1 Administrative Topics

- We take the quiz

2 Project 11

We made a 3D box:

```python
# boxy thing
# draw the back face first
# if the orientation is the "home" orientation,
# then width is in x-direction, height in y, depth in z
# and (x,y,z) is at the bottom back corner

class ThreeDSquare( Shape ):
    def __init__( self, width, height, depth, color=(0,0,0) ):
        Shape.__init__( self, distance=1, angle=90, color=color,
                        istring = 2*("("+str(width)+")F+"+str(height)+")F")+
                        ""+str(depth)+")F&"+
                        "("+str(width)+")F[&("+str(depth)+")F]+")+
                        "("+str(height)+")F[&("+str(depth)+")F]+")+
                        "("+str(width)+")F[&("+str(depth)+")F]+")+
                        "("+str(height)+")F")
```

```python
```
We didn’t talk about this in class, but here is some advice for drawing shapes with the 3D turtle:

- If you want to fill a shape, you need to construct and fill each face separately. Otherwise, the turtle won’t know what the outside surface is.

- To draw wire mesh shapes (which can look quite nice with the NPR features), I frequently draw the bottom, then pitch back to go up one edge, then right my turtle and draw the top, but at each vertex on the top, I use a branch (‘[’ and ‘]’) to draw the connecting line back down to the corresponding bottom vertex.

- Take advantage of parameterization – you can use multiple distances and multiple angles when creating your shapes. This is particularly useful for pyramids.

- Start with simple examples. I worked for quite awhile trying to get my pyramid to work and found that, along the way I needed to prove to myself that I could draw a triangle (of a given height and base) first.

- When doing trigonometry, remember that the math module uses radians but that the turtle uses degrees. You can convert from radians to degrees by multiplying by 180 and dividing by math.pi.

- If you draw a small object in front of a large object, you may find the large object is drawn “in front” of the small object. This has to do with the algorithm for determining “nearness” to the viewer. To solve the problem, don’t draw large objects. For example, you can draw the front of a house by tiling small, identical squares, instead of by using one large square.