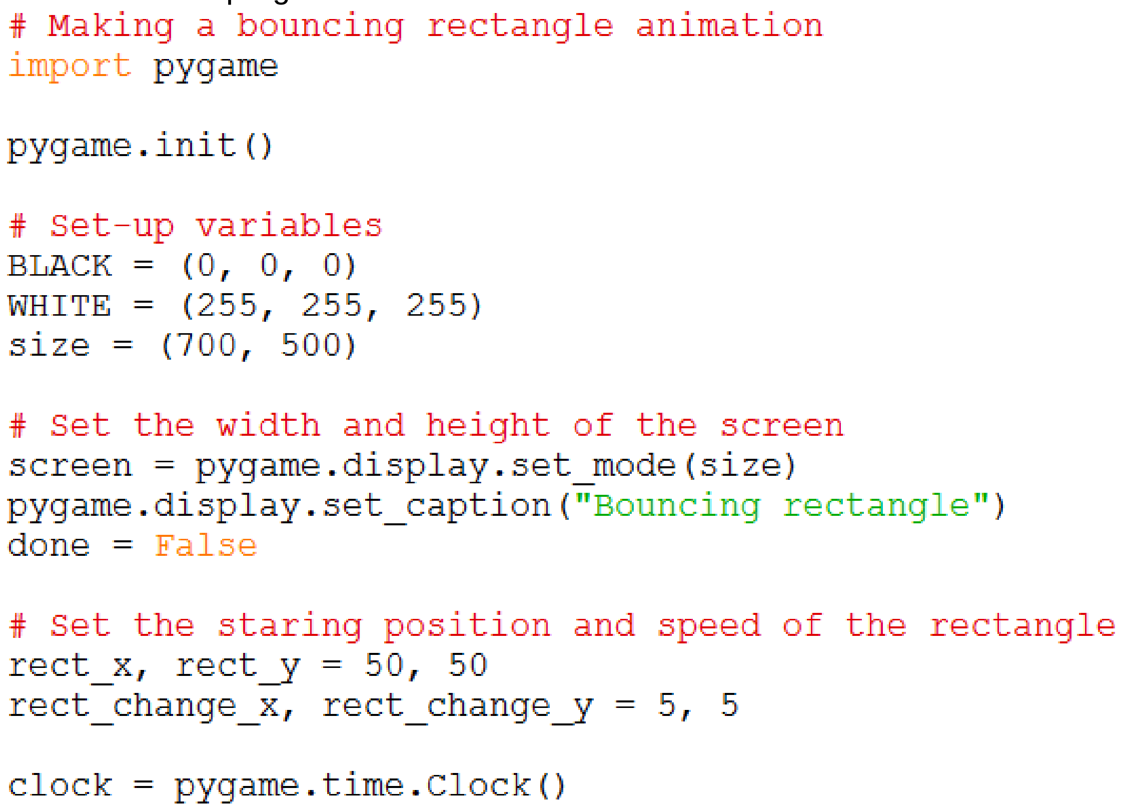
Pygame lesson 2

In the previous lesson you learned about drawing, and how to set up the basic game loop. The next thing we want to do is learn how to make things bounce off the screen. This turns out to be really easy.

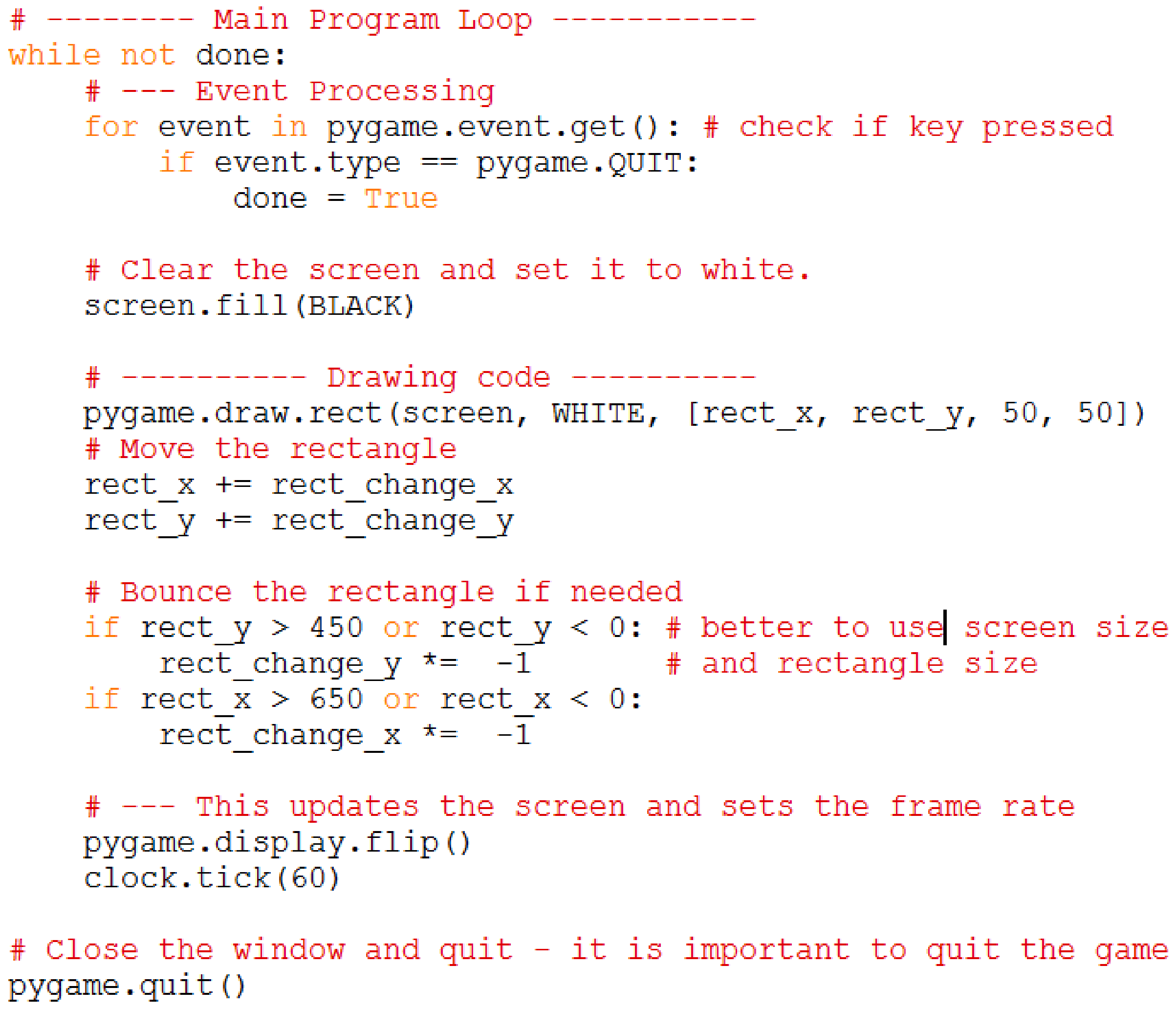
Bouncing rectangle

The start of the program:



```
# Making a bouncing rectangle animation
import pygame

pygame.init()

# Set-up variables
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
size = (700, 500)

# Set the width and height of the screen
screen = pygame.display.set_mode(size)
pygame.display.set_caption("Bouncing rectangle")
done = False

# Set the staring position and speed of the rectangle
rect_x, rect_y = 50, 50
rect_change_x, rect_change_y = 5, 5

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

The rest of the program:



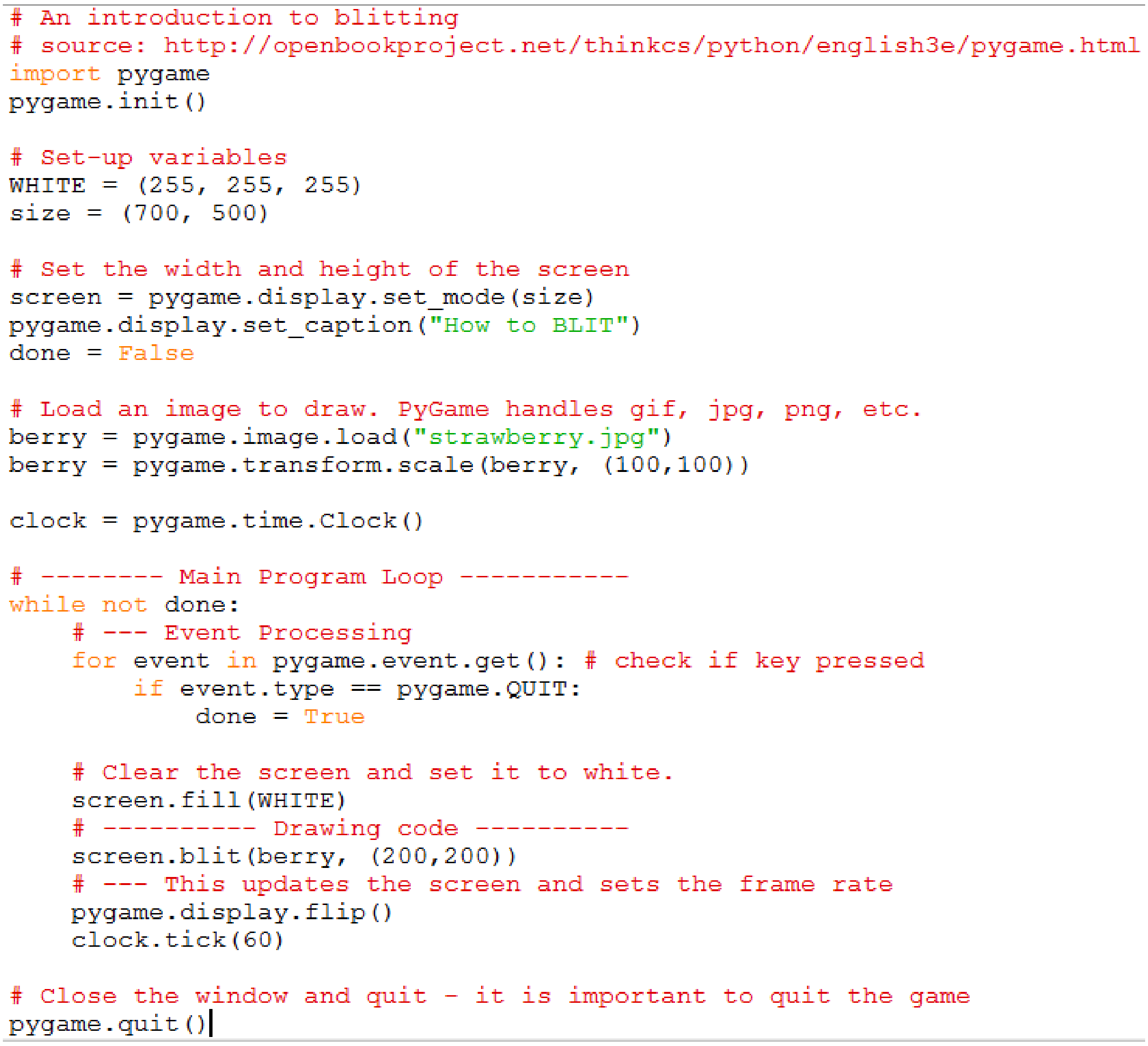
```
# -------- Main Program Loop -----------
while not done:
    # --- Event Processing
    for event in pygame.event.get(): # check if key pressed
        if event.type == pygame.QUIT:
            done = True

    # Clear the screen and set it to white.
    screen.fill(BLACK)

    # ---------- Drawing code ----------
    pygame.draw.rect(screen, WHITE, [rect_x, rect_y, 50, 50])
    # Move the rectangle
    rect_x += rect_change_x
    rect_y += rect_change_y

    # Bounce the rectangle if needed
    if rect_y > 450 or rect_y < 0: # better to use screen size
        rect_change_y *=  -1       # and rectangle size
    if rect_x > 650 or rect_x < 0:
        rect_change_x *=  -1

    # --- This updates the screen and sets the frame rate
    pygame.display.flip()
    clock.tick(60)

# Close the window and quit - it is important to quit the game
pygame.quit()
```

The ‘bounce’ code really should use the screen size and rectangle size. To see why, change the screen size at the top of the program to 800, 600 and see what happens.

Importing images

Bringing images into Pygame is called BLITTING – it means making a fast copy from one area of memory to another.

I downloaded an image for this (strawberry.jpg), you will need to find one yourself. Because my image was too big I used pygame's transform option to resize it. See the code below:



```
# An introduction to blitting
# source: http://openbookproject.net/thinkcs/python/english3e/pygame.html
import pygame
pygame.init()

# Set-up variables
WHITE = (255, 255, 255)
size = (700, 500)

# Set the width and height of the screen
screen = pygame.display.set_mode(size)
pygame.display.set_caption("How to BLIT")
done = False

# Load an image to draw. PyGame handles gif, jpg, png, etc.
berry = pygame.image.load("strawberry.jpg")
berry = pygame.transform.scale(berry, (100,100))

clock = pygame.time.Clock()

# -------- Main Program Loop -----------
while not done:
    # --- Event Processing
    for event in pygame.event.get(): # check if key pressed
        if event.type == pygame.QUIT:
            done = True

    # Clear the screen and set it to white.
    screen.fill(WHITE)
    # ---------- Drawing code ----------
    screen.blit(berry, (200,200))
    # --- This updates the screen and sets the frame rate
    pygame.display.flip()
    clock.tick(60)

# Close the window and quit - it is important to quit the game
pygame.quit()
```

Do this for yourself. Then use what you have learned previously and make the image move around the screen