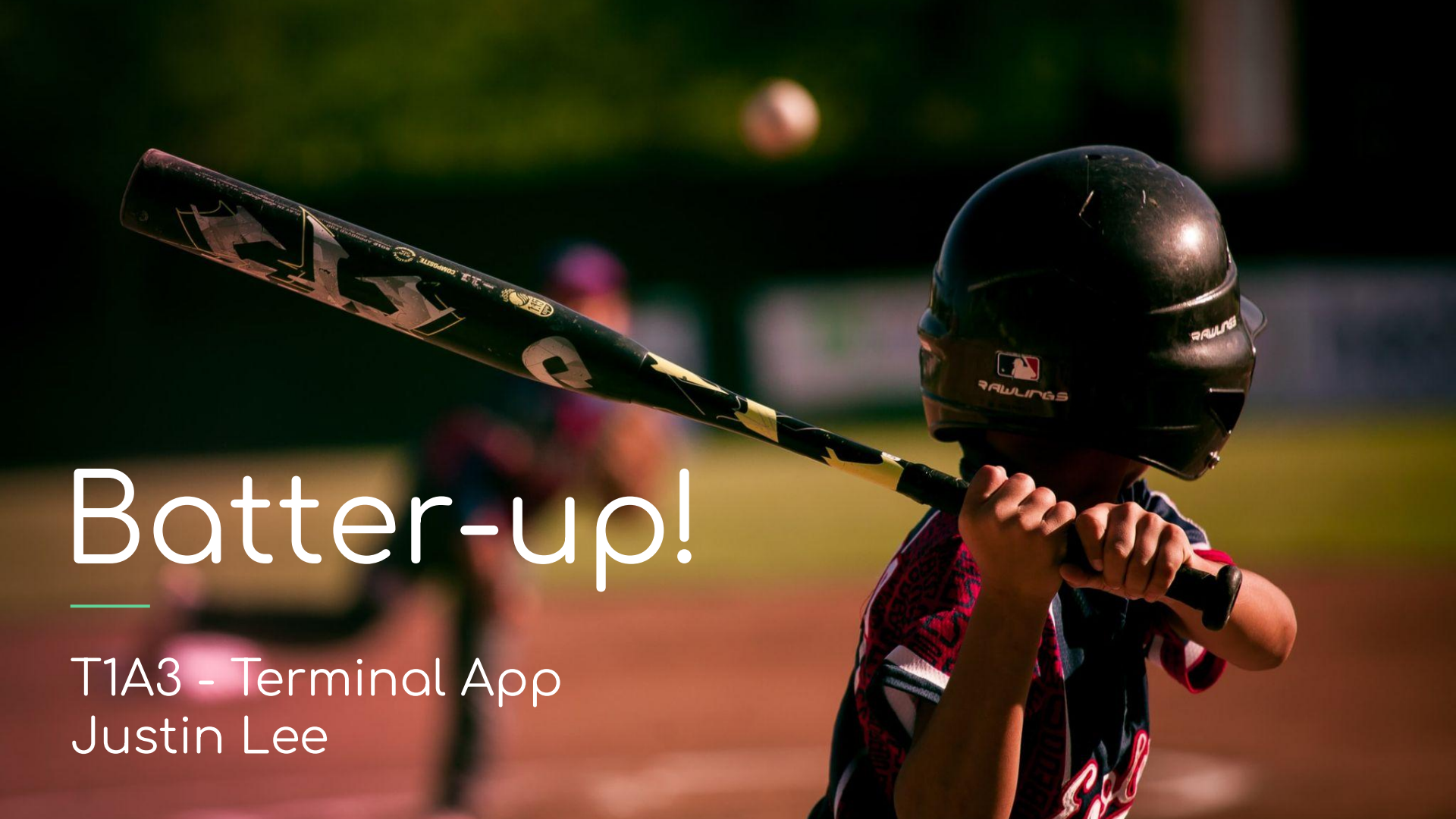


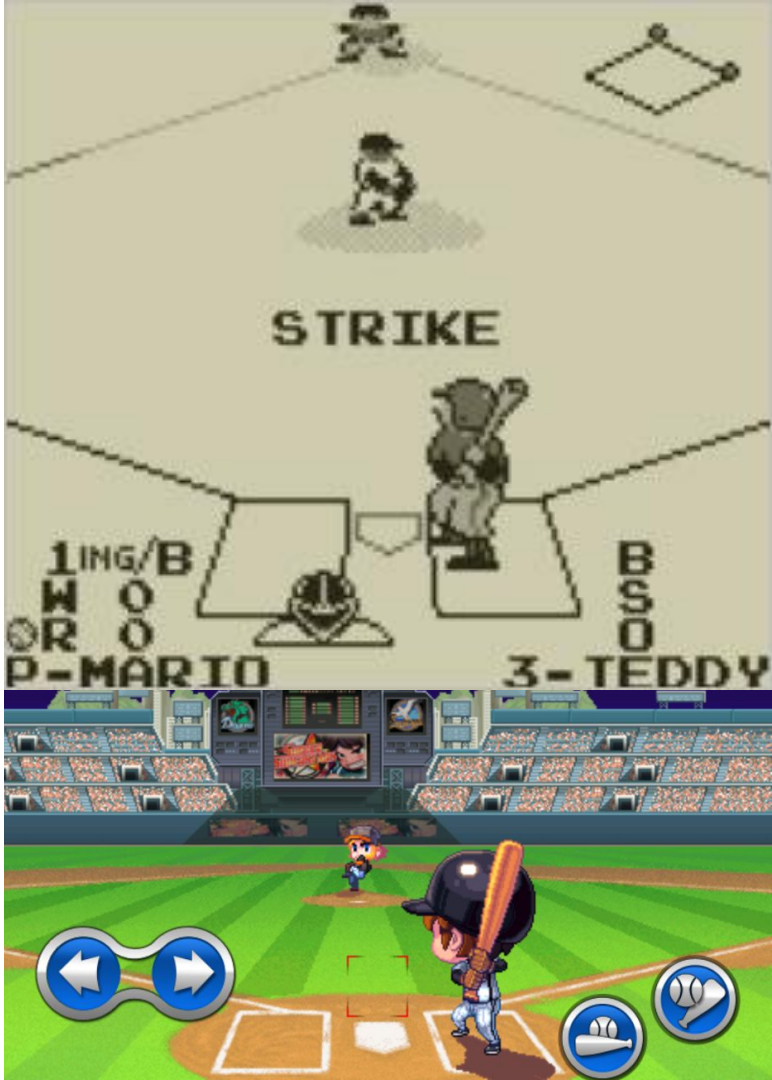
Batter-up!

T1A3 - Terminal App

Justin Lee

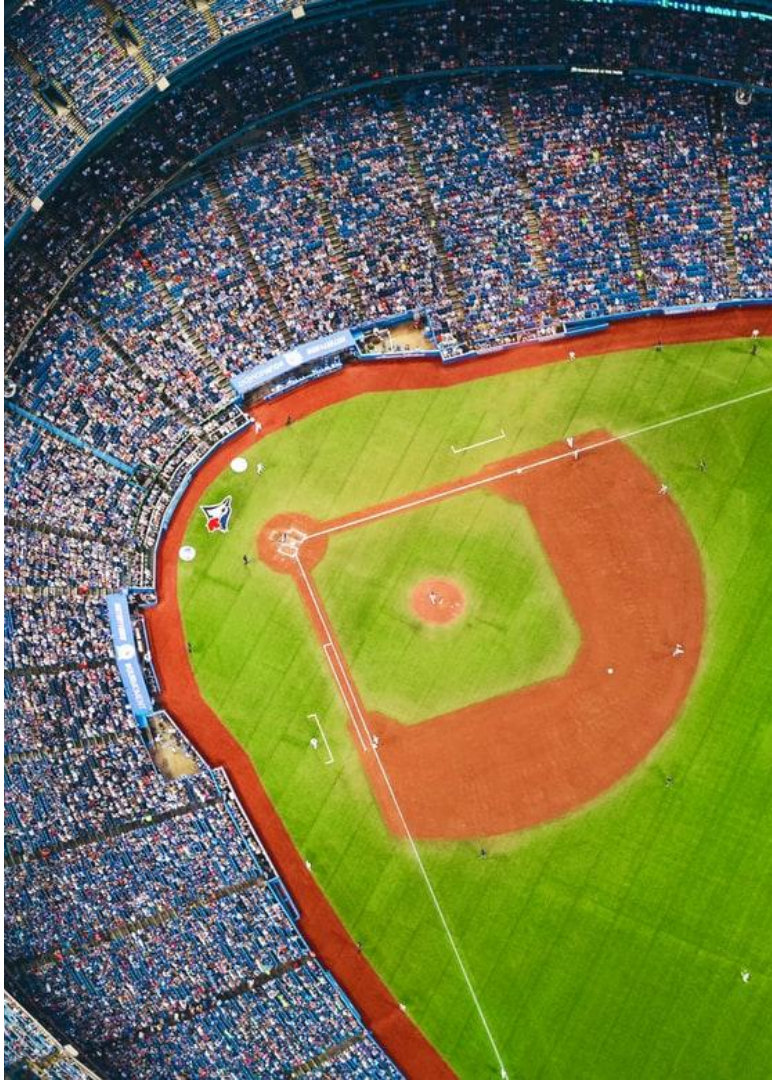
Inspiration

- 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!

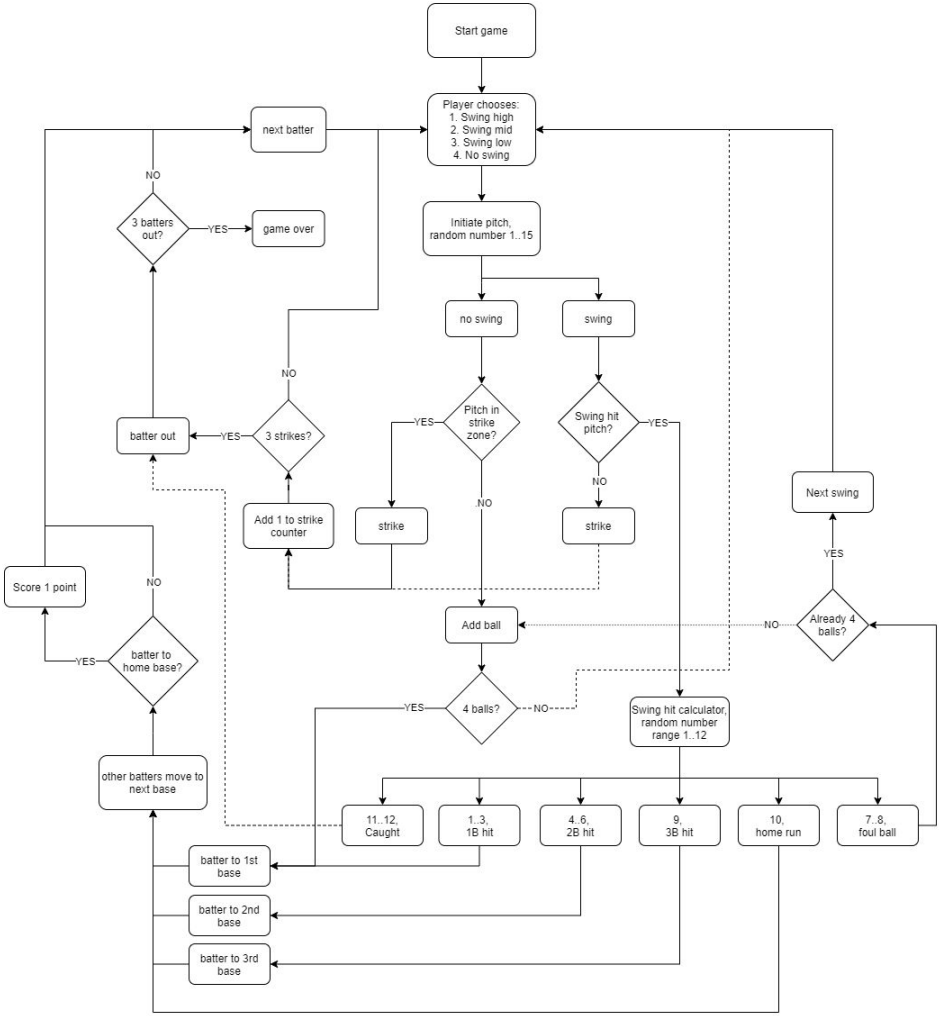


MVP features
(minimum viable product)

- Batting system
  - Includes pitches, strikes, balls & outs
  - 3 outs is game over
- Batter class
- High Scores
- Scoreboard
  - Baserunning tracker
  - Keeping track of strikes, balls and outs
- Basic graphic interface
- Interactive menu

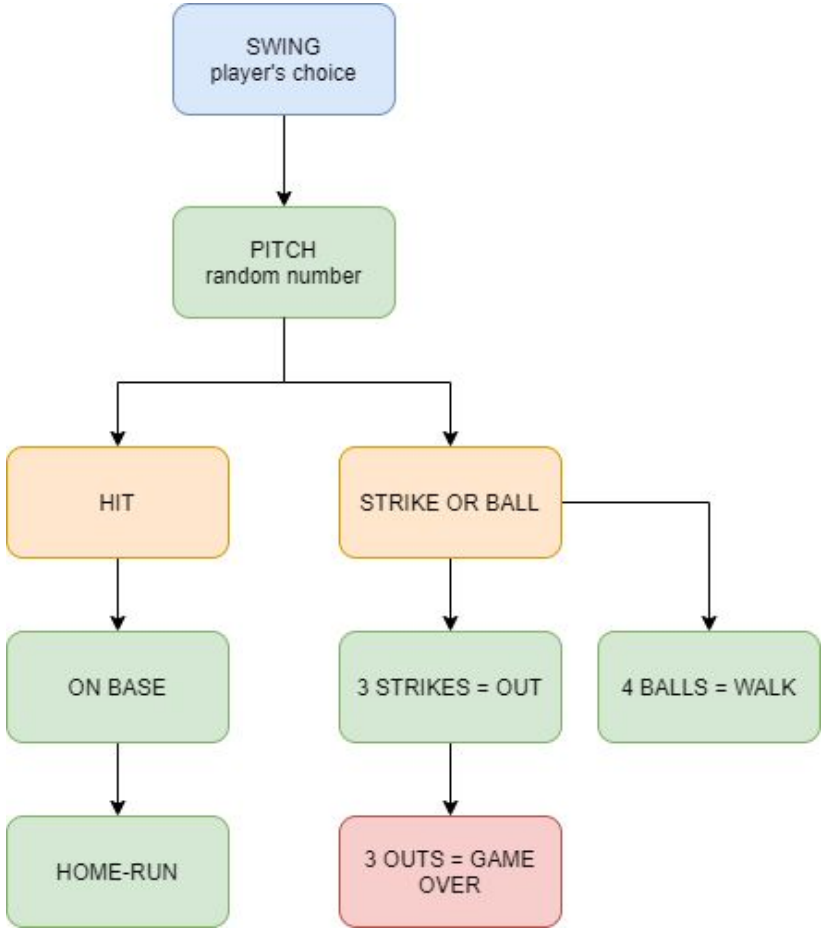
Batting system feature logic
(comprehensive control flow)

- Control flow for a single inning at-bat
  - Random pitch
  - Player chooses swing
- Game over at 3 outs



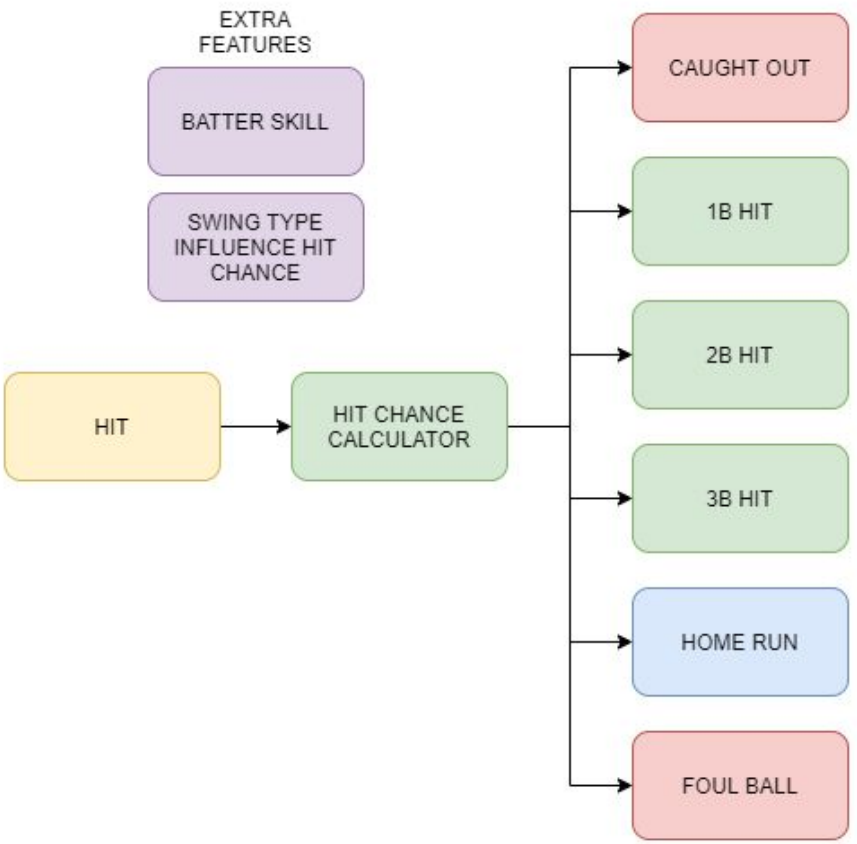
Batting system feature logic (simplified)

- 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 display:
  - Strikes
  - Balls
  - Outs
- Baserunning tracker will visualise position of runners
- Utilize gems tty-font for the game title and tty-box for framing

Batter Class

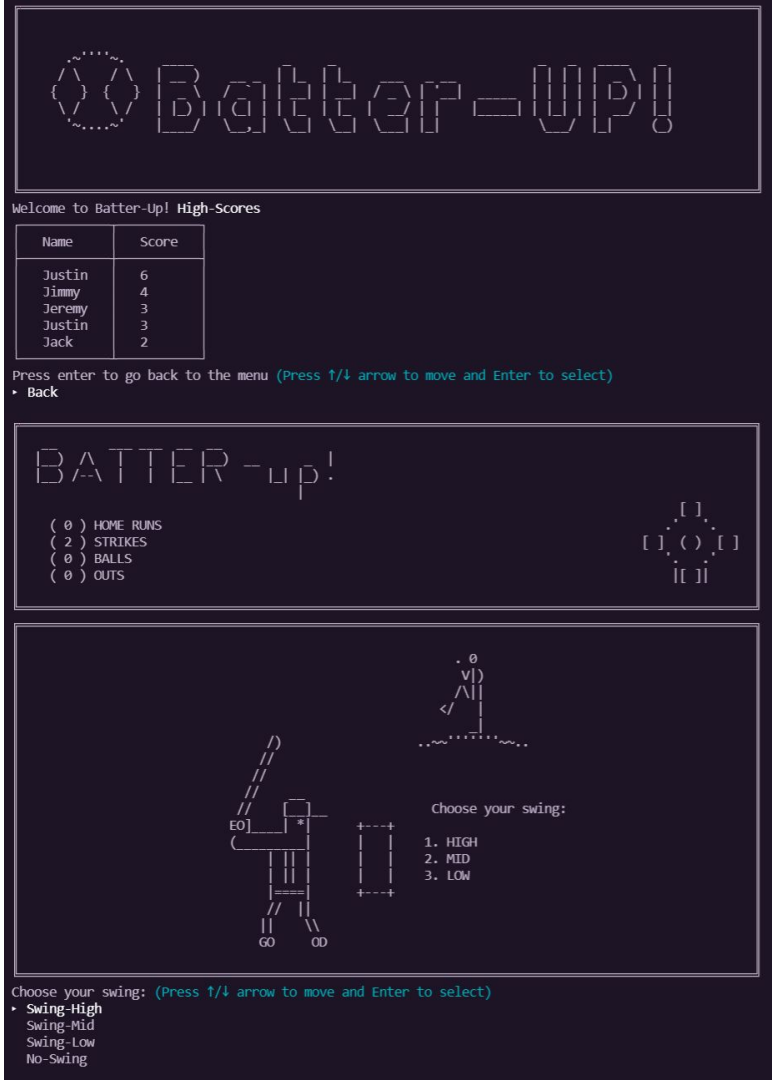
- Instance of class will track:
  - Name
  - Strikes
  - Balls
  - Outs
  - Home Runs
  - Bases

- Methods in instance will be called in main.rb to affect instance variables

```ruby
class PlayerBatter
    attr_accessor :strikes, :balls, :outs, :home_runs, :bases
    attr_reader :name

    def initialize(name)
        @name = name
        @strikes = 0
        @balls = 0
        @outs = 0
        @home_runs = 0
        @bases = []
    end

    def hit(hit_value)···
    end

    def reset()···
    end

    def strike_count()···
    end

    def foul_or_ball(v)···
    end
end
```

Menu & Graphics

- Basic graphic interface created from ascii and ruby gems tty-box, tty-font, tty-prompt and tty-table
- Provides visual interest and feedback to the player
- Animated menu title!



High-Scores feature

- High scores are stored as hashes in an array
- Methods retrieve and save new records into a yaml file

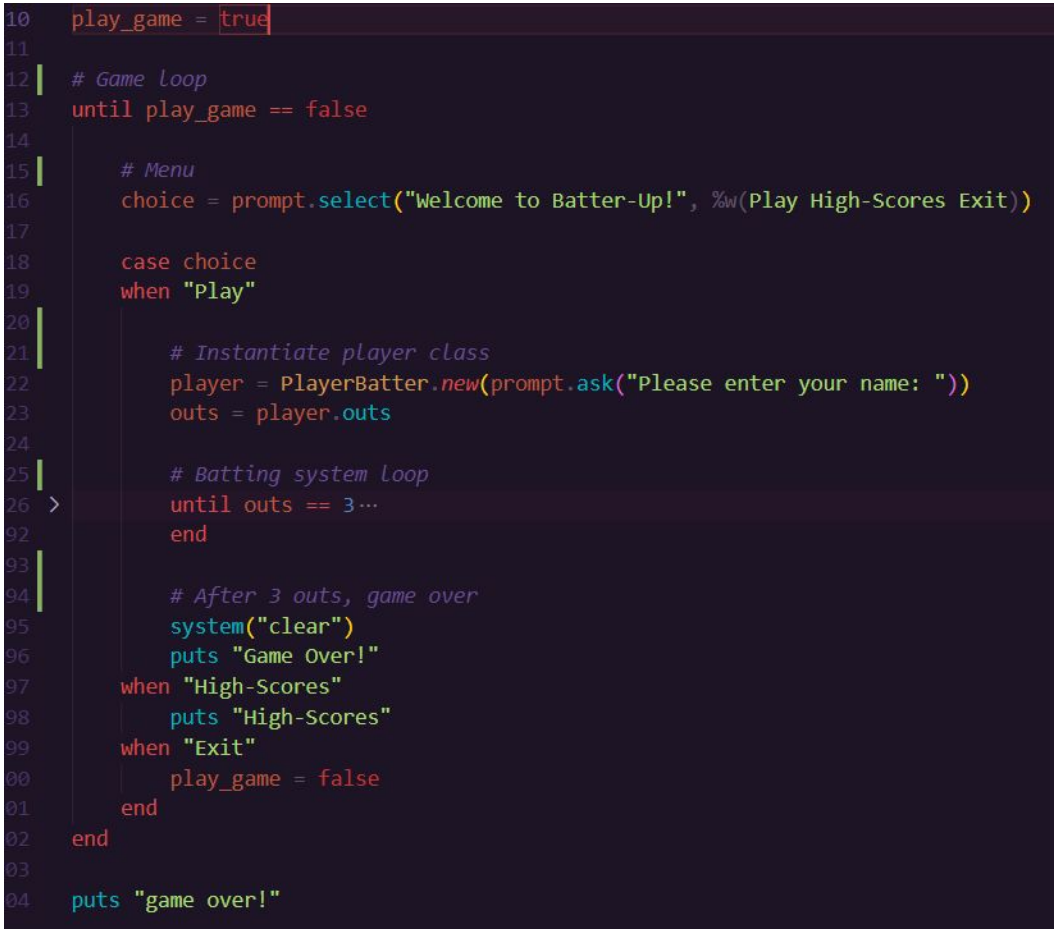
```
Welcome to Batter-Up! High-Scores
```

Name	Score
Justin	6
Jimmy	4
Jeremy	3
Justin	3
Jack	2

```
Press enter to go back to the menu (Press ↑/↓ arrow to move and Enter to select)
‣ Back
```

Main Structure

- Main loop controlled by boolean play_game
- Menu options case statement
- Batting system controlled by until outs === 3 loop



```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!"
```

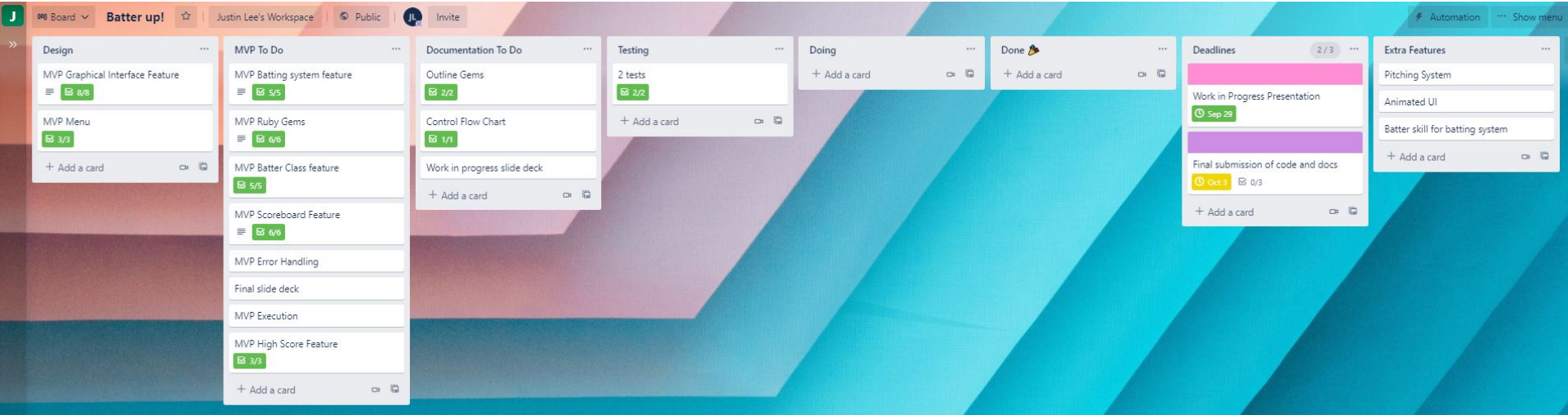
User Interaction

- 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

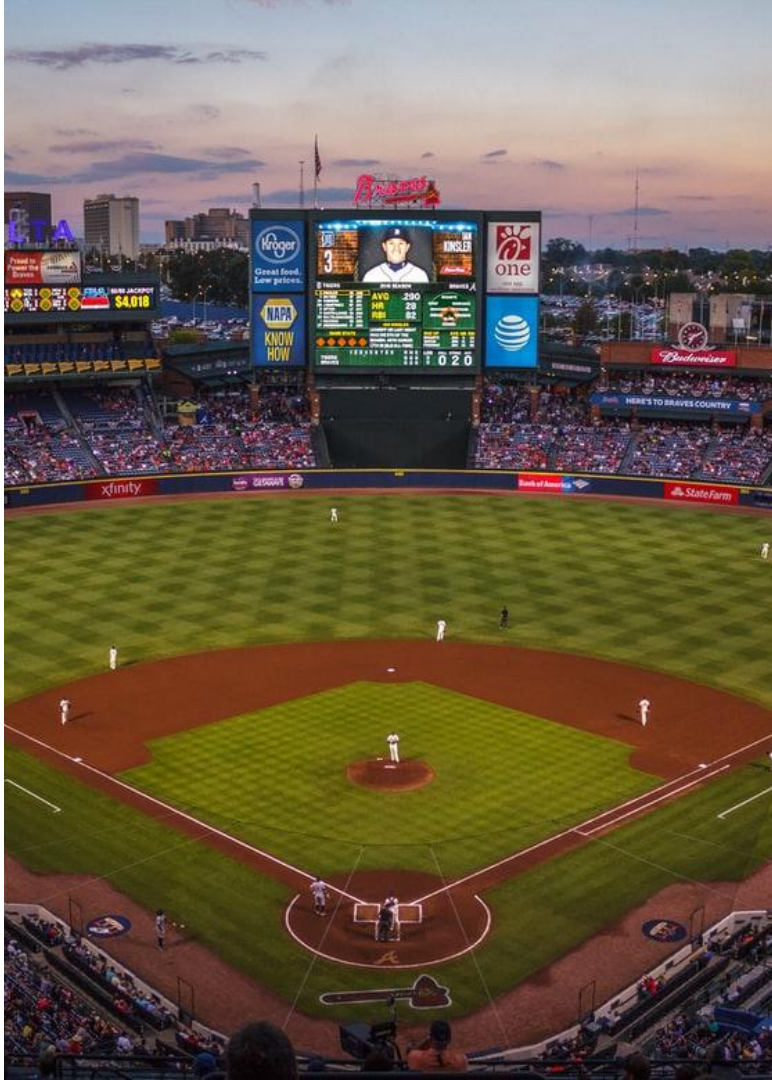


Gems

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

Challenges

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



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