TECH 1711 - Mixed Reality Studio

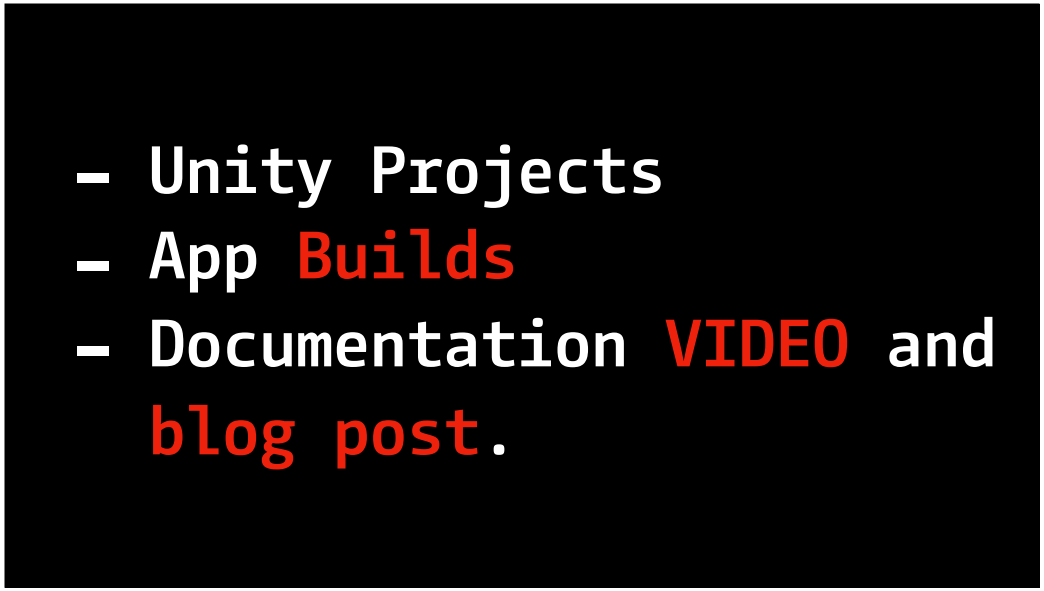
Final Project!

Here we go!

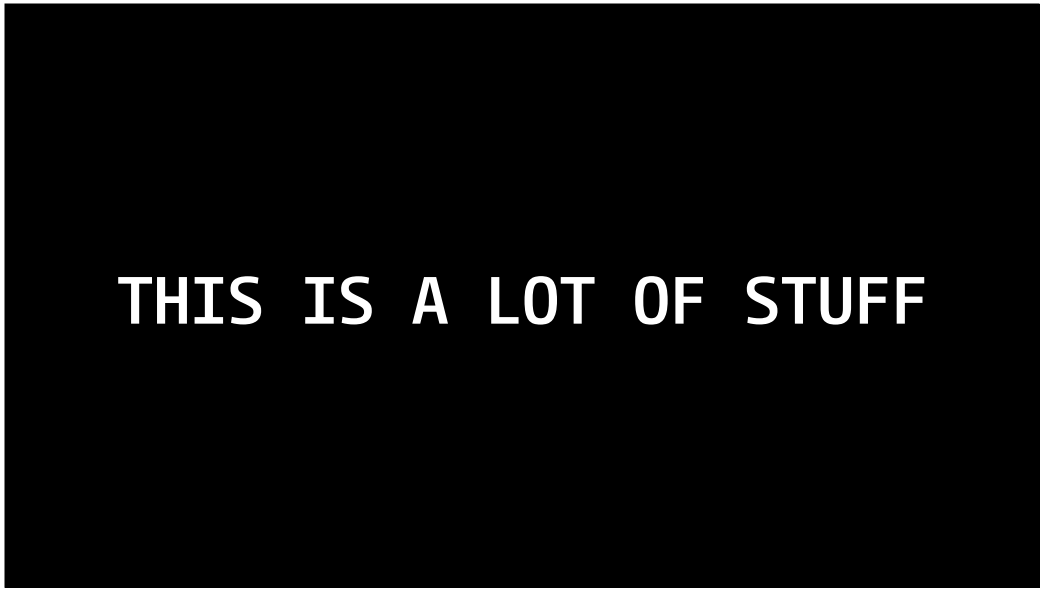
6 classes left! (Plus a saturday session)

Due 4/25?

- Buuuuuut…..we'll have a showcase in the first week of May.
- The project I ***grade*** will be the one you turn in at the showcase
- However, your grade will be *severely* penalized if you don't have some version of your working project by 4/25 (one whole letter-grade down)

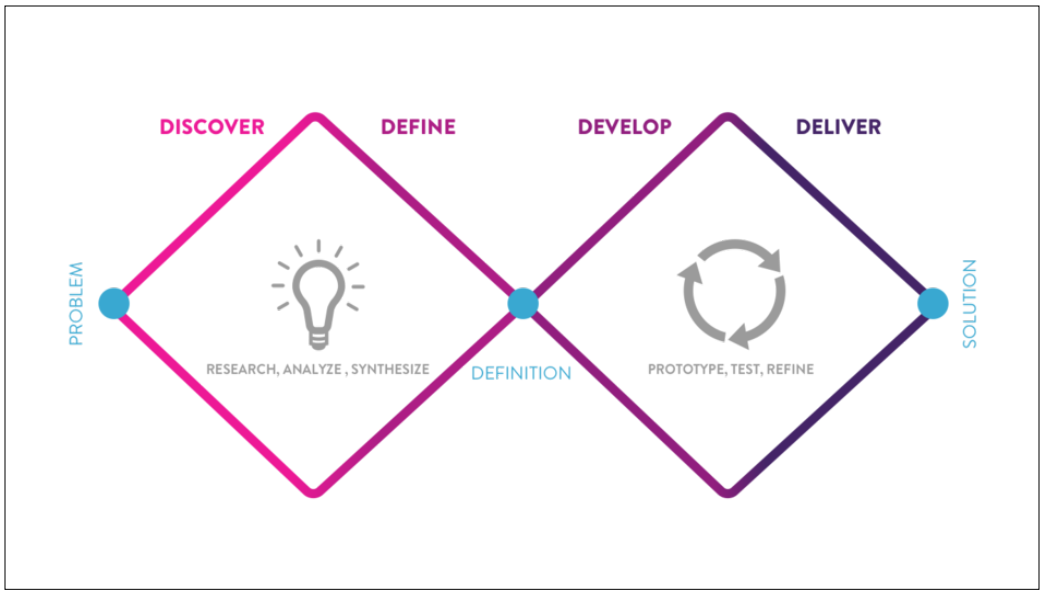


- Final deliverables due by public start of showcase.
- Must Blog post doesn't have to be public before showcase so that you can edit it and incorporate pictures of people using your project.
    - I don't care what format/service you use for post (I suggest medium.com or even just a github repo with a single text file)



- Don’t underestimate the work that goes into documentation
- Documentation journals can become first draft of blog post

- Don’t try to do too much. Focus on designing a simple but smooth user experience.
- People outside of this class should know what to do.



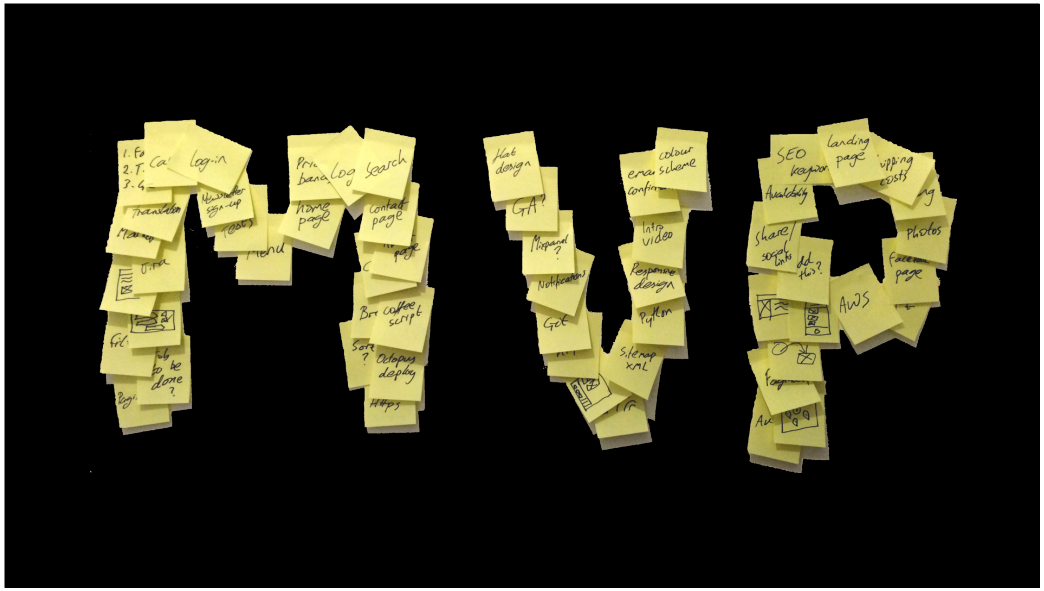
Not equal time….equal weight

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

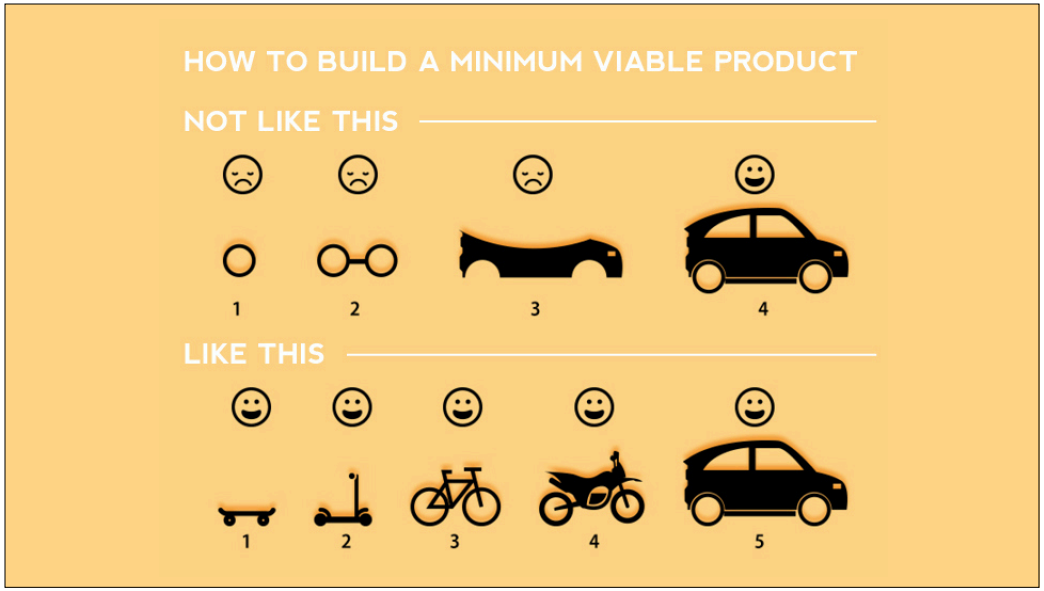
The Lean Startup - by Eric Ries

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.

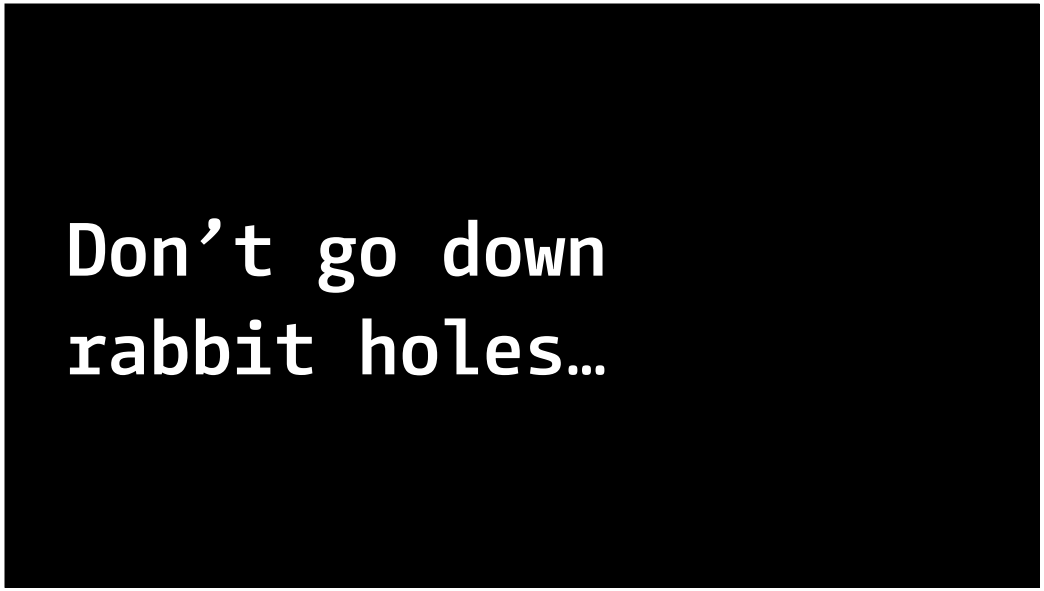
No expectation of perfect code or design. No danger it will prematurely become a product.



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

Use What You Know

- Break problem into smaller pieces
- ‘Hack-y’ solutions are OK!
- I *know* there is more out there than what I covered in this class. I picked what I covered very intentionally
    - For example: we built our teleport script in a very specific way. If you want to use a different one, think about *why* the one we made won’t work…



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

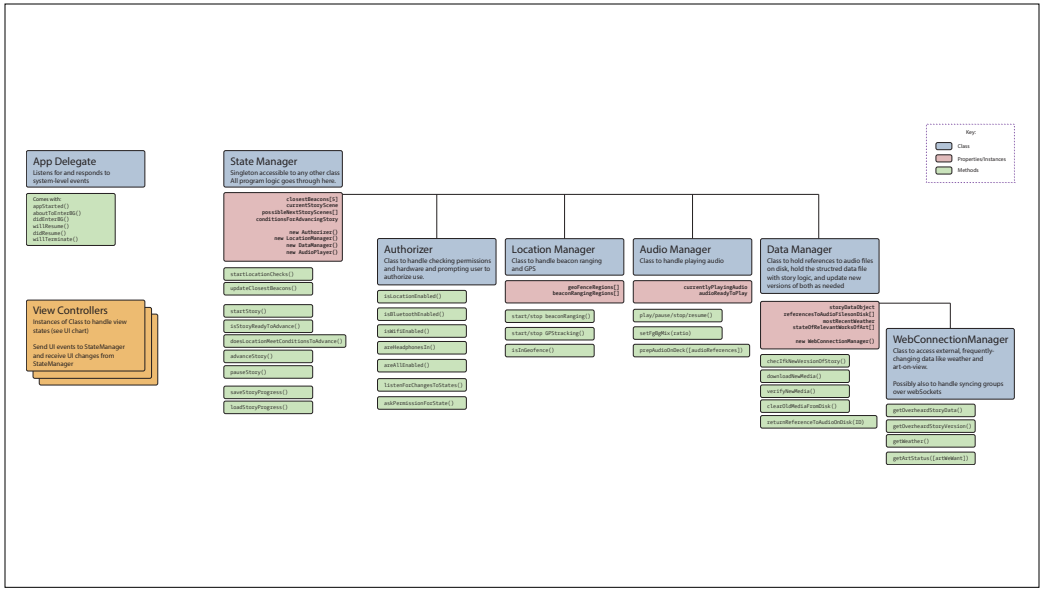
Don’t forget the boring stuff

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

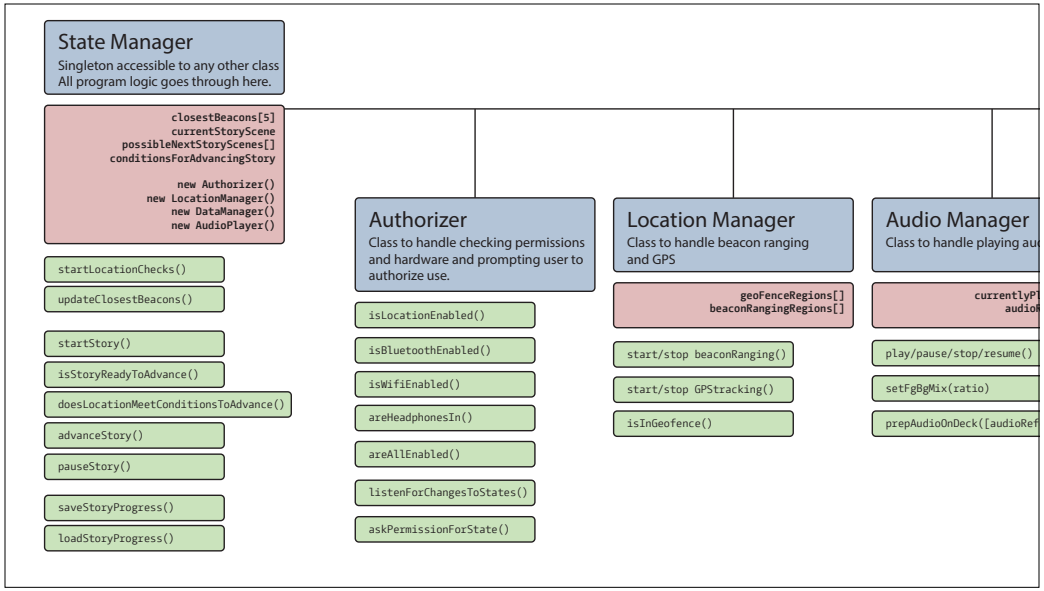
```
bool isReady = false;
```

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

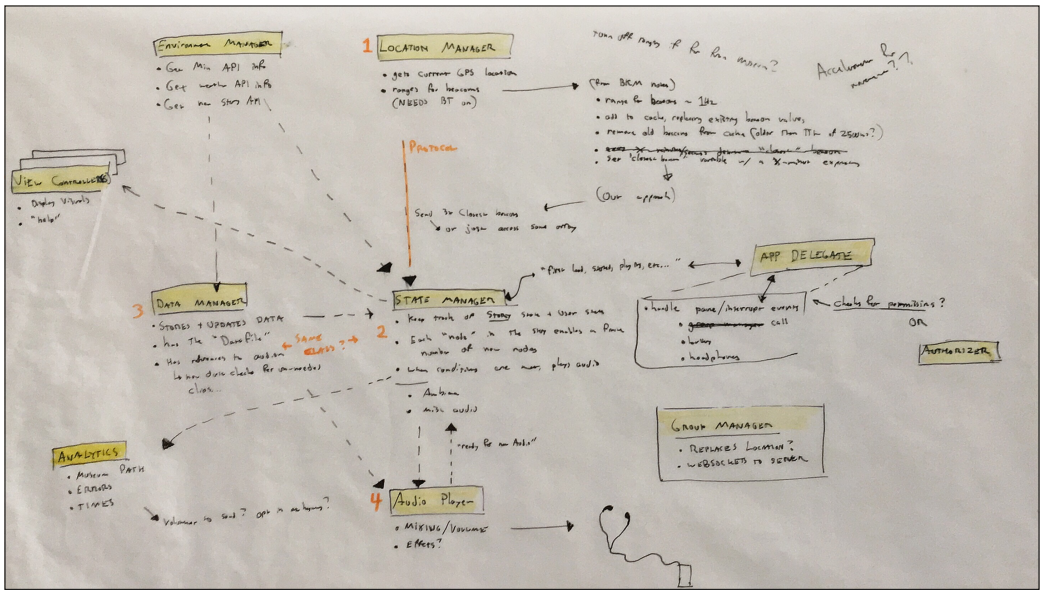
```
if (isReady == true) {
    // Do something here
} else {
    // Do something different here
}
```



- PLAN IT OUT
- Here is example of a way to organize an app with lots of components



Even for simple programs, helps to organize into well-defined pieces (classes) and the variables & functions that go along with each piece.



- Plan doesn’t have to be beautiful.
- Here is the same chart, before it was cleaned up and finalized

Final Project Proposals

Good luck!