



PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100 Feet Ring Road, BSK III Stage, Bengaluru-560 085

Department of Computer Science & Engineering

Title: Snake Game

Team member details: PES2UG21CS283 – Maryam Khan

PES2UG21CS351 – Niveditha Deepak

PES2UG21CS355 – P Niveditha

Abstract

This project, titled 'Snake Game', is an interesting and skilful computer action game. This project gave us an opportunity to learn and explore various python functions. We learnt how to implement Python's Graphic User Interface, Tkinter and turtle functions. The turtle function is a standalone function that can also be embedded in a larger Tkinter program.

The objective of this project is to create a source of entertainment, "Disconnect to Reconnect". It requires players to quickly assess their surroundings and find the safest route to a point. This is an excellent opportunity to learn about spatial awareness and plan ahead for your next move. It improves hand-eye coordination. It requires patience in order to grow and a cool head once you inevitably lose. These valuable skills promote one's personal development.

Contents

Introduction:	4
Design/Implementation:	5
Testing:	14
Result and Analysis:	16
Conclusions & future enhancements:	19
References:	20

Introduction:

Snake first appeared in 1997 on the Nokia 6110, along with the games Logic and Memory. It was programmed by Taneli Armanto, a design engineer in Nokia. The concept originated from the 1976 arcade game 'Blockade', developed and published by Gremlin. The original snake game was a famous mobile game and became an entire generation's obsession.

Snake game is a computer action game, whose goal is to control a snake to move and collect food. The player controls a long, thin creature, resembling a snake, which roams around on a bordered plane, picking up food, trying to avoid hitting its own tail or the edges of the playing area. Each time the snake eats a piece of food, its tail grows longer, and its speed increases making the game increasingly difficult. The user controls the direction of the snake's head (up, down, left, or right), and the snake's body follows.

With the evolution of graphics and python, we attempted to innovate the game with developed app-like features. We are excited to present the game to you.

Our snake game, comes with user input, instructions on how to play the game and even a 'play again' button which gives user the option to play the game any number of times he/she wishes to.

Design/Implementation:

```
#import module
import turtle
import time
import random
from tkinter import *

#creating window1
wn = Tk()
wn.title("Snake game")
wn.geometry("600x600")
bg = PhotoImage(file = "python-bg.png")
lb1 = Label(wn, image = bg)
lb1.place(x = 0, y = 0)

#creating window2
def launch():
    global top
    top = Toplevel()
    wn.withdraw()
    top.geometry("600x600")
    top['bg'] = '#88a704'
    global l2
    #creating entry boxes
    l1 = Label(top, text = 'Enter user id: ',bg = '#88a704',font = 20)
    l1.grid(row = 50,column = 40,pady = 5)
    l2 = Entry(top, font = ('calibre',20))
    l2.grid(row = 50,column = 49,pady = 5)
    l6 = Label(top, text = 'Enter age: ',bg = '#88a704',font = 20)
    l6.grid(row = 60,column = 40,pady = 5)
```

```

17 = Entry(top,font = ('calibre',20))
17.grid(row = 60,column = 49,pady = 5)
18 = Label(top, text = 'Enter country: ',bg = '#88a704',font =20)
18.grid(row = 70,column = 40,pady = 5)
19 = Entry(top, font = ('calibre',20))
19.grid(row = 70,column = 49,pady = 5)
Button(top, text="CONTINUE", command=launch2,font = 24).grid(row =80,column = 50)

```

#giving instructions

```
def launch2():
```

```
    top.withdraw()
```

```
    global top1
```

```
    top1 = Toplevel()
```

```
    top1.geometry("600x600")
```

```
    top1['bg'] = '#88a704'
```

```
    name = l2.get()
```

```
    top1.title(name + "'s snake game ")
```

```

    l3 = Label(top1,text ="How To Play:\n\nThe player controls a long, thin creature, resembling a
snake, \nwhich roams around on a bordered plane, picking up food, trying to \navoid hitting its own tail
or the edges of the playing area.\n Each time the snake eats a piece of food, \nits tail grows longer, the
score increases and its speed increases \nmaking the game increasingly difficult.\n The user controls
the direction of the snake's head and the \nsnake's body follows.\nUse the keys:\nW or w (Up) \nA or a
(Left)\nS or s(Down) \nD or d (Right) \n All the best :)",bg = '#88a704',font = 20 ).grid()

```

```
    Button(top1, text="CONTINUE", command=launch3,font = 24).grid()
```

```
def close():
```

```
    top4.withdraw()
```

#creating snake game

```
def launch3():
```

```
    top1.withdraw()
```

```
    window = turtle.Screen()
```

```
    turtle.setup(600,600)
```

```
window.title('Snake Game')
window.colormode(255)
window.bgcolor(136,167,4)
window.tracer(0)
delay = 0.1
# setting up the snake head
the_snake = turtle.getturtle()
the_snake.speed(0)
the_snake.fillcolor('pink')
the_snake.shape('square')
the_snake.resizemode
the_snake.turtlesize(1,1)
the_snake.showturtle()
the_snake.penup()
the_snake.direction = "stop"
#setting up the snake food
snake_food = turtle.Turtle()
snake_food.speed(0)
snake_food.shape("square")
snake_food.color("light blue")
snake_food.turtlesize(1,1)
snake_food.penup()
snake_food.goto(0,100)
snake_food.showturtle()
snake_food = turtle.Turtle()
snake_food.speed(0)
snake_food.shape("square")
snake_food.color("orange")
snake_food.turtlesize(1,1)
snake_food.penup()
snake_food.goto(0,100)
#score display
```

```
score = 0
High_Score = 0
pen = turtle.Turtle()
pen.speed(0)
pen.shape("square")
pen.color("black")
pen.hideturtle()
pen.penup()
pen.goto(0, 260)
pen.write("Score: 0 High Score: 0", align="center", font=("Times New Roman", 24, "bold"))
#functions
def go_up():
    if the_snake.direction != "down":
        the_snake.direction = "up"

def go_down():
    if the_snake.direction != "up":
        the_snake.direction = "down"

def go_left():
    if the_snake.direction != "right":
        the_snake.direction = "left"

def go_right():
    if the_snake.direction != "left":
        the_snake.direction = "right"

def move():
    if the_snake.direction == "up":
        y = the_snake.ycor()
        the_snake.sety(y + 20)
```



```
if the_snake.direction == "down":  
    y = the_snake.ycor()  
    the_snake.sety(y - 20)
```

```
if the_snake.direction == "left":  
    x = the_snake.xcor()  
    the_snake.setx(x - 20)
```

```
if the_snake.direction == "right":  
    x = the_snake.xcor()  
    the_snake.setx(x + 20)
```

```
#keyboard bindings
```

```
window.listen()
```

```
window.onkeypress(go_up, "w")
```

```
window.onkeypress(go_up, "W")
```

```
window.onkeypress(go_down, "s")
```

```
window.onkeypress(go_down, "S")
```

```
window.onkeypress(go_left, "a")
```

```
window.onkeypress(go_left, "A")
```

```
window.onkeypress(go_right, "d")
```

```
window.onkeypress(go_right, "D")
```

```
segments = []
```

```
#main game
```

```
while True:
```

```
    window.update()
```

```
#checking for collision with the border
```

```
    if the_snake.xcor()>290 or the_snake.xcor()<-290 or the_snake.ycor()>290 or the_snake.ycor()<-  
290:
```

```

time.sleep(1)
the_snake.goto(0,0)
the_snake.direction = "stop"

#hiding the segments
for segment in segments:
    segment.goto(1000, 1000)

#clearing the segments list
segments.clear()

#reseting the score
score = 0

#reseting the delay
delay = 0.1

pen.clear()
pen.write("Score: {} High Score: {}".format(score, High_Score), align="center", font=("Times
New Roman", 12, "bold"))
turtle.Screen().bye()
global top4
top4 = Toplevel()
top4.title("Snake game")
top4.geometry("600x600")
bg = PhotoImage(file = "GAME-OVER.png")
lb5 = Label(top4, image = bg)
lb5.place(x = 0, y = 0)
Button(top4, text="PLAY AGAIN",command = lambda: [close(), launch3()]).place(x=180,
y=470)

```

```

#check for a collision with the food
if the_snake.distance(snake_food) < 20:
    #moving the food to a random spot
    x = random.randint(-290, 290)
    y = random.randint(-290, 290)
    snake_food.goto(x,y)

#adding a body segment
body_segment = turtle.Turtle()
body_segment.speed(0)
body_segment.shape("square")
body_segment.color("purple")
body_segment.penup()
segments.append(body_segment)

#shortening the delay
delay -= 0.01

#increasing the score
score += 10

if score > High_Score:
    High_Score = score

pen.clear()
pen.write("Score: {} High Score: {}".format(score, High_Score), align="center", font=("Times
New Roman", 12, "bold"))

#moving the end segments first in reverse order
for index in range(len(segments)-1, 0, -1):
    x = segments[index-1].xcor()

```

```
y = segments[index-1].ycor()
segments[index].goto(x, y)

#moving segment 0 to where the head is
if len(segments) > 0:
    x = the_snake.xcor()
    y = the_snake.ycor()
    segments[0].goto(x,y)
move()

#checking for head collision with the body segments
for segment in segments:
    if segment.distance(the_snake) < 20:
        time.sleep(1)
        the_snake.goto(0,0)
        the_snake.direction = "stop"

#hiding the segments
for segment in segments:
    segment.goto(1000, 1000)

#clearing the segments list
segments.clear()

#reseting the score
score = 0

#reseting the delay
delay = 0.1

#updating the score display
pen.clear()
```

```
pen.write("Score: {} High Score: {}".format(score, High_Scoreigh_score), align="center",  
font=("Courier", 24, "normal"))
```

```
time.sleep(delay)
```

```
window.mainloop()
```

```
Button(wn, text="CONTINUE", command=launch, font = 25).pack(pady=10)
```

```
wn.mainloop()
```

Testing:

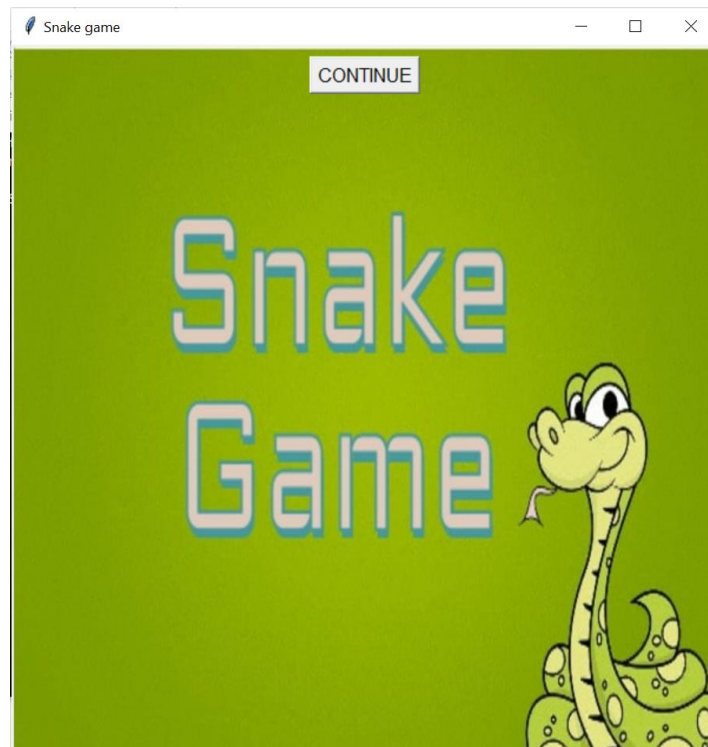
- 1) We create Window 1 using tkinter graphic user interface. It has a 'continue' button that takes the user to window 2.
- 2) **Launch()** - It is a function created using tkinter used to take an input about Username, Age and Country from the user. It also has a 'continue' button that takes the user to window 3.
- 3) **Launch2()** - It is a function that takes the username input and displays it as the title for window 3. Window 3 contains the instructions on how the Snake Game works. A 'continue' button is added to start the Snake game which opens window 4.
- 4) **Launch3()** - It is a function that opens window 4 created using the turtle function. We set up the game screen, snake head, snake food and display. It runs in window 4
- 5) **go_up()** - It is a function to make the snake head turn upwards if it is not facing downwards. It is present inside launch3(). It runs in window 4
- 6) **go_down()** - It is a function to make the snake head turn downwards if it not facing upwards. It is present inside Launch3(). It runs in window 4
- 7) **go_left()** - It is a function to make the snake head turn left if it is not facing right. It is present inside Launch3(). It runs in window 4
- 8) **go_right()** - It is a function to make the snake head turn right if it is not facing left. It is present inside Launch3(). It runs in window 4
- 9) **move()** - It is a function to move the snake upwards, downwards ,leftward or rightward using keys (W,S,A,D or w,s,a,d) according to the function used. It increases the x or y coordinate by 20 units. It is present inside Launch3() function. It runs in window 4.
- 10) Window 4 is the Snake game which is seen by the user where he can control the movement of the snake and collect food, trying to avoid hitting its own tail or the edges of the playing area. If he is not careful the game will end.
- 11) Window 5 will open which has a 'Game Over' screen with a 'play again' button which will take the user to window 4 to play the game. This window is created using tkinter.

The features used in the Snake Game are mentioned below:

- ▶ **Turtle**- It is a pre-installed library in python which is used for creating shapes, picture, and game.
- **Time** – It is used for counting the number of seconds elapsed
- **Random** – This module is used to generate snake food at random positions on the screen
- **penup()** – It stops drawing of the turtle pen
- **speed()** – It is used to control the speed of the snake
- **color()** – It returns or set pen color and fill color
- **shape()** – It set turtle shape to the shape of a given name
- **hideturtle()** – It makes the turtle invisible
- **xcor()** – Return the turtle's x coordinate
- **ycor()** – Return the turtle's y coordinate
- ▶ **Graphical User Interface with Tkinter package**- Different geometric methods – Tk, mainloop, Creating simple GUI - buttons, canvas, labels, entry fields, dialogs Widgets - sizes, fonts, colours layouts, nested frames

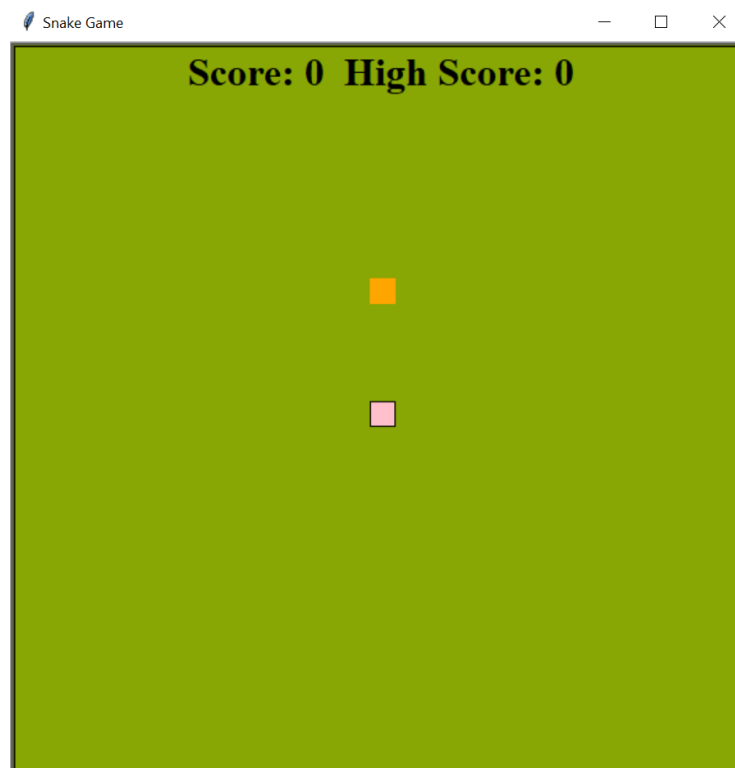
Result and Analysis:

Window 1: Start screen using tkinter graphic user interface

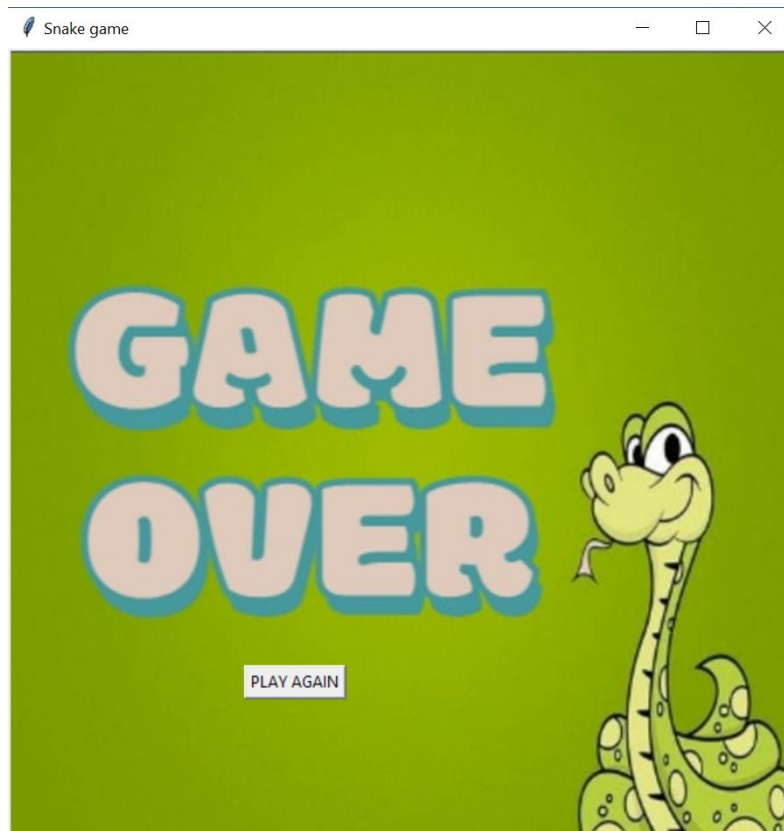


Window 2: User input screen using tkinter to make the game more personalized

A screenshot of a Tkinter window titled "Snake game". The window has a green background. On the left side, there are three input fields with labels: "Enter user id:", "Enter age:", and "Enter country:". Each label is followed by a white rectangular input box. On the right side, there is a white button labeled "CONTINUE".

Window 3: Instructions manual screen made using tkinter**Window 4:** Snake Game window made using python turtle functions

Window 5: Game over window with a Play Again option made using tkinter which goes to Window 4



Conclusions & future enhancements:

In our Snake Game, with the evolution of graphics and python, we attempted to innovate the game with developed app-like features. Our snake game, comes with user input, 'instructions' on how to play the game and even a 'play again' button which gives user the option to play the game any number of times he/she wishes to.

In future, we'd like to personalize the game by creating a data base where the users can store their game achievements and scores. We would also like to add power ups and detriments to the game to enhance the difficulty of the game.

We are thankful to the faculty for the opportunity to do this project. Through the process, our ability to work in a team has certainly improved and our knowledge of python coding has intensified.

References:

Python turtle graphics:

<https://docs.python.org/3/library/turtle.html>

Python tkinter tutorial:

<https://www.google.co.in/amp/s/www.geeksforgeeks.org/python-tkinter-tutorial/amp/>

We have referred to the Notes of Unit 3: Functions and GUI to learn about tkinter