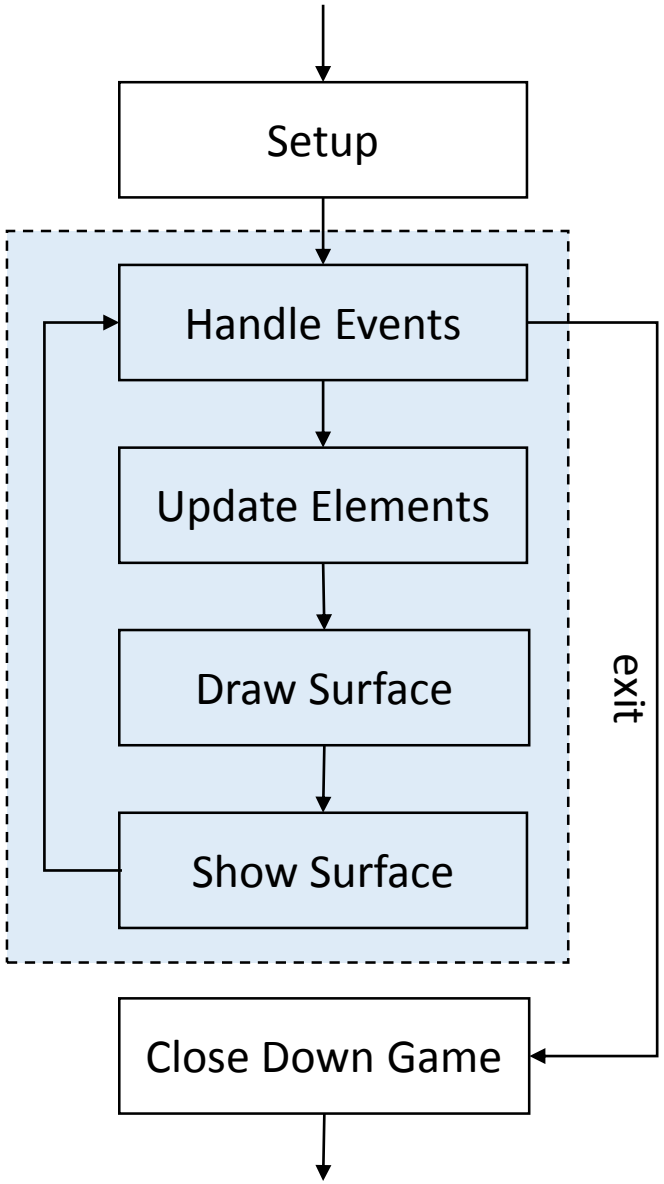
برنامه سازی پیشرفته
(برنامه نویسی شیءگرا: مفاهیم بیشتر)

صادق اسکندری - دانشکده علوم ریاضی، گروه علوم کامپیوتر

eskandari@guilan.ac.ir

یادآوری ...

چرخه بازی در pygame



ایجاد یک پنجره و بارگذاری برخی محتویات

در صورت بروز یک رخداد (مانند کلیک بر روی یک شیء، بستن پنجره و ...) به آن رسیدگی می شود.

اعمال تغییرات مورد نیاز بر روی عناصر بازی

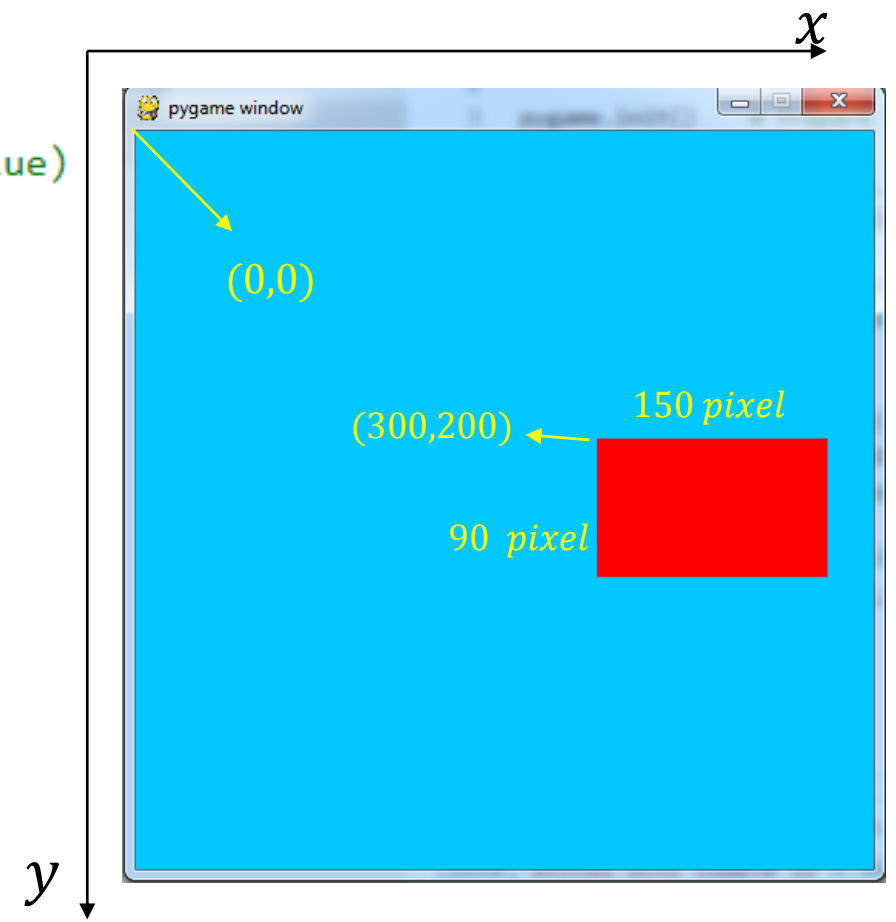
رسم عناصر بازی در پس زمینه

نمایش عناصر بازی

اتمام بازی

یادآوری ...

```
1   import pygame
2
3   pygame.init()    # Prepare the pygame module for use
4
5   # Create surface of (width, height), and its window.
6   main_surface = pygame.display.set_mode((480, 480))
7
8   small_rect = [300, 200, 150, 90]
9   some_color = [255, 0, 0]  # A color is a mix of (Red, Green, Blue)
10
11  while True:
12      ev = pygame.event.poll() # Look for any event
13      if ev.type == pygame.QUIT: # Window close button clicked?
14          break # ... leave game loop
15
16      main_surface.fill([0, 200, 255])
17      main_surface.fill(some_color, small_rect)
18
19      pygame.display.flip()
20
21  pygame.quit() #
```



یادآوری ...

افزودن عکس به صفحه بازی

خواندن فایل عکس (این عکس در کنار فایل اصلی برنامه قرار دارد.

اعمال تغییرات مختلف بر روی عکس خوانده شده

افزودن عکس خوانده شده به مختصات خاصی از پنجره برنامه

```
ball = pygame.image.load("ball.png")
ball =  pygame.transform.scale(ball,(50,50))
main_surface.blit(ball, (100, 120))
```

افزودن متن به صفحه بازی

تعریف یک فونت

ایجاد متن با استفاده از فونت ایجاد شده

افزودن متن ایجاد شده به مختصات خاصی از پنجره برنامه

```
my_font = pygame.font.SysFont('Courier', 16)
the_text = my_font.render('Hello World', True, (0,0,0))
main_surface.blit(the_text, (10, 10))
```

یادآوری ...

مثال: توپ رقصان (ایجاد یک کلاس توپ)

```
class Ball:
    def __init__(self, source='ball.png', scale=(100,100), vx = 1, vy = 1, x = 0, y = 0):
        self.ball=ball = pygame.image.load(source)
        self.ball = pygame.transform.scale(self.ball,scale)
        self.vx = vx
        self.vy = vy
        self.x = x
        self. y =y

    def move(self, surface):
        self.x += self.vx
        self.y += self.vy
        if self.x  >= surface.get_width() or self.x <=0 :
            self.vx = - self.vx
        if self.y >= surface.get_height() or self.y <=0 :
            self.vy = -self.vy
        surface.blit(self.ball, (self.x, self.y))
```

نوع داده: Ball

صفات

x , y: موقعیت

vx,vy: سرعت

ball: تصویر توپ

رفتارها

move(surface): جابجایی و نمایش این توپ بر روی صفحه

pygame

رخدادها

یک رخداد، دستوری است که از جانب کاربر یا سیستم به برنامه در حال اجرا ارسال می شود.

تابحال فقط رخداد Exit را مورد استفاده قرار دادیم ولی رخدادهای بسیار زیادی قابل دریافت توسط pygame است. مانند کلیک، حرکت ماوس، فشردن یک کلید، رها کردن یک کلید و ...

هر رخداد در pygame، یک شیء از نوع event است که حاوی یک نام و یک دیکشنری است.

این دیکشنری، اطلاعات اضافی درباره رخداد را در خود دارد.

در صورتی که رخدادی دریافت نشده باشد، نام رخداد برابر NOEVENT و دیکشنری آن تهی خواهد بود.

pygame

مثال: با استفاده از دستور زیر می توان رخدادهای مختلفی را که در جریان اجرای برنامه رخ می دهند، نمایش داد:

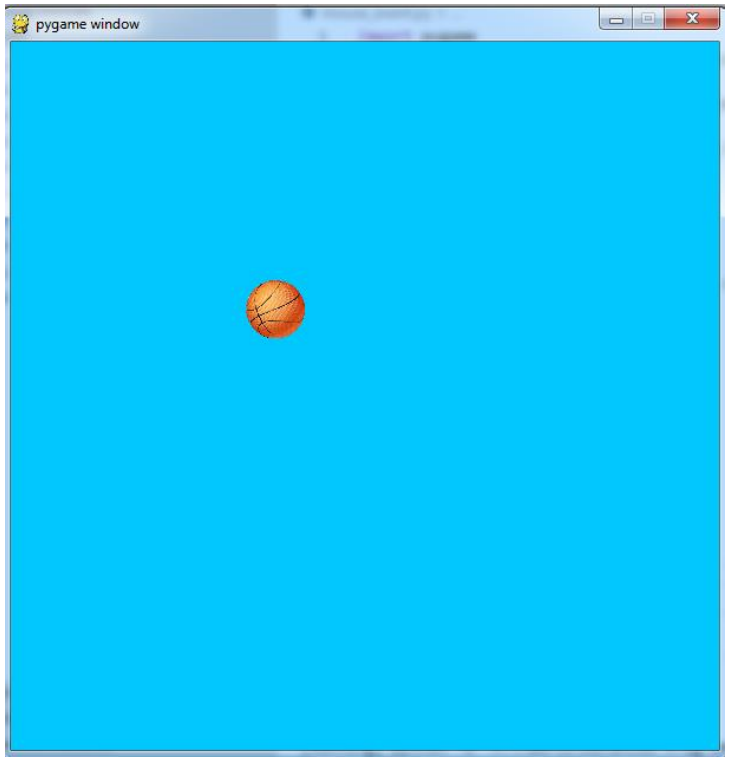
```
while True:
    ev = pygame.event.poll()
    if ev.type != pygame.NOEVENT:
        print(ev)
```

```
<Event(17-VideoExpose {})>
<Event(16-VideoResize {'h': 600, 'w': 600, 'size': (600, 600)})>
<Event(1-ActiveEvent {'state': 1, 'gain': 0})>
<Event(4-MouseMotion {'buttons': (0, 0, 0), 'pos': (599, 0), 'rel': (600, 0)})>
<Event(1-ActiveEvent {'state': 1, 'gain': 1})>
<Event(4-MouseMotion {'buttons': (0, 0, 0), 'pos': (585, 76), 'rel': (-14, 76)})>
<Event(4-MouseMotion {'buttons': (0, 0, 0), 'pos': (555, 82), 'rel': (-30, 6)})>
<Event(4-MouseMotion {'buttons': (0, 0, 0), 'pos': (525, 89), 'rel': (-30, 7)})>
<Event(4-MouseMotion {'buttons': (0, 0, 0), 'pos': (495, 96), 'rel': (-30, 7)})>
<Event(4-MouseMotion {'buttons': (0, 0, 0), 'pos': (467, 104), 'rel': (-28, 8)})>
<Event(4-MouseMotion {'buttons': (0, 0, 0), 'pos': (446, 108), 'rel': (-21, 4)})>
<Event(4-MouseMotion {'buttons': (0, 0, 0), 'pos': (426, 114), 'rel': (-20, 6)})>
<Event(4-MouseMotion {'buttons': (0, 0, 0), 'pos': (410, 118), 'rel': (-16, 4)})>
```

pygame

مثال: استفاده از اطلاعات حرکتی ماوس

```
1   import pygame
2   pygame.init() # Prepare the PyGame module for use
3   max_x, max_y = 600,600
4   main_surface = pygame.display.set_mode((max_x,max_y))
5   ball = pygame.image.load("ball.png")
6   ball =  pygame.transform.scale(ball,(50,50))
7   x,y = 300,300
8
9
10  while True:
11      ev = pygame.event.poll()
12      if ev.type == pygame.QUIT: # Window close button clicked?
13          break # Leave game loop
14      if ev.type == pygame.MOUSEMOTION:
15          x,y= ev.pos
16
17      main_surface.fill((0, 200, 255))
18
19      main_surface.blit(ball, (x,y))
20
21      pygame.display.flip()
22
23  pygame.quit()
```

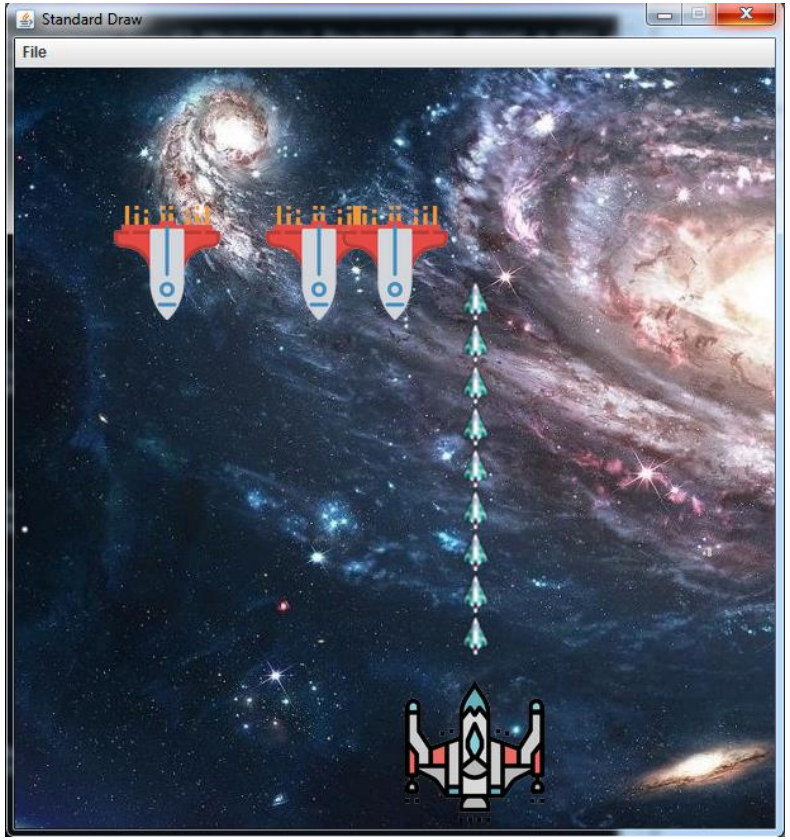


pygame

مثال: افزودن توپ به صفحه با رخداد کلیک

```
pygame.init() # Prepare the PyGame module for use
max_x, max_y = 600,600
main_surface = pygame.display.set_mode((max_x,max_y))
balls =[]


while True:
    ev = pygame.event.poll()
    if ev.type == pygame.QUIT: # Window close button clicked?
        break # Leave game loop
    if ev.type == pygame.MOUSEBUTTONUP:
        scale = random.randint(1,100)
        b = Ball(scale=(scale,scale), vx = random.random()*5, vy = random.random()*5, x =
        random.randint(0,max_x), y=random.randint(0,max_y))
        balls.append(b)

    main_surface.fill((0, 200, 255))

    for b in balls:
        b.move(main_surface)

    pygame.display.flip()

pygame.quit()
```

تمرین



تمرین: بازی جنگ ستارگان را بنویسید.

سعی کنید برای هر نوع موجودیتی (سفینه، دشمنان مختلف، گلوله و ...) یک کلاس طراحی کنید.

نکات بیشتر درباره برنامه نویسی شیء گرا

مثال: نوع داده زمان

کلاس اولیه

```
1   class MyTime:
2       def __init__(self, h = 0, m = 0, s = 0):
3           self.hours = h
4           self.minutes = m
5           self.seconds = s
6
7       def __str__(self):
8           return '%d : %d : %d'%(self.hours, self.minutes, self.seconds)
```

مشکل MyTime چیست؟

```
t1 = MyTime(3,12,54)
print(t1)

t2 = MyTime()
print(t2)
```

```
3 : 12 : 54
0 : 0 : 0
```

مثال: نوع داده زمان

کلاس اولیه

```python
class MyTime:
    def __init__(self, h = 0, m = 0, s = 0):
        self.hours = h
        self.minutes = m
        self.seconds = s

    def __str__(self):
        return '%d : %d : %d'%(self.hours, self.minutes, self.seconds)
```

مشکل MyTime چیست؟

امکان تعریف زمان های نامعتبر

```python
t3 = MyTime(133,5404,434)
print(t3)
```

```
133 : 5404 : 434
```

مثال: نوع داده زمان

```python
class MyTime:
    def __init__(self, h = 0, m = 0, s = 0):
        while s >= 60:
            s -= 60
            m += 1

        while m >= 60:
            m -= 60
            h += 1

        self.hours = h
        self.minutes = m
        self.seconds = s

    def __str__(self):
        return '%d : %d : %d'%(self.hours, self.minutes, self.seconds)
```

```python
t3 = MyTime(1,61,61)
print(t3)
```

```
2 : 2 : 1
```

راهکار اول:

تبدیل ثانیه های اضافی به دقیقه و دقیقه های اضافی به ساعت

مشکل این راهکار:

در زمان نوشتن متدهای مختلف، نیاز به نظر گرفتن سه مقدار ساعت، دقیقه و ثانیه داریم. ☹☹

مثال: نوع داده زمان

مثال: مقایسه دو زمان

```
    def after(self, time2):
        """ Return True if I am strictly greater than time2 """
        if self.hours > time2.hours:
            return True
        if self.hours < time2.hours:
            return False

        if self.minutes > time2.minutes:
            return True
        if self.minutes < time2.minutes:
            return False

        if self.seconds > time2.seconds:
            return True
        return False
```

راهکار اول:

تبدیل ثانیه های اضافی به دقیقه و دقیقه های اضافی به ساعت

مشکل این راهکار:

در زمان نوشتن متدهای مختلف، نیاز به در نظر گرفتن سه مقدار ساعت، دقیقه و ثانیه داریم. ☹☹

مثال: نوع داده زمان

راهکار اول:

تبدیل ثانیه های اضافی به دقیقه و دقیقه های اضافی به ساعت

مشکل این راهکار:

در زمان نوشتن متدهای مختلف، نیاز به نظر گرفتن سه مقدار ساعت، دقیقه و ثانیه داریم. ☹☹

مثالی دیگر: افزودن زمان به زمان فعلی

```
    def increment(self,time2):
        self.seconds += time2.seconds
        self.minutes += time2.minutes
        self.hours += time2.hours

        while self.seconds >= 60:
            self.seconds -= 60
            self.minutes += 1

        while self.minutes >= 60:
            self.minutes -= 60
            self.hours += 1
```

مثال: نوع داده زمان

متدهای کمکی برای تبدیل زمان به ثانیه و بالعکس

```
def to_seconds(self):
    return self.hours * 3600 + self.minutes * 60 + self.seconds

def seconds_to_time(self,s):
    h = s // 3600
    m = (s % 3600) // 60
    s = (s % 3600) % 60
    return h,m,s
```

راهکار دوم:

تبدیل زمان به ثانیه و انجام عملیات در قالب ثانیه

بنابراین، سازنده کلاس به صورت زیر قابل بیان است:

```
def __init__(self, h = 0, m = 0, s = 0):
    self.hours = h
    self.minutes = m
    self.seconds = s
    self.hours, self.minutes, self.seconds = self.seconds_to_time(self.to_seconds())
```

مثال: نوع داده زمان

متدهای کمکی برای تبدیل زمان به ثانیه و بالعکس

```
def to_seconds(self):
    return self.hours * 3600 + self.minutes * 60 + self.seconds

def seconds_to_time(self,s):
    h = s // 3600
    m = (s % 3600) // 60
    s = (s % 3600) % 60
    return h,m,s
```

راهکار دوم:

تبدیل زمان به ثانیه و انجام عملیات در قالب ثانیه

و حال متدهای **increment** و **after** به صورت زیر قابل تعریف هستند.

```
    def after(self, time2):
        """ Return True if I am strictly greater than time2 """
        return self.to_seconds() > time2.to_seconds()

    def increment(self,time2):
        self.hours, self.minutes, self.seconds = self.seconds_to_time(self.to_seconds()+time2.to_seconds())
```