Yılan game

| Date started | Date completed |
| --- | --- |
| 01.05.2020 | 01.06.2020 |

ŞÜKRÜ TAHA DEMİRER – TOLGA GÜLDÜTUNA

Analysis

Try and create 3 or more key success criteria for your program.

**Success Criteria:**

1. The program should work interactive.
2. The program should work without any crashes.
3. The program will have a simple GUI

Yılan Game

PsuedoCode:

Renkler are "orange", "red", "yellow", "green", "brown", "purple”.

a,b,c random & choice in renkler.

Puan is 0

Enyuksek is 0.

delay is 0.05.

oyunAlani created in turtle.Screen()

yilan,yiyecek,yilanvucut,bildirim are created in turtle.Turtle().

New list created for yilanVucutParcalari.

oyunAlani gived title then CRAZY PYTHON

oyunAlani of the background color is determined black.

oyunAlani of the frame size are 600 and 600.

oyunAlani was traced 0

yilan’s speed is 0

yilan’s shape is circle.

yilan’s color is “a”

yiyecek’s speed is 0

yiyecek’s shape is square.

yiyecek’s color is “b”

yiyecek is created in penup.

yiyecek is determined 0 to 100.

yilanvucut’s speed is 0

yilanvucut’s shape is square.

yilanvucut’s color is gray.

yilanvucut is created in penup.

yilanvucut send in hideturtle.

yilanvucut is determined 0 to 260.

yilanvucut was writed "Puan: 0 En Yüksek Puan: 0", align="center", font=("Arial", 22, "normal")".

#yukari section

if yilan.direction is not equal asagi

yilan.direction is equal yukari

endif

#asagi section

if yilan.direction is not equal yukari

yilan.direction is equal asagi

endif

#sola section

if yilan.direction is not equal sag

yilan.direction is equal sol

endif

#saga section

if yilan.direction is not equal sol

yilan.direction is equal sag

endif

#hareket section

if yilan.direction is equal yukari

y is equal yilan in ycor

yılan's of the sety is increased 20.

endif

if yilan.direction is equal asagi

y is equal yilan in ycor

yılan's of the sety is deccreased 20.

endif

if yilan.direction is equal sol

x is equal yilan in ycor

yılan's of the setx is deccreased 20.

endif

if yilan.direction is equal sag

x is equal yilan in ycor

yılan's of the setx is increased 20.

endif

#oyunuKapat section

oyunAlani is determined bye.

oyunAlani is listened.

oyunAlani's onkeypress set Up,Down,Left,Right and ESC keys.

while true

oyunAlani's updated.

count increase plus 1

if yilan's xcor is greater than 290 or less than 290 and yilan's ycor is greater than 290 or less than 290

time of sleep is 2

turtle wrote ""OYUN YENIDEN BAŞLIYOR!", align="center", font=("Arial", 26, "normal")"

yılan go to 0 to 0

yiyecek go to 100 to 100

yilan's direction is equal stop.

For each parca yilanVucutParcalari do

parca go to 1000 to 1000

End For

yilanVucutParcalari is clear

puan is 0.

delay is 0.05.

yilanvucut is clear

yilanvucut write "Skor: {} En Yüksek Skor: {}".format(puan, enyuksek), align="center", font=("Courier", 24, "normal")

end if

if yilan's distance of the yiyecek is less than 20.

x is randint -290 to 290.

y is randint -290 to 290.

yiyecek go to 0 to 0

yeniParca is determined turtle.

yeniParca's speed is entered 0

yeniParca's shape is square.

yeniParca's color is entered c.

yeniParca.penup()

yeniParca

yilanVucutParcalari append yeniParca

delay is decrease 0.001

puan is increase 10

if puan is greater en yuksek

enyuksek is equal paun.

end if

yilanvucut is clear.

yilanvucut write ""Skor: {} En Yüksek Skor: {}".format(puan, enyuksek), align="center", font=("Courier", 24, "normal")"

end if

For each index do range in yilanVucutParcalari's length -1, 0, -1.

x is yilanVucutParcalari's index remove 1 in xcor.

y is yilanVucutParcalari's index remove 1 in ycor.

yilanVucutParcalari's index go to x to y.

End for.

if yilanVucutParcalari's length is greater than 0.

x is yilan in xcor.

y is yilan in ycor.

yilanVucutParcalari's 0 go to x to y.

endif

hareket run.

For Each parça do yilanVucutParcalari

if parça's distance in yilan is less than 20.

turtle write "("OYUN YENIDEN BAŞLIYOR!", align="center", font=("Arial", 26, "normal"))"

time in sleep is determined 1.

yilan's 0 go to 0 to 0.

yilan's direction is equal stop.

For each parca do yilanVucutParcalari

parca's 0 go to 1000 to 1000.

End for

puan is 0.

delay is 0,1.

yilanvucut is clear

yilanvucut write "Score: {} High Score: {}".format(puan, enyuksek), align="center-top", font=("Courier", 24, "normal")"

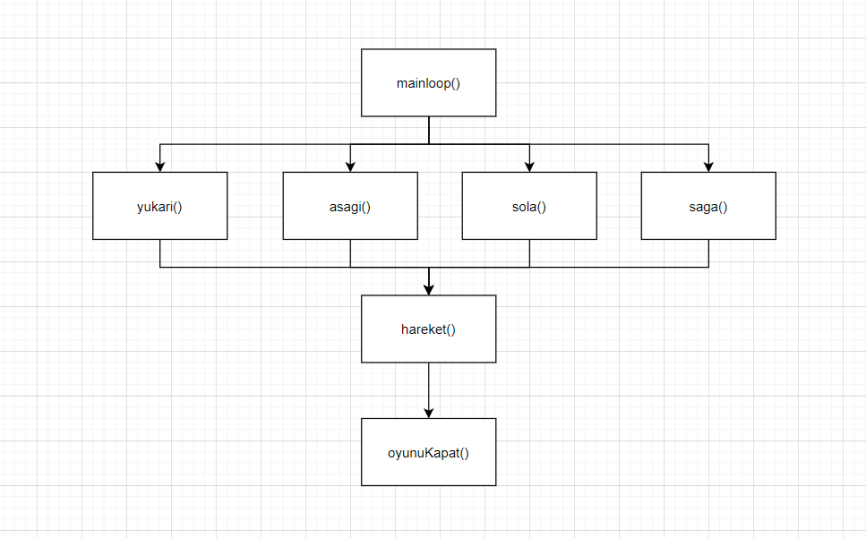
endif

time's sleep is determined delay.

End for

oyunAlani is runned.

UML



Test design

**My tests:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | What am I testing? | What data will I use? | Normal/Boundary/Erroneous? | Expected Result |
| 1 | Arrow Keys | User Input | Normal | Arrows should change the direction. |
| 2 | Function Tests | Collision Statements | Normal | The program should give reaction if any collision occurs. |
| 3 | Quit Program | User Input | Normal | The program should close if user presses the key “Q”. |
| 4 | Max High Score | High Score | Boundary | The maximum score can be max size of integer. But in the program, it should be 10\* max snake size can fit the screen. |
| 5 |  |  |  |  |

Development

**My program code:**

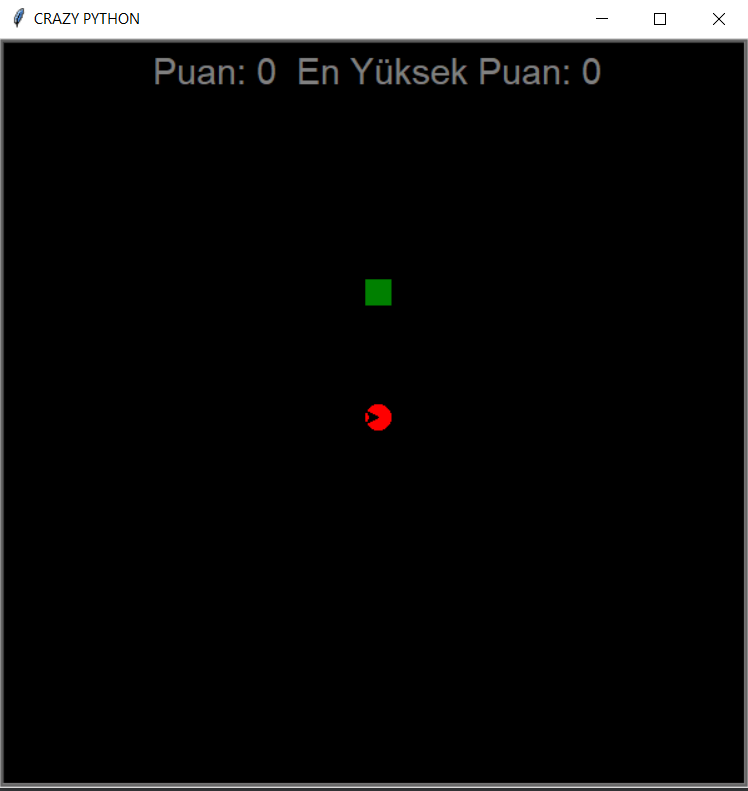
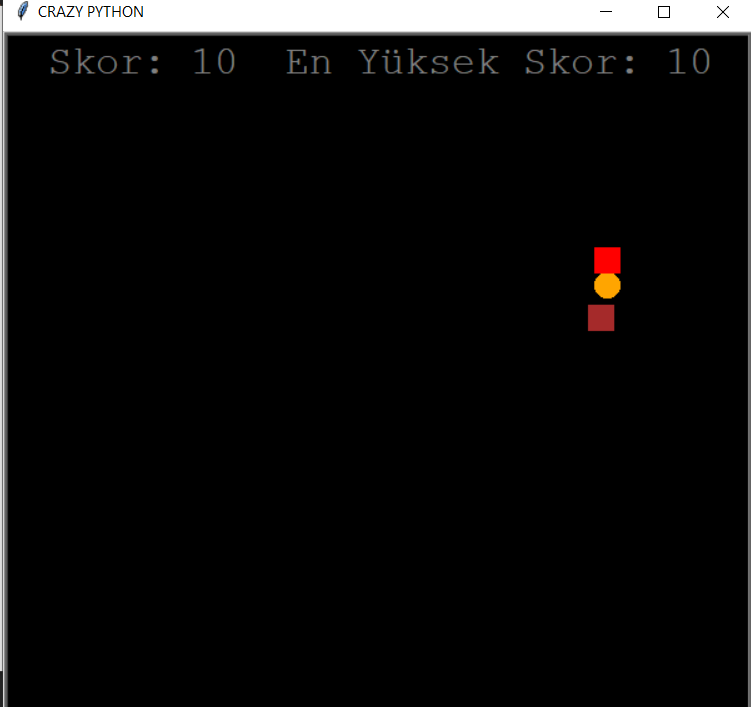
[**https://github.com/TolgaGuldu/Py-Snake-Game**](https://github.com/TolgaGuldu/Py-Snake-Game)

Testing

**My tests:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | What am I testing? | Expected result | Pass/Fail | Do I need to change my program? If so, how? |
| 1 | Food Collision | Score +10 | Pass | No |
| 2 | Border Collision | Exit The Game | Pass | No |
| 3 | Food Collision | Increase Snake Size | Pass | No |
| 4 | Food Point | Change The Food Point | Pass | No |
| 5 | Colors | Change Food and Snake’s Colors | Pass | No |

**My test screenshots:**

Food Collision and Changing Colors 

Increasing Snake Size

Yılan Game

**How successful was my program?**

The program is successful, it gets the commands with input, executes as it expected. The snake’s size increases in every collision with the food, the game stops every collision with the borders. Color of the food and snake changes in every new start of program. Snake game program keeps the instants score and high score. High score renews in every new high score. The program doesn’t need to hard tests because there is nothing but user inputs execution. The executed functions are working as it is expected, processed user input correctly and makes the changes of snake’s direction, food’s position, collision detecting etc. The program passed every test as a result of these situations. The GUI is so simple, so everyone can play the game easily. There is no spaghetti in the code, thus, the code is easy to understand and change.

**What new skills have I developed?**

We developed the skill of object oriented programming with Python. Learned the logic of Turtle framework, functions and commands in it. Learned how to work with sets, generate objects and use them.