

Team Mezewn (Michael Ruvinshteyn, Jake Zaia, Adeebur Rahman)  
APCS2 pd1

Our project is a series of relatively simple games modelled after the \*.io brand of games that have been popular on the internet recently. This would involve several separate files (one for each game, and probably one as a main driver). Needless to say this would be written in Processing, and despite its simple graphics would probably need time considering it has graphics at all. There are several areas where term 2 data structures and algorithms could be used, but most clearly we would need a sorting algorithm to constantly maintain a scoreboard. Provided we have time/ the ability to/ are physically able to, we may add LAN multiplayer??? (VERY UNLIKELY, but definitely worth a shot if there's time)

Title: "\*.ui" (IO Bootlegs)

- Sagar.ui
  - <http://agar.io>
  - Basic Ball Game
  - MVP
    - Basic Player Sagar that can be controlled by the mouse
    - Basic Mass Balls that can be eaten to increase Player Mass.
    - Player can spit out some of their mass with Q press.
    - Fixed map, player can move freely around.
    - Sagar can not go past the border
  - After MVP
    - Basic AIs running around as enemies. (Varying difficulty) (Likely)
      - Easiest AI: Enemy always goes towards nearest Sagar.
      - Harder AI: Enemy targets Sagars it can eat.
    - Leaderboard (Likely)
    - Sagars can split into two balls? (Semi Likely)
    - Green Spike balls (Small Sagars can hide behind, large ones can eat it but get split.) (Semi Likely)
    - Map centered around player, map updates based on player movement. (Semi Likely)
  - Plan and Stuff with Concepts
    - AISagar and HumanSagar extends Sagar
      - draw()
      - movement()
      - split()
      - compareTo() to compare to other Sagars when consuming
    - Store Sagars balls in ArrayList
    - Mass

- Random value mass
  - Iterate through Sagars to randomly spawn mass.
- split()
  - Divide each ball into 2 add to ArrayList
  - After some time passes, merge to smallest balls.
- Snek.ui
  - <http://slither.io>
  - Basic snake game
  - MVP
    - Basic snek that can be controlled by the mouse
    - Basic mass that can be eaten to increase snek length.
    - Hold down mouse to increase speed while expelling some mass.
    - Fixed map, player can move freely around.
    - Going outside of border = instant death.
  - After MVP
    - Basic AIs running around as enemies. (Likely)
    - Leaderboard (Likely)
    - Map centered around player, map updates based on player movement. (Semi Likely)
  - Plan and Stuff with Concepts
    - AISnek and HumanSnek extend Snek
      - draw()
      - Movement
    - Snek
      - Body can be stored as LinkedList of segments
      - Segments can easily follow the node preceding them.
- Chrench.ui
  - <http://diep.io>
  - A basic tank warfare game
  - MVP
    - Basic Tank that can be controlled by the mouse.
    - Basic shooting via space key or mouse click.
    - Shape classes (triangle, square, pentagons) that can be attacked to increase score.
    - Fixed map, player can move freely around.
    - Tank can not go out of the border
  - After MVP
    - Basic AIs running around as enemies. (Likely)
    - Leaderboard (Likely)

- Map centered around player, map updates based on player movement. (Semi Likely)
  - There would be an interface Part. Many classes would extend Part, and Tank would contain all of these Parts as instance variables (user customization (to an extent)) (Not Likely)
- Plan and Stuff with Concepts
  - Store Tanks in ArrayList?
  - AITank and HumanTank extends Tank
    - draw()
    - shoot()
    - Movement: AI - Random, Human - Mouse-based
    - Class upgrades: Stored in a tree?
  - Triangle, Square, Pentagon extends Block
    - Weak - Strong: Triangle, Square, Pentagon
    - Iterate through Tanks to randomly spawn shapes around them?
- Mar.ui (NOT PART OF MVP, SEMI UNLIKELY)
  - A 2D sidescrolling platformer. (A ripoff of Mario)
  - Store levels in a randomized queue?
    - Either that or procedurally generated terrain
  - If we want another bilingual pun, 丸い *marui* means circular in Japanese, we can have all characters be circles
  - Classes Player and Enemy extend Character
  - Mostly just a Player with no enemies, jumping onto platforms is hard :c
    - Enemies will exist, just only for later difficulties
- War.ui\*
  - (Language note: 悪い *warui* means “bad” in Japanese)
  - Runs MVP version of one of the other games.
  - Only Exists if we get past the other MVP's