

</> CSC1024 PROGRAMMING PRINCIPLE

Programming Project: A Mastermind Computer Game

Student Name: Karen Ng Kai En

Student ID: 19043934

Programme: Bachelor of Software Engineering

Date: 26 November 2020

Presentation Video Web Link: <https://youtu.be/XDgiIh-NSCQ>

TABLE OF CONTENTS

01

Introduction

03

Design

02

Demonstration

04

Conclusion

01

INTRODUCTION

INTRODUCTION

- </> Recreating the mastermind game but on computer using python programming language.
- </> Mastermind Computer Game is a program to let user guess a secret code of 4 colors.
- </> A feedback will be given to the user as a hint to narrow down the possibilities of winning.
- </> This game require logic thinking and brain power to win which could improve analysis strategic and critical thinking skills.



02

DEMONSTRATION

```
/users/Karenng/MyncharProjects/testing/venv/bin/python /users/Karenng/MyncharProjects/testing/testing.py
```

* * * PYTHON MASTERMIND COMPUTER GAME * * *

-----Rules of this game -----

- > I'm thinking of a 4 color code. Try and guess what is it!
- > Try to guess the color in order as few times as possible.
- > You have only 10 attempts for each round.
- > You will get a hint after every guesses.
- > This game have 5 colors, Green, Blue, Yellow, White and Red
- > Please input G(Green), B(Blue), Y(Yellow), W(White) and R(Red).
- > Example Answer: GBGY

NOTED: The color may be repeated

Please key in your guessed color below

['G', 'R', 'B', 'B']

I

Enter your answer:

03

DESIGN

Interactive User Menu

User Guild Line

* * * PYTHON MASTERMIND COMPUTER GAME * * *

Rules of this game

- I'm thinking of a 4 color code. Try and guess what is it!
- Try to guess the color in order as few times as possible.
- You have only 10 attempts for each round.
- You will get a hint after every guesses.
- This game have 5 colors, Green, Blue, Yellow, White and Red
- Please input G(Green), B(Blue), Y(Yellow), W(White) and R(Red).
- Example Answer: GBGY

NOTED: The color may be repeated

Please key in your guessed color below

Enter your answer:

User Guild line was provided when the game had started.

The purpose of this guild line is to give a short brief of how this game looks like and the rules of this game before user input their answer.

Enter your answer: *qqqq*

You had entered the wrong color. We only have green(G), blue(B), yellow(Y), white(W) and red(R).

^ when user enter wrong color code

Enter your answer: *www*

You must enter four color of your guesses. Please try again :)

^ when user enter less than 4 color code

Enter your answer: *RBYG*

WOW! You guessed at the first attempt!

^ when user are correct at the first attempt

Number of attempts used: 6

You have 4 attempts left.

^ Reminder of the attempts used and left

Enter your answer: *gggg*

Oh No! You didn't guess it right. The secret color code was:['W', 'R', 'W', 'W'].

^ when user didn't manage to answer correctly within 10 attempts

Do you wish to play a new round? (Please reply YES or NO)

>> *yes*

^ ask user do they want to play a new round?



FEEDBACK  HINT

Generated Color Code: G, B, Y, Y

User Input: G, Y, Y, B

FEEDBACK:

2 Correct color and Right Position

2 Correct color but Wrong Position

List

```
# List of colors  
colors = ["G", "B", "Y", "W", "R"]  
attempts = 0  
game = True
```

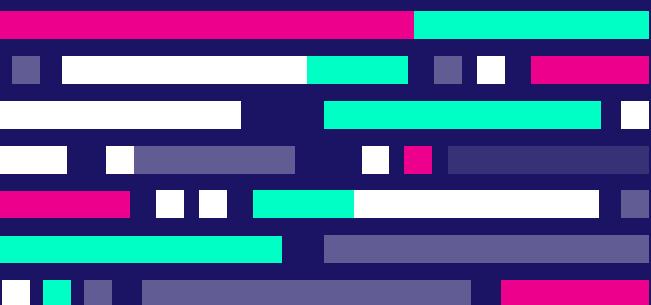
<< This is a list of colors available in the game.

```
# Randomise the color  
color_code = random.choices(colors, k=4)
```

<< Randomize 4 values from the list

- A list is a number of items in an ordered or unordered structure. A list can be used for a number of things like storing items or deleting and adding items.
- And for this program, list allow program to perform the different tasks such as generating randomize values from the list.
- Lists can be helpful for computational problems when different set of data were needed to be stored, access and compared with each other.

Generating Randomize Value



Generating random value in programming functions helps us generate new set of randomize code which is hard to generates manually every round because this game contains many possible orders.

```
import random # Importing random module from python library
```

Before that, I use *random.sample* to generate random color for user, but this code will generate colors without repetition.

```
# List of colors
colors = ["G", "B", "Y", "W", "R"]
attempts = 0
game = True

# Randomise the color
color_code = random.sample(colors, 4)
```

I use *random.choices* to generate random color code with 4 randomize color that have repetitions

```
# List of colors
colors = ["G", "B", "Y", "W", "R"]
attempts = 0
game = True

# Randomise the color
color_code = random.choices(colors, k=4)
```

« Revised Version

```

# During the game (Check player's input)
if len(player_guess) != len(color_code):
    print("\n You must enter four color of your guesses. Please try again :)")
    continue
for i in range(4):
    if player_guess[i] not in colors:
        print("\nYou had entered the wrong color. We only have green(G), blue(B), yellow(Y), white(W) and red(R).")
        print("Please try again :)")
        break

```

For example the player enter unknown character as the guess such as symbols of number, the player will be asked to input again.

```

# If user guessed it correctly
if right_ans() == 4:
    if attempts == 1:
        print("WOW! You guessed at the first attempt!")
    else:
        print("Well Done.. You needed " +str(attempts) + " attempts to guess.")
game = False

# If user didn't guessed it correctly below 10 attempts
if attempts >= 1 and attempts < 10 and right_ans() != 4:
    feedback(right_ans(),wrong_ans())
    print("Number of attempts used: " + str(attempts))
    print("You have", int(10 - attempts), "attempts left.\n")

# If user didn't guessed it correctly after 10 attempts
elif attempts >= 10:
    print("Oh No! You didn't guess it right. The secret color code was:" + str(color_code) +".")
game = False

```

Example 2:

- If the user guessed it correctly in the first attempts, the program will print "Wow! You guessed at the first attempt!"
- And when user guessed it correctly or didn't guessed it correctly , within 10 attempts, the **program will make suitable decision and execution for each situation**

Decision Making

If-Else Statement

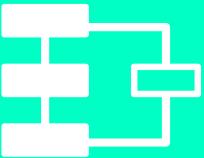
Decision making is applied by using if-else statement. This is very useful when there is many potential condition and statements. By using if-else statements, it allow python to check the condition is true or false before the executing the instruction and which statement should be executed.



Iterative Process

While Loops

After the game



```
# Continue or not?
while game == False:
    end_game = input("\nDo you wish to play a new round? (Please reply YES or NO)\n> ").upper()
    attempts = 0

    if end_game == "NO":
        print("Thanks for playing the game! Bye Bye !")
        break

    elif end_game == "YES":
        game = True

        color_code = random.choices(colors, k=4)
        print("So, lets play again... Guess the secret code:")
        print("Please key in your guessed color below\n")
        print(color_code) # Delete this if you don't want answer to be display.
        print("")

    << Program will end the game if user input "No"
```

<< The program will ask user whether they want to play a new round after every round

```
<< Program will generate new set of color code for every new round
```

```
# Let player guess the number
while game:
    player_guess = list(input("Enter your answer: ").upper())
    attempts += 1
```

Repetitive computational problem where multiple time of checking is needed will need iterative process. The loop will iterate while the condition is true. When the game is True, the program will run all the code after the condition unless user input "No" and the program will terminate

User Define Function

Compare

For example, the condition if `player_guess` is equal to `color_code`, then the variable "right" will plus 1 , and the final value is then return and store in the variable "right". And the same for `def wrong_ans ()`:

Feedback

The variable right and wrong which had been assigned previously will then used in `def feedback(right, wrong)`:

```
# Compared player guesses with computer guesses
```

```
# Determine the right color and right position
def right_ans():
    right = 0
    for i in range(4):
        if player_guess[i] == color_code[i]:
            right += 1
    return right
```

<< Compare Position

```
# Determine right color but wrong position
def wrong_ans():
    check_ans = color_code[:]
    wrong = 0
    for i in range(4):
        if player_guess[i] in check_ans:
            check_ans.remove(player_guess[i])
            wrong += 1
    return wrong
```

Right Position

Wrong Position

```
# Feedback (Hint) to user at every guesses.
```

```
def feedback(right, wrong):
```

```
    feedback1 = str(right) + " Correct Color and Right Position "
    feedback2 = str(wrong - right) + " Correct Color but Wrong Position "
    feedback = feedback1 + "\n" + feedback2 + "\n"
    return print(feedback)
```

After processing this user define function, the value will return and print it out for user as a feedback for this game

User define function that had been written before can be call out every time instead of writing the whole piece of code again.

Conclusion & Improvement

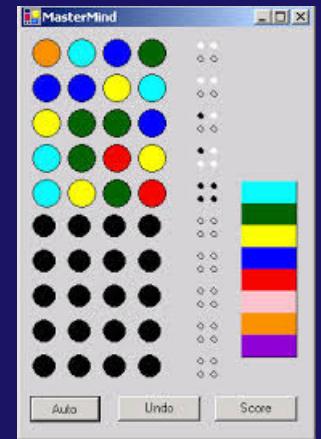
Conclusion

I had learned so much about python programming throughout the semester.
Other than that, I had applied all my programming knowledge, time and efforts in this Mastermind Game Project.

Although, I had made a lot of errors in the beginning, but I managed to create a complete version of Mastermind Computer Game that is zero-error after several times of debugging and enhancing.

Improvement

For further improvement, I wish that I can create a 2-player version of mastermind game.
And a version of mastermind game with Graphic User Interface(GUI) which will make this game looks better visually.





THANKS!

19043934@imail.sunway.edu.my
Karen Ng Kai En
19043934