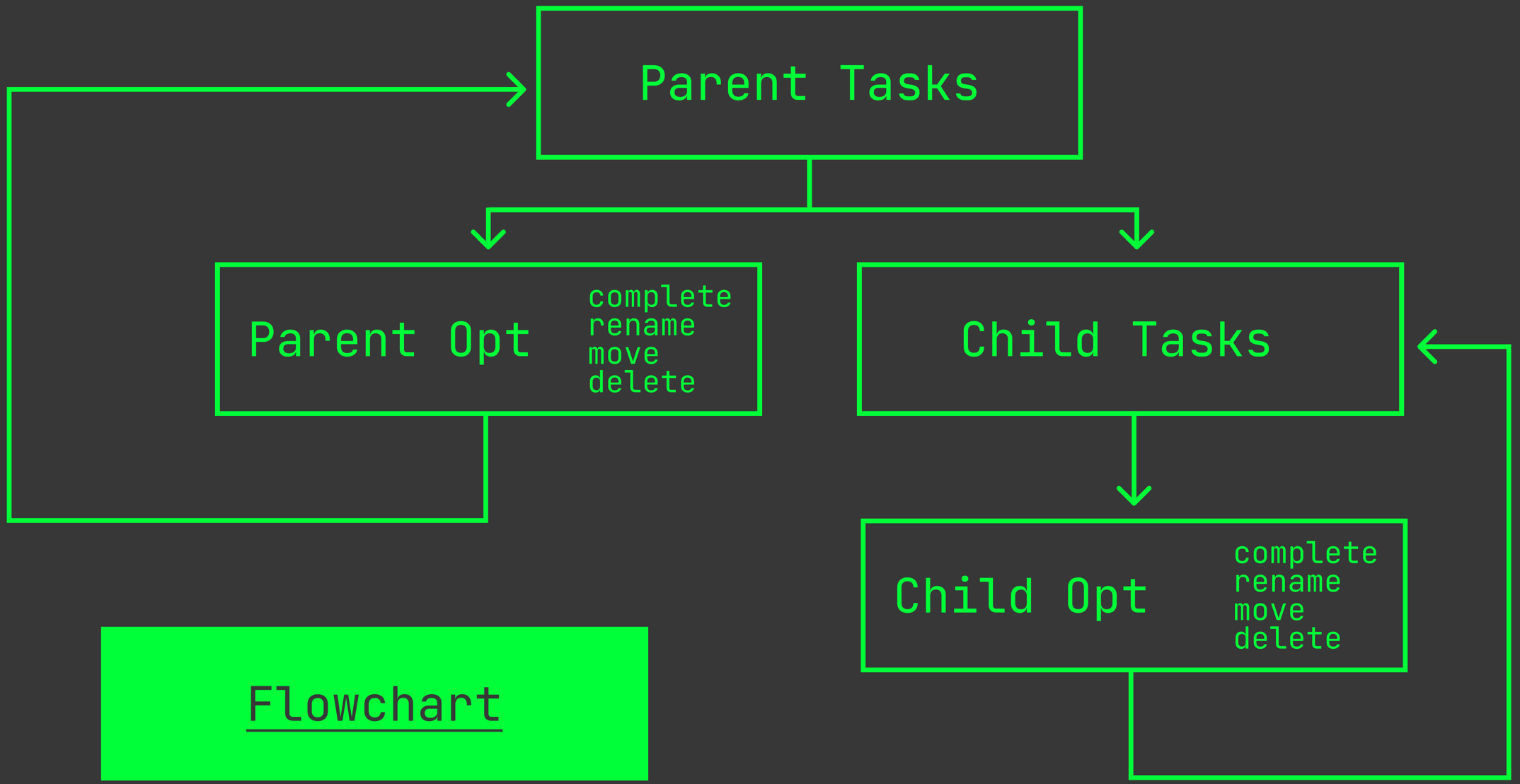
~ T1A3 - Terminal Application

→ by Stephen Smith

→ ~ Organisational Task Tracker [OTTR]

→ Goals:
    → track project specific tasks
    → based in repository
        → embrace cmd line workflow
    → fully functional from cmd line arguments...
    → optional user interface

→ ~ Structure


Parent Tasks
Parent Opt
complete
rename
move
delete
Child Tasks
Child Opt
complete
rename
move
delete
Flowchart

→ ~ Features

→ Add tasks & add children tasks
  → Check / uncheck tasks
  → Rename tasks
  → Re-arrange tasks
  → Delete tasks
  → ID based!!!
→ Auto-save and auto-load task list
→ Do all of the above from command line

```
→ ~
→ ~
→ ~
→ ~    d8888                      888       888          888 888      888    888                                                  888
      d88888                      888   o   888          888 888      888    888                                                  888
     d88P888                      888  d8b  888          888 888      888    888                                                  888
    d88P 888 88888b.  88888b.     888 d888b 888  8888b.  888 888  888 888888 88888b.  888d888 .d88b.  888  888  .d88b.  88888b.
   d88P  888 888 "88b 888 "88b    888d88888b888     "88b 888 888 .88P 888    888 "88b 888P"  d88""88b 888  888 d88P"88b 888 "88b
  d88P   888 888  888 888  888    88888P Y88888 .d888888 888 888888K  888    888  888 888    888  888 888  888 888  888 888  888
 d8888888888 888 d88P 888 d88P    8888P   Y8888 888  888 888 888 "88b Y88b.  888  888 888    Y88..88P Y88b 888 Y88b 888 888  888
d88P     888 88888P"  88888P"     888P     Y888 "Y888888 888 888  888  "Y888 888  888 888     "Y88P"   "Y88888  "Y88888 888  888
            888      888                                                                                    888
            888      888                                                                                Y8b d88P
                                                                                                         "Y88P"
```

→ ~ List Class

```
def initialize
    @tasks = []
end
```

```
def add_task(description)
  @tasks.push({
                'id' => unique_id,
                'description' => description,
                'is_complete?' => false,
                'is_parent?' => false,
                'is_selected?' => false,
                'child_tasks' => []
              })
end
```

```
def unique_id
  if @tasks.length > 0
    task_ids.max + 1
  else
    1
  end
end
```

```
def select_task(id)
  @tasks.each { |t| t['id'] == id ? t['is_selected?'] = true : nil }
end
```

```
def rename_task(name)
  @tasks[selected_task]['description'] = name
end
```

```
def reopen_task
  @tasks[selected_task]['is_complete?'] = false
end
```

```
def deselect_all_tasks
  @tasks.each do |t|
    t['is_selected?'] = false
    t['child_tasks'].each do |ct|
      ct['is_selected_child?'] = false
    end
  end
end
```

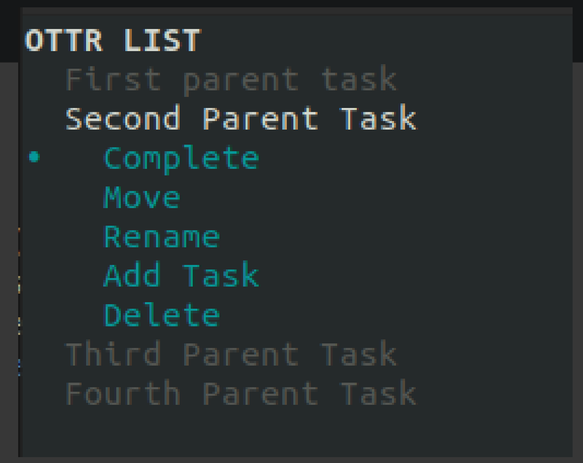
→ ~ Menu Class < SubMenu Class

```ruby
@PARENT_DEFAULTS = [
  { '  Complete'.colorize(:cyan) => :COMPLETE },
  { '  Move'.colorize(:cyan) => :MOVE },
  { '  Rename'.colorize(:cyan) => :RENAME },
  { '  Add Task'.colorize(:cyan) => :ADD },
  { '  Delete'.colorize(:cyan) => :DELETE }
]
@CHILD_DEFAULTS = [
  { '  Complete'.colorize(:cyan) => :COMPLETE },
  { '  Move'.colorize(:cyan) => :MOVE },
  { '  Rename'.colorize(:cyan) => :RENAME },
  { '  Delete'.colorize(:cyan) => :DELETE }
]
```

```ruby
@options = []
@loads = 0
```

```ruby
def populate_options(tasks, list)
  @options = tasks
  @options.insert(list.selected_task + 1, @DEFAULTS).flatten!
end
```

```ruby
def construct(list)
  system('cls') || system('clear')
  TTY::Prompt.new.select('OTTR LIST'.bold, symbols: { marker: '•', cross: ' ' }, active_color: :cyan)
do |menu|
    menu.default list.selected_task + list.selected_child_task + 2
    menu.per_page 20
    menu.help ''
    menu.choices @options
  end
end
```



→ ~ JSONHandler Module

```ruby
module JSONHandler
  def load_json(file)
    File.read(file)
  end

  def write_json(file, data)
    File.write(file, JSON.dump(data))
  end

  def json_array(file)
    JSON.parse!(load_json(file))
  end
end
```

→ ~ ProcessARGV Module

```ruby
module ProcessARGV
  @@init_status = File.exist?('./.ottr.json')

  def initialize_ottr
    if @@init_status == true
      puts 'already initialized'
    else
      List.new.write_tasks
      puts 'ottr initialized'
    end
  end

  def self.init_status
    @@init_status
  end
end
```

→ ~ RSPEC

```
describe List do
  context "When a list object's tasks contain 3 items:" do
    before(:each) do
      @list = List.new
      @list.instance_variable_set(:@tasks, [{ 'id' => 6,
                                              'description' => 'Do the laundry',
                                              'is_complete?' => false,
                                              'is_parent?' => false,
                                              'is_selected?' => false },
                                            {
                                              'id' => 8,
                                              'description' => 'Go for walk',
                                              'is_complete?' => false,
                                              'is_parent?' => false,
                                              'is_selected?' => false
                                            },
                                            {
                                              'id' => 4,
                                              'description' => 'Wash the car',
                                              'is_complete?' => false,
                                              'is_parent?' => false,
                                              'is_selected?' => false
                                            }])
    end
```

```
it 'should return array of hashes containing {task_description => id}' do
  expect(@list.list_task_descriptions).to eq([
                                               { 'Do the laundry' => 6 },
                                               { 'Go for walk' => 8 },
                                               { 'Wash the car' => 4 }
                                             ])
end
```

```
it 'it should return array of task ids' do
  expect(@list.task_ids).to eq([6, 8, 4])
end

it 'it should return a unique id that is 1 larger than the largest id' do
  expect(@list.unique_id).to eq(9)
end

it 'it should take two ids and swap their position in task list' do
  @list.move_task(4, 8)
  expect(@list.task_ids).to eq([6, 4, 8])
end
}
```

→ ~ Ethical
  → Colors?

→ ~ Challenges

  → Sub menus and child lists get repetitive
  → Rspec & TDD (loading external files)
  → Brainstorming = messy

→ ~ Gems

  → TTY Prompt → TTY Platform
  → Colorize → TTY Progress Bar

→ Thank you :]