TIA3 -Terminal Application

Zop

A Ruby Quiz Game

By Nathan Blaga

What Is Zop?

- Zop is a Trivia/Quiz game
- Consists of 3 levels
- Each level has 30 unique questions
- The difficulty increases with each level

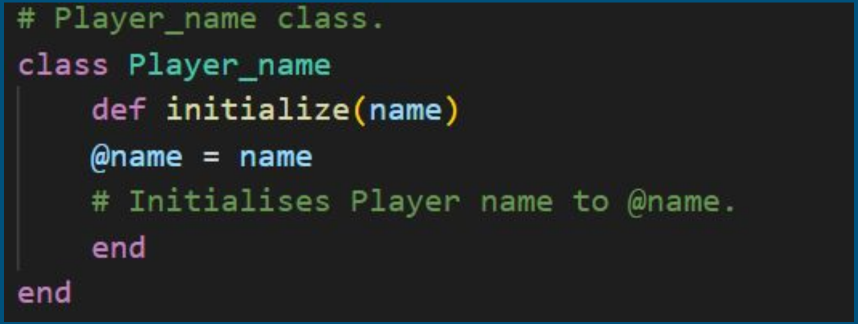
- The player must answer 10 correct questions to beat the level
- If they answer 5 incorrect answer, it’s game over
- All questions are MCQS
- Each question only contains 1 correct answer

Main Features

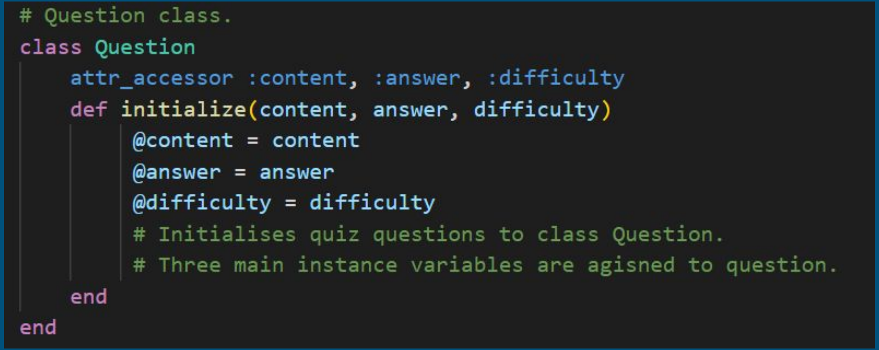
- Menu System
- Level Selection
- Level Difficulty

- Rules/Instruction menu
- Displays player progression after each question, including:
  - Number of correct answers
  - Attempts remaining
  - The correct answer

Logic Behind Code

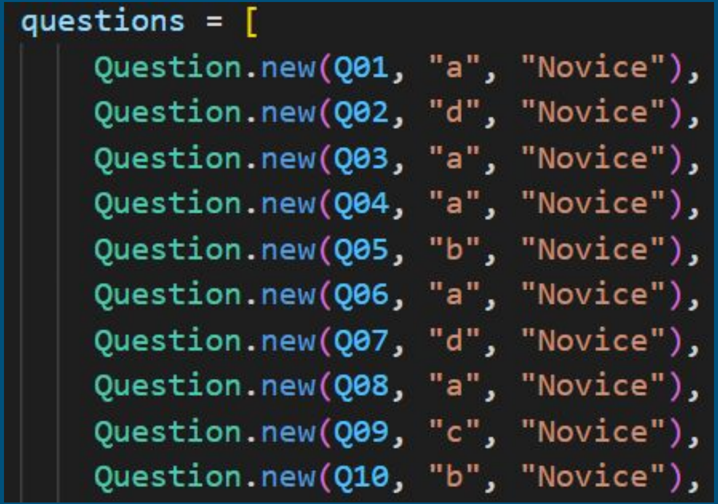


```ruby
# Player_name class.
class Player_name
    def initialize(name)
    @name = name
    # Initialises Player name to @name.
    end
end
```

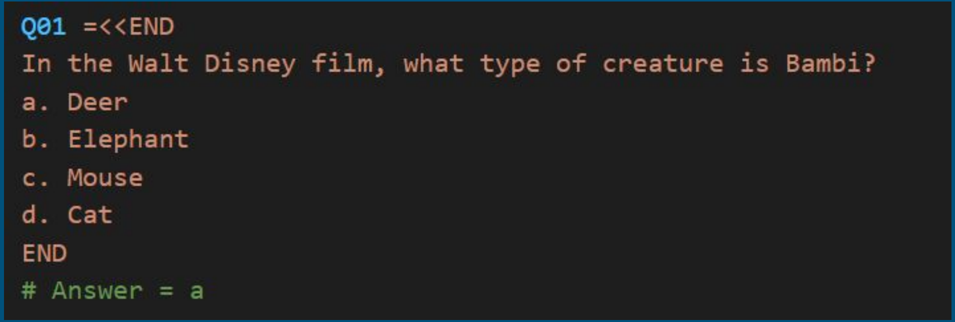


```ruby
# Question class.
class Question
    attr_accessor :content, :answer, :difficulty
    def initialize(content, answer, difficulty)
        @content = content
        @answer = answer
        @difficulty = difficulty
        # Initialises quiz questions to class Question.
        # Three main instance variables are agisned to question.
    end
end
```

Logic Behind Code

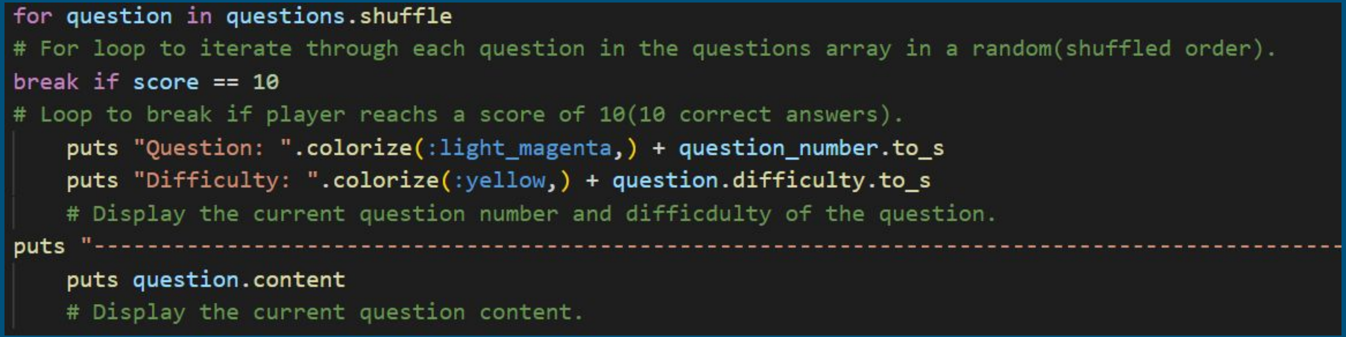


```
questions = [
        Question.new(Q01, "a", "Novice"),
        Question.new(Q02, "d", "Novice"),
        Question.new(Q03, "a", "Novice"),
        Question.new(Q04, "a", "Novice"),
        Question.new(Q05, "b", "Novice"),
        Question.new(Q06, "a", "Novice"),
        Question.new(Q07, "d", "Novice"),
        Question.new(Q08, "a", "Novice"),
        Question.new(Q09, "c", "Novice"),
        Question.new(Q10, "b", "Novice"),
```



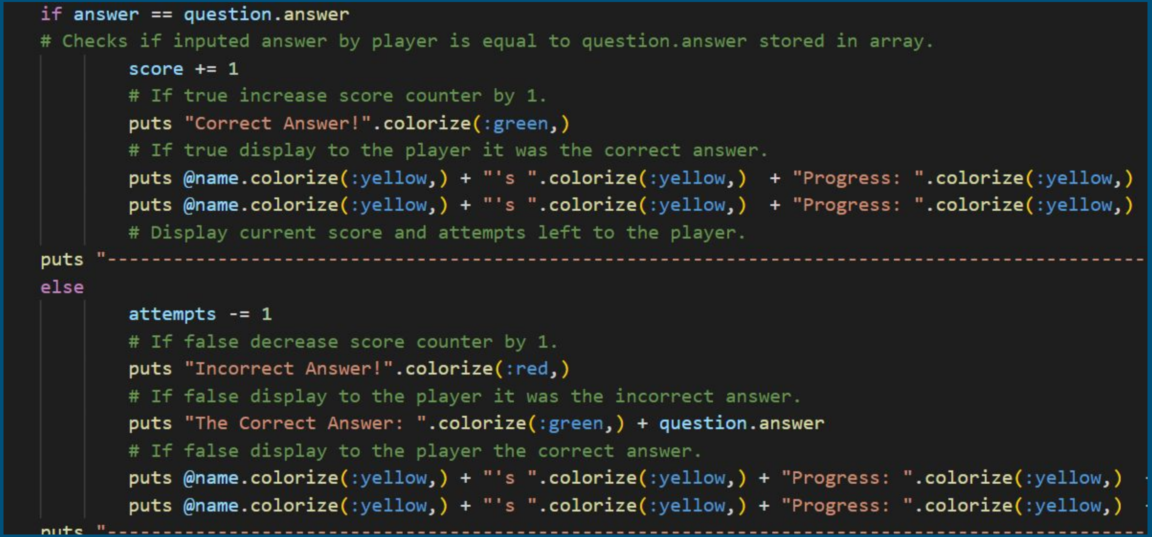
```
Q01 =<<END
In the Walt Disney film, what type of creature is Bambi?
a. Deer
b. Elephant
c. Mouse
d. Cat
END
# Answer = a
```

Logic Behind Code



```
for question in questions.shuffle
# For loop to iterate through each question in the questions array in a random(shuffled order).
break if score == 10
# Loop to break if player reachs a score of 10(10 correct answers).
    puts "Question: ".colorize(:light_magenta,) + question_number.to_s
    puts "Difficulty: ".colorize(:yellow,) + question.difficulty.to_s
    # Display the current question number and difficdulty of the question.
puts "----------------------------------------------------------------------------------------------
    puts question.content
    # Display the current question content.
```

Logic Behind Code



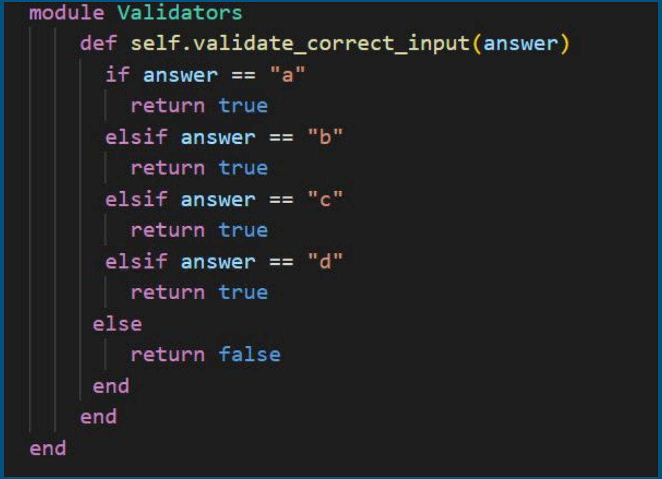
```
if answer == question.answer
# Checks if inputed answer by player is equal to question.answer stored in array.
        score += 1
        # If true increase score counter by 1.
        puts "Correct Answer!".colorize(:green,)
        # If true display to the player it was the correct answer.
        puts @name.colorize(:yellow,) + "'s ".colorize(:yellow,)  + "Progress: ".colorize(:yellow,)
        puts @name.colorize(:yellow,) + "'s ".colorize(:yellow,)  + "Progress: ".colorize(:yellow,)
        # Display current score and attempts left to the player.
puts "----------------------------------------------------------------------------------------------
else
        attempts -= 1
        # If false decrease score counter by 1.
        puts "Incorrect Answer!".colorize(:red,)
        # If false display to the player it was the incorrect answer.
        puts "The Correct Answer: ".colorize(:green,) + question.answer
        # If false display to the player the correct answer.
        puts @name.colorize(:yellow,) + "'s ".colorize(:yellow,) + "Progress: ".colorize(:yellow,)
        puts @name.colorize(:yellow,) + "'s ".colorize(:yellow,) + "Progress: ".colorize(:yellow,)
puts "----------------------------------------------------------------------------------------------
```

Ruby Gems

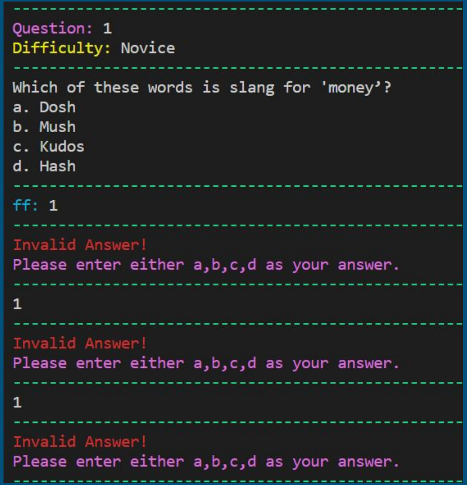
- Menu System utilised tty-prompt
- Zop also runs the following gems:
    - tty -spinner
    - colorize
    - artii

Error Handling

- A validator module was run for the quiz
- Used in a while loop to continuously prevent errors



```
module Validators
    def self.validate_correct_input(answer)
      if answer == "a"
        return true
      elsif answer == "b"
        return true
      elsif answer == "c"
        return true
      elsif answer == "d"
        return true
     else
        return false
     end
    end
end
```



```
----------------------------------------------------------------
Question: 1
Difficulty: Novice
----------------------------------------------------------------
Which of these words is slang for 'money’?
a. Dosh
b. Mush
c. Kudos
d. Hash
----------------------------------------------------------------
ff: 1
----------------------------------------------------------------
Invalid Answer!
Please enter either a,b,c,d as your answer.
----------------------------------------------------------------
1
----------------------------------------------------------------
Invalid Answer!
Please enter either a,b,c,d as your answer.
----------------------------------------------------------------
1
----------------------------------------------------------------
Invalid Answer!
Please enter either a,b,c,d as your answer.
----------------------------------------------------------------
```

Testing

Manual testing was performed throughout the development of Zop

2 Tests were run:

- Correct_answer_testing.rb
- Score_attempt_testing.rb

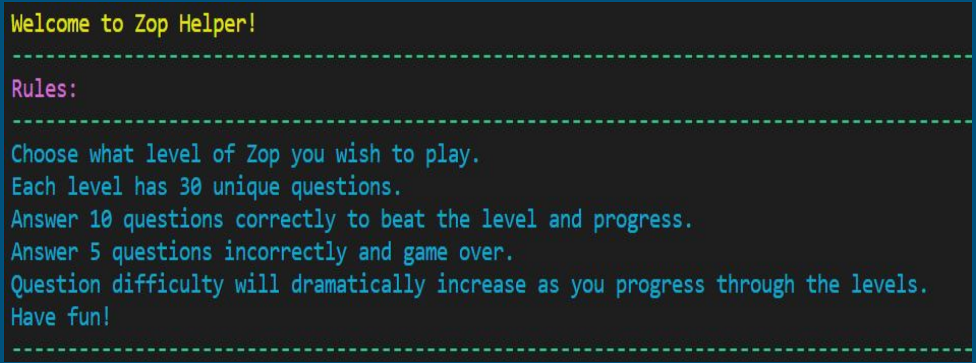
Both tests aided in resolving errors and led to the implementation of the validator module

Install & Run

Install Zop:
./install_zop.sh

Run Zop:
./run_zop.sh

Zop-Helper
Ruby zop_helper.rb
-r for rules
-i for instructions
-d for difficulty



```
Welcome to Zop Helper!
----------------------------------------------------------------------------------------------------
Rules:
----------------------------------------------------------------------------------------------------
Choose what level of Zop you wish to play.
Each level has 30 unique questions.
Answer 10 questions correctly to beat the level and progress.
Answer 5 questions incorrectly and game over.
Question difficulty will dramatically increase as you progress through the levels.
Have fun!
----------------------------------------------------------------------------------------------------
```

Review

Challenges:

- Seeing a plan/feature through until the end

Favourite Parts:

- Solving errors and issues with the code
- Was frustrating at times but worth it
- Watching a player run through the game for the first time without errors

GitHub

https://github.com/NJBLAGA/NathanBlaga_T1A3