Term project:

use all skills that learned in lecture, be complex.

use experience: interactive, be able to test.

do not start by coding

avoid procrastination & use resources!

make your own smaller deadlines

trello

notion

google sheets

google calendar

reminders

resources:

OH/TP Mentor

use OOP

break project into different componenets (ps still not coding)

write features you want, separate them into classes.

pre-work recap:

1create a plan for yourself

2understand the main parts of your project

a.basic gameplay

b.complexity component

c.UI

timeline:

make the basic work

everything your project needs in order to be considered itself at the most basic level

everything needed in order to start on the most complex part of your project

anything needed for testing

TP Ideation TP1:Algorithmic Research

figure our what algorithm you needed to use

google your algorithm and find pages with either explanations or pseudocode

NO REAL CODE

work to understand those explanations and pseudocode as they are

google unfamiliar lines/methods

think about how this algorithm would work in the context in your project

what each line means in your context & what changes you might need to make

write pseudocode for your algorithm within your own code

MAKE SURE TO CITE RESOURCES

TP! - TP2 Algorithmic Complexity Coding

Now we tackle coding the hardest part of your project - the algorithmically difficult bit!

Don’t start his late

Get to MVP!

After you’re done with your algorithmically complex parts, all you need to do is fill in any last thins needed for MVP!

How much you needed to do for this step depends heavily on your specific mvp definition.

TP2 - TP3: Finish Things Up

Once you’ve been confirmed for MVP you can have fun making your project whatever you imagine it to be!

This is the spot where you make the UI beautiful.

\*\*\*Code Organization

What to put in different files

Mian file to run the main animation (import all other files into this file)

Major classes in their own file

Big modes in their own file

General helper functions/baby modes in one file

Style and Comments

Avoid bugs!

More credits!

Git!

powerful software tool used for version control

what is version control?

the practice of tracking cha nges to a repository of code

Save snapshots of your code at various stages of prohress

better than

copy-pasting code

emailing the code to yourself

take..

easily keep track of different versions of a software project

rollback to old versions of code with a single command

useful when you break something :)

easy to share

git terminology

repository

your code base-saved to a folder on your computer. Also backed up to Gitbun

branch

One continually updated version of your repository

repository begins with just the “master” or “main” branch - you can add more branches

useful for prototyping parts of a project withour having to make changes to the stable branch

Git commands

init - initialize a new repository

clone - download a repository form the internet to your computer. This only happen once.

status

commit -m

add

push

git workflow example

create a new repository on Github.com

Clone the repository to local machine

Write some code

git through the terminal

on Mac, open op your terminal and type in the commands on the next slide