

# **Geetanjali Institute of Technical Studies, Udaipur**



*An*

***Internship Report***

*submitted to*

**Department of Computer Science & Engineering**

Submitted By:

Mayank Joshi

Roll No.: 19EGICS065

**Internship Carried Out**

**at**

**Department of Computer Science & Engineering**

**Geetanjali Institute of Technical Studies**

**Rajasthan Technical University**

**July, 2020**

## **Vision of Department of Computer Science & Engineering**

To nurture the students to become employable graduates who can provide solutions to the societal issues through ICT.

## **Mission of Department of Computer Science & Engineering**

**M1.** To focus on practical approach towards learning and exposing the students on the latest ICT technologies.

**M2.** To foster logical thinking among the students to solve real-time problems using innovative approaches.

**M3.** To provide state-of-the-art resources that contributes to inculcate ethical & life-long learning environment.

## Candidate's Declaration

*I hereby declare that the work, which is being presented in the Internship Report, entitled "Color Changing using tkinter module in python" in partial fulfilment for the award of Degree of "Bachelor of Technology" in Deptt. of Computer Science & Engineering, and submitted to the Department of Computer Science & Engineering, Geetanjali Institute of Technical Studies, Rajasthan Technical University is a record of my own investigations carried at Name of organization / Company & address /online platform..*

*I have not submitted the matter presented in this Project Report anywhere for the award of any other Degree.*



Mayank Joshi

(Name and Signature of Candidate)

Computer Science Engineering

Roll No.: 19EGICS065

Geetanjali Institute of Technical Studies

# **Internship Report 2021**

## **Student Details:**

**Name:** Mayank Joshi

**Branch:** CSE

**Year:** 3<sup>rd</sup> Year

**Section:** B

**Roll no.:** 19EGICS065

## **Table of Contents**

| <b>Chapter</b> | <b>Content</b>                                | <b>Page No.</b> |
|----------------|---|-----------------|
| Chapter 1      | Title & Student Detail                        | 1-3             |
| Chapter 2      | Table of Contents                             | 4               |
| Chapter 3      | Project title                                 |                 |
| Chapter 4      | Problem statement                             | 16              |
| Chapter 5      | Objective                                     | 17              |
| Chapter 6      | Introduction to project (project description) | 17              |
| Chapter 7      | Language used (technology stack)              | 18              |
| Chapter 8      | Literature Review                             | 19-21           |
| Chapter 9      | Module Flow Diagram & UML diagram             | 21-23           |
| Chapter 10     | Implementation of module with methodology     | 24              |
| Chapter 11     | Result & testing                              | 25              |
| Chapter 12     | Conclusion                                    | 26              |
| Chapter 13     | References                                    | 27              |

## PROJECT TITLE

### COLOR CHANGING GAME USING TKINTER MODULE IN PYTHON



## **Problem Statement**

The game sequence. Each turn is composed of five steps: To select one coloured peg, the player clicks one of the circles and chooses the desired colour from the table of colours. When a colour is selected (clicked), the circle is replaced by a peg of the desired colour.

Let's try to make a game using **Tkinter**. In this game player has to enter color of the word that appears on the screen and hence the score increases by one, the total time to play this game is 30 seconds. Colors used in this game are Red, Blue, Green, Pink, Black, Yellow, Orange, White, Purple and Brown

## **Objective**

TKinter is widely used for developing GUI applications. Along with applications, we can also use Tkinter GUI to develop games.

Let's try to make a game using Tkinter. In this game player has to enter color of the word that appears on the screen and hence the score increases by one, the total time to play this game is 30 seconds. Colors used in this game are Red, Blue, Green, Pink, Black, Yellow, Orange, White, Purple and Brown. Interface will display name of different colors in different colors. Player has to identify the color and enter the correct color name to win the game.



## **Introduction to project**

**Color changing game using tkinter module in python.**

**In this game player has to enter color of the word that appears on the screen and hence the score increases by one, the total time to play this game is 30 seconds. Colors used in this game are Red, Blue, Green, Pink, Black, Yellow, Orange, White, Purple and Brown. Interface will display name of different colors in different colors. Player has to identify the color and enter the correct color name to win the game**

**Need of the project Because of coding too much and stuck much time on screen. Our mind is frustrated so this is the main reason for making this game to reduce the stress level of mind.**

# Language used: Python

- Python is a widely used general-purpose, high-level programming language.
- It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation.
- It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.
- In 1999, Guido van Rossum defined his goals for Python:
  - an easy and intuitive language just as powerful as those of the major competitors;
  - open source, so anyone can contribute to its development;
  - code that is as understandable as plain English;
  - suitable for everyday tasks, allowing for short development times.

## Why Python?

What makes Python so special? How does it happen that programmers, young and old, experienced and novice, want to use it? How did it happen that large companies adopted Python and implemented their flagship products using it?

There are many reasons – we’ve listed some of them already, but let’s enumerate them again in a more practical manner:

it’s easy to learn – the time needed to learn Python is shorter than for many other languages; this means that it’s possible to start the actual programming faster;

it’s easy to teach – the teaching workload is smaller than that needed by other languages; this means that the teacher can put more emphasis on general (language-independent) programming techniques, not wasting energy on exotic tricks, strange exceptions and incomprehensible rules;

it’s easy to use for writing new software – it’s often possible to write code faster when using Python;

it’s easy to understand – it’s also often easier to understand someone else’s code faster if it is written in Python;

## **Python in Action**

It's used extensively to implement complex Internet services like search engines, cloud storage and tools, social media and so on. Whenever you use any of these services, you are actually very close to Python, although you wouldn't know it.

Many developing tools are implemented in Python.

More and more everyday use applications are being written in Python. Lots of scientists have abandoned expensive proprietary tools and switched to Python.

Lots of IT project testers have started using Python to carry out repeatable test procedures. The list is long.

## **Feature of Python**

### **1. Easy to code:**

Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, JavaScript, Java, etc. It is very easy to code in python language and anybody can learn python basics in a few hours or days. It is also a developer-friendly language.

### **2. Free and Open Source:**

Python language is freely available at the official website and you can download it from the given download link below click on the Download Python keyword. Since it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

### **3. Object-Oriented Language:**

One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation, etc.

#### 4. GUI Programming Support:

Graphical User interfaces can be made using a module such as PyQt5, PyQt4, python, or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.

#### 5. High-Level Language:

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

#### 6. Python is Portable language:

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

#### 7. Python is Integrated language:

Python is also an integrated language because we can easily integrated python with other languages like c, c++, etc.

#### 9. Interpreted Language:

Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

#### 10. Large Standard Library

Python has a large standard library which provides a rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers, etc.

## 11. Dynamically Typed Language:

Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

# Literature Review

## 1. Python

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages.

Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, Data science etc. Python has a lot of libraries for every need of this project. For JARVIS, libraries used are speech recognition to recognize voice, Pyttsx3 for text to speech, selenium for web automation etc.

Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time, and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

## 2. TKINTER MODULE

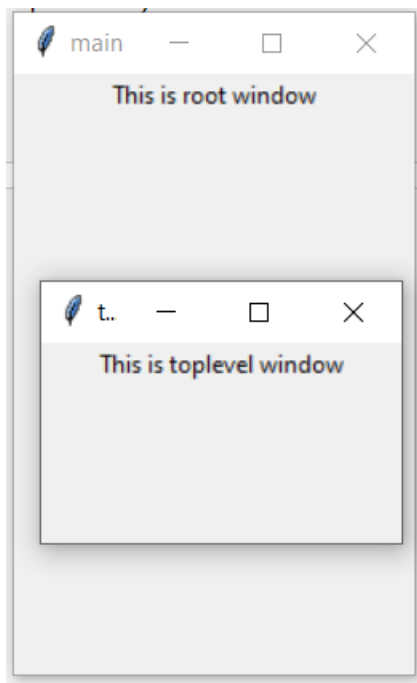
Tkinter is **the standard GUI library for Python**. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Import the Tkinter module.



## Small Activity using Tkinter

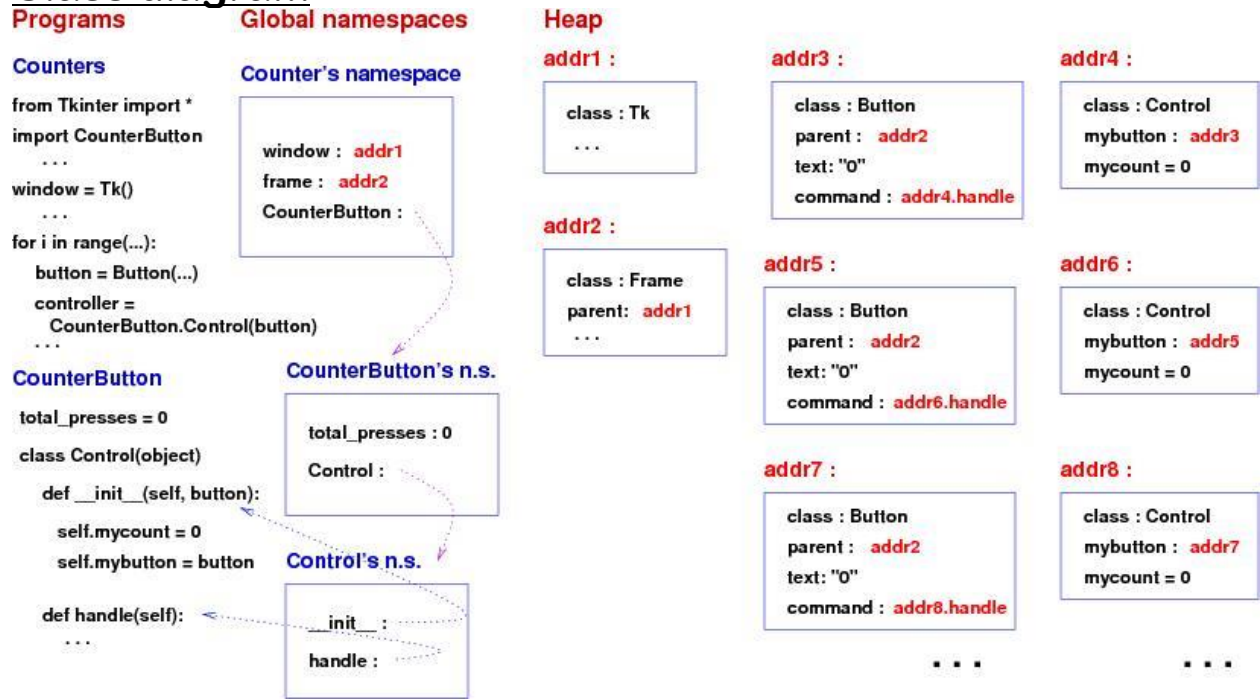
```
from tkinter import *  
root = Tk()  
root.geometry("200x300")  
root.title("main")  
l = Label(root, text = "This is root window")  
top = Toplevel()  
top.geometry("180x100")  
top.title("toplevel")  
l2 = Label(top, text = "This is toplevel window")  
l.pack()  
l2.pack()  
top.mainloop()
```

Output :-

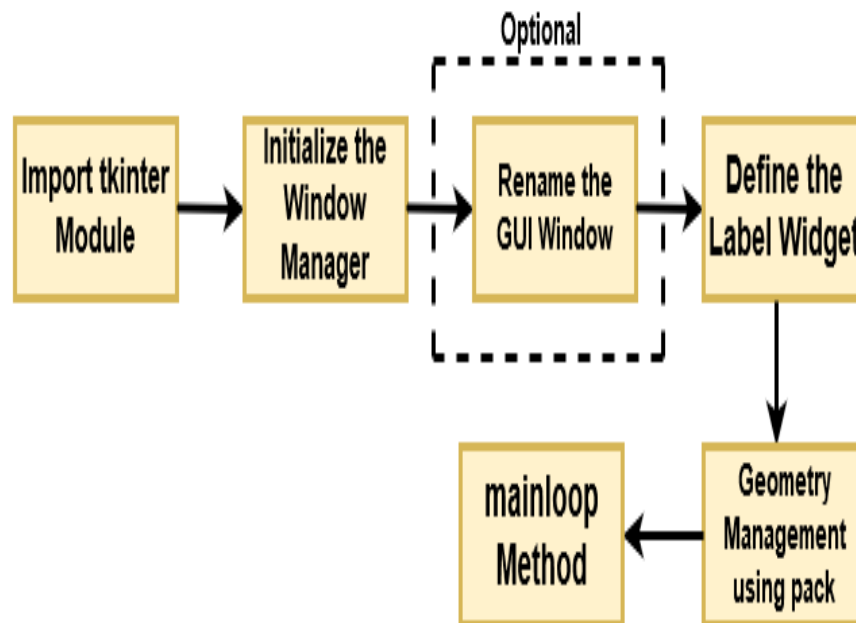


# Module Flow Diagram & UML Diagram

- Class diagram**

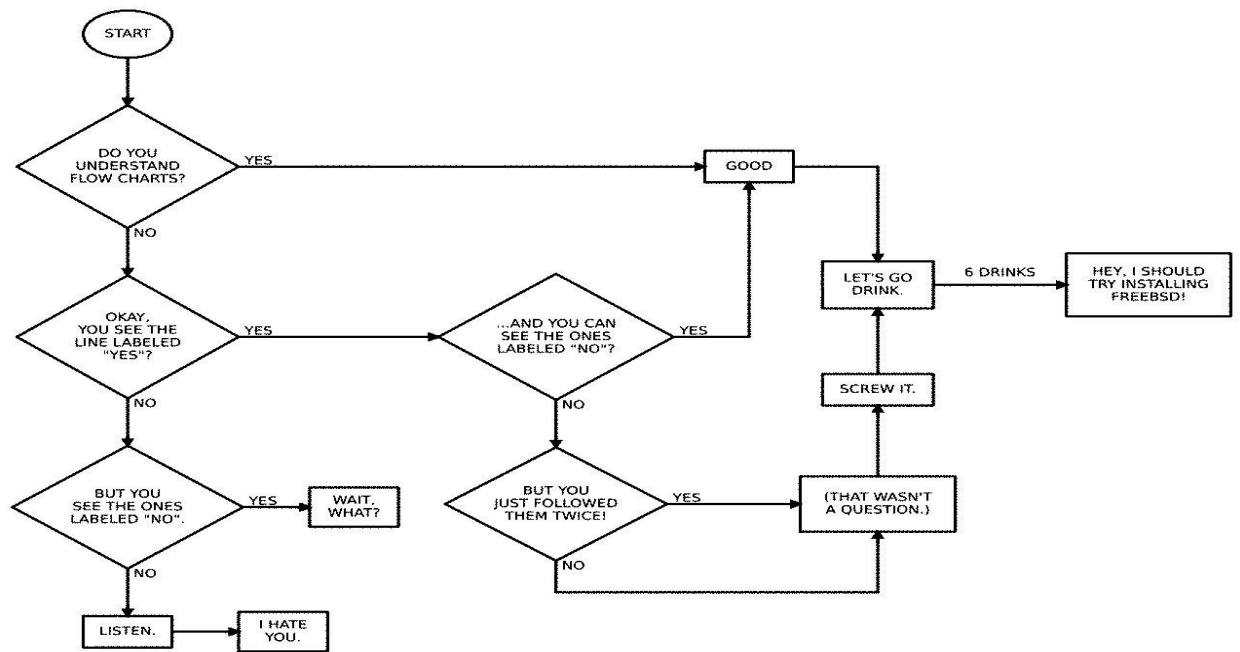


- FLOW DIAGRAM**

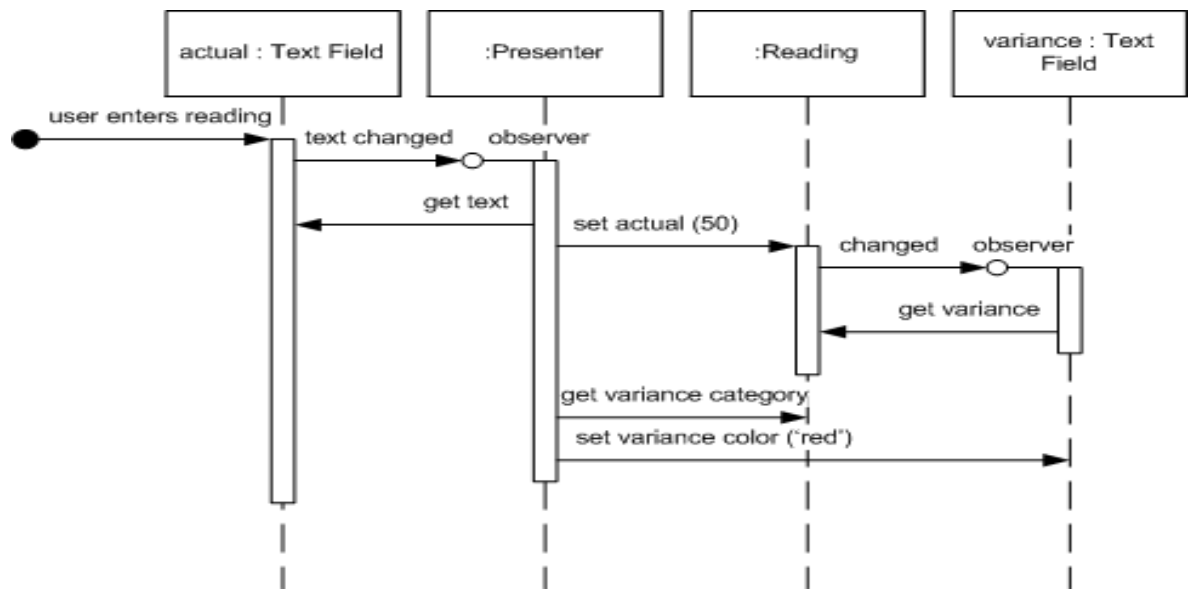


- UML DIAGRAM**

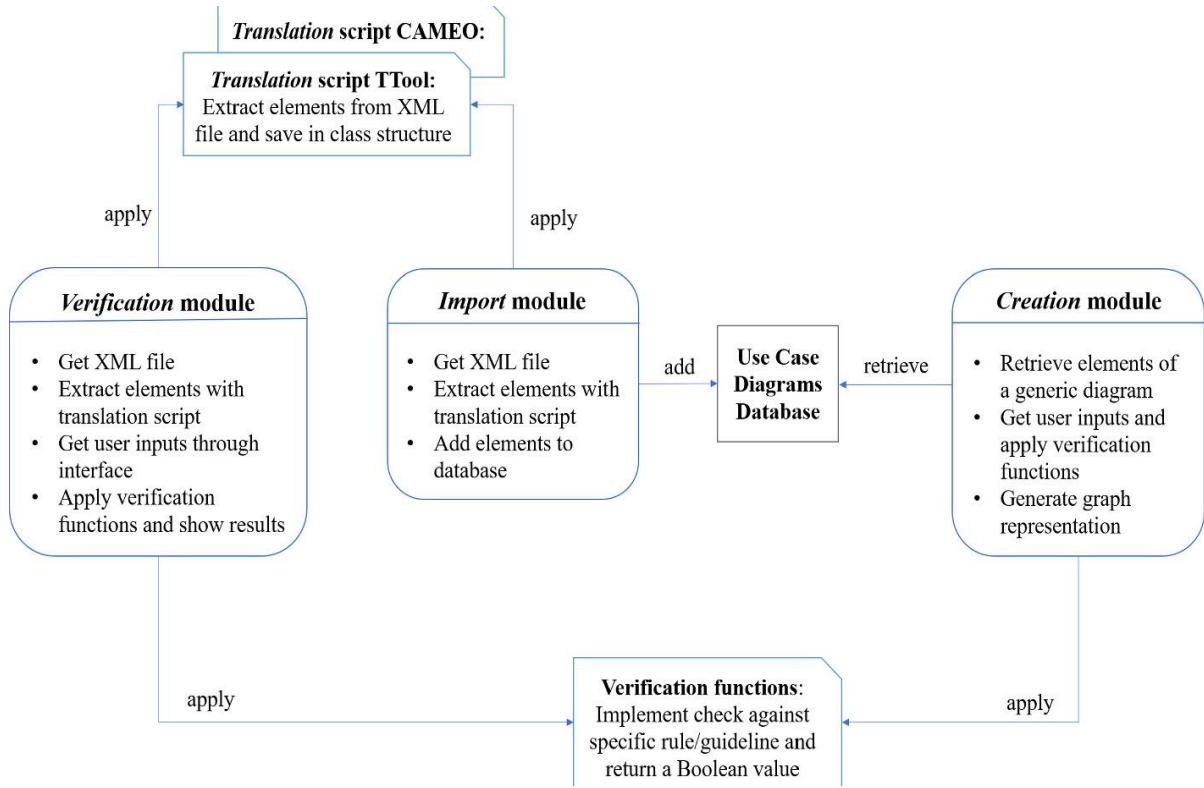




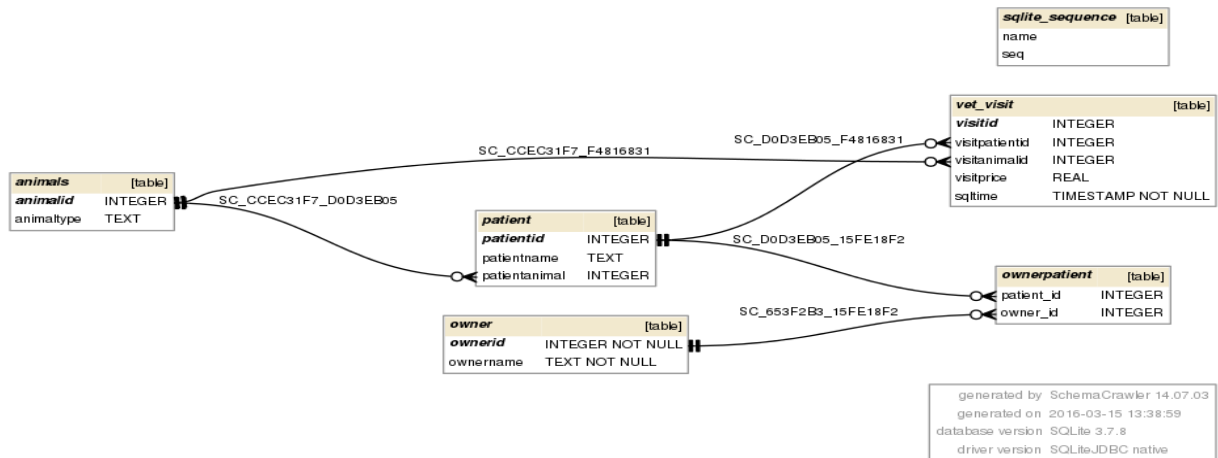
- **SEQUENCE DIAGRAM**



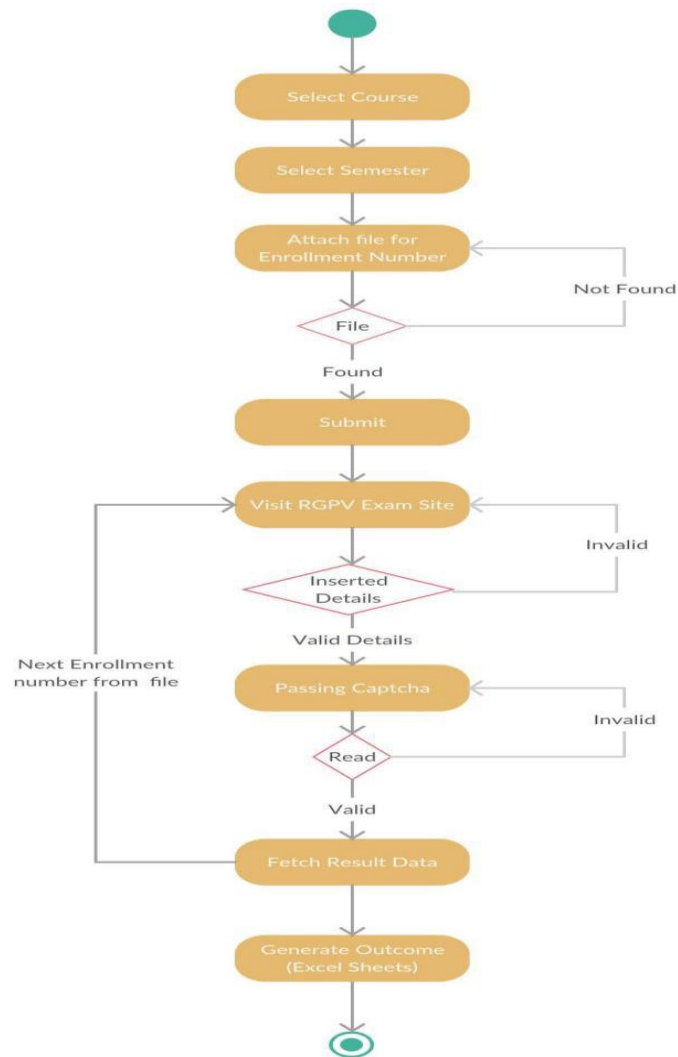
- **USE CASE DIAGRAM**



## • ER DIAGRAM



- **ACTIVITY DIAGRAM**



# **Implementation of Module with Methodolgy**

## **Technology Stack :**

- Python
- Python Packages
- Tkinter Package
- Chrome Driver
- Python OS module
- PyCharm
- Python V-3.9

# Code of the project

```
1  import tkinter
2  import random
3
4  colours = ['Red', 'Blue', 'Green', 'Pink', 'Black',
5            'Yellow', 'Orange', 'White', 'Purple', 'Brown']
6
7  score = 0
8
9  timeleft = 30
10
11 def startGame(event):
12     if timeleft == 30:
13         countdown()
14
15     nextColour()
16
17 def nextColour():
18
19     global score
20     global timeleft
21
22
23     if timeleft > 0:
24
25         e.focus_set()
```

```

28         if e.get().lower() == colours[1].lower():
29             score += 1
30
31         e.delete(0, tkinter.END)
32
33         random.shuffle(colours)
34
35         label.config(fg=str(colours[1]), text=str(colours[0]))
36
37
38         scoreLabel.config(text="Score: " + str(score))
39
40     def countdown():
41         global timeleft
42
43
44         if timeleft > 0:
45
46             timeleft -= 1
47
48             timeLabel.config(text="Time left: "
49                               + str(timeleft))
50
51
52         timeLabel.after(1000, countdown)

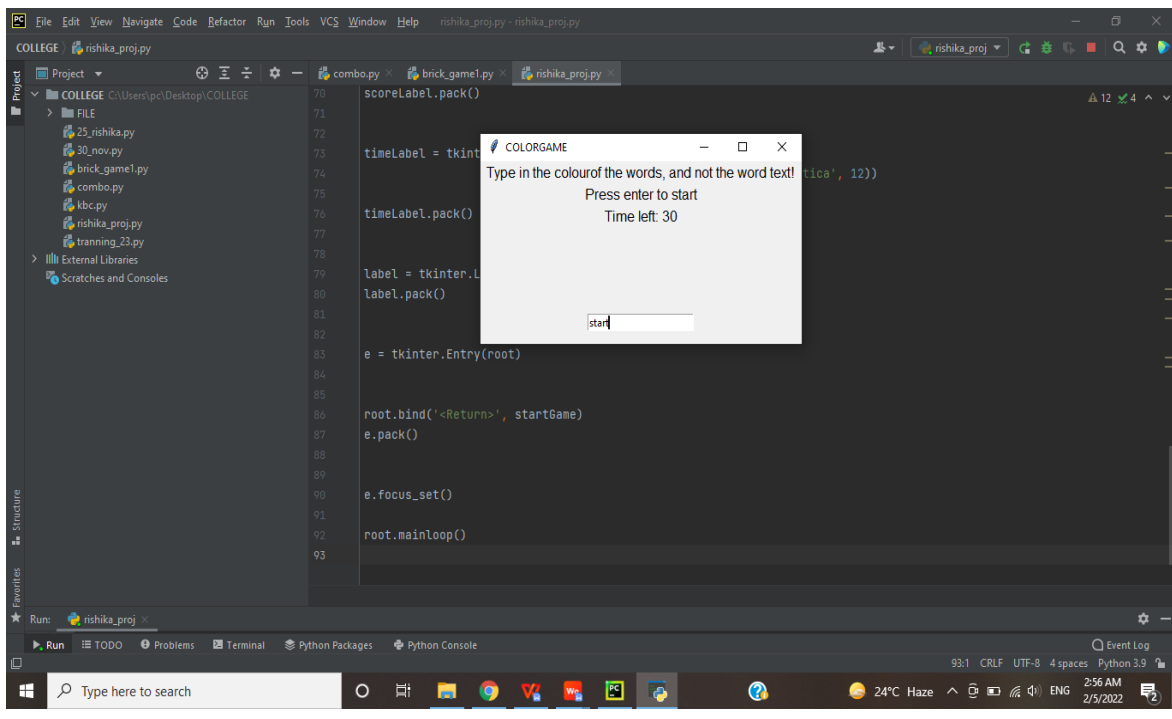
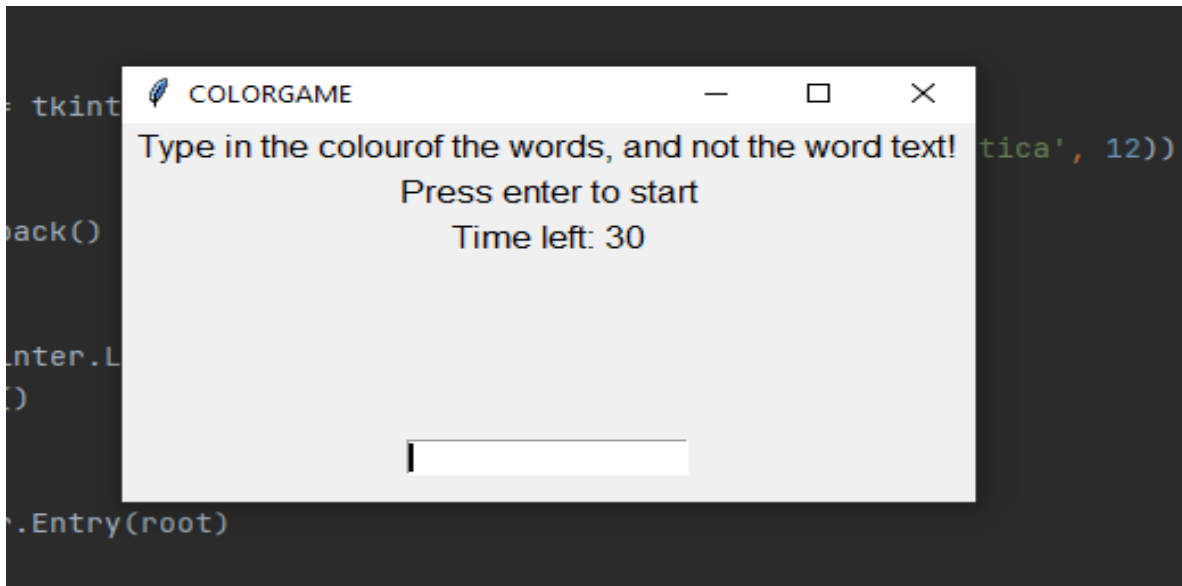
```

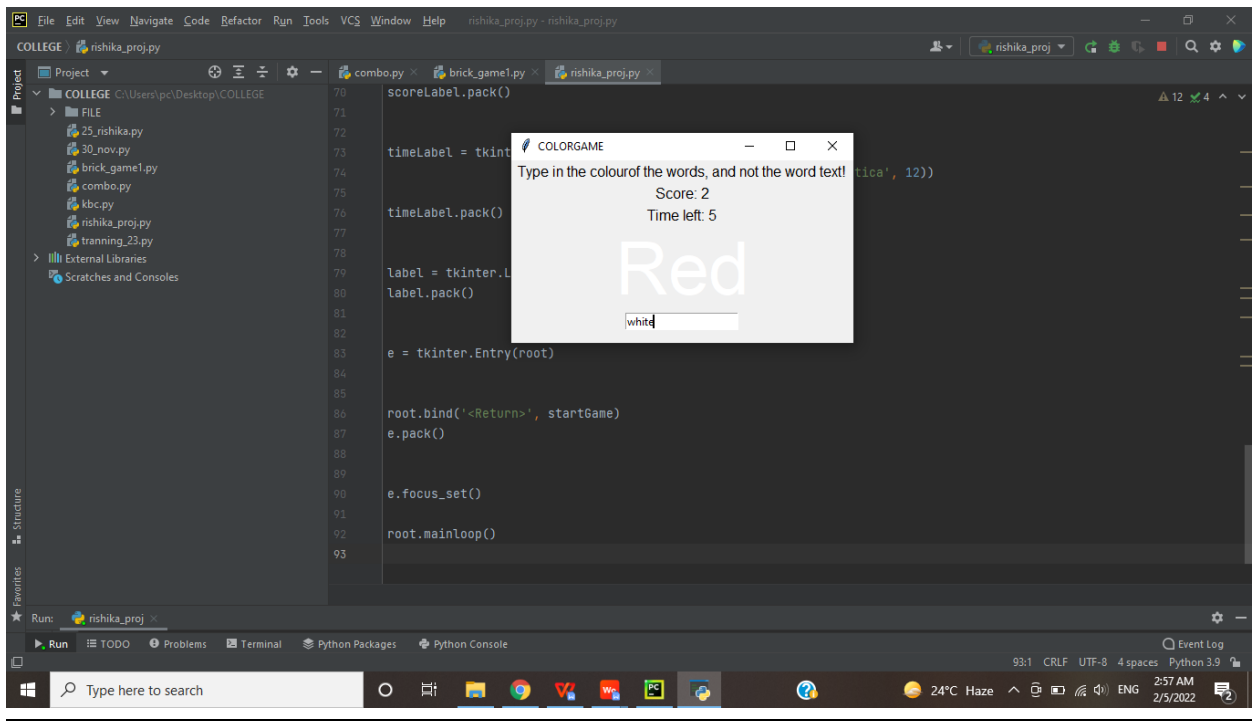
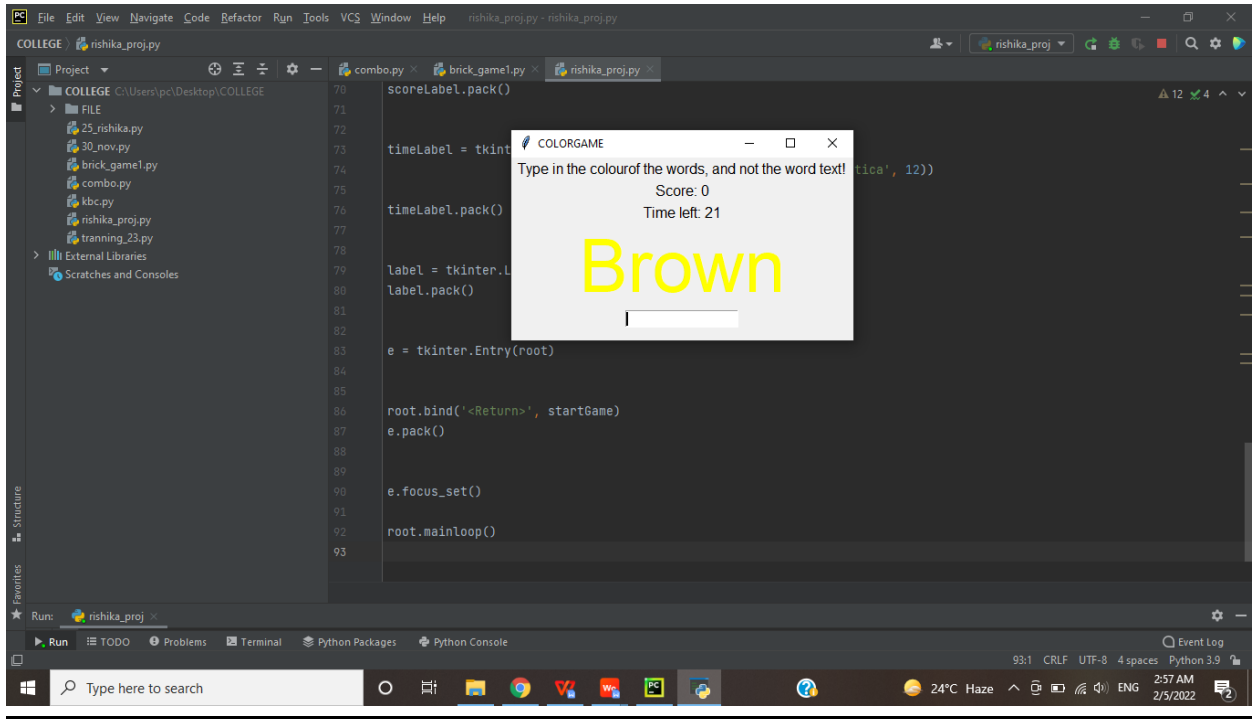
```
54  root = tkinter.Tk()
55
56  root.title("COLORGAME")
57
58
59  root.geometry("375x200")
60
61
62  instructions = tkinter.Label(root, text="Type in the colour"
63                                "of the words, and not the word text!",
64                                font=('Helvetica', 12))
65  instructions.pack()
66
67
68  scoreLabel = tkinter.Label(root, text="Press enter to start",
69                                font=('Helvetica', 12))
70  scoreLabel.pack()
71
72
73  timeLabel = tkinter.Label(root, text="Time left: " +
74                                str(timeleft), font=('Helvetica', 12))
75
76  timeLabel.pack()
```

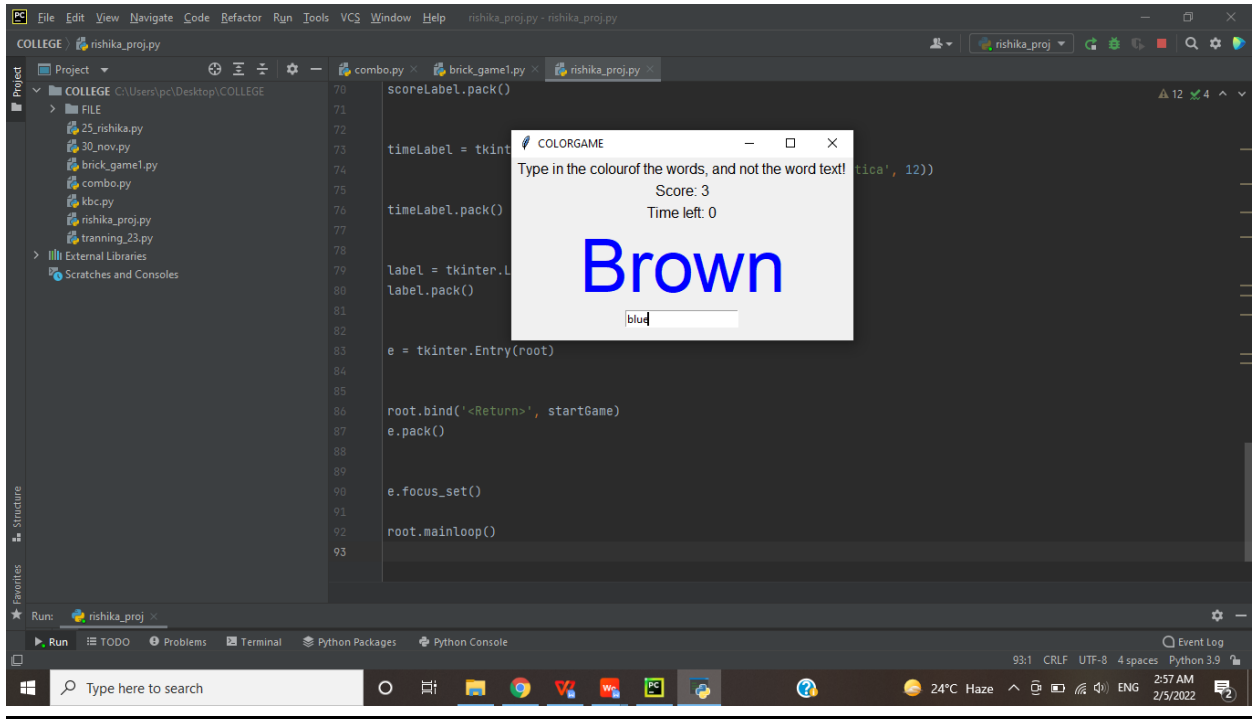
```
79 label = tkinter.Label(root, font=('Helvetica', 60))
80 label.pack()
81
82
83 e = tkinter.Entry(root)
84
85
86 root.bind('<Return>', startGame)
87 e.pack()
88
89
90 e.focus_set()
91
92 root.mainloop()
```



# Output







# Conclusion

Great introduction to Python and programming as a whole. Everything is broken down making it easy to understand. This was a well-designed course, especially for the beginners, who are really enthusiastic and passionate about programming. It really helps you to learn from scratch and it does not require any prerequisites. I get beginner level experience from this course. It was very good introduction to basic programming. Very easy for beginners in python who have already some programming background - but still extremely useful to quickly and efficiently learn python basics.

# References

[https://www.w3schools.com/python/python\\_lists.asp](https://www.w3schools.com/python/python_lists.asp)  
[https://www.tutorialspoint.com/python/python\\_modules.htm](https://www.tutorialspoint.com/python/python_modules.htm)  
<https://www.geeksforgeeks.org/python-gui-tkinter/>  
[https://www.w3schools.com/python/python\\_functions.asp](https://www.w3schools.com/python/python_functions.asp)