Slides for the Week

CS273 Laboratory 11

This week's lab focuses on inheritance

You will transform this sea of homogenous faces:

face

embarassed

happy

face

sad

jealous

groucho

sad

happy

embarassed

jealous

groucho

into this:

face

embarassed

happy

face

sad

groucho

sad

jealous

happy

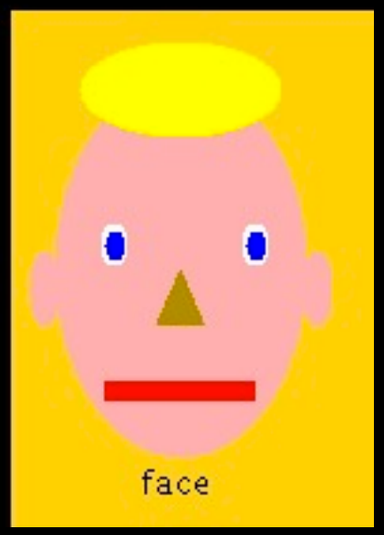
embarassed

jealous

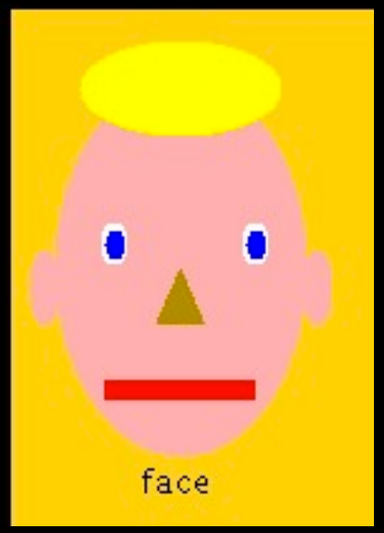
groucho

using inheritance.

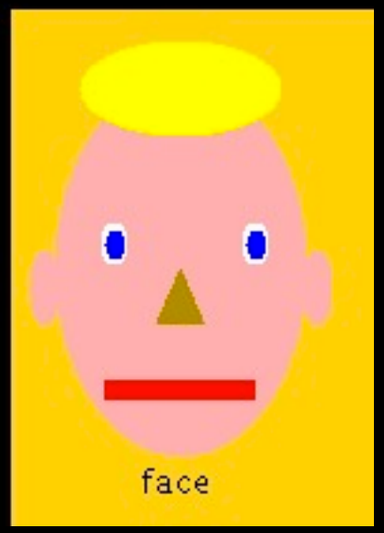
Suppose that this is the result of the parent class, Face:



The Face class has a number of methods such as drawMouth() and eyeColor().



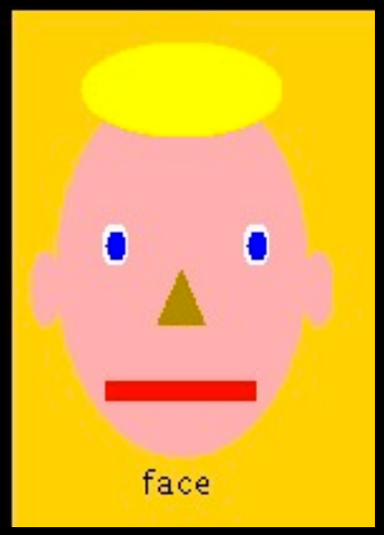
To get through this lab quickly, it's best to review the contents of Face.java.

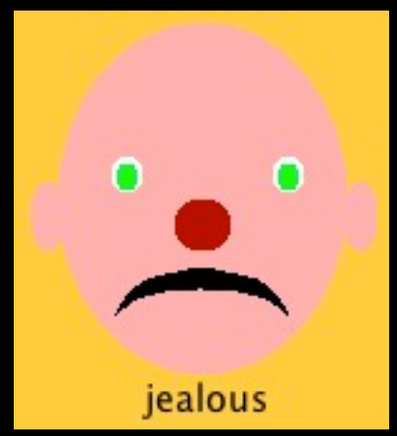


Go do it right now.

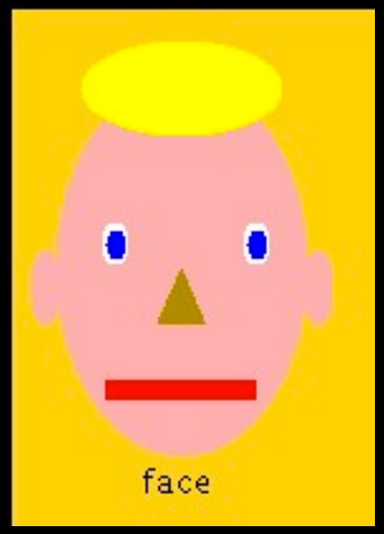
I'll wait.

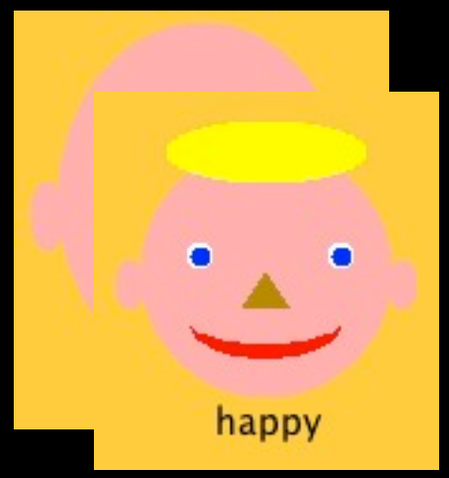
The JealousFace class is a child of the Face parent class.





But to get green eyes, a red nose, and a black sad mouth will require overriding certain methods.





Similarly, the HappyFace class will need override certain methods to make a happy mouth.

FOR EXAMPLE:

```
// HappyFace extends Face, that is, HappyFace is a child class
// of the Face class.
public class HappyFace extends Face
{
    public HappyFace(int w, int h)
    {
        super(w, h);
    }

   // Override the way the mouth is created.
    protected Polygon2 createMouth()
    {
       // STUDENTS WRITE STUFF HERE.
    }

   // Override the way the mouth is drawn.
    protected void drawMouth(Graphics g)
    {
       // STUDENTS WRITE STUFF HERE.
    }


}
```

For almost all checkpoints, the solution is a 2-5 line method that overrides a method in the Face class.

For almost all checkpoints, the solution is a 2-5 line method that overrides a method in the Face class.

If you are copy-pasting code, or re-writing existing code,

you are doing it wrong.

You are supplied with some helper functions to assist with writing less code.

In Face.java

```
    ////////////////////////////////////////////////////////
    // distX - computes a horizontal pixel-distance, as a percentage of
    //         the width of the head
    //
    protected int distX(double xv) {
        return (int)Math.round(xv*width/100.0);
    }


    ////////////////////////////////////////////////////////
    // distY - computes a vertical pixel-distance, as a percentage of
    //         the height of the head
    //
    protected int distY(double yv) {
        return (int)Math.round(yv*height/100.0);
    }

    ////////////////////////////////////////////////////////
    // pixelY - computes a pixel location that is the given percentage
    //   of the way down from the top of the head
    //
    protected int pixelY(double yv) {
        return (int)Math.round(y + yv*height/100.0);
    }

    ////////////////////////////////////////////////////////
    // pixelX - computes a pixel location that is the given percentage
    //   of the way across the head (from the left)
    protected int pixelX(double xv) {
          // compute/return the appropriate value
        return (int)Math.round(x + xv*width/100.0);
    }
```

There are four methods to help you locate elements on a given face.

```
// Draw a rectangle that is 8% into the left side of the face,
// 80% from the top of the face, and is 10% the width of
// the face and 3% the height of the face.
g.fillRect(pixelX(8), pixelY(80), distX(10), distY(3));
```

In Polygon2.java

```
// Create a polygon that is a one-quarter moon.
Polygon2 p = new Polygon2(25);

// Reassign polygon p a one-quarter moon that fits
// in the given bounding box on the face.
p = p.fitIn(pixelX(40), pixelY(20), distX(20), distY(40))

// Reassign polygon p a 60 degree clockwise rotation
// of its previous self.
p = p.rotateBy(60);
```

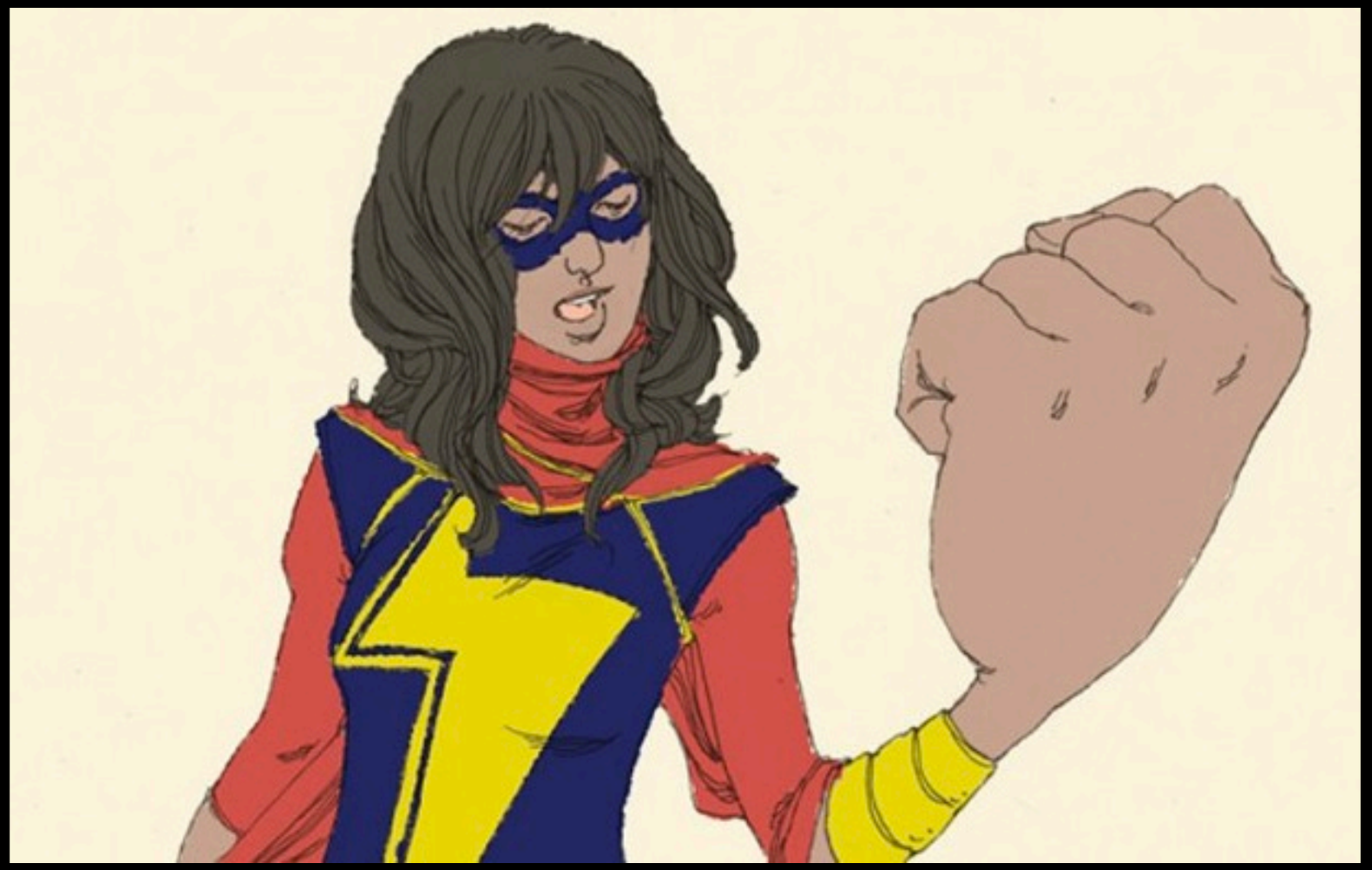


Image credit: Marvel

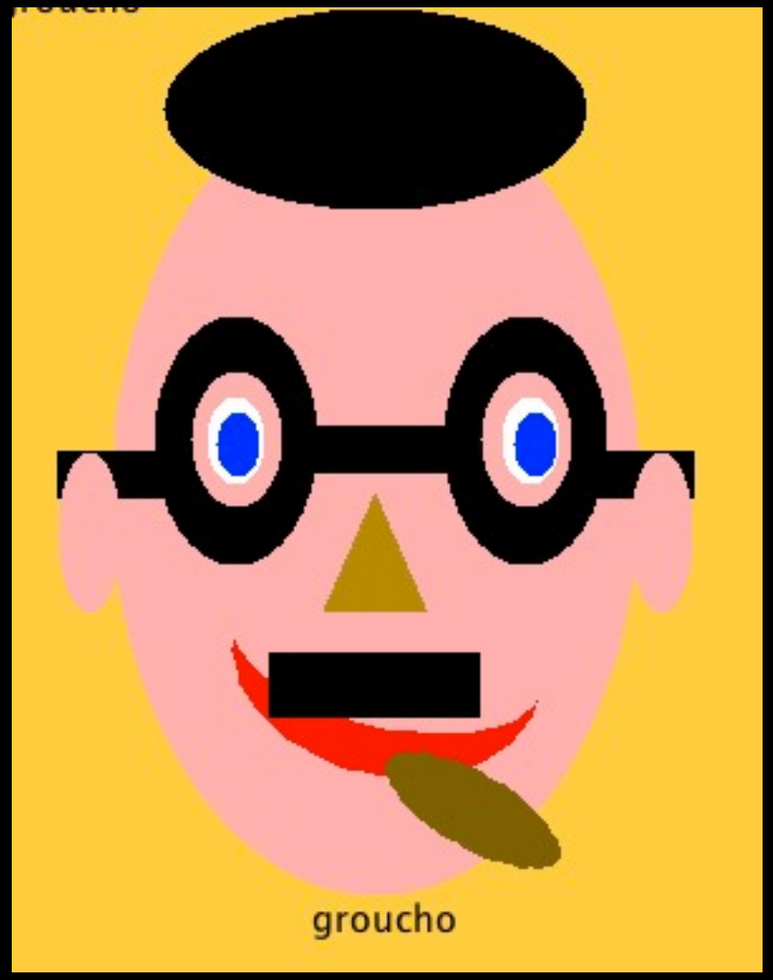
A note on using super

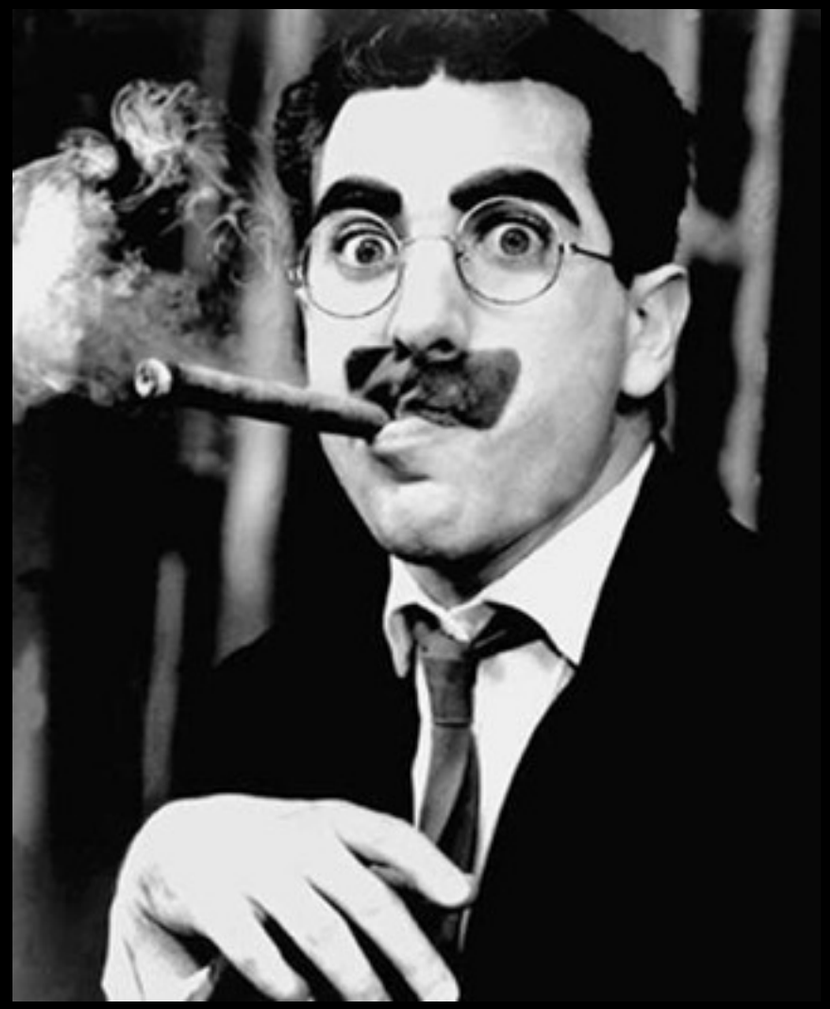
super( ) calls the parent constructor, for example:

```
public class HappyFace extends Face {
    public HappyFace(int w, int h) {
        super(w, h);
    }
```

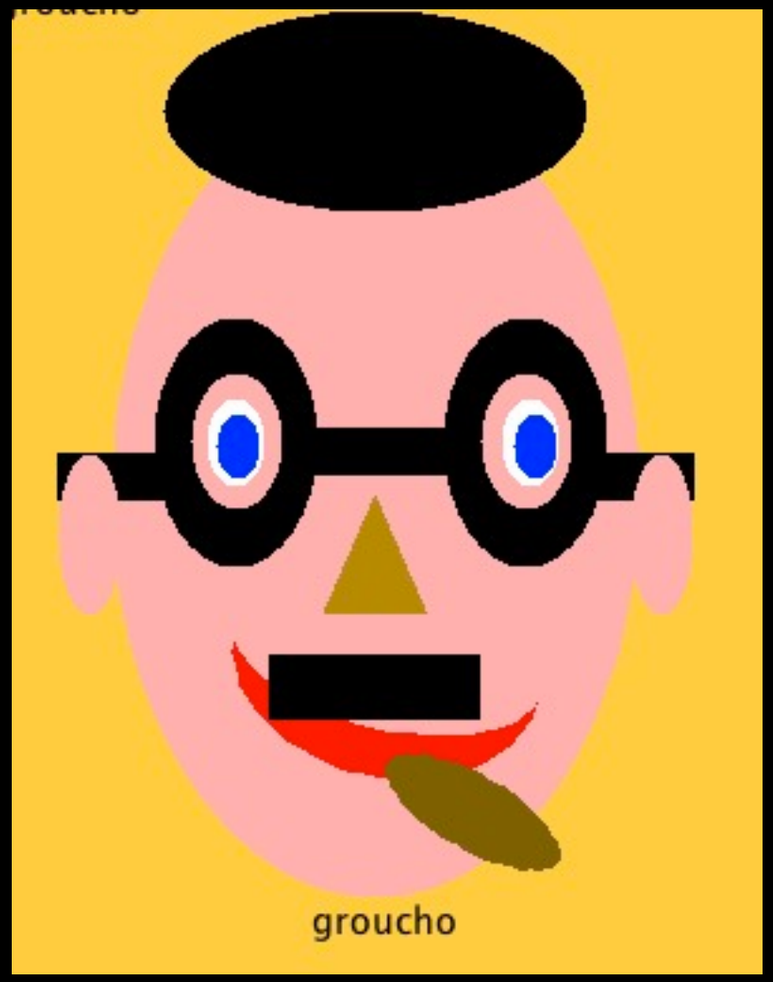
Conversely, super.method()
calls a method on the parent

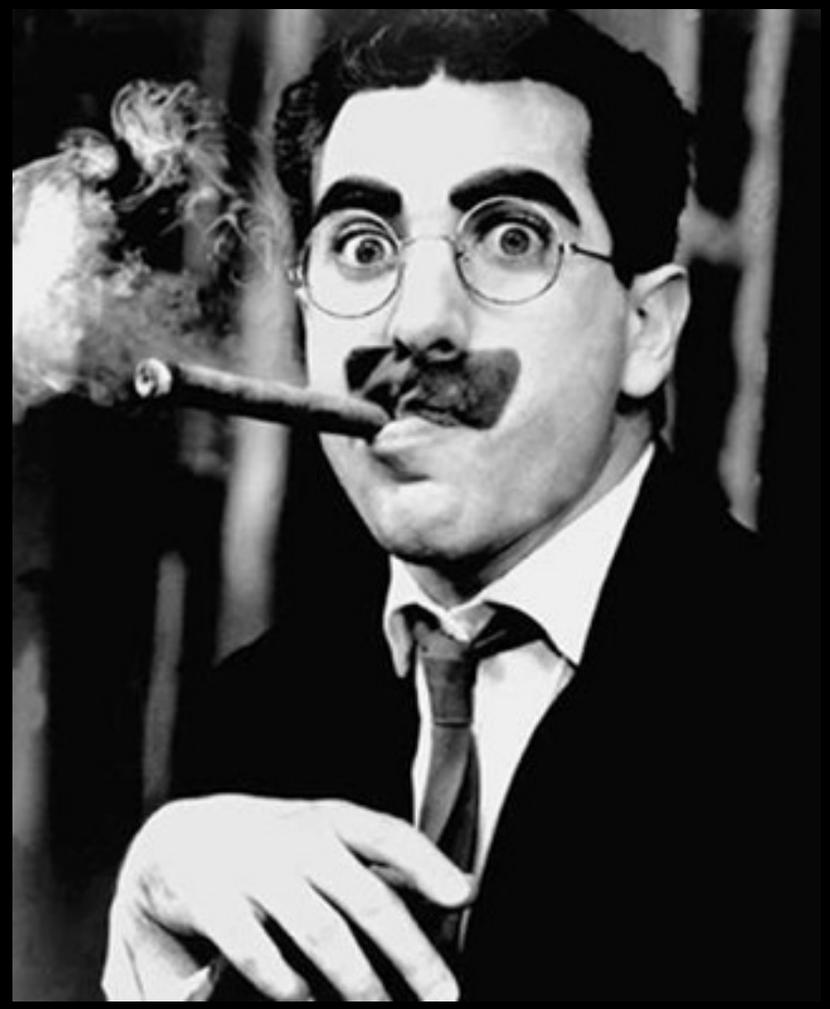
(and, in case you were wondering, Groucho Marx was a popular comedian of black and white TV time)



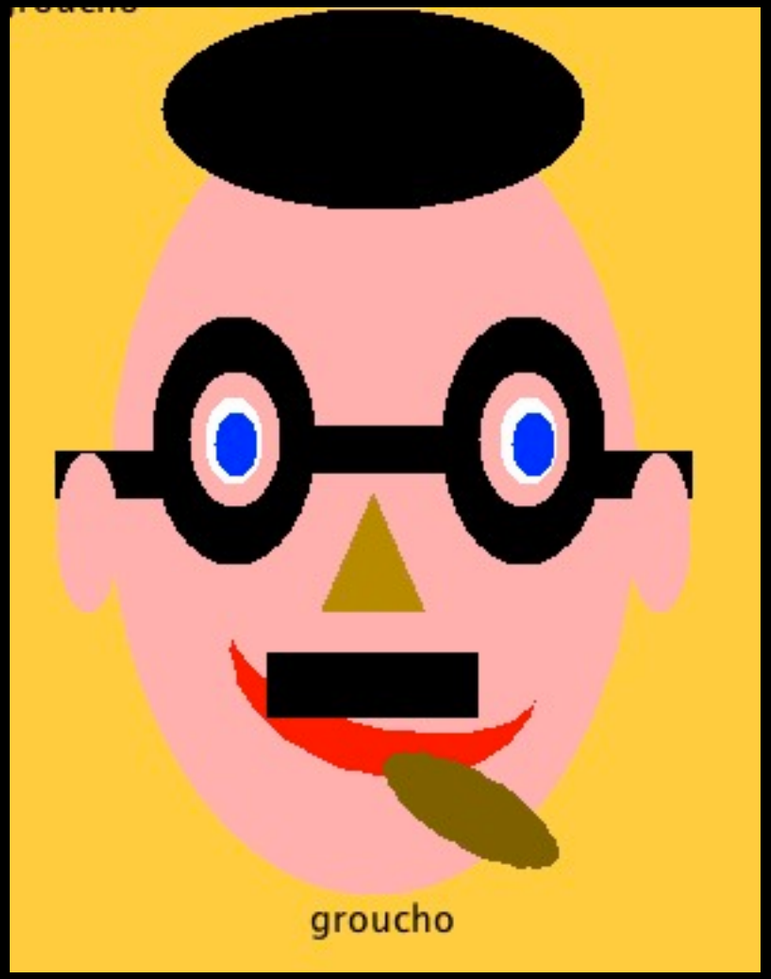


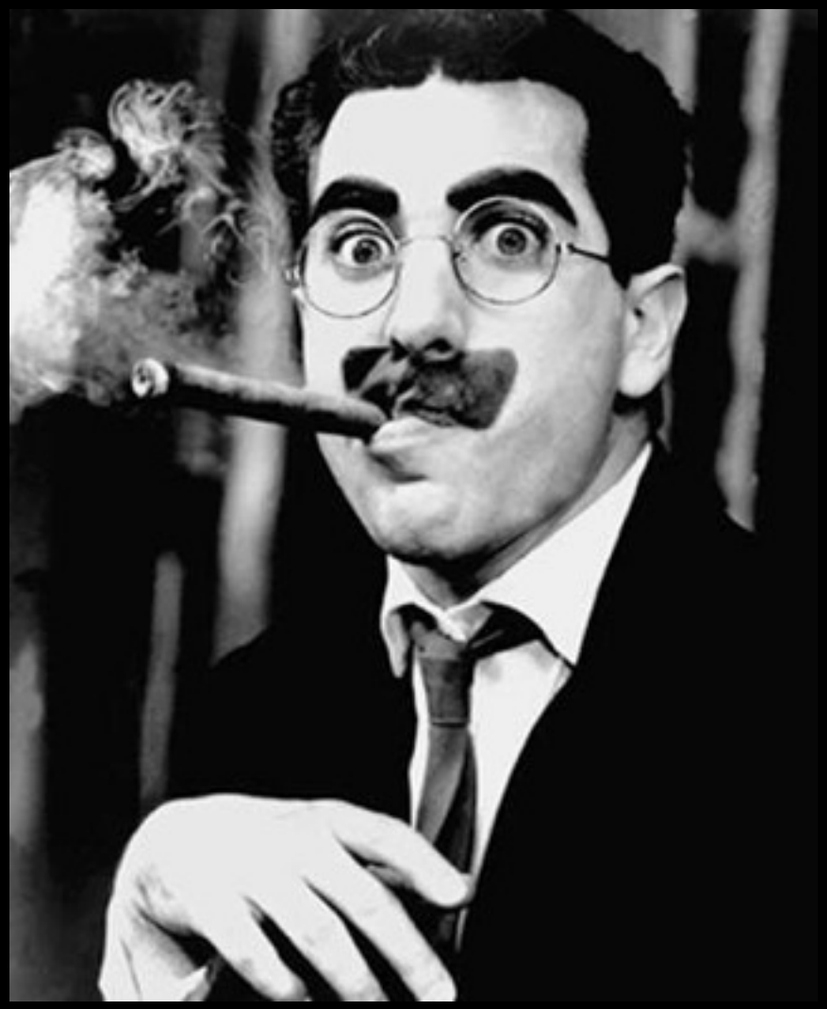
(notice that he has glasses, not a monocle)



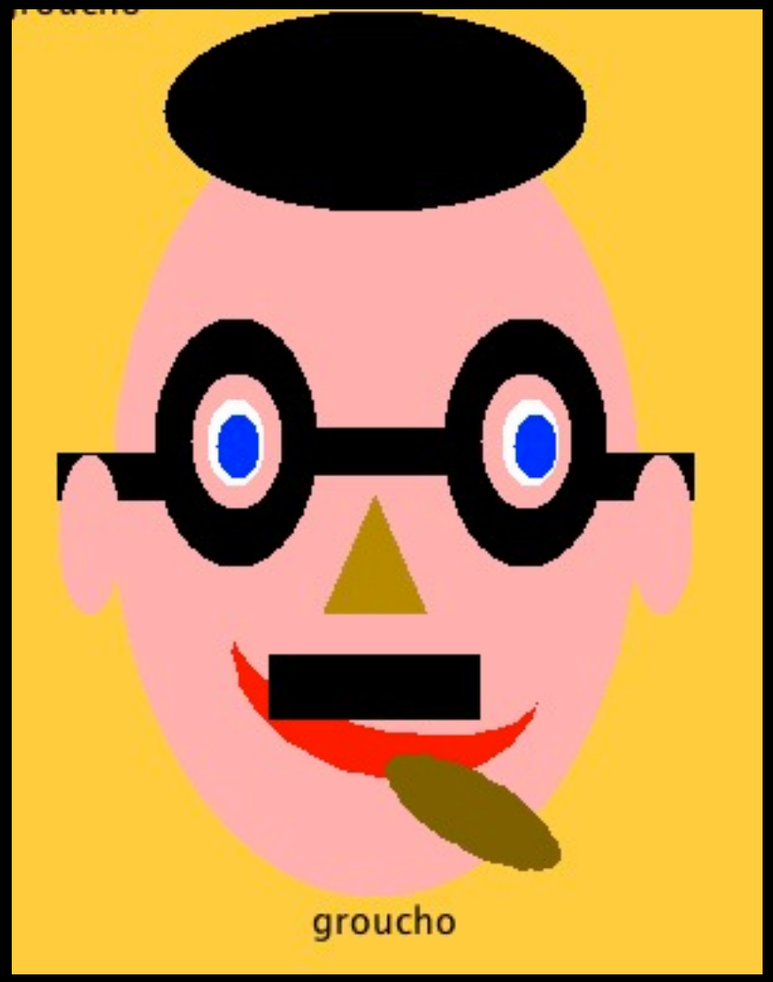


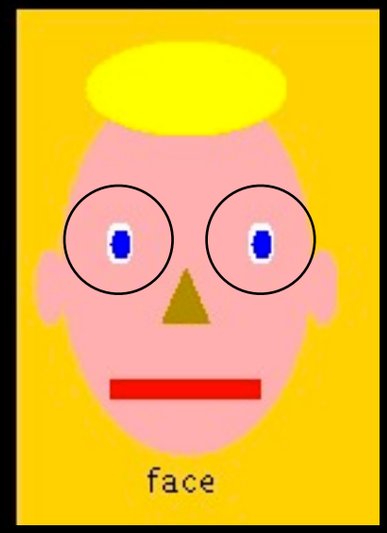
(notice that his glasses go behind his ears)





This **will** earn checkpoint 6.





This will **not** earn checkpoint 6.

Good Luck!

If you have any questions the TAs and I are happy to help.