pygame-lab

Michele Pratusevich

January 27, 2014

1 Lab4, part 2: Pygame

1.1 Exercise 1

Play with pygame, since that is what we are going to be using for the group project. Copy the following code into a new file under Lab4 (name it pygame_example.py) and try to understand exactly what is happening in the code. Change some parameters - see what changes.

```
import pygame

if __name__=="__main__":
    pygame.init()
    screen_size = 400
    main_screen = pygame.display.set_mode((screen_size, screen_size))
    main_screen.fill((255,255,255))
    square = pygame.Surface([20, 20])

    while True:
        ev = pygame.event.poll()
        if ev.type == pygame.QUIT:
            sys.exit()
        if ev.type == pygame.MOUSEBUTTONDOWN:
            x, y = ev.pos
            if square.get_bounding_rect().collidepoint(x, y):
                print "clicked"
        main_screen.blit(square, (0, 0))
        pygame.display.flip()
```

1.2 Exercise 2

1. In the previous exercise you had the variable `square = pygame.Surface([20, 20])` that creates a box on the screen. Instead of doing this with an instance of the `Surface` class, we want to do this with a custom class of ours. In the `pygame_example.py` file, above the main method but below the `import pygame` statement, create a new class that subclasses the pygame Surface class. The class header should look like this: `class MySurface(pygame.Surface): ....`
2. Your custom surface class should store it's own color in a variable.
3. Change the `square` from the original main code to use your own custom `MySurface` class.

4. Test to make sure it works, and when you are done, call an instructor / TA over for a demo