

游戏：小猪快跑

组名：Emmmm

成员：徐浩展 黄宝宣 曹晨旭

指导老师：鲍扬

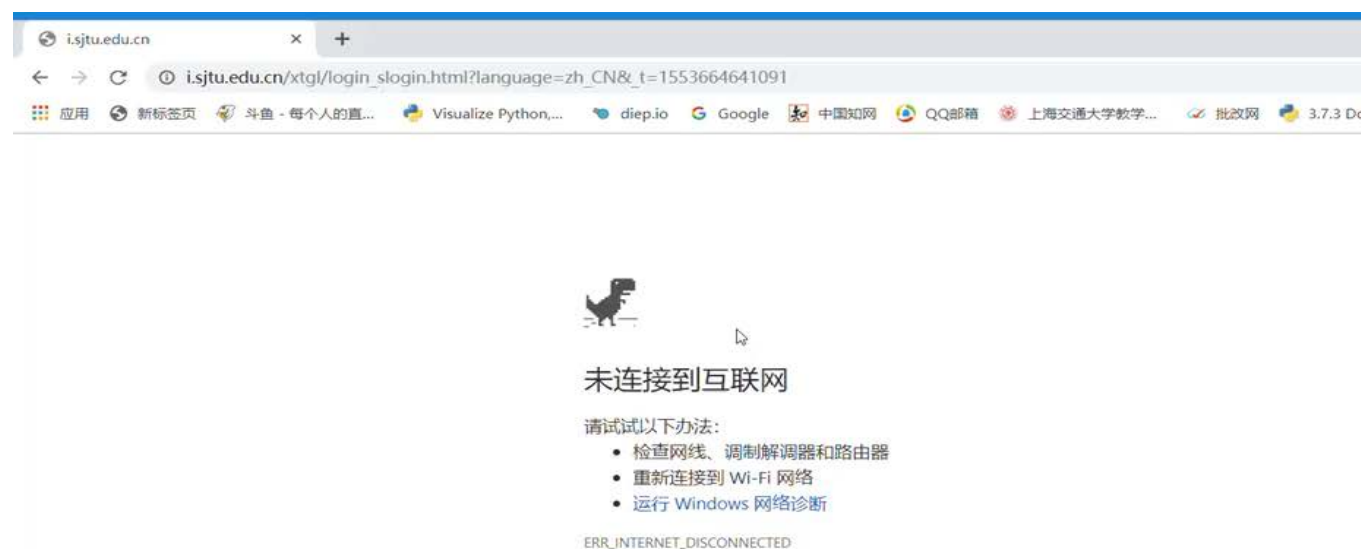
第一部分：背景介绍和整体思路

游戏介绍

我们想设计一种有音乐元素的横板 2D 像素风小游戏，主要内容为玩家使用键盘控制游戏主角（一只猪）在向前奔跑的同时躲避根据音乐节奏出现的障碍物（云朵）。随着程序运行时间的增加，小猪向前的速度会不断增加，当小猪碰到障碍物时便会死亡，游戏结束，系统记录玩家生存的时间，并根据一定比例转化为该玩家的得分。

灵感来源

我们的灵感来源于往年优秀的团队作品、老师介绍的 Flappy-bird 和谷歌浏览器断网时的等待小游戏。



设计方法

根据对 pygame 的学习，我们将游戏设计分为两个部分。

1. 游戏主体的设计：包括小猪对象的创建和跑动的实现、云朵对象的创建和运动的实现、跳跃、碰撞与重启功能、暂停与开始功能等
2. 游戏音乐的添加和节奏设计，根据音乐节奏设置障碍，使游戏的难度上升并增加游戏趣味性。

具体操作：

1. 寻找或制作障碍物的形象和动态效果的素材。
2. 选定若干首合适的歌曲并分析记录其节奏。
3. 编写猪的一系列动作（跑动、跳、碰撞、死亡等）的代码。
4. 编写障碍物代码并手动将障碍物根据音乐节奏加入到猪的行进路线上
5. 编写开始和结束界面的代码并将音乐于对应的行进路线相匹配。
6. 对几部分代码进行整合优化。
7. 测试及最终完善。

第二部分程序具体内容：

2.1 插入库、常量等

在制作本程序的过程中，我们决定使用 pygame 作为主要制作工具，与此同时，为了完成时间相关内容，故插入 time 库。

```
7 #导入
8 import pygame, sys, time
9 from pygame.locals import K_ESCAPE, MOUSEMOTION, MOUSEBUTTONUP, KEYDOWN, K_SPACE, K_r, K_e, K_DOWN, K_UP
```

2.2 初始化

将 surface 与背景音乐初始化。

```
31 #初始化
32 pygame.init()
33 pygame.mixer.init()
```

2.3 小猪对象的创建和跑动的实现

```
#创建物体
x=pygame.Rect(0,207,80,68)

95     #小猪跑动
96     if not (delay%5):
97         switch_image = not switch_image
98     delay -=1
99     if not delay:
100         delay = 100
101     if switch_image:
102         ball = pygame.image.load("3.png").convert_alpha()
103     else:
104         ball = pygame.image.load("4.png").convert_alpha()
105     pig.blit(ball,x)
```

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。

2.4 云朵对象的创建和运动的实现

```

38 # 节奏
39 timelist=[]
40 ti=open("timelist.txt","r")
41 for i in ti:
42     timelist.append(float(i))
43 ti.close()

71 clouds=[pygame.Rect(375+60*6*(t-0.98),220,40,32) for t in timelist]
72 clouds_save=list(clouds)

90     # 云朵
91     cloud = pygame.image.load("cloud.png")
92     for y in clouds:
93         pig.blit(cloud,y)
94         y.left=y.left+speedy[0]

```

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。

2.5 指令输入部分（109 到 145 行）

```

109 #指令输入
110 for event in pygame.event.get():
111     if event.type == pygame.QUIT:
112         sys.exit()
113     elif event.type==KEYDOWN and event.key==K_ESCAPE:
114         pygame.quit()
115         sys.exit()
116         exit
117     elif event.type==MOUSEMOTION:
118         mousex,mousey=event.pos
119     elif event.type==MOUSEBUTTONDOWN and speed!=[0,0] and speedy!=0 and is_c==False :
120         is_stop=True
121         time3=time.time()
122     elif event.type==MOUSEBUTTONDOWN and speed==[0,0] and speedy==[0,0] and is_c==False and is_start==True and is_stop==True:
123         speed=s
124         speedy=sy
125         is_stop=False
126         time4=time.time()
127         pausetime=pausetime+time4-time3
128         pygame.mixer.music.unpause()
129     elif event.type==KEYDOWN and (event.key==K_SPACE or event.key==K_UP):
130         jump=True
131         jumpnum=jumpnum+1
132         if speed[1]==5 and jumpnum==2:
133             speed[1]=-5
134     elif event.type==KEYDOWN and event.key==K_r:
135         x,clouds,jump,is_stop,is_c,speed,speedy,time1,time4,time3,pretime,endtime,pausetime,is_start,jumpnum,s,sy=reset()
136     elif event.type==KEYDOWN and event.key==K_e and is_start==False:
137         speed=s
138         speedy=sy
139         is_stop=False
140         time4=time.time()
141         pausetime=pausetime+time4-time3
142         is_start=True
143         pygame.mixer.music.play()
144     elif event.type==KEYDOWN and event.key==K_DOWN and 145<x.bottom<275 and jump==True:
145         speed[1]=15

```

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。

2.6 跳跃及二级跳动作

```
129         elif event.type==KEYDOWN and (event.key==K_SPACE or event.key==K_UP):
130             jump=True
131             jumpnum=jumpnum+1
132             if speed[1]==5 and jumpnum==2:
133                 speed[1]=-5
147         #跳跃动作
148         if jump==True:
149             if x.bottom<=275:
150                 if x.bottom<=145:
151                     speed[1]=-speed[1]
152             if x.bottom>275:
153                 jump=False
154                 speed[1]=-speed[1]
155                 jumpnum=0
156                 speed=[0,-5]
157                 s=[0,-5]
158                 x.bottom=275
159             x=x.move(speed)
```

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。

2.7 碰撞与重启功能

```
10 def reset():
11     x=pygame.Rect(0,207,80,68)
12     clouds=[pygame.Rect(375+6*60*(t-0.98),220,40,32) for t in timelist]
13     jump=False
14     is_stop=True
15     is_c=False
16     speed=[0,-5]
17     speedy=[-6,0]
18     time1=time.time()
19     time4=time1
20     time3=time1
21     pretime=0
22     endtime=None
23     pausetime=0
24     is_start=True
25     pygame.mixer.music.play(-1,0.0)
26     jumpnum=0
27     s=[0,-5]
28     sy=[-6,0]
29     return x,clouds,jump,is_stop,is_c,speed,speedy,time1,time4,time3,pretime,endtime,pausetime,is_start,jumpnum,s,sy

134         elif event.type==KEYDOWN and event.key==K_r:
135             x,clouds,jump,is_stop,is_c,speed,speedy,time1,time4,time3,pretime,endtime,pausetime,is_start,jumpnum,s,sy=reset()
```

```

173 #碰撞与重启
174 for y in clouds:
175     for a,b in [y.bottomleft,y.bottomright,y.topleft,y.topright]:
176         if x.bottom>=b>=x.top and x.left<=a<=x.right:
177             is_c=True
178 if is_c==True:
179     time3=time.time()
180     if endtime==None:
181         endtime=time3
182     pretime="{0:0.2f}".format(endtime-time1-pausetime)
183     timetext=str(pretime)
184     speed=[0,0]
185     speedy=[0,0]
186     chongqi=pygame.font.Font('freesansbold.ttf',32)
187     chongqisur=chongqi.render('Press R to restart',True,black)
188     chongqirect=chongqisur.get_rect()
189     chongqirect.center=(175,100)
190     pig.blit(chongqisur,chongqirect)
191     pygame.mixer.music.stop()

```

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后再开始。

2.8 暂停与开始功能

```

192 #暂停与开始
193 if is_stop==True:
194     speed=[0,0]
195     speedy=[0,0]
196     pygame.mixer.music.pause()
197     if is_start==False:
198         kaishi=pygame.font.Font('freesansbold.ttf',32)
199         kaishisur=kaishi.render('Press E to start',True,red)
200         kaishirect=kaishisur.get_rect()
201         kaishirect.center=(175,100)
202         pig.blit(kaishisur,kaishirect)

```

```

119 elif event.type==MOUSEBUTTONDOWN and speed!=[0,0] and speedy!=[0,0] and is_c==False :
120     is_stop=True
121     time3=time.time()
122 elif event.type==MOUSEBUTTONDOWN and speed==[0,0] and speedy==[0,0] and is_c==False and is_start==True and is_stop==True:
123     speed=s
124     speedy=sy
125     is_stop=False
126     time4=time.time()
127     pausetime=pausetime+time4-time3
128     pygame.mixer.music.unpause()

```

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。

2.9 时间与分数

```
62 #时间②
63 time1=time.time()
64 time4=time1
65 time3=time1
66 pretime=0
67 endtime=None
68 pausetime=0

86 #时间②
87 time2=time.time()

119 elif event.type==MOUSEBUTTONUP and speed!=[0,0] and speedy!=[0,0] and is_c==False :
120     is_stop=True
121     time3=time.time()
122 elif event.type==MOUSEBUTTONUP and speed==[0,0] and speedy==[0,0] and is_c==False and is_start==True and is_stop==True:
123     speed=s
124     speedy=sy
125     is_stop=False
126     time4=time.time()
127     pausetime=pausetime+time4-time3

136 elif event.type==KEYDOWN and event.key==K_e and is_start==False:
137     speed=s
138     speedy=sy
139     is_stop=False
140     time4=time.time()
141     pausetime=pausetime+time4-time3
142     is_start=True

160 #暂停与时间②
161 if is_c==False:
162     if is_stop==True:
163         pretime="{0:0.2f}".format(time3-time1-pausetime)
164         timetext=str(pretime)
165     elif is_stop==False:
166         timetext="{0:0.2f}".format(float(pretime)+time2-time4)
167 fontobj=pygame.font.Font("freesansbold.ttf", 28)
168 score="score:"+ "{0:0.0f}".format((((max(float(timetext),0))*10)**0.5)*100)
169 textsurobj=fontobj.render(score,True,green)
170 textrectobj=textsurobj.get_rect()
171 textrectobj.center=(300,50)
172 pig.blit(textsurobj, textrectobj)

178 if is_c==True:
179     time3=time.time()
180     if endtime==None:
181         endtime=time3
182     pretime="{0:0.2f}".format(endtime-time1-pausetime)
183     timetext=str(pretime)
```

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。

2.10 音乐的播放和暂停

```
10 def reset():
11     x=pygame.Rect(0,207,80,68)
12     clouds=[pygame.Rect(375+6*60*(t-0.98),220,40,32) for t in timelist]
13     jump=False
14     is_stop=True
15     is_c=False
16     speed=[0,-5]
17     speedy=[-6,0]
18     time1=time.time()
19     time4=time1
20     time3=time1
21     pretime=0
22     endtime=None
23     pausetime=0
24     is_start=True
25     pygame.mixer.music.play(-1,0.0)
26     jumpnum=0
27     s=[0,-5]
28     sy=[-6,0]
```

```
33 pygame.mixer.init()
34 #音乐
35 pygame.mixer.music.load("sound1.mp3")
```

```
122 elif event.type==MOUSEBUTTONDOWN and speed==[0,0] and speedy==[0,0] and is_c==False and is_start==True and is_stop==True:
123     speed=s
124     speedy=sy
125     is_stop=False
126     time4=time.time()
127     pausetime=pausetime+time4-time3
128     pygame.mixer.music.unpause()
```

```
136 elif event.type==KEYDOWN and event.key==K_e and is_start==False:
137     speed=s
138     speedy=sy
139     is_stop=False
140     time4=time.time()
141     pausetime=pausetime+time4-time3
142     is_start=True
143     pygame.mixer.music.play()
```

```

178     if is_c==True:
179         time3=time.time()
180         if endtime==None:
181             endtime=time3
182         pretime="{0:0.2f}".format(endtime-time1-paus
183         timetext=str(pretime)
184         speed=[0,0]
185         speedy=[0,0]
186         chongqi=pygame.font.Font('freesansbold.ttf',
187         chongqisur=chongqi.render('Press R to restar
188         chongqirect=chongqisur.get_rect()
189         chongqirect.center=(175,100)
190     pig.blit(chongqisur,chongqirect)
191     pygame.mixer.music.stop()
192 #暂停与开始
193     if is_stop==True:
194         speed=[0,0]
195         speedy=[0,0]
196         pygame.mixer.music.pause()

```

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。

第三部分：存在的不足和可能的改进

实 验 的 结 果 如 图 所 示 :



1. 背景比较单调，我们希望给小猪运动加上背景，但由于小猪图片是由红色的小猪和白色的底色组成，所以只好用白色作为背景以达到只突出红色小猪的效果。未来可能会用更有效的方法进行处理，方便添加背景。
2. 小猪碰撞是以矩形框为边缘进行碰撞判断的，这与小猪与云朵实际碰撞可能会有误差，所以更好的改进是用新的函数提取图片中的主体部分进行碰撞判断
3. 背景音乐比较单调，我们希望能添加更多的音乐来使得游戏更富有趣味性。

总结:

我们根据对 `pygame` 的学习，自主设计了小猪快跑的游戏，尽管我们的游戏设计还存在许多不足，但通过游戏的设计使得我们对 `python` 知识应用的更加熟练，并掌握了 `pygame` 设计游戏的基本框架，还有对游戏的基本优化方法。在此过程中我们付出了努力并获得了成长。最后，由衷的感谢我们的鲍扬老师对我们悉心的指点和教导，使我们可以完满的完成这次游戏的设计。