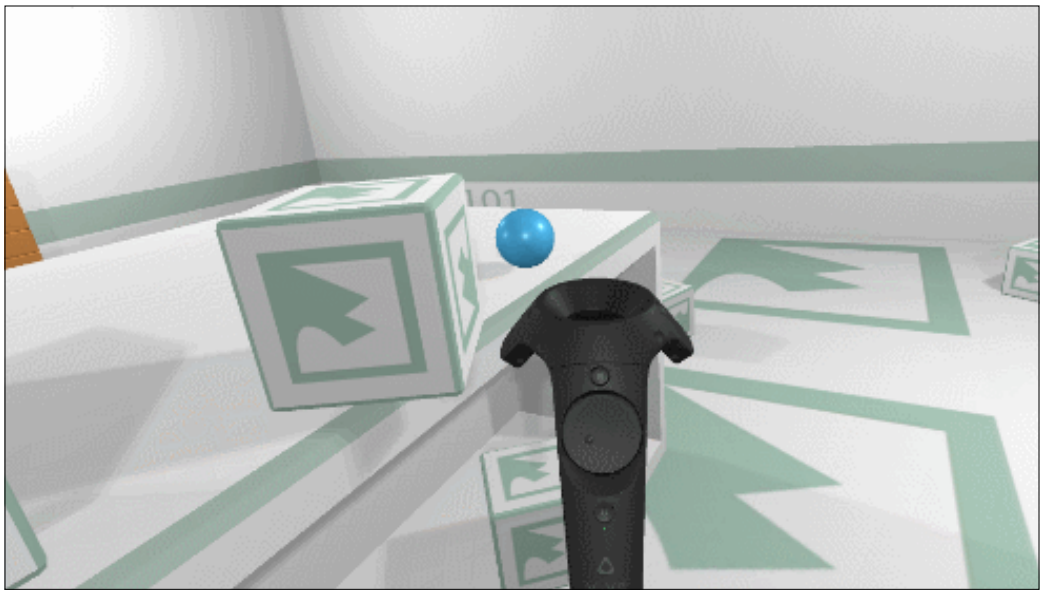
The most common symptoms are general discomfort, headache, stomach awareness, nausea, vomiting, pallor, sweating, fatigue, drowsiness, disorientation, and apathy.[2] Other symptoms include postural instability and retching.[2] Virtual reality sickness is different from motion sickness in that it can be caused by the visually-induced perception of self-motion; real self-motion is not needed.[1] It is also different from simulator sickness; non-virtual reality simulator sickness tends to be characterized by oculomotor disturbances, whereas virtual reality sickness tends to be characterized by disorientation.[3]

Another trigger of virtual reality sickness is when there is disparity in apparent motion between the visual and vestibular stimuli. Essentially what happens is there is a disagreement between what the stimuli from the eyes send to the brain and what the stimuli from the inner ear are sending to the brain. This is what is essentially at the heart of both simulator and motion sickness. In virtual reality, the eyes transmit that the person is running and jumping through a dimension, however, the ears transmit that no movement is occurring and that the body is sitting still. Since there is this discord between the eyes and the ears, a form of motion sickness can occur.

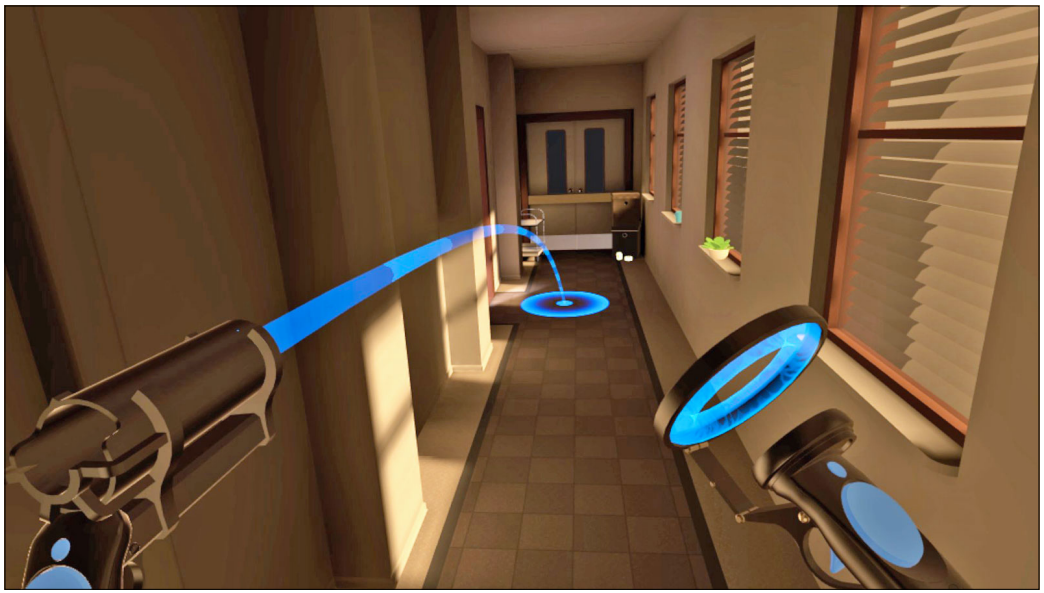
"getting your sea legs”.

individual susceptibility
https://en.wikipedia.org/wiki/Virtual_reality_sickness?oldformat=true

Like a poison



Teleporting allows us to virtually move without *physically moving.*

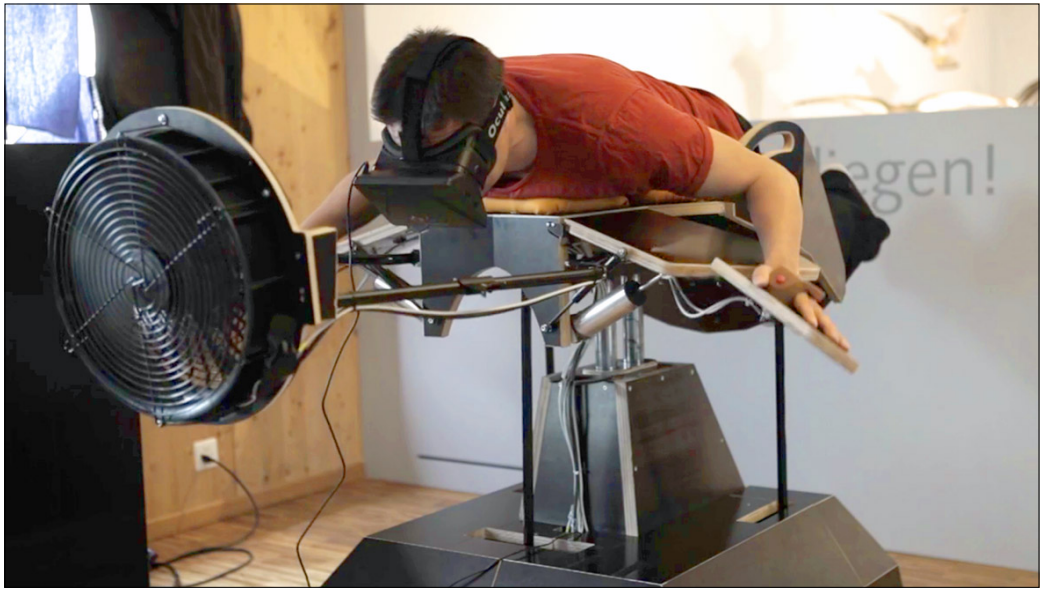


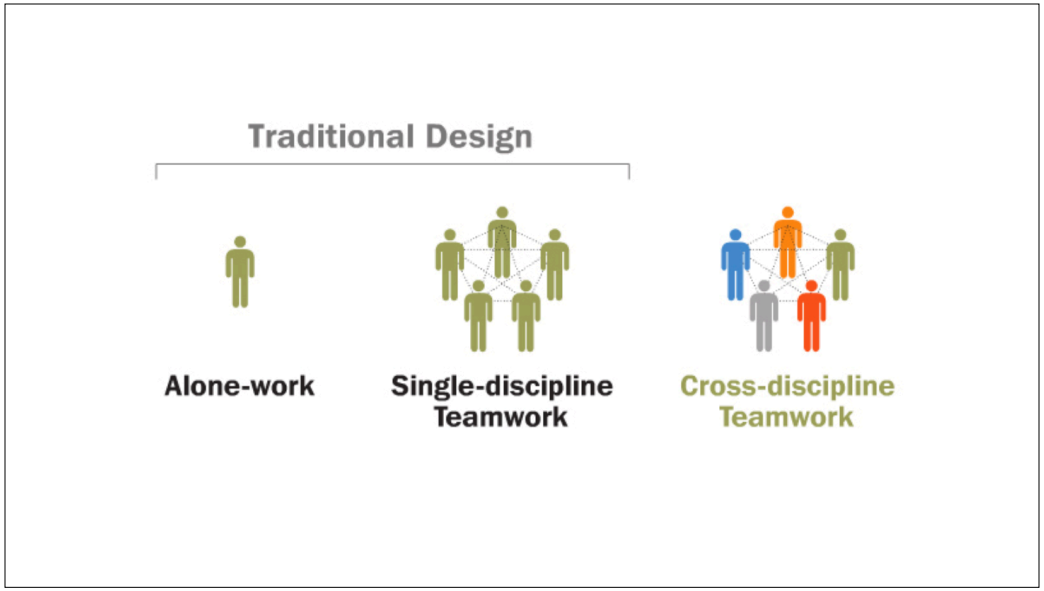
Modality

n. - A particular mode in which something exists or is experienced or expressed.

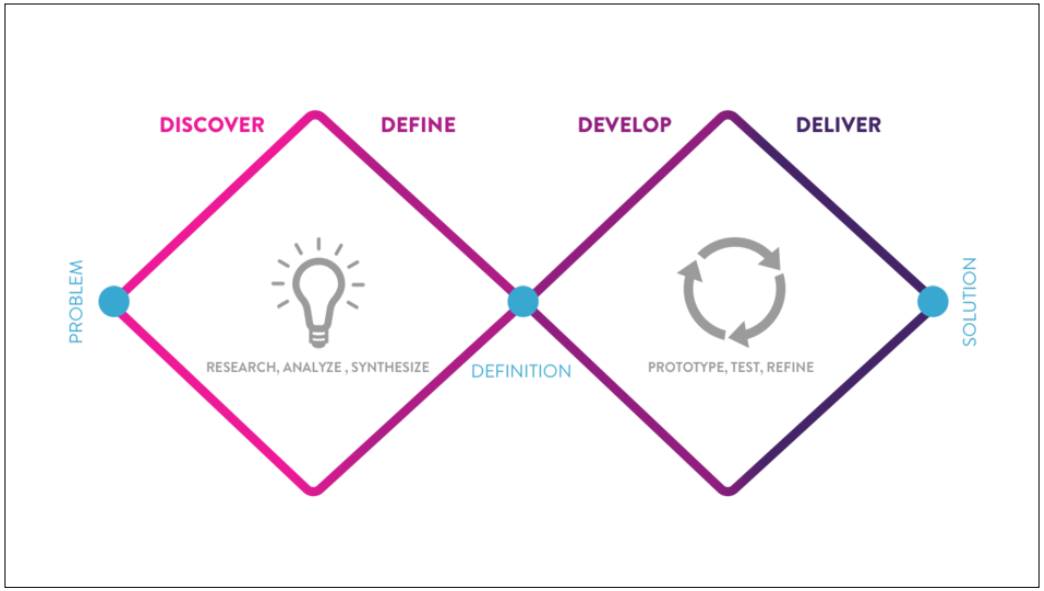
Basically, the ***way*** that you do something.

Not just what you are doing in the experience! Think about **how** the user uses it.


Oculus
egen!



Design Process

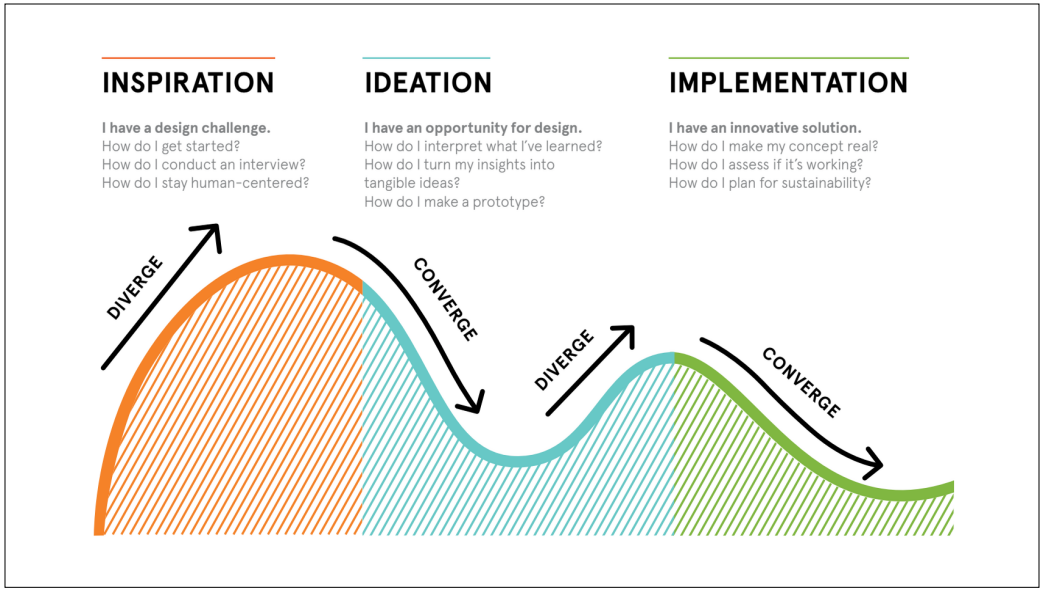


Double Diamond is the name of a design process model developed by the British Design Council in 2005

Alessi, BSkyB, BT, LEGO, Microsoft, Sony, Starbucks, Virgin Atlantic Airways, Whirlpool, Xerox, Yahoo!

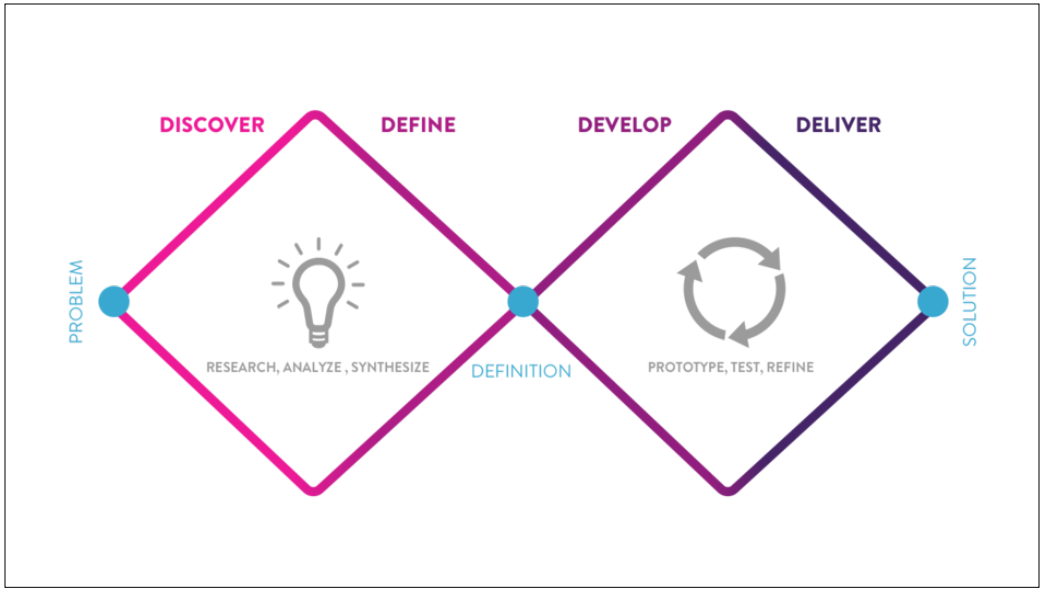
Different designers manage the process of design in different ways. But when we studied the design process in eleven leading companies, we found striking similarities and shared approaches among the designers we talked to.

Very useful to structure development of ideas when otherwise so open ended



Ideo Human Centered Design

Phases of this process are either diverging or converging. During a diverging phase, you try to open up as much as possible without limiting yourself, whereas a converging phase focuses on condensing and narrowing your findings or ideas.



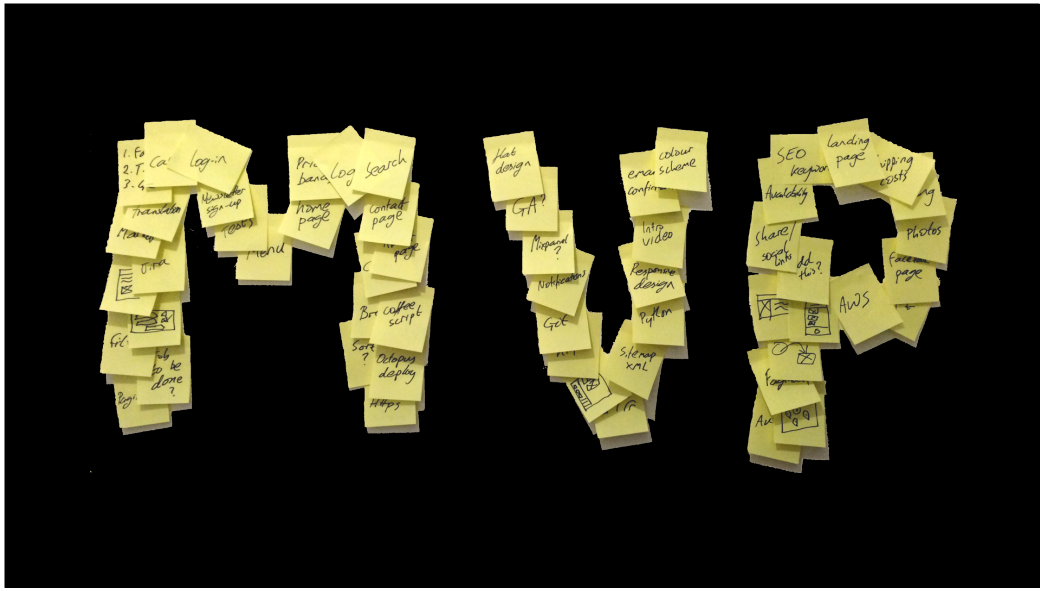
Not equal time….equal weight

Discover – The first quarter of the Double Diamond model covers the **start of the project**. Designers try to **look at the world in a fresh way, notice new things and gather insights.**

Define – The second quarter represents the definition stage, in which designers try to **make sense of all the possibilities identified in the Discover phase**. Which matters most? **Which should we act on first? What is feasible?** The goal here is to develop a clear creative brief that frames the fundamental design challenge.

Develop – The third quarter marks a period of development where s**olutions or concepts are created, prototyped, tested and iterated**. This process of trial and error helps designers to improve and refine their ideas.

Delivery – The final quarter of the double diamond model is the delivery stage, where the r**esulting project** (a product, service or environment, for example) is **finalized, produced and launched.**



Minimum Viable Product

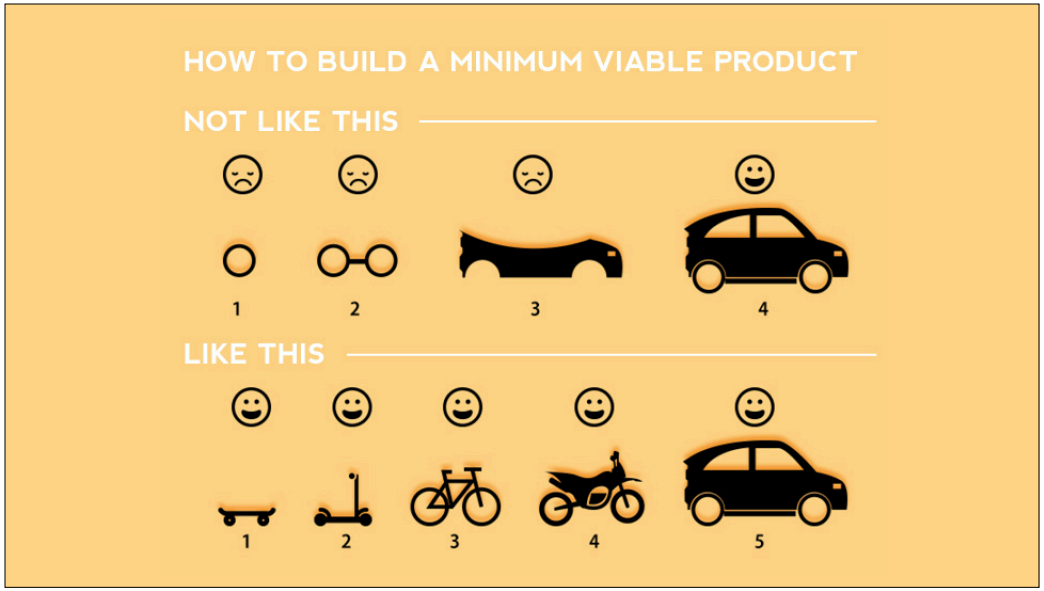
The Lean Startup - by Eric Ries

Maximum amount of validated learning about customers with least effort.

(Grown to mean a lot more than it was originally)

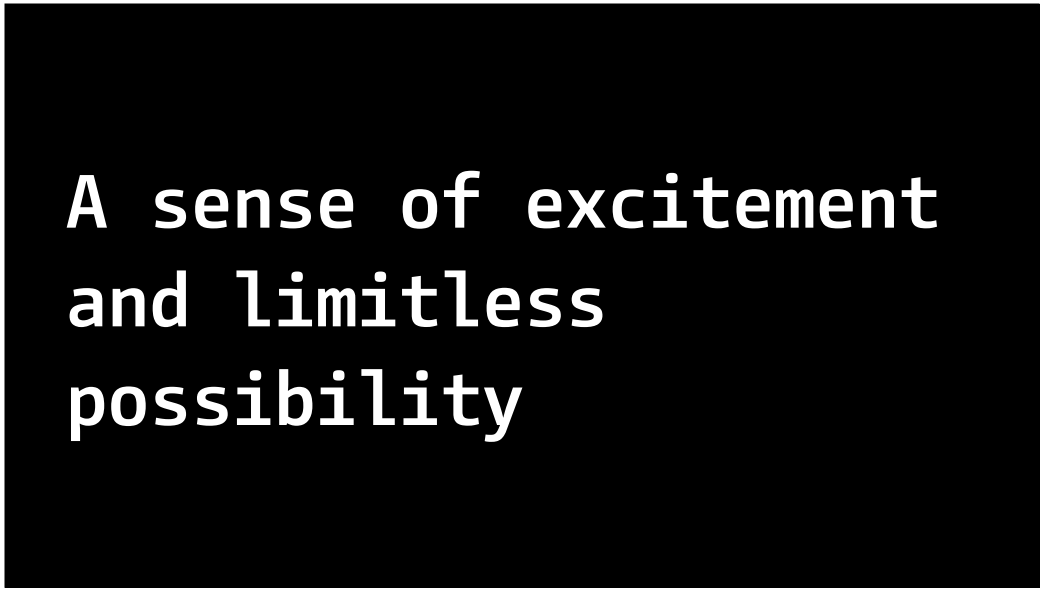
Riskiest Assumption Test - There is no need to build more than what's required to test your largest unknown.

No expectation of perfect code or design. No danger it will prematurely become a product.



Very quickly get to something usable that gets you “from point A to point B”

Implementation horrors



A sense of excitement and limitless possibility

An idea or discovery arrives shiny and new, without the eventual scuff marks inherent in the process of bringing it to fruition. The sense of potential is empowering, yet rarely realistic. It’s like buying a Ferrari without yet having found a mechanic to service it. Enjoy this part as it happens, but not so much that you are afraid to let it go.

As soon as you dig into a [project], you'll uncover a seemingly endless pile of emotional and logistical factors simultaneously at play. This is quicksand. Too much thinking here means trouble. When you are in this territory, focus on doing. Get right to prototyping through quick mock-ups and experiences. Acknowledge and categorise new issues as they arise, but prioritise – you'll never be able to resolve all of them. Keep your eyes open for inspiration and direction away from the soup of complexity.

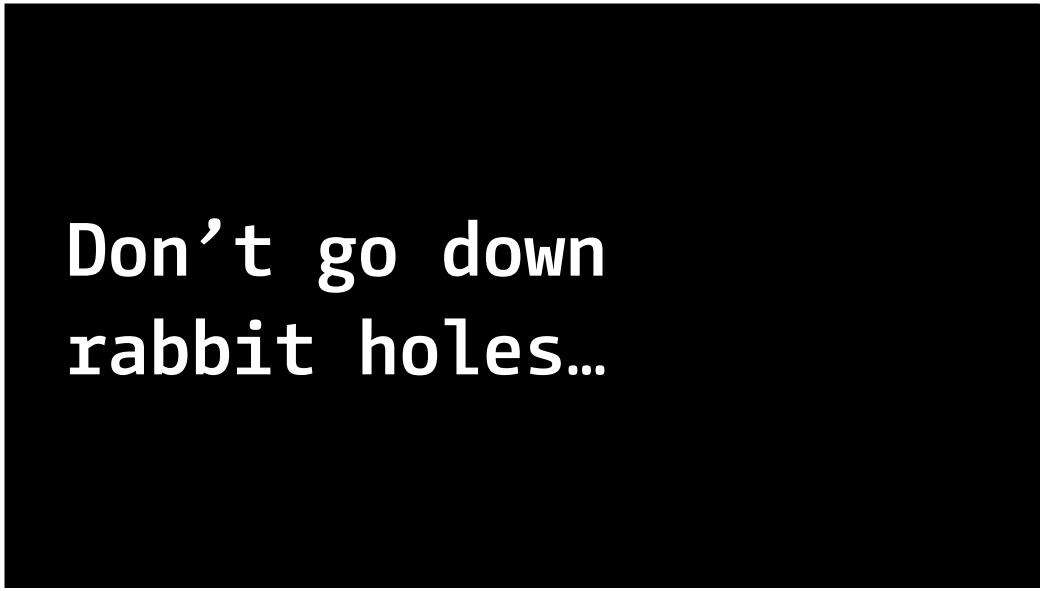
These are moments of clarity when you feel you’ve got it all figured out. They are the siren songs: glorious but potentially derailing. Strive with all your might to get to this point, but diligently question this clarity when you arrive.

The brutal realities of implementation

Almost everything you do will take longer than you think because there is a lot more to consider than is apparent. You will have to make compromises. Have the strength to stand up for the right things and the wisdom to let go of the meaningless bits. The ability to know how and when to do this often comes from the proof and intuition you'll acquire through prototyping.

Use What You Know

- Break problem into smaller pieces
- ‘Hack-y’ solutions are OK!



A beautiful 3D Model is great, but not if your project doesn’t do anything.

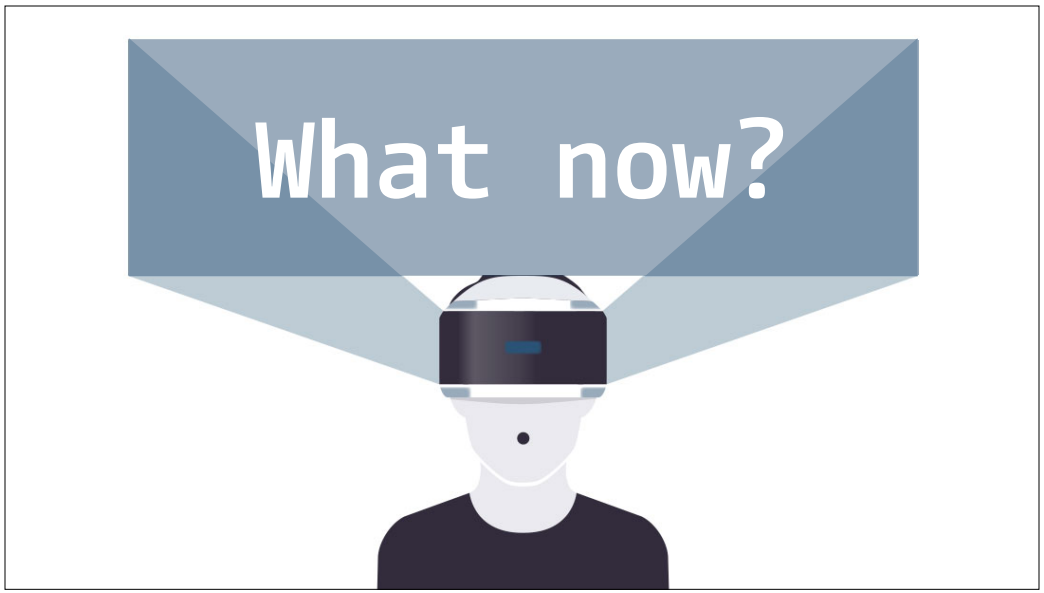
Don’t forget the boring stuff

- Organizing your program helps you make changes
- Keep track of the **state** of your program
  - For example:

```
bool isReady = false;
```

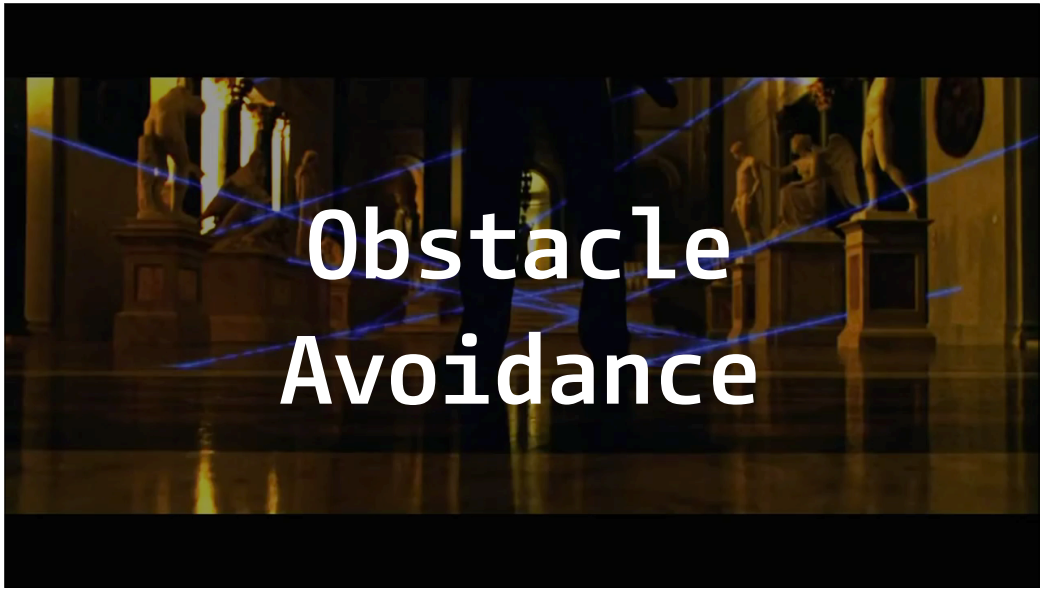
- Use if/else statements to check the state and change behavior accordingly:

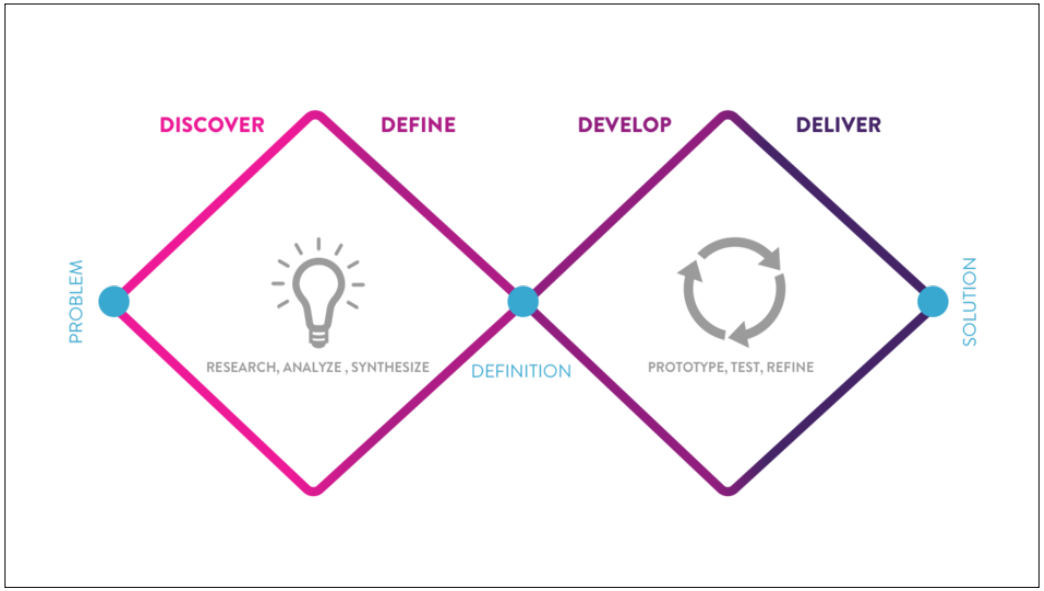
```
if (isReady == true) {
    // Do something here
} else {
    // Do something different here
}
```

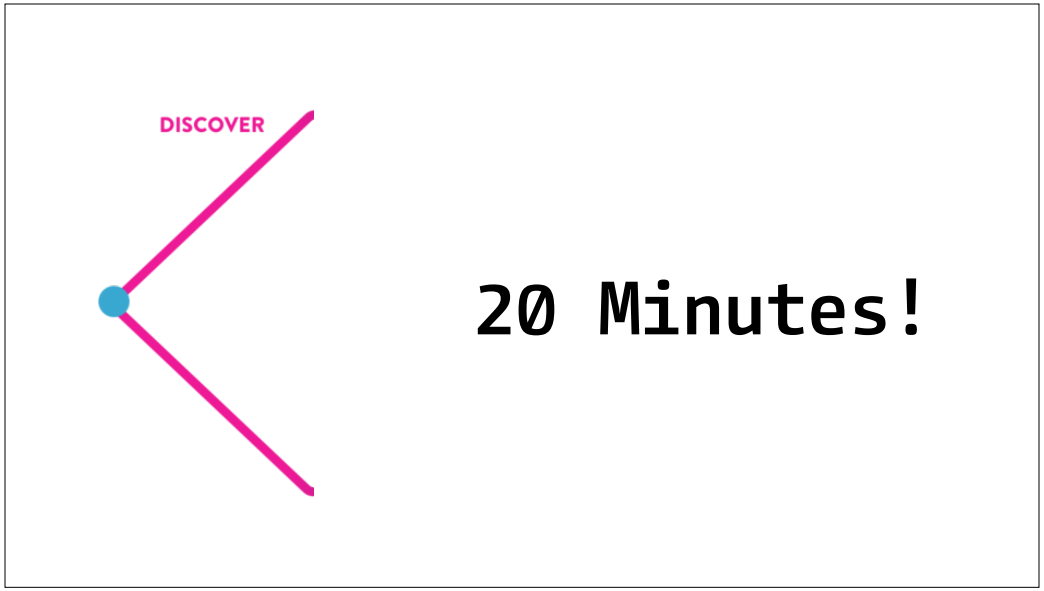


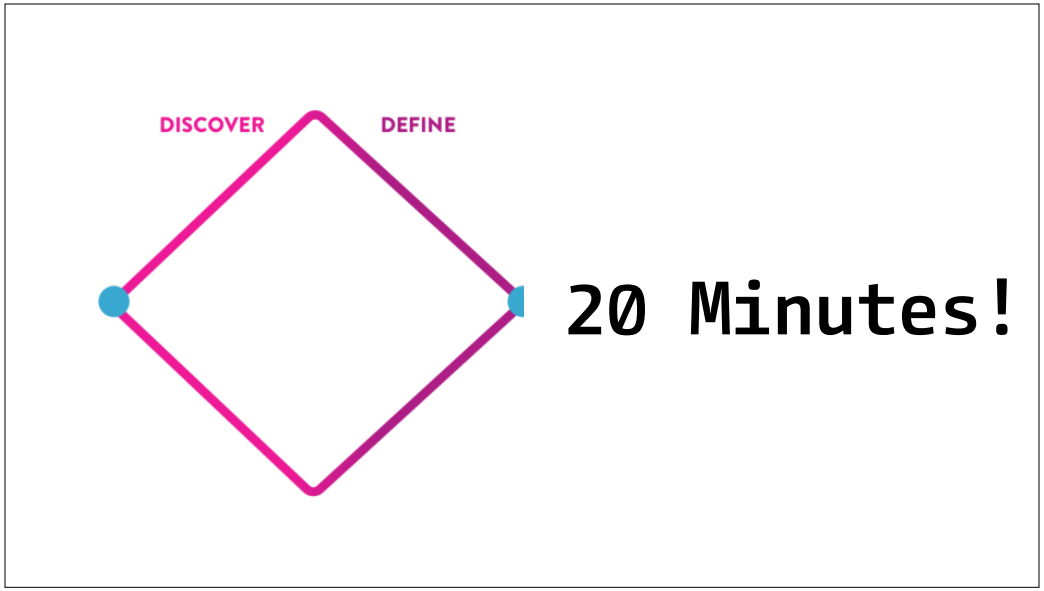
Workshop project.

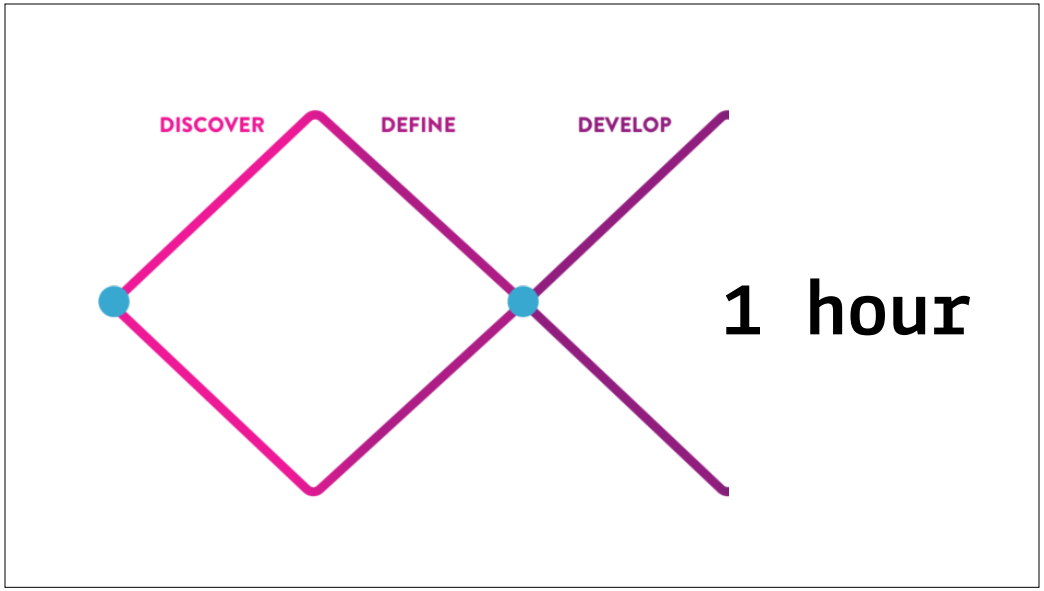

Escape Room

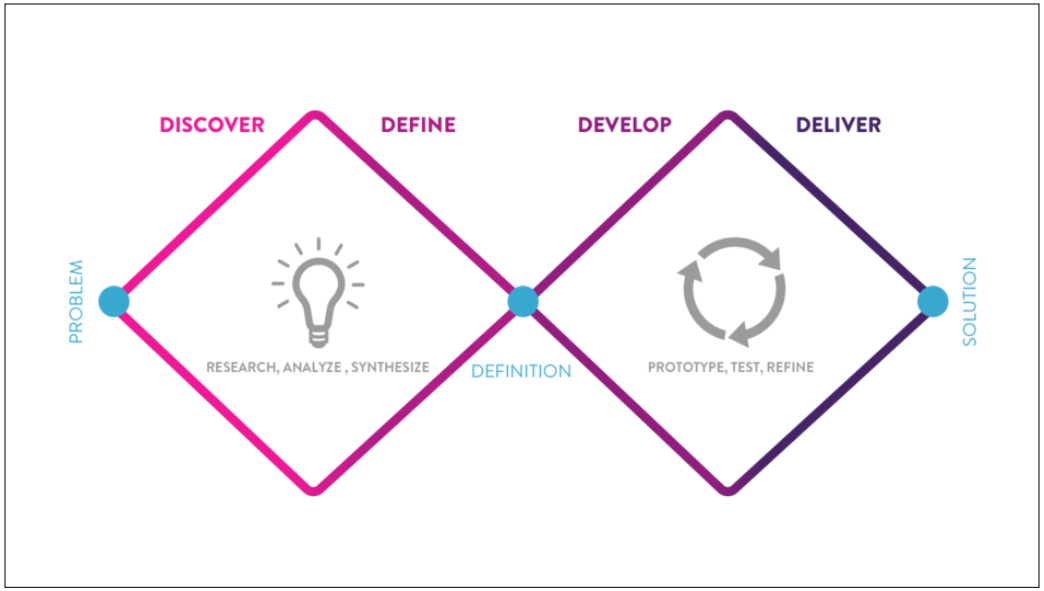

Obstacle
Avoidance


DISCOVER
DEFINE
DEVELOP
DELIVER
PROBLEM
RESEARCH, ANALYZE , SYNTHESIZE
DEFINITION
PROTOTYPE, TEST, REFINE
SOLUTION


DISCOVER
20 Minutes!


DISCOVER
DEFINE
20 Minutes!


DISCOVER
DEFINE
DEVELOP
1 hour


DISCOVER
DEFINE
DEVELOP
DELIVER
PROBLEM
RESEARCH, ANALYZE , SYNTHESIZE
DEFINITION
PROTOTYPE, TEST, REFINE
SOLUTION


TECH 1711 - Mixed Reality Studio