T1A3 - Terminal Application

Small Marketing Event Check-in Application

Present by Vivian Zhang

25 of May 2021

Lolcats - clapping - LOL at Funny Cat ...
icanhas.cheezburger.com

Funny Face (9/9) Movie CLIP - Clap Yo ...
youtube.com

Funny Clap - Animated Gif Image...
gifscenter.com

Related searches


clapping gif


animated clapping gif

clapping meme

ZoneAlarm Results
search.zonealarm.com

clapping funny gif | WiffleGif
wifflegif.com

funny GIFs - Primo GIF - Latest ...
primogif.com

Thats Some Funny Shit Clap GIF ...
pinterest.com

Funny Guy Film Clap Stock Photo (Edit ...
shutterstock.com

Best Funny Clap GIFs | Gfycat
gfycat.com

T1A3 - Terminal Application

Agenda

What will going through?

- A walk-through of Terminal application, features and how it used
- A walk-through of the logic of Terminal application and code
- A review of development/build process including challenges, ethical issues, favourite parts, etc

T1A3 - Terminal Application

Why I did that

- Working experience
- Use the fun and simple way to solve the problem ---- like rock-paper-scissors . (It is alway fun and works.)

T1A3 - Terminal Application

Features

- The app will be reaction User's name input and will get a greeting with colorize name.
- User Check-in system, that user input their name and check if they are new user.
- New user will get a index check-in number output from a file.
- The application will mention them if the index number of this client is "lucky number"
- The user have option to choose play a tossing-coin-game that pretty simple but very helpful small and fun when a group of people getting together.
  (Not to much like a "TOO FUN GAME! " every one enjoy to play byself.)
- User can choose the tossing times like 1 times, 3 times or 5 times.

T1A3 - Terminal Application

This terminal application design for provide service for support to organize event and function that 5~35 people's Check in and also with lottery function, pick the client name randomized. Also with a toss coin game for entertainment purpose.

1. It will instead of paper,easy to check in and clients data management more safety and also will return a check-in number
2. The check-in number also will be the certificate for lottery game.
3. The app also provide the small game (toss a coin) function for entertainment, the client can play it when they can't pick up their decision that choose the 1 of 2 option.

T1A3 - Terminal Application

The target audience is small~ medium size event and function organizer.

Help them to check-in guest, and also provide simple entertainment function for guest.

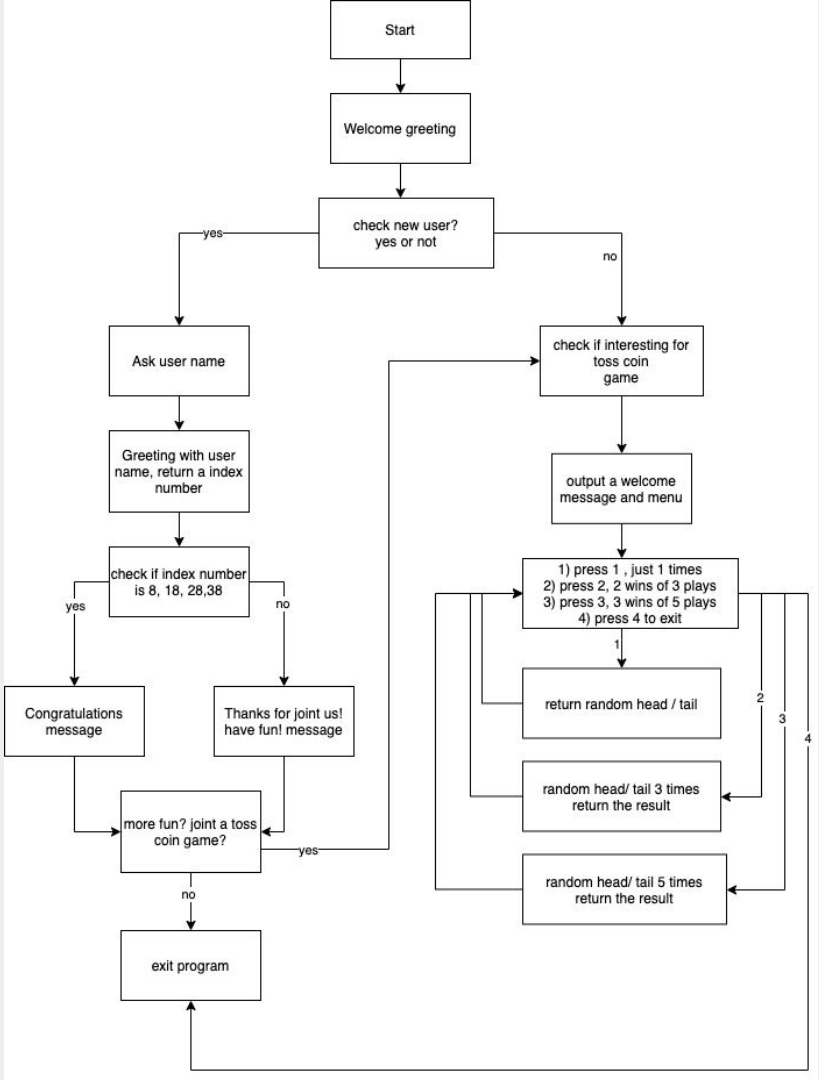
otherwise, the organizer could set lottery number as check-in order, for example: 8, 18, 28, 38.

-target audience number ---- 5 ~ 35

T1A3 - Terminal Application

Flowchart

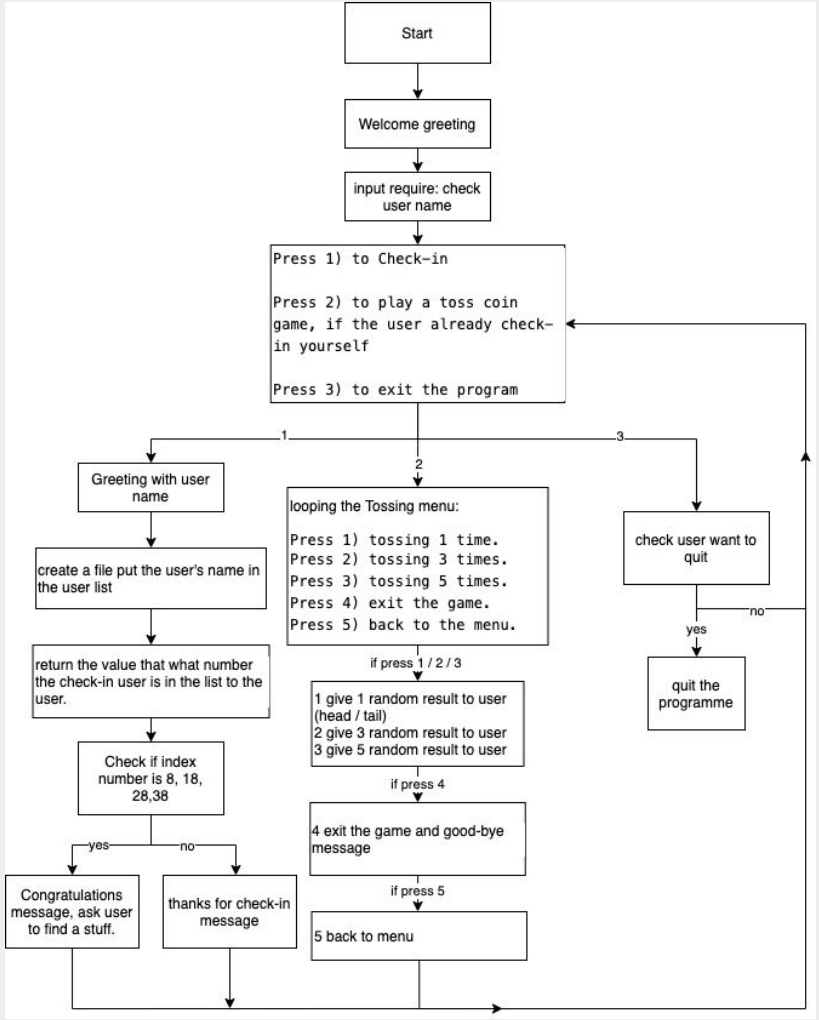
Version 1



T1A3 - Terminal Application

Flowchart

Version 2



Coding part ----- Gem: colorize; Variable;

T1A3 - Terminal Application

```ruby
def greeting
    puts ColorizedString["Hi there!"].colorize(:light_magenta)
    sleep(1)
    puts ColorizedString["Welcome to our check-in app!"].colorize(:yellow)
    sleep(1)
    puts ColorizedString["What's your full name?"].colorize(:light_black)
    sleep(1)
    #take input from user
     print  print ColorizedString["> "].colorize(:blue)
    @user_name = gets.strip.upcase.colorize(greeting_color)
        puts "Hi #{@user_name}, nice to meet you!"

end
```

T1A3 - Terminal Application

```
Hi there!
Welcome to our check-in app!
What's your full name?
> Sophie
Hi SOPHIE, nice to meet you!
Press 1) to Check-in, if you are new user.
Press 2) to play a toss coin game, if you already check-in yourself.
Press 3) to exit the program.
```

```
Press 1) toss 1 time.
Press 2) toss 3 times.
Press 3) toss 5 times.
Press 4) exit the game.
Press 5) back to the menu.
> 1
_Start_
3
2
1
You got a 'Head'
```

Coding part ----- File I/O

```ruby
def new_user_check
    # add the user to user list, check if the name is already in.
    # if the name is in, double check message
    # if the name is not in the list, add the name, and give back a client number.
    #if client number is 8, 18, 28, 38, give a congratulation message, ask them to contect staff.

    puts "Hi there. Glad to see you #{@user_name}!"
    sleep(2)
    puts "One second, let's check your number now."
    sleep(2)
    print "."
    sleep(1)
    print "."
    sleep(1)
    print "."
    sleep(1)
 #input user name to client_data.txt
#  File.open("client_data.txt", "a") { |f| f.write "#{@user_name}\n" }
 File.write("client_data.txt" , "\n#{@user_name}" , mode: "a")

 # check the user's index number
 filename = 'client_data.txt'
 line_count = `wc -l "#{filename}"`.strip.split(' ')[0].to_i
```

T1A3 - Terminal Application

Coding part ----- If else statement

T1A3 - Terminal Application

```
#  FileIO.write(@user_string)
#return the index of the user in the user list
if line_count == 8 || line_count == 18 ||  line_count == 28 ||  line_count == 38

   puts ColorizedString[" Oh, lucky you! "].colorize(:light_yellow ).colorize( :background => :red)
   puts " your number is 00#{line_count}."
   sleep(1)
   puts ColorizedString[" You Won the Lottery! Congratulations! "].colorize(:light_yellow ).colorize( :backg
   sleep(1)
   puts "Find our staff member and show him this message."

else
   puts "Thanks for waiting, your number is..00#{line_count}."
   sleep(2)
   puts "Have fun!"
   sleep(2)
end
end
```

```ruby
case check_user_name
 when 1
    new_user_check
#push clients to the group and reture clients a number
#let them know if their number is 8, 18, 28,38, they will get congratulation message.
#else, puts the thanks check-in message, reture to the menu.

 when 2 #tossing coin game.
    tossing_coin_loop = true
 while tossing_coin_loop
    original_result = ["Head", "Tail", "Head", "Tail", "Head", "Tail", "Head","Tail", "Head", "Tail"]
    case tossing_coin_method

    when 1
        count_down
        puts "You got a '#{original_result.shuffle[0]}'"

    when 2
        count_down
        puts "You got '#{original_result.shuffle[0]}', '#{original_result.shuffle[1]}', '#{original_resul
    when 3
        count_down
        puts "You got '#{original_result.shuffle[0]}', '#{original_result.shuffle[1]}', '#{original_resul
    when 4
        puts "Type yes to quit"
        quit_choice = gets.chomp.downcase
        break if quit_choice == "yes"
    when 5
        tossing_coin_loop = false
    else
        puts "Invalid input please try again"
    end
 end
```

Coding part ----- case statement and while loop

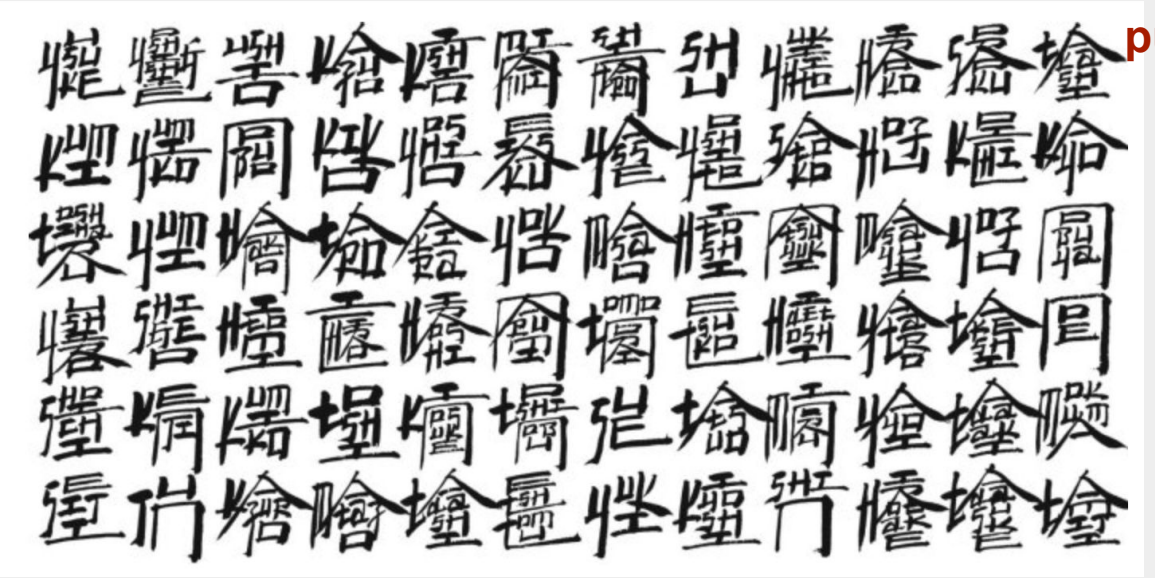
T1A3 - Terminal Application

def ChallengesEthicalIssuesFavouriteParts


```
challenge_array = [ " gem", "File I/O", unfinished_part,

                    "Learning language by second language"]


unfinished_part = "test, another gem, welcome promt…."

favourite_parts = {
                    1 : "First time use the gem and file system successfully"
                    2 : "Flowchart to make logical is right and smooth"
                    3 : "really enjoy to break up task and step by step to solve it"

                   }

end
```

def GreatThingsILearnedFromWork

T1A3 - Terminal Application



Xu Bing 1988 <***A Book from the Sky***>

puts “ First week learning ruby

When I read documentation pretty same with you read left picture

You don’t know any words on it.

so do I….

I find my own way to learn my first coding language, that is keep slow...”

end

T1A3 - Terminal Application

What I am achieve in this Application design?

List of features that will be included in the application

- Use of variables and the concept of variable scope -- I can use it, but still with a very simple way
- Loops and conditional control structures -- I can use it, but still need documentation
- File input / output -- God, I've done it, hope next time I still remember it.
- Error handling-- Still working on it

More haste less speed --- If you hasted, you can't eat hot Tofu.(Chinese proverb)


Thank you
for all your
HELP!