

基于 Python 平台的贪吃蛇游戏开发设计探索

张灿熙

北京市西城区黄城根小学

DOI:10.12238/acair.v1i1.6130

[摘要] 贪吃蛇游戏的设计内容分为游戏界面设计和游戏功能设计两大部分,其中游戏界面设计以满足视听觉体验为主,游戏功能设计是在满足原有的贪吃蛇游戏基本功能的基础上添加一些其他的功能。

[关键词] 贪吃蛇; 游戏; Python

中图分类号: TS952.83 **文献标识码:** A

Exploration of the Development and Design of Snake Game Based on Python Platform

Canxi Zhang

Huangchenggen Primary School in Xicheng District, Beijing

[Abstract] The design of the Snake game is mainly divided into two parts: game interface design and game function design. The game interface design is mainly designed to meet the audio-visual experience, and the game function design is to add some other functions on the basis of meeting the basic functions of the original Snake game.

[Key words] Snake; games; Python

游戏是当下社会的一种娱乐方式,近年来,各式各样的游戏如雨后春笋般拔节而出,有些游戏披荆斩棘,脱颖而出,也有些游戏精疲力竭,隐于尘埃。但不管是哪种游戏,都为人们带去了欢声笑语。在工作学习之余,身心疲惫,大多数人会选择来一款有趣的游戏来放松疲惫不堪的内心;在生活闲暇时候,百无聊赖,很多人会打开久违的游戏来打发生锈的时光。当然,对人们来说,电影、音乐、阅读、旅游这些都是不错的娱乐方式,但随着社会发展的步伐在加快,这些传统的娱乐方式已经渐渐跟不上人们的娱乐需求了,游戏理所当然地走进了人们的视野。可以这么说,游戏,已经成为了生活中的一种重要娱乐方式,它能为人们带来很多生活中得不到的乐趣和满足。

贪吃蛇游戏作为一款深受大家喜爱的经典游戏,伴随着很多人的童年和青春,是很多年轻人甚至中年人的游戏起点,发展到现在,更是受到各阶层广大人群的热衷和追捧。

1 开发语言的简介

Python是一种解释性语言,它不需要如C或C++之类的被编译成二进制代码,它可以直接从源代码运行程序。当我们运行Python文件程序的时候,文件源代码首先被Python解释器翻译成字节码,然后再交给Python字节码虚拟机来解释这些字节码,最后由Python运行输出Result。

2 概要设计

2.1 功能介绍

贪吃蛇游戏主要就是在屏幕区域内绘制出一条小蛇,并且

在与蛇不重合的屏幕区域内随机出现一个食物,游戏者通过键盘操纵游戏小蛇上下左右移动来吃到食物,当食物成功被小蛇吃到后,会在另外的区域再重新刷新出食物,而小蛇由于吃到食物,身体会增长。在游戏中,小蛇如果撞到墙壁和障碍物或者交叉撞到自己的身体,则游戏结束。

贪吃蛇游戏设计的关键点在于蛇的图形组建和蛇的运动轨迹绘制。使用一个小的矩形块(rect属性)代表蛇的身体的一部分。身体每增长一部分,添加一个新的矩形块。贪吃蛇游戏运行从蛇的头部开始移动,所以蛇不能反方向移动,也就是尾巴不能变成蛇头。如果你不按任何键,蛇就会以当前的方向向前移动。当玩家按下有效的箭头键时,蛇头按指定的方向移动,一次移动身体一部分。当玩家按下有效的箭头键时,首先要确定蛇头的位置,然后身体随着蛇头运动,实现图形就是蛇头从新的位置开始画一条蛇。蛇头按指定的方向移动,一次移动身体一部分。所以当按下有效的箭头键时,首先要确定蛇头的位置,然后身体的每一部分都随着前一部分的变化而变化,后一个覆盖前一个的位置,使贪吃蛇看起来好像是在到处移动。食物的出现和消失也是按照绘制矩形块并且覆盖矩形块来实现。

2.2 画面设计

(1) 游戏的界面构造主要包括游戏运行界面尺寸(bg_size)的定义,背景幕布(background)的覆盖,以及墙壁(wall)和障碍物(obstacle)的添加。其中墙壁是沿着界面的边缘四周依次排列,障碍物是在墙壁以内的位置随机生成。

(2)另外,在界面设计中还添加了开始界面、暂停界面和结束界面。开始界面时,界面上出现英文:操作提示和“点击任意键开始游戏”提示;暂停界面时,界面上出现“点击空格键继续游戏”的英文提示;结束界面时,会出现得分和历史记录显示,另外还添加了“重新开始”和“退出游戏”两个功能按钮。

(3)贪吃蛇的构造分为两部分,一个是蛇头,另一个是蛇的身体。蛇头是一个矩形块(rect),具有贪吃蛇的大部分功能,例如移动、变换方向、碰撞、吃食物等,游戏过程中整个界面只能存在一个蛇头;蛇的身体也是由一个小矩形块构成,与蛇头不同的是,身体的数量不限,随着贪吃蛇不断吃到食物,蛇的身体会逐渐增加,身体会随着蛇头的移动而移动,不具备蛇头所具有的功能。

(4)食物的构造也分为两部分。一部分为普通食物,会随机刷新在界面内,且不与其他物体重合,再与蛇头触碰之后会消失,并在其他区域重新随机生成;还有一部分是功能性食物,能使贪吃蛇具有某些非正常的能力。

2.3 游戏过程

(1)贪吃蛇必须从蛇的头部开始移动,所以蛇不能反方向移动,也就是尾巴不能变成蛇头。如果你不按任何键,蛇就会以当前的方向向前移动。当玩家按下有效的箭头键时,蛇头按指定的方向移动,一次移动身体一部分。所以当按下有效的箭头键时,首先要确定蛇头的位置,然后身体随着蛇头运动,实现图形就是蛇头从新的位置开始画一条蛇,此时,由于没有清屏,程序在不停地更新屏幕(flip),原来的蛇差了一个单元,看起来蛇的身体会多一部分,所以蛇的最后部分被覆盖了背景颜色,使贪吃蛇看起来好像是在到处移动。

(2)碰撞检测包含三种情况。第一种是蛇头和障碍物的碰撞,在正常情况下蛇会死亡;蛇会第二种是蛇头和身体的碰撞,正常情况下蛇也会死亡;第三种是蛇头和食物的碰撞,蛇的身体会增长一个矩形块(rect)。

2.4 游戏结束

当蛇头与四周的墙壁、随机刷新的障碍物和自己的身体发生碰撞时会死亡,这个时候,游戏进程停止,跳转到结束界面,音效停止。

2.5 添加功能

(1)添加游戏背景音乐、吃普通食物音效、生成功能性食物音效、吃功能性食物音效以及死亡音效。

(2)功能性食物区别于普通食物的是它的生成速度为20秒/个,且它的对象(Surface)图片不同,当蛇头碰撞到功能性食物的时候,蛇会获得4秒钟无敌状态的功能,在与障碍物和身体碰撞后不会触发碰撞检测,简单说,就是蛇不会死亡,游戏会继续进行下去。

(3)在游戏设计中添加了很多常规功能:

(4)开始:游戏程序运行前,界面会出现提示,点击任意键开始游戏。

(5)记录分数:游戏运行过程,吃到食物会得分,功能性食物

则会扣分,得分会记录在界面左上角。

(6)暂停/开始:在游戏过程中可以点击Space键来控制游戏的暂停和开始。

(7)结束:显示最终得分和最高纪录;会出现“Restart”和“Finish”两个功能按钮。

3 详细设计与实现

3.1 设计游戏界面

程序设计了三个界面:开始界面、主界面和结束界面,这里只给出主界面构建过程。

主界面:

#绘制屏幕函数,先将屏幕填充为bg_color(在本设计中为黑色,或者填充background,但是需要加上位置坐标参数),然后在四周边界填充墙壁。

3.2 构建贪吃蛇

构成贪吃蛇的代码主要由两部分,一部分是构建蛇头,另一部分是构建身体,移动函数,控制身体的移动路径,第一节身体刷新到蛇头的位置,后一节会刷新到前一节的位置,删除最后一个。限制是移动速度受到身体尺寸和游戏帧频的影响。

```
def move(self, xy):
    self.rect[0]=xy[0]
    self.rect[1]=xy[1]
    self.moves.append(xy)
delfself.moves[0]
```

3.3 构建食物

构建食物的代码也有两部分,分别是普通食物和功能性食物,但是这两部分代码大致相同。

3.4 碰撞检测

程序的碰撞检测一共包含了如下四个部分:

#控制食物刷新与身体不重合,如果出现重合,则食物重置(如果是食物预障碍物检测,则判断变量为food_obstacles_alive)。

```
while food_bodies_alive:
    for food in foods:
        if me.rect.colliderect(food.rect) or
pygame.sprite.spritecollide\
    (food, bodies, 0) != []:
        food.kill()
        food.Reset()
    else:
        food_bodies_alive = 0
```

4 程序的调试和测试

调试和测试程序的目的是检测和纠正潜在的错误。必须要经过仔细的调试和测试程序,以尽可能地减少错误,确保程序满足功能和性能要求,并达到原始的设计和使用要求,从而确保程序开发的质量。

Python程序的调试需要设置断点,需要首先在程序导入调试模块import pdb,然后再代码中适当的地方插入测试

代码 `pdb.set_trace()`, 最后需要执行调试代码 `python-m pdbmain.py`。

4.1 遇到的困难及解决过程

4.1.1 蛇头image的变换

贪吃蛇的蛇头需要实现上下左右四个方向移动的功能, 当贪吃蛇改变了方向之后, 其头部的图像也相应地需要发生适应其运动方向的变化, 刚开始的时候, 我采用了四张面向不同方向的蛇头图片, 当蛇头方向发生改变的时候, 利用指定函数对其 `image` 属性进行修改, 但是, 在运行的时候, 由于其不是规则的 `rect` 图形, 在进行碰撞检测的时候会有视觉上的差异, 看上去像是两者还没碰到就发生了碰撞检测, 导致游戏体验比较差。

解决过程: 将原有的四张不同方向的蛇头图片换成两张不同表情的规则图形, 并优化碰撞检测。

4.1.2 surface对象位置的刷新

在刚开始的设计中, 有很多对象的生成是采用 `random` 随机生成, 在运行过程中无法控制各个 `surface` 对象的位置, 导致有不同的 `surface` 对象出现重合的情况, 使游戏无法正常进行下去。

解决过程: 对某些对象的刷新限定区域, 对象与对象之间进行碰撞检测来限制刷新重合, 对数量不单一的对象内部进行互相碰撞检测。

4.1.3 身体的移动

对于贪吃蛇身体的移动刚开始是一筹莫展的, 因为贪吃蛇的蛇头和身体是不同的对象, 要想让他们连接起来如同一个整体来移动其实并不是那么容易。起初我对蛇头和身体分别进行部分移动设计, 然后再把他们两部分通过对速度和位置的强行限定来连接在一起, 让后一部分完全模仿前一部分运动, 但是这种设计最终并没有很好的实现。由于在对键盘事件的捕捉和设定中, 只对蛇头进行了事件匹配, 当蛇头转弯的时候身体会按照原有的方向移动, 修改之后, 却又出现身体与蛇头同时转弯的情况, 没有实现身体随着蛇头一部分一部分转向的功能。

解决过程: 将蛇头和身体各部分看成一个整体进行设计, 建立一个数组 `moves[]`, 将蛇头和身体各部分的移动坐标全部放进去, 利用迭代的算法, 让第一个对象移动一个速度(固定的坐标值, 大小等于每个部分的直径), 再让后面的对象依次相应地

覆盖前一个对象的坐标, 通过界面的不断刷新, 就能形成各部分依次向前移动的视觉效果。

4.1.4 碰撞检测的实现

本程序中涉及到的碰撞检测比较多, 刚开始的时候对 Python 函数库不熟悉, 所以自己编写碰撞检测函数。首先给参与检测的两个对象分别设置圆心值坐标 `center` 和半径值 `radius`, 利用数学方法中根据坐标求两点间距离的公式, 比较两个圆心之间的距离与半径之和的大小, 来判断两个对象是否发生重合(碰撞), 具体设计如图7-1所示。但是这样设计出来的方法只能满足规则的圆形对象之间的碰撞, 对待其他形状的对象之间的碰撞检测时, 很难得到准确的计算, 且这个方法对待数量单一的对象之间的碰撞检测可以大致实现, 但是对待数组对象之间的检测就无能为力了。

解决过程: 利用 Python 函数库中的 `collidect()` 函数(精灵图形的 `rect` 属性碰撞函数)和 `spritecollide()` 函数(参数为一个精灵 `sprite` 和一个精灵数组 `group`, 返回一个列表, 记录数组中与 `sprite` 发生碰撞的精灵), 所有的对象之间的碰撞问题都解决了。

5 结论

目前, 使用 Python 软件开发游戏是主流, 但对于有着高性能、高安全性要求的游戏来说, 硬件平台至关重要并起着决定作用。未来游戏平台将向“高速、低成本、安全”的方向发展, Python 平台无疑满足这些条件, 是首选平台之一。

本文提出的“贪吃蛇”游戏完全基于 Python 平台, 使用 HDL 实现, 并使用标准键盘和 VGA 显示器作为输入和显示设备, 是在 Python 平台上开发游戏的有益的尝试。基于 Python 平台的游戏开发, 速度快、可拓展性好, 在同一个 Python 硬件平台上, 可以开发多款独立的游戏, 游戏的速度快, 且游戏间互不影响。

[参考文献]

[1] 吕橙. 贪吃蛇游戏的开发设计[J]. 电脑知识与技术, 2021, 17(03):118-121.

[2] 李倩. 贪吃蛇游戏设计[J]. 无线互联科技, 2019, 16(2):162-163.

[3] 刘港毓, 龙波, 欧阳朝煌, 等. 浅析《贪吃蛇》游戏设计与制作[J]. 科技风, 2017, (05):68.