Monster Smash Turn-based Game
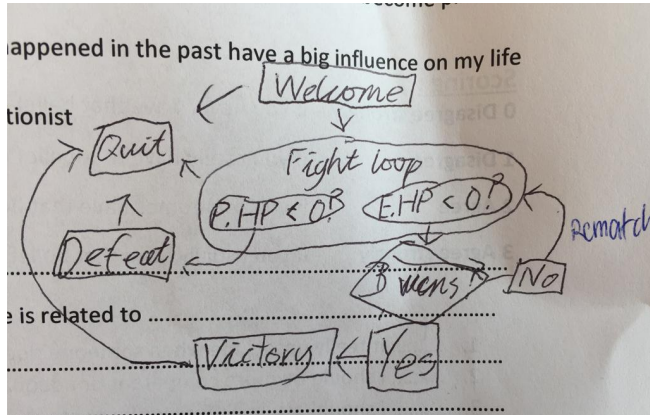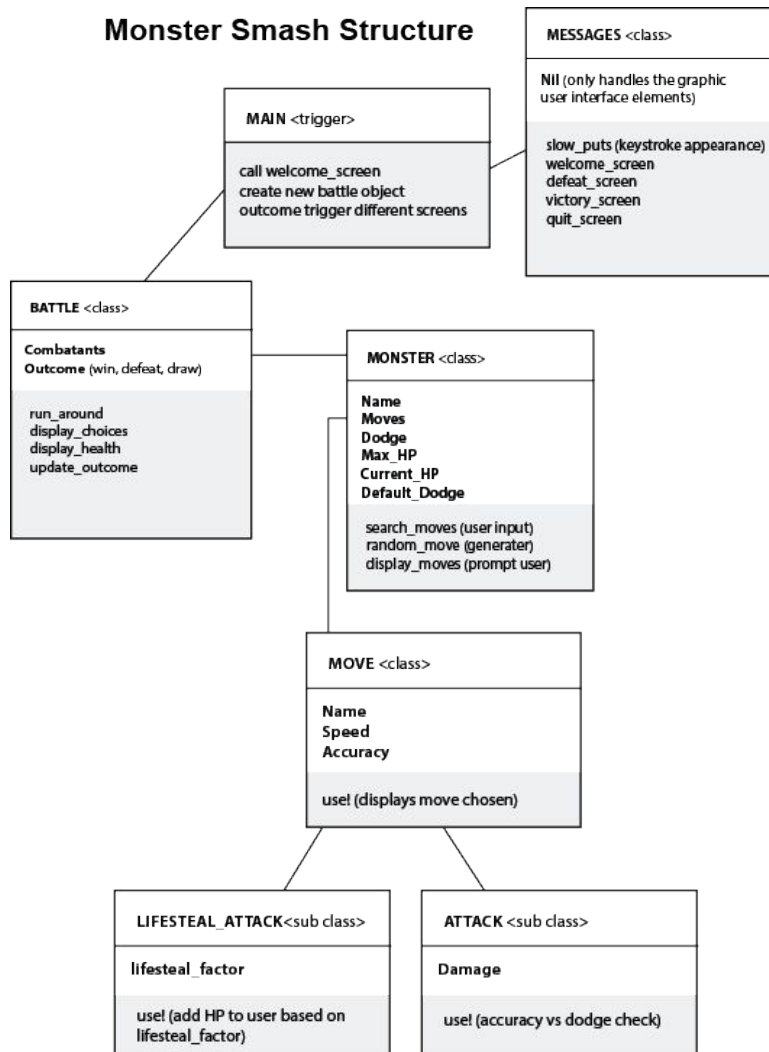
Matthew Puku
Rachel Wong

Concept

- Do something fun
- Challenge coding ability
- Similar to arcade game aesthetics

Monster Smash Structure

MESSAGES <class>

Nil (only handles the graphic user interface elements)

slow_puts (keystroke appearance)
welcome_screen
defeat_screen
victory_screen
quit_screen

MAIN <trigger>

call welcome_screen
create new battle object
outcome trigger different screens

BATTLE <class>

Combatants
Outcome (win, defeat, draw)

run_around
display_choices
display_health
update_outcome

MONSTER <class>

Name
Moves
Dodge
Max_HP
Current_HP
Default_Dodge

search_moves (user input)
random_move (generater)
display_moves (prompt user)

MOVE <class>

Name
Speed
Accuracy

use! (displays move chosen)

LIFESTEAL_ATTACK<sub class>

lifesteal_factor

use! (add HP to user based on lifesteal_factor)

ATTACK <sub class>
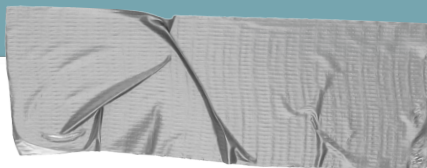
Damage

use! (accuracy vs dodge check)

In a nutshell

Each time you play:

Creates a new battle obj

Each player is a Monster

Each Monster has a series of moves

Future features

➔ Chain multiple battles together in a gauntlet by altering `main.rb`
➔ Allow user to "learn a move" from each monster they defeat, by creating a `loot.rb` file with the required logic
➔ Extra monsters e.g. skeleton, zombie in `monster.rb`
➔ Extra moves e.g. stat buffs, debuffs in `moves.rb`
➔ Randomly choose a monster opponent to play against, which may require the creation of a new class like a gallery (which might sort monsters according to difficulty or theme)
➔ Player vs. player combat
➔ Nicer visualisation using Colorize, ruby-Progressbar