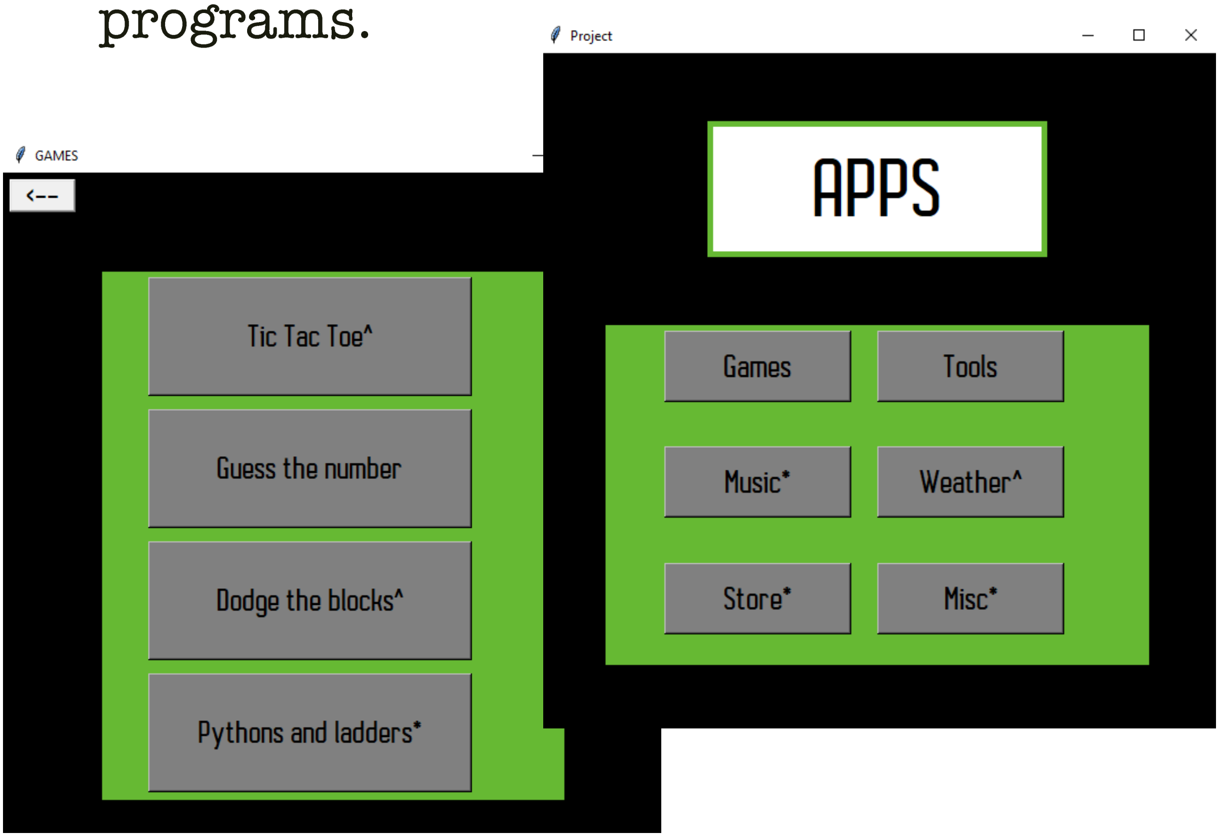
PROJECT

Menu Navigation App

Table Of Content

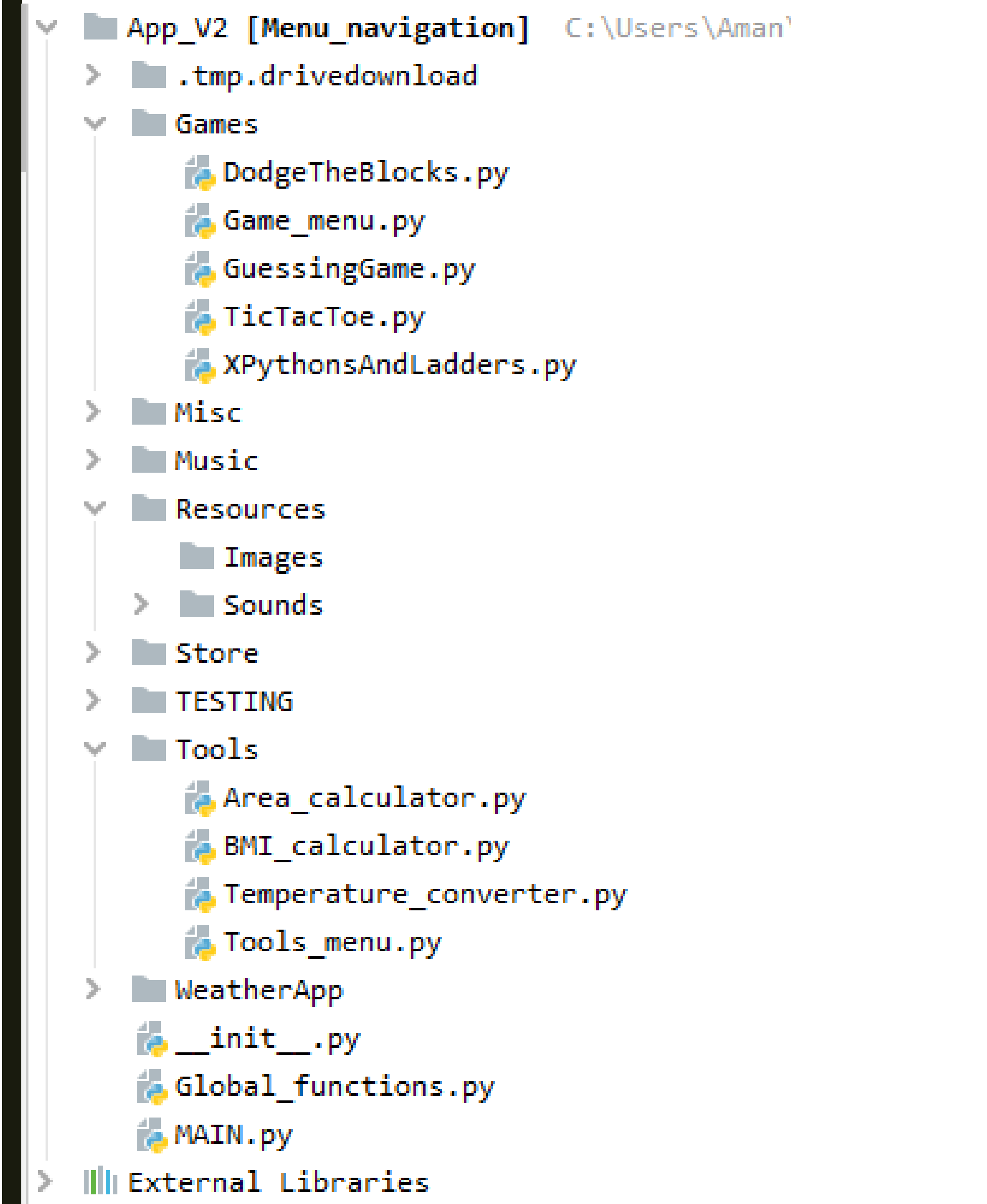
INTRODUCTION

- This is a GUI based program to open and navigate through different apps/programs.
- Tkinter is used to create the GUI and other such modules like Pygame are used to make the apps and programs.



KEY:-
'*' → Incomplete code
'^' → Outsourced program that
has been modified by me

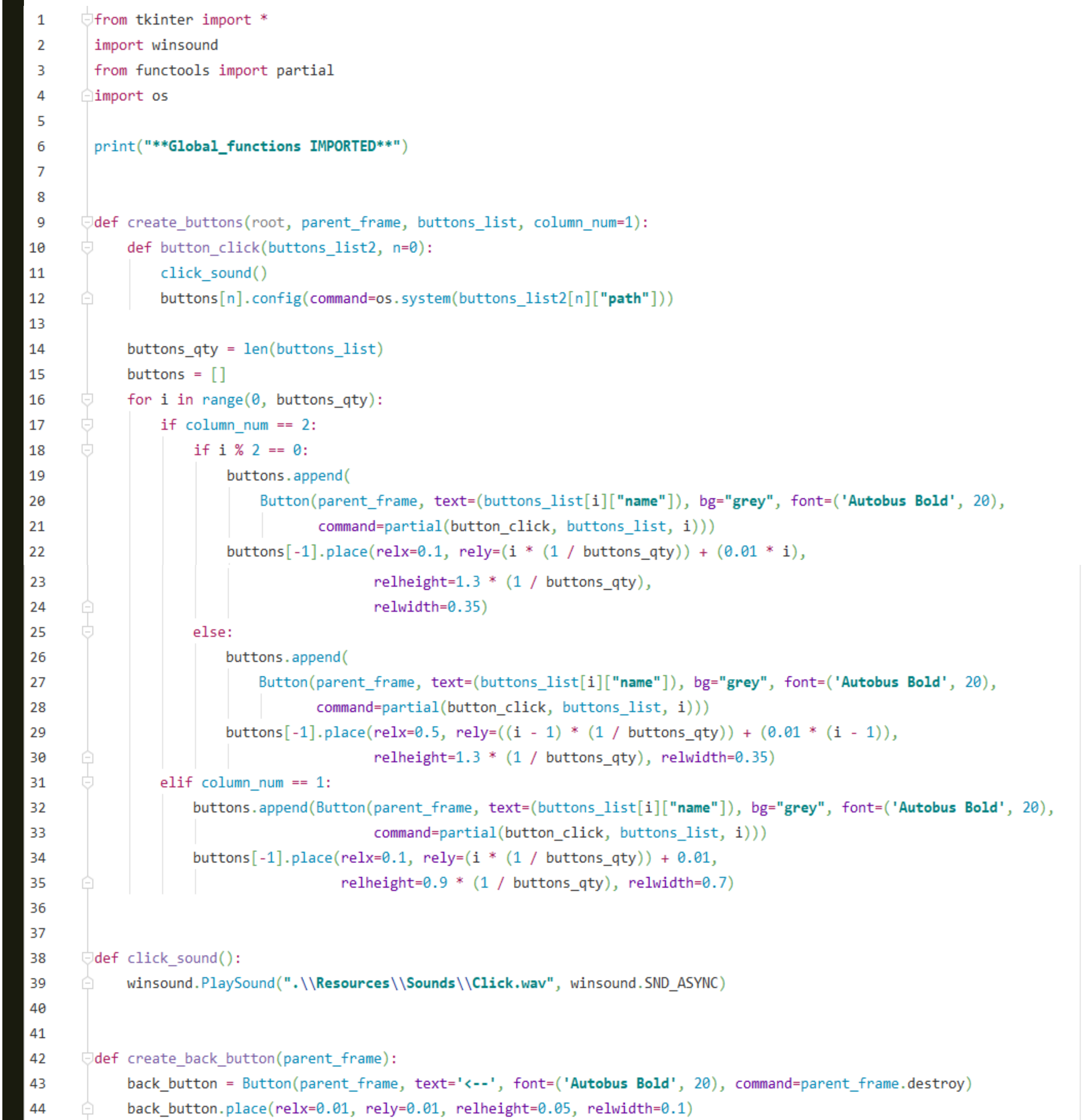
Directory Overview



```
App_V2 [Menu_navigation]  C:\Users\Aman'
  .tmp.drivedownload
  Games
    DodgeTheBlocks.py
    Game_menu.py
    GuessingGame.py
    TicTacToe.py
    XPythonsAndLadders.py
  Misc
  Music
  Resources
    Images
    Sounds
  Store
  TESTING
  Tools
    Area_calculator.py
    BMI_calculator.py
    Temperature_converter.py
    Tools_menu.py
  WeatherApp
  __init__.py
  Global_functions.py
  MAIN.py
External Libraries
```

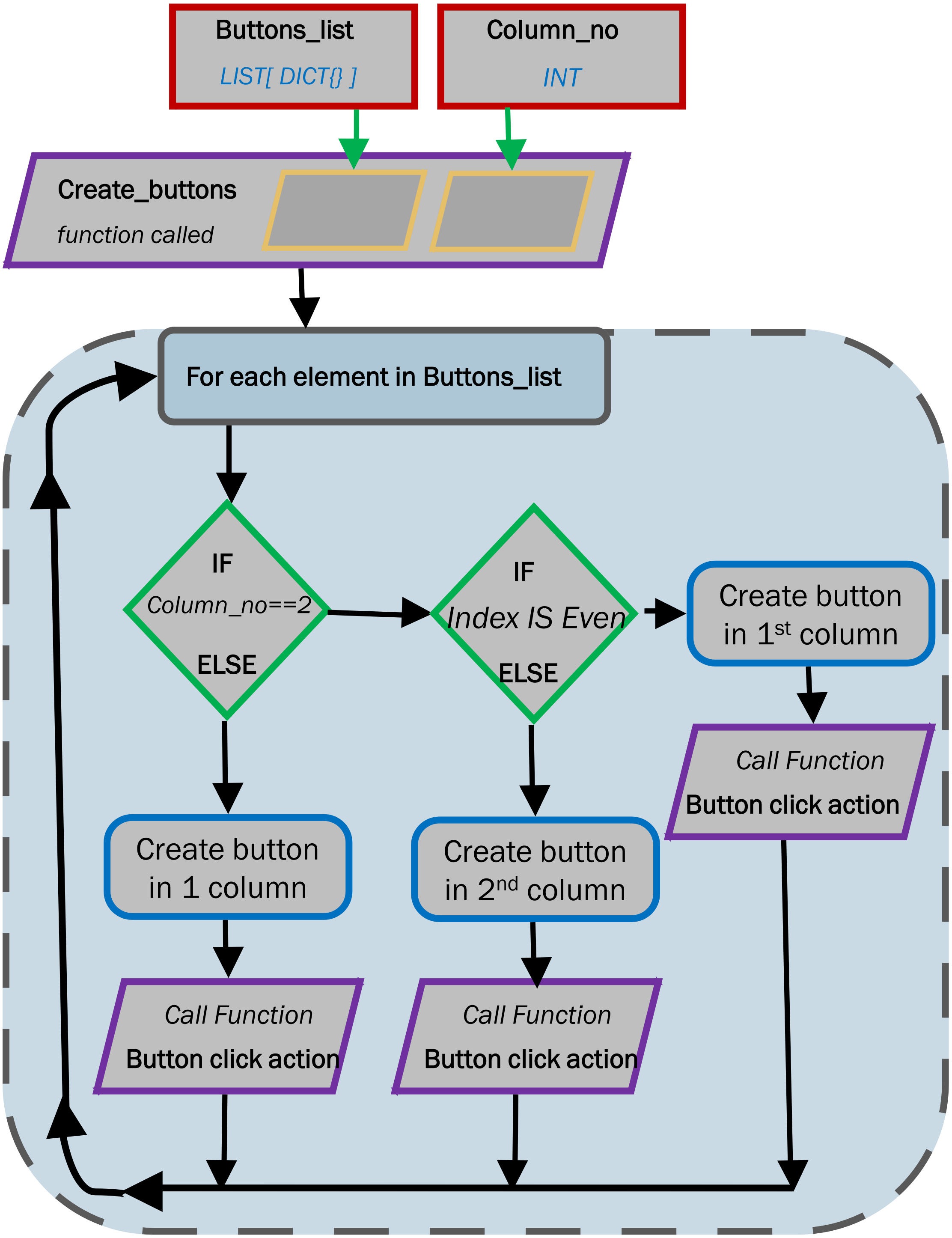
PROGRAMS

- **Global_functions.py**
  - Custom module that is referenced by other programs in the project.



```python
from tkinter import *
import winsound
from functools import partial
import os

print("**Global_functions IMPORTED**")


def create_buttons(root, parent_frame, buttons_list, column_num=1):
    def button_click(buttons_list2, n=0):
        click_sound()
        buttons[n].config(command=os.system(buttons_list2[n]["path"]))

    buttons_qty = len(buttons_list)
    buttons = []
    for i in range(0, buttons_qty):
        if column_num == 2:
            if i % 2 == 0:
                buttons.append(
                    Button(parent_frame, text=(buttons_list[i]["name"]), bg="grey", font=('Autobus Bold', 20),
                           command=partial(button_click, buttons_list, i)))
                buttons[-1].place(relx=0.1, rely=(i * (1 / buttons_qty)) + (0.01 * i),
                                  relheight=1.3 * (1 / buttons_qty),
                                  relwidth=0.35)
            else:
                buttons.append(
                    Button(parent_frame, text=(buttons_list[i]["name"]), bg="grey", font=('Autobus Bold', 20),
                           command=partial(button_click, buttons_list, i)))
                buttons[-1].place(relx=0.5, rely=((i - 1) * (1 / buttons_qty)) + (0.01 * (i - 1)),
                                  relheight=1.3 * (1 / buttons_qty), relwidth=0.35)
        elif column_num == 1:
            buttons.append(Button(parent_frame, text=(buttons_list[i]["name"]), bg="grey", font=('Autobus Bold', 20),
                                  command=partial(button_click, buttons_list, i)))
            buttons[-1].place(relx=0.1, rely=(i * (1 / buttons_qty)) + 0.01,
                              relheight=0.9 * (1 / buttons_qty), relwidth=0.7)


def click_sound():
    winsound.PlaySound(".\\Resources\\Sounds\\Click.wav", winsound.SND_ASYNC)


def create_back_button(parent_frame):
    back_button = Button(parent_frame, text='<--', font=('Autobus Bold', 20), command=parent_frame.destroy)
    back_button.place(relx=0.01, rely=0.01, relheight=0.05, relwidth=0.1)
```

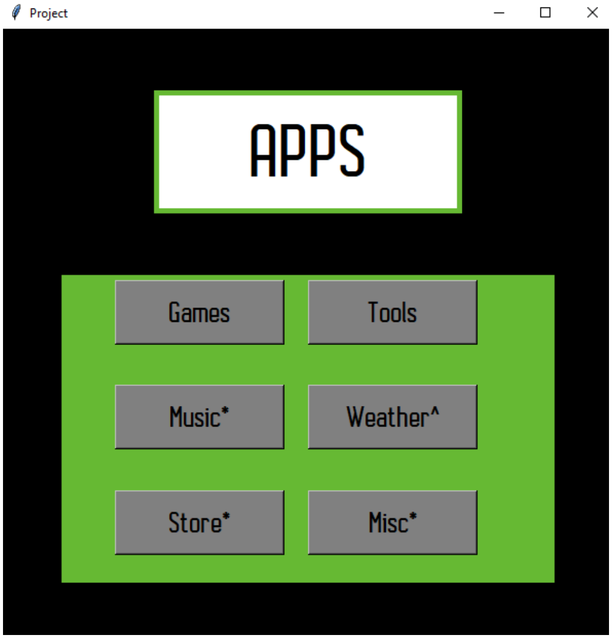
- Create_Buttons() Function
  - Creates buttons inside a parent frame based on the number of buttons and number of columns specified. It also creates separate functions for each button to open a specified program when button is clicked.
    - Pseudo Code :-

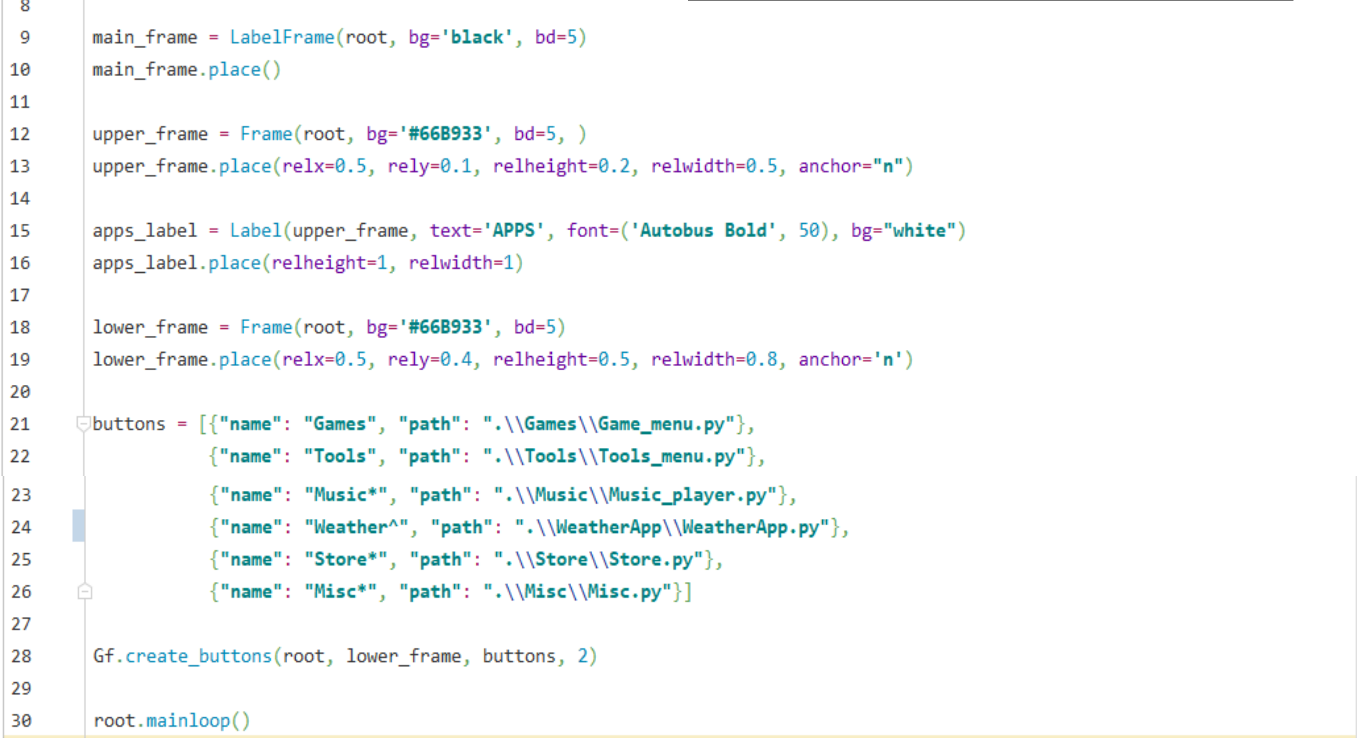


- Create_back_button()
  - Creates a back button which when pressed destroys/closes the current Tkinter window.
- Click_sound()
  - Plays a .wav file which is of a Click sound.
  - Called when a button is pressed

MAIN.py

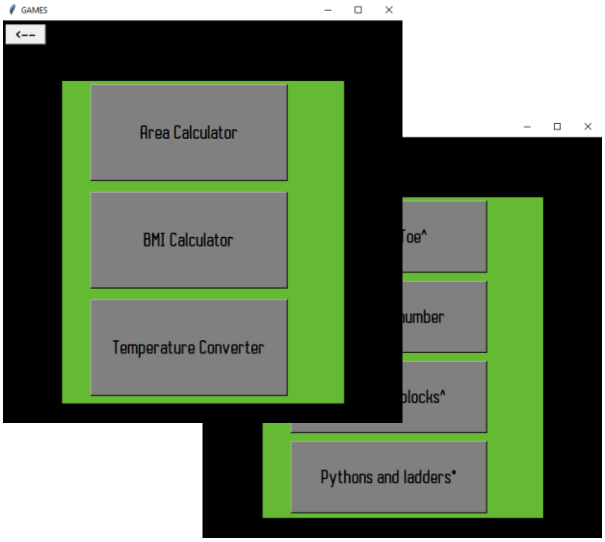
- Creates a Heading label and frames to contain the buttons.





```python
import Global_functions as Gf
from tkinter import *

root = Tk()
root.title('Project')
root.geometry("600x600")
root.configure(bg="black")

main_frame = LabelFrame(root, bg='black', bd=5)
main_frame.place()

upper_frame = Frame(root, bg='#66B933', bd=5, )
upper_frame.place(relx=0.5, rely=0.1, relheight=0.2, relwidth=0.5, anchor="n")

apps_label = Label(upper_frame, text='APPS', font=('Autobus Bold', 50), bg="white")
apps_label.place(relheight=1, relwidth=1)

lower_frame = Frame(root, bg='#66B933', bd=5)
lower_frame.place(relx=0.5, rely=0.4, relheight=0.5, relwidth=0.8, anchor='n')

buttons = [{"name": "Games", "path": ".\\Games\\Game_menu.py"},
           {"name": "Tools", "path": ".\\Tools\\Tools_menu.py"},
           {"name": "Music*", "path": ".\\Music\\Music_player.py"},
           {"name": "Weather^", "path": ".\\WeatherApp\\WeatherApp.py"},
           {"name": "Store*", "path": ".\\Store\\Store.py"},
           {"name": "Misc*", "path": ".\\Misc\\Misc.py"}]

Gf.create_buttons(root, lower_frame, buttons, 2)

root.mainloop()
```

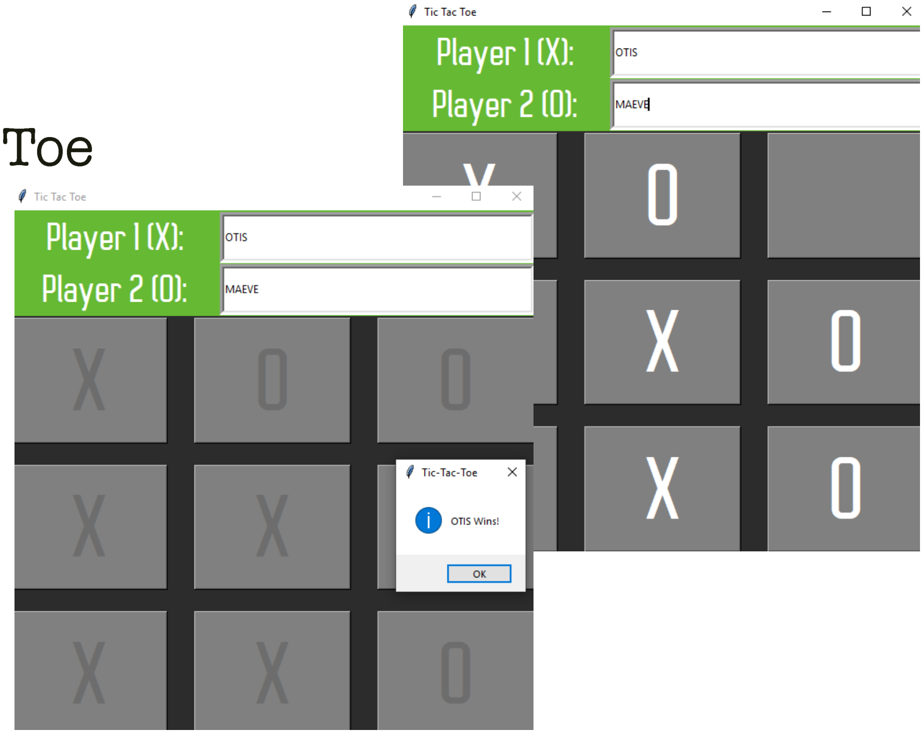
- **%_menu.py**
  - Creates frames to contain the buttons.



```python
from Global_functions import *

tools_menu = Tk()
tools_menu.title('GAMES')
tools_menu.geometry("600x600")
tools_menu.configure(bg="black")

tools_menu_frame = Frame(tools_menu, bg='#66B933')
tools_menu_frame.place(relx=0.5, rely=0.15, relheight=0.8, relwidth=0.7, anchor='n')

buttons = [{"name": "Area Calculator", "path": ".\\Tools\\Area_calculator.py"},
           {"name": "BMI Calculator", "path": ".\\Tools\\BMI_calculator.py"},
           {"name": "Temperature Converter", "path": ".\\Tools\\Temperature_converter.py"}]

create_buttons(tools_menu, tools_menu_frame, buttons, 1)

create_back_button(tools_menu)

mainloop()
```

APPs

- **Games**
  - Tic Tac Toe


Tic Tac Toe

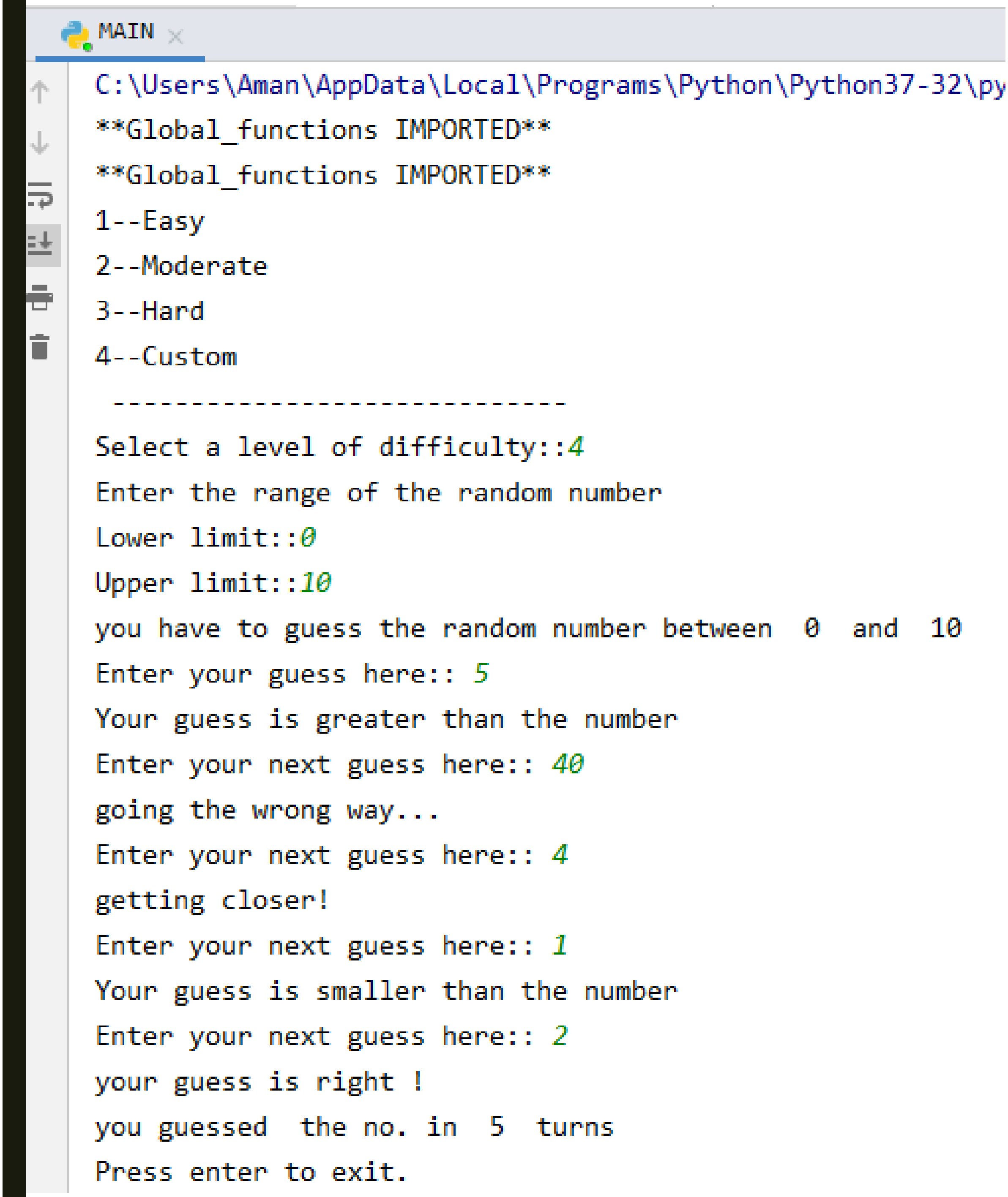
Player 1 (X): OTIS

Player 2 (O): MAEVE

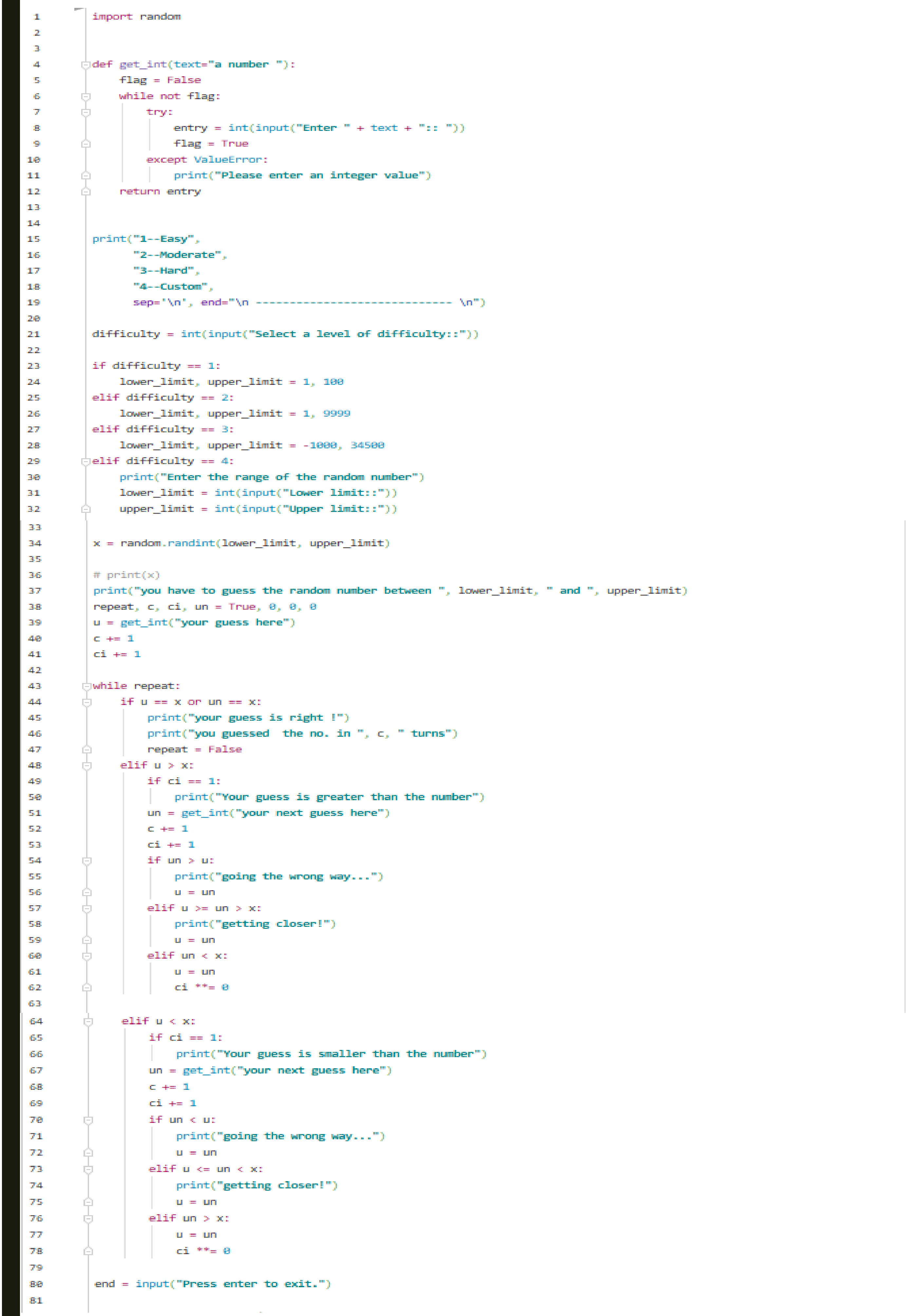
Tic-Tac-Toe

OTIS Wins!

- Guessing game
  - The user has to guess a number between a given range while the program provides hints

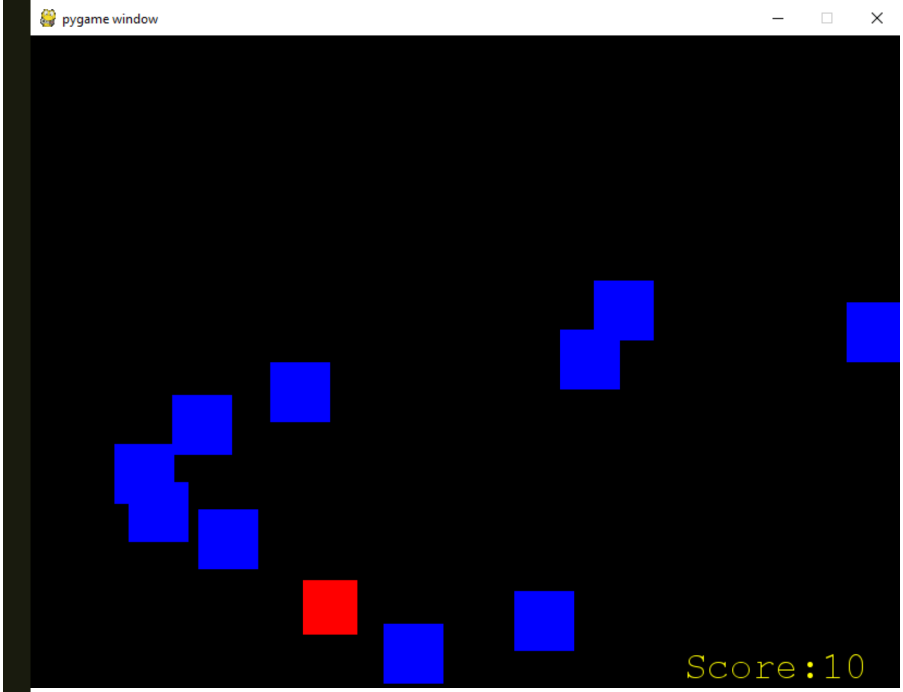


```
C:\Users\Aman\AppData\Local\Programs\Python\Python37-32\py
**Global_functions IMPORTED**
**Global_functions IMPORTED**
1--Easy
2--Moderate
3--Hard
4--Custom
 -----------------------------
Select a level of difficulty::4
Enter the range of the random number
Lower limit::0
Upper limit::10
you have to guess the random number between  0  and  10
Enter your guess here:: 5
Your guess is greater than the number
Enter your next guess here:: 40
going the wrong way...
Enter your next guess here:: 4
getting closer!
Enter your next guess here:: 1
Your guess is smaller than the number
Enter your next guess here:: 2
your guess is right !
you guessed  the no. in  5  turns
Press enter to exit.
```

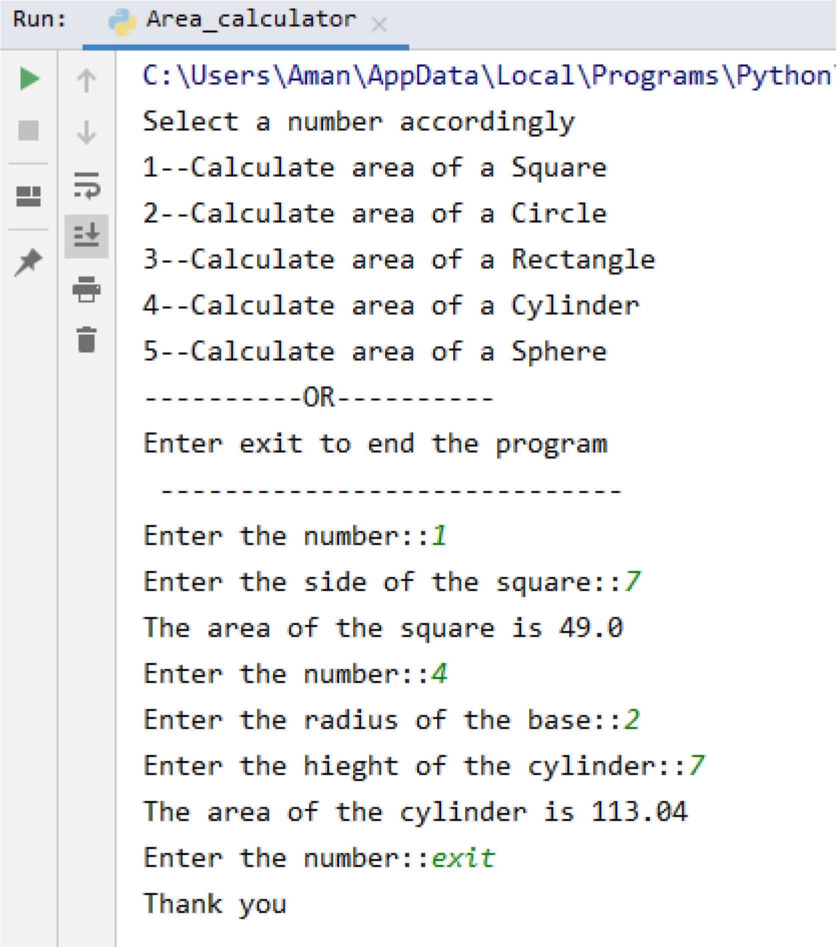


```python
import random


def get_int(text="a number "):
    flag = False
    while not flag:
        try:
            entry = int(input("Enter " + text + ":: "))
            flag = True
        except ValueError:
            print("Please enter an integer value")
    return entry


print("1--Easy",
      "2--Moderate",
      "3--Hard",
      "4--Custom",
      sep='\n', end="\n ----------------------------- \n")

difficulty = int(input("Select a level of difficulty::"))

if difficulty == 1:
    lower_limit, upper_limit = 1, 100
elif difficulty == 2:
    lower_limit, upper_limit = 1, 9999
elif difficulty == 3:
    lower_limit, upper_limit = -1000, 34500
elif difficulty == 4:
    print("Enter the range of the random number")
    lower_limit = int(input("Lower limit::"))
    upper_limit = int(input("Upper limit::"))

x = random.randint(lower_limit, upper_limit)

# print(x)
print("you have to guess the random number between ", lower_limit, " and ", upper_limit)
repeat, c, ci, un = True, 0, 0, 0
u = get_int("your guess here")
c += 1
ci += 1

while repeat:
    if u == x or un == x:
        print("your guess is right !")
        print("you guessed  the no. in ", c, " turns")
        repeat = False
    elif u > x:
        if ci == 1:
            print("Your guess is greater than the number")
        un = get_int("your next guess here")
        c += 1
        ci += 1
        if un > u:
            print("going the wrong way...")
            u = un
        elif u >= un > x:
            print("getting closer!")
            u = un
        elif un < x:
            u = un
            ci **= 0

    elif u < x:
        if ci == 1:
            print("Your guess is smaller than the number")
        un = get_int("your next guess here")
        c += 1
        ci += 1
        if un < u:
            print("going the wrong way...")
            u = un
        elif u <= un < x:
            print("getting closer!")
            u = un
        elif un > x:
            u = un
            ci **= 0

end = input("Press enter to exit.")

```

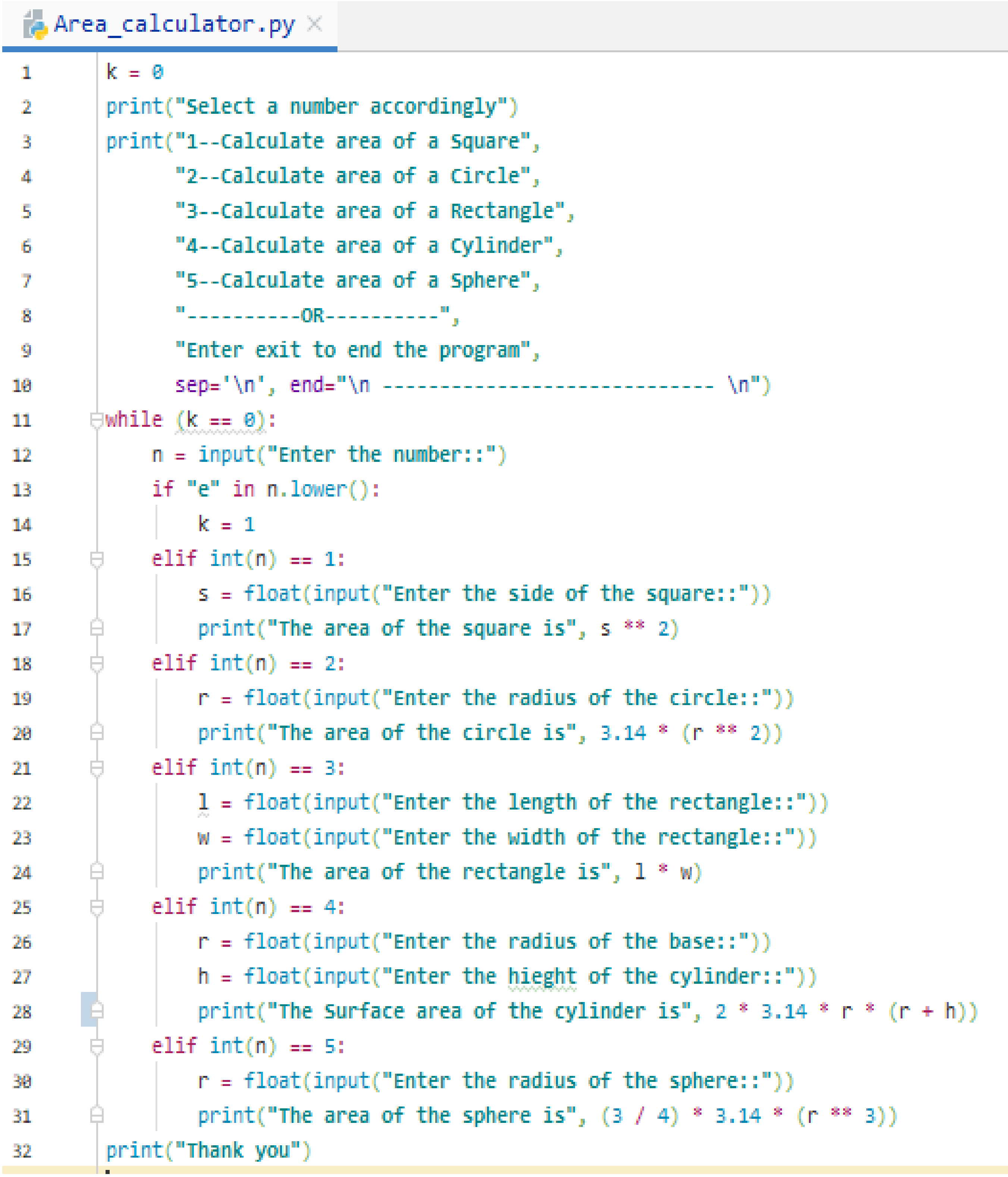
- Dodge the blocks
  - The user has to Move the Red (player) block using arrow keys to avoid the Blue(Enemy) Blocks



- **Tools**
  - Area Calculator

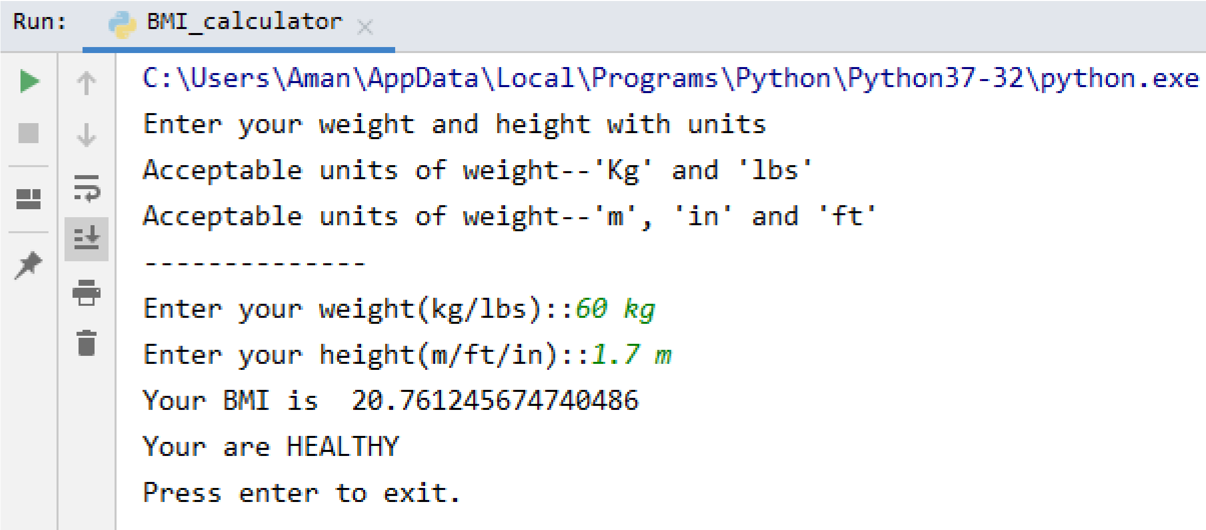


```
C:\Users\Aman\AppData\Local\Programs\Python
Select a number accordingly
1--Calculate area of a Square
2--Calculate area of a Circle
3--Calculate area of a Rectangle
4--Calculate area of a Cylinder
5--Calculate area of a Sphere
----------OR----------
Enter exit to end the program
 ------------------------------
Enter the number::1
Enter the side of the square::7
The area of the square is 49.0
Enter the number::4
Enter the radius of the base::2
Enter the hieght of the cylinder::7
The area of the cylinder is 113.04
Enter the number::exit
Thank you
```

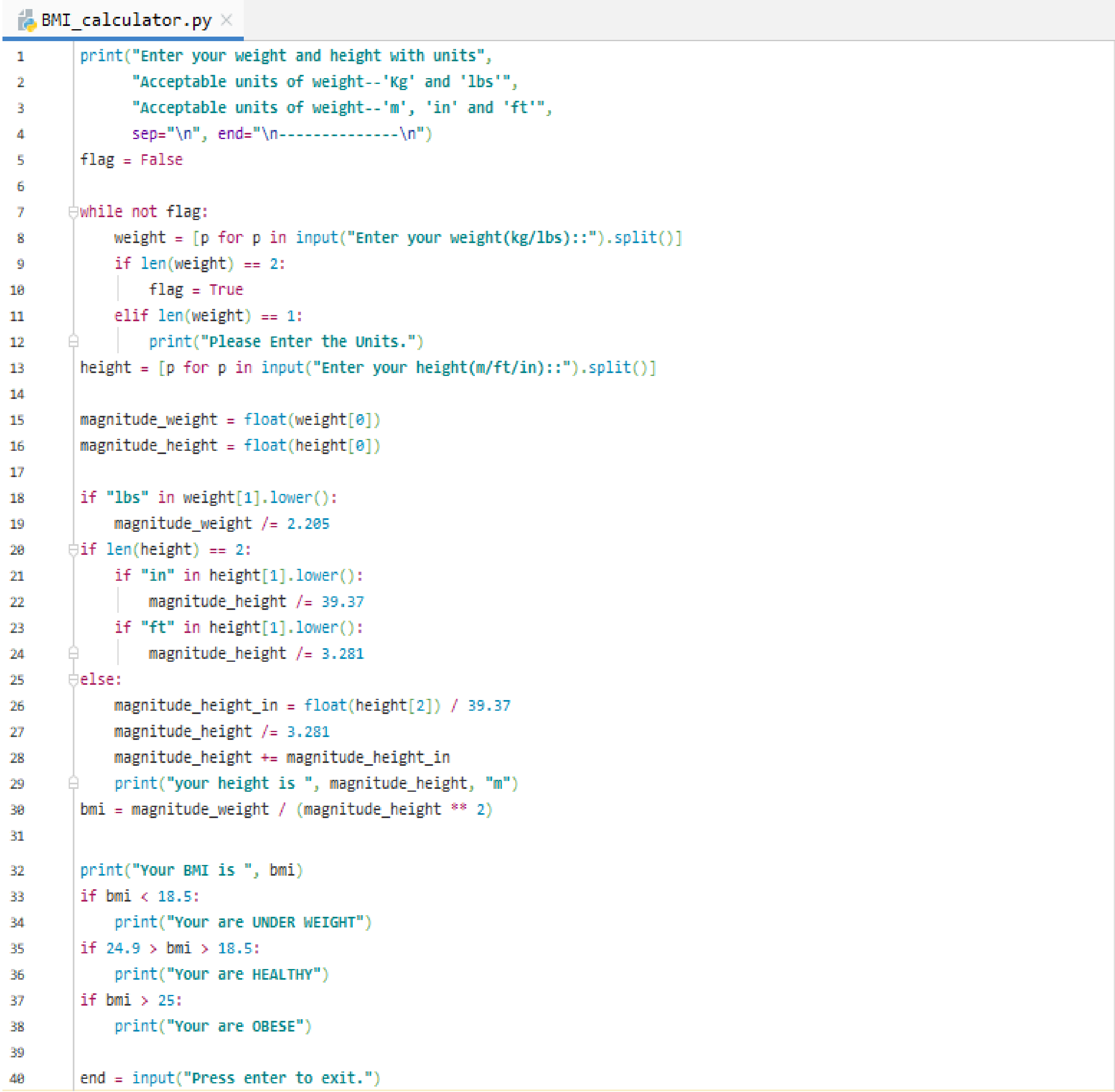


```python
k = 0
print("Select a number accordingly")
print("1--Calculate area of a Square",
      "2--Calculate area of a Circle",
      "3--Calculate area of a Rectangle",
      "4--Calculate area of a Cylinder",
      "5--Calculate area of a Sphere",
      "----------OR----------",
      "Enter exit to end the program",
      sep='\n', end="\n ----------------------------- \n")
while (k == 0):
    n = input("Enter the number::")
    if "e" in n.lower():
        k = 1
    elif int(n) == 1:
        s = float(input("Enter the side of the square::"))
        print("The area of the square is", s ** 2)
    elif int(n) == 2:
        r = float(input("Enter the radius of the circle::"))
        print("The area of the circle is", 3.14 * (r ** 2))
    elif int(n) == 3:
        l = float(input("Enter the length of the rectangle::"))
        w = float(input("Enter the width of the rectangle::"))
        print("The area of the rectangle is", l * w)
    elif int(n) == 4:
        r = float(input("Enter the radius of the base::"))
        h = float(input("Enter the hieght of the cylinder::"))
        print("The Surface area of the cylinder is", 2 * 3.14 * r * (r + h))
    elif int(n) == 5:
        r = float(input("Enter the radius of the sphere::"))
        print("The area of the sphere is", (3 / 4) * 3.14 * (r ** 3))
print("Thank you")
```

– BMI Calculator

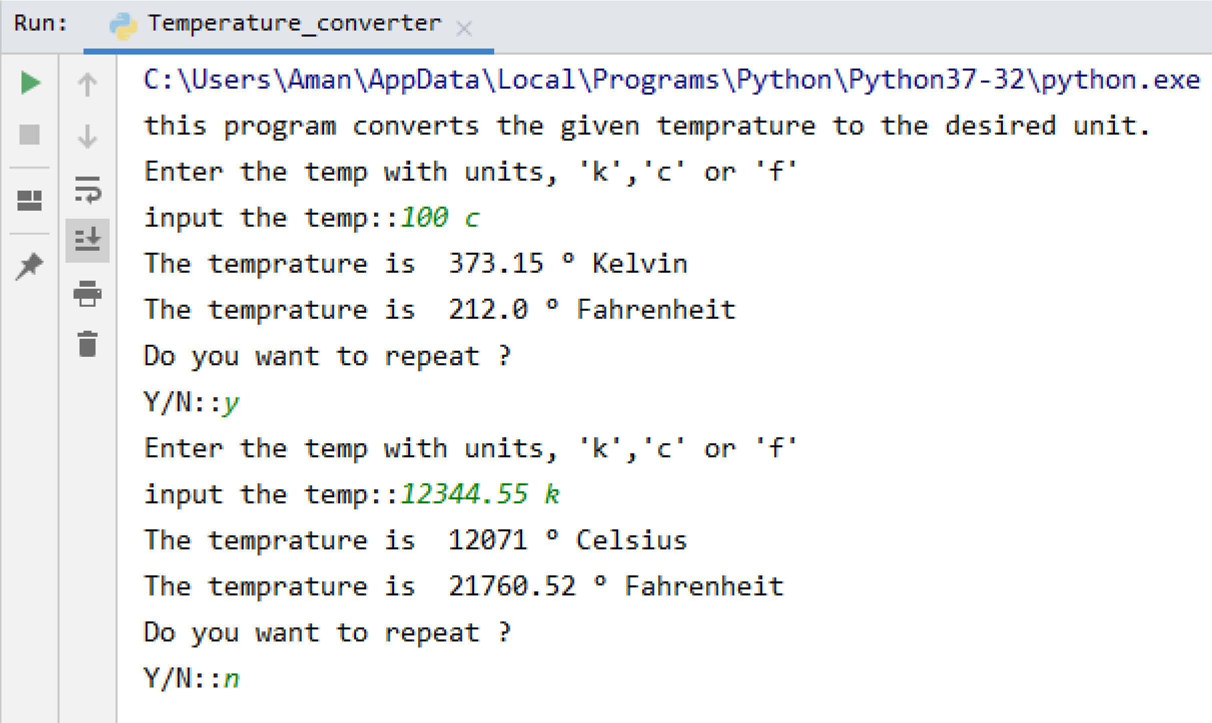


```
C:\Users\Aman\AppData\Local\Programs\Python\Python37-32\python.exe
Enter your weight and height with units
Acceptable units of weight--'Kg' and 'lbs'
Acceptable units of weight--'m', 'in' and 'ft'
--------------
Enter your weight(kg/lbs)::60 kg
Enter your height(m/ft/in)::1.7 m
Your BMI is  20.761245674740486
Your are HEALTHY
Press enter to exit.
```

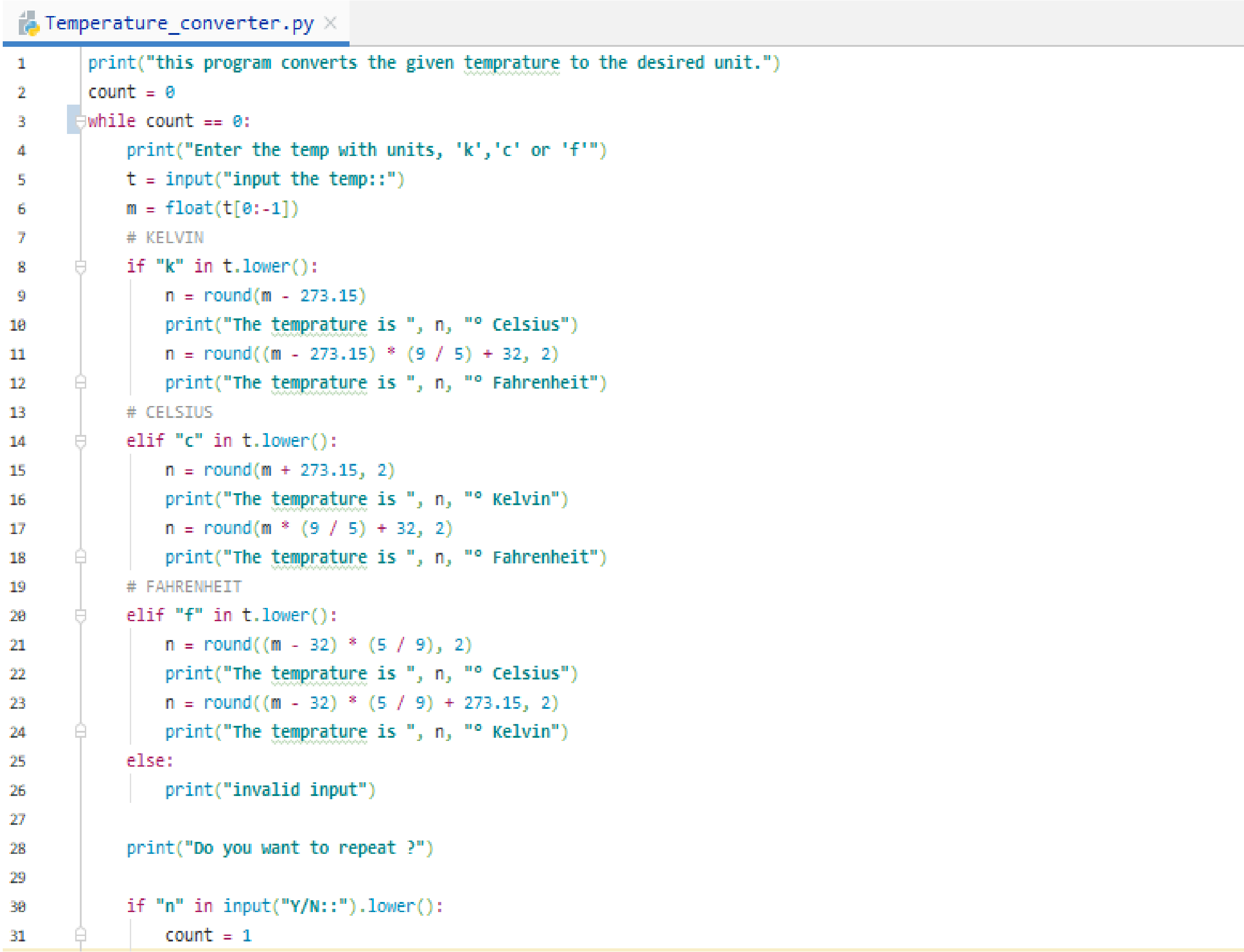


```python
print("Enter your weight and height with units",
      "Acceptable units of weight--'Kg' and 'lbs'",
      "Acceptable units of weight--'m', 'in' and 'ft'",
      sep="\n", end="\n--------------\n")
flag = False

while not flag:
    weight = [p for p in input("Enter your weight(kg/lbs)::").split()]
    if len(weight) == 2:
        flag = True
    elif len(weight) == 1:
        print("Please Enter the Units.")
height = [p for p in input("Enter your height(m/ft/in)::").split()]

magnitude_weight = float(weight[0])
magnitude_height = float(height[0])

if "lbs" in weight[1].lower():
    magnitude_weight /= 2.205
if len(height) == 2:
    if "in" in height[1].lower():
        magnitude_height /= 39.37
    if "ft" in height[1].lower():
        magnitude_height /= 3.281
else:
    magnitude_height_in = float(height[2]) / 39.37
    magnitude_height /= 3.281
    magnitude_height += magnitude_height_in
    print("your height is ", magnitude_height, "m")
bmi = magnitude_weight / (magnitude_height ** 2)


print("Your BMI is ", bmi)
if bmi < 18.5:
    print("Your are UNDER WEIGHT")
if 24.9 > bmi > 18.5:
    print("Your are HEALTHY")
if bmi > 25:
    print("Your are OBESE")

end = input("Press enter to exit.")
```

- Temperature Converter

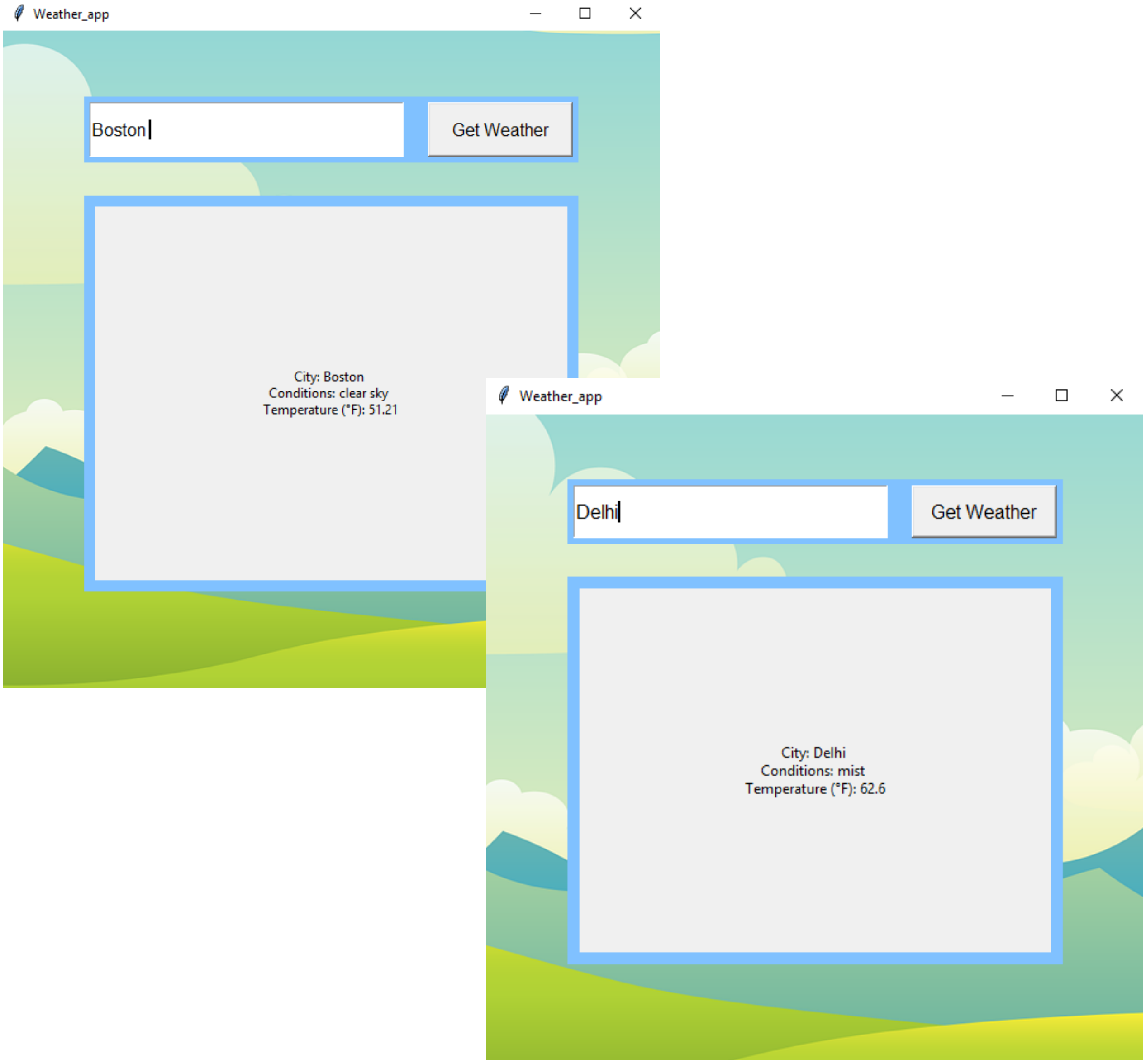


```
C:\Users\Aman\AppData\Local\Programs\Python\Python37-32\python.exe
this program converts the given temprature to the desired unit.
Enter the temp with units, 'k','c' or 'f'
input the temp::100 c
The temprature is  373.15 ° Kelvin
The temprature is  212.0 ° Fahrenheit
Do you want to repeat ?
Y/N::y
Enter the temp with units, 'k','c' or 'f'
input the temp::12344.55 k
The temprature is  12071 ° Celsius
The temprature is  21760.52 ° Fahrenheit
Do you want to repeat ?
Y/N::n
```



```python
print("this program converts the given temprature to the desired unit.")
count = 0
while count == 0:
    print("Enter the temp with units, 'k','c' or 'f'")
    t = input("input the temp::")
    m = float(t[0:-1])
    # KELVIN
    if "k" in t.lower():
        n = round(m - 273.15)
        print("The temprature is ", n, "° Celsius")
        n = round((m - 273.15) * (9 / 5) + 32, 2)
        print("The temprature is ", n, "° Fahrenheit")
    # CELSIUS
    elif "c" in t.lower():
        n = round(m + 273.15, 2)
        print("The temprature is ", n, "° Kelvin")
        n = round(m * (9 / 5) + 32, 2)
        print("The temprature is ", n, "° Fahrenheit")
    # FAHRENHEIT
    elif "f" in t.lower():
        n = round((m - 32) * (5 / 9), 2)
        print("The temprature is ", n, "° Celsius")
        n = round((m - 32) * (5 / 9) + 273.15, 2)
        print("The temprature is ", n, "° Kelvin")
    else:
        print("invalid input")

    print("Do you want to repeat ?")

    if "n" in input("Y/N::").lower():
        count = 1
```

- **Weather App^**
  - When a user enters name of a city, it retrieves weather information from an external link and displays it.



Boston

```
City: Boston
Conditions: clear sky
Temperature (°F): 51.21
```

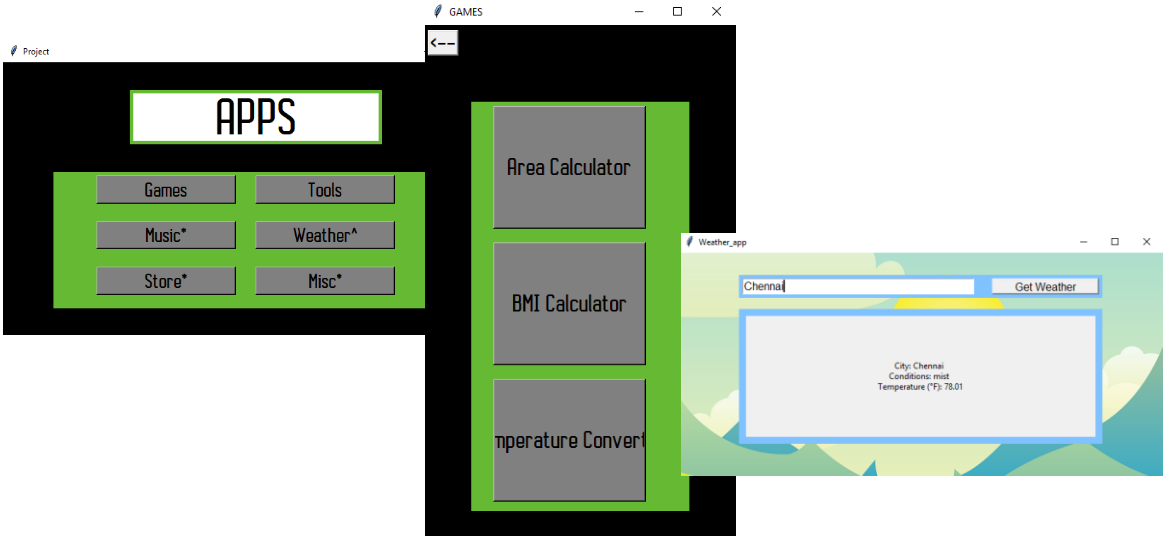
Delhi

```
City: Delhi
Conditions: mist
Temperature (°F): 62.6
```

NOTEABLE FEATURES

- ## Responsive

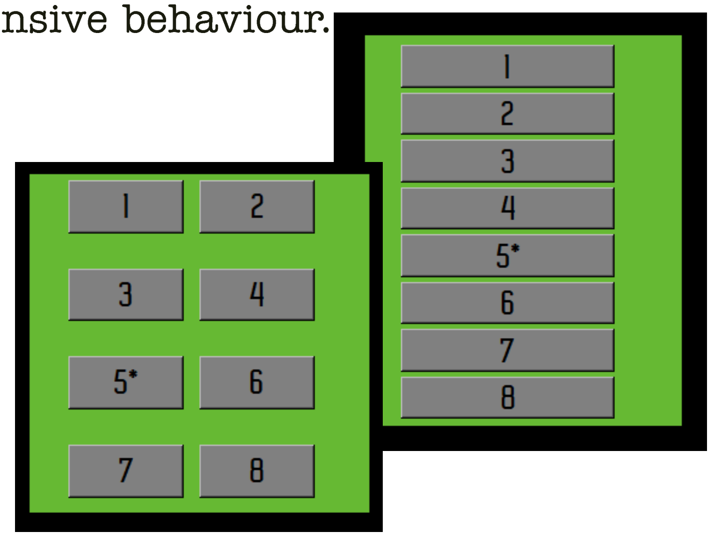
  - Each of the Tkinter windows can be resized without affecting the objects inside the frames.



City: Chennai
Conditions: mist
Temperature (°F): 78.01

- ## Easy to create buttons

  - Any number of buttons can be Added and they will fit in their parent frames and will still show responsive behaviour.



CREDITS & BIBLIOGRAPHY

- Libraries and modules used
  - Tkinter
  - Winsound
  - Random
  - Functools – Partial
  - Os
  - Sys
  - Pygame
  - Requests
  - Pillow (PIL)
  - Urllib (Urllib.request)
- Weather app information API
  - https://api.openweathermap.org/data/2.5/weather
- Program credits
  - *Tic Tac Toe* – https://github.com/abhishek305
  - *Dodge the blocks* - https://github.com/KeithGalli/
  - *Weather App* - https://github.com/KeithGalli/