



小猪快跑

Emmmm组

- 徐浩展 曹晨旭 黄宝萱



Contents

- 1 关于游戏
- 2 灵感来源 & 我们的设计与优化
- 3 成果展示



关于我们的小游戏



“小猪快跑“是一个简化版的障碍类游戏，一只可可爱爱的小猪要想闯关成功，需要跳过前进路上所有的云朵。



设计灵感

01

往年的优秀Team Project

02

谷歌浏览器断网时的等待小游戏





我们的设计与优化

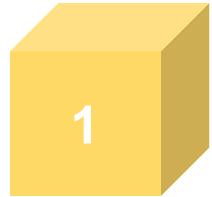
01

主要功能与设计

02

音乐的加入

主要功能与设计



小猪对象的创建和跑动的实现（70, 95到105行）

运用delay变量控制两幅小猪图片更迭的时间间隔，并同时有规律地控制switch_image变量的值以达到切换图片的目的。



云朵对象的创建和运动的实现（38到43, 71, 72, 90到94行）

首先将音乐中的节奏点记录在文本文件timelist.list中，然后用读取其中的内容构成列表，并运用这一数据通过list comprehension 的方法根据时间批量创建对象，最后利用循环遍历所有对象，使其向左运动。



指令输入部分（109到145行）

本程序一共包含八个可输入的指令，分别是按下Escape(退出程序)、鼠标点击（使程序暂停）、鼠标点击（使程序从暂停中恢复）、按下空格或方向键上键（跳跃和二级跳）、按方向键下键（快速下落）、按下e键（开始游戏）、按下r键（重新开始），具体实现过程在下文中将会详细解释。

4

跳跃及二级跳动作 (129到133, 147到159行)

当跳跃指令发出时，跳跃动作的指示变量jump的值更新为True，此时判断小猪是否达到最高点，若否，则依据设定好的速度变量speed使小猪向上移动，若是，则speed改变方向，令小猪向下移动，同时判断小猪是否抵达地面，若是则将jump更改为False，表示不处于跳跃状态，并将速度方向重置，特别的，为了防止小猪因为速度和高度无法整除等原因进入地面以下使程序出错，故在每一次落地时将小猪的位置重置回原位。二级跳指令的实现是在原有跳跃程序的基础上增加jumpnum变量以记录跳跃次数，使得程序允许一次额外的跳跃发生在空中的下落阶段，即在此阶段中改变一次速度方向，再实现部分跳跃过程。

5

碰撞与重启功能 (10到29, 134, 135, 173到191行)

由于云朵的面积比小猪小，故选取云朵的四个顶点作为比较点，当四个点中的任意一点碰到小猪时，is_c变量的值为True意为碰撞发生，此时根据游戏规则，游戏结束故所有变化速度归零，并显示Press R to restart。此时若重启指令下达，则运行reset () 函数，其功能是将所有变量重置为初始值，使游戏重新开始并暂停游戏以等待玩家准备好后再开始。

6

暂停与开始功能 (192到202, 119到128行)

一开始，当程序被启动时，游戏默认处于暂停状态，`is_stop`等于True，意为目前正处于暂停中，`is_start`等于False意为游戏未开始，此时显示Press E to start，当开始指令下达后，暂停状态解除，`is_stop`变为False，`is_start`变为True，游戏开始运行。与此同理，点击鼠标时也会进行类似操作以转换暂停状态。当`is_stop`为True时，`speed`(小猪速度)和`speedy` (云朵速度) 都变为零以使画面暂停，当`is_stop`变量变化为False时，会将`speed`和`speedy`重置回原值以使游戏继续。

7

音乐的播放和暂停 (25, 33到35, 128, 143, 191, 196行)

先装载背景音乐，当游戏开始时开始播放，暂停时音乐暂停，暂停恢复时音乐跟着恢复，碰撞时音乐停止，重启时将音乐再次从头开始播放（但因为重启后为暂停状态所以此时音乐也处于暂停状态）。

时间与分数

(62到68, 87, 121, 126, 127, 140, 141, 161到172, 179到183行)

当游戏开始时，用time1记录开始时间，并将time2, time3, time4, pretime,endtime,pausetime都设置成初始值，在每个游戏循环中，用time2记录当前时间，当暂停开始时，用time3记录暂停开始时间，当暂停解除时，用time4记录解除时间，并将time4与time3的差记录为pausetime即暂停时间，当碰撞发生时，用time3记录碰撞时间（此处将碰撞理解为一种不可恢复的暂停），当暂停发生时，游戏总时间为暂停开始时间（time3）减开始时间（time1）减之前暂停的时间（pausetime），同时，该时间也将被记为pretime即在该暂停发生前的总时间；当暂停未发生时，总时间等于pretime加当前时间（time2）减最近一次暂停恢复的时间（time4）；当碰撞发生，游戏结束时，碰撞时间（time3）即为结束时间（endtime），为使此赋值只发生一次，故需要判断条件if endtime==None，此时总时间的计算方法是结束时间（endtime）减开始时间(time1)减暂停的时间(pausetime)。游戏的分数由时间得出，计算公式为：分数= (时间*10) **0.5*100，保留整数，最后，将分数以文本形式输出在画面中。

音乐的加入

1

根据音乐节奏设置障碍，使游戏的难度上升

2

糅合音乐元素，游戏趣味增加



成果展示





谢谢观看