

Python 大作业实习报告

《WALL·E 历险记》










作者：董欣然

WALL·E历险记

游戏简述：小球WALL·E在太空中游走，他会收获途经的所有宝物，不同的宝物功能不同，WALL·E进入禁区会失去一点生命值，WALL·E想要得到尽量高的积分，快来帮助他吧(*^▽^*)

小组成员：董欣然 邹远鑫 祝希尧



<p>WALL·E，在太空中游走的小可爱，遇到平板会反弹</p> 		<p>反弹板 游戏中反弹板由三个组件构成，鼠标控制反弹板位置，键盘上下左右键控制两端反弹板角度</p> 
<p>红宝石 增加一点生命值</p> 		<p>旋转的平板 可反弹</p> 
<p>移动的毛毛虫 碰到失去一点生命值</p> 		<p>钥匙 引发机关，可能出现的情况有：反弹板增长、缩短，球速增快，减慢，爆出新的道具，增减球</p> 
<p>生命值 爱心的数量体现实时生命值</p> 		<p>普通宝物 WALL·E碰到宝物会获得积分</p> 

<h3>WALL·E历险记</h3> <p>游戏简述：小球WALL·E在太空中游走，他会收获途经的所有宝物，不同的宝物功能不同，WALL·E进入禁区会失去一点生命值，WALL·E想要得到尽量高的积分，快来帮助他吧(*^▽^*)</p>		<p>小组成员：董欣然 邹远鑫 祝希尧</p>	<p>Python大作业 多媒体软件</p> 
<p>WALL·E，在太空中游走的小可爱，遇到平板会反弹</p> 	<p>触碰钥匙引发机关： 1.反弹板伸缩 2.球速增减 3.爆出新的道具 4. WALL·E的小伙伴出现/消失</p> 		<p>第一关： 反弹板在底部，禁区在底部以下 得分策略：将反弹板调成碗状，根据宝物的位置调整侧板的角度</p> 
<p>红宝石，增加一点生命值</p> 	<p>反弹板</p> 		
<p>旋转的平板，可反弹</p> 	<p>游戏中反弹板由三个平板构成，鼠标控制反弹板位置，键盘上下左右键控制两端反弹板角度</p>		
<p>移动的毛毛虫，碰到失去一点生命值</p> 	<p>普通宝物，WALL·E碰到宝物会获得积分</p> 		

摘要:

WALL·E 的设计以弹珠为原型，通过游戏角色在界面上的游走解锁不同的道具，获得积分和惩罚。在游戏基本框架上我们加入了许多创新元素，比如有三种模式可选；反弹板设计为 H 形，且侧面的两个板可旋转；加入了生命值设计；平面中加入自动旋转的平板；小球碰到特殊道具“钥匙”会随机触发一些特殊效果。

一、选题及创意介绍

基于 Pgzzero 库支持的功能，我们小组设计了一款弹球类型的小游戏，在积分制的基础上加入了许多创新元素，比如有三种模式可选，三种模式难度逐渐增大，模式一反弹板设在底部，可在底部水平移动，小球落到最底部损失生命值；模式二中地图分为左右两部分，反弹板设在中间分界处，可沿中间分界线移动，小球越过分界损失生命值；模式三为模式二进阶，地图设为可旋转（此时分界线也会一起旋转）；反弹板设计为 H 形，且侧面的两个板可旋转；加入了生命值设计，小球碰到毛毛虫会损失生命值（其中毛毛虫可移动），碰到红宝石会增加生命值；平面中加入自动旋转的平板，可反弹小球；小球碰到特殊道具“钥匙”会随机触发一些特殊效果，如反弹板增长、缩短，小球速度增加、减慢等。

二、设计方案

此程序有控制分支模块、游戏界面模块、绘图模块、游戏运行模块、音乐和音效模块以及玩家操作模块。

控制分支模块的功能是控制游戏的运行，决定运行代码中的哪一部分，比如是控制进入游戏说明界面还是开始游戏的运行，游戏结束后重新开始还是返回主菜单等。

游戏界面模块由五个界面组成，分别是主菜单界面、游戏说明界面、选择关卡界面、游戏运行界面、游戏结束界面。此模块通过控制分支模块和玩家操纵模块来控制响应，通过绘图模块将界面展示出来。

绘图模块即 draw 函数和它的几个子函数，每个界面都对应一个函数来绘制，通过控制分支模块来决定进行哪个界面的绘制。此模块将游戏界面展示出来，是与玩家最直接的交互方式。

游戏运行模块即在游戏运行过程中发挥作用，此模块包括增加宝物、增加（减少）小球、小球反弹、生命值增加减少等多种功能，接收玩家操纵模块传来的玩家操作，进行响应，并通过绘图模块展示每一次的运行结果。

音乐与音效模块即播放背景音乐以及游戏运行中产生的音效，背景音乐为全程播放，音效在小球碰撞、吃到钥匙产生特殊效果、生命值增加或减少、游戏结束时播放。

玩家操纵模块即通过鼠标和键盘接受玩家的输入，并将输入传递给控制分支模块和游戏运行模块。

三、实现方案及代码分析

（1）实现方案

控制分支模块主要是通过 flag 变量进行控制，flag 取不同的值游戏将进入不同的界面和模块，在其他模块中皆有关于 flag 值的分支，以此来进行游戏的

控制。

游戏界面模块主要通过制图软件来绘制一些文字图片，调用每个 Actor 对象的 draw 函数来绘制出来。

绘图模块即通过 Actor 对象的 draw 函数堆叠来实现。

游戏运行模块通过一系列函数来实现，具体函数解释在代码分析部分呈现。

音乐与音效模块通过 music 和 sounds 两个轨道实现，在 music 和 sounds 文件夹中放入想要播放的音乐文件，通过 music.play(filename) 函数以及 sounds.filename.play() 函数进行播放。

玩家操纵模块即接收鼠标和键盘的输入，通过 on_mouse_down 来接收鼠标的点击，通过 Actor 的 collidepoint 函数进行处理；通过 on_mouse_move 函数来接收鼠标移动的信息，通过计算确定反弹板应该移动到的位置；通过 on_key_down 函数接收键盘输入的信息，实现对反弹板旋转的操纵。

(2) 代码分析（从前到后，只分析主要函数）

flag 用于控制分支模块，用于控制进入哪个界面和模块，flag 为 1 进入主菜单界面，flag 为 2 进入选择关卡界面，flag 为 3 进入游戏运行界面，flag 为 4 进入游戏结束界面，flag 为 5 进入游戏说明界面。

task 用于控制进入三个模式中的哪一个。

start 函数用于游戏运行模块开始时初始化一些数据，此函数在进入游戏运行模块时调用，在选择关卡完成时运行。

select 函数在选择关卡之后调用，功能是选择背景图以及设置背景图的角度，其中模式一用 background1，模式二和模式三用 background3，且角度设为 30 度。

Judge 函数用于判断球在分界线哪一侧，在模式二和模式三中用到（因为越过分界会损失生命值），判断方式是根据向量外积的正负。

Addball, Addbox, Add_box_canes 为增加球、道具、板的函数。

draw_start, draw_select, draw_play, draw_over, draw_instructions 为绘制五个界面所用的函数，通过 Actor 的 draw 函数实现。

draw 即是在 flag 取不同值时，调用上述五个函数，绘制界面。

on_mouse_down 函数用于接收鼠标点击信息，此块信息主要用于控制分支模块。on_mouse_move 函数用于感知鼠标移动，用在游戏运行时，由鼠标位置向分界线作垂线，垂足位置即为反弹板移动到的位置，实现方式是几何和计算。

on_key_down 函数用于感知键盘信息，通过方向键操纵 H 形反弹板两侧板的角速度，上下键用于控制右侧板，左右键用于控制左侧板。

touch_wall 为撞到地图边界反弹的函数，反弹规则为平行边界方向速度不变，垂直边界速度相反。

key_rand 为吃到钥匙后触发随机效果，通过产生一个随机数，根据这个数决定产生什么效果。

touch_box 在小球吃到道具时调用，遇到红宝石增加生命值，遇到毛毛虫损失生命值，遇到钥匙触发随机效果。

touch_cane 的功能是撞到板后反弹，rotate_cane 为板的自动旋转函数。

Judge2 用于判断球是否碰到反弹板，判断方式为根据板和球的中心位置 pos 以及大小进行计算。若球碰到板，则调用 touch_board 函数使球反弹。

update 函数每次更新当前数据，其内容主要有球的移动，板的旋转，毛毛虫的移动，背景图的旋转，以及小球是否吃到道具和小球的反弹。生命值为 0 时，播放 gameover 音效，进入游戏结束界面。

四、后续工作展望

在游戏进行界面中增加返回主界面按钮，即可以中途退出。

增加游戏难度选择部分，玩家可根据自身能力选择难度，其中难度可通过刷宝物的概率以及球的速度，板变长、变短的概率进行调整。

增加闯关模式，将不同游戏模式和难度设置一个次序，通过当前关卡方可进入下一关。

增加暂停和存档功能，考虑到低难度可能会进行很长时间，玩家中途退出而又不想放弃本局游戏时可以选择暂停或存档。

增加排行榜功能，记录玩家每次游戏的得分，进行排行。

五、小组分工合作

游戏设计：共同完成

功能实现及程序编写：

邹远鑫：

董欣然：

祝希尧：

程序修缮及 debug：共同完成