Program and Design

Develop a mini game by Python language

Author: CHUANG, CHIH-HSUAN
Supervisor: Heng Zhong

Word count: 972

Date: April 5, 2021

Introduction

This is a basic course of learning fundamental knowledge of Python. The learning outcomes are to understand the features, advantages and applications of Python language and learn Python syntax. The purpose is to write a mini game by using Python language.

Ball Collision Game Design and Programming

Brief Introduction of Python

General speaking, Python is a very beginner-friendly programming language. First of all, Python is an interpreted language, so written code can be run without being compiled (Dimitriou & Hatzitaskos, 2015). In addition, when a programmer wants to run a program, as long as this part of the code has no syntax errors, they can run this code. Unlike other compiled language such as c + +, the programmer needs to compile all codes. If only one syntax error happens at compiling, all the codes cannot run.

Python, on the other hand, is an object-oriented programming (OOP) language. Brief speaking, OOP is to use such a language to create different objects, give them instructions, and let them execute those instructions (ibid). Compared with a process-oriented language, which has a disadvantage that a coder must simulate the process in their mind, and it makes it harder for multiple people to co-write the program; therefore, using an object-oriented language, more than one programmer can write code for different objects at the same time, which can be much more productive.

Python can develop web back-end. For instance, crawlers, one of powerful Python tools can collect data automatically and present them as user wishes (Lecture by Mirror Edu, 2020). Python makes data analysis more easily accessible for scholars who have never learned a programming language, than other computer language.

Admittedly, Python has its drawbacks. One of its biggest weaknesses is that while Python is easy to write, it is very inefficient for running compared to other languages (Lecture by Mirror Edu, 2020). During this course, I learned two compilers of Python - Idle and VSCode.

Learning Process--Basic Syntax

All the following contents are notes I took from the lectures delivered by Xian Zhong in Mirror Education.

Basic Variable

Use Integer to store a data type of integer number;

Float to store a data type of decimal number;

String to store a data type of character (include numbers and symbols);

And bool has only two types- True and False.

Basic Operational Rule

“+” represents addition; “-” means subtraction; “*” stands for multiplication; “/” for division; “//” for exact division; “%” for finding remainder; “**” for calculating power.

Features of List

As with Python ‘s variables, the types of data in a list can be changed flexibly, and different types of data can exist in the same list. And the list doesn't have to declare a length. Every element in list has particular index which indicated its position in the list. And index starts from 0.

E.g. a = {1, ”a”, True} // index for 1, ”a”, True are 0, 1, 2 respectively

A list of Common Methods

Table. 1 Function of methods

Method	Function
Append()	add elements at the end of list
Insert()	insert elements in any specific position
Len()	obtain quantity of elements in the list
Pop()	delete default last elements or specify which to delete by index
Remove()	Remove(): search and delete any element in the list

Function

Functions are the internal encapsulation of Python. We can break a large piece of code into different functions. By calling different functions, we can break a complex task into simpletasks.

E.g.
```
def addsum(number 1, number 2, number 3 ):
    total = 0
    For i in range(number+1):
        total = total + i
    return total
```

// this is a function for calculating the sum of several numbers

File

IO programming means interaction between user and program like input() and output(), or the interaction between software and hardware system for example, file system.

Reading and writing to files is the most common IO programming, and Python has built-in functions for reading and writing to files.

We must know that the function of read and write files on disk is provided by the operating system, and modern operating systems do not allow the ordinary procedure directly disk operation, so, speaking, reading and writing file requires to the operating system to open a file object, then through from this number to read from the file object or the data is written to the file object.

So, how to write a file? Actually, we have two modes for file writing: one is “w” mode, which can clear all contents in the file and write at beginning of the file. The other is “a” mode, which can add content to the end of the data in file. The two modes will also automatically help construct new file if you haven’t got one.

Format is: f = file (“file name”, “mode”)

E.g. f = file (“1.txt”, “w”)

Turtle

Turtle is like as a painter on the screen. Programmer can use code and functions to control it to move.

E.g.

```
import turtle

littleturtle.color("blue")

for i in range(4):
    littleturtle.forward(200)
    littleturtle.right(90)
```

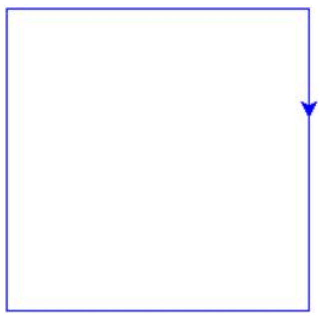

Fig.1 Graph of left-hand side example

Final project

After learning syntax of Python, I wrote a set of codes for the Pong Game (See Appendix). The game contains two players - A and B, Player A uses keys of “w” and “s” to shift Paddle A upwards and downwards respectively. Player B uses keys of “up” and “down” to shift Paddle B upwards and downwards respectively. Two players use their paddle to bounce the ball to their opposite side. If one doesn’t catch the ball, the other player will acquire 1 point.

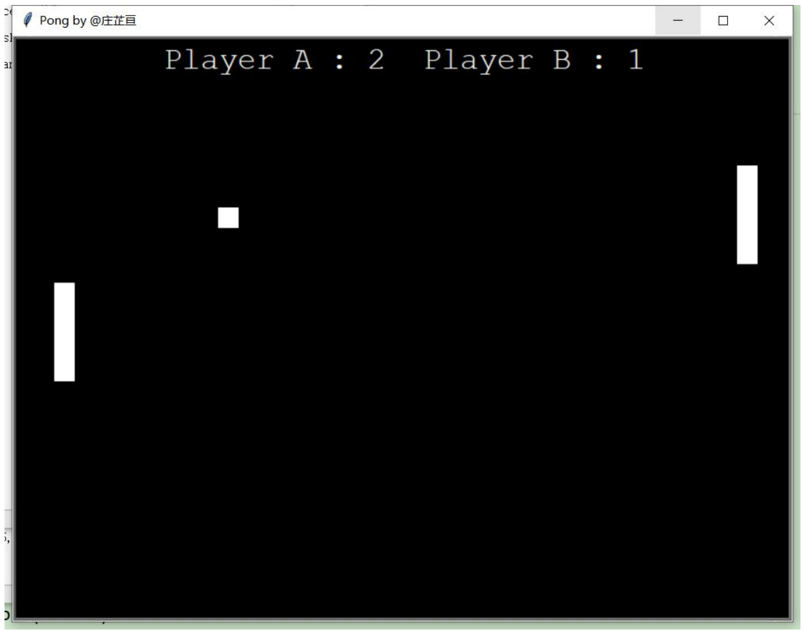

Fig.2 Graph of game windows

Conclusion

All in all, after knowing the features, advantages and syntax Python language, I make an attempt to apply them to produce the mini game successfully even though there is much room to improve in the game. This program made me lay a foundation for my future study of coding and the room of improvement is a drive for me to study further.

Reference

[1] Dimitriou, K. and Hatzitaskos, M. , 2015, Core Computer Science For the IB Diploma Program, Express Publishing, Newbury, United Kingdom

Appendix

```
import turtle

#window
wn = turtle.Screen()
wn.title("Pong by @庄芷亘")
wn.bgcolor("black")
wn.setup(width=800,height=600)
wn.tracer(0)
```

//define the author information, size, background format

```
#ball
ball = turtle.Turtle()
ball.speed(0)
ball.penup()
ball.shape("square")
ball.color("white")
ball.goto(0,0)
ball.dy = 0.1
ball.dx = 0.1
```

//define the speed, shape, color,initial position of the ball for collision

```
#paddle A
paddle_a = turtle.Turtle()
paddle_a.speed(0)
paddle_a.shape("square")
paddle_a.color("white")
paddle_a.penup()
```

//define shape, size speed, color of paddle A

```
paddle_a.goto(-350,0)

paddle_a.shapesize(stretch_wid=5,stretch_len=1)


#paddle B

paddle_b = turtle.Turtle()

paddle_b.speed(0)

paddle_b.shape("square")

paddle_b.color("white")

paddle_b.penup()

paddle_b.goto(350,0)

paddle_b.shapesize(stretch_wid=5,stretch_len=1)
```

//define shape, size speed, color of paddle B

```
#score board

pen = turtle.Turtle()

pen.speed(0)

pen.shape("square")

pen.color("white")

pen.penup()

pen.hideturtle()

pen.goto(0,260)

pen.write("Player A : 0    Player B : 0",align = "center",font=("Courier",24,"normal"))



score_a =0

score_b =0
```

//define shape, size speed, color of score board. Define the content, position and font of text on the board

```
#function control board to move

def paddle_a_up():
```

//define the function for moving paddle . move paddle up and down by adding or subtracting 20 unit of distance at y coordinate of paddle for every time click on the control key

```
    y = paddle_a.ycor()
    y += 20
    paddle_a.sety(y)


def paddle_a_down():
    y = paddle_a.ycor()
    y -= 20
    paddle_a.sety(y)


def paddle_b_up():
    y = paddle_b.ycor()
    y += 20
    paddle_b.sety(y)


def paddle_b_down():
    y = paddle_b.ycor()
    y -= 20
    paddle_b.sety(y)
```

#keyboard bindings

```
wn.listen()
wn.onkeypress(paddle_a_up,"w")
```

//define which control key correspond to which locomotion

//A use keys of “w”and “s” to shift paddle A upward and downwards respectively. b use keys of “up”and “down” to shift paddle B upward and downwards respectively.

```
wn.onkeypress(paddle_a_down,"s")

wn.onkeypress(paddle_b_up,"Up")

wn.onkeypress(paddle_b_down,"Down")



#main loop

while True:
```

//The ball move by repetition of adding same difference to its x and y coordinates at same time

```
    ball.setx(ball.xcor()+ ball.dx)

    ball.sety(ball.ycor()+ ball.dy)


    wn.update()


# Top and bottom

    if ball.ycor()>290:

        ball.sety(290)

        ball.dy = ball.dy* -1
```

// if the ball reach the top or bottom side of the windows, turn the difference negatively at direction on y axis , then the ball will bounce at the edge of window.

```
    if ball.ycor()<-290:

        ball.sety(-290)

        ball.dy = ball.dy* -1
```

// if the ball reach the paddle, turn the difference negatively at direction on x axis , then the ball will bounce at the paddle.

```
#left and right


#catch by paddles-> bounce

    if ball.xcor()<-340 and ball.ycor()<paddle_a.ycor()+50 and ball.ycor()>paddle_a.ycor()-50:

        ball.dx*=-1
```

// if the ball doesn't reach the paddle, place the ball to its initial position at beginning of the game

//then if A doesn't catch the ball, score of B add 1 point, else if B doesn't catch the board, score of A at 1 point

```
    elif ball.xcor()>340 and ball.ycor()<paddle_b.ycor()+50 and ball.ycor()>paddle_b.ycor()-50:
        ball.dx*=-1
```

#Not catch by paddles-> move the ball to origin and modify score

```
    if ball.xcor()>390:
        score_a = score_a + 1
        ball.goto(0,0)
        ball.dx = ball.dx* -1
        pen.clear()
        pen.write("Player A : {}    Player B : {}".format(score_a,score_b),align = "center",font=("Courier",24,"normal"))


    if ball.xcor() < -390:
        score_b = score_b + 1
        ball.goto(0,0)
        ball.dx = ball.dx* -1
        pen.clear()
        pen.write("Player A : {}    Player B : {}".format(score_a,score_b),align = "center",font=("Courier",24,"normal"))
```