目录

[1. 背景介绍 3](#_Toc91653724)

[2. 需求分析 3](#_Toc91653725)

[3. 设计目的 3](#_Toc91653726)

[4. 技术简介 4](#_Toc91653727)

[4.1 开发工具 4](#_Toc91653728)

[4.2开发语言简介 4](#_Toc91653729)

[5. 设计思路 5](#_Toc91653730)

[5.1 功能分析 5](#_Toc91653731)

[5.2 模块划分 5](#_Toc91653732)

[5.3 程序流程图 5](#_Toc91653733)

[6. 项目实现及代码 7](#_Toc91653734)

[6.1 游戏主窗体的实现 7](#_Toc91653735)

[6.2 实现跳跃的小恐龙 9](#_Toc91653736)

[6.3 实现障碍物的出现 11](#_Toc91653737)

[6.4 实现碰撞与加分 14](#_Toc91653738)

[6.5 项目效果 17](#_Toc91653739)

[7. 心得总结 17](#_Toc91653740)

# 1. 背景介绍

chrome是谷歌开发的一款开源浏览器，基于谷歌Chromium浏览器开源核心研发。支持最新的网页标准和大量扩展。是和Internet Explorer(Windows 系统集成 简称IE浏览器)、Firefox并列拥有自己核心的三大浏览器之一。在使用电脑中的chrome浏览器的时候，许多用户可能都不知道chrome浏览器中自带的一个小游戏，Chrome 浏览器中的小恐龙出自谷歌的设计师 Sebastien Gabriel 之手，能够给予用户在空闲时间中放松一下。本文通过对Chrome小恐龙游戏的设计与分析，掌握Python语言在游戏开发领域的优势。

# 2. 需求分析

1.功能分析：

1. 加载背景地图，实现地图的更新
2. 初始化小恐龙图形，并实现小恐龙位置、大小、跳跃等动作状态
3. 初始化障碍物及得分图形，并实现图片的更新及分数显示

通过以上功能的实现，让使用者可以按下键盘中的空格键就能激活该恐龙,然后可以通过空格键来控制小恐龙的跳跃进行障碍物的躲避,再游戏结束后可以显示查看游戏得分。

2.非功能性需求：

本次设计目的主要为了多python基础知识的掌握和项目整体流程的学习，在项目设计上并不完善，对性能方面的要求、运行环境要求等各方面较低，只需要满足，电脑已安装python3.6及以上环境即可。

# 3. 设计目的

通过简化版Chrome小恐龙项目的完成,来提升和巩固自己的Python技术水平。通过Python基础语法的学习以及对pygame模块的学习,自行完成简化版Chrome小恐龙项目。

# 4. 技术简介

1.对python基础语法的使用，比如选择，循环，字符串,列表,元组,函数的使用等

2. python的面向对象编程技术的使用，使得代码可以复用，程序比较简洁

3.对Pygame模块的相关学习

## 4.1 开发工具

* Python 3.9
* Pycharm 2021.2(Professional Edition)

## 4.2开发语言简介

Python是一种跨平台的、开源的、免费的、解释型的高级编程语言。它具有丰富的、强大的库,能够把其他语言制作的各种模块很轻松地联结在-起，所以Python常被称为“胶水”语言。Python语 言近几年发展势头迅猛,应用领域也非常广泛,在Web编程、图形处理、黑客编程、大数据处理、网络爬虫、科学计算和游戏编程等领域都能找到Python的应用。Python语言的研究和发明主要是为了便于学习和应用,因此Python语言是开源的,并且语法非常简单、容易编写和理解。

Python是-种面向对象的动态类型语言,最初被设计用于编写自动化脚本(shell),随着版本的不断更新和语言新功能的添加，Python越来越多被用于独立的大型项目的开发。Python语言在游戏开发领域有着自己的优势,可以通过相对于其他编程语言更加人性化的语法和更加少的代码量来完成相同的任务。

# 5. 设计思路

## 5.1 功能分析

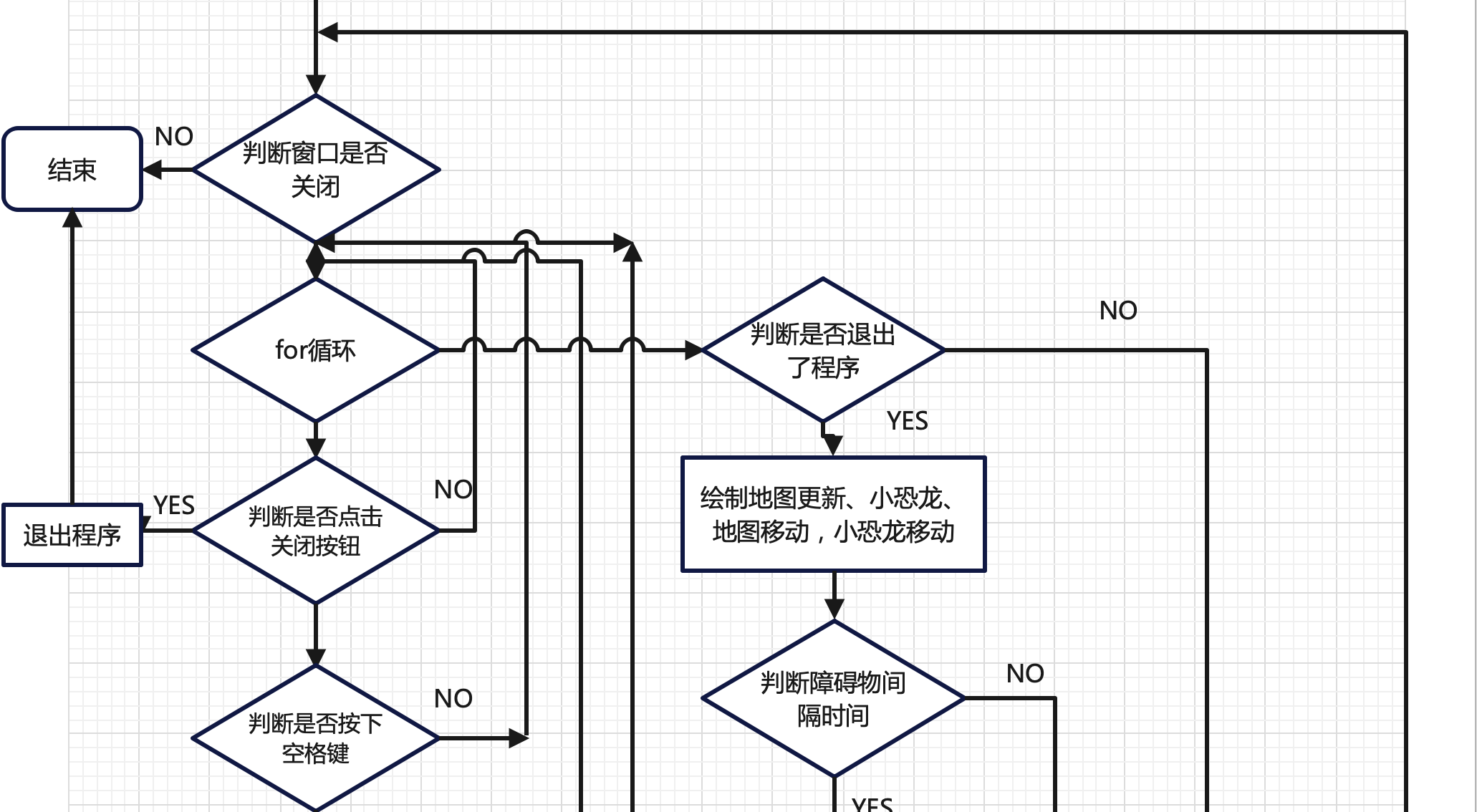
在本文设计的基于Python的游戏中会看见一只小恐龙,按下键盘中的空格键就能激活该恐龙,然后可以通过空格键来控制小恐龙的跳跃进行障碍物的躲避,并且躲避障碍物的数量进行分数的统计。

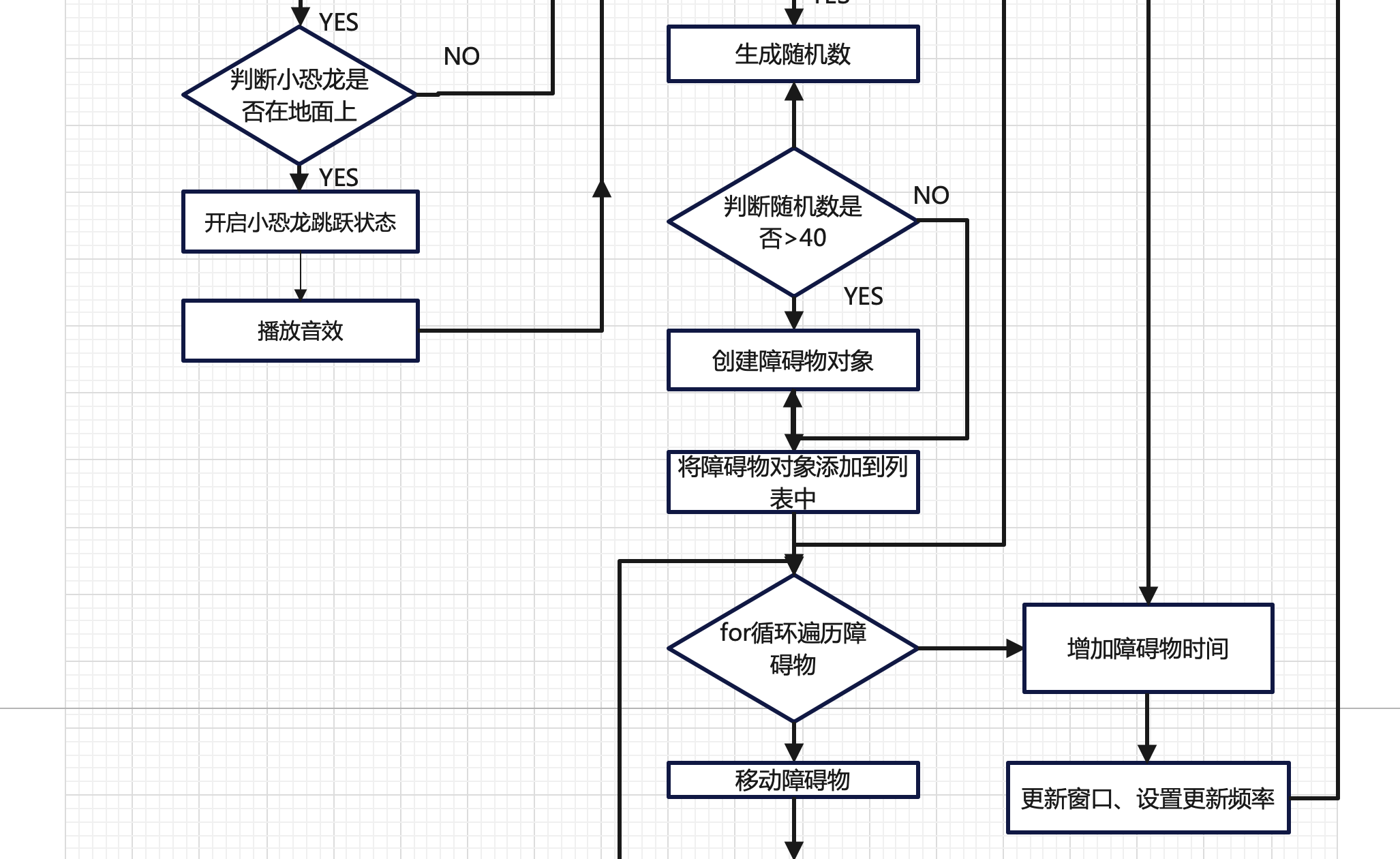
## 5.2 模块划分

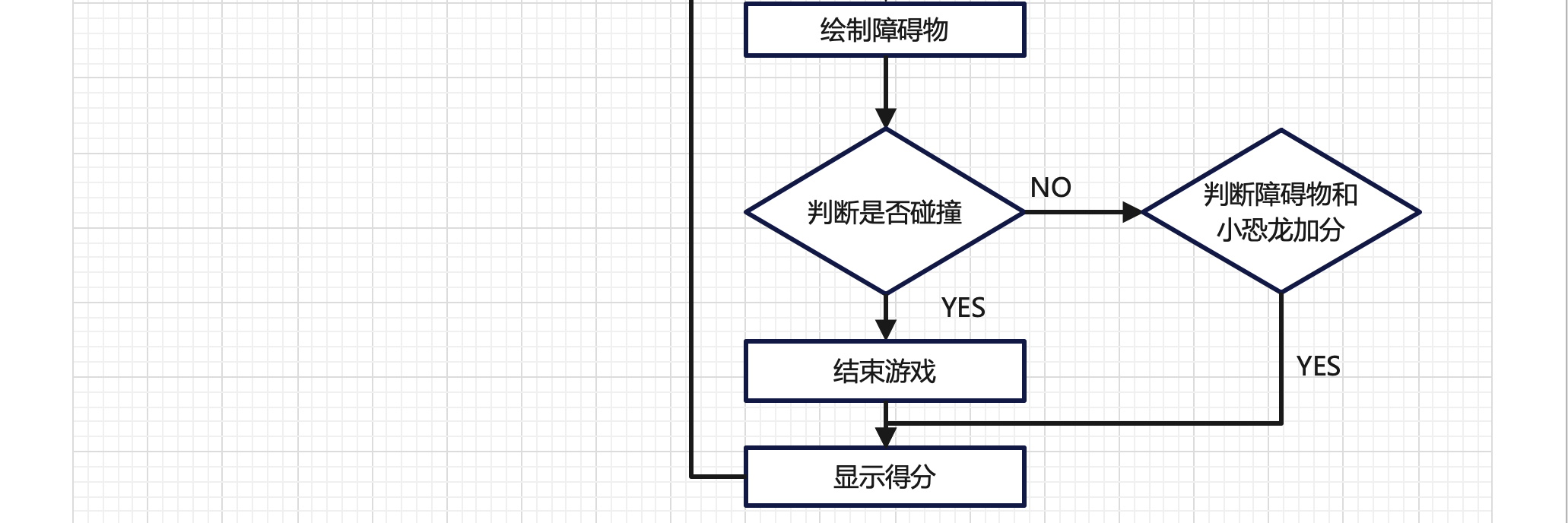
* + - 1. 在游戏主窗体下完成显示小恐龙、显示滚动地图、随机显示障碍物。
      2. 在显示小恐龙的同时编写小恐龙的跳跃功能和音效。
      3. 在b）实现同时判断游戏中移动的障碍物是否与小恐龙发生碰撞。
      4. 如果小恐龙与障碍物发生碰撞则播放碰撞音效、显示游戏结束画面、显示游戏积分。
      5. 如果小恐龙没有与障碍物发生碰撞,则只显示跳过障碍物后的积分变化。

## 5.3 程序流程图





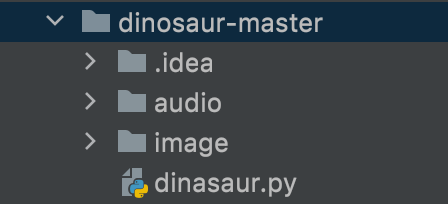




# 6. 项目实现及代码

## 6.1 游戏主窗体的实现

在Python游戏开发中,主要使用Pygame模块,这样可以使小恐龙游戏的开发变得简单很多。创建名称为dinsosaur-master的文件夹,该文件夹用于保存小恐龙游戏的项目文件,然后在该文件夹中分别创建两个文件夹, -个命名为audio,用于保存游戏中的音频文件,另一个命名为image,用于保存游戏中所使用的图片资源。最后在dinosaur文件夹中创建dinosaur.py文件,在该文件中保存实现小恐龙游戏的代码。

图6-1 项目文件结构

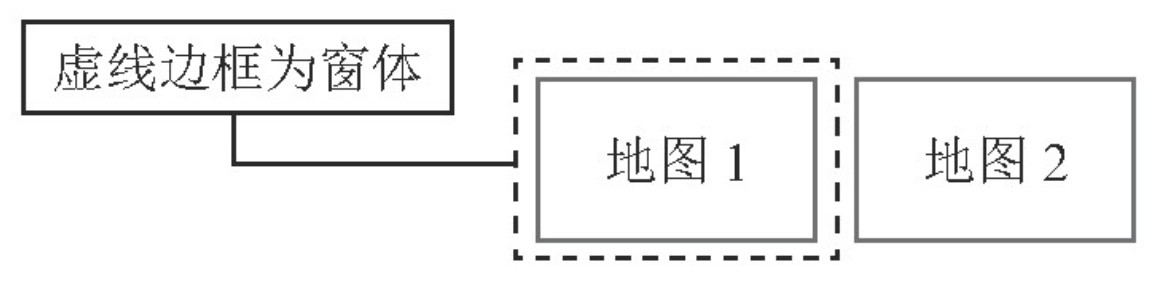
首先导入Pygame库与Pygame中的.常量库,然后定义窗体的宽度与高度。其次,创建mainGame()方法,在该方法中首先进行Pygame的初始化工作,然后创建时间对象用于更新窗体中的画面,再创建窗体实例并设置窗体的标题文字,最后通过循环实现窗体的显示与刷新,经过初始化以后就可以尽情地使用Pygame了。使用Pygame时钟之前，必须先创建Clock对象的一个实例，从而控制每个循环多长时间运行一次。 先创建一个窗体,方便与代码交互。在实现一个可以循环滚动的地图时,首先需要渲染两张地图的背景图片，然后地图1的背景图片展示在窗体中,而另一张地图2的背景图片需要在窗体的外面进行准备，如图6-2所示。

图6-2 游戏主窗体

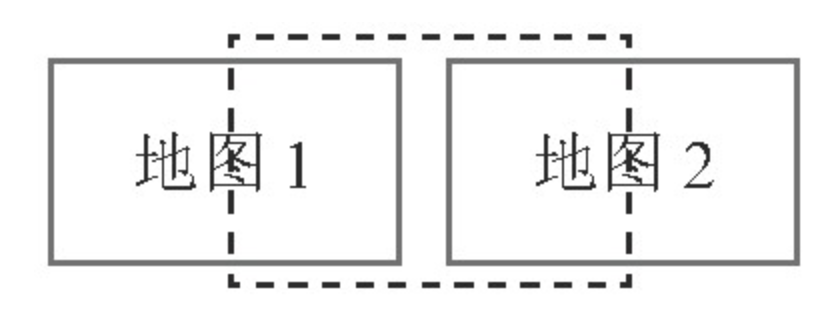
然后两张地图同时以相同的速度向左移动,此时窗体外的地图2背景图片将跟随地图1背景图片进入窗体中,这样就完成了地图的循环播放,如图6-3所示。

图6-3 游戏背景移动

当地图1完全离开窗体的时候,将该图片的坐标设置为准备状态的坐标位置。通过不断颠倒两张图片的位置,然后再进行平移,就可以在用户的视觉中形成一张不断滚动的地图。通过代码实现滚动地图的具体步骤如下:

(1)创建一个名称为MyMap的滚动地图类, 然后在该类的初始化方法中加载背景图片、定义X与Y的坐标。

(2)在MyMap类中创建名为map\_rolling()的方法,在该方法中根据地图背景图片的X坐标判断图片是否移出窗体,如果移出就给图片设置--个新的坐标点，否则按照每次5个像素的跨度向左移动。

(3)在MyMap类中创建名为map\_update () 的方法,这样就可以实现地图无限滚动的效果。

(4)在mainGame ()方法中,设置标题文字代码,在下面创建两个背景图片对象。

(5)在mainGame ()方法中循环,实现无限循环滚动的地图。

代码如下：

#定义一个地图类

class MyMap:

#加载背景图片

def \_\_init\_\_(self,x,y):

self.bg=pygame.image.load("image/sdyu1.png")

self.x=x

self.y=y

def map\_rolling(self):

if self.x<-790: #说明地图已经移动完毕

self.x=800 #给地图新坐标

else:

self.x -=5 # 移动5个像素

#更新地图

def map\_update(self):

SCREEN.blit(self.bg,(self.x,self.y))

## 6.2 实现跳跃的小恐龙

在实现小恐龙跳跃时,首先需要设置小恐龙在地图上默认的固定位置,然后判断是否按下了键盘中的空格键。如果按下了就开启小恐龙的跳跃开关,让小恐龙以5个像素的距离向.上移动。当小恐龙到达窗体的顶部边缘时,让小恐龙以5个像素的距离向下移动,回到地面后关闭跳跃的开关。具体的实现步骤如下:

(1) 导入迭代工具,创建一个名称为Dinosaur的恐龙类, 然后在该类的初始化方法中首先定义小恐龙跳跃时所需要的变量,然后加载小恐龙跑动的三张图片，最后加载小恐龙跳跃时的音效并设置小恐龙默认显示的坐标位置。

(2)在Dinosaur类中创建jump ()方法,通过该方法实现开启跳跃的开关。

(3) 在Dinosaur类中创建move () 方法,该方法先判断小恐龙的跳跃开关是否开启,再判断小恐龙是否在地面上,如果满足了，上面的两个条件,小恐龙就以6个像素的距离向上移动。当小恐龙到达窗体顶部时会以5个像素的距离向下移动,当小恐龙回到地面后关闭跳跃开关。

(4) 在Dinosaur类中创建dinosaur ()方法,在该方法中首先匹配小恐龙跑步的动图，然后进行小恐龙的绘制。

(5)在mainGame ()方法中，在创建地图对象的代码下面创建小恐龙。

(6)在mainGame ()方法的while循环中，判断关闭窗体的下面判断是否按下了空格键,如果按下了就可以开启小恐龙开关,并且播放跳跃音效。

(7)在mainGame ()方法中绘制地图代码下面实现小恐龙的移动与绘制功能。

代码如下：

#恐龙类

class Dinasaur:

def \_\_init\_\_(self):

#初始化小恐龙矩形

self.rect=pygame.Rect(0,0,0,0)

self.jumpState=False #跳跃的状态

self.jumpHeight=140 #跳跃高度

self.lowest\_y=140 #最低坐标

self.jumpValue=0 #跳跃增变量

self.dinasaurIndex=0

self.dinasaurIndexGen=cycle([0,1,2])

self.dinasaur\_image=(pygame.image.load('image/dinosaur1.png').convert\_alpha(), pygame.image.load('image/dinosaur2.png').convert\_alpha(), pygame.image.load('image/dinosaur3.png').convert\_alpha(),)

self.jump\_audio=pygame.mixer.Sound('audio/jump.wav')

#加载音效

self.rect.size=self.dinasaur\_image[0].get\_size()

#设置小恐龙矩形大小

self.x=50 #设置小恐龙的x坐标

self.y=self.lowest\_y #设置小恐龙的y坐标

self.rect.topleft=(self.x,self.y) #设置左上角为准

#跳跃

def jump(self):

self.jumpState=True

#小恐龙的移动

def move(self):

if self.jumpState: #可以起跳

if self.rect.y>=self.lowest\_y:

self.jumpValue =- 5 #以5个像素向上移动

if self.rect.y<=self.lowest\_y-self.jumpHeight:

self.jumpValue=5

self.rect.y+=self.jumpValue#通过循环改变恐龙的Y值

if self.rect.y >=self.lowest\_y: #恐龙回到地面

self.jumpState=False #关闭跳跃状态

#绘制恐龙

def draw\_dinasour(self):

#匹配恐龙动图

dinasaurindex=next(self.dinasaurIndexGen)

#实现绘制

SCREEN.blit(self.dinasaur\_image[dinasaurindex],(self.x,self.rect.y))

## 6.3 实现障碍物的出现

在实现障碍物的出现时，首先需要考虑到障碍物的大小以及障碍物的不同,如果每次出现的障碍物都是相同的,那么该游戏将失去游戏的乐趣。所以，需要加载两个大小不同的障碍物图片,然后随机抽选并显示。此外,还需要通过计算来设置多久出现-个障碍物并将其显示在主窗体中。具体的实现步骤如下:

(1) 导入随机数,创建一个名称为Obstacle的障碍物类,在该类中定义-个分数,然后在初始化方法中加载障碍物图片、分数图片以及加分音效。创建0至1的随机数数字,根据该数字抽选障碍物是石头还是仙人掌,最后根据图片的宽高创建障碍物矩形的大小，并且设置障碍物的绘制坐标。

(2) 在Obstacle类中首先创建map\_move ()方法用于实现障碍物的移动,然后创建Obstacle ()方法用于实现绘制障碍物。

(3) 在mainGame()方法中创建恐龙对象的代码，下面定义添加障碍物的时间与障碍物对象列表。

(4) 在mainGame () 方法中绘制恐龙的代码下面,计算障碍物出现的间隔时间。

(5) 在mainGame()方法中计算障碍物间隔时间代码的下面，循环遍历障碍物并进行障碍物的绘制。

(6) 在mainGame ()方法中更新整个窗体代码的上面，增加障碍物时间。

代码如下：

class Obstacle:

score=1 #分数

def \_\_init\_\_(self):

#初始化障碍物的矩形

self.rect=pygame.Rect(0,0,0,0)

#加载障碍物的图片

#加载石头

self.stone=pygame.image.load('image/stone1.png').convert\_alpha()

#加载仙人掌

self.cacti=pygame.image.load('image/cacti.png').convert\_alpha()

#创建0，1之间的随机数,0是石头，1是仙人掌

r=random.randint(0,1)

if r ==0:

self.image=self.stone

else:

self.image=self.cacti

#根据障碍物位图的宽高设置矩形

self.rect.size=self.image.get\_size()

#获取位图的宽高

self.width,self.height=self.rect.size

#障碍物绘制坐标

self.x=800

self.y=200-(self.height/2)

self.rect.center=(self.x,self.y)

#移动障碍物

def obstacle\_move(self):

self.rect.x -=5

#绘制障碍物

def draw\_obstacle(self):

SCREEN.blit(self.image,(self.rect.x,self.rect.y))

main\_Game()方法部分代码,如图6-4所示：



图6-4 main\_Game()方法部分代码实现

## 6.4 实现碰撞与加分

在实现碰撞.与积分时,首先需要判断小恐龙与障碍物的两个矩形图片是否发生了碰撞,如果发生了碰撞就证明该游戏已经结束，否则判断小恐龙是否越过了障碍物,确认跃过后就可以进行加分操作,并将分数显示在窗体顶部中间的位置了。具体步骤如下:

(1) 在Obstacle类中，Obstacle()方法的下面创建getscore()方法用于获取分数并播放加分音效,然后创建showScore()方法用于在窗体顶部中间的位置显示分数。

(2) 在mainGame()方法的上面创建game\_over()方法,在该方法中首先需要加载与播放撞击的音效,然后获取窗体的宽度与高度,最后加载游戏结束的图片并将该图片显示在窗体的中间位置。

(3) 在mainGame()方法中，绘制障碍物代码的下面判断小恐龙与障碍物是否发生碰撞,如果发生了碰撞,就开启游戏结束的开关,并调用game\_over()方法显示游戏结束的图片;否则判断小恐龙是否跳过了障碍物,跃过就进行分数的增加并显示当前得分。

(4) 为了实现在游戏结束后再按下空格键时重新启动游戏,需要在mainGame()方法中播放小恐龙跳跃音效代码的下面判断游戏结束的开关是否开启,如果开启,则重新调用mainGame()方法重新启动游戏。

代码如下：

1、Obstacle类中getScore()和showScore()方法如下：

#获取分数

def getScore(self):

self.score

tmp=self.score

if tmp==1:

self.score\_audio.play()

self.score=0

return tmp

#显示分数

def showScore(self,score):

self.scoreDigits=[int(x) for x in list(str(score))]

totalWidth=0 #要显示的数字的总宽度

for digit in self.scoreDigits:

#获取积分图片的宽度

totalWidth+=self.numbers[digit].get\_width()

#分数横向位置

xoffset=(SCREENWITDH - totalWidth)/2

for digit in self.scoreDigits:

#绘制分数

SCREEN.blit(self.numbers[digit],(xoffset,SCREENHEIGHT\*0.1))

#随着数字增加改变位置

xoffset+=self.numbers[digit].get\_width()

2、 创建game\_over()方法结束游戏

def game\_over():

bump\_audio=pygame.mixer.Sound('audio/bump.wav')

bump\_audio.play()

#获取窗口宽高

screen\_w=pygame.display.Info().current\_w

screen\_h=pygame.display.Info().current\_h

#加载游戏结束的图片

over\_img=pygame.image.load('image/gameover.png').convert\_alpha()

#绘制游戏结束的图标在窗体中间

SCREEN.blit(over\_img,((screen\_w-over\_img.get\_width())/2,(screen\_h-over\_img.get\_height())/2))

3、 （3）（4）功能在mainGame()方法的实现如图6-5、图6-6：

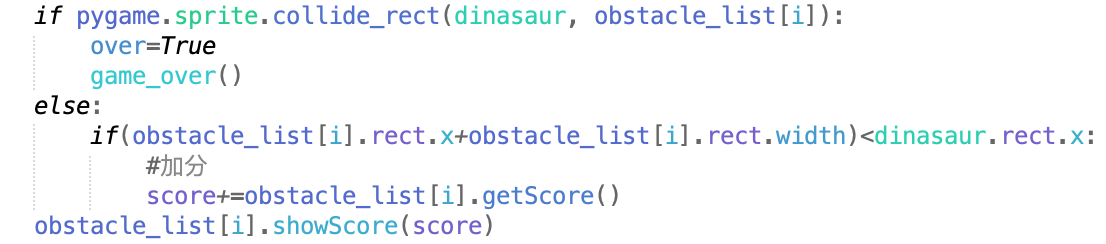


图6-5 碰撞与加分中步骤（3）具体实现



图6-6 碰撞与加分中步骤（4）具体实现

## 6.5 项目效果



图 6-5-1 游戏主窗体初始状态



图 6-5-2 正常游戏状态



图 6-5-3 碰撞后游戏结束状态

# 7. 心得总结

随着人工智能的发展,Python语言的应用变得越来越普遍。通过Python语言中的Pygame模块来设计小恐龙游戏，可以发现Python语言在开发游戏方面非常简便,相比其他语言用了更少的代码量完成了相同的功能。Python拥有 一个强大的标准库。Python语言的核心只包含数字、字符串、列表、字典、文件等常见类型和函数,而由Python标准库提供了系统管理、网络通信、文本处理、数据库接口、图形系统、XML处理等额外的功能。Python标准库命名接口清晰、文档良好,很容易学习和使用。

本次大作业收获很大，虽然技术有待提高，但可以说是用心了，我对程序编程有了新的认识，且在这个过程提高了自身的水平。通过这次大作业对计算机这门学科有了许多新的认识,活用书本的知识是不易的，程序的编写也是要经过数次的分析、思考和修正的，我利用计算机求解问题的能力得到了提升，理论知识也得以在实践中运用。