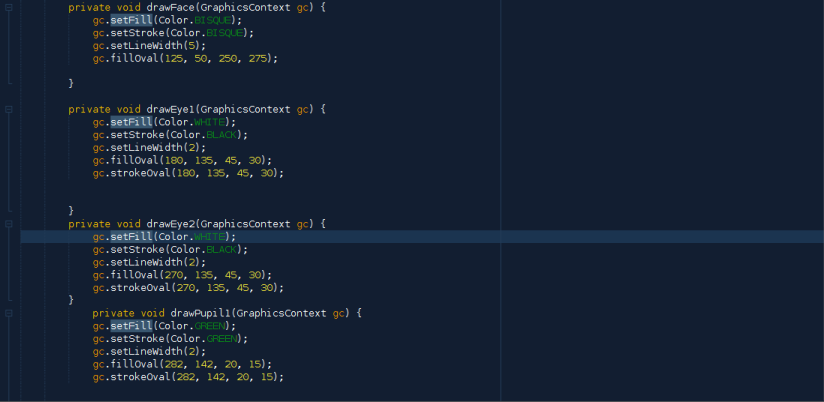
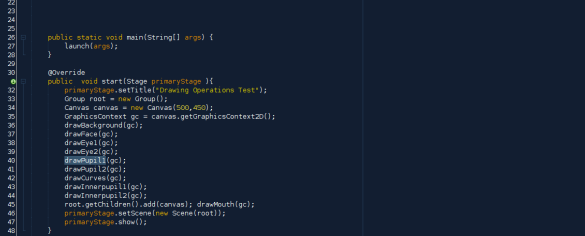
Essay for Challenge1 Project

by Alexander Johnson

this essay is a description of what types of code and methods i used for my “*challenge1*” project for my Computer Science Pre-AP class. My project’s goal is to create art on a digital canvas using methods. The first step i took to creating art was deciding what i should make . I eventually decided on creating a portrait of a fictional person , although making a face is rather complicated when using code to make art. I then started coding , I began by using the *private void* class to create a bunch of different sections and made the corresponding code in the registry . I then filled specific classes with *drawShape* which allowed me to declare my method. in another one i created a registry where I could store my shapes , i then specified which shapes would become the face and which ones would be the features as you can see here :



as you can see i used mostly colored Ovals to create my face , infact all of my methods had the code *gc.setfill(Color.\*\*\*\*\*)* and *gc.setfill(Color.\*\*\*\*\*)* which allowed me to choose the color that the ovals and other shapes were , both the inside and the actual line. another necessity in the code was the

*gc.fillOval(x , y , q , w , )* with the *x* and *y* being the coordinates where the shape is located and *q* and *w* being the actual size of the shape , by length and height. another portion of the code stores the names of my draw methods 

Near the top of the program there is a method called *public static void main*

which is the method from which all other methods are connected , literally!

*void main* allows my methods to run by giving the green light to each method allowing the it run if it works , though it does in a certain order . If you look under section of code titled *public void start* it shows the size of the canvas and the order by which methods are ran , some examples are *drawFace(gc); and drawEye1(gc);* if you notice the order of the code it shows that *Face* comes before *Eye* and so in the actual picture the eye will show up above the face instead of being under it and not visible.

So, my program creates a face that uses many different methods and pieces of code. as you could see in the previous pictures.