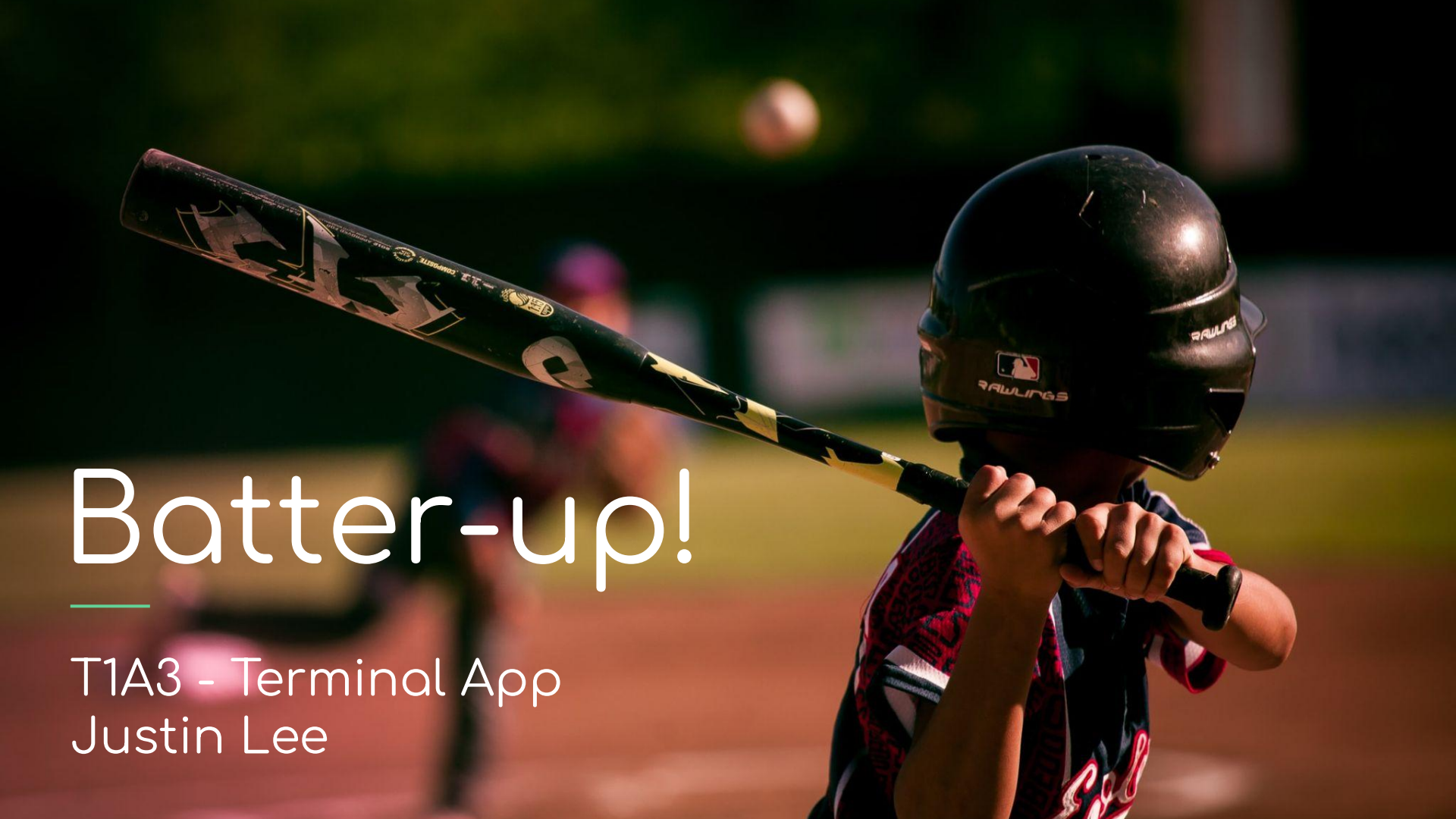


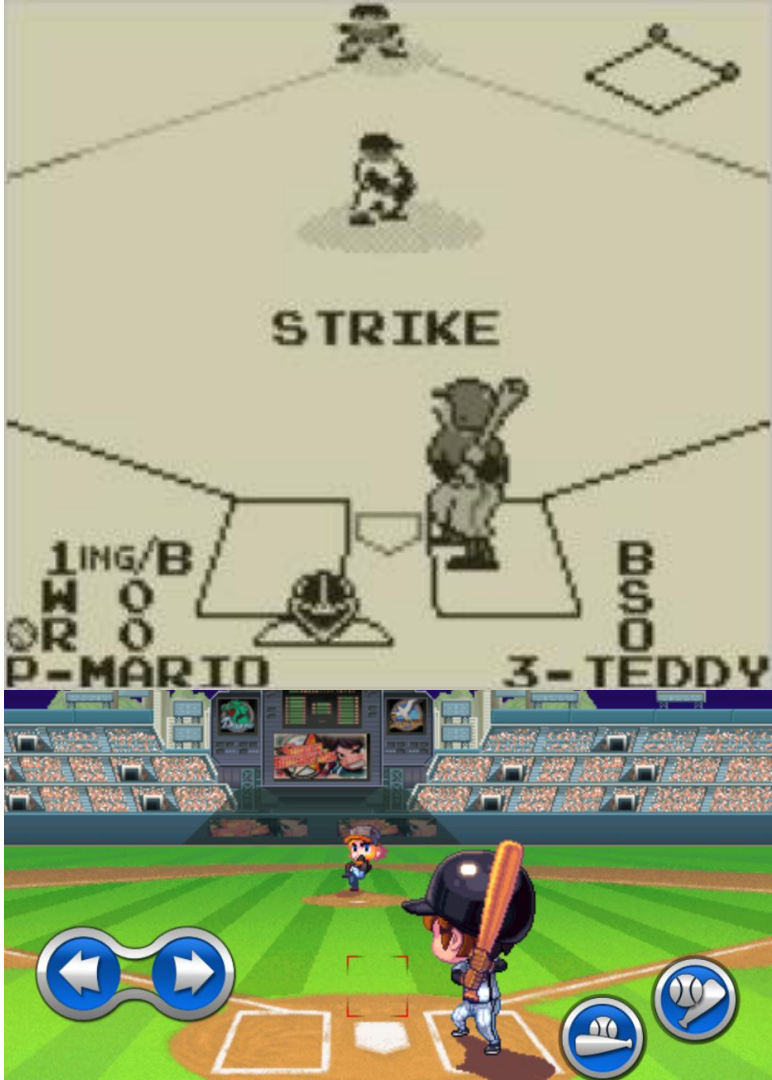
Batter-up!

T1A3 - Terminal App

Justin Lee

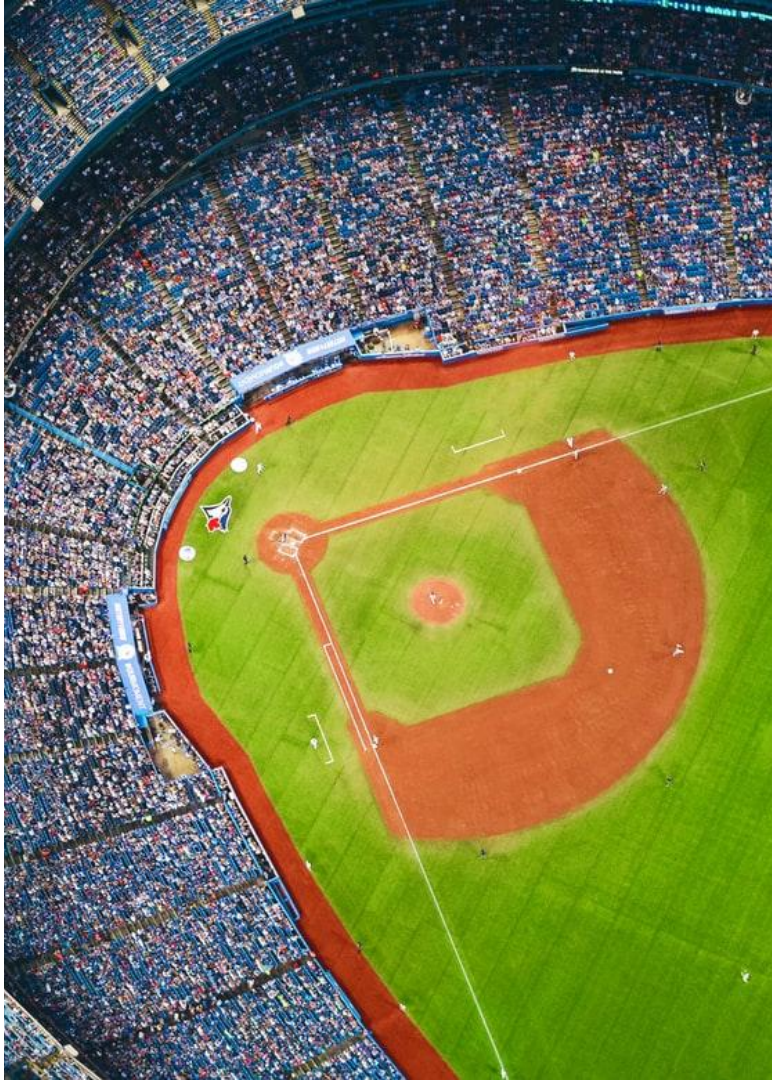
Inspiration

- Inspired from classic baseball games growing up
- One of my first gaming experiences was playing baseball on the original brick gameboy



Purpose & Scope

- Game to simulate one inning at-bat
    - 3 outs is game over
- Advantage of established game logic
    - No new creation of game logic
- Play for the high-score!

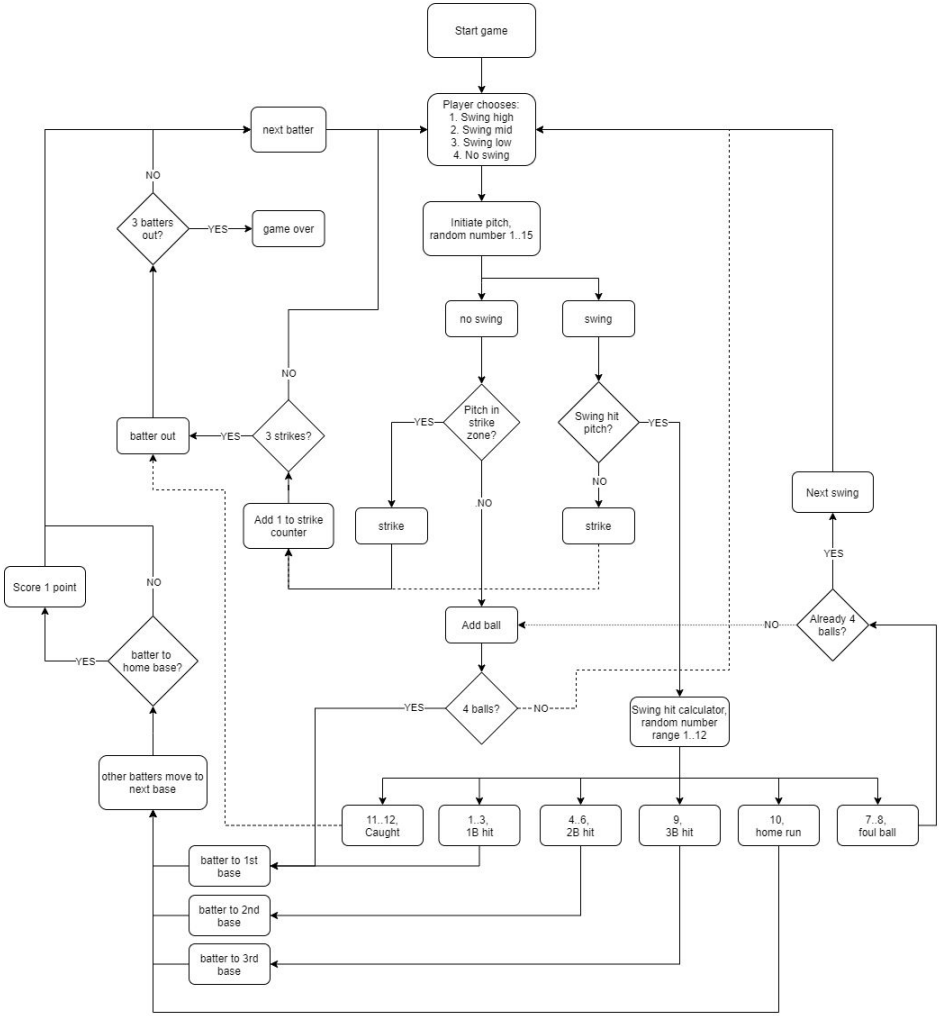


MVP features
(minimum viable product)

- Batting system
  - Random generated pitches
  - Includes strikes, balls & outs
  - 3 outs is game over
- Baserunning tracker
  - Shows current runners on base
- Scoreboard
  - Keeping track of strikes, balls and outs
- Basic graphic interface

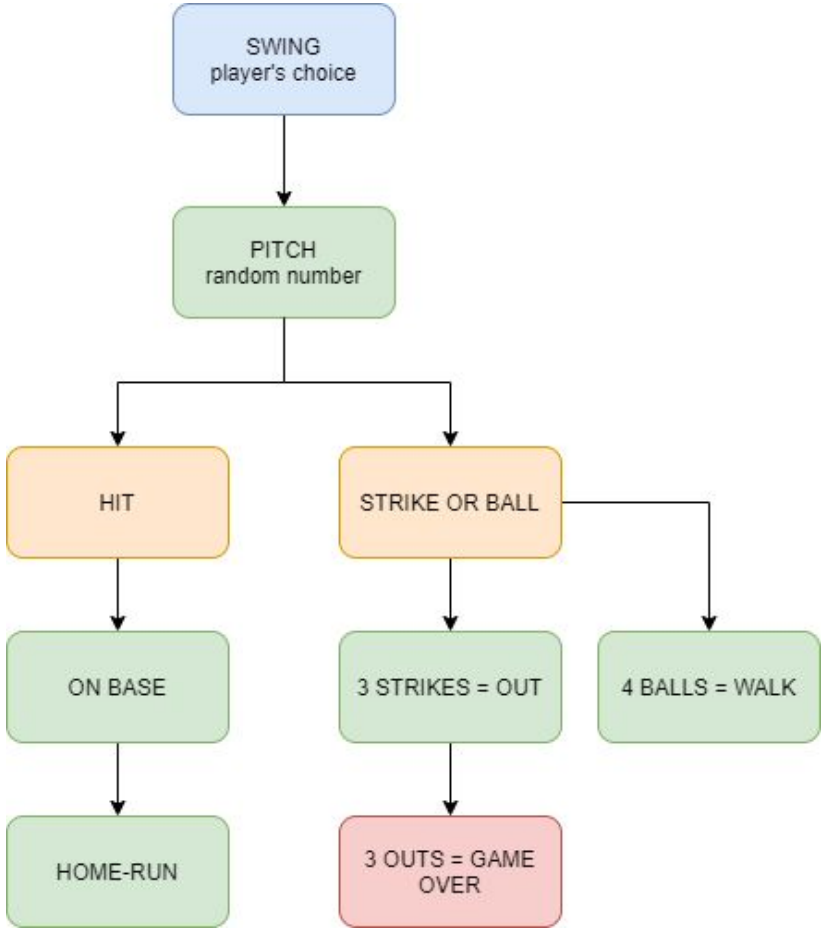
Batting system feature logic
(comprehensive control flow)

- Control flow for a single inning at-bat
- Game over at 3 outs



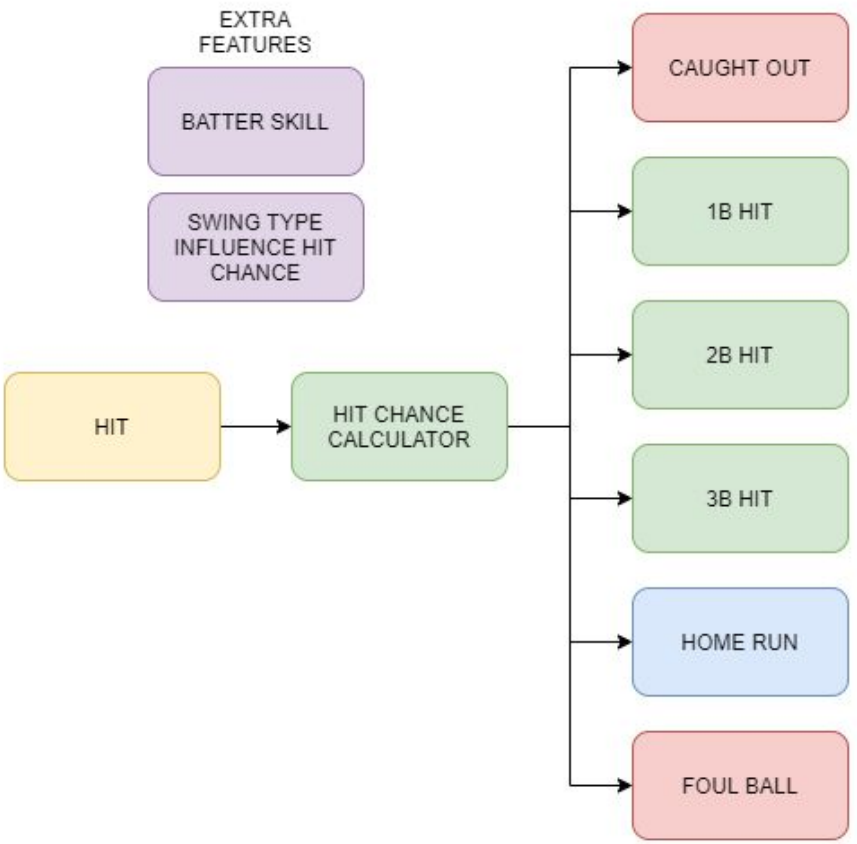
Batting system feature logic (simplified)

- Player chooses swing
- Either hit, strike or ball
- 3 strikes, out
- 3 outs, game over
- Random hit outcome
  - 1 base hit, 2 base hit etc.



Hit Chance Calculator

- Outcome of hit calculated
- Higher chance of 1B and 2B hit, low chance of home run and 3B hit



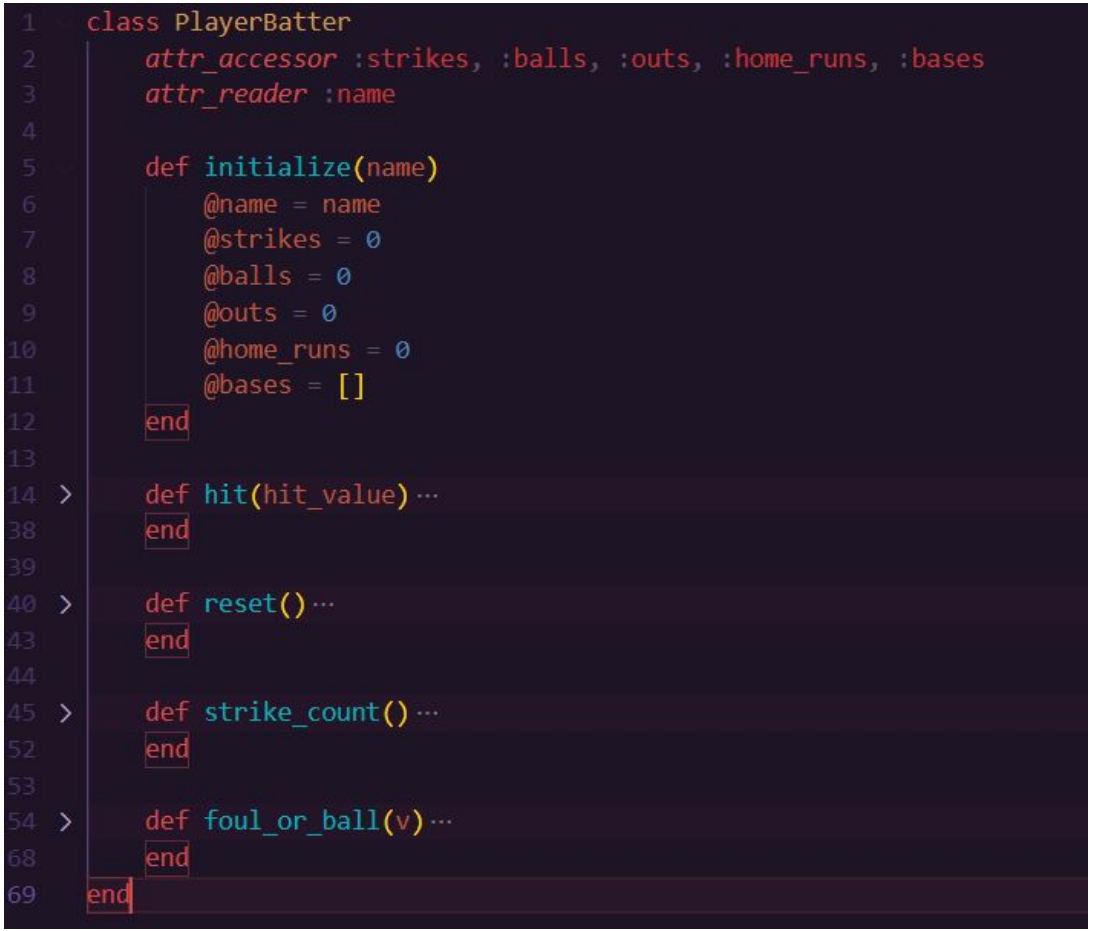
Scoreboard & Baserunning Tracker

- Scoreboard will keep track of:
  - Strikes
  - Balls
  - Outs
- Baserunning tracker will visualise position of runners

App Structure

```ruby
play_game = true

# Game loop
until play_game == false

    # Menu
    choice = prompt.select("Welcome to Batter-Up!", %w(Play High-Scores Exit))

    case choice
    when "Play"

        # Instantiate player class
        player = PlayerBatter.new(prompt.ask("Please enter your name: "))
        outs = player.outs

        # Batting system loop
        until outs == 3···
        end

        # After 3 outs, game over
        system("clear")
        puts "Game Over!"
    when "High-Scores"
        puts "High-Scores"
    when "Exit"
        play_game = false
    end
end

puts "game over!"
```

Batter Class



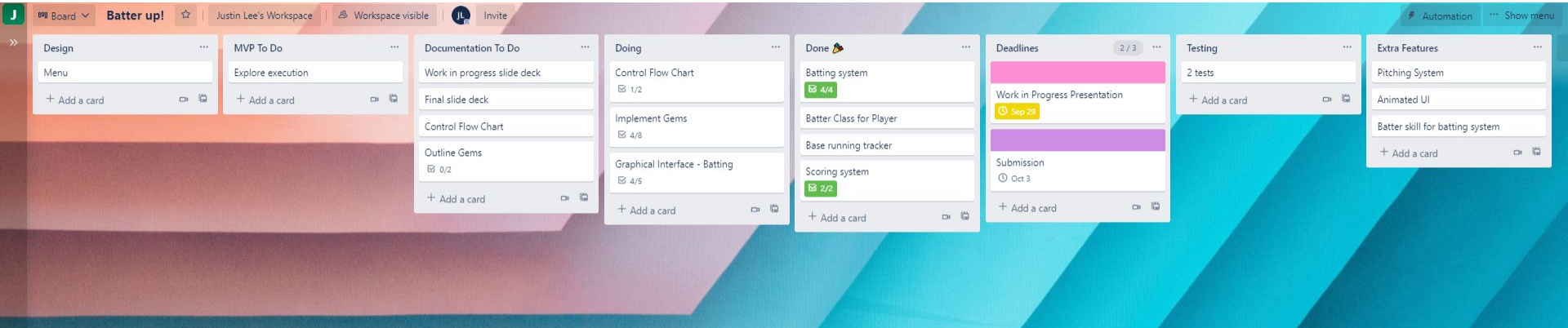
User Interaction

- Player can select swing:
  - High, Mid, Low or No swing
- Depending on swing (or no-swing), app will calculate outcome
- Player must strategize
  - go for swing-hits (high-risk, high reward)
  - Or no-swing for balls to walk-in runs (low-risk, low reward)

Project management & Implementation

- Trello: To-do, doing, done
- Prioritize MVP

- Divided items into:
  - Design (graphic interface)
  - MVP features
  - Documentation
  - Deadlines
  - Testing
  - Extra features



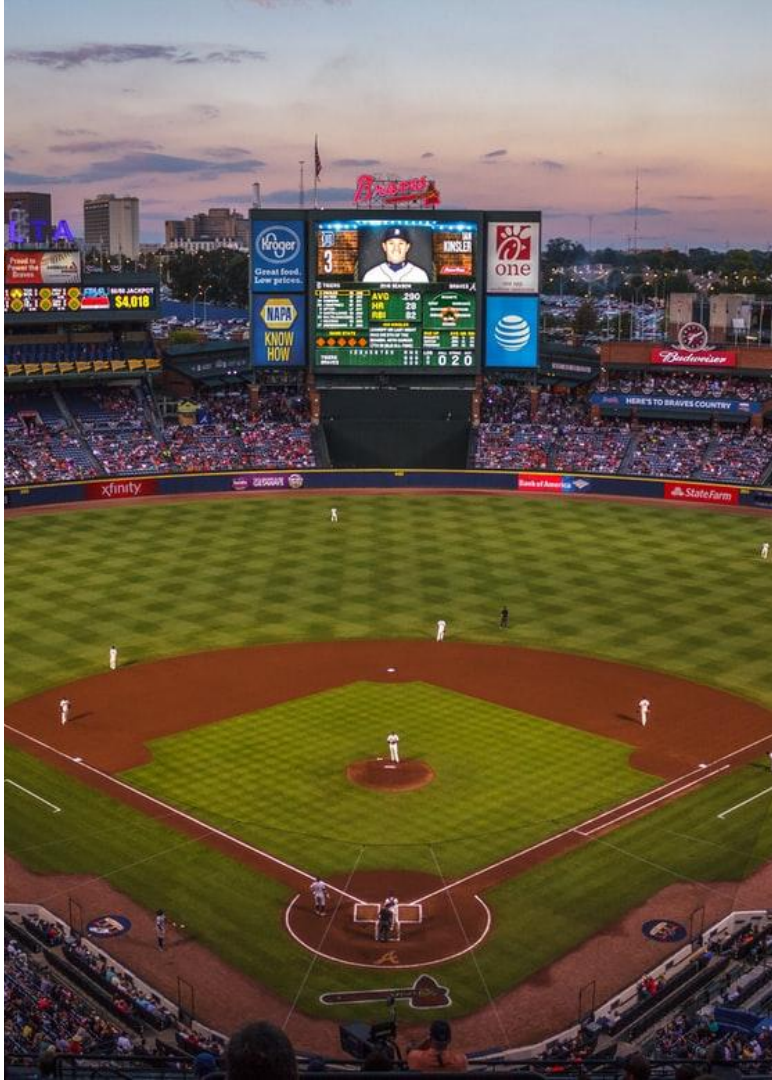
Gems

- MVP gems:
  - TTY-prompt
  - TTY-box
  - TTY-font
  - Bundler

- To test/add:
  - Artii
  - Colorize
  - Paint

Challenges (so-far)

- Converting baseball logic to game logic
    - How to utilize Ruby to express the game logic
- Challenging & entertaining for the player
- Finding relevant gems



To-do

- Menu
- Documentation
- Implement batter skill
- Chance to get caught out
- High-scores
- Testing
- Additional gems
- Execution

Features wishlist

- Swing-type to affect hit outcome
  - E.g high swing = high chance to hit home runs, but also be caught
- Batter skill
  - Skill affects hit calculations
- Upgradable batter
- Animated graphic interface
- Pitching feature
- Batting + pitching = whole game!

Demonstration