Terminal App

Ruby
Terminal
App
Presentation

Leeroy vs The World

The app is inspired by Pokemon and D&D. It has a pokemon style battle menu with options for 'run' and 'attack' however damage is delt similar to D&D with a dice roll.

The player plays as my dog Leeroy who fights enemies who enter his territory(my house)

Leeroy

Features

on the app

- **Dice Roll**: I have installed a ruby gem to handle the dice rolling. When the player and enemy attack, a dice will automaticaly be rolled in which the number it lands on is the amount of damange delt.

- **Stats**: The player can see what Hp they are on as well as the enemy.

- **Battle Menu:** This is a menu created with TTY gem which will help the player choose options they would like to do (Attack, Run).

- **Main Menu**: This is the first menu the user will see and again has been made with TTY.

- **Story flow:** The story was written and implemented into the code to create a background and a sense of purpose as to why you are fighting.

Sprinkles

- Features i would like to add time permitted

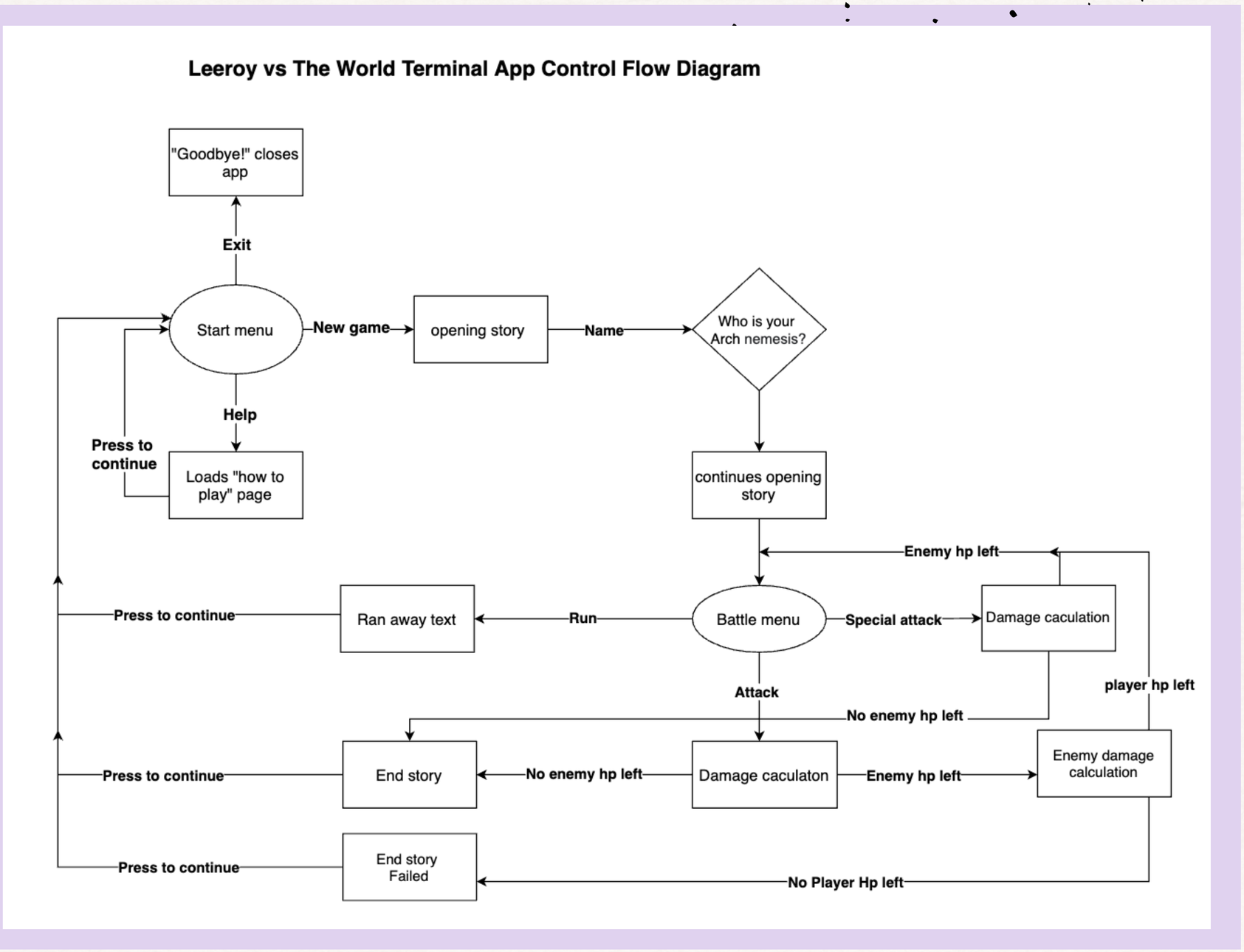
- **Block feature:** Block action where dice is rolled to see if it is a miss or succeed. If miss goes back to the battle
menu if succeed you block an incoming attack.

- **Special Attacks:** Instead of just having the one attack having a 'special attack' for a higher chance of damage given for both classes. (I have just added this in)

- **More levels:** At the moment there is only one enemy I would like to create more which go up in difficulty.

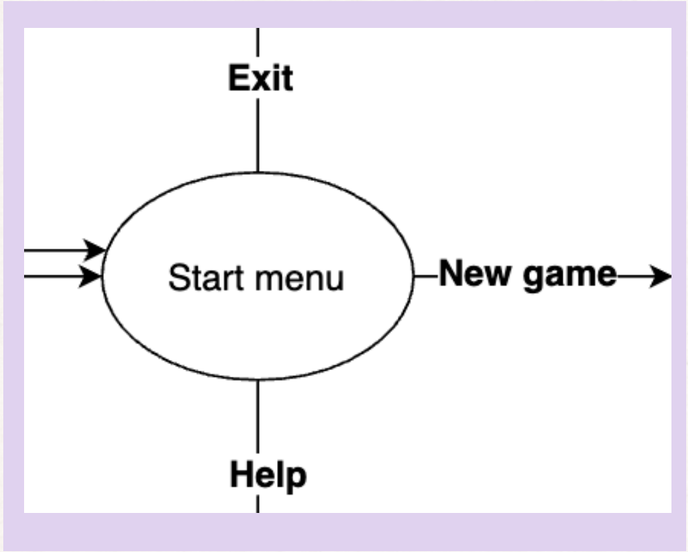
- **Battle music:** If I can find a gem to help with this I will add it in.

Logic Walkthrough



Control flow diagram as a whole. It contains multiple loops. first loop for the main menu, second loop for battle choice of run or attack. Then a third loop for the attack option.

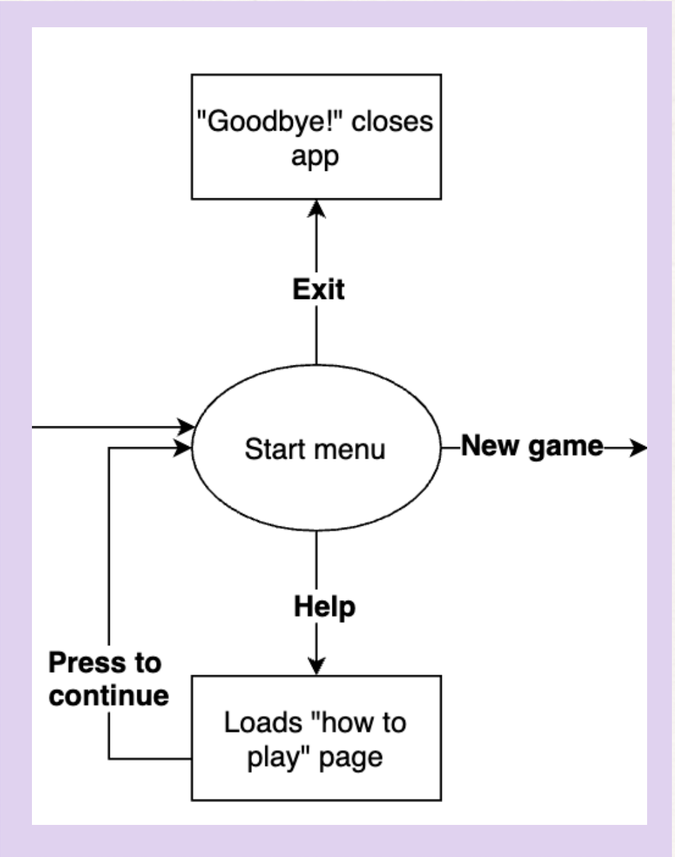
Main menu Loop



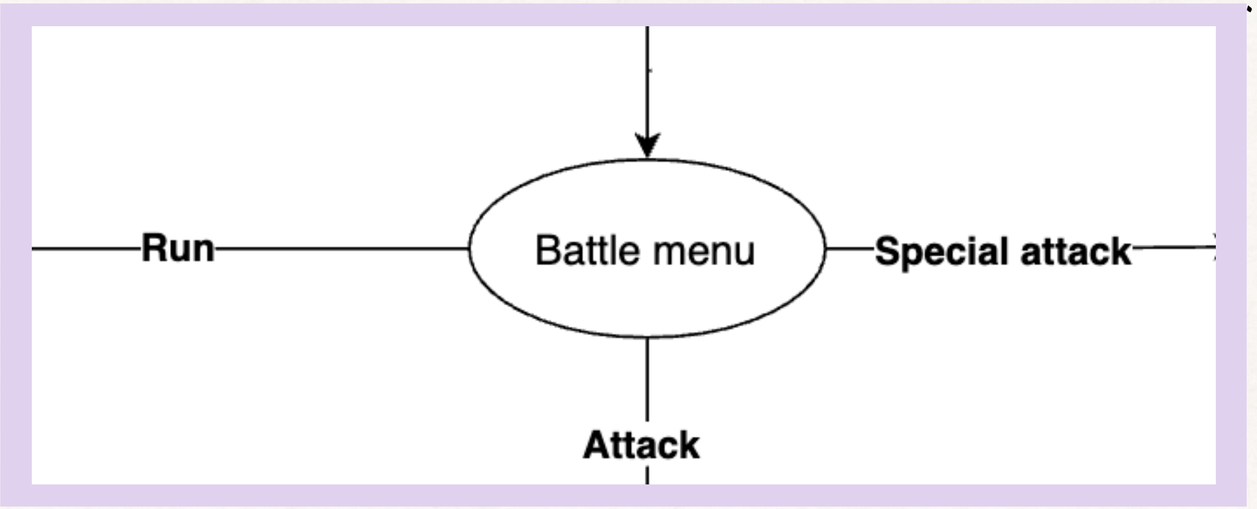
The first loop is that of the start menu. Giving the user three options to choose from. Depending on what they choose will result in exiting the game, starting a new game or getting info on how to play the game.

When on the Help it loads the page with a "press to continue" prompt at the bottom which will bring the user back to the start menu

Exit, exits the program all together



Battle menu loop



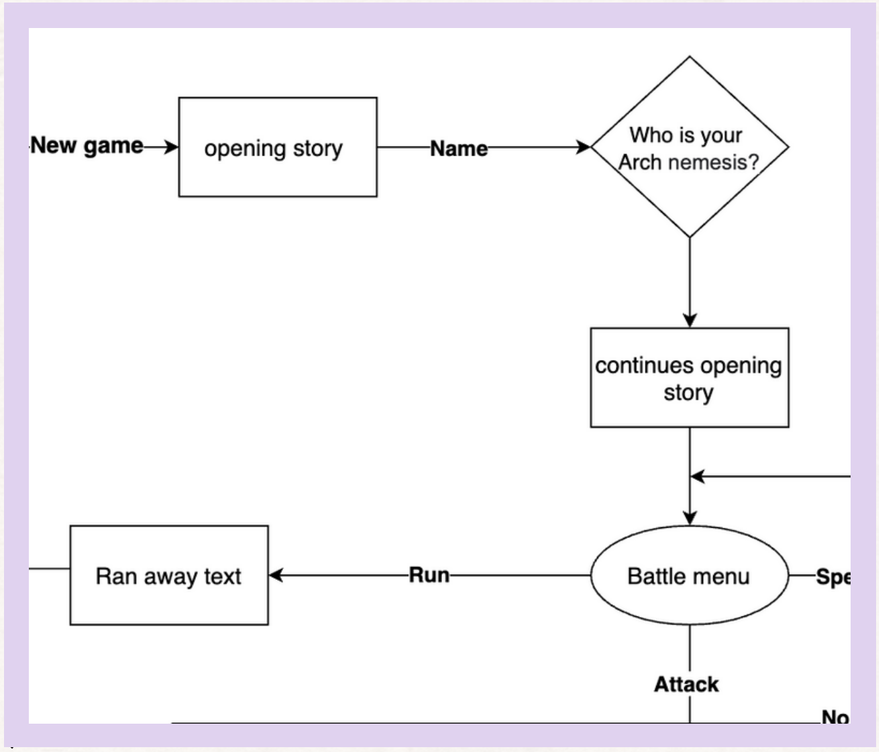
The second loop is that of the battle menu which has three options:

Run
Special Attack
Attack

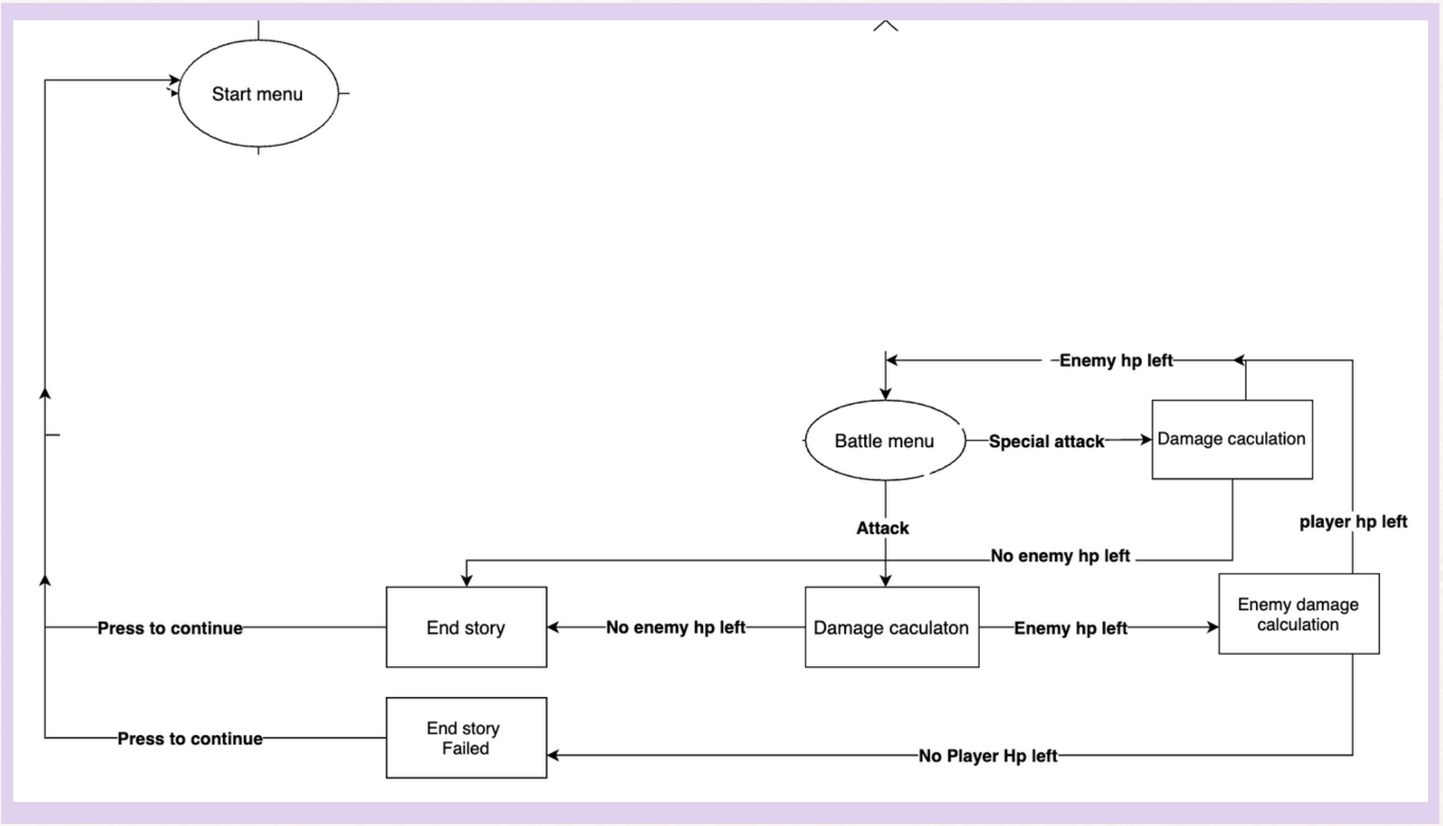
Run will exit the scenario write show some script and then will return the user back to the main menu.

Special Attack, is an attack the user can only use once, as the dice gem will roll two dice for a chance of higher damage. If the user selects to use it again an error message pops up.

Attack will attack the enemy and can be used until the user or enemy has 0 hp.



Battle menu loop cont



The third loop shows, depending on what the user selects will loop.

All loops will eventually go back to the start menu once the battle is over. This is determined by hp, when the user to enemy reaches 0 the battle will end. Depending on how who lost is what branch of story-line they will receive

```ruby
#Loop for the main menu
answer = ""
while answer != "Exit"
    answer = main_menu
    system "clear"
    case answer
    when "New Game!"
        player = Player.new(30, "1d10") #created player class with their HP and Attack dice used
        name = opening_story
        enemy = Enemy.new(30, "1d8", name) #created enemy class with its hp and attack dice used
        battle = ""
        special = ""
        while battle != "Run!"
            battle = battle_menu
            case battle
            when "Attack"
                damage = player.attack
                enemy.take_damage(damage)
                if enemy.hp > 0
                    puts "#{enemy.name} has taken #{damage} damage they have #{enemy.hp} hp left"
                else
                    victory_story(enemy.name)
                    press_continue
                    battle = "Run!"
                end
                damage = enemy.attack
                player.take_damage(damage)
                if player.hp > 0
                    page_break
                    sleep(1)
                    puts "You take #{damage} damage, you now have #{player.hp} hp left"
                else
                    fail_story(enemy.name)
                    press_continue
                    battle = "Run!"
                end
            when "Special Attack!!"
                damage = player.special_attack
                enemy.take_damage(damage)
                if special == "selected"
                    page_break
                    puts "You have already used your special, select another option".upcase
                elsif enemy.hp > 0 && special != "selected"
                    page_break
                    puts "#{enemy.name} has taken #{damage} special damage they have #{enemy.hp} hp left"
                    puts "#{enemy.name} is now stunned"
                    special = "selected"
                else
                    victory_story(enemy.name)
                    press_continue
                    battle = "Run!"
                end
            when "Run!"
                puts "Leeroy hears his name being called and uses that as an excuse to run away. "
                page_break
                press_continue
                next
            end
        end
    when "Help"
        help_text
        press_continue
    else
        page_break
        puts "GoodBye! ᕦ( ͡° ͜ʖ ͡°)ᕤ "
        page_break
        next
    end
    system "clear"
end
```

The Loop nest

```ruby
#Loop for the main menu, battle menu, attack, and run.
answer = ""
while answer != "Exit"
    answer = main_menu
    system "clear"
    case answer
    when "New Game!"
        player = Player.new(30, "1d10") #created player class with their HP and Attack dice used
        name = opening_story
        enemy = Enemy.new(30, "1d8", name) #created enemy class with its hp and attack dice used
        battle = ""
        special = ""
        while battle != "Run!"
            battle = battle_menu
            case battle
            when "Attack"
                damage = player.attack
                enemy.take_damage(damage)
                if enemy.hp > 0
                    puts "#{enemy.name} has taken #{damage} damage they have #{enemy.hp} hp left"
                else
                    victory_story(enemy.name)
                    press_continue
                    battle = "Run!"
                end
                damage = enemy.attack
                player.take_damage(damage)
                if player.hp > 0
                    page_break
                    sleep(1)
                    puts "You take #{damage} damage, you now have #{player.hp} hp left"
                else
                    fail_story(enemy.name)
                    press_continue
                    battle = "Run!"
```

Loop nest snippet

From the loop snippet, you can see i used a lot of methods to keep the code neat and easy to read.

The most important part of my game was setting up the player and enemy class which help methods for "attack, special attack & run"

```ruby
#method for page breaks to make blocks of text look neat
def page_break
    puts ""
end


# Body of text for story
def opening_story

    puts "It was a warm summers day and Leeroy the sausage dog was sniffing the kitchen floor for pieces of food his hoomans might have dropped. He sauntered past the open kitchen door when suddenly movement in his peripherals made him stop in his tracks. His tail went into shark mode as his fur began to prickle, outside on his lawn he spotted a trespasser. A lizard, sunbaking in the grass! How dare it!"

    pastel = Pastel.new
    puts pastel.red("What is your archnemesis name?")
    name = gets.chomp.to_s.upcase
    empty_name_string
    page_break

    puts "Leeroy's eyes squint as he sizes up the lizard, he can just make out that the lizard is wearing a collar with the word #{name}. His eyes narrow as #{name} slowly turns to stare back at Leeroy, Leeroy lets out a warning growl as he takes a step forward. #{name} the lizard takes a challenging step forward, not wanting to give up his sunbaking spot."
    return name
end

#text if player hp is 0
def fail_story(name)…
end

#text is enemy hp is 0
def victory_story(name)…
end

#text for when enemy attacks
def enemy_attack(name)…
```

Methods

```ruby
require_relative './battle'
require 'games_dice'

class Player
    attr_reader :hp
    def initialize (hp,dice)
        @hp = hp
        @dice = dice
    end

    def attack
        dice = GamesDice.create @dice
        return dice.roll
    end

    def special_attack
        dice = GamesDice.create @dice
        return dice.roll + dice.roll
    end

    def take_damage(i)
        @hp -= i
    end

end


class Enemy < Player
    attr_reader :name
    def initialize(hp, dice, name)
        super(hp, dice)
        @name = name
    end
end
```

P Class
E Class

```ruby
def attack
    dice = GamesDice.create @dice
    return dice.roll
end
```

Dice gem

Dice gem was very important as that created what damage was received both by player and enemy.

I had methods for every part of the story so when adding it to the loop it would look neat.

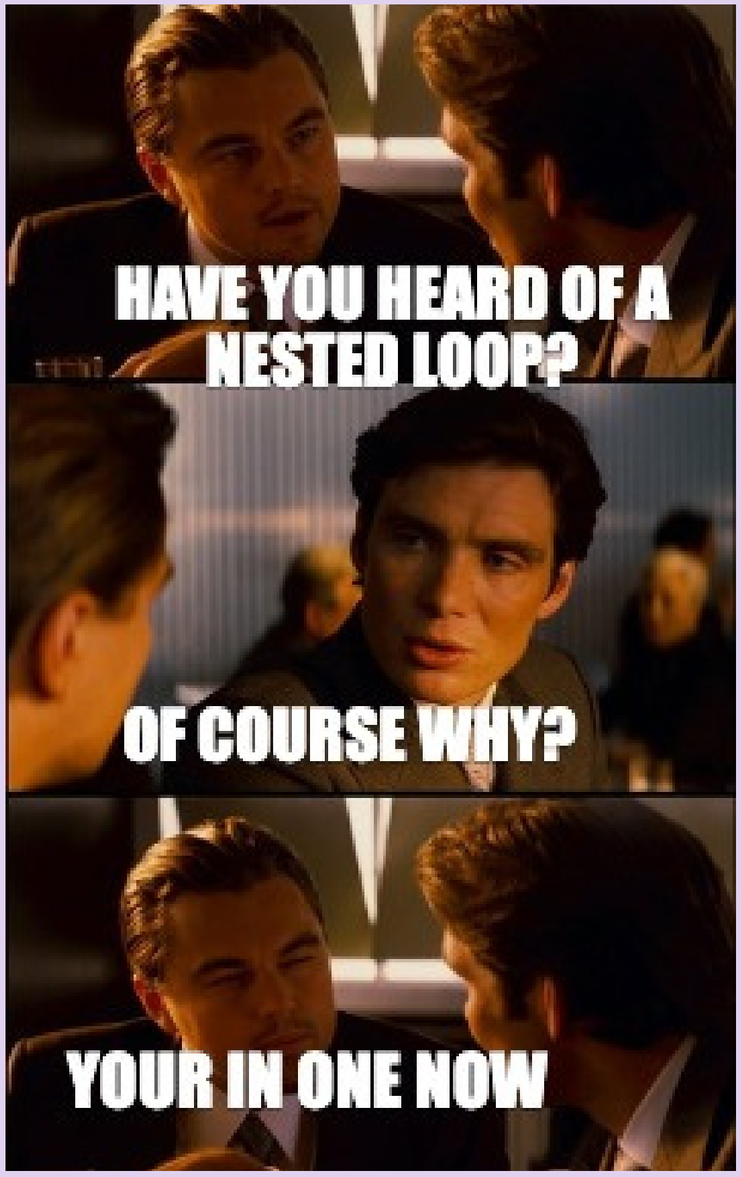
The player class and enemy class were super important as they held all the methods to make the game playable.

Challenges

Loop-ception: I have about 4 nested loops.

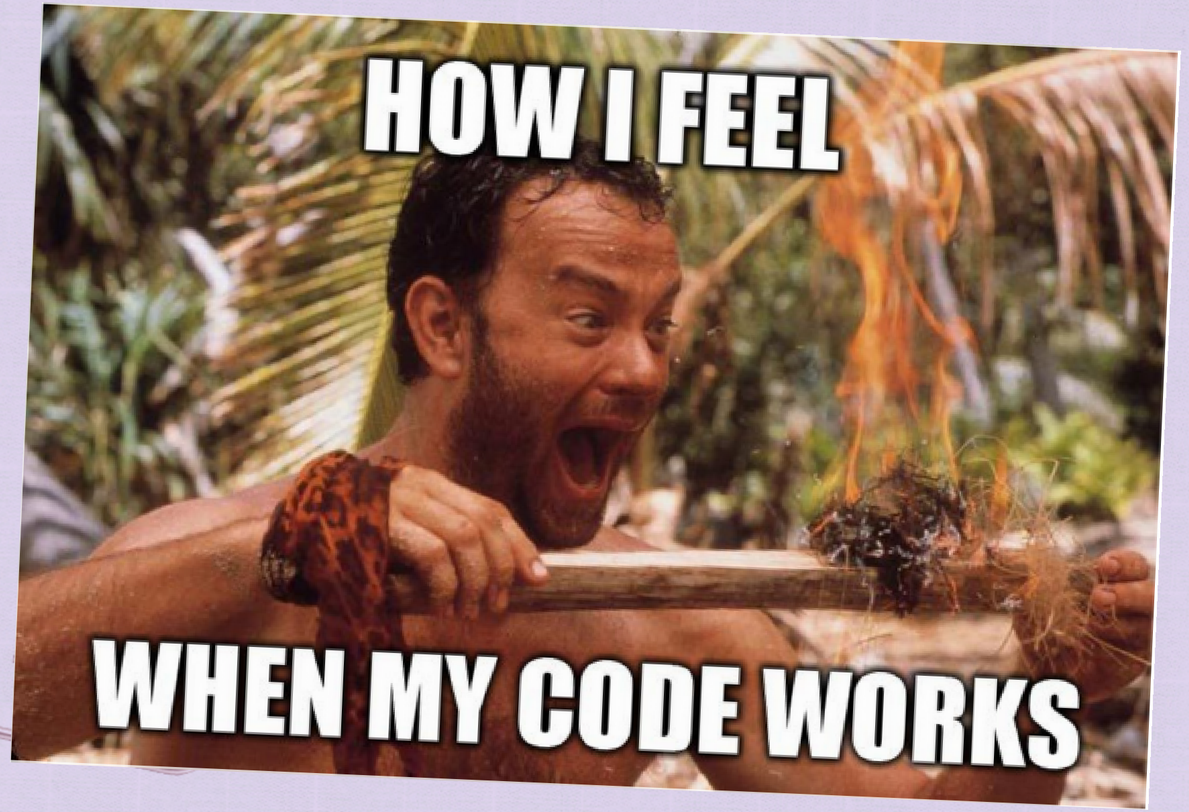
Syntax: knowing what I wanted to write than finding out the correct way on doing so.

Naming: giving objects and classes relevant names.



Enjoyable parts

Finding and using ruby gems!



Thank you!

Have a great day ahead.