Python: Images

Computer Science ICS20

Images

Instead of simply drawing circles, rectangles and lines, pygame can also be used to display images. The code below is our standard template for our 60 fps animation loop. It draws a simple red circle on the screen. We will now add some code to load an image and display it alongside the circle. Images can be obtained from many sources: you can create them yourself with a painting program or a camera. You can also download existing images from the Internet, but you must be careful not to infringe on copyright laws, especially if you wish to sell your program or even just distribute it online for free. Some images are available without any restrictions, such as those identified as Creative Commons and the use of these types of images should be encouraged.

```
import pygame
pygame.init()     # initializes the graphics module
window = pygame.display.set_mode((800,600))     # define window size
pygame.display.set_caption('PyGame Images & Sound')  # title of program that
                                                     # appears on window frame
clock = pygame.time.Clock()     # used to track time within the game (FPS)
quit = False
while not quit:                                 # main program loop
    for event in pygame.event.get():            # check if there were any events
        if event.type == pygame.QUIT:           # check if user clicked the upper
            quit = True                         # right quit button

      # your code that draws to the window goes here
      pygame.draw.circle(window, red, (100, 100), 50)


    pygame.display.update()         # refresh your display
    clock.tick(60)                  # wait a certain amount of time that
                                    # ensures a frame rate of 60 fps
pygame.quit()                       # shutdown module
```

For this example, we will use an image of a billiard ball that was found online and that has a Creative Commons license. It is available on the course website.

We will also use a background image of a billiard table that was found on the same website. In this example we want an image to use as a background so that we can incorporate it into a game.

How Images are Stored in Computer Memory

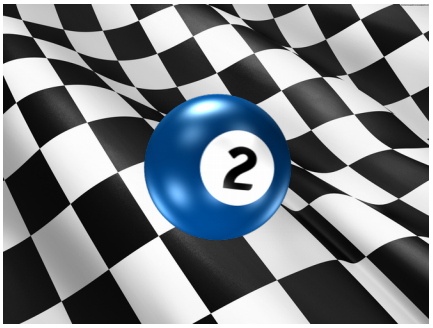
An image from the point of view of a computer is simply a list of RGB values. Each RGB value represents one pixel on the screen. Images can be stored in different file formats in order to compress them so that they take up

less space in memory. These types of images are called **lossy**, because during the compression of the image, some information is lost. This may be ok for some very high resolution photos where the loss won't be noticeable, such as on websites, or where the images are used as thumbnails or icons. An example a of lossy file format is JPEG.

There are other file formats that retain all of the information of the original image. These are called **lossless** file formats. Examples of lossless file formats are PNG are BMP.

Images and Transaparency

Some image file formats have an extra "channel" in addition to RGB. This is normally referred to as an alpha channel and it is used for transparency information. In this case, the image is a list of RGBA values, where the alpha channel is denoted by the 'A' in RGBA. It stores the **opacity** for a pixel which means that each pixel could have a certain amount of transparency. Below are two examples of a billiard ball against a checkered background:



Both images of the billiard ball are rectangular images even though the image on the right doesn't appear to be. The billiard ball on the left is a JPEG file and does not have an alpha channel. You can see that its white background is not transparent. The image on the right is a PNG file with an alpha channel. Its background was set to be 100% transparent and, as a result, the checkered background is visible.

So if you are using an image for a background then it doesn't matter what type of file format you use since transparency is not a requirement. However if you need to draw an image on top of another image, then choosing an image with an alpha channel is recommended. PNG or BMP files can have alpha channels, so look for these. Otherwise you will need to use a program like Photoshop or GIMP to add an alpha channel.

Loading Images

In order to use an image, pygame requires that you load the file into memory first. This is done by the following line of code:

```
ball = pygame.image.load('billiard_blue_PNG10930.png')
```

The name of the file is given as a String to the load() function. The location of the file in the above example is assumed to be in the same location as the python file itself. If this is not the case then the relative or absolute path to the file should be provided.

Drawing Images to the Screen

Once the image is successfully loaded into memory you can display it on the screen using the following command:

```
window.blit(ball, (100, 200))  # draw the ball at location (100, 200)
```

The modified template code now looks like this with the changes shown in red:

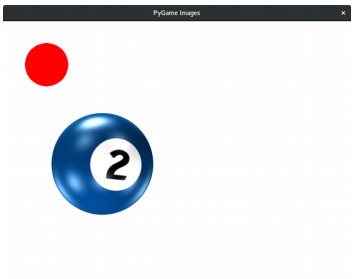
```
import pygame
pygame.init()     # initializes the graphics module
window = pygame.display.set_mode((800,600))     # define window size
pygame.display.set_caption('PyGame Images & Sound')  # title of program that
                                                     # appears on window frame
red=[255,0,0]
white = [255,255,255]
ball = pygame.image.load('billiard_blue_PNG10930.png')

clock = pygame.time.Clock()     # used to track time within the game (FPS)
quit = False
while not quit:                             # main program loop
    for event in pygame.event.get():        # check if there were any events
        if event.type == pygame.QUIT:       # check if user clicked the upper
            quit = True                     # right quit button

      window.fill(white)
      # your code that draws to the window goes here
      pygame.draw.circle(window, red, (100, 100), 50)
      window.blit(ball, (100, 200))

    pygame.display.update()         # refresh your display
    clock.tick(60)                  # wait a certain amount of time that
                                    # ensures a frame rate of 60 fps
pygame.quit()                       # shutdown module
```

Type this in to your program and run it. You should get the following:



Displaying Text using PyGame

Displaying text in PyGame consists of two two steps:

1. Choose and create a font.
2. Use the font in step 1. to render an image of the text.

For example, to create a font object of type “monospace” and a font size of 25:

```
font = pygame.font.SysFont("monospace", 25)
```

Then, to render some text we can use the following function:

```
label = myfont.render("Hello World!", 1, black)
```

The format of the render function is as follows:

```
render(text, antialias, color, background=None)
```

where:

text = text to be displayed

antialias = 1 to turn antialias on, 0 to turn it off

color = colour of text

background = optional background

Check out the pygame font documentation for more details.

Exercises

1. Write a Python program that draws a ball image with a transparent alpha channel background on the screen. You will need to obtain a ball image by either creating one yourself or downloading it from the Internet.
2. Modify your Python program from question 1 to include a background image. Once again you will need to find a suitable image yourself. Hint: you will also need to replace the window.fill() command with a window.blit() command.
3. Modify your Python program from question 2 so that the first image travels across the screen and bounces off both the right and left edges. Add some text centered near the bottom of the window which displays the number of times the ball hits an edge.
4. Modify your program so that the speed of the ball speeds up by one unit every 5 seconds.
5. Modify your Python program again so that if the user clicks on the ball, it changes direction.
6. Finally modify your program so that if the ball hits either edge 3 times the animation stops and a score appears in the center of the screen showing the elapsed time. See how long you can play the game!