

Chance Game - Ruby Terminal Application

This is my ruby based terminal application, it is a chance based game, with 3 seperate games within.

I Started with a concept of what I wanted to do, then moved onto planning it out with a trello board to help me keep to pre set requirements. I then made a flow chart for how I wanted the program to run, researched gems that would fit my application and started coding.

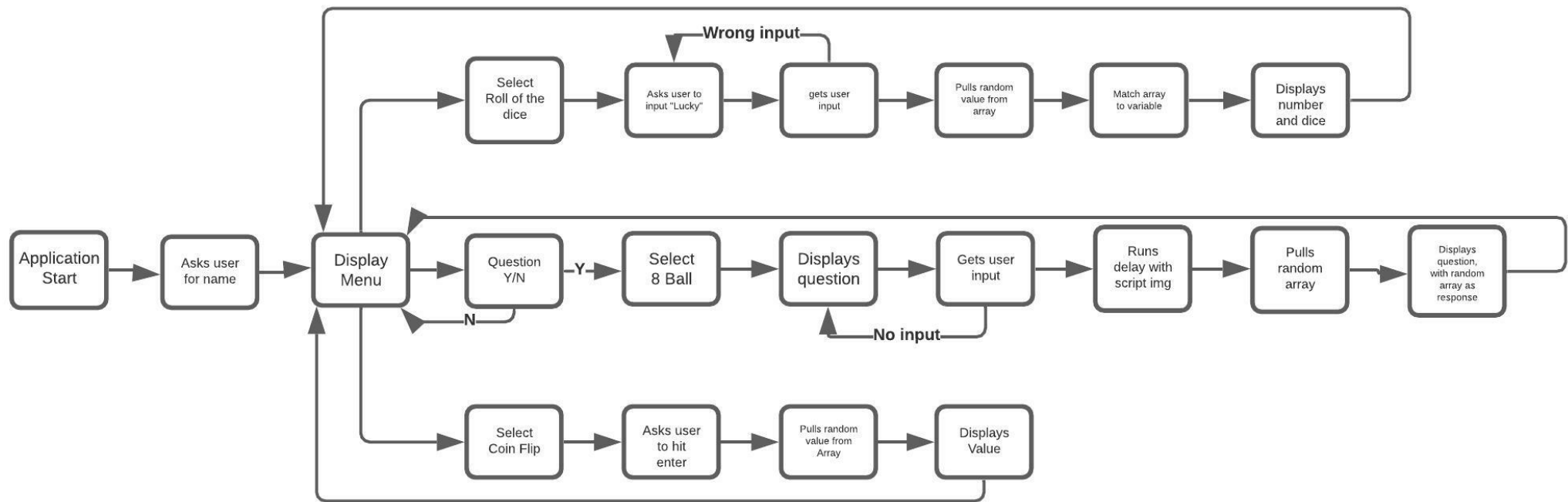
I started by coding each of my sub games, putting them in methods to be able to call them remotely to keep my code nice and clean. I then added the main menu system and joined all the games. I went in after and added styling and extra little run features.

```

  /  _  \ | | | | /  _  \ | | | | /  _  \ | | | |
 | /  \ | | | | /  \ | | | | \  /  \ | | | |
 | |  | |  | |  | |  | |  | |  | |  | |  | |
 | \  / | | | | | | | | | | \  / | | | |
 \    / | | | | | | | | | | \    / | | | |

Hello Kyle, Welcome to the Chance Game!!
Select what game you would like to play: (Press ↑/↓ arrow to move and Enter to select)
▶ Magic 8 Ball
  Coin Flip
  Roll The Dice
  Exit
```

Flow Chart



Initial Setup & Command Line Arguments

When you run the bash script it will automatically install any required gems, bundle and bring up a section for you to enter your name and supply it to the main script. It then runs the ruby terminal application.

```
Hello, what is your name?
```

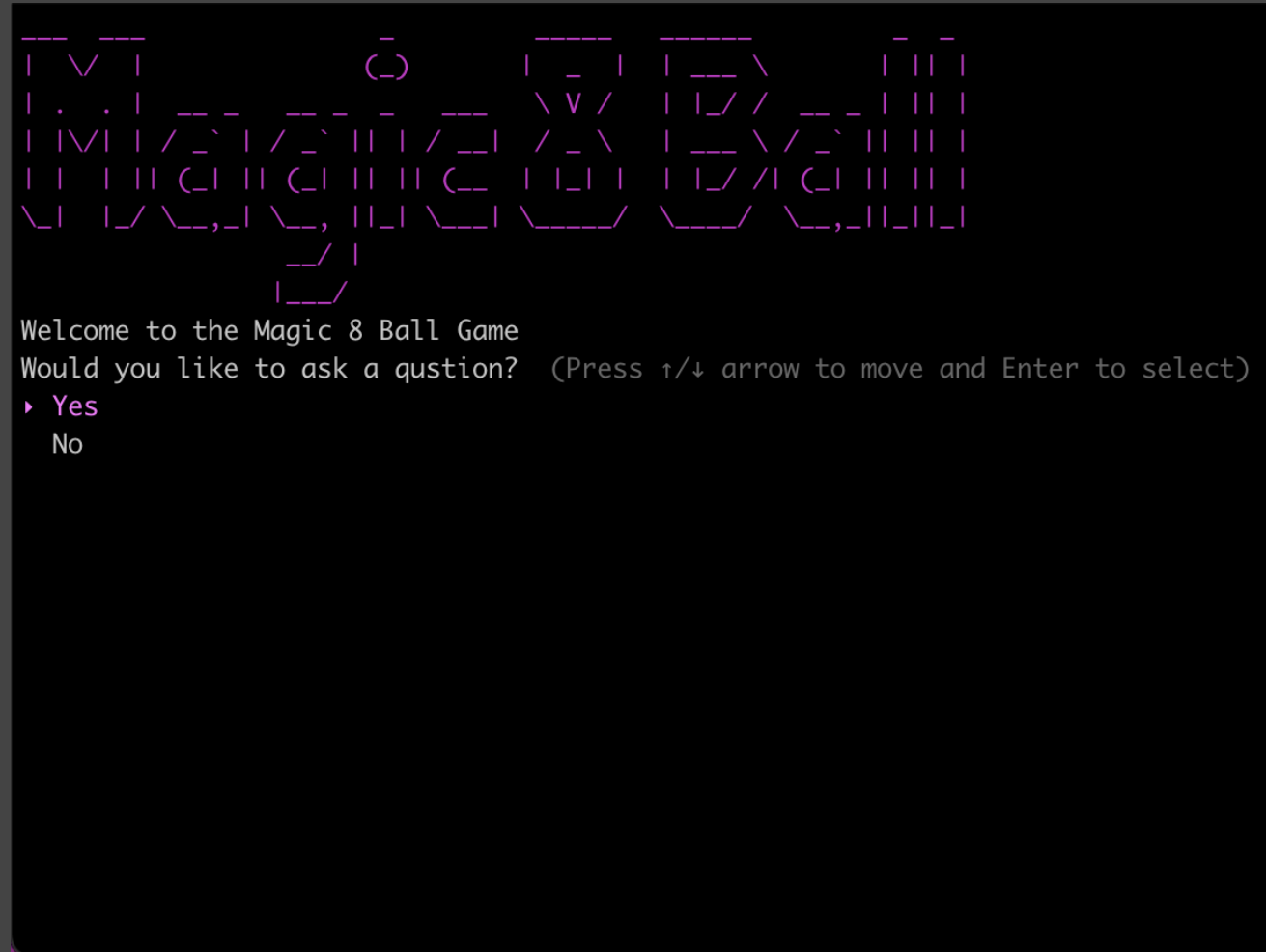


Page 10 of 10

```
Hello Kyle, Welcome to the Chance Game!!
Select what game you would like to play: (Press ↑/↓ arrow to move and Enter to select)
▶ Magic 8 Ball
  Coin Flip
  Roll The Dice
  Exit
```

Magic 8 Ball

When you select the Magic 8 Ball app, it runs another question verifying whether or not you would like to ask a question. No will take you back to the main menu, where Yes will continue in the application. Once you ask a question it brings up a quick loading screen of an 8 Ball and then displays your question with a randomly generated answer afterwards.



Coin Flip

The Coin flip is pretty simple, it asks the user to hit Enter (Any key will work) and the application spits out either heads or tails.

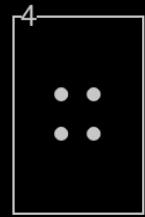
[illegible]

Roll The Dice

To start the dice roll you need to type Lucky when you are asked, it will then roll the dice and display your number with a box representing the dice face using TTY-Box.



To roll the dice please type Lucky: lucky
Your result is:



'Press any key to continue'
█

Code Run Through

This is my main code block that brings in the argument statement and turns it into a variable. Uses a case statement in conjunction with tty prompt to chose and run menu options. I have loops for error handling if no name is entered, and for when you enter the Magic 8 Ball.

I put the coin flip into the main code as it was such a small element. I used an array to store heads or tails, then .sample to pull a random element from that array and display it.

```
src > main.rb
1 require_relative "../app_pages/8ball.rb"
2 require_relative "../app_pages/dice.rb"
3
4 require "tty-prompt"
5 require "pastel"
6 require "tty-font"
7
8 prompt = TTY::Prompt.new(active_color: :bright_magenta)
9 pastel = Pastel.new
10 font = TTY::Font.new(:doom)
11
12 user_name = ARGV[0]
13
14 while true
15
16     system "clear"
17     puts pastel.magenta(font.write("CHANCE!"))
18
19     if ARGV.size == 0
20         puts "Welcome to the Chance Game!!"
21     else
22         puts "Hello #{user_name.capitalize}, Welcome to the Chance Game!!"
23     end
24
25     ARGV.clear
26
27     choice = prompt.select("Select what game you would like to play:", ["Magic 8 Ball", "Coin Flip", "Roll The Dice", "Exit"])
28
29     case choice
30     when "Magic 8 Ball"
31
32         system "clear"
33         puts pastel.magenta(font.write("Magic 8 Ball"))
34         puts "Welcome to the Magic 8 Ball Game"
35         enter = prompt.select("Would you like to ask a question? ", ["Yes", "No"])
36
37         if enter == "Yes"
38             system "clear"
39             magic_8_ball
40         end
41
42     when "Coin Flip"
43
44         system "clear"
45         puts pastel.magenta(font.write("Coin Flip"))
46
47         results = ["Heads", "Tails"]
48         print "To flip the coin hit the 'Enter' key: "
```

```
src > main.rb
44
45 when choice == "Coin Flip"
46     system "clear"
47     puts pastel.magenta(font.write("Coin Flip"))
48
49     results = ["Heads", "Tails"]
50     print "To flip the coin hit the 'Enter' key: "
51
52     STDIN.getch
53
54     system "clear"
55
56     puts pastel.magenta(font.write("Coin Flip"))
57     print "Your result is: "
58     puts results.sample
59     puts "'Press any key to continue'"
60     STDIN.getch
61
62 when choice == "Roll The Dice"
63     system "clear"
64     puts pastel.magenta(font.write("Roll The Dice"))
65     dice_roll
66
67 when choice == "Exit"
68     system "clear"
69     return
70 end
```


Dice Roll Code

This is the method to run my dice roll section. It uses variables to store each of the dice numbers and the TTY-box images. It uses a loop with rescue to handle user input error and pull a random array and match it to the variable to display to the user.

```
src / app_pages / dice.rb
You, 15 hours ago | 1 author (You)
1  require "tty-box"
2
3  def dice_roll
4
5      one = TTY::Box.frame " • ", title: {top_left: "1"}, align: :center, padding: [1,2,1,2]
6      two = TTY::Box.frame " • ", title: {top_left: "2"}, align: :center, padding: [1,2,1,2]
7      three = TTY::Box.frame " • \n • ", title: {top_left: "3"}, align: :center, padding: [1,2,1,2]
8      four = TTY::Box.frame " • \n • ", title: {top_left: "4"}, align: :center, padding: [1,2,1,2]
9      five = TTY::Box.frame " • \n • \n • ", title: {top_left: "5"}, align: :center, padding: [0,2,0,2]
10     six = TTY::Box.frame " • \n • \n • ", title: {top_left: "6"}, align: :center, padding: [0,2,0,2]
11
12     dice = [one, two, three, four, five, six]
13     print "To roll the dice please type Lucky: "
14
15     roll = gets.chomp.capitalize
16
17     # system "clear"
18
19     if roll == "Lucky"
20         puts "Your result is: "
21         puts dice.sample
22         puts "'Press any key to continue'"
23         STDIN.getch
24     else
25         raise
26     end
27
28     rescue
29         system "clear"
30         print "Please retry: "
31     retry | You, 2 days ago • Fixed dice roll run error
32     return
33 end
```

Magic 8 Ball Code

The magic 8 ball game is stored in a method, it uses an array to store all the answers. When you answer the question I have it auto capitalize the question and remove any question marks as they are added later. It uses a loop and rescue for error handling and an added bonus for a specific question.

```
app_pages > 8ball.rb
You, 11 hours ago | 1 author (You)
require "pastel"
require "tty-font"

def magic_8_ball

  answers = ["It is certain", "It is decidedly so", "Without a doubt", "Yes, definitely",
    "You may rely on it", "As I see it, yes", "Most likely", "Outlook good",
    "Signs point to yes", "Yes", "Reply hazy, try again", "Ask again later",
    "Better not tell you now", "Cannot predict now", "Concentrate and ask again",
    "Don't bet on it", "My reply is no", "My sources say no", "Outlook not so good",
    "Very doubtful"]

  print "What is your question? "
  question = gets.chomp.capitalize.delete("?")

  if question != ""
    system "clear"

    puts "
      _____
      dP9CGG88@b
      IP   _  Y888@b
      dIi  (_)  G8888@b
      dCII  (_)  G8888@b
      GCCII   GG8888@@
      GGGCCCCCGGG88888@@@
      GGGGGCCGGGG88888@@@
      Y8GGGGGG8888888@@@P
      Y888888888888@@@P
      Y888888888888@@@P
      @@@@@@@@@P
    "

    sleep (1)
    system "clear"
    if question == "What is the meaning of life"
      print "#{question}? "
      puts "42"
      puts "'Press any key to continue'"
      STDIN.getch
    else
      print "#{question}? "
      puts "#{answers.sample}"
      puts "'Press any key to continue'"
      STDIN.getch
    end
  end
end
```

```
src > app_pages > 8ball.rb
43       puts "'Press any key to continue'"
44       STDIN.getch
45     end
46   else
47     raise
48   end
49
50   rescue
51     system "clear"
52     print "You want to ask something? "
53     retry
54   return
55 end
```

Challenges & Favorite Parts

Challenges

- Command argument was messing with my gets.chomp
- Getting my menu to load back to the main menu and not automatically exit the application
- Choosing what gems to use

Favourite parts

- Coding my own idea
- The code doing what I wanted to do
- Problem solving as issues came up
- Adding styling

Thank you

Presentation by me