必要的变量的声明

第一个变量会跟踪玩家的精度（这个精度的变量本质上是一个数字组成的列表，记录了射出的箭头数和被击中的獾的数量）

acc=[0,0]

第二个变量会跟踪箭头

arrows=[]

加载箭头的图片

arrow = pygame.image.load("resources/images/bullet.png")

当玩家点击鼠标，就需要射出一支箭头 (这段代码会检查是否鼠标被点击了，如果点击了，它就会得到鼠标的位置并且根据玩家和光标的位置计算出箭头旋转角度。旋转角度的值存放在arrows这个数组里。)

if event.type==pygame.MOUSEBUTTONDOWN:#检查是否点击了鼠标

position=pygame.mouse.get\_pos()#获取鼠标的位置

acc[1]+=1#箭头数加一

arrows.append([math.atan2(position[1]-(playerpos1[1]+32),position[0]-(playerpos1[0]+26)),playerpos1[0]+32,playerpos1[1]+32])#向箭头的列表里加入箭头的旋转角度

画出箭头来

for bullet in arrows:

index=0

vely和velx的值是根据三角定理算出来的。

velx=math.cos(bullet[0])\*10 （10是箭头的速度）

vely=math.sin(bullet[0])\*10

bullet[1]+=velx

bullet[2]+=vely

if表达式是检查箭头是否超出了屏幕范围，如果超出，就删除这个箭头。

if bullet[1]<-64 or bullet[1]>640 or bullet[2]<-64 or bullet[2]>480:

arrows.pop(index)

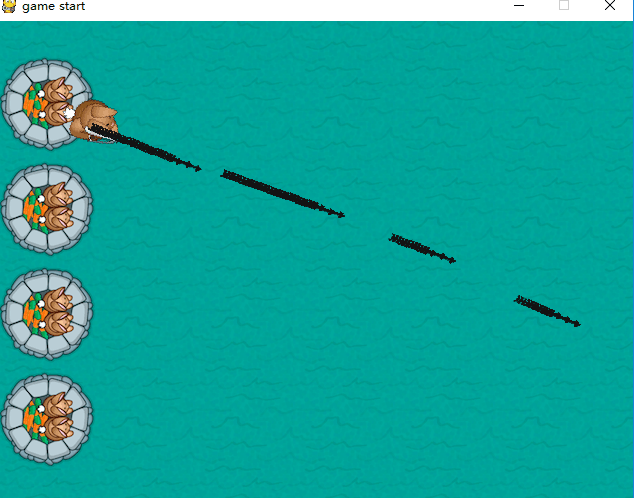
index+=1

循环来把箭头根据相应的旋转画出来。

for projectile in arrows:

arrow1 = pygame.transform.rotate(arrow, 360-projectile[0]\*57.29)

screen.blit(arrow1, (projectile[1], projectile[2]))



用户点击鼠标的时候，兔子就会射出箭，同时记录箭的精度。