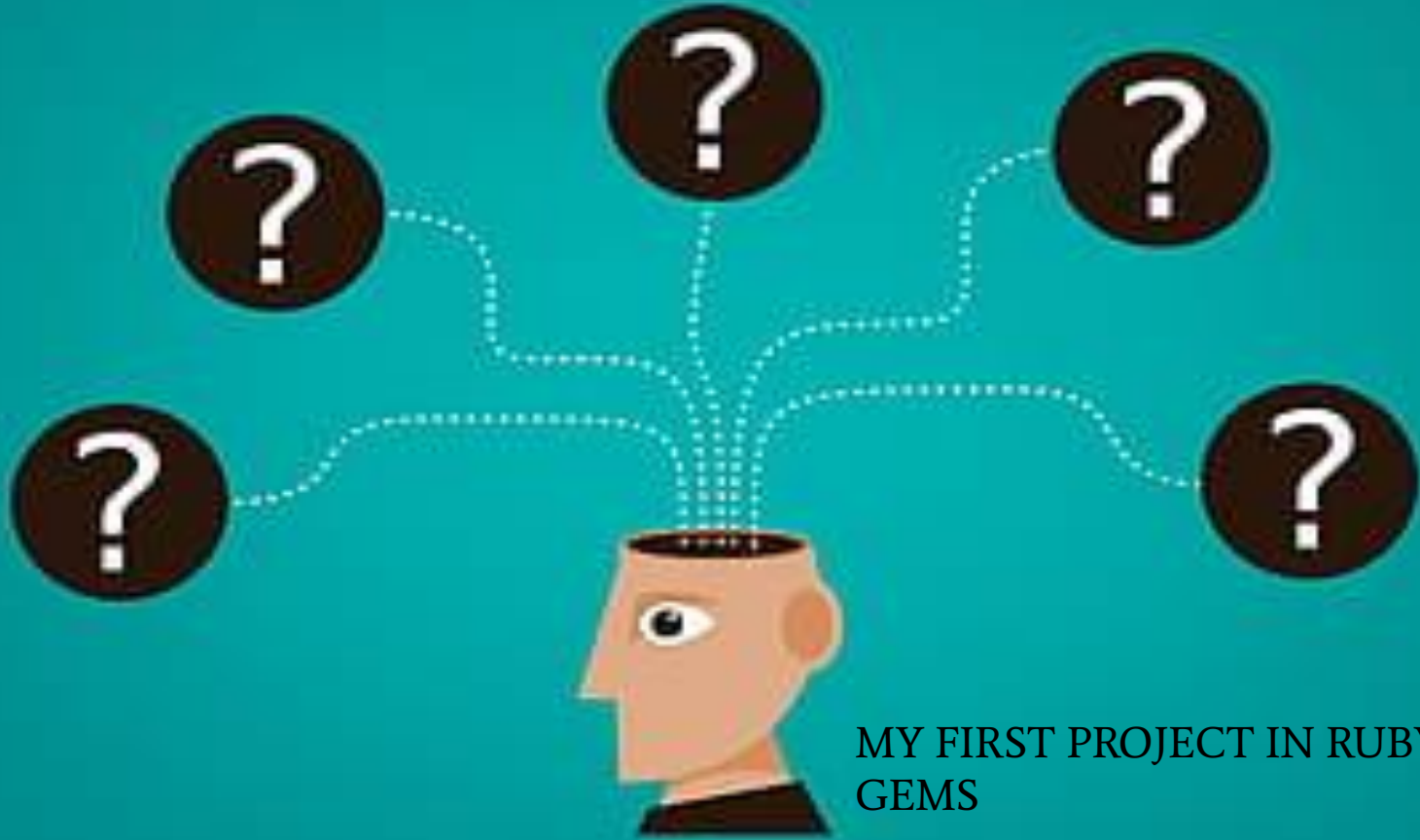


HANGMAN-GUESSING GAME



MY FIRST PROJECT IN RUBY WITH
GEMS

PROJECT PLAN

1. BRAINSTORM
2. PLAN
3. DEVELOP
4. TEST
5. PRESENT

@ BRAINSTORM

1. Bounce my ideas around and narrow down to one or two that I like.

2. Discuss with my Educator's & Finalize.

3. Then start planning to build up my application.

@PLAN

- a. Clearly define solution: What is it? What will it do?
- b. Describe the flow of our application. What might the user do or expect and how will they interact with it.
- c. Decide which features are part of core functionality and which should be stretch goals to be completed if time allows.
- d. Create a user story map to describe the user's experience.
- e. Create a workflow diagram or flowchart to describe the order of operations.
- f. Write pseudocode for the application

@ DEVELOP

WRITE CONTENT

IMPLEMENT REQUIRED FILE STRUCTURE.

WRITE CODE FOR STRETCH GOAL & core features.

Development tool is VS code.

@TEST

1. Manually test the application

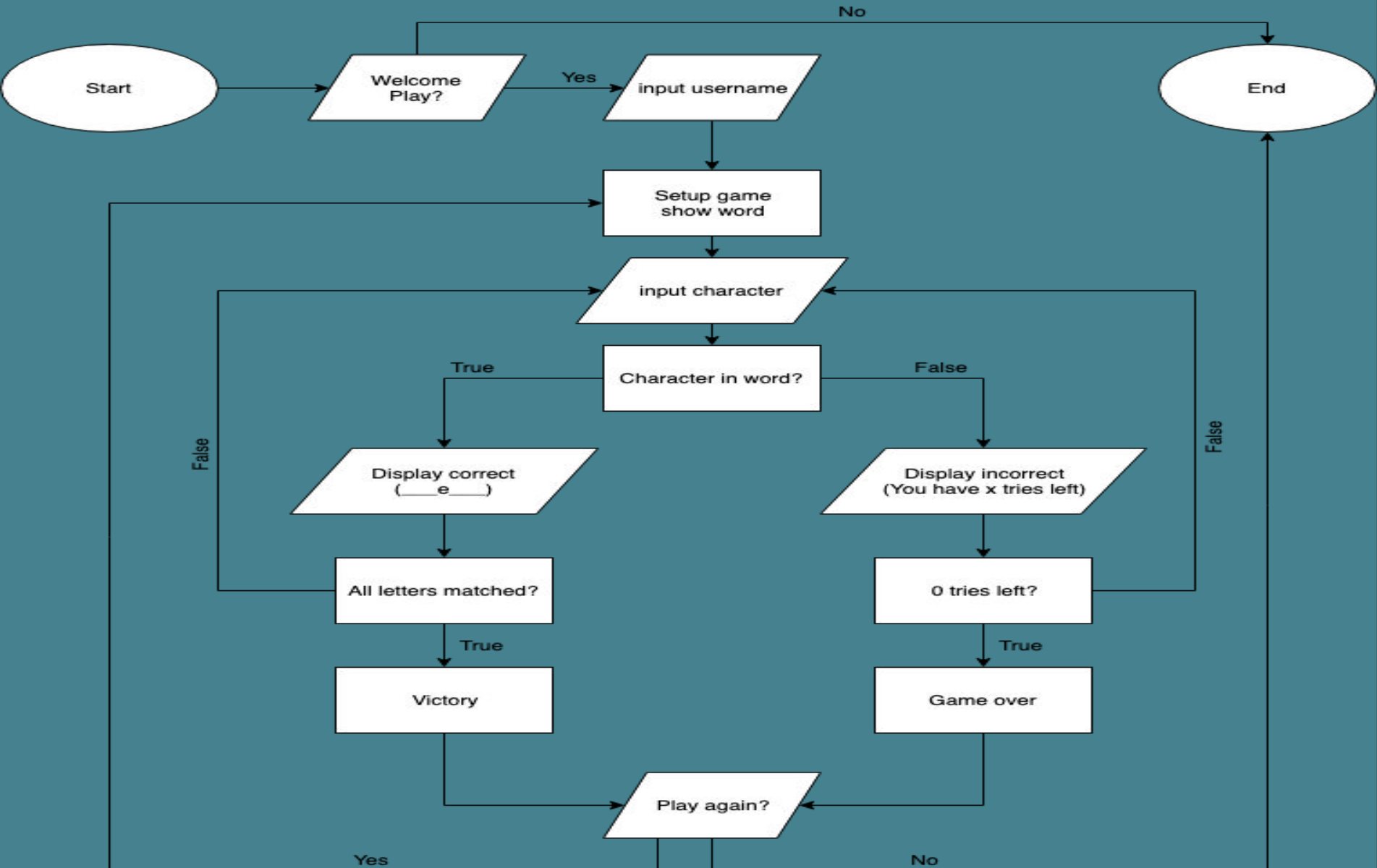
2. Fix any bugs that arise

3. Try to break the application with user interactions.

@PRESENT

Create a google doc
presentation to show
demo of the
presentation.

Flow Chart



Introduction

User would get 7 chance to guess the word , every wrong letter guess , will lose one chance and every right letter guess , will get the message that would guess the right word.

Every time user will see the Hangman movement according to words guess.

When user guess correct word

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

```
chaitee27@Chaitee:~/ChaiteeDuttaGuhaAssessments/ChaiteeDuttaGuha_T1A3$ ruby src/Strat/console
ruby: No such file or directory -- src/Strat/console (LoadError)
chaitee27@Chaitee:~/ChaiteeDuttaGuhaAssessments/ChaiteeDuttaGuha_T1A3$ ruby src/Start/console
```

```
Welcome to Hangman!
To win, you need to guess the mystery word or you die.
You can have up to 8 incorrect guesses, before you're hanged.
Let's begin!
```

```
-----
Guess a letter
```

```
>
```

```
a
```

```
That's correct!
```

```
Here are your correct guesses: _ _ a _ _ _ _ _
```

```
Guess a letter
```

```
>
```

```
[]
```

Let's begin!

Guess a letter

>

a

That's correct!

Here are your correct guesses: _ _ a _ _ _ _

Guess a letter

>

r

```
+-----+
|       |
|       |
|       |
|       |
|       |
|       |
+-----+
```

You have 7 guesses left.

Here are your incorrect guesses: r

Here are your correct guesses: _ _ a _ _ _ _

Guess a letter

>

f

```
+-----+
|       |
|       |
|       |
|       |
|       |
|       |
+-----+
```

You have 6 guesses left.

Here are your incorrect guesses: r f

Here are your correct guesses: _ _ a _ _ _ _

Guess a letter

>

□

When user guess wrong word

Motivation

- People can learn new words through playing.
- Also helpful for remembering the spelling of new words.

Things I have learned

- Ruby language , logic , syntax .
- Using branches in Git
- Methods call and compile the files.

Things I have struggled with

- Testing output with RSpec.
- Scope of instance variables.
- Knowing when to switch off and take breaks.

THANK YOU