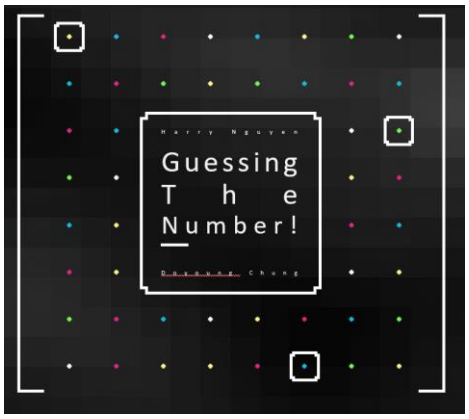
Guessing The Number



BICC Birmingham International Collegiate of Canada

INTRODUCTION

The game “Guessing the number” is literally about guessing the number. The game is suitable for two mature friends to strengthen their relationship without being aimed at two kids. Also, the game “Guessing the number” would be a great source to develop early learning skills for younger children. While guessing the number, children can improve their level of detection ability, intelligence, cognitive, memory, and even critical thinking with given hints. It seems to be a simple game, but it includes a high level of competition between two players and critical thinking.

TABLE OF CONTENT

Chapter 1: Requirement analysis (5 pages)

Chapter 2: Feasibility Study and challenges (3 pages)

Chapter 3: Design diagrams

Chapter 4: Code implementation

Chapter 5: Testing

Chapter 6: Help guide

Chapter 7: Conclusion

Chapter 1: SRSD

[Doyoung & Harrry corp]

Guessing the number

Software Requirements Specification Document

Doyoung Chung/Harry Nguyen

3/16/2021

Contents

1 INTRODUCTION 2
1.1 DOCUMENT PURPOSE..... 2
1.2 PRODUCT SCOPE..... 2
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW.....2
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....2
1.5 DOCUMENT CONVENTIONS..... 2
1.6 REFERENCES AND ACKNOWLEDGMENTS..... 3

2 OVERALL DESCRIPTION.....3
2.1 PRODUCT PERSPECTIVE 3
2.2 PRODUCT FUNCTIONALITY 3
2.3 USERS AND CHARACTERISTICS 3
2.4 OPERATING
ENVIRONMENT.....3
2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS 3
2.6 USER
DOCUMENTATION…..... 3
2.7 ASSUMPTIONS AND DEPENDENCIES..... 3

3 SPECIFIC REQUIREMENTS..... 4
3.1 EXTERNAL INTERFACE REQUIREMENTS..... 4
3.2 FUNCTIONAL REQUIREMENTS.....4

4 OTHER NON-FUNCTIONAL REQUIREMENTS.....4
4.1 PERFORMANCE REQUIREMENTS 4

1. INTRODUCTION

The following subsections of the Software Requirements Specifications (SRS) document provide an overview of the entire SRS.

1.1 DOCUMENT PURPOSE

The purpose of this document is to show the software requirements of the game named "Guessing the number". The functionality and scope of this game are described in this SRS document.

1.2 PRODUCT SCOPE

The game "Guessing the number" aims at helping the user to feel happy and relax. Nowadays people are too busy to make a relationship, but the game can help you to make a relationship while playing the game. It contains the level of competition and achievement. Ultimately, we can expect increases in the level of cognitive, intelligence, and critical thinking. Additionally, the game shows the incredible effectiveness of killing time.

1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW

This SRS document is intended for developers , professors, and students for reading. The rest of the document contains the functional and nonfunctional requirements of the game "Guessing the number".

1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

SRS- Software Requirement Specification.
GTS – Guessing The Number
Servers: Machines that store all the information and records.

1.5 DOCUMENT CONVENTIONS

The entire document is in Times New Roman font. The headings are numbered 1,2,3... and so on and sub-headings are numbered x.1,x.2.... and so on. Both headings and subheadings are in bold.

Main title : Font Times New Roman and size 36
Subtitles : Font Times New Roman and size 21.5
Content : Font Times New Roman and size 12

1.6 REFERENCES AND ACKNOWLEDGMENTS

Ms.Garima – given SRS template

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

The game “Guessing the number” is literally about guessing the number. The game is suitable for two mature friends to strengthen their relationship without being aimed at two kids. Also, the game “Guessing the number” would be a great source to develop early learning skills for younger children. While guessing the number, children can improve their level of detection ability, intelligence, cognitive, memory, and even critical thinking with given hints. It seems to be a simple game, but it includes a high level of competition between two players and critical thinking.

2.2 PRODUCT FUNCTIONALITY

Some major product functionalities of the system are as follows:
-Information about the process
-Information about the answer
-Show whether the number is correct or not

2.3 USERS AND CHARACTERISTICS

Primary users of the system will be employees working in company /students /staff, manager , HOD, Admin. Very little technical expertise is required for reading the outputted data since it is in graphical/tabular form.
Educational level of LMS computer software – Low
Experience of LMS software – None
Technical Expertise – Little

2.4 OPERATING ENVIRONMENT

Google colab

2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

The design of the game is built in google colab. The design is simple and straight.

2.6 USER DOCUMENTATION

A link is provided for help and a very easy User Interface.

2.7 ASSUMPTIONS AND DEPENDENCIES

Assume that all the information entered by the user will be whether wrong or correct. If any wrong information is found then the system will tell you “the number is wrong” and return the value.

3. SPECIFIC REQUIREMENTS

3.1 External Interface Requirements

Google Colab is used as the interface of the game.

3.2 Functional Requirements

- Take input function
- Function to calculate the hints
- Loops
- Function to reply appropriate message

4. NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements define the needs in terms of performance, availability, maintainability, and bug-free.

4.1 PERFORMANCE REQUIREMENTS

Performance requirements define acceptable response times for system functionality.

- Fast loading time
- The system shall consume very little of primary memory

Chapter 2: Feasibility Study ang challenges

1. Finalized requirement

- The game should have no bugs and run smoothly
- Make the Hint() function

```
def add(m):
  global x
  for i in range(len(str(m))):
    x+=m%10
    m = m//10
  print("The sum of terms is",x)
```

```
def even_or_odd(m):
  global e
  global o
  for i in range(len(str(m))):
    t=m%10
    m = m//10
    if t%2==1:
      o+=1
    else:
      e+=1
  print("The number have", o, "odd number",". The number have", e,"even number")
```

```
def mul_fal(m):
  for i in range(len(str(m))):
    mul_fal_list.append(m%10)
    m = m//10
  print("the multiplicaation of the first and the last number is", mul_fal_list[0]*mul_fal_list[-1])
```

```
def add_2l(m):
  for i in range(len(str(m))):
    add_2l_list.append(m%10)
    m = m//10
  print("the sum of two last numbers is", add_2l_list[0] + add_2l_list[1])
```

- Make the Take Input() function

```
found=0
def take_input():
  input_sn_list=[]
  a=0
  global y
  global found
  ca=input_number=int(input("Enter your guess: "))      #Take input
  y+=1
  m=10-y
  for u in range(len(str(input_number))):               #Create list of input
    input_sn_list.append(ca%10)
    ca = ca//10
  for i in range(g):
    for u in range(len(str(input_number))):             #Reply appropriate message
      if number_list[i]==input_sn_list[u]:
        if i == u:
          a+=1
  if input_number == n:                                 #Reply answer
    print("You won in", y, "guesses")
    found=1
  else:
    print("You have",a,"number in the correct position")
    print("You have",m,"more guess")
    found=0
```

- Make the main loop

```
take_input()                #First loop len
if found == 0:
  print("-------------------------------------------------------------")
  add(n)
  take_input()              #Second loop add
  if found ==0:
    print("-------------------------------------------------------------")
    even_or_odd(n)
    take_input()            #Third loop even and odd
    if found ==0:
      print("-------------------------------------------------------------")
      mul_fal(n)
      take_input()          #Forth loop the multiplication of the first and the last number in the serial number
      if found ==0:
        print("-------------------------------------------------------------")
        add_2l(n)
        take_input()        #Fifth loop the sum of 2 last number
        if found ==0:
          print("-------------------------------------------------------------")
          take_input()      #Sixth loop
          if found ==0:
            print("-------------------------------------------------------------")
            take_input()    #Seventh loop
            if found ==0:
              print("-------------------------------------------------------------")
              take_input()  #Eighth loop
              if found ==0:
                print("-------------------------------------------------------------")
                take_input()  #Nineth loop
                if found ==0:
                  print("-------------------------------------------------------------")
                  take_input()  #Tenth loop
                  if found ==0:
                    print("The number is",n,". Wish you luck next time!")
```

2. Challenges encountered

- Global and Local variable challenge (Decide variable should be Local or Global for the code to work).

- Global and Local challenge can be solved by understand the main purpose of the variable and their continuity to use or reuse in the code.

```
found=0
input_sn_list=[]
def take_input():
  a=0
  global y
  global found
  ca=input_number=int(input("Enter your guess: "))      #Take input
  y+=1
  m=10-y
  for u in range(len(str(input_number))):               #Create list of input
    input_sn_list.append(ca%10)
    ca = ca//10
  for i in range(g):
    for u in range(len(str(input_number))):             #Reply appropriate message
      if number_list[i]==input_sn_list[u]:
        if i == u:
          a+=1
  if input_number == n:                                 #Reply answer
    print("You won in", y, "guesses")
    found=1
  else:
    print("You have",a,"number in the correct position")
    print("You have",m,"more guess")
    found=0
```

```
----------------------------------------------------------------
your number have 5 charactors
Enter your guess: 12345
You have 0 number in the correct position
You have 9 more guess
----------------------------------------------------------------
The sum of terms is 18
Enter your guess: 45450
You have 0 number in the correct position
You have 8 more guess
----------------------------------------------------------------
The number have 4 odd number . The number have 1 even number
Enter your guess: 13572
You have 0 number in the correct position
You have 7 more guess
----------------------------------------------------------------
the multiplicaation of the first and the last number is 18
Enter your guess: 99132
You have 0 number in the correct position
You have 6 more guess
----------------------------------------------------------------
the sum of two last numbers is 10
Enter your guess: 23119
You have 0 number in the correct position
You have 5 more guess
----------------------------------------------------------------
Enter your guess: 61173
You won in 6 guesses
```

```
found=0
def take_input():
  input_sn_list=[]
  a=0
  global y
  global found
  ca=input_number=int(input("Enter your guess: "))      #Take input
  y+=1
  m=10-y
  for u in range(len(str(input_number))):               #Create list of input
    input_sn_list.append(ca%10)
    ca = ca//10
  for i in range(g):
    for u in range(len(str(input_number))):             #Reply appropriate message
      if number_list[i]==input_sn_list[u]:
        if i == u:
          a+=1
  if input_number == n:                                 #Reply answer
    print("You won in", y, "guesses")
    found=1
  else:
    print("You have",a,"number in the correct position")
    print("You have",m,"more guess")
    found=0
```

```
----------------------------------------------------------------
your number have 5 charactors
Enter your guess: 12345
You have 0 number in the correct position
You have 9 more guess
----------------------------------------------------------------
The sum of terms is 18
Enter your guess: 45450
You have 0 number in the correct position
You have 8 more guess
----------------------------------------------------------------
The number have 4 odd number . The number have 1 even number
Enter your guess: 13572
You have 1 number in the correct position
You have 7 more guess
----------------------------------------------------------------
the multiplicaation of the first and the last number is 18
Enter your guess: 91132
You have 2 number in the correct position
You have 6 more guess
----------------------------------------------------------------
the sum of two last numbers is 10
Enter your guess: 23119
You have 1 number in the correct position
You have 5 more guess
----------------------------------------------------------------
Enter your guess: 61173
You won in 6 guesses
```

- Loop challenge. Python does not understand which function is true or false.
- Loop challenge can be solved by declared a variable represents for the true and false statement and after each loop the variable may change

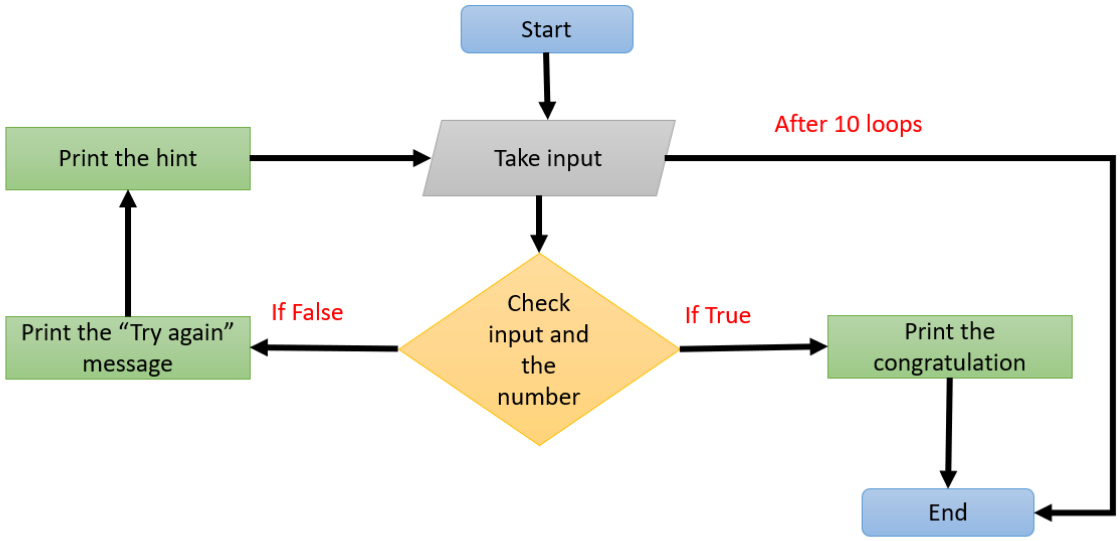
```
print("----------------------------------------------------------")
print("your number have", len(str(n)),"charactors")
take_input()
if take_input is True:
  print ("You won")
  break
if take_input is False :
  print("----------------------------------------------------------")
  print("your number have the sum of terms is", add(n))
  take_input()
  if take_input is True:
    print ("You won")
    break
  if take_input is False :
    print("----------------------------------------------------------"
    even_or_odd(n)
```

```
take_input()                    #First loop len
if found == 0:
  print("----------------------------------------------------------")
  add(n)
  take_input()                  #Second loop add
  if found ==0:
    print("----------------------------------------------------------")
    even_or_odd(n)
    take_input()                #Third loop even and odd
    if found ==0:
      print("----------------------------------------------------------")
      mul_fal(n)
      take_input()              #Forth loop the multiplication of the first and the last number in the serial number
      if found ==0:
        print("----------------------------------------------------------")
        add_2i(n)
        take_input()            #Fifth loop the sum of 2 last number
        if found ==0:
          print("----------------------------------------------------------")
          take_input()          #Sixth loop
          if found ==0:
            print("----------------------------------------------------------")
            take_input()        #Seventh loop
            if found ==0:
              print("----------------------------------------------------------")
              take_input()      #Eighth loop
              if found ==0:
                print("----------------------------------------------------------")
                take_input()    #Nineth loop
                if found ==0:
                  print("----------------------------------------------------------")
                  take_input()  #Tenth loop
                  if found ==0:
                    print("The number is",n,". Wish you luck next time!")
```

- Wrong hint challenge.
- Wrong hint challenge can be solve by testing and rechecking the code.

Chapter 3: Design diagrams

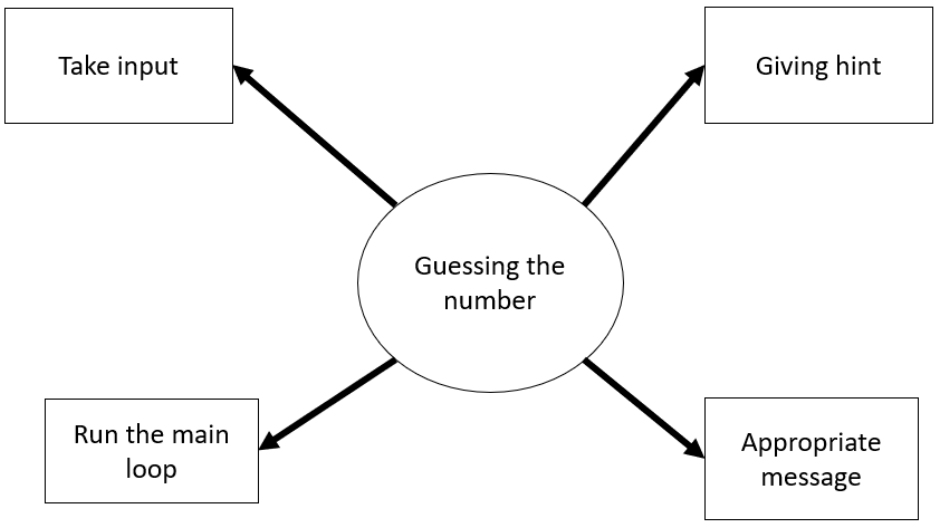
Flowchart diagram



Start the program => Taking input from the user => put the in put into a loop

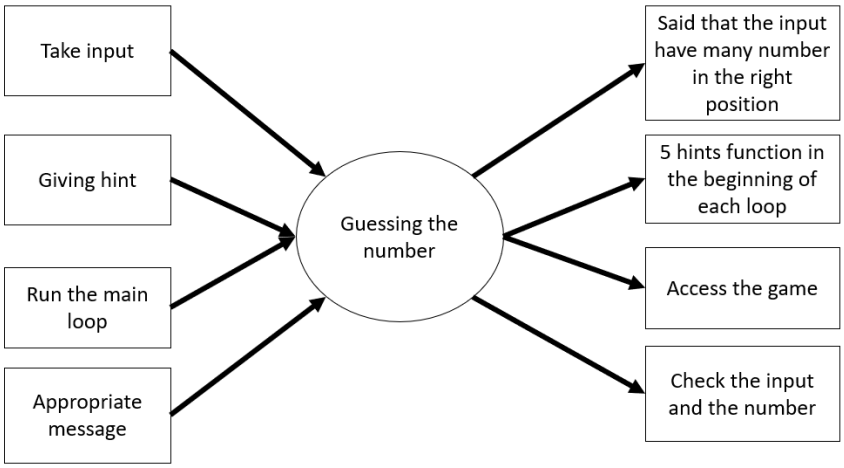
- ⇨ If true print the congratulation message => End the program
- ⇨ If False return into the loop => After 10 loops => End the program

DFD diagram



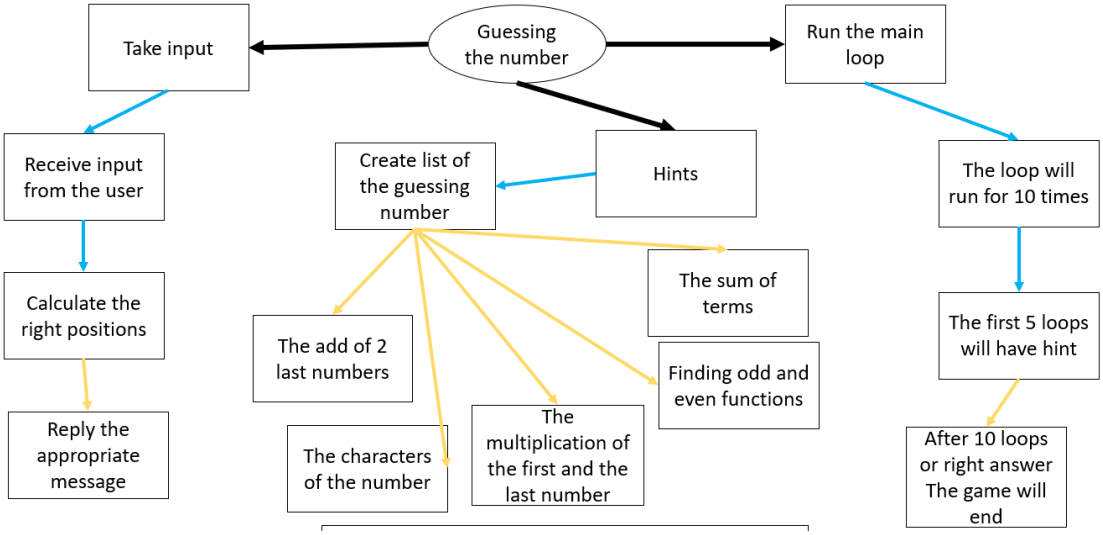
Zero level DFD – Guessing the number

Zero level DFD have 4 categories is Take input, Giving hint, Run the main loop, and Appropriate message



First level DFD – Guessing the number

First level DFD have more function about each category
Take input => reply the hint message
Giving hint => reply 5 hints
Run the main loop => Access the game
Appropriate message => Check and reply the message



Second level DFD – Guessing the number

It contains more function and step

Take input => Function to calculate the position of the number => reply message

Hints => Compare the input number with the guessing number

The main loop => Access the game and rules

Chapter 4: Code implementation

1.Python concept

- Game

2. Functions and libraries

2.1 Function

- print()
- len()

Hint functions:

- add() – the sum of terms

```
def add(m):
  global x
  for i in range(len(str(m))):
    x+=m%10
    m = m//10
  print("The sum of terms is",x)
```

- even_or_odd() – find the number of even and odd number

```
def even_or_odd(m):
  global e
  global o
  for i in range(len(str(m))):
    t=m%10
    m = m//10
    if t%2==1:
      o+=1
    else:
      e+=1
  print("The number have", o, "odd number",". The number have", e,"even number")
```

- mul_fal() – find the multiplication of the first and the last numbers

```
def mul_fal(m):
  for i in range(len(str(m))):
    mul_fal_list.append(m%10)
    m = m//10
  print("the multiplicaation of the first and the last number is", mul_fal_list[0]*mul_fal_list[-1])
```

- add_2l()

```
def add_2l(m):
  for i in range(len(str(m))):
    add_2l_list.append(m%10)
    m = m//10
  print("the sum of two last numbers is", add_2l_list[0] + add_2l_list[1])
```

- Take_input() – take input and compare them with the guessing number

```
found=0
def take_input():
  input_sn_list=[]
  a=0
  global y
  global found
  ca=input_number=int(input("Enter your guess: "))      #Take input
  y+=1
  m=10-y
  for u in range(len(str(input_number))):               #Create list of input
    input_sn_list.append(ca%10)
    ca = ca//10
  for i in range(g):
    for u in range(len(str(input_number))):             #Reply appropriate message
      if number_list[i]==input_sn_list[u]:
        if i == u:
          a+=1
  if input_number == n:                                 #Reply answer
    print("You won in", y, "guesses")
    found=1
  else:
    print("You have",a,"number in the correct position")
    print("You have",m,"more guess")
    found=0
```

2.2 Libraries

- Random

```
from random import randint
n = randint(1000,100000)
```

CHAPTER 5: TESTING

```
#Create random number

from random import randint
n = randint(1000,100000)
```

- **set the range of given random number in case the number is too b ig to guess it**

```
q=n
x=0
y=0
e=0
o=0
t=0
```

- declare various variables to organize the code without any error

```
y+=1
m=10-y
for u in range(len(str(input_number))):                #Create list of input
  input_sn_list.append(ca%10)
  ca = ca//10
for i in range(g):
  for u in range(len(str(input_number))):              #Reply appropriate messa
    if number_list[i]==input_sn_list[u]:
      if i == u:
        a+=1
if input_number == n:                                  #Reply answer
  print("You won in", y, "guesses")
  found=1
else:
  print("You have",a,"number in the correct position")
  print("You have",m,"more guess")
  found=0
```

- In case the player can play the ga me with out any limits, set the limit of shots.

```python
#Hint 1:

def add(m):
  global x
  for i in range(len(str(m))):
    x+=m%10
    m = m//10
  print("The sum of terms is",x)

#Hint 2:

def even_or_odd(m):
  global e
  global o
  for i in range(len(str(m))):
    t=m%10
    m = m//10
    if t%2==1:
      o+=1
    else:
      e+=1
  print("The number have", o, "odd number",". The number have", e,"even number")

#Hint 3:

def mul_fal(m):
  for i in range(len(str(m))):
    mul_fal_list.append(m%10)
    m = m//10
  print("the multiplicaation of the first and the last number is", mul_fal_list[0]*mul_fal_list

#Hint 4:

def add_2l(m):
  for i in range(len(str(m))):
    add_2l_list.append(m%10)
    m = m//10
  print("the sum of two last numbers is", add_2l_list[0] + add_2l_list[1])
```

- Given hints, could be possible to guess the number within shots.

```
---------------------------------------------------------------------
your number have 5 charactors
Enter your guess: 39232
You have 1 number in the correct position
You have 9 more guess
---------------------------------------------------------------------
The sum of terms is 21
Enter your guess: 13833
You have 0 number in the correct position
You have 8 more guess
---------------------------------------------------------------------
The number have 1 odd number . The number have 4 even number
Enter your guess: 47214
You have 1 number in the correct position
You have 7 more guess
---------------------------------------------------------------------
the multiplicaation of the first and the last number is 36
Enter your guess: 59321
You have 1 number in the correct position
You have 6 more guess
---------------------------------------------------------------------
the sum of two last numbers is 8
Enter your guess: 39251
You have 1 number in the correct position
You have 5 more guess
---------------------------------------------------------------------
Enter your guess: 39283
You have 1 number in the correct position
You have 4 more guess
---------------------------------------------------------------------
Enter your guess: 30021
You have 1 number in the correct position
You have 3 more guess
---------------------------------------------------------------------
Enter your guess: 39214
You have 1 number in the correct position
You have 2 more guess
---------------------------------------------------------------------
Enter your guess: 02124
You have 1 number in the correct position
You have 1 more guess
---------------------------------------------------------------------
Enter your guess: 20141
You have 0 number in the correct position
You have 0 more guess
The number is 65226 . Wish you luck next time!
```

- Diagram : output when we are out of 10 guess chancees

Chapter 6: Help guide

How to run(play) our code?

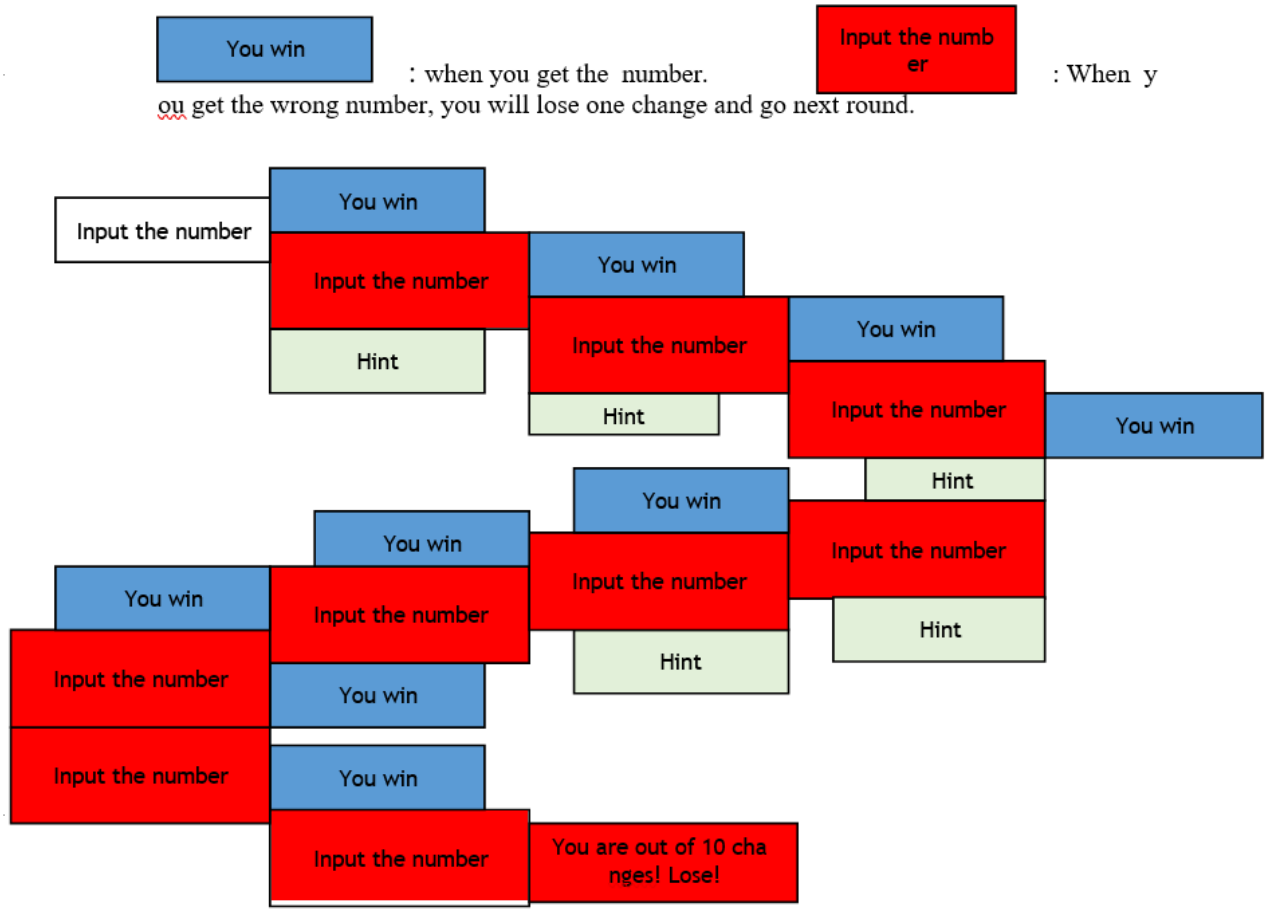
- Our product 'Guessing The Number" is basically a fun educational game that challenges kids to find a number based on hints and given feedback. Therefor e, since the guideline is given, what players have to do is to input the numbe r(guess) based on the guideline. Player's guessing chances are limited as 10.

Can We Play Together?

- 'Guessing The Number" can be played with your friends or even your parents. Rather, We are looking forward to playing together.

The Program Flow

Basically the code will not end until you got the number. But if you are out of 10 chances, you will lose. Also, as we proceed, hints are given (not more than 5). Below is the basic process of the program



CHAPTER 7 : CONCLUSION

What have we achieved?

- What we aimed at is to make a game which can be played by all ages, especially university students or businessmen . Nowadays, people are too busy to do something else even without being businessmen or university students. People only care about their businesses. It seems that all of those are so overwhelmed by so many things. So, the ultimate purpose of the game is that it can be played anytime for a while. We are delighted that it seems “Guessing The Number” is able to make it.

- When we came up with the idea, we thought that the code would be really easy and simple. However, the code was complex and harder than we thought it would be. But we were able to make the perfect code to implement it. We've handled edge cases , errors, and etc. We believe that what we did itself(making the game) was in the sense of achievement.

What have we not achieved?

- We originally wanted to make the game in a sort of software including the button and the fancy design. But the problem is that we do not have enough technology and knowledge to handle them. So, we had decided just to make it as a python code. But we believe that one day we are able to make it once we’ve learned the software.

Final Project Report Rubrics

Achievement category	Marks Given	50–59% (Level 1)	60–69% (Level 2)	70–79% (Level 3)	80–100% (Level 4)
Knowledge & Understanding - The student demonstrates a thorough understanding of the concepts related to python programming and software development in the report.	/20	Demonstrates limited understanding of python programming and software development concepts in the report.	Demonstrates some understanding of python programming and software development concepts in the report.	Demonstrates a considerable understanding of the python programming and software development concepts in the report.	Demonstrates a high degree of understanding of the python programming and software development concepts in the report.
Thinking – The student is able to come up with a creative project idea and present the idea with the devised solution in project report	/20	Demonstrates limited ability in presenting the project idea and its solution in a creative way in the project report.	Demonstrates some critical understanding in presenting the project idea and its solution in a creative way in the project report.	Demonstrates a considerable understanding in presenting the project idea and its solution in a creative way in the project report.	Demonstrates a high degree of critical understanding in presenting the project idea and its solution in a creative way in the project report.
Communication - follows the project report template and all the chapters should be well structured and clearly written. Design diagrams should be in proper format	/40	Demonstrates limited ability to follow the project report template and structure it. Design diagrams also created with limited effectiveness.	Demonstrates some ability to follow the project report template and structure it. Design diagrams were also created with some effectiveness.	Demonstrates considerable ability to follow the project report template and structure it. Design diagrams were also created with considerable effectiveness.	Demonstrates a high degree of understanding, follows the project report template and structures it. Design diagrams also created with effectiveness.
Application – The student applies their understanding of python programming with software development concepts in their project report	/20	Apply knowledge of python programming and software development concepts with limited effectiveness.	Apply knowledge of python programming and software development concepts with some effectiveness.	Apply knowledge of python programming and software development concepts with considerable effectiveness.	Apply knowledge of python programming and software development concepts with a high degree of effectiveness.

Comments :