explain_hw1

20171300 지용환

```
86    screenWidth = 1000
87    screenHeight = 900
```

First, this is window size. (1000*900). Width is 1000, and height is 900. Since this is global variable, I can use this variable in some function.

```
94    current_path = os.path.dirname(__file__)
95    assets_path = os.path.join(current_path, 'abe')
```

For find the image, and sound file I use os.path.dirname(), and join.

```
96    sound = pygame.mixer.Sound(os.path.join(assets_path, 'sound.wav'))
84    pygame.init()
```

By using, Sound method, I can get sound file, but must do pygame.init() before use Sound method.

```
99    for i in range(101):
100       if(i<7):
101
102           image = pygame.image.load(os.path.join(assets_path, 'a.png'))
103       elif(i<14):
104           image = pygame.image.load(os.path.join(assets_path, 'b.png'))
105       elif(i<21):
106           image = pygame.image.load(os.path.join(assets_path, 'c.png'))
107       elif(i<28):
108           image = pygame.image.load(os.path.join(assets_path, 'd.png'))
109       elif(i<35):
110           image = pygame.image.load(os.path.join(assets_path, 'e.png'))
111       elif(i<42):
112           image = pygame.image.load(os.path.join(assets_path, 'f.png'))
113       elif(i<49):
114           image = pygame.image.load(os.path.join(assets_path, 'g.png'))
115       elif(i<56):
116           image = pygame.image.load(os.path.join(assets_path, 'h.png'))
117       elif(i<63):
118           image = pygame.image.load(os.path.join(assets_path, 'i.png'))
119       elif(i<70):
120           image = pygame.image.load(os.path.join(assets_path, 'j.png'))
121       elif(i<77):
122           image = pygame.image.load(os.path.join(assets_path, 'k.png'))
123       elif(i<84):
124           image = pygame.image.load(os.path.join(assets_path, 'l.png'))
125       elif(i<91):
126           image = pygame.image.load(os.path.join(assets_path, 'm.png'))
```

This is how I got the image. Using for loop, I get 14 image. This image sizes are 20*20 or 30*30 or 40*40.

```
    if(image.get_width()==20):
        r=10
        m=2
    elif(image.get_width()==30):
        r=15
        m=4
    elif(image.get_width()==40):
        r=20
        m=7
```

Since image width is one of 20,30,40, using get_witdh I get image size and set up r (radius), and m(mass). The radius here is just the length of one side/2.

```
ball=Ball(image,random.randint(70,800), random.randint(70,800), r, m, random.randint(1, 10), random.randint(1,10), WHITE)
ballList.append(ball)
```

Using ball class, make image, and add this class in list.

```
class Ball:
    def __init__(self,ii, px, py, r, m, vx, vy, col):
        self.posX = px+r  #사진의 중심
        self.posY = py+r
        self.velX = vx
        self.velY = vy
        self.colour = col
        self.rad = r
        self.mass = m
        self.im=ii
        #self.sound=s
#         print(vx,vy)
```

This is ball constructor. posX, posY mean center of image, and r,m is already explained. vx,vy are velocity of image.

Now, I explain about keyboard_image.

```
keyboard_x = int(screenWidth / 2)
keyboard_y = int(screenHeight / 2)
keyboard_dx = 0
keyboard_dy = 0
keyboard_image = pygame.image.load(os.path.join(assets_path, 'keyboard.png'))
```

This is same. location, velocity.

```
while True:
    main()
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            break
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                keyboard_dx = -3
            elif event.key == pygame.K_RIGHT:
                keyboard_dx = 3
            elif event.key == pygame.K_UP:
                keyboard_dy = -3
            elif event.key == pygame.K_DOWN:
                keyboard_dy = 3
        elif event.type == pygame.KEYUP:
            if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
                keyboard_dx = 0
            elif event.key == pygame.K_UP or event.key == pygame.K_DOWN:
                keyboard_dy = 0
    keyboard_x += keyboard_dx
    keyboard_y += keyboard_dy

    fps.tick(60)
```

Using while loop, First main() is all about drawing pygame. And then get key event. If we push esc. then pygame is quit. If we push arrow keys then key_image velocity chagenge and release key arrow, then velocity become 0. Using this velocity changing, location of key_image changed, and will be drawed in main funciton.

```
def main():
    screen.fill(BLACK)
    for ball in ballList:
        ball.draw()
        ball.wallColl()
        for ball2 in ballList:
            if ball != ball2 and distanceSq(ball, ball2) <= (ball.rad + ball2.rad) ** 2:
                collision(ball, ball2)
        ball.setPos()
    screen.blit(keyboard_image, [keyboard_x, keyboard_y])
    pygame.display.flip()
```

This is main function. Always fill screen black first(this mean clean screeen), and draw images, in balllist using ball.draw(). check ball collision and wall collision by wallColl() and collision, and change location using setPos. (image velocity change in wallColl() or collision). Also, draw key_image in this main function. Then we can

darw 101 movinge images and key_image.

```
23      def draw(self):
24          screen.blit(self.im, [self.posX-self.rad, self.posY-self.rad])  #
25          #screen.blit(self.im, [self.posX, self.posX])
```

This is ball.draw(). Just darw image using location.

```
26      def wallColl(self):     # Checks and handles wall collisions
27          if self.posX - self.rad <= 0:
28              sound.play()
29              self.posX = self.rad + 1
30              self.velX = - self.velX/5*4
31          elif self.posX + self.rad >= screenWidth:
32              sound.play()
33              self.posX = screenWidth - self.rad - 1
34              self.velX = - self.velX/5*4
35          if self.posY - self.rad <= 0:
36              sound.play()
37              self.posY = self.rad + 1
38              self.velY = - self.velY/5*4
39          elif self.posY + self.rad >= screenHeight:
40              sound.play()
41              self.posY = screenHeight - self.rad - 1
42              self.velY = - self.velY/5*4
43
```

This is Ball.wallColl(). self.posX-self.rad mean left side of image. If this value <=0, this mean crashing left side of window, then make a sound, and change a velocity. self.posX=self.rad+1 is just avoid noise. If we not use this code, when crash the wall, it will continue to sound. Similary, check all 4 corners crash and make sound, change velocity and change location little.

```
    def setPos(self):
        self.posX = toInt(self.posX + self.velX)
        self.posY = toInt(self.posY + self.velY)
```

This is setPos. Use velocity and previous location, to create the next location. In this case, toInt fucntion is rounding function.

```
def toInt(x):
    if x - int(x) < 0.5:
        return int(x)
    else:
        return int(x+1)
```