Pac-Man

Contents

1 Background of Pac-Man

2 Goals of the Program

3 Approaches to the Goal

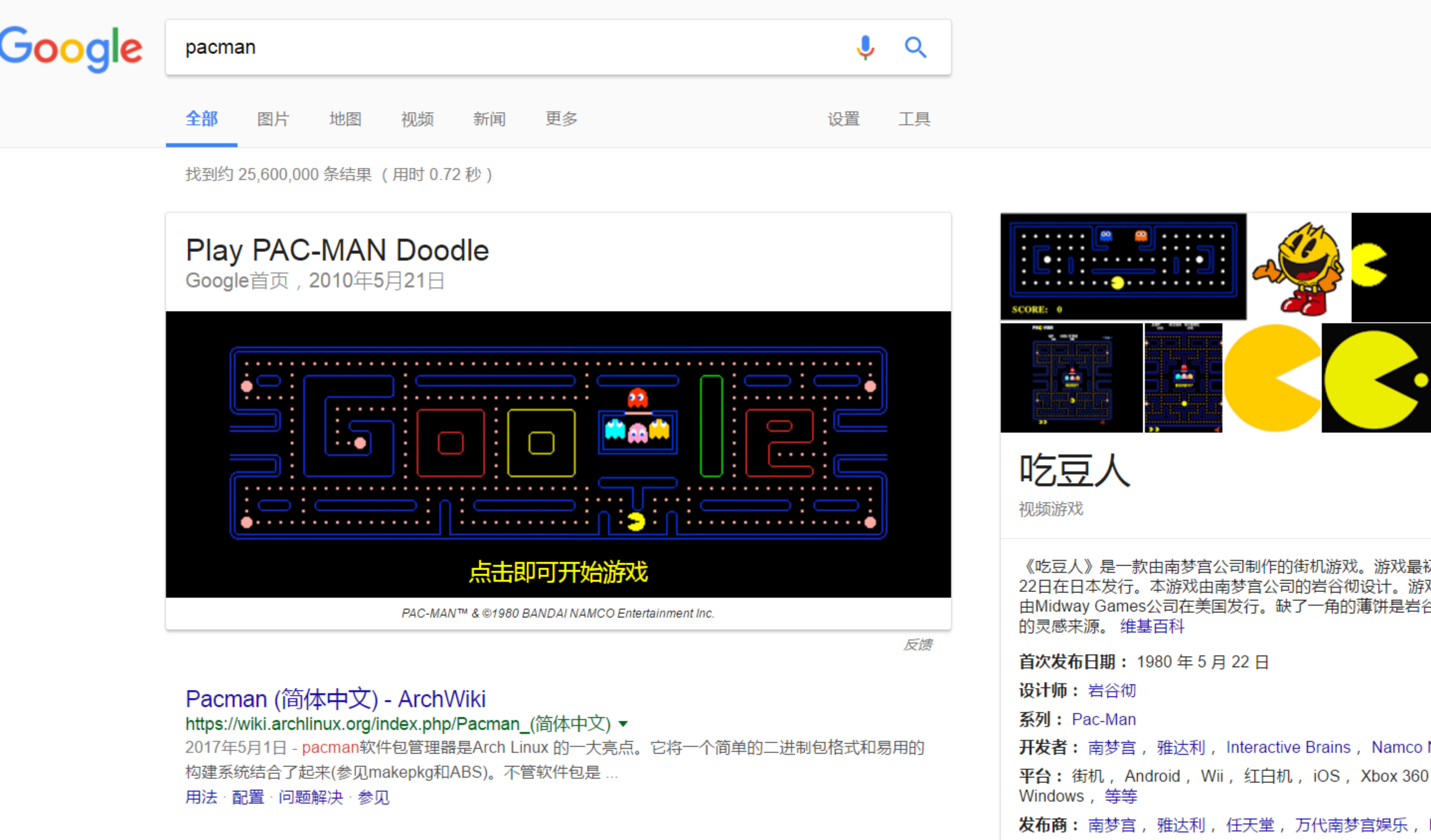
4 Improvements

Background

- Pac-Man is an arcade game developed by **Namco** and first released in Japan in May 1980.
- It was created by Japanese video game designer **Toru Iwatani.**

Background

- licensed for distribution in the United States by **Midway**
- generated more than $2.5 billion in quarters by the 1990s
- regarded as one of the **most influential video games** of all time
- Still popular nowadays



pacman

全部 图片 地图 视频 新闻 更多 设置 工具

找到约 25,600,000 条结果（用时 0.72 秒）

Play PAC-MAN Doodle

Google首页，2010年5月21日

点击即可开始游戏

PAC-MAN™ & ©1980 BANDAI NAMCO Entertainment Inc.

反馈

Pacman (简体中文) - ArchWiki

https://wiki.archlinux.org/index.php/Pacman_(简体中文) ▾

2017年5月1日 - pacman软件包管理器是Arch Linux 的一大亮点。它将一个简单的二进制包格式和易用的构建系统结合了起来(参见makepkg和ABS)。不管软件包是 ...

用法 · 配置 · 问题解决 · 参见

吃豆人

视频游戏

《吃豆人》是一款由南梦宫公司制作的街机游戏。游戏最初
22日在日本发行。本游戏由南梦宫公司的岩谷彻设计。游
由Midway Games公司在美国发行。缺了一角的薄饼是岩谷
的灵感来源。 维基百科

首次发布日期： 1980 年 5 月 22 日

设计师： 岩谷彻

系列： Pac-Man

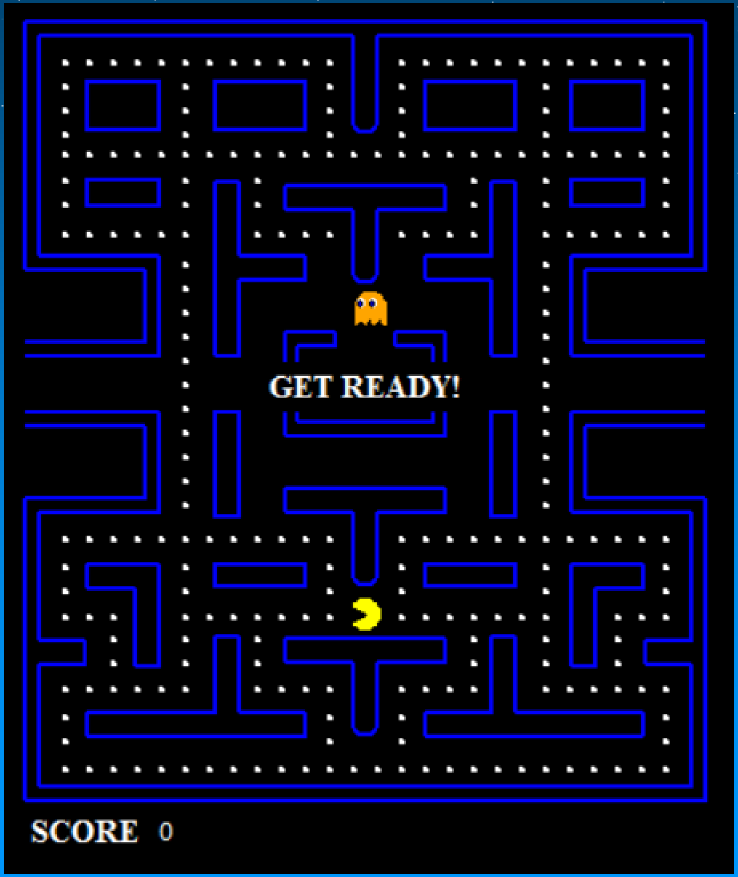
开发者： 南梦宫， 雅达利， Interactive Brains， Namco

平台： 街机， Android， Wii， 红白机， iOS， Xbox 360
Windows， 等等

发布商： 南梦宫， 雅达利， 任天堂， 万代南梦宫娱乐，

Game playing

- **Operation interface**



Goal

- Goal One

  Description of Beans

- Goal Two

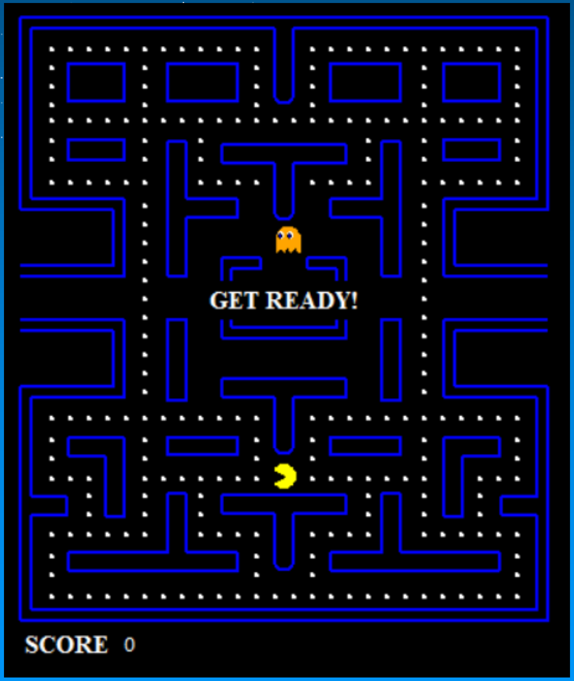
  Achieving the Function of Walls

- Goal Three

  Enabling the Control of Pac-Man

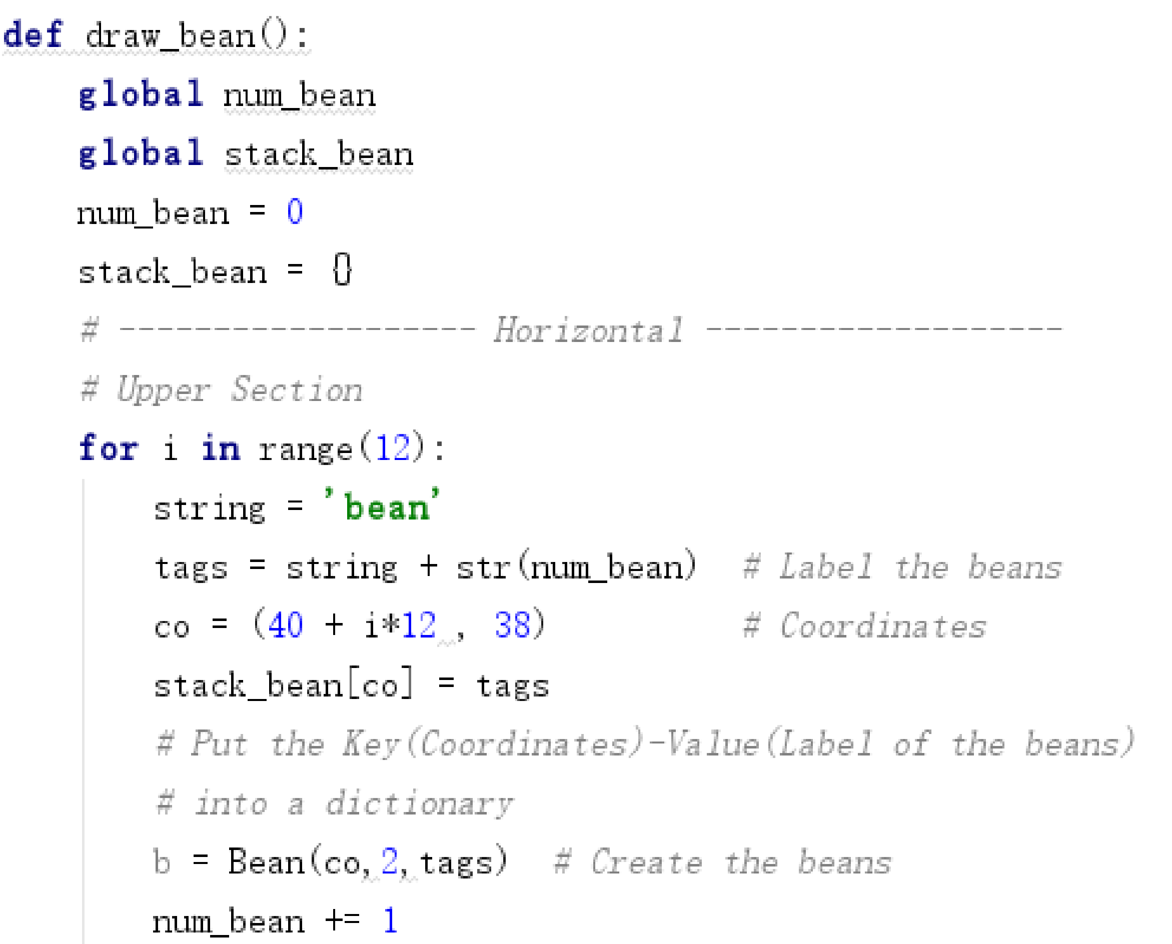
- Goal Four

  Setting Routes for the Enemies



Approach-beans

- **How to establish bean models?**



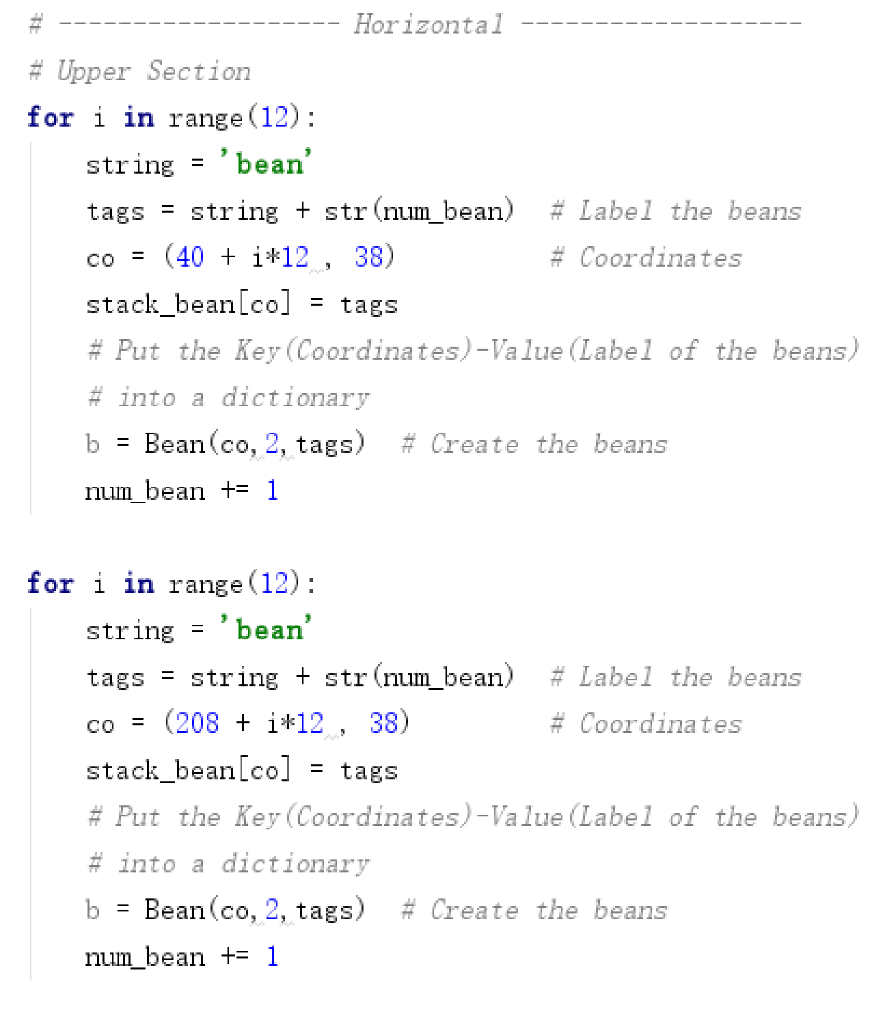
```
def draw_bean():
    global num_bean
    global stack_bean
    num_bean = 0
    stack_bean = {}
    # ------------------- Horizontal -------------------
    # Upper Section
    for i in range(12):
        string = 'bean'
        tags = string + str(num_bean)  # Label the beans
        co = (40 + i*12 , 38)          # Coordinates
        stack_bean[co] = tags
        # Put the Key(Coordinates)-Value(Label of the beans)
        # into a dictionary
        b = Bean(co, 2, tags)  # Create the beans
        num_bean += 1
```

# Global

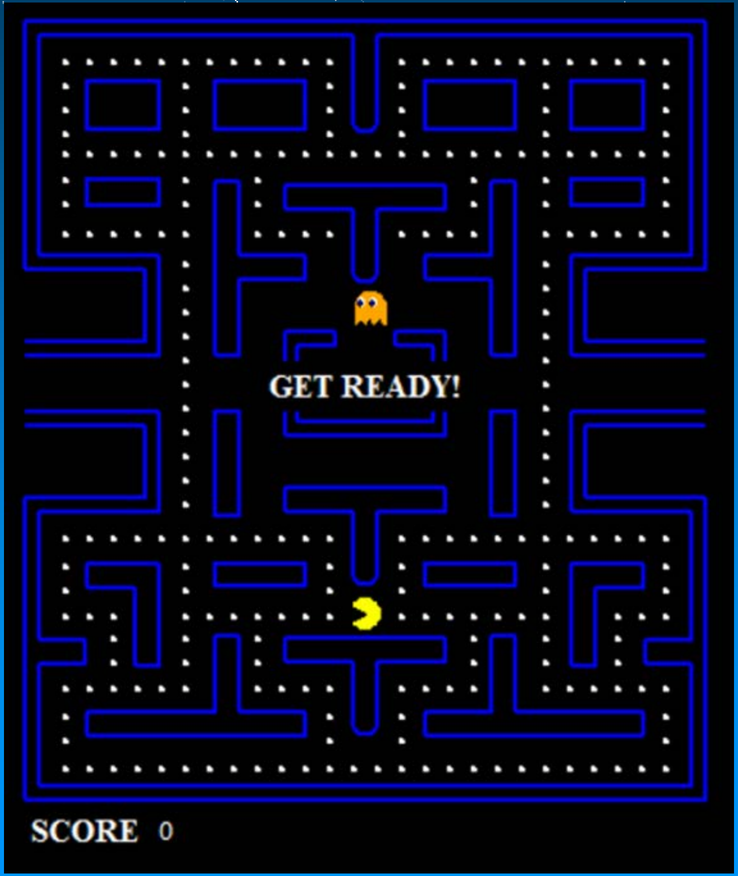
The variables will not be swapped after the function call.

Approach-beans

- **How to establish bean models?**



```
# -------------------- Horizontal --------------------
# Upper Section
for i in range(12):
    string = 'bean'
    tags = string + str(num_bean)  # Label the beans
    co = (40 + i*12 , 38)          # Coordinates
    stack_bean[co] = tags
    # Put the Key(Coordinates)-Value(Label of the beans)
    # into a dictionary
    b = Bean(co,2,tags)  # Create the beans
    num_bean += 1

for i in range(12):
    string = 'bean'
    tags = string + str(num_bean)  # Label the beans
    co = (208 + i*12 , 38)         # Coordinates
    stack_bean[co] = tags
    # Put the Key(Coordinates)-Value(Label of the beans)
    # into a dictionary
    b = Bean(co,2,tags)  # Create the beans
    num_bean += 1
```



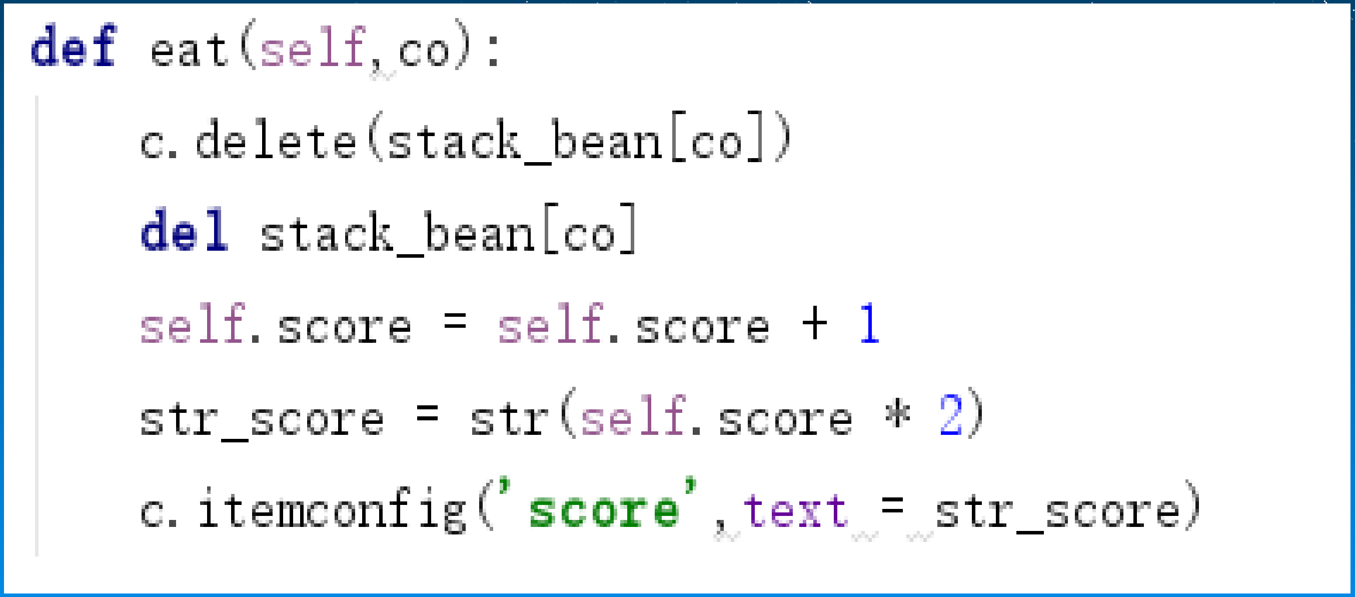
For intersections, beans can only be created once.

Approach-beans

- **How to make pacman eat beans?**

  1.”eat” function——delete the coordinates of beans and accumulate points

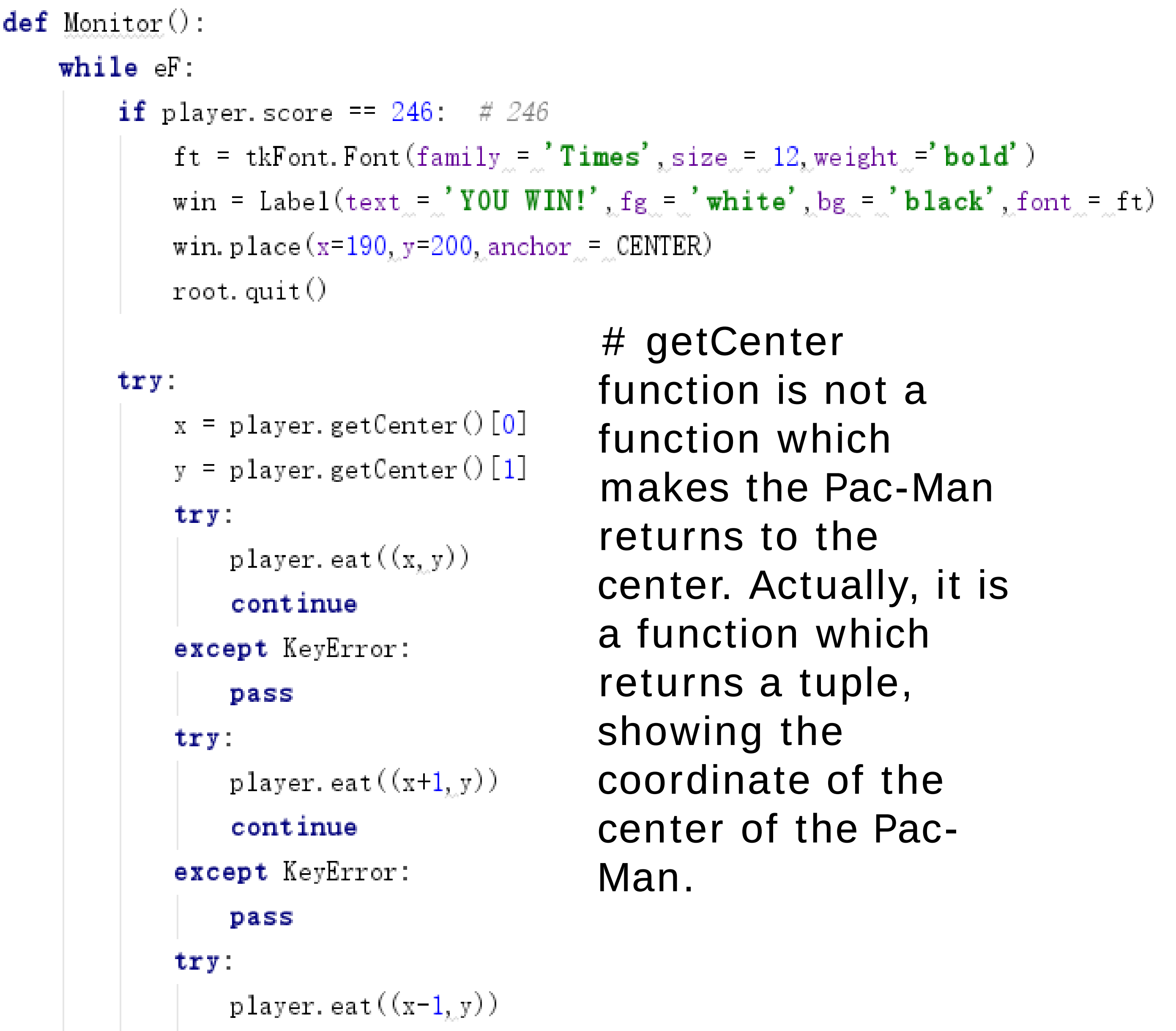
  2.monitoring function——whether beans coincide with its position.



```
def eat(self,co):
    c.delete(stack_bean[co])
    del stack_bean[co]
    self.score = self.score + 1
    str_score = str(self.score * 2)
    c.itemconfig('score',text = str_score)
```

# delete the image

# delete the key-value pair



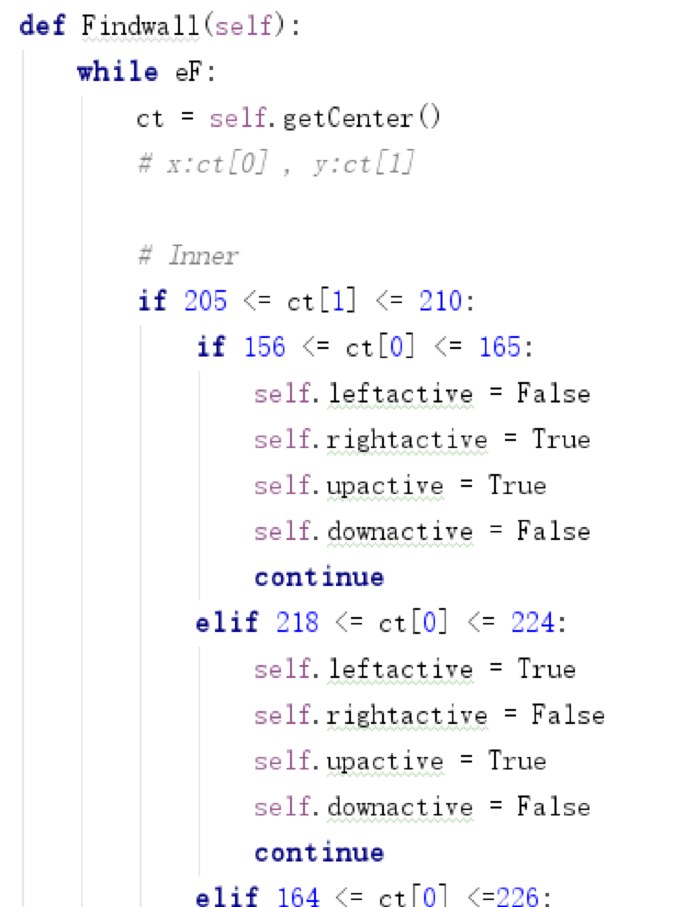
```
def Monitor():
    while eF:
        if player.score == 246:  # 246
            ft = tkFont.Font(family = 'Times',size = 12,weight ='bold')
            win = Label(text = 'YOU WIN!',fg = 'white',bg = 'black',font = ft)
            win.place(x=190,y=200,anchor = CENTER)
            root.quit()

        try:
            x = player.getCenter()[0]
            y = player.getCenter()[1]
            try:
                player.eat((x,y))
                continue
            except KeyError:
                pass
            try:
                player.eat((x+1,y))
                continue
            except KeyError:
                pass
            try:
                player.eat((x-1,y))
```

# getCenter function is not a function which makes the Pac-Man returns to the center. Actually, it is a function which returns a tuple, showing the coordinate of the center of the Pac-Man.

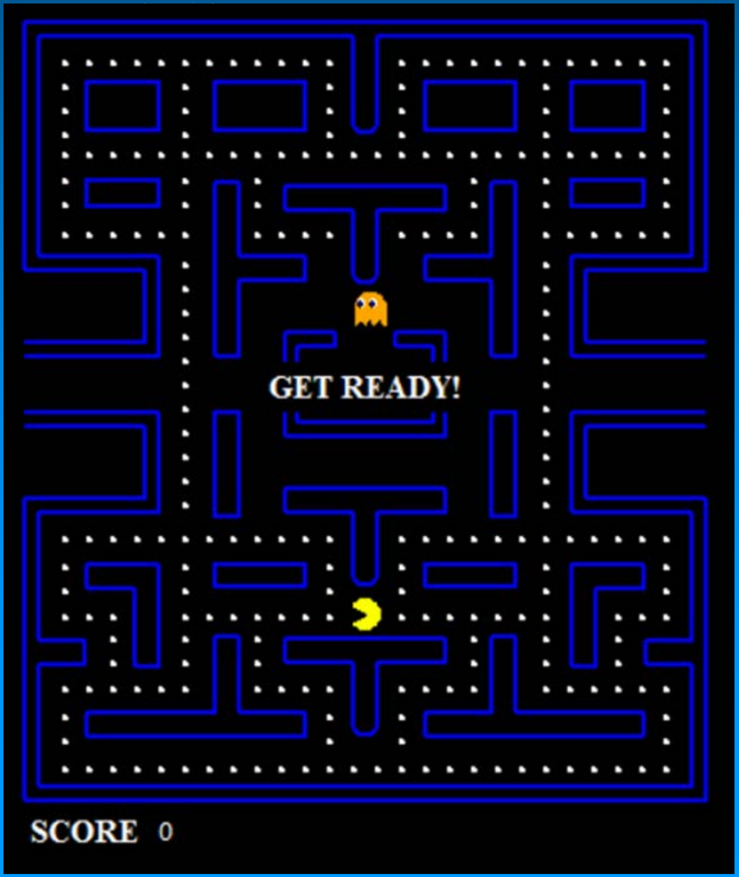
Approach-walls

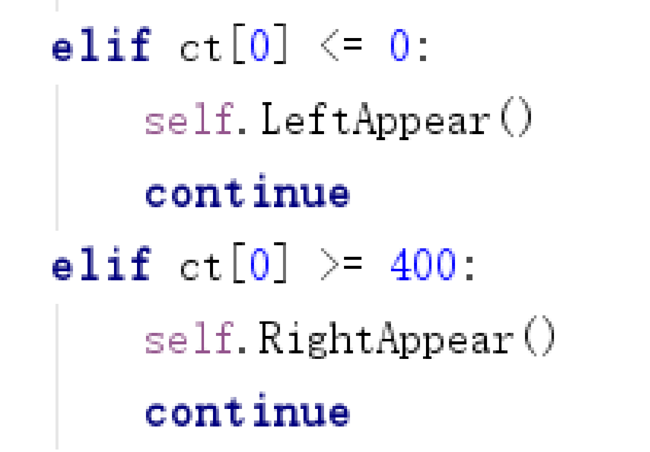
- **How to block the Pac-Man ?**

  Define the values “upactive”, “downactive”, “leftactive”, “rightactive” in different places.

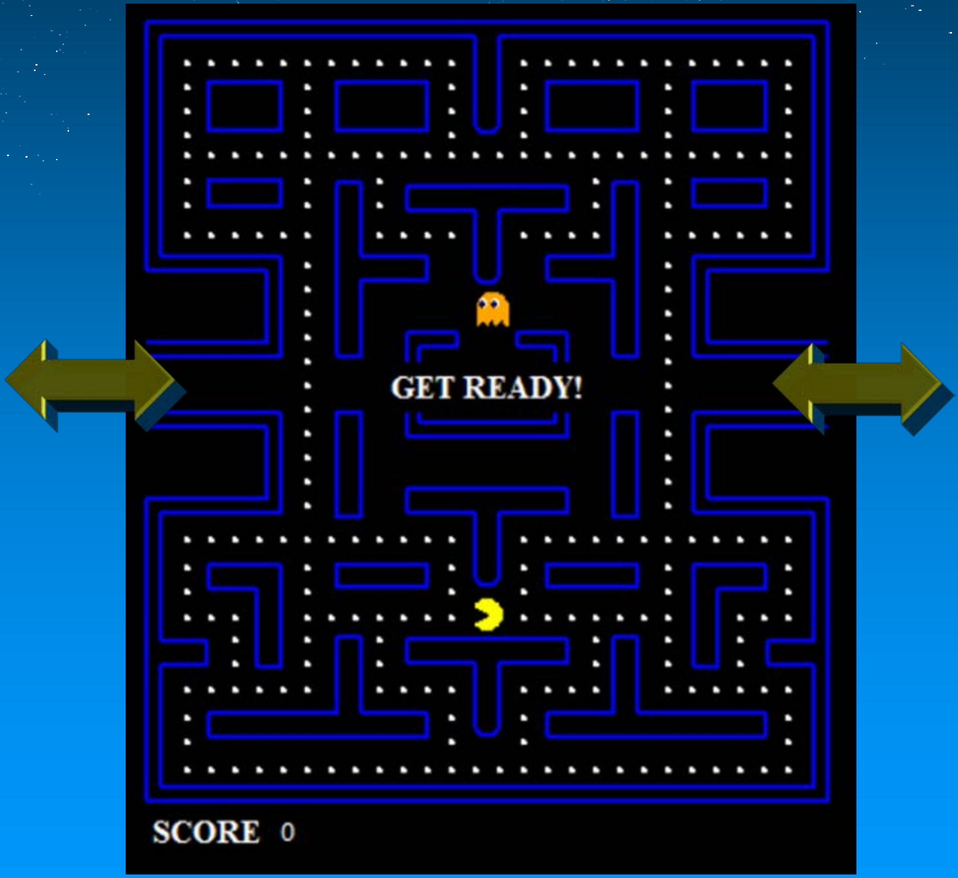


```
def Findwall(self):
    while eF:
        ct = self.getCenter()
        # x:ct[0] , y:ct[1]

        # Inner
        if 205 <= ct[1] <= 210:
            if 156 <= ct[0] <= 165:
                self.leftactive = False
                self.rightactive = True
                self.upactive = True
                self.downactive = False
                continue
            elif 218 <= ct[0] <= 224:
                self.leftactive = True
                self.rightactive = False
                self.upactive = True
                self.downactive = False
                continue
            elif 164 <= ct[0] <=226:
```





```
elif ct[0] <= 0:
    self.LeftAppear()
    continue
elif ct[0] >= 400:
    self.RightAppear()
    continue
```



Approach-pacman

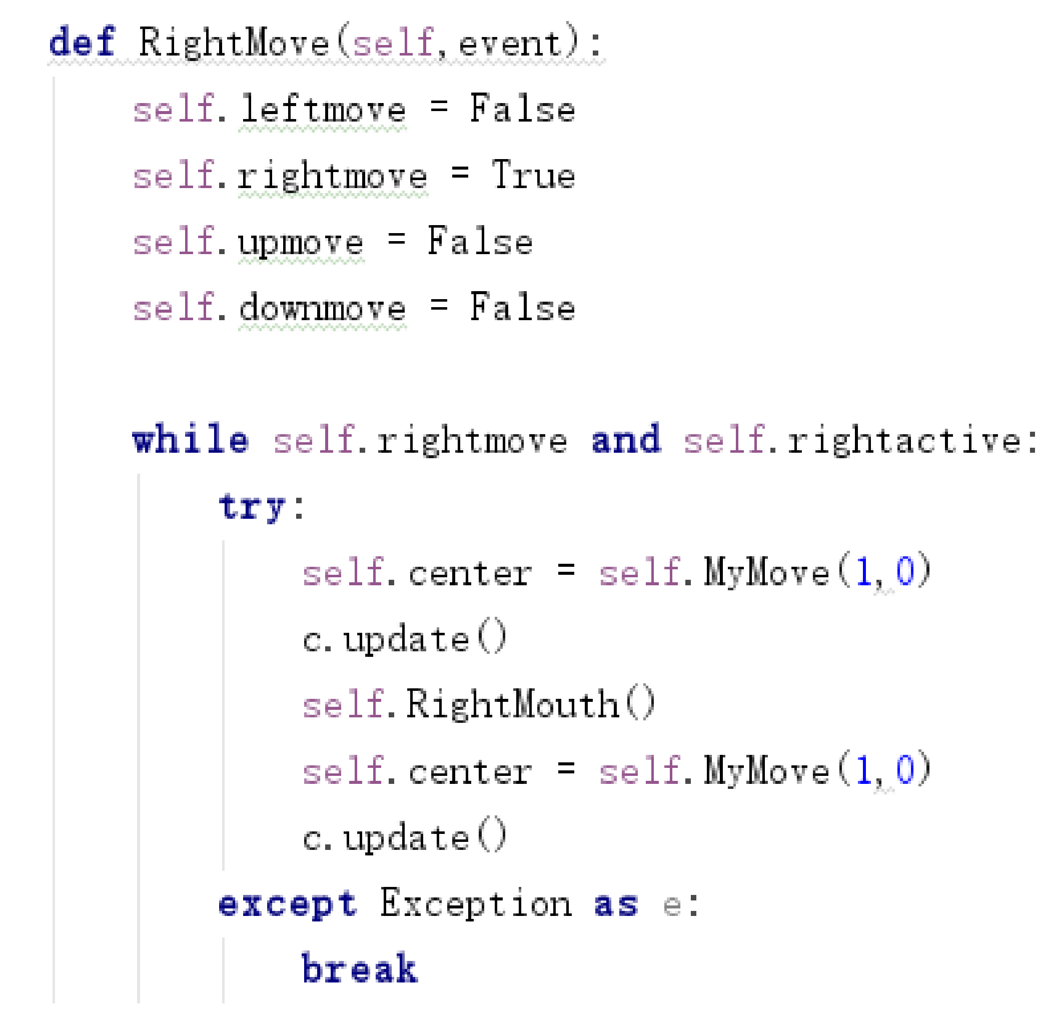
- **How to make pacman controllable**

Define the functions “UpMove”, “DownMove”, “LeftMove”, “RightMove”.

1.define four variables——”leftmove “”rightmove” “downmove “““upmove”

2.corresponding direction changes to “true” if the player input command

Approach-pacman



```
def RightMove(self,event):
    self.leftmove = False
    self.rightmove = True
    self.upmove = False
    self.downmove = False

    while self.rightmove and self.rightactive:
        try:
            self.center = self.MyMove(1,0)
            c.update()
            self.RightMouth()
            self.center = self.MyMove(1,0)
            c.update()
        except Exception as e:
            break
```

# switch the mouth to the right

Approach-enemy

- **How to set routines for ghosts?**

  Method 1

  the enemy in accordance with the provisions of the route, repeated in the route to move.

  Method 2

  the enemy move randomly

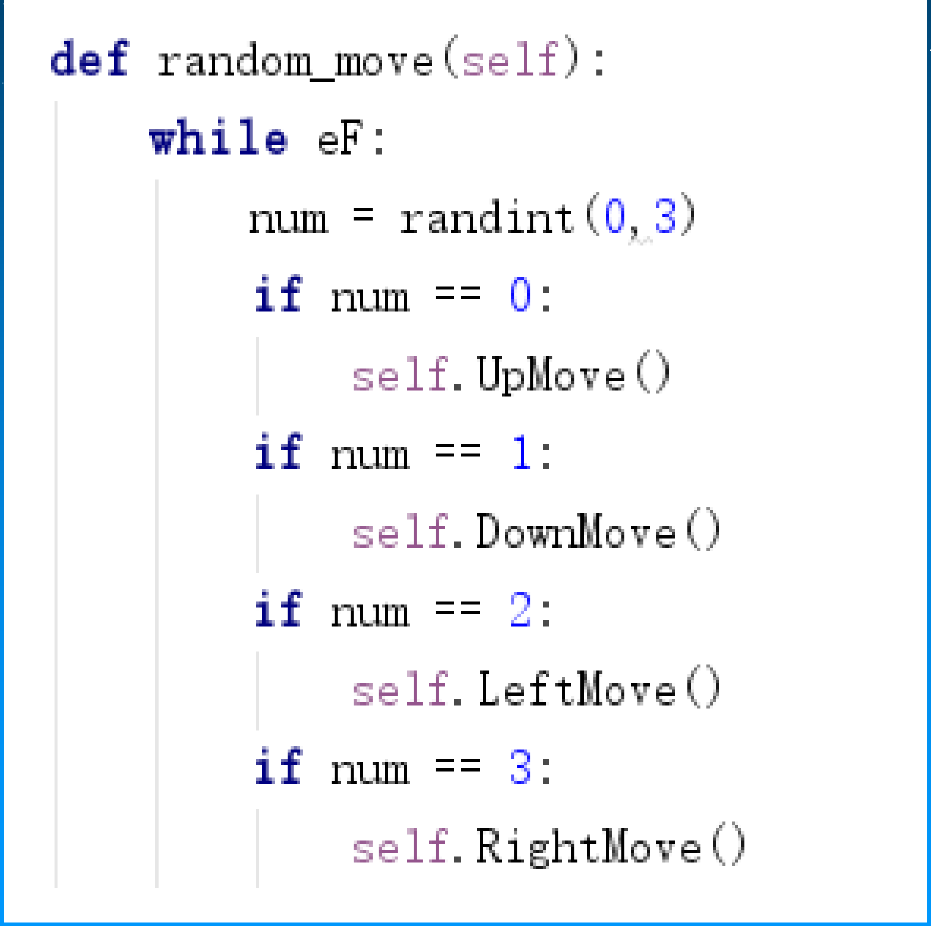
  randomly generate four numbers to represent the four directions

  Method3

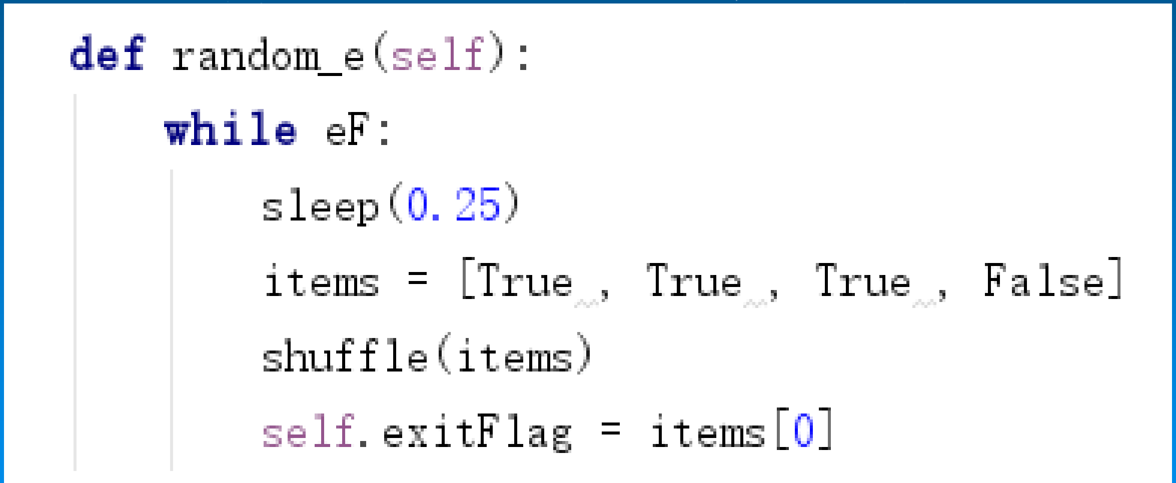
  the enemy calculate the optimal path according to the location where the “Pacman” is

Approach-enemy

Method 2



```
def random_move(self):
    while eF:
        num = randint(0,3)
        if num == 0:
            self.UpMove()
        if num == 1:
            self.DownMove()
        if num == 2:
            self.LeftMove()
        if num == 3:
            self.RightMove()
```

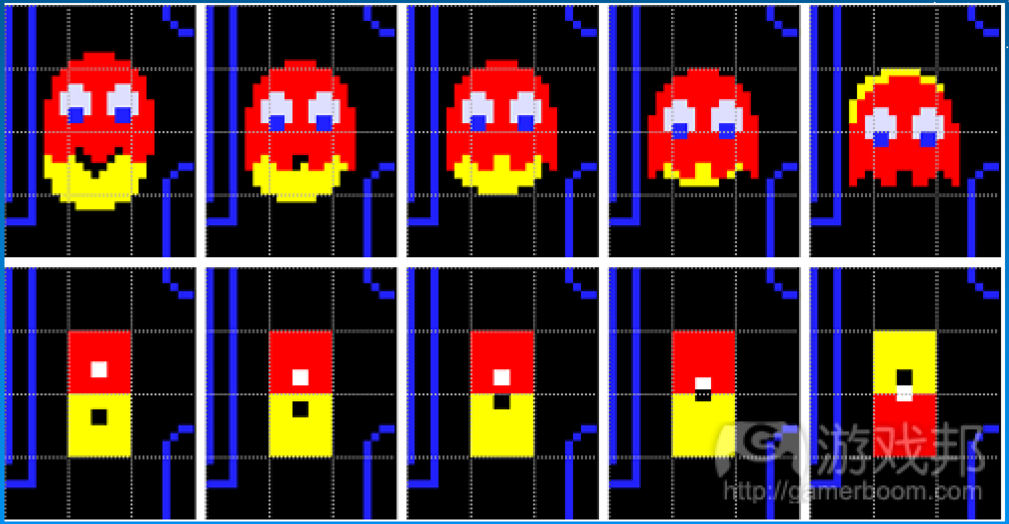


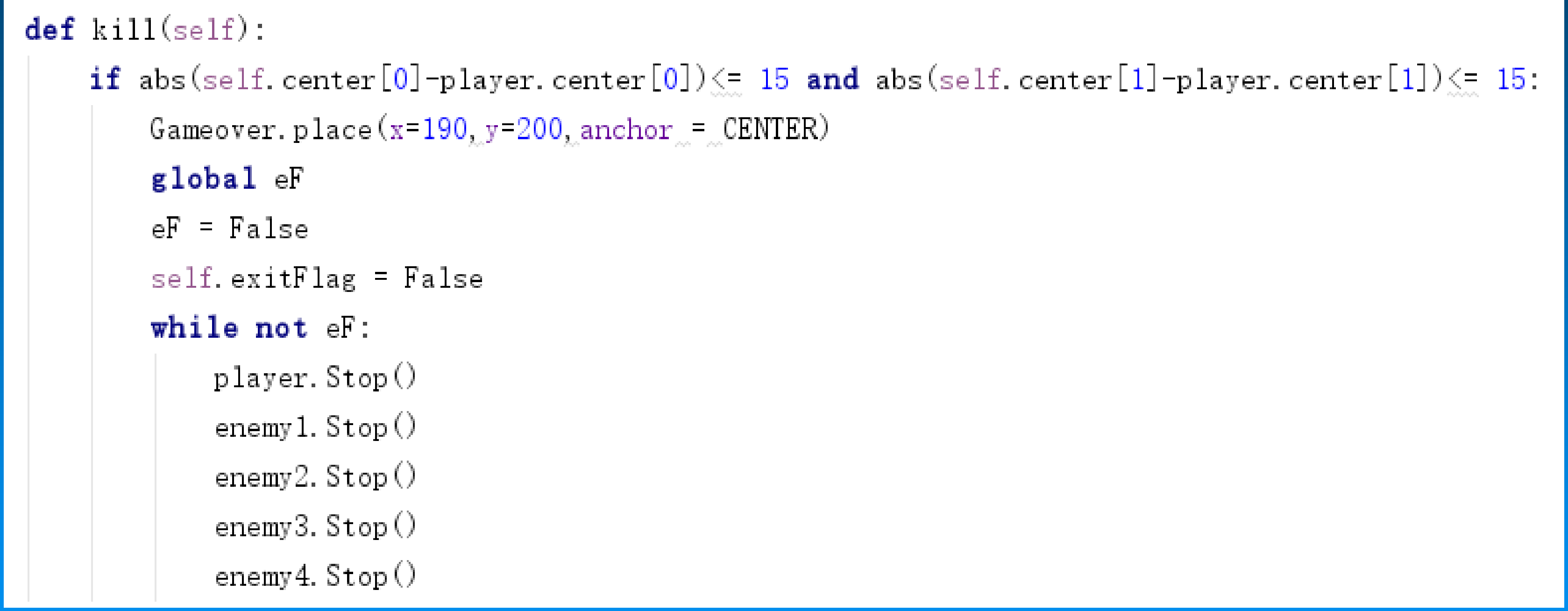
```
def random_e(self):
    while eF:
        sleep(0.25)
        items = [True , True , True , False]
        shuffle(items)
        self.exitFlag = items[0]
```

Problem remained

- **The problem of crossing the ghost**

  pacman move up and ghost moved down at the same speed





```python
def kill(self):
    if abs(self.center[0]-player.center[0])<= 15 and abs(self.center[1]-player.center[1])<= 15:
        Gameover.place(x=190,y=200,anchor = CENTER)
        global eF
        eF = False
        self.exitFlag = False
        while not eF:
            player.Stop()
            enemy1.Stop()
            enemy2.Stop()
            enemy3.Stop()
            enemy4.Stop()
```

- As game enthusiasts, we made the project not only to try to write Pac-Man in the python environment, but also to pay tribute to such a world-famous game. Throughout the process, we found a lot of problems, and finally solved them. We have a deeper understanding of how to write Python programs, and most importantly, we have lots of fun during the process.

Thank you