
T1A3 - Terminal App Assessment

— -Mitchell Boyle- —

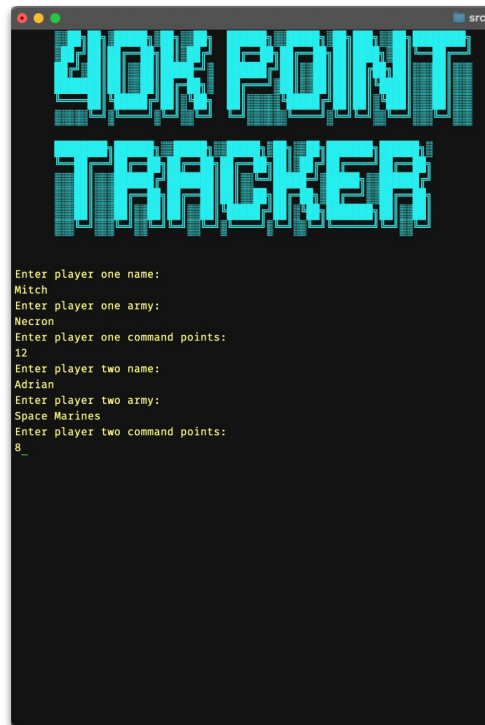
Overview

- The target audience: Tabletop war game players and peers.
- Used to assist players in keep track of Victory points and Command points while playing Warhammer 40k.
- Showcase my knowledge using the Ruby language.

Front-end walk through

When App is launched - Prompt user to for player information.

Once input is collected - Display the information.



```
40K POINT TRACKER

Enter player one name:
Mitch
Enter player one army:
Necron
Enter player one command points:
12
Enter player two name:
Adrian
Enter player two army:
Space Marines
Enter player two command points:
8
_
```



```
40K POINT TRACKER

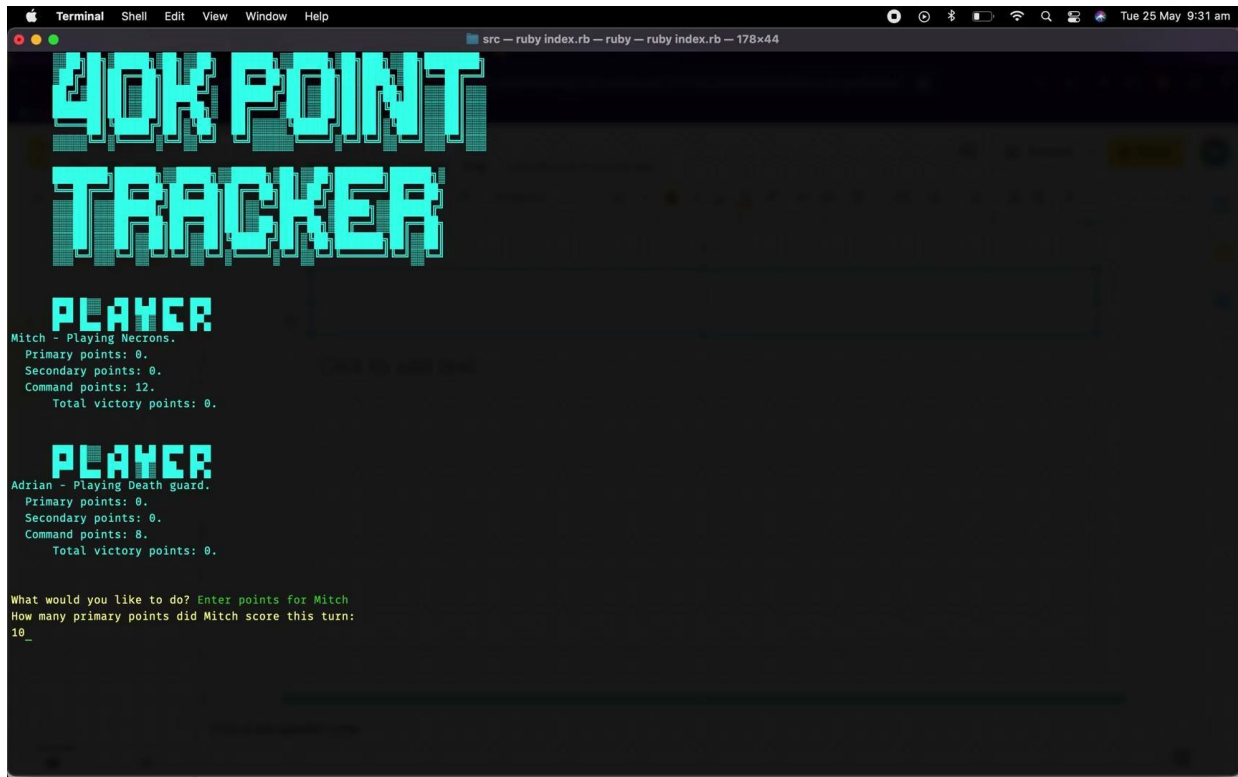
PLAYER
Mitch - Playing Necrons.
Primary points: 0.
Secondary points: 0.
Command points: 12.
Total victory points: 0.

PLAYER
Adrian - Playing Death guard.
Primary points: 0.
Secondary points: 0.
Command points: 8.
Total victory points: 0.

What would you like to do? (Press ↑/↓ arrow to move and Enter to select)
* Enter points for Mitch
  Enter spent command points for Mitch
  Enter points for Adrian
  Enter spent command points for Adrian
  Exit
```

Front-end walk through

Menu demonstration



```
Terminal Shell Edit View Window Help
src — ruby index.rb — ruby — ruby index.rb — 178x44

40K POINT
TRACKER

PLAYER
Mitch - Playing Necrons.
Primary points: 0.
Secondary points: 0.
Command points: 12.
Total victory points: 0.

PLAYER
Adrian - Playing Death guard.
Primary points: 0.
Secondary points: 0.
Command points: 8.
Total victory points: 0.

What would you like to do? Enter points for Mitch
How many primary points did Mitch score this turn:
10
_
```

Back-end walk through

Gather user information.

Initialize a new player using
the Player class.

```
#input for player 1
puts "Enter player one name: "
player1_name = gets.chomp
puts "Enter player one army: "
player1_army = gets.chomp
puts "Enter player one command points: "
player1_cp = gets.chomp
player1_pp = 0
player1_sp= 0

player1 = Player.new(player1_name, player1_army, player1_pp, player1_sp, player1_cp)

#input for player 2
puts "Enter player two name: "
player2_name = gets.chomp
puts "Enter player two army: "
player2_army = gets.chomp
puts "Enter player two command points: "
player2_cp = gets.chomp
player2_pp = 0
player2_sp = 0

player2 = Player.new(player2_name, player2_army, player2_pp, player2_sp, player2_cp)
```

Back-end walk through

Player class break down.

```
class Player

  def initialize(name, army, prim_points, sec_points, cmd_points)
    @name = name.capitalize
    @army = army.capitalize
    @prim_points = prim_points.to_i
    @sec_points = sec_points.to_i
    @cmd_points = cmd_points.to_i
  end

  def display_score
    puts "      [X][X][X][X][X][X].light_cyan
    puts "      [X][X][X][X][X][X].light_cyan
    puts "#{@name} - Playing #{@army}."light_cyan
    puts "    Primary points: #{@prim_points}."light_cyan
    puts "    Secondary points: #{@sec_points}."light_cyan
    puts "    Command points: #{@cmd_points}."light_cyan
    puts "      Total victory points: #{@prim_points + @sec_points}."light_cyan
  end

  attr_reader :name, :army
  attr_accessor :prim_points, :sec_points, :cmd_points
end
```

You, 11 hours ago • Moved user menu out of method

Back-end walk through

TTY-Prompt Menu

```
menu_input = prompt.select("What would you like to do?") do |menu|  
  menu.choice name: "Enter points for #{player1.name}", value: 1  
  menu.choice name: "Enter spent command points for #{player1.name}", value: 2  
  menu.choice name: "Enter points for #{player2.name}", value: 3  
  menu.choice name: "Enter spent command points for #{player2.name}", value: 4  
  menu.choice name: "Exit", value: 5  
end
```

Dependencies

Currently requires the following Gems:

Colorize

TTY-Prompt

Challenges

Command points

Artii

Instance variable scope

Favourite Parts

Creating the play Class

Understanding Getters and Setters

TTY-Prompt / TTY Library

Ethical Issues

- Legality behind the app
- Selecting colours



Thank You!

