



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)

- Unity Projects
- App Builds
- Documentation VIDEO and blog post.

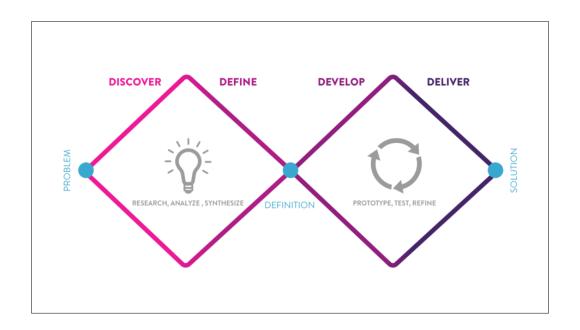
- Final deliverables due by public start of showcase.
- Must Blog post doesn't have to be public before showcase so that you can edit it and incorporate pictures of people using your project.
 - I don't care what format/service you use for post (I suggest <u>medium.com</u> or even just a github repo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

THIS IS A LOT OF STUFF

- 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

<u>Discover</u> – 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.

<u>Define</u> – 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.

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

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



Minimum Viable Product

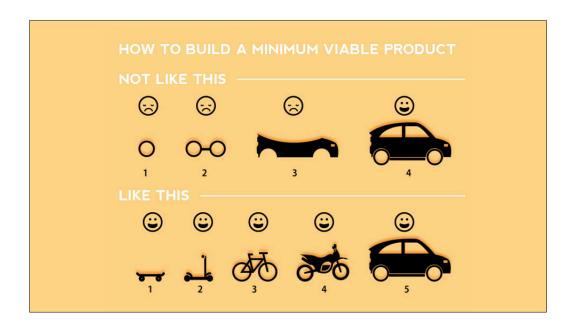
The Lean Startup - by Eric Ries

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...

Don't go down rabbit holes...

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

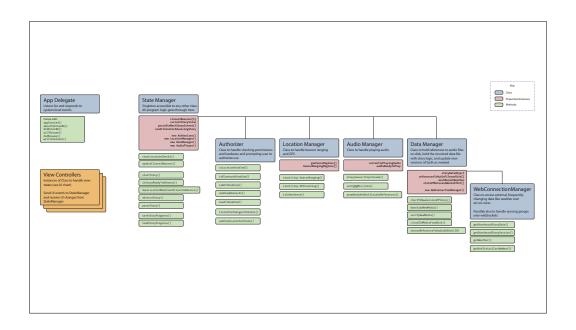
Don't forget the boring stuff

- 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:

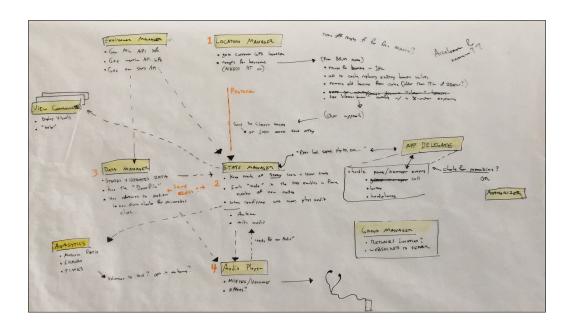
```
if (isReady == true) {
      // Do something here
} else {
      // Do something different here
}
```



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

State Manager Singleton accessible to any other class All program logic goes through here.			
closestBeacons[5]			
new LocationManager() new DataManager() new AudioPtayer() startLocationChecks()	Authorizer Class to handle checking permissions and hardware and prompting user to authorize use.	Location Manager Class to handle beacon ranging and GPS	Audio Manager Class to handle playing au
updateClosestBeacons()	isLocationEnabled()	geoFenceRegions[] beaconRangingRegions[]	currentlyP audio
startStory()	isBluetoothEnabled()	start/stop beaconRanging()	play/pause/stop/resume()
isStoryReadyToAdvance() doesLocationMeetConditionsToAdvance()	isWifiEnabled()	start/stop GPStracking()	setFgBgMix(ratio)
advanceStory()	areHeadphonesIn()	isInGeofence()	prepAudioOnDeck([audioRef
pauseStory()	areAllEnabled()		
saveStoryProgress()	listenForChangesToStates()		
loadStoryProgress()	askPermissionForState()		

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!