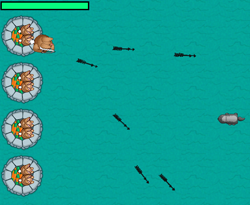
12岁的少年教你用Python做小游戏

2013/07/10 · [Python](http://blog.jobbole.com/category/python/), [开发](http://blog.jobbole.com/category/programmer/) · [37 评论](http://blog.jobbole.com/46308/#article-comment)· [Julian Meyer](http://blog.jobbole.com/tag/julian-meyer/), [Python](http://blog.jobbole.com/tag/python/), [游戏](http://blog.jobbole.com/tag/%e6%b8%b8%e6%88%8f/)

分享到：**222**

本文由 [伯乐在线](http://blog.jobbole.com/) - [贱圣OMG](http://www.jobbole.com/members/fwg1989) 翻译。未经许可，禁止转载！  
英文出处：[Julian Meyer](http://www.raywenderlich.com/24252/beginning-game-programming-for-teens-with-python)。欢迎加入[翻译组](http://group.jobbole.com/category/feedback/trans-team/)。

你有没有想过电脑游戏是怎样制作出来的？其实它没有你想象的那样复杂！在这个教程里，你要学做一个叫《兔子和獾》的塔防游戏，兔子作为英雄，需要在城堡里抵御獾的进攻。



为了写这个游戏的代码，你将会用Python。好吧，我不是指一条大蟒蛇！

Python是一种计算机语言。我们在这篇教程里选择Python是因为这门语言很容易上手，学习起来也很简单和有趣。

如果你是个Python方面的新手，在开始看教程之前你可以看看这本书《 [Think Python: How to Think Like a Computer Scientist](http://greenteapress.com/thinkpython/html/index.html)》。这能让你看教程的时候不那么吃力。

在看了那本书后回到这里并且准备好——兔子和獾之间有一场大战爆发，一起来加入到这场战斗中来吧！

**起步：安装Python**

如果你想在Windows PC上尝试这篇教程里讲到的东西，你需要安装Python。确保你安装的是2.7.3版本，而不是3.3.0版本！在安装程序运行完之后，在开始按钮的“所有程序”里就会有IDLE了。首先启动IDLE。

如果你是用的Mac，上面已经是把Python装好了！打开终端，输入python，然后按回车就行了

注意：如果你是安装的从python.org下载的安装包，那么你在Mac上也可以启动IDLE了，它应该是在 /Application/Python2.7 这个文件夹里。

如果你按以上的步骤执行了，那么你可以看到一下的东西：

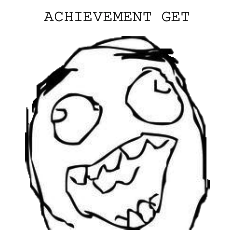
Python



|  |  |
| --- | --- |
| 1  2  3  4 | Python 2.7.3 (v2.7.3:70274d53c1dd, Apr  9 2012, 20:52:43)  [GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin  Type "help", "copyright", "credits" or "license" for more information.  >>> |

注意：如果你想迅速终止Python，你可以输入exit()然后按回车，或者是按Control+D。

现在很迅速的把Python环境配置好了，为了测试下Python是否正常工作，输入 print 1+1 然后按回车，应该会打印出2。你刚才就写了一个简单的Python程序！

[](http://jbcdn2.b0.upaiyun.com/2013/08/HGLbF.png)

那么，现在你知道Python已经能正常工作了，为了能用Python来写一个游戏，你需要安装PyGame。

PyGame是一个Python的库，能够让你更容易的写出一个游戏。它提供的功能包括图片处理和声音重放的功能，并且它们能很容易的整合进你的游戏里。

[点击这里](http://www.pygame.org/download.shtml)下载适合你的PyGame安装包。确保你下载的是针对2.7版本的库。

注意：从以上链接下载的PyGame安装包是无法与Mac上安装的Python一起正常工作的。你需要从 python.org 上下载一个Python安装包才能使用PyGame。或者，你可以用MacPorts来让你的Python和PyGame正常工作。

为了确定PyGame是否安装好了，打开IELD或者是在终端里运行Python，然后输入import pygame 并且回车。如果没有打出任何输出，说明没有问题。

但是，如果打出了如下图的输出，那么PyGame就没有被安装上。

Python



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | Python 2.7.2 (default, Jun 20 2012, 16:23:33)  [GCC 4.2.1 Compatible Apple Clang 4.0 (tags/Apple/clang-418.0.60)] on darwin  Type "help", "copyright", "credits" or "license" for more information.  >>> import pygame  Traceback (most recent call last):    File "<stdin>", line 1, in <module>  ImportError: No module named pygame  >>> |

如果你发现了这样的错误，把出错的情况发到论坛上，我会帮助你把它运行起来

**运行文件里的Python代码**

现在你已经能在Python快捷终端里写一点代码了，如果你想要写一个更大一点的程序（比如游戏），那么你就需要把代码保存到文件里了。

有好几种运行文件里的Python代码的方法。其中一种方法是使用一个简单的文本编辑器，比如Windows系统上的nodepad，或者Mac上的TextEdit。打开一个新的文本文件，输入Python代码（比如print 1+1）。然后将其保存为XXX.py 。（XXX表示文件名）

在Windows上，双击这个py文件来运行它。在Mac上，打开终端然后输入python，然后把代码文件拖到终端上最后再按回车。

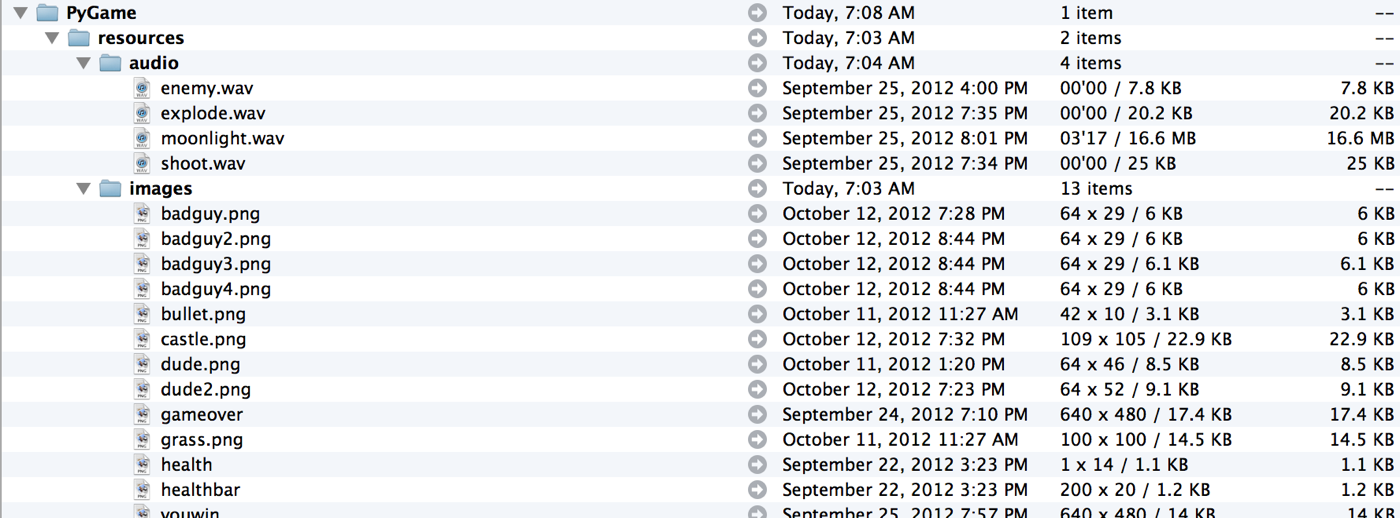
另一种运行代码的方法是使用IDLE编辑器，这是在这篇教程里主要使用的方法。要运行IDLE，需要在终端里输入idle，然后在菜单里选择新建文件，然后就可以利用这个文本编辑器来写Python代码了。你可以通过 File/Save 来保存文件，通过点击 Run/Run（F5）来运行代码。

要注意运行的菜单在编辑器里打开了文件之后才可用。

**添加游戏资源**

你现在已经基本上准备好来创建一个游戏了。但是哪个游戏没有很棒的图像和音效呢？我把游戏里需要的图像和音像保存在了zip档案里，[点击这里下载](http://www.raywenderlich.com/downloads/BB_Resources.zip)。

在下载好文件后，为游戏创建一个文件夹然后把压缩文件解压到这个文件夹的子文件夹里，将其命名为resources，具体如下图：



我们现在可以开始做兔子和獾这个游戏啦！

**第一步：你好，兔子**

运行IDLE，打开一个新的文本编辑窗口。输入以下的代码：

Python

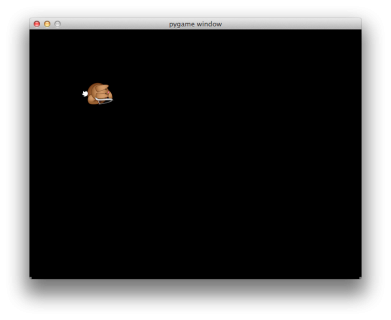


|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | # 1 - Import library  import pygame  from pygame.locals import \*    # 2 - Initialize the game  pygame.init()  width, height = 640, 480  screen=pygame.display.set\_mode((width, height))    # 3 - Load images  player = pygame.image.load("resources/images/dude.png")    # 4 - keep looping through  while 1:      # 5 - clear the screen before drawing it again      screen.fill(0)      # 6 - draw the screen elements      screen.blit(player, (100,100))      # 7 - update the screen      pygame.display.flip()      # 8 - loop through the events      for event in pygame.event.get():          # check if the event is the X button          if event.type==pygame.QUIT:              # if it is quit the game              pygame.quit()              exit(0) |

把文件保存到你的游戏文件夹里，把它命名为 game.py 。我们现在看看这段代码做了什么：

1. 导入pygame库，这一步能让你使用库里提供的功能
2. 初始化pygame，设置展示窗口
3. 加载作为兔子的图片
4. 不停地循环执行接下来的部分
5. 在给屏幕画任何东西之前用黑色进行填充
6. 在屏幕的（100，100）坐标出添加你加载的兔子图片
7. 更新屏幕
8. 检查一些新的事件，如果有退出命令，则终止程序的执行。

在运行这段代码后，你会看到一下的画面：



可以看到一个兔子在这里，准备好要战斗了！

但是暂时兔子看起来还是很孤独的，背景上只有它一个。现在是时候让它更进一步了。

**第二步：添加背景**

我们现在开始给游戏的背景加上一些风景。这可以通过一些 screen.blit() 的调用来实现。

在#3的结尾处，在添加玩家图片的后面，加上以下的代码：

Python



|  |  |
| --- | --- |
| 1  2 | grass = pygame.image.load("resources/images/grass.png")  castle = pygame.image.load("resources/images/castle.png") |

这段代码加载图片并将它们放在变量里。现在它们需要被画在屏幕上。但是当你检查了草的图片后，发现它不会覆盖整个屏幕，它的分辨率是640 x 480。

添加到 game.py 里#6的代码：

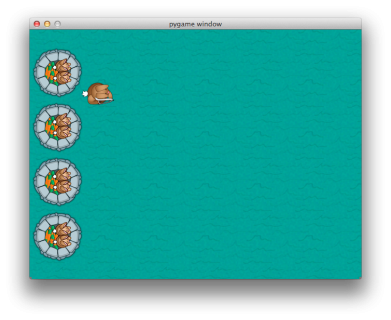
Python



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | for x in range(width/grass.get\_width()+1):          for y in range(height/grass.get\_height()+1):              screen.blit(grass,(x\*100,y\*100))      screen.blit(castle,(0,30))      screen.blit(castle,(0,135))      screen.blit(castle,(0,240))      screen.blit(castle,(0,345 )) |

你可以看到，这段代码首先是依次通过x进行循环。又是一个依次通过y的循环并且根据循环里x和y的值来画上草的效果。接下来的几行就是在屏幕上画出城堡。

如果你现在运行你的代码，效果应该如下图：



好多了——现在开始看起来不错了！

**第三步：让兔子能够移动**

接下来你需要加上一些真正的游戏的元素了，比如让兔子能够随着按键移动。

为了做到这一点，首先，你需要实现一个方法，用来记录在某一时刻那个键被按下。可以通过新建一个按键状态的队列来存放每个你想在游戏里用到的按键。

把以下代码加入到 game.py 里#2后面：

Python



|  |  |
| --- | --- |
| 1  2 | keys = [False, False, False, False]  playerpos=[100,100] |

这段代码是不言自明的，key这个队列用来记录几个按键的情况：WASD。队列中的每一项对应一个按键——第一个对应W，第二个对应A等等。

playerpos这个变量表示程序画出玩家的位置。因为这个游戏会让玩家向不同的方向移动，用一个变量来表示玩家的位置并且能够依据这个变量将玩家画出来的方案是非常适合的。

现在你需要修改现在画出玩家的代码，需要用上playerpos变量。把#6部分的代码进行修改：

从

Python



|  |  |
| --- | --- |
| 1 | screen.blit(player, (100,100)) |

改成：

Python



|  |  |
| --- | --- |
| 1 | screen.blit(player, playerpos) |

接下来，根据按下的键来更新按键记录数组。PyGame里用给按键添加事件的方法来检测按键。

在#8八月份的结尾，就在event.py == pygame.QUIT后面，添加一下的代码（与pygame.QUIT保持同级别缩进）：

Python



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | if event.type == pygame.KEYDOWN:              if event.key==K\_w:                  keys[0]=True              elif event.key==K\_a:                  keys[1]=True              elif event.key==K\_s:                  keys[2]=True              elif event.key==K\_d:                  keys[3]=True          if event.type == pygame.KEYUP:              if event.key==pygame.K\_w:                  keys[0]=False              elif event.key==pygame.K\_a:                  keys[1]=False              elif event.key==pygame.K\_s:                  keys[2]=False              elif event.key==pygame.K\_d:                  keys[3]=False |

Wow!又加了这么多代码。但是如果你仔细看看这些表达式，并没有那么复杂。

首先，你检查是否有一个键被按下或放开。然后，检查是哪一个键被按下或放开了，如果被按下或放开的键是你使用的，你就更新记录按键的变量。

最终，你需要更新playerpos变量作为按键后的反应。这实际上是非常简单的。

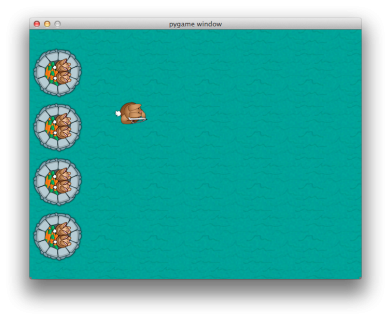
把一下的代码加到game.py的结尾：（让它与for 循环保持同级别缩进）



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | # 9 - Move player      if keys[0]:          playerpos[1]-=5      elif keys[2]:          playerpos[1]+=5      if keys[1]:          playerpos[0]-=5      elif keys[3]:          playerpos[0]+=5 |

这段代码简单的检查了哪个键被按下，然后增加或减少玩家的x和y坐标。

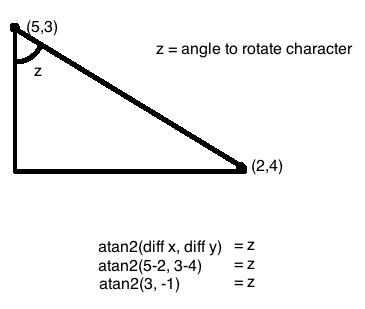
运行这个游戏，那么你应该会看到一下的画面。试着按WASD，耶！好使了！



**第四步：让兔子转向**

好的，现在你的兔子在你按键的时候可以移动了，但是如果你能用鼠标让兔子朝向你选择的方向不是更酷吗?这样它就不会总是朝向一个方向了。用三角定理实现它会非常简单。

看一下下面的插图：



在上图中，如果（5，2）是兔子的位置，（2，4）是现在鼠标的位置，你可以通过三角定理来得出需要旋转的角度。然后，你知道了旋转的角度后，以就可以来旋转你的兔子了。

如果你对这部分感到有点疑惑，不要担心——这没有关系。但这是你为什么需要在数学课上集中精力的原因。在游戏编程中会用得到它的。

现在，你需要接受你的游戏里的概念。为了实现它，你可以使用PyGame Surface.rotate(degrees) 函数。

atatn2函数是来自Python 的math库。所以把一下代码加到#1部分：

Python



|  |  |
| --- | --- |
| 1 | import math |

然后，把#6部分的最后一行用一下代码替换：

Python

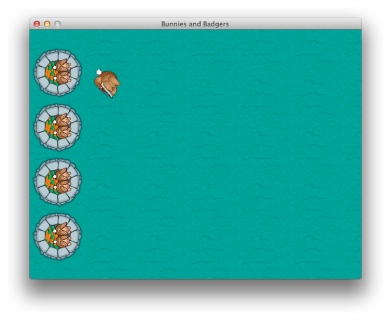


|  |  |
| --- | --- |
| 1  2  3  4  5  6 | # 6.1 - Set player position and rotation      position = pygame.mouse.get\_pos()      angle = math.atan2(position[1]-(playerpos[1]+32),position[0]-(playerpos[0]+26))      playerrot = pygame.transform.rotate(player, 360-angle\*57.29)      playerpos1 = (playerpos[0]-playerrot.get\_rect().width/2, playerpos[1]-playerrot.get\_rect().height/2)      screen.blit(playerrot, playerpos1) |

我们来浏览一下以上代码的基本结构。首先获取鼠标和玩家的位置。然后将它们使用atan2函数。然后，获取通过atan2函数得出的角度和弧度。

当兔子被旋转的时候，它的位置将会改变。所以你需要计算兔子新的位置，然后将其在屏幕上显示出来。

再次运行游戏，如果你只是按WASD这几个键，那么这个游戏跟以前还是一样的。但是如果你移动鼠标，兔子也开始旋转了，相当酷！



**第五步：射吧！兔子**

现在你的兔子可以四处移动了，是时候来添加更多的功能了。让兔子用箭头射向它们的敌人怎么样?

这一步会有一点复杂，因为你需要跟踪所有的箭头，更新它们，旋转它们，在它们跑出屏幕的时候删除它们。

首先，在#2声明的部分加上必要的变量的声明。

Python



|  |  |
| --- | --- |
| 1  2 | acc=[0,0]  arrows=[] |

第一个变量会跟踪玩家的精度，第二个变量会跟踪箭头。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。之后我们会用到这些信息用来计算射击精确度。

接下来，在#3部分结尾加载箭头的图片。

Python



|  |  |
| --- | --- |
| 1 | arrow = pygame.image.load("resources/images/bullet.png") |

现在，当玩家点击鼠标，就需要射出一支箭头。在#8部分加上以下代码：

Python



|  |  |
| --- | --- |
| 1  2  3  4 | if event.type==pygame.MOUSEBUTTONDOWN:              position=pygame.mouse.get\_pos()              acc[1]+=1              arrows.append([math.atan2(position[1]-(playerpos1[1]+32),position[0]-(playerpos1[0]+26)),playerpos1[0]+32,playerpos1[1]+32]) |

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。

接下来，你需要真的在屏幕上画出箭头来。在#6部分加上以下代码：

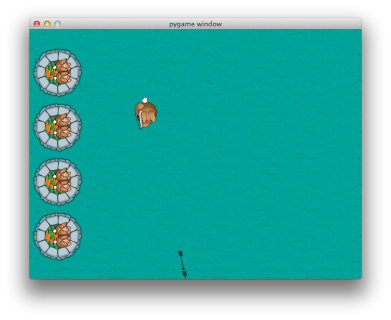
Python



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | # 6.2 - Draw arrows      for bullet in arrows:          index=0          velx=math.cos(bullet[0])\*10          vely=math.sin(bullet[0])\*10          bullet[1]+=velx          bullet[2]+=vely          if bullet[1]<-64 or bullet[1]>640 or bullet[2]<-64 or bullet[2]>480:              arrows.pop(index)          index+=1          for projectile in arrows:              arrow1 = pygame.transform.rotate(arrow, 360-projectile[0]\*57.29)              screen.blit(arrow1, (projectile[1], projectile[2])) |

vely和velx的值是根据三角定理算出来的。10是箭头的速度。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。第二个for表达式是循环来把箭头根据相应的旋转画出来。

试着运行游戏。当你点击鼠标时，你的兔子就可以射出剪头了:D



**第六部：獾，拿上武器!**

好吧，现在有了城堡，并且你有一个英雄可以移动和射出箭头。还差什么呢？攻击城堡的敌人可以被英雄用箭头射了！



在这一步，你将会随机创建出一些獾冲向城堡。在游戏的进程中会有越来越多的獾冒出来。所以，我们来列个接下来要做的事情的清单。

1. 添加一个坏蛋的列表
2. 更新坏蛋的信息，并且检查它们是否超出屏幕范围
3. 展示这些坏蛋

第一步，在#2部分加上一下代码：

Python



|  |  |
| --- | --- |
| 1  2  3  4 | badtimer=100  badtimer1=0  badguys=[[640,100]]  healthvalue=194 |

以上的代码里定义了一个定时器，使得游戏里可以经过一段时间后就新建一只獾。

在#3部分结尾处添加以下代码：

Python



|  |  |
| --- | --- |
| 1  2 | badguyimg1 = pygame.image.load("resources/images/badguy.png")  badguyimg=badguyimg1 |

第一行跟前面加载图片的代码很相似。第二行声明了一个图片的复制。

接下来，你需要更新并且显示这些坏蛋了。在#6.2部分加上以下代码：

Python



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | # 6.3 - Draw badgers      if badtimer==0:          badguys.append([640, random.randint(50,430)])          badtimer=100-(badtimer1\*2)          if badtimer1>=35:              badtimer1=35          else:              badtimer1+=5      index=0      for badguy in badguys:          if badguy[0]<-64:              badguys.pop(index)          badguy[0]-=7          index+=1      for badguy in badguys:          screen.blit(badguyimg, badguy) |

上面的代码看起来有不少。第一行是来检查badtime是否为0，如果为0，创建一个獾然后重新设置badtime。第一个循环更新獾的x坐标，检查獾是否超出屏幕范围，如果超出范围，将獾删掉。第二个循环是来画出所有的獾。

为了在以上代码里用到随机的功能，你需要导入random库。所以在#1部分加上导入的代码：

Python



|  |  |
| --- | --- |
| 1 | import random |

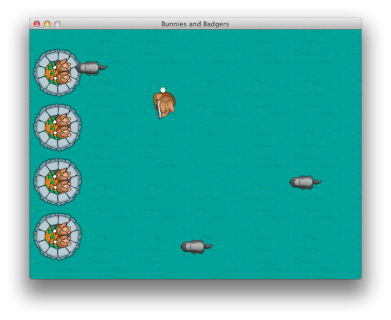
最后，把一行代码添加到#4部分的while表达式后面：

Python



|  |  |
| --- | --- |
| 1 | badtimer-=1 |

试着运行这段代码。现在它应该像个真正的游戏了——你可以移动、射箭、转向，然后有獾冲向你。



但是先慢着！为什么獾没有炸掉城堡？我们快速的把这功能加进来。

把以下代码加到#6.3部分index+=1之前的第一个循环里：

Python



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | # 6.3.1 - Attack castle          badrect=pygame.Rect(badguyimg.get\_rect())          badrect.top=badguy[1]          badrect.left=badguy[0]          if badrect.left<64:              healthvalue -= random.randint(5,20)              badguys.pop(index)          # 6.3.3 - Next bad guy |

这段代码相当简单。如果獾的x坐标离右边小于64，就删除坏蛋并且减少游戏里的健康值，减少的大小为5至20里的一个随机数。

再次运行这个游戏，你就会有一些獾冲过来并且在碰到城堡的时候会消失。尽管你看不到，獾实际上会降低你的健康值。



**第七部：獾与箭头的碰撞**

獾们冲向你的城堡，但是你的箭头对它们完全没有作用！这让兔子怎么放手它的家园呢？

是时候来让箭头能够杀死獾让你能保护你的城堡并且赢得这场游戏。基本上，你需要循环所有的坏蛋，你需要循环所有的箭头来检查是否有碰撞。如果碰撞上，删除獾，删除箭头，并且添加精确度的变量里面加1。

在#6.3.1部分后面加这些：

Python



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | #6.3.2 - Check for collisions          index1=0          for bullet in arrows:              bullrect=pygame.Rect(arrow.get\_rect())              bullrect.left=bullet[1]              bullrect.top=bullet[2]              if badrect.colliderect(bullrect):                  acc[0]+=1                  badguys.pop(index)                  arrows.pop(index1)              index1+=1 |

这段代码里面只有一个地方需要提一下，这个if表达式使用了PyGame内建功能来检查两个矩形是否交叉。接下来的一些代码跟上面说的一样。

再运行下代码，现在你就可以用箭头来杀死獾了。

**第八步：添加健康值和时间的显示**

现在游戏运行起来相当不错了。现在有攻击者，有防守者。现在，你需要的就是通过一个方法来显示兔子的得分。

最简单的方法就是添加一个HUD来显示当前城堡的生命值。你也可以加一个计时来记录城堡存活下来的时间。

首先，添加一个计时。在#7部分加上下面的代码：

Python



|  |  |
| --- | --- |
| 1  2  3  4  5  6 | # 6.4 - Draw clock      font = pygame.font.Font(None, 24)      survivedtext = font.render(str((90000-pygame.time.get\_ticks())/60000)+":"+str((90000-pygame.time.get\_ticks())/1000%60).zfill(2), True, (0,0,0))      textRect = survivedtext.get\_rect()      textRect.topright=[635,5]      screen.blit(survivedtext, textRect) |

上面的代码使用了PyGame默认的大小为24的字体。这个字体用来显示时间信息。

Python



|  |  |
| --- | --- |
| 1  2 | healthbar = pygame.image.load("resources/images/healthbar.png")  health = pygame.image.load("resources/images/health.png") |

接下来添加代码到#6.4部分后面：

Python



|  |  |
| --- | --- |
| 1  2  3  4 | # 6.5 - Draw health bar      screen.blit(healthbar, (5,5))      for health1 in range(healthvalue):          screen.blit(health, (health1+8,8)) |

这段代码首先画了一个全红色的生命值条。然后根据城堡的生命值往生命条里面添加绿色。

运行下代码，现在就会有计时和生命值条了。



**第九步：赢或输**

现在怎么样？如果你玩的时间够长，即使你的生命值已经变成0了，游戏仍然是继续的！不仅如此，你仍然可以用箭头射向这些獾。这肯定不太对。你需要加上一些赢或者输的情况来让你的游戏值得玩。

那么我们来加上胜利或者失败的条件。你可以通过终止主循环，进入胜利/失败的循环来实现它。你需要指出玩家是否胜利，并将其显示在屏幕上。

下面是一些胜利/失败的一些基本条件：

如果时间到了（90秒）那么：

* 停止运行游戏
* l设置游戏的输出

如果城堡被毁，那么：

* 停止运行游戏
* l设置游戏的输出

精确度是一直都需要计算的。

在game.py 结尾添加这些代码：

Python



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | #10 - Win/Lose check      if pygame.time.get\_ticks()>=90000:          running=0          exitcode=1      if healthvalue<=0:          running=0          exitcode=0      if acc[1]!=0:          accuracy=acc[0]\*1.0/acc[1]\*100      else:          accuracy=0  # 11 - Win/lose display  if exitcode==0:      pygame.font.init()      font = pygame.font.Font(None, 24)      text = font.render("Accuracy: "+str(accuracy)+"%", True, (255,0,0))      textRect = text.get\_rect()      textRect.centerx = screen.get\_rect().centerx      textRect.centery = screen.get\_rect().centery+24      screen.blit(gameover, (0,0))      screen.blit(text, textRect)  else:      pygame.font.init()      font = pygame.font.Font(None, 24)      text = font.render("Accuracy: "+str(accuracy)+"%", True, (0,255,0))      textRect = text.get\_rect()      textRect.centerx = screen.get\_rect().centerx      textRect.centery = screen.get\_rect().centery+24      screen.blit(youwin, (0,0))      screen.blit(text, textRect)  while 1:      for event in pygame.event.get():          if event.type == pygame.QUIT:              pygame.quit()              exit(0)      pygame.display.flip() |

这是最长的一段代码！不过它并不复杂。

第一个ifb表达式是检查是否时间到了。第二个是检查城堡是否被摧毁了。第三个计算你的精准度。之后，一个if表达式是检查你是赢了还是输了，然后显示出相应的图片。

当然，如果你想在赢或输的时候显示图片，这些图片首先需要加载。所以在#3部分加上这些代码：

Python



|  |  |
| --- | --- |
| 1  2 | gameover = pygame.image.load("resources/images/gameover.png")  youwin = pygame.image.load("resources/images/youwin.png") |

还有需要改的地方，把#4部分从：

Python



|  |  |
| --- | --- |
| 1  2  3 | # 4 - keep looping through  while 1:      badtimer-=1 |

改成：

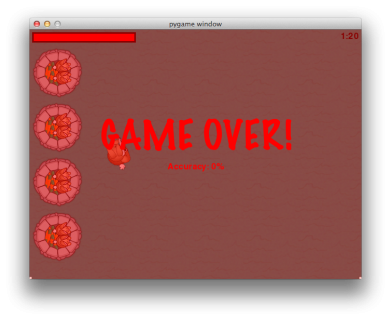
Python



|  |  |
| --- | --- |
| 1  2  3  4  5 | # 4 - keep looping through  running = 1  exitcode = 0  while running:      badtimer-=1 |

这个running变量会跟踪游戏是否结束，exitcode变量会跟踪玩家是否胜利。

再次运行游戏，你就发现你可以胜利或者失败了！酷



**第十步：免费的音乐和声音效果**

这个游戏现在相当不错了，但是没有声音。有点太安静了。加上一点声音效果，能让你更好地感受这个游戏。

PyGame能够让加载和播放声音非常简单。首先，你在#2部分结尾加上这些代码：

Python



|  |  |
| --- | --- |
| 1 | pygame.mixer.init() |

然后在#3部分加载声音然后设置声音：

Python



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | # 3.1 - Load audio  hit = pygame.mixer.Sound("resources/audio/explode.wav")  enemy = pygame.mixer.Sound("resources/audio/enemy.wav")  shoot = pygame.mixer.Sound("resources/audio/shoot.wav")  hit.set\_volume(0.05)  enemy.set\_volume(0.05)  shoot.set\_volume(0.05)  pygame.mixer.music.load('resources/audio/moonlight.wav')  pygame.mixer.music.play(-1, 0.0)  pygame.mixer.music.set\_volume(0.25) |

上面大部分代码就是加载声音文件并且配置音量。但是注意 pygame.mixer.music.load这行代码——这一行加载游戏的背景音乐然后下一行让背景音乐一直不停的播放。

现在注意声音的配置。现在你所有需要做的就是在需要的时候播放不同的音乐。

Python



|  |  |
| --- | --- |
| 1  2  3  4  5  6 | # section 6.3.1 after if badrect.left<64:  hit.play()  # section 6.3.2 after if badrect.colliderect(bullrect):  enemy.play()  # section 8, after if event.type==pygame.MOUSEBUTTONDOWN:  shoot.play() |

再次运行游戏，你会发现现在有背景音乐了，并且在射出箭头和碰撞的时候会有音效。这游戏就更加逼真啦！

**那接下来呢？**

你应该为自己感到自豪：你刚刚完成了一个游戏；里面包含了音乐，音效，一个杀手兔子，自杀冲锋的獾等等。我跟你说了这确实是可以完成的！

你可以从[这里下载](http://cdn2.raywenderlich.com/downloads/BB_FinalGame.zip)最终的游戏代码。

在这基础上，你可以根据你自己的创意来扩展游戏！你可以试着重置里面的各种图片，加上不同的枪或者是不同的怪物！

如果你对这篇教程有什么问题和建议。请留言。