Routinely

A Terminal App

By

Ryan Wise

Contents

What is Routinely?

In Brief:

Routinely aims to provide a method for users to quickly create and reorder a sequence of events or tasks.

The Long Answer:

Sometimes you need to plan out a series of tasks or events. Usually, the best and quickest way to do this is jot it down on a piece of paper. However, this becomes a problem if that sequence is something you want to refer to later or even refine. You could lose the paper and have to start again, or in the case of reshuffling need to rewrite the list.

All well and good while the list is short, but once it reaches a certain length this becomes a tedious process. Routinely aims to solve this problem by allowing a user to quickly add a name and estimated time for each event and storing the sequence in a routine. This will allow the user to save, edit and even shuffle the sequence of events as they see fit.

The Audience

The
Everyday User

Maybe you're somebody trying to work out a morning routine or plan the perfect exercise plan, Routinely's got you.

The
Event Organiser

Planning big events and need work out what order things need to happen? Maybe you're frontrunners running late and you need to shuffle the plan around at the last minute to keep the show running. Routinely's got you.

The
Developer

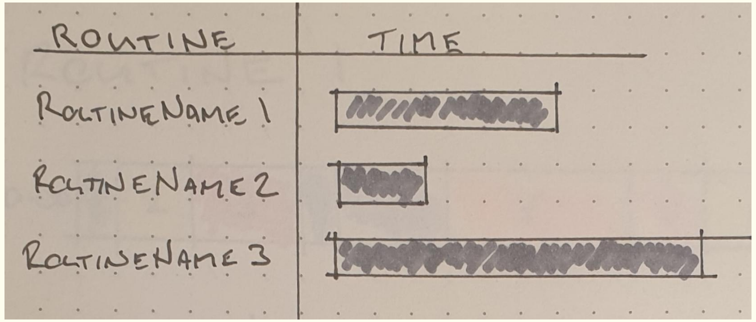
Doing a routine task and need to know how long it'll take? Or starting a new project and want to be sure you don't skip a step? Stop guessing and use Routinely.

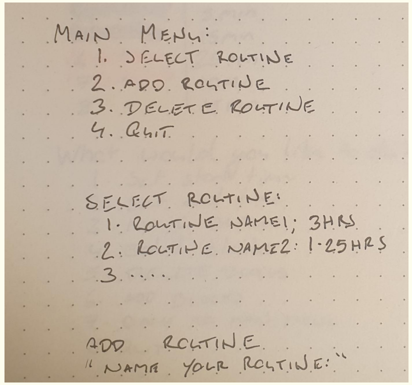
Minimum Viable Product

- Create, Read, Update, and Delete routines
- Quick, Easy and Efficient data entry
- Visual display of routines for fast reference
- Time calculations

Design Process

First Impression Sketches of what the app would look like:

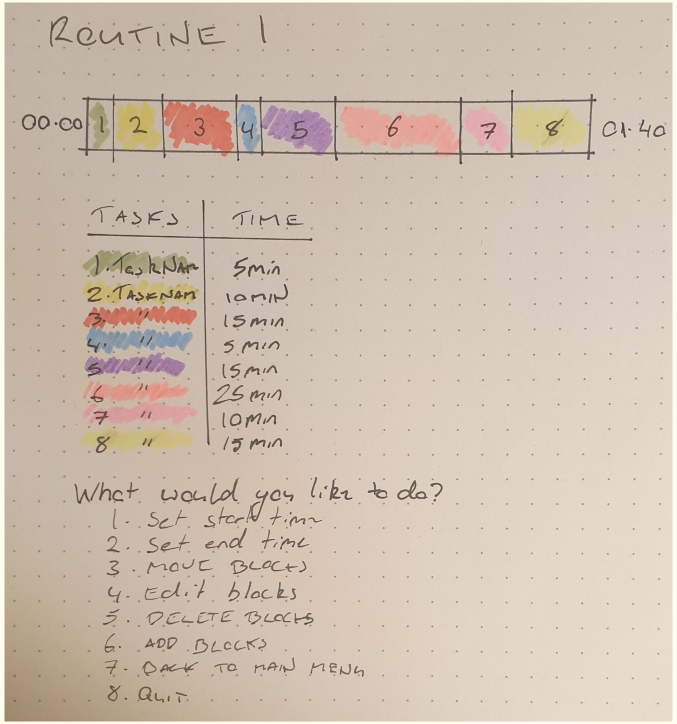




MAIN MENU:
1. SELECT ROUTINE
2. ADD ROUTINE
3. DELETE ROUTINE
4. QUIT

SELECT ROUTINE:
1. ROUTINE NAME1; 3HRS
2. ROUTINE NAME2: 1·25HRS
3. ....

ADD ROUTINE
"NAME YOUR ROUTINE:"



ROUTINE 1

TASKS	TIME
1. TaskNam	5min
2. TASKNAM	10MIN
3. "	15min
4. "	5min
5 "	15min
6 "	25min
7 "	10min
8 "	15min

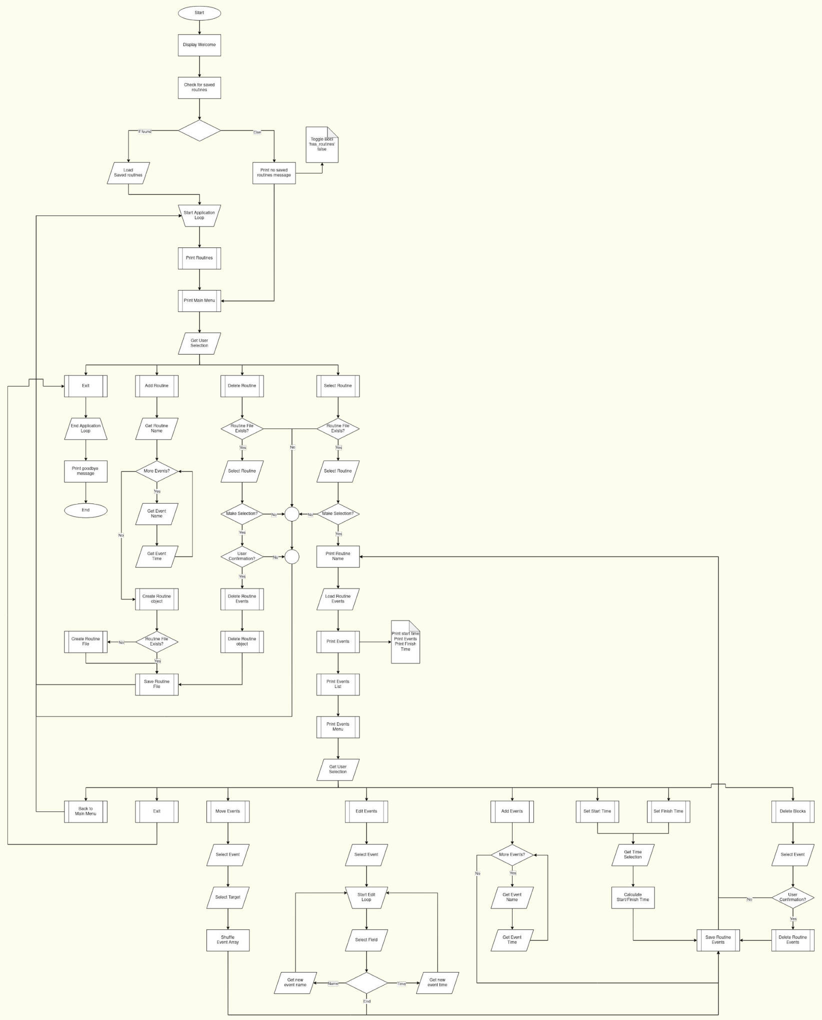
What would you like to do?
1. Set start time
2. Set end time
3. MOVE BLOCKS
4. Edit blocks
5. DELETE BLOCKS
6. ADD BLOCKS
7. BACK TO MAIN MENU
8. QUIT

Flowchart

Thought through initial implementations with a flowchart

Not married to the plan, just using it as a way to organise thoughts.

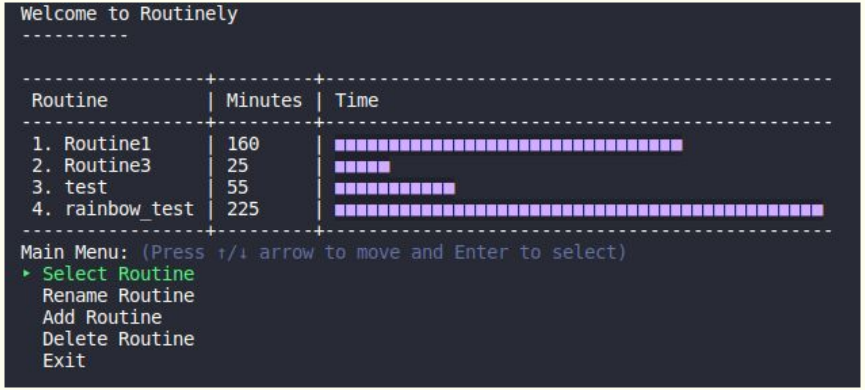
- How the loops work?
- Where are the branches?
- How do the menu's work together?



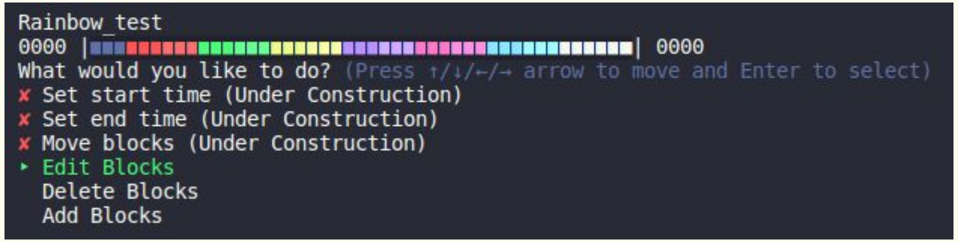
The Application

Really pleased with how it's coming together.

There's still a few bugs to iron out and a few core features missing, but I think it's coming along well!



```
Welcome to Routinely
----------

-----------------+---------+-----------------------------------------------
 Routine         | Minutes | Time
-----------------+---------+-----------------------------------------------
 1. Routine1     | 160     | ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
 2. Routine3     | 25      | ■■■■■
 3. test         | 55      | ■■■■■■■■■■■
 4. rainbow_test | 225     | ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
-----------------+---------+-----------------------------------------------
Main Menu: (Press ↑/↓ arrow to move and Enter to select)
▸ Select Routine
  Rename Routine
  Add Routine
  Delete Routine
  Exit
```



```
Rainbow_test
0000 |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■| 0000
What would you like to do? (Press ↑/↓/←/→ arrow to move and Enter to select)
✘ Set start time (Under Construction)
✘ Set end time (Under Construction)
✘ Move blocks (Under Construction)
▸ Edit Blocks
  Delete Blocks
  Add Blocks
```

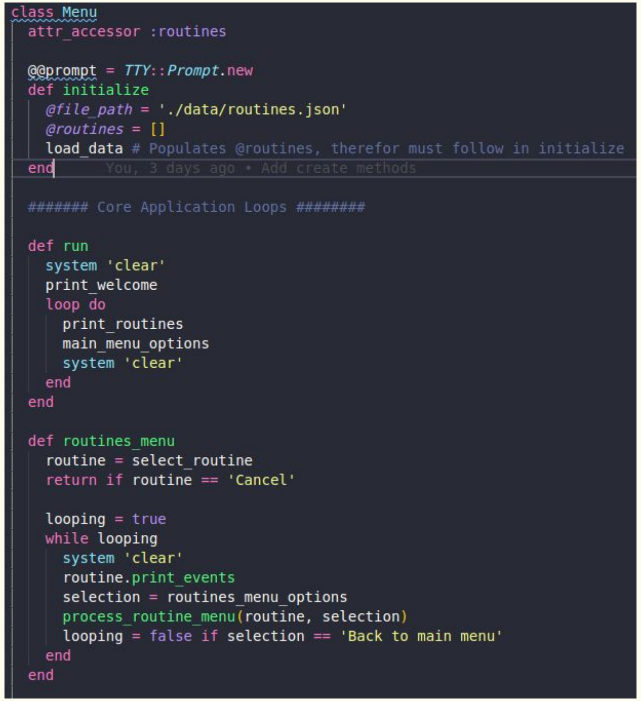
Under The Hood

Consists of two classes:

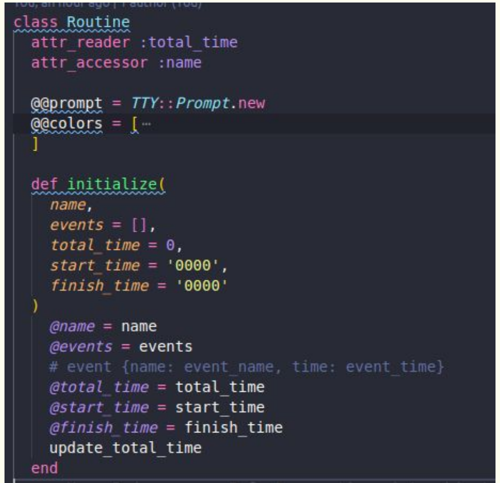
- The Menu
- The Routines

Ideally I would like to further abstract events and have routines stored as separate files.

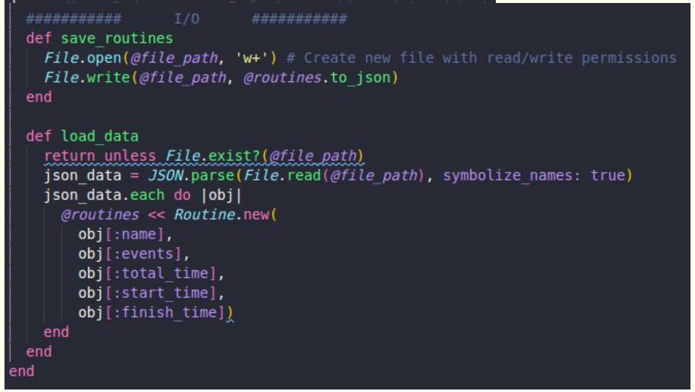
This is not part of the MVP however.



```ruby
class Menu
  attr_accessor :routines

  @@prompt = TTY::Prompt.new
  def initialize
    @file_path = './data/routines.json'
    @routines = []
    load_data # Populates @routines, therefor must follow in initialize
  end        You, 3 days ago • Add create methods

  ####### Core Application Loops ########

  def run
    system 'clear'
    print_welcome
    loop do
      print_routines
      main_menu_options
      system 'clear'
    end
  end

  def routines_menu
    routine = select_routine
    return if routine == 'Cancel'

    looping = true
    while looping
      system 'clear'
      routine.print_events
      selection = routines_menu_options
      process_routine_menu(routine, selection)
      looping = false if selection == 'Back to main menu'
    end
  end
```



```ruby
class Routine
  attr_reader :total_time
  attr_accessor :name

  @@prompt = TTY::Prompt.new
  @@colors = [ ⋯
  ]

  def initialize(
    name,
    events = [],
    total_time = 0,
    start_time = '0000',
    finish_time = '0000'
  )
    @name = name
    @events = events
    # event {name: event_name, time: event_time}
    @total_time = total_time
    @start_time = start_time
    @finish_time = finish_time
    update_total_time
  end
```



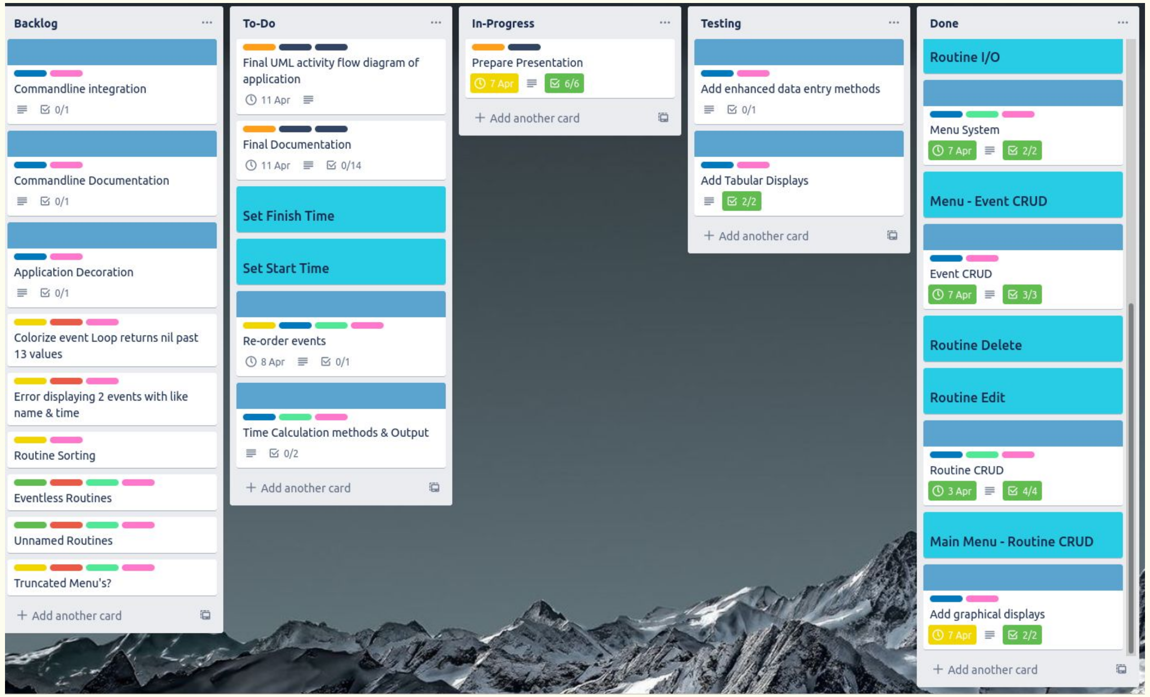
```ruby
  ###########      I/O      ###########
  def save_routines
    File.open(@file_path, 'w+') # Create new file with read/write permissions
    File.write(@file_path, @routines.to_json)
  end

  def load_data
    return unless File.exist?(@file_path)
    json_data = JSON.parse(File.read(@file_path), symbolize_names: true)
    json_data.each do |obj|
      @routines << Routine.new(
        obj[:name],
        obj[:events],
        obj[:total_time],
        obj[:start_time],
        obj[:finish_time])
    end
  end
end
```

Challenges

- File IO
  - A lot of trouble writing and reading objects back in, expected to take maybe 2hrs, ended up spending almost 2 days on it.
- Rspec
  - Started with TDD, however encountered difficulties with object creation routines early, and have put Rspec on hold.
- Colorize
  - Limited colour options

Still to come

- Tabulated Data Display (Events)
- Add ability to reorder events
- Time calculation methods
- Bug squashing
- Lots of tests!



Backlog

- Commandline integration 0/1
- Commandline Documentation 0/1
- Application Decoration 0/1
- Colorize event Loop returns nil past 13 values
- Error displaying 2 events with like name & time
- Routine Sorting
- Eventless Routines
- Unnamed Routines
- Truncated Menu's?

+ Add another card

To-Do

- Final UML activity flow diagram of application 11 Apr
- Final Documentation 11 Apr 0/14
- Set Finish Time
- Set Start Time
- Re-order events 8 Apr 0/1
- Time Calculation methods & Output 0/2

+ Add another card

In-Progress

- Prepare Presentation 7 Apr 6/6

+ Add another card

Testing

- Add enhanced data entry methods 0/1
- Add Tabular Displays 2/2

+ Add another card

Done

- Routine I/O
- Menu System 7 Apr 2/2
- Menu - Event CRUD
- Event CRUD 7 Apr 3/3
- Routine Delete
- Routine Edit
- Routine CRUD 3 Apr 4/4
- Main Menu - Routine CRUD
- Add graphical displays 7 Apr 2/2

+ Add another card