

آموزش بازی سازی دوبعدی با زبان پایتون

و کمک گرفتن از کتابخانه pygame

تابستان 1391 ، نویسنده : منصور بهزادی

www.behzadim.ir

مقدمه

در این کتاب ما نکات کلیدی ای را که برای ساخت یک بازی دوبعدی نیاز است را آموزش می دهیم. پیش نیاز این کتاب این است که شما با زبان پایتون آشنا شده باشید و همچنین بهترست استعداد و سابقه ای در بازی سازی دو بعدی داشته باشید. کتاب بصورت کلی شما را با امکانات پایتون آشنا می کند ولی یاد گرفتن نکات تکمیلی را به خودتان واگذار می کند.

نصب نرم افزار های مورد نیاز

ابتدا شما باید خود زبان پایتون را نصب کنید. سپس نسخه ای از pygame که با نسخه پایتون نصب شده یکن منطبق باشد را نصب کنید. این دو نرم افزار را می توانید به راحتی و رایگان در سایت های رسمیشان دریافت و نصب کنید. من در حال حاضر از این دو نسخه استفاده می کنم:

Python 3.1.2

pygame-1.9.1.win32-py3.1

پس از نصب پایتون وارد محیط محاوره ای پایتون شوید و با دستور زیر مطمئن شوید که pygame هم نصب شده:

```
>>> import pygame
```

آنچه برای ساخت یک بازی لازم است

پیش از هرچیز باید شما ذوق و استعداد داشته باشید. بعد با کدنویسی پیاده سازی الگوریتم و بخصوص با شئی گزایی آشنا باشید. سپس در یک زبان یا محیط خاص نحوه رسم اشکال ، تصاویر و آنچه برای یک بازی است را بیاموزید. آنگاه شما می توانید یک ایده بازی بنویسید آنرا در سبک شئی گزایی تحلیل و تنظیم کنید و سپس در محیط برنامه نویسی بازی را طراحی و تست کنید. در این کتاب فقط بخش سوم یکی کار با گرافیک آن هم در پایتون آموزش داده می شود. امید وارم که برایتان مفید باشد.

الگوریتم کلی کار(پیشنهاد من)

شما برای طراحی یک بازی نیاز به یک حلقه بی نهایت دارید. پیش از اجرای این حلقه شما باید فایل های مورد نیاز(عکس،موزیک ...) را لود کنید و هرکدام را در متغیرهایی قرار دهید. سپس در حلقه بینهایت ، سه تابع که بیرون حلقه ساخته اید را صدا بزنید. این سه حلقه یکی حلقه دریافت رویداد ، یکی تابع منطقی و دیگری تابع گرافیکی است. شما در حلقه منطقی که اصل کار شما هست باید اشیاء مکان ها و به طور کلی منطق بازی را کنترل کنید (که خود این تابع توسط تابع رویداد کنترل می شود). و در تابع گرافیک اشیاء مورد نظر یا موزیک مورد نظر را (احتمالا در مختصات جدید) پیاده سازی کنید. برای مثال فرض کنید که می خواهید یک دایره را روی صفحه حرکت دهید ، برای این کار نخست دو متغیر به عنوان مختصات میانه دایره را تعریف

سپس در بخش تابع گرافیک دایره را نسبت به دو متغیر رسم کنید. سپس با توجه به منطق مورد نظر تان دو متغیر را در تابع منطق تغییر دهید.

آغاز کد نویسی

صدا زدن کتابخانه های مورد نیاز

```
Import pygame,sys
```

```
From pygame.locals import *
```

ابتدا کتابخانه های sys و pygame را فراخوانی میکنیم. تا بتوانیم از توابع و اشیائی که درون آنهاست استفاده کنیم.

```
pygame.init()
```

همیشه پس از افزودن کتابخانه پای گیم به کد خود این متد از پای گیم را فراخوانی کنید. خیلی مهم نیست که کار این متد چیست فقط کافیست بدانید این متد در هنگام لود کردن فایل های بازی نقش مهمی دارد. مهم: بهترست همیشه پیش از استفاده از توابع پای گیم حتما این متد را فراخوانی کنید.

ساختن و باز کردن پنجره بازی

```
Screen = pygame.display.set_mode((400, 300))
```

این دستور یک شیء از نوع پنجره را می سازد و درون متغیر Screen قرار می دهد. همزمان پنجره ساخته شده را نمایش می دهد. پس از این خط ما برای دسترسی به پنجره بازیمان باید از متغیر Screen استفاده کنیم.مثلا:

```
Screen.fill((255,255,255))
```

با این دستور ما تمام صفحه پنجره مان را رنگ کردیم(سفید)=(255,255,255))

```
pygame.display.set_caption('نخستین بازی من')
```

دستور بالا برای مشخص کردن عنوانی پنجره ای است که باز می شود.

نوشتن حلقه بی نهایت (هسته بازی)

```
while True: # main game loop
```

```
    pygame.display.update()
```

در تکه کد بالا ما یک حلقه بینهایت که همان هسته بازیمان هست را نوشته ایم. اما تا حالا مگر در جایی ما از حلقه ای استفاده کرده ایم که نیاز با پایان نداشته باشد.

دریافت و مدیریت رویدادها

شما می توانید با دستور `break` و یا `pygame.quit()` به ترتیب از حلقه و یا از کل بازی خارج شوید. برای خارج شدن از بازی شما باید مثلا منتظر رویداد خاصی از طرف کاربر باشید پس ما با تکه کد زیر حلقه مان را تکمیل می کنیم:

```
while True: # main game loop

    for event in pygame.event.get():

        if event.type == QUIT:

            pygame.quit()

            sys.exit()
```

این حلقه تا زمانی ادامه می یابد که کاربر دکمه X بالای پنجره بازی را بزند. تکه کد اضافه شده تمام رویدادهای کاربر را میخواند سپس با رویداد QUIT یعنی زدن دکمه خروج پنجره مقایسه می کند. اگر زده شد برنامه را می بندد. اگر شما بخواهید بازیتان نسبت به دیگر ورودی های موس و صفحه کلید نیز عکس العمل داشته باشد کافی است زیر شرط `if` یک شرط دیگر (ترجیحا `elif`) برای آن رویداد قرار دهید.

همیشه آخرین خط حلقه باید دستور زیر باشد:

```
pygame.display.update()
```

این دستور محتوای `Screen` یا همان پنجره را بهنگام می کند. یعنی فرض کنید شما می خواهید 5 خط روی صفحه بکشید پس از نوشتن دستوران برای این 5 خط با این دستور پنجره را بروز کرده تا نمایان شوند.

تا به ایجای کار شما می توانید یک پنجره با رنگ پس زمینه دلخواه رسم کنید و رویدادهای آن را مدیریت کنید.

برای تمرین ، برنامه ای بنویسید که مختصات موس را روی عنوان پنجره نمایش دهد. یا تعداد کلیک های موس را در عنوان نمایش دهد.

رنگ ها

(red,green,blue,alpha)

هنگامی که ما بخواهیم رنگی بسازیم از دستور بالا استفاده می کنیم. ما می توانیم از سه رنگ قرمز ، آبی و سبز رنگ مورد نظرممان را بسازیم، پارامتر چهارم اختیاری است که میزان شفافیت رنگ را می توانیم مشخص کنیم . همه پارامترها اعدادی بین 0 تا 255 را می گیرند.

همچنین می میتوانیم با استفاده از متد color در python خودمان یک رنگ بسازیم:

```
myColor = pygame.Color(255, 0, 50, 100)
```

اکنون می توان از متغیر myColor به عنوان یک رنگ استفاده نمود.

شیء مستطیل

در pygame یک شیء با چهار ویژگی X,Y,Height,Width وجود دارد که شما می توانید متغیری از نوع آن را تعریف کنید و هر جا که خواستید آنرا تغییر یا از آن استفاده نمایید.

```
spamRect = pygame.Rect(10, 20, 200, 300)
```

توجه کنید که R هست نه r . پس از تعریف این متغیر شئی از نوع مستطیل دارید که ویژگیهای زیادی دارد که قابل تغییر است اگر کمی جلو تر برویم متوجه فایده این شیء می شوید، قبل از اینکه من با این شیء آشنا شوم بجای آن میبایست خودم کلاس مستطیل را تعریف می کردم.

رسم اشکال هندسی

توابعی در شیء draw در کتابخانه ی pygame وجود دارد که برای رسم اشکال هندسی از آنها استفاده می شود. چند مثال از چند تا از توابع مهم رو نوشتم که باتوجه به اسمشون و مثالها ، نحوه استفاده و کارشون مشخصه و من توضیحی دربارشون نمیدم:

```
pygame.draw.line(Screen, BLUE, (120, 60), (60, 120))
```

```
pygame.draw.circle(Screen, BLUE, (300, 50), 20, 0)
```

```
pygame.draw.rect(Screen, RED, (200, 150, 100, 50))
```

```
pygame.draw.ellipse(Screen, RED, (300, 250, 40, 80), 1)
```

```
pygame.draw.polygon(Screen, GREEN, ((146, 0), (291, 106), (236, 277), (56, 77), (0, 106)))
```

اولی رسم خط ، دومی دایره ، سومی مستطیل ، چهارمی بیضی و پنجمی هم چند ضلعی(در اینجا 5 ضلعی)

توجه کنید که با تابع aaline بهتر میتوانید خط رسم کنید با این دستور خطی که می کشید نرم است یعنی حاشیه ی دندانان دندانان ای ندارد.

دسترسی به حافظه گرافیکی

شیء `pygame.PixelArray` یک شیء خیلی جالب برای دسترسی به حافظه صفحه نمایش(درون پنجره) بصورت آرایه دو بعدی:

```
scrrarray=pygame.PixelArray(screen)
```

```
scrrarray[5][5]=(0,255,0)
```

```
del scrrarray
```

همونطور که می بینید با تعریف یک شیء از این نوع ما می توانیم مستقیم به یک پیکسل از پنجره یک رنگ اختصاص دهیم. از این ویژگی برای مثال در حرکت یک شیء می توان استفاده نمود. نکته مهم اینجاست که همیشه پس از استفاده از شیء ایجاد شده از این نوع باید آن را پاک کنیم. زیرا در هنگام چیات این شیء `screen` قفل میشود. برای اینکه در هر لحظه بدانیم شیء `screen` قفل است یا نه میتوان از دستور `(get_locked)` استفاده کرد ،که بله یا خیر برمی گرداند.

استفاده از عکس در بازی

لود کردن فایل عکس : برای اینکه ما بازی جذابی داشته باشیم یقینا می بایست از عکس های زیبا هم استفاده کنیم. برای لود کردن یک عکس ابتدا باید فایل عکس را درون یک متغیر بریزیم سپس آن را رسم کنیم.

تعریف و ریختن فایل در یک متغیر:

```
mypic = pygame.image.load('pic.png')
```

رسم متغیر عکی روی صفحه نسبز به ایتصورت است:

```
screen.blit(mypic, (x, y))
```

توجه شود که `x` و `y` مکان عکس را مشخص می کنند.

ایجاد وقفه در بازی

FPS به معنی فریم در ثانیه است. یعنی تصاویر روی صفحه ما هر ثانیه چند بار بروز بشوند. ما این امکان را داریم که تعیین کنیم هر چند ثانیه یک بار شیء screen بروز شود. اگر تعداد فریم در ثانیه کمتر باشد سی پی یو کمتر اشغال شده و در نتیجه سرعت بازی بیشتر می شود. به تکه کد زیر توجه کنید:

```
fpsClock = pygame.time.Clock()
```

```
while True:
```

```
...
```

```
    fpsClock.tick(10)
```

ما یک شیء ساعت تعریف کردیم(بیرون از حلقه) سپس در داخل حلقه با مقداری دهی به متد tick حلقه را نگه داشته و زمان را به سی پی یو داده ایم. هر چه عدد کمتر باشد سرعت تکرار حلقه هم کمتر می شود. استاندارد عدد 30 مناسب است ولی شما می توانید این عدد را تغییر دهید. بهتر است همیشه بعد از دستور `pygame.display.update()` دستور `fpsClock.tick()` را قرار دهید.

تابع دیگری هم وجود دارد که به میزانی که ما میگوییم درنگ می کند:

```
time.sleep(0.5)
```

ورودی برحسب ثانیه است وقتی برنامه به این خط می رسد به میزان ورودی ، ثانیه صبر می کند سپس ادامه میدهد. اگر میخواهید از این دستور استفاده کنید. `Import time` را در بالای برنامه فراموش نکنید.

خودم هنوز زیاد تفادت ویژگی های استفاده از `time.sleep()` و `Clock.tick()` را متوجه نشدم. ولی هر دو این دو دستور علاوه بر ایجاد وقفه وقت را به سیستم (سی پی یو) می دهند.

نوشتن متن روی صفحه نمایش

برای این کار باید مراحل زیر رو طی کنید.

```
fontObj = pygame.font.Font('نام فونت مورد نظر.ttf', اندازه متن مورد نظر به پوینت)
```

ابتدا فونت مورد نظر را لود کنید.

```
textSurfaceObj = fontObj.render('متن مورد نظر', True, رنگ متن, رنگ پس زمینه)
```

سپس با شیء فونت یک تصویر را از متن مورد نظر به صورت یک تصویر رندر می کنیم. اگر دوست ندارید رنگی را برای پس زمینه انتخاب کنید این پارامتر را ننویسید (, قبل از آنرا هم پاک کنید). در

صورتی که می خواهید در مکان خاصی نوشته تان رسم شود به این صورت مکان آن را تنظیم کنید.

```
textRectObj = textSurfaceObj.get_rect()
```

```
textRectObj.center = (x, y)
```

محدوده متن را بصورت یک شیء مستطیل برگردانید سپس با تغییر ویژگی های آن مکان نوشته را تغییر دهید. در مثال بالا به اینگونه مکان مشخص شده که مختصات ویانه ی مستطیل را تعیین کردیم.

پس از انجام تنظیمات بالا میتوانین مانند یک عکس متنتان را روی صفحه چاپ کنید:

```
DISPLAYSURF.blit(textSurfaceObj, textRectObj)
```

استفاده از صدا در بازی

این امکان در جذاب کردن بازی شما نقش مهمی دارد. شما می توانید دو نوع صدا داشته باشید ، یکی صدایی در پس زمینه و یکی صداهایی در هنگام رویداد ها(مثلا در هنگام شلیک کردن).

برای پخش کردن صدایی در پسزمینه بصورت زیر عمل کنید:

```
pygame.mixer.music.load('backgroundmusic.mp3')
```

```
pygame.mixer.music.play(-1, 0.0)
```

با این دو دستور یک فایل صوتی را لود کرده و آن را اجرا می کنید. و هر وقت خواستید می توانید آن را قطع کنید:

```
pygame.mixer.music.stop()
```

همچنین می توانید صدای آنرا کم و زیاد کنید:

```
pygame.mixer.music.set_volume(صدای)
```

امکانات دیگر را خودتان می توانید با دستکاری دیگر خاصیت های `pygame.mixer.music` یاد بگیرید.

صدا هنگام رویداد

صدایی که شما برای پس زمینه انتخاب می کنید در هر زمان فقط می توان یک صدا را پخش کرد چون با نسبت دادن یک فایل جدید به `pygame.mixer.music` فایل پیشین پاک میشود. برای رفع این مشکل شما باید یک متغیر صدا تعریف و مقدار دهی کنید سپس هر وقت خواستید آن را اجرا و قطع کنید.(به همین راحتی)


```
import time
soundObj = pygame.mixer.Sound('پسوند.صدا')
soundObj.play()
time.sleep(ثانیه)
soundObj.stop()
```

فکر می کنم این چند خط دیگر نیاز به توضیح نداشته باشد.

بنابراین شما اکنون می توانید یک بازی دوبعدی گرافیکی ، کم حجم ، شامل صدا و جذاب را طراحی کنید.

به امید سربلندی کشور عزیزمان

پیروز باشید

تابستان 1391 ، نویسنده : منصور بهزادی

www.behzadim.ir