1 Administrative Topics

- We take the quiz.
- Any questions about project 5?

2 Organizing code for animation with Zelle’s graphics objects

In class today we rewrote code that would bounce a funky beach ball up and down. Our goal was to organize the code better to allow for larger groups of objects. Our solution was to place the parts of each complex object in lists.

Below is the code we wrote. It relies on a set of helper functions that Stephanie wrote. It is a module called basic_shapes.py and the functions make it easy to create a graphics object and set the colors and widths in one fell swoop.

```python
import graphics
import sys
import time
import basic_shapes as ba
import aggregate as ag
```
def main1(nsteps):
    # create the window
    win = graphics.GraphWin("Silence", 300, 300)

    # create and fancify my objects
    beach_ball_part1 = ba.createCircle(150, 150, 20, "blue", "yellow", 4)
    beach_ball_part2 = ba.createCircle(165, 150, 20, "blue", "yellow", 4)
    beach_ball = [beach_ball_part1, beach_ball_part2]
    beach_part1 = ba.createRectangle(0, 170, 300, 300, graphics.color_rgb(210, 180, 140), None, None)
    beach = [beach_part1]

    # draw my objects
    ag.draw(beach_ball, win)
    ag.draw(beach, win)

    # bounce the ball
    idx = 0
    while not win.checkMouse():
        time.sleep(0.1)
        if idx % (2*nsteps) < nsteps:
            for item in beach_ball:
                item.move(3, 5)
        else:
            for item in beach_ball:
                item.move(-3, -5)
        idx = idx + 1

    # wait till the user is sick of looking at it
    #win.getMouse()
    win.close()

if __name__ == '__main__':
    if len(sys.argv) > 1:
        nsteps = int(sys.argv[1])
    else:
nsteps = 5
main1(nsteps)