CSC290: Technical Writing Excerpts

The following two exerpts both explain line 8 of the code below. (The exerpts are based on the tutorial at `https://www.pygame.org/docs/tut/PygameIntro.html`)

```
1  import sys, pygame
2  pygame.init()
3
4  size = width, height = 320, 240
5  speed = [2, 2]
6  black = 0, 0, 0
7
8  screen = pygame.display.set_mode(size)
9
10 ball = pygame.image.load("intro_ball.gif")
11 ballrect = ball.get_rect()
12 # ...
```

Activity 1: Version A

`pygame.display.set_mode()` in line 8 creates a window. It defaults to the best graphical modes based on hardware. You can override it if you want. Each image is represented as a `Surface`. `display.set_mode()` creates a new surface that represents the actual displayed graphic. It makes drawings you do to this `Surface` visible on the monitor.

Activity 1: Version B

The call to `pygame.display.set_mode()` in line 8 creates a new graphical window. Pygame defaults to using the best graphical mode for your graphics hardware. You can override the mode to compensate for anything the hardware cannot do. Pygame represents images as `Surface` objects. The `display.set_mode()` function creates a new `Surface` object that represents the actual displayed graphics. Any drawings you make on this `Surface` will become visible on the monitor.

Activity 2

A commonly used architectural pattern for developing user interfaces is the Model-view-controller pattern. It divides an application into three interconnected parts. It is based on the way information is presented to and accepted from the user. Input is converted to commands for the model or view by the controller. Data and logic and rules of the application, which are independent from the user interface, the is managed by the model, The view can be any output representation of information. There can be more than one view. It is commonly used in web applications.