TERMINAL APP

NATHAN O'DONNELL

RUBY POKEMON RUBY

- Nostalgia
- Was using Pokemon types to learn about classes

And inheritance

- Didn’t want to come up with app
    - No storyline
    - No ‘what should be included’
- Can always be expanded on
- Open to a long term project

STARTING FROM THE START

- Slowly print words to the screen

- Have to keep pressing a button to wipe the text and see more

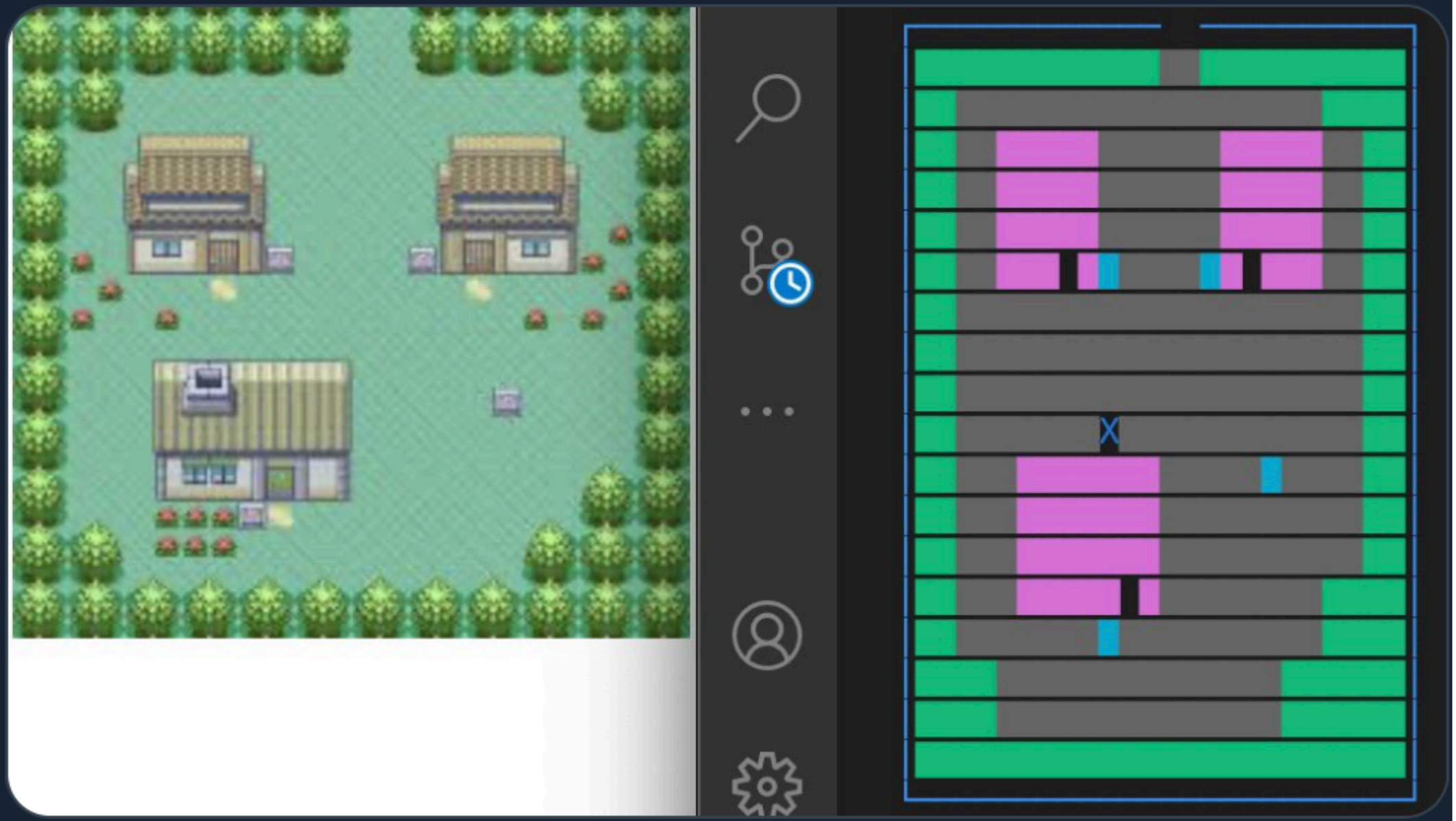
- Character creation

```
def slowly(str)
    str.each_char do |c|
        sleep 0.01
        print c
    end
    gets
end
```

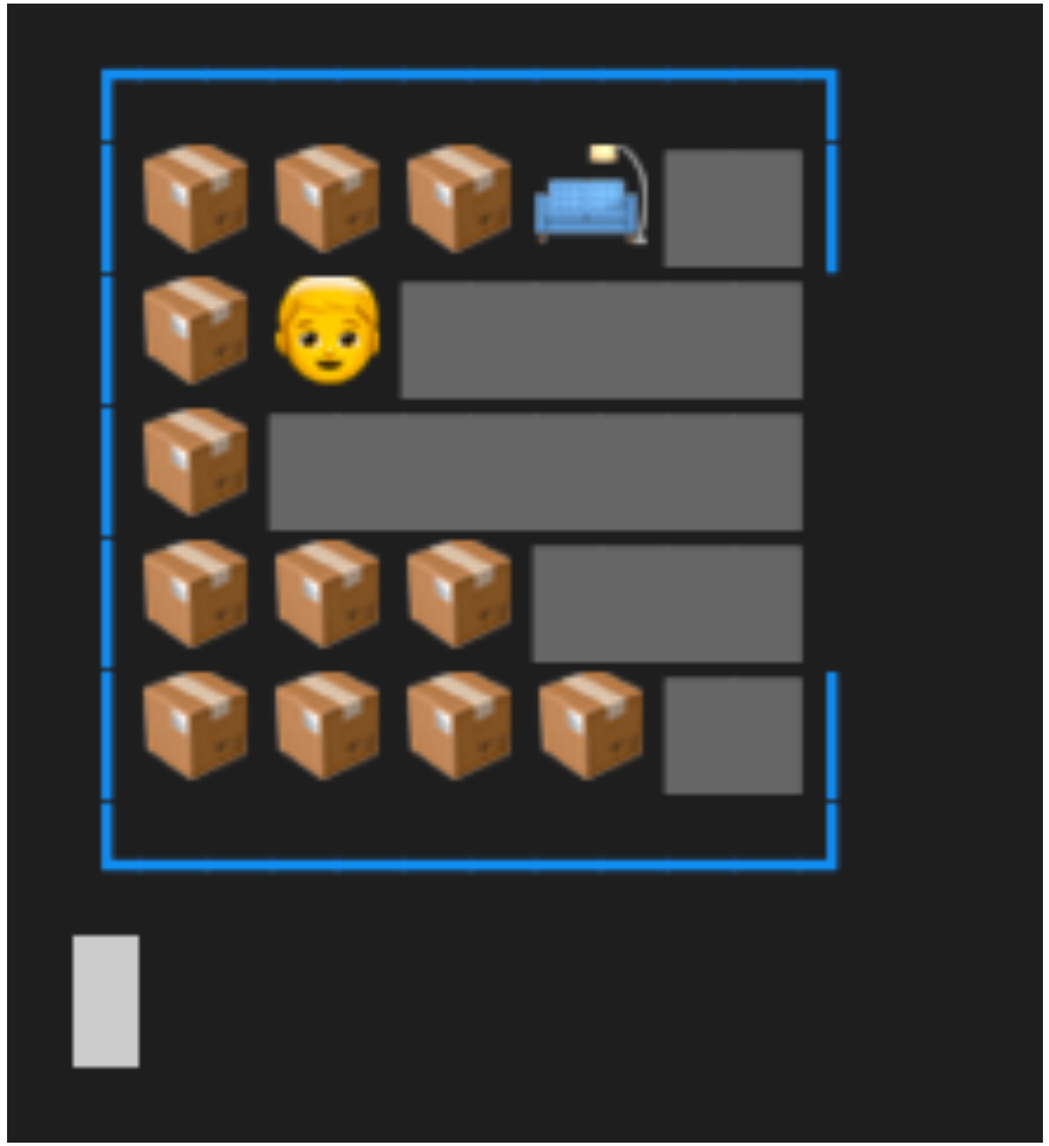
```ruby
name =
gender = ''
loop do
    slowly("Are you a boy? Or are you a girl?")
    gender = prompt.select("", %w(BOY GIRL))
    system('clear')
    if gender == 'BOY'
        slowly("All right. What's your name?")
        name = prompt.select("", %w(NEW LANDON TERRY SETH TOM))
    else
        slowly("All right. What's your name?")
        name = prompt.select("", %w(NEW TERRA KIMMY NICOLA SARA))
    end
    system('clear')
    if name == 'NEW'
        slowly("Enter your name")
        name = prompt.ask("") do |q|
            q.required true
            q.validate /^[a-zA-Z.,]+$/
            # /regex/ works like quotation marks for print; they encapsulate data
            # ^..$ is the start and stop of the regex
            # + means you can have more then one character matching
            # [a-zA-z] is checking for characters a..z + caps
            # ., allows the symbols
            q.modify :capitalize
        end
    end
    system('clear')
    slowly("So it's #{ name }?")
    check = prompt.select("", %w(YES NO))
    if check == 'YES'
        break
    end
end
player = Player.new(name, gender)
```

nathanodonnell @nathano64732109 · Sep 24

Look familiar? #ruby @CoderAcademy_au #100daysofcode

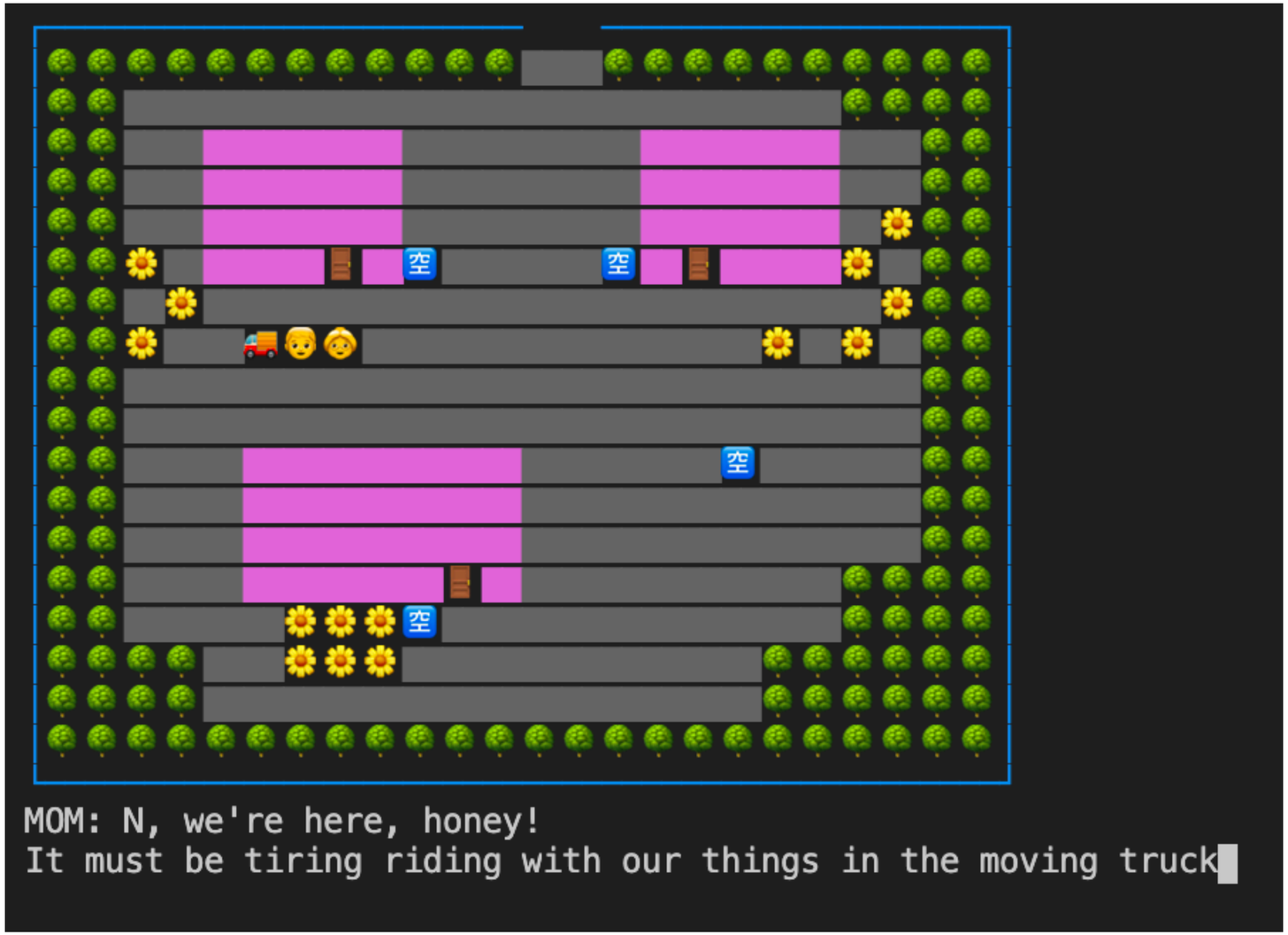


MAPS



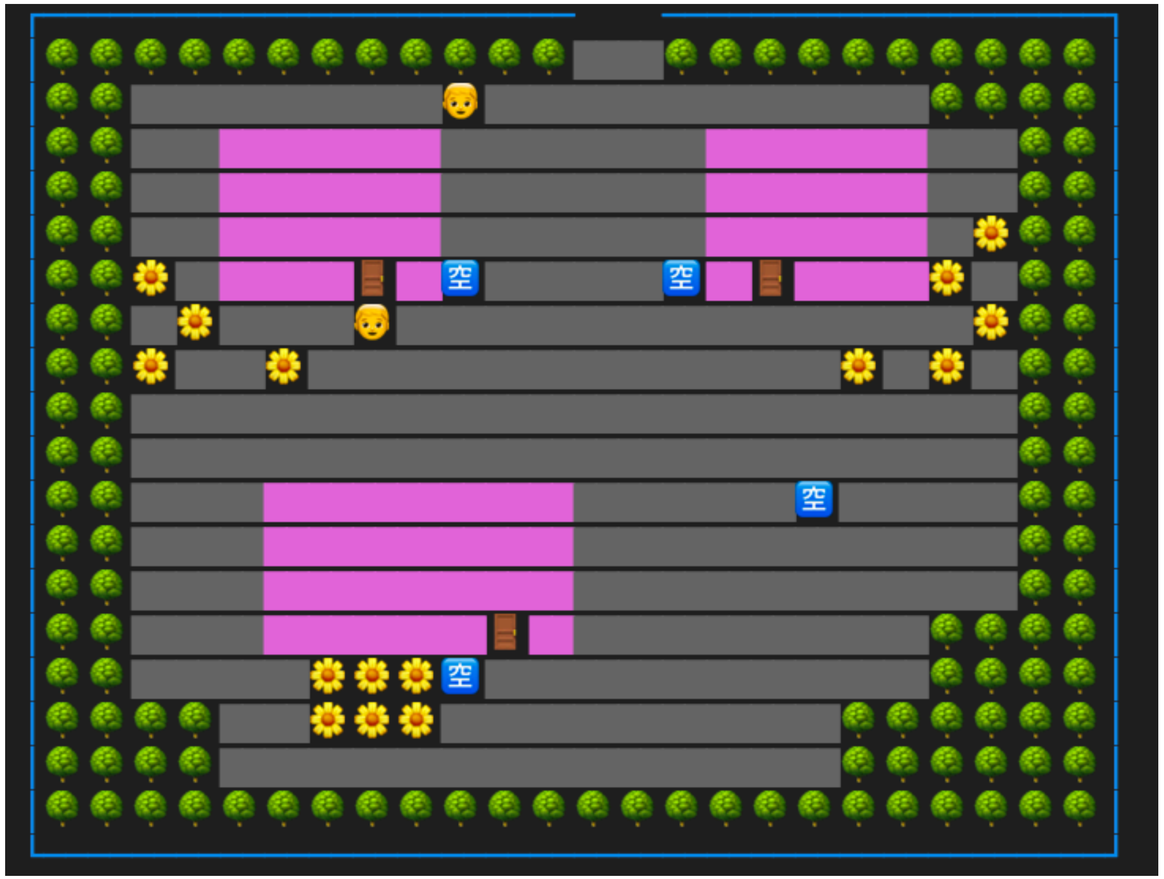
```
class LittleRoot < Map
    def initialize(player, x, y)
        @name='Littleroot'
        @map=[
    ['1','`','`','`','`','`','`','`','`','`','`','`','`','E','E','`','`','`','`','`','`','`','`','`','`','2',],
    ['|','T','T','T','T','T','T','T','T','T','T','T','T','T','S','S','T','T','T','T','T','T','T','T','T','T','|',],
    ['|','T','T','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','T','T','T','T','|',],
    ['|','T','T','S','S','H','H','H','H','H','S','S','S','S','S','S','S','H','H','H','H','H','S','S','T','T','|',],
    ['|','T','T','S','S','H','H','H','H','H','S','S','S','S','S','S','S','H','H','H','H','H','S','S','T','T','|',],
    ['|','T','T','S','S','H','H','H','H','H','S','S','S','S','S','S','S','H','H','H','H','H','S','flower','T','T','|',],
    ['|','T','T','flower','S','H','H','H','D','H','I','S','S','S','S','I','H','D','H','H','H','flower','S','T','T','|',],
    ['|','T','T','S','flower','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','flower','T','T','|',],
    ['|','T','T','flower','S','S','flower','S','S','S','S','S','S','S','S','S','S','S','S','S','S','flower','S','flower','S','T','T','|',],
    ['|','T','T','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','T','T','|',],
    ['|','T','T','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','T','T','|',],
    ['|','T','T','S','S','S','H','H','H','H','H','H','H','S','S','S','S','S','I','S','S','S','S','T','T','|',],
    ['|','T','T','S','S','S','H','H','H','H','H','H','H','S','S','S','S','S','S','S','S','S','S','T','T','|',],
    ['|','T','T','S','S','S','H','H','H','H','H','H','H','S','S','S','S','S','S','S','S','S','S','T','T','|',],
    ['|','T','T','S','S','S','H','H','H','H','H','D','H','S','S','S','S','S','S','S','S','S','T','T','T','T','|',],
    ['|','T','T','S','S','S','S','flower','flower','flower','I','S','S','S','S','S','S','S','S','S','S','S','T','T','T','T','|',],
    ['|','T','T','T','T','S','S','flower','flower','flower','S','S','S','S','S','S','S','S','S','S','T','T','T','T','T','T','|',],
    ['|','T','T','T','T','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','S','T','T','T','T','T','T','|',],
    ['|','T','T','T','T','T','T','T','T','T','T','T','T','T','T','T','T','T','T','T','T','T','T','T','T','T','T','|',],
    ['3','`','`','`','`','`','`','`','`','`','`','`','`','`','`','`','`','`','`','`','`','`','`','`','`','4',],
    ]
    @pos_x=x
    @pos_y=y
    @saved_variable='S'
    @player=player
    @player_icon=@player.player_icon
    @map[@pos_x][@pos_y] = "X"
    @talk = 0
    end
```

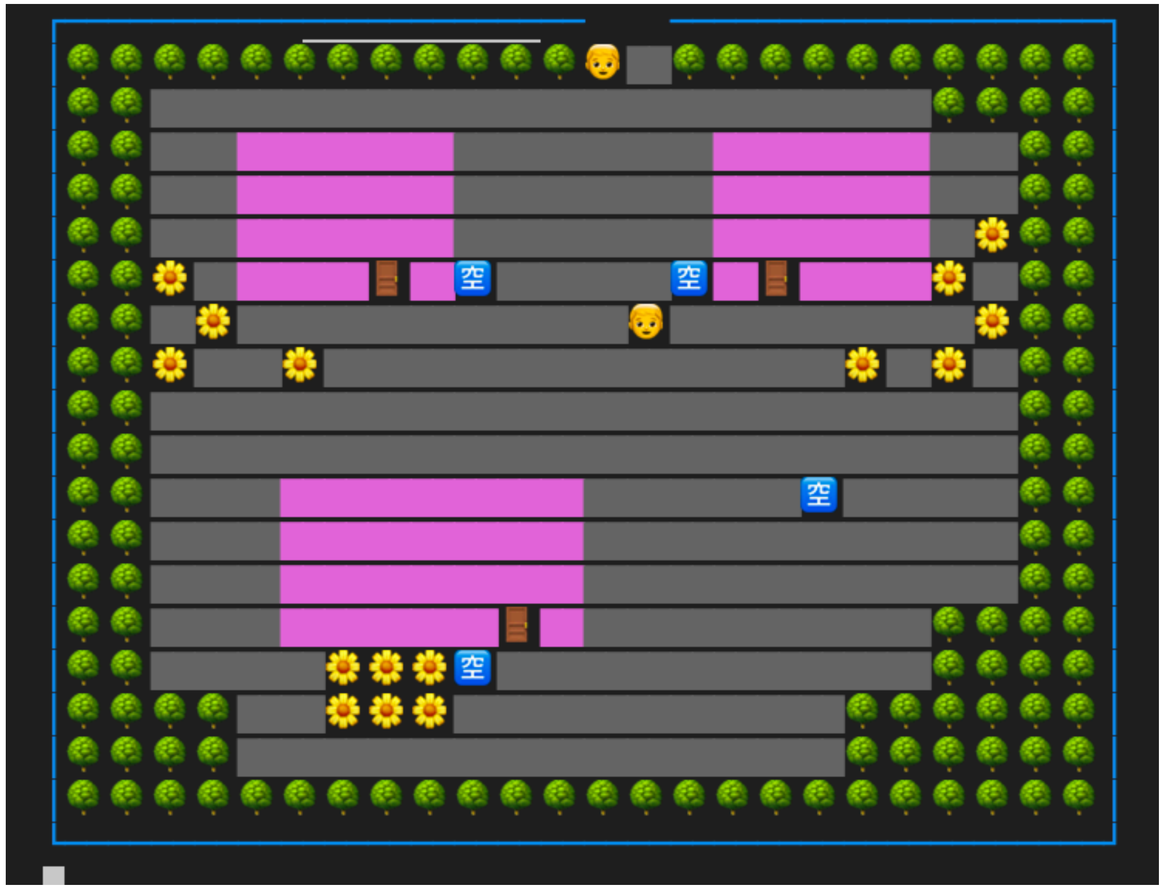
ANIMATION


空
空
空
空
MOM: N, we're here, honey!
It must be tiring riding with our things in the moving truck

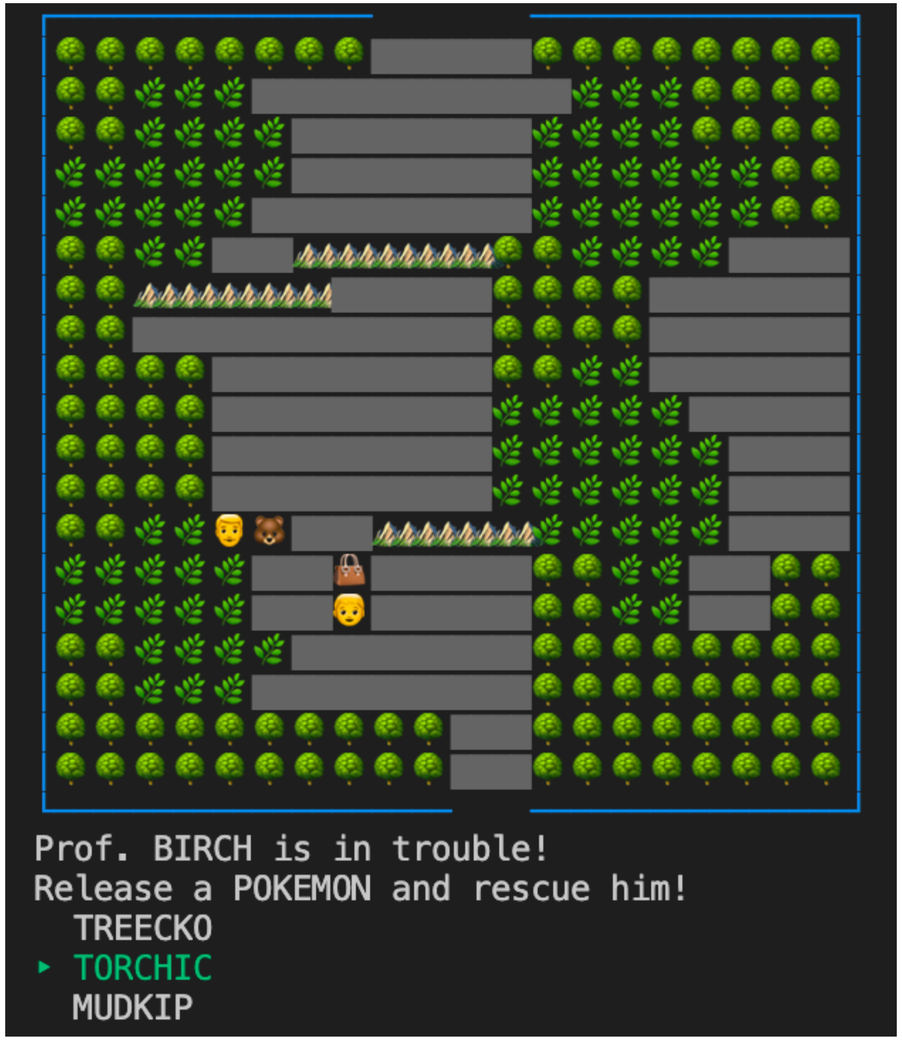
```
def time_setup
    case @player.littleroot
    when 'first' #Places car, mum animation and forces player to next map
        @map[8][6] = 'C'
        print_map
        @map[7][8] = 'lady'
        sleep 1
        print_map
        @map[7][8] = 'S'
        @map[8][8] = 'lady'
        sleep 1
        print_map
        slowly("MOM: #{ @player.name }, we're here, honey!")
        slowly("It must be tiring riding with our things in the moving truck")
        reset_map
```

ONE MAP... THREE TIMES

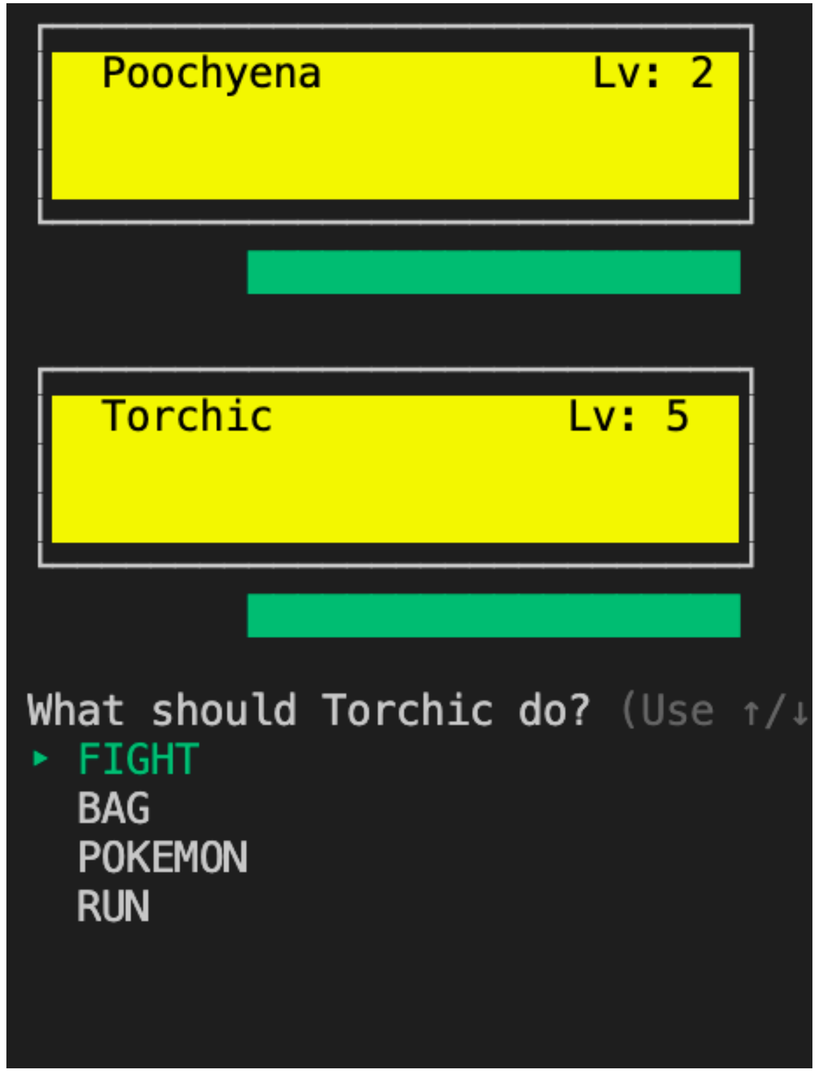


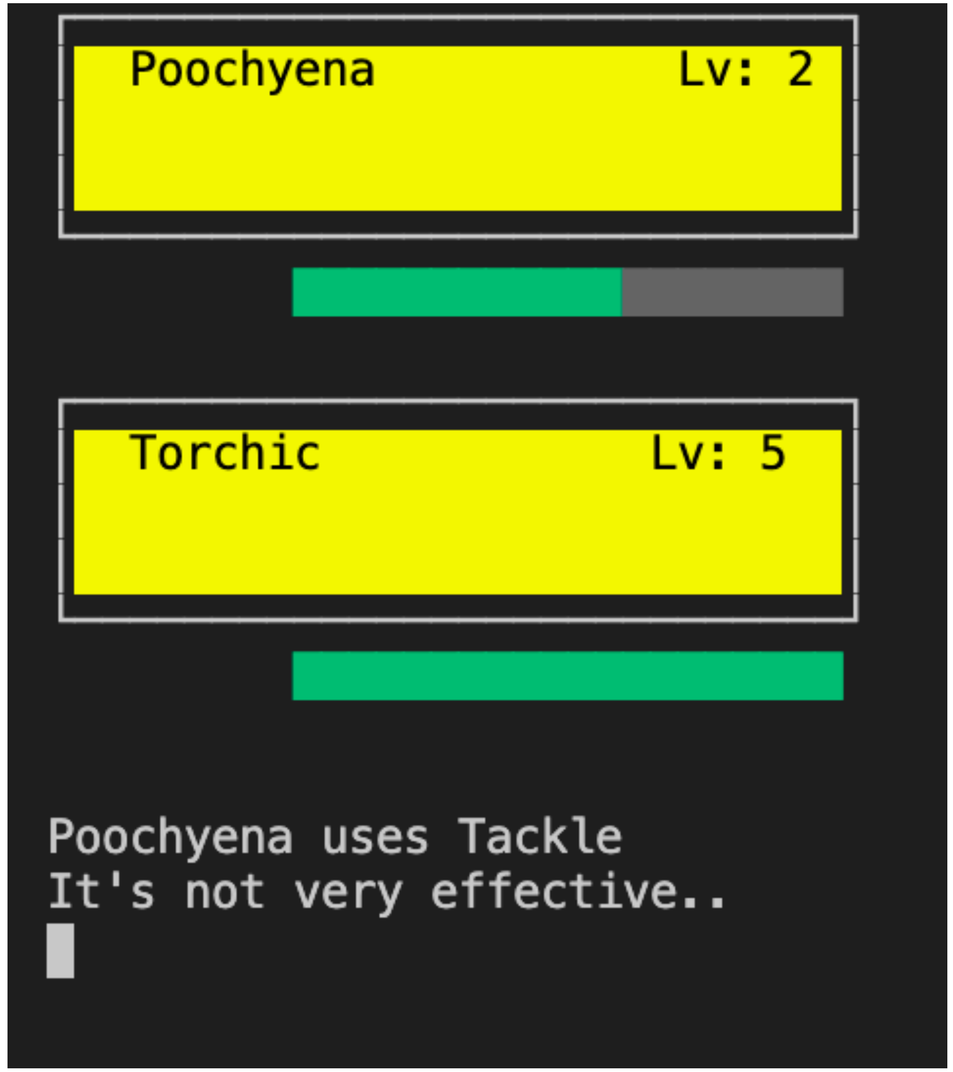


CHOOSE A POKEMON

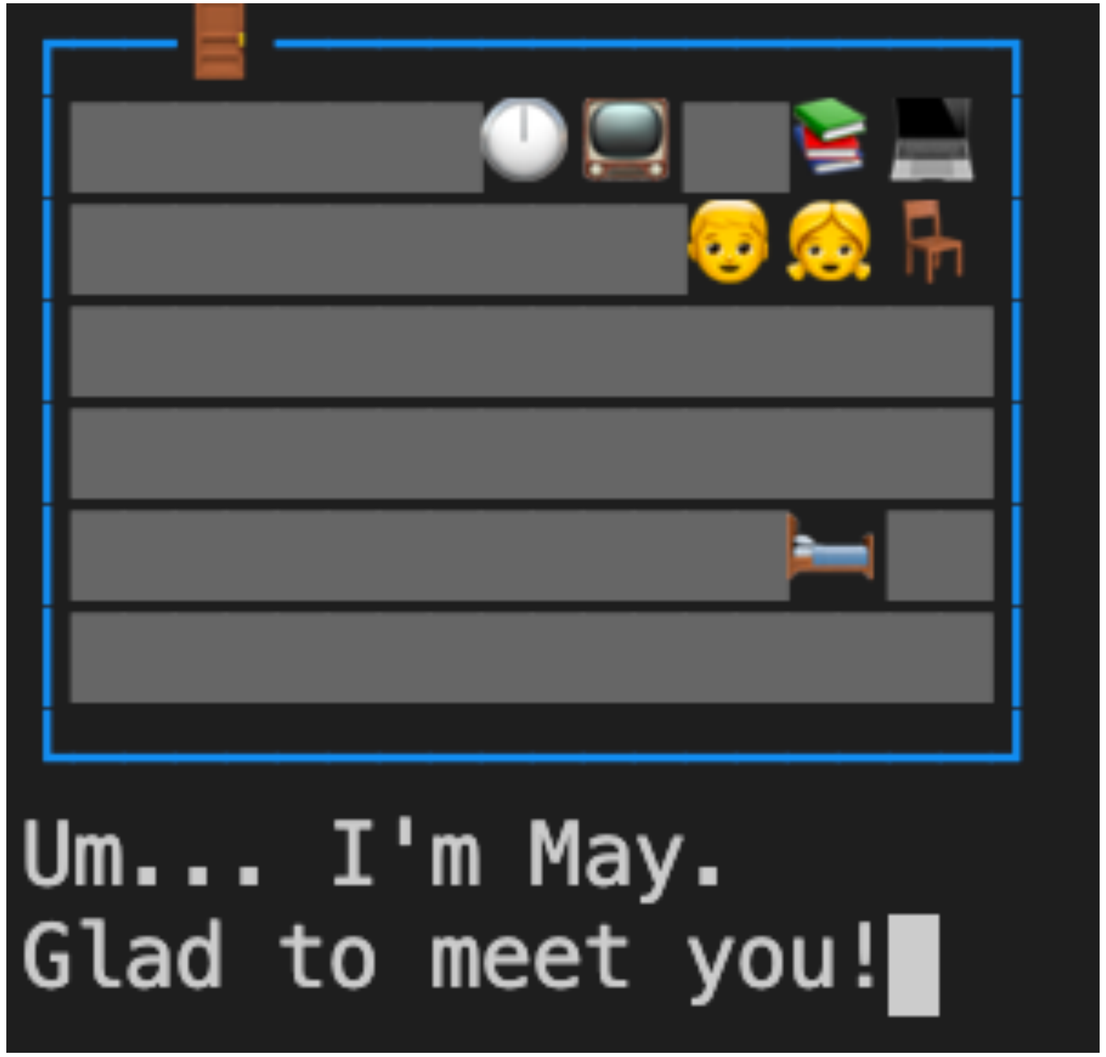


BATTLE


Poochyena Lv: 2
Torchic Lv: 5
What should Torchic do? (Use ↑/↓
▸ FIGHT
BAG
POKEMON
RUN


Poochyena Lv: 2
Torchic Lv: 5
Poochyena uses Tackle
It's not very effective..

DYNAMIC PLAY



TO DO

- Error Test
- Bundle
- Applicationise
- Start over?

WHAT NOW?

- Easy elements to incorporate: Different pokemon, Random chance of finding them

- Challenges: Saving, Time, Limit on pokemon on hand/Computer storage

- Hard: Creating the maps and animation, using all the different features of pokemon battling and capture rates

> More precisely, damage is calculated as
>
> $$Damage = \left(\frac{\left(\frac{2 \times Level}{5} + 2 \right) \times Power \times A/D}{50} + 2 \right) \times Modifier$$
>
> where
>
> - *Level* is the level of the attacking Pokémon (or twice the level for a critical hit in Generation I).
> - *A* is the effective Attack stat of the attacking Pokémon if the used move is a physical move, or the effective Special Attack stat of the attacking Pokémon if the used move is a special move (ignoring all[Gen. II]/negative[Gen. III+] stat stages for a critical hit).
> - *D* is the effective Defense stat of the target if the used move is a physical move or a special move that uses the target's Defense stat, or the effective Special Defense of the target if the used move is an other special move (ignoring all[Gen. II]/positive[Gen. III+] stat stages for a critical hit).
> - *Power* is the effective power of the used move.
>
> and *Modifier* is
>
> $$Modifier = Targets \times Weather \times Badge \times Critical \times random \times STAB \times Type \times Burn \times other$$
>
> where