



TERMINAL
APP
PRESENTATION

2021

PREPARED BY
JASMINE OFAOLAIN

I would like to acknowledge the traditional custodians of the land on which I study, work and live - the Wurundjeri People of the Kulin Nation - whose sovereignty was never ceded. I pay my respects to Elders past, present and emerging.

CONCEPT

Modern day parenting is a balancing act! Be “too caring” towards your child and you are judged as being “crunchy”. Breastfeed your child and be judged as a lactivist! Bottle feed your child and be judged for not giving your child the best start to life. Feed your kid processed food? Look out, you will be judged! It really is a minefield out there!

This game is a tongue-in-cheek game considering some of those concepts - lets see if you really **are** a terrible parent.

CONCEPT

The Judged Parent Game -
How will you fare after one day as a parent??

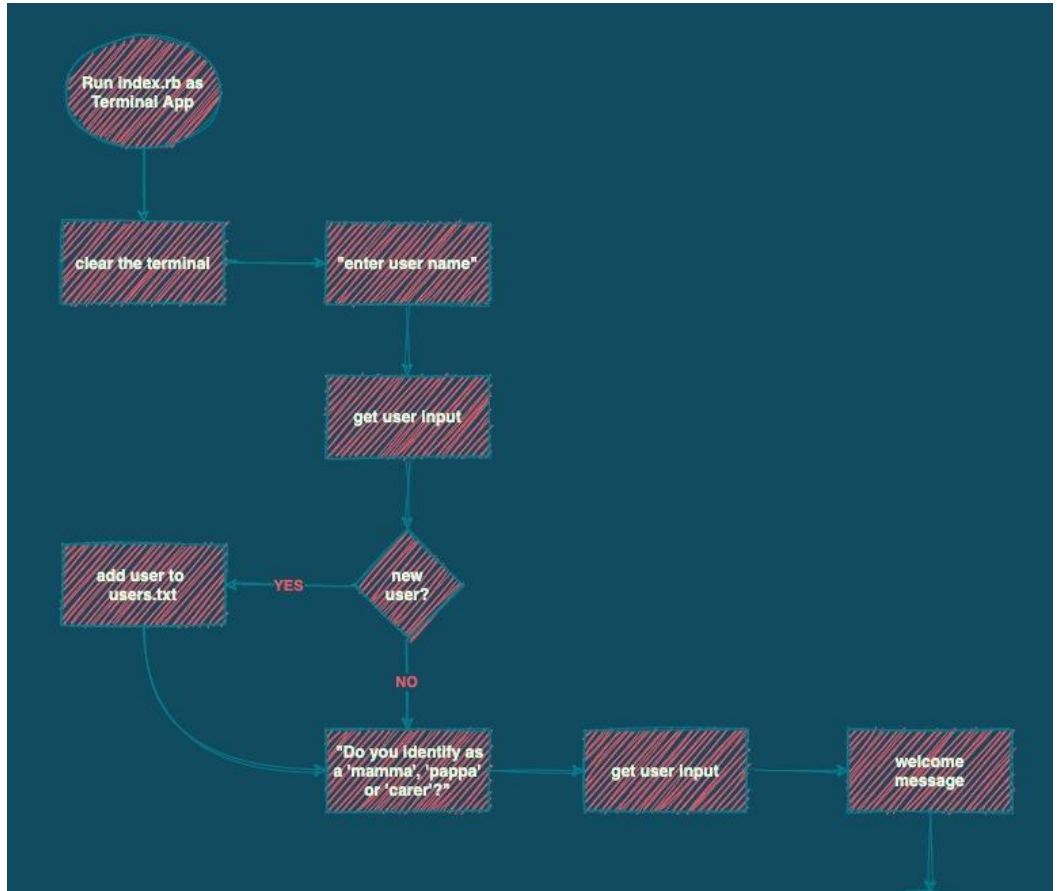
Will you succeed in being a crunchy parent? Or a bad parent? Or a middle of the road parent. Play along to find out!

So, which parent will you be??



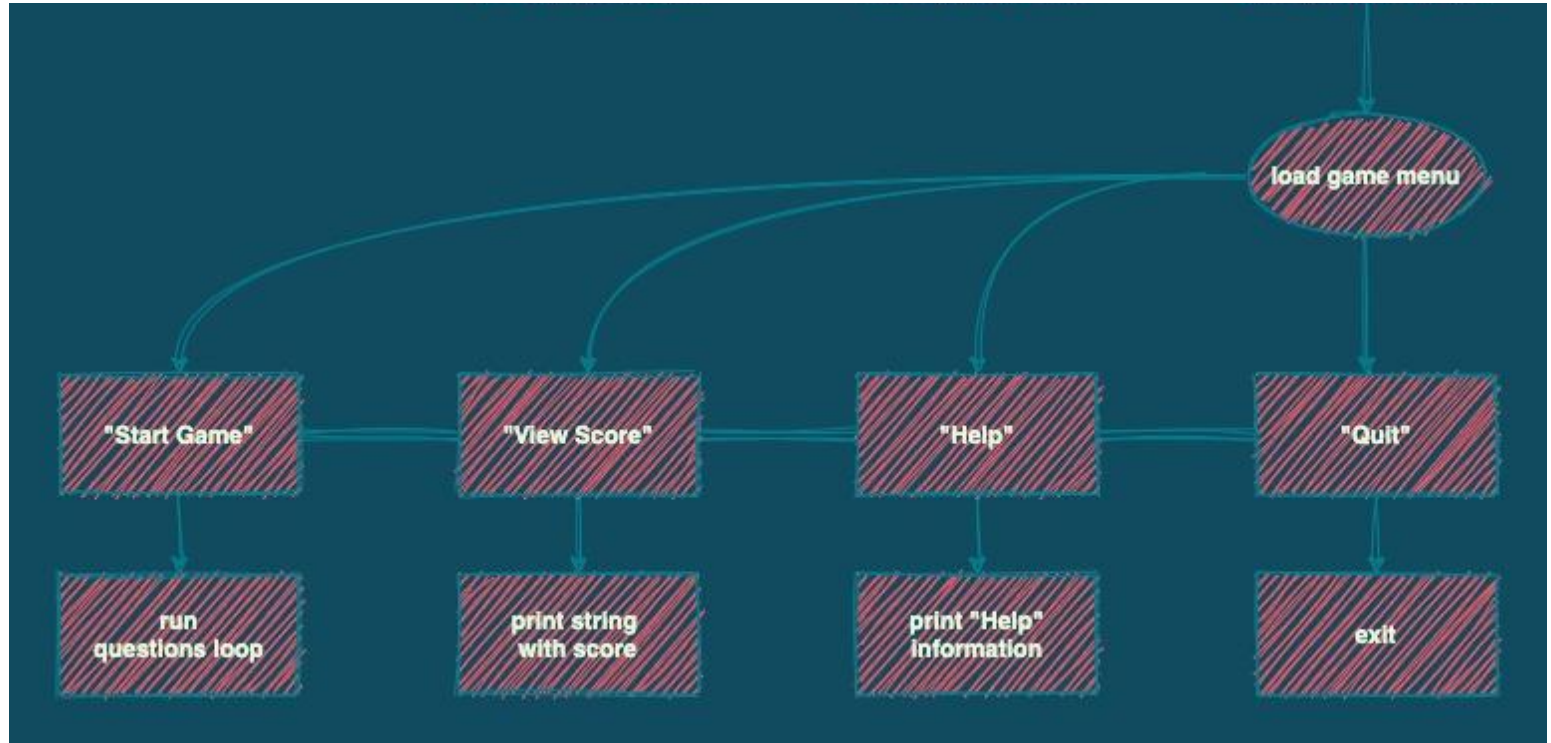
FLOW CHART:

RUN THROUGH TO WELCOME



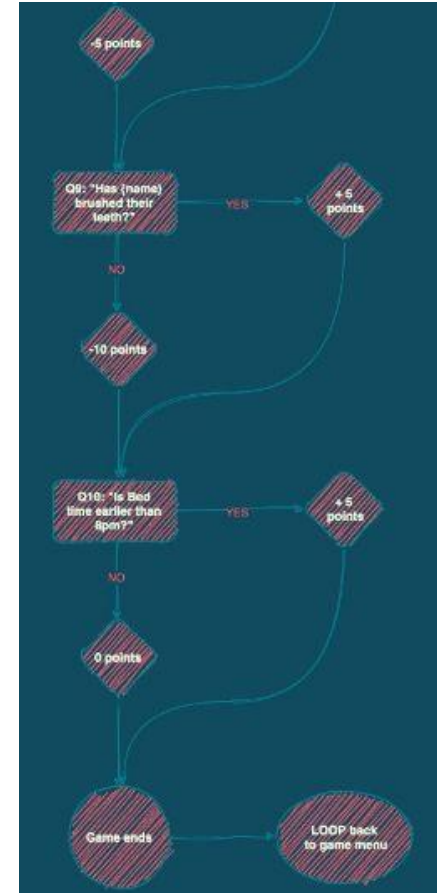
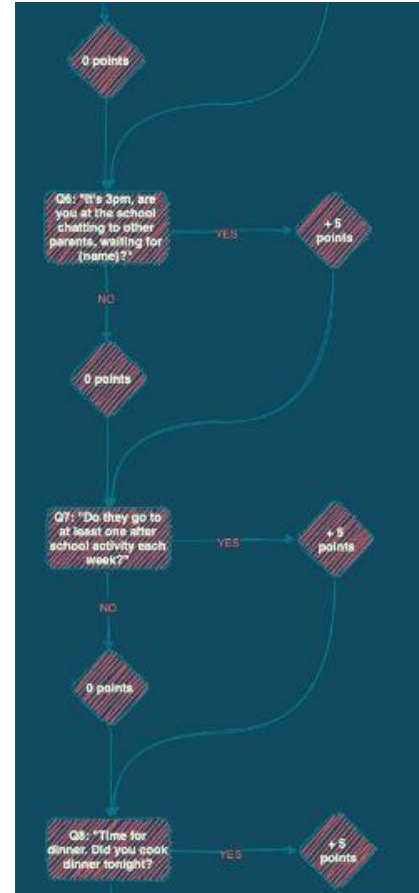
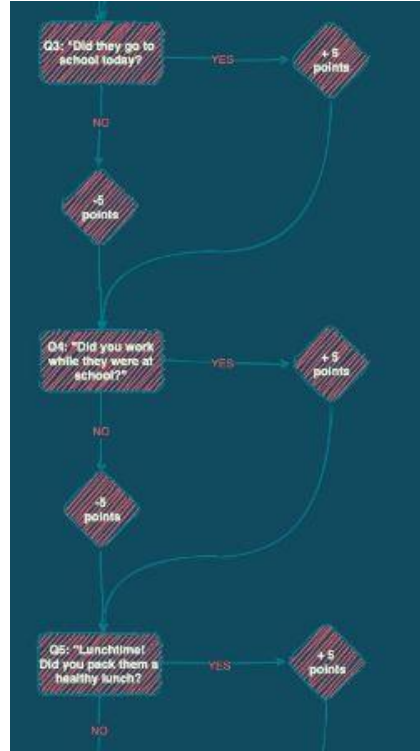
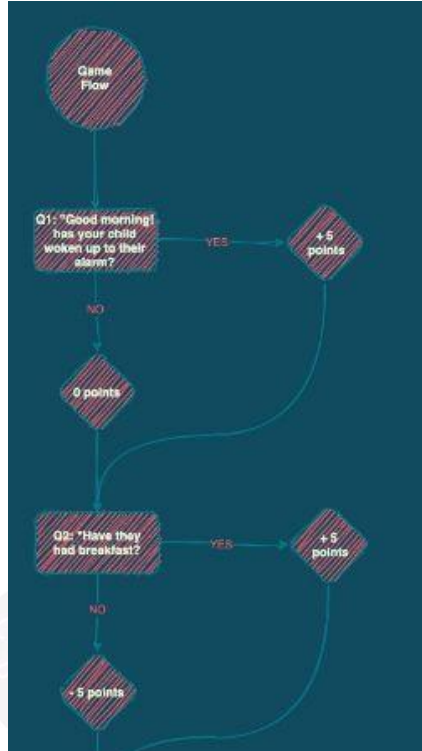
FLOW CHART:

GAME MENU LAYOUT



FLOW CHART:

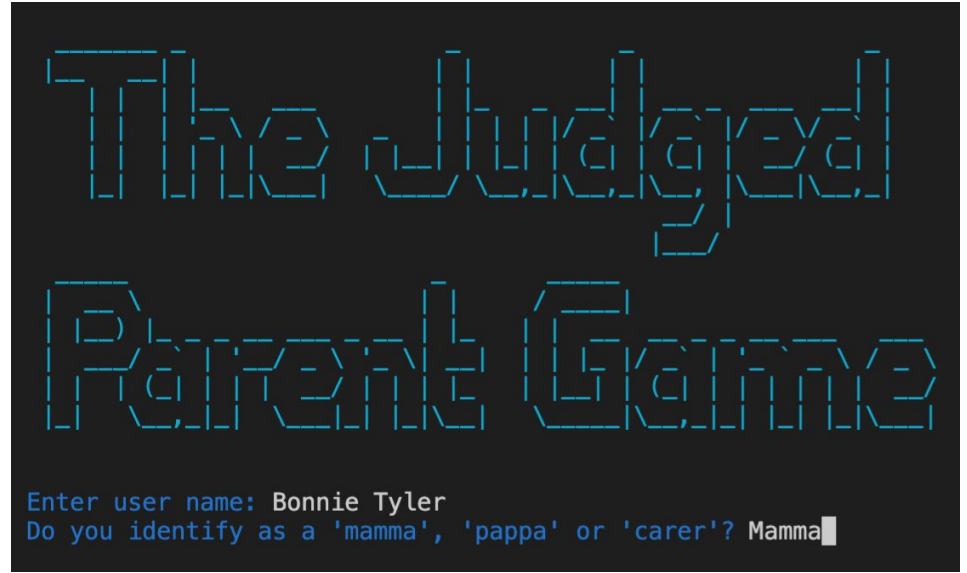
QUESTION FLOW LAYOUT



FEATURE 1:

USING FILE FOR USER SYSTEM

```
# -----#  
#  "LOGIN"  #  
# -----#  
print "Enter user name: ".colorize(:blue)  
@user_name = gets.chomp.downcase  
print "Do you identify as a 'mamma', 'pappa' or 'carer'? ".colorize(:blue)  
@user_role = gets.chomp.downcase  
  
@new_user = true  
  
@lines = File.readlines("users.txt")  
  
@lines.map {|user| @new_user = false if user.strip == @user_name}
```



- User input (gets method) used to gather “name” and “caring “role” data
- User name is pushed to text file
- Different greetings are printed, depending on whether user is new or returning
- Variables are used in string interpolation to personalise interactions with user



FEATURE 2:

INTERACTIVE MENU SYSTEM

```
loop do
  puts "=====
  puts
  case menu(prompt)
  when 1
    questions
    all_questions
  when 2
    view_score
  when 3
    help
  when 4
    exit
  end
end
```

- TTY-Prompt used for simple, clean aesthetics
- Choices array used to list the 4 menu options, within the menu method
- Loop used on case method to keep the menu looping through and displaying the list whenever the previous item had completed running

```
Enter user name: Bonnie Tyler
Do you identify as a 'mamma', 'pappa' or 'carer'? Mamma
```

```
Welcome to 'The Judged Parent Game' Bonnie tyler!
```

```
Have fun analyzing your potential parenting abilities, I'm sure you will
be the best Mamma ever!!
```

```
=====
```

```
What would you like to do? (Press ↑/↓ arrow to move and Enter to select)
```

```
▶ Start-game
View-score
Help
Quit
```

```
# -----#
#  GAME MENU  #
# -----#
prompt = TTY::Prompt.new(active_color: :blue)

def menu(prompt)
  choices = [
    {name: "Start-game", value: 1},
    {name: "View-score", value: 2},
    {name: "Help", value: 3},
    {name: "Quit", value: 4},
  ]
  prompt.select("What would you like to do?",
    choices)
end
```

FEATURE 3:

TOTAL SCORE BASED ON ANSWER

```
def all_questions
  score = 0
  #q1
  puts
  puts "Good moring, has your child woken up to their
alarm? (yes/no)"
  answer = gets.chomp.downcase
    if answer == "yes"
      score += 5;
    else
      score += 0;
    end
end
```

Scored

Mamma Bonnie tyler, you scored !=====

What would you like to do? (Press ↑/↓ arrow to move and Enter to select)

► Start-game
View-score
Help
Quit

- The 'yes' or 'no' answer of each Question has a score value associated with it
- User input received using gets.chomp.downcase
- The method calculates the score correctly, but I struggled to get the total score variable extracted from the data
- The output still has an error I need to work through



GEMS

Colorize - for making the app colourful and pleasing to the eye

TTY-Prompt - to make the user interface user friendly

Artii - to create the game name banner for the game

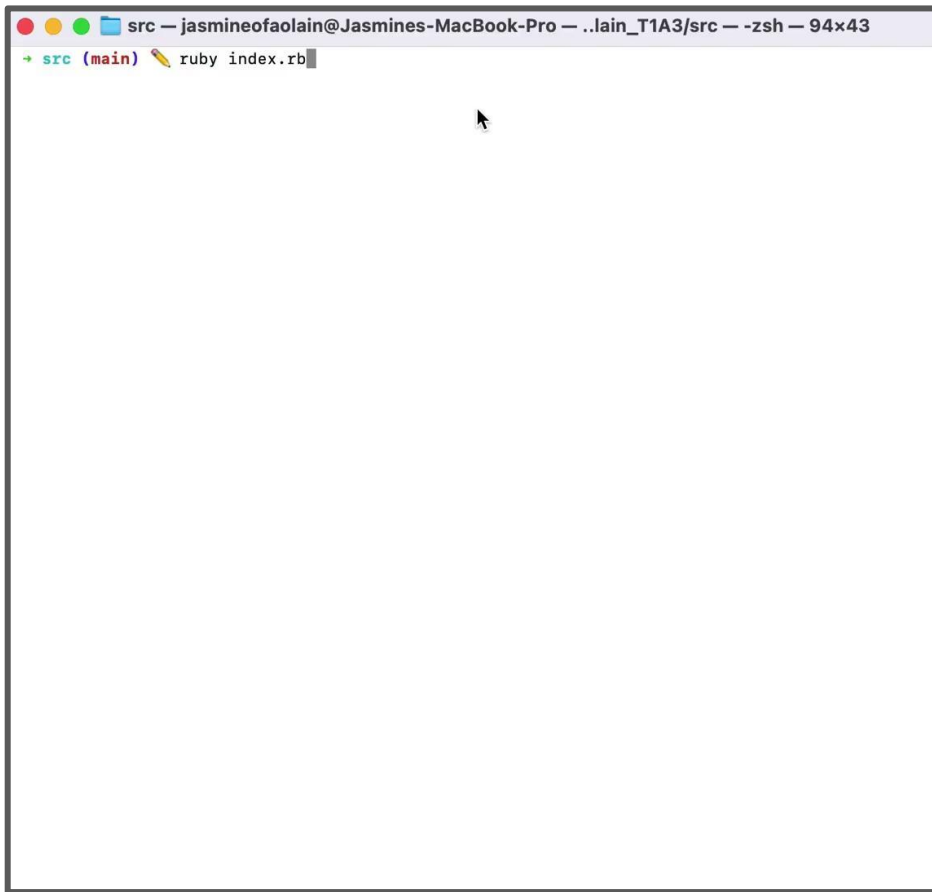
Espeak-Ruby - to send the message home



APP WALKTHROUGH

Highlights:

- Welcome page
- User input
- Menu usage
- Game feature
- Score feature
- Help feature
- Quit to return to Terminal



POSSIBLE EXTENSIONS (SPRINKLES)

- Making a more complicated question/scoring system
 - some questions were meant to have multiple choice answers which would score different amounts, depending on the option
- Add extra outputs after each question is answered (either encouraging or heckling the player)
- Tracking over time - Today's score is, yesterday you scored...
- Set Profile - so there can be set tasks for Parent Vs Parent
- Random parenting tip generated throughout the game
 - V1 - hard code from a list locally
 - V2 - API from internet



REVIEW OF BUILD PROCESS

CHALLENGES

- Getting my head around Ruby
- Being sick :(
- Missing some foundations in Ruby while away sick

ETHICAL ISSUES

- Some people are triggered by being critiqued, gentle or 'kind' ending required
- Privacy - make sure the data is secure

FAVOURITE PARTS

- Fun & a bit tongue-in-cheek
- It is working- just
- Lots of things I can refactor as I learn more



Let's stop being so damn
judgemental & crucifying
everyone who doesn't fit into our
boxed-in perceptions of what is
right.

Gillian Anderson

Thank you!