

COMPUTER GRAPHICS

Student names and ids: Andreas Jensenius (s093199) & Jannick (s09)

Title : Final project

1 Exercise 1

1.1 Part 1

1.1.1 A

The function `display` is the display function used by OpenGL, as defined by `glutDisplayFunc(display)`, and is called continuously. It defines which shaders to use, binds the VAO, draws the triangle and handles the swapping of buffers.

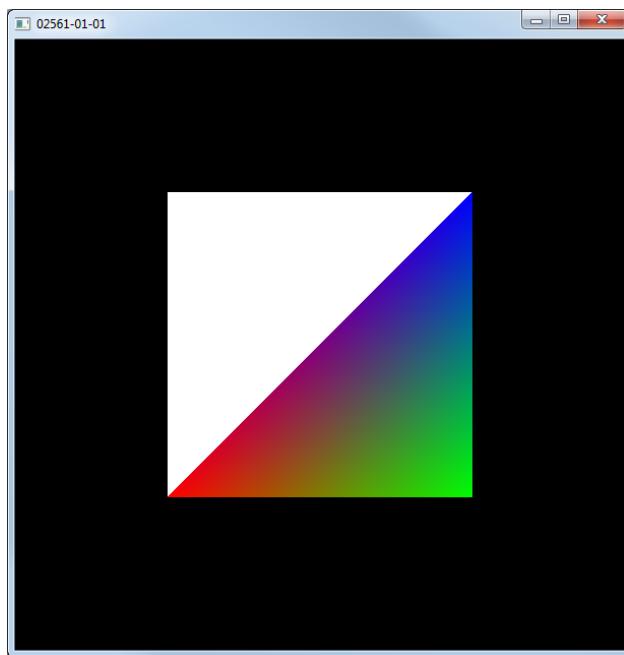
The function `reshape` handles the resizing of the window.

The function `loadBufferData` sets up the buffers and attaches the VAO.

The function `loadShader` sets up the shader and sets up the transfer of data to the shader.

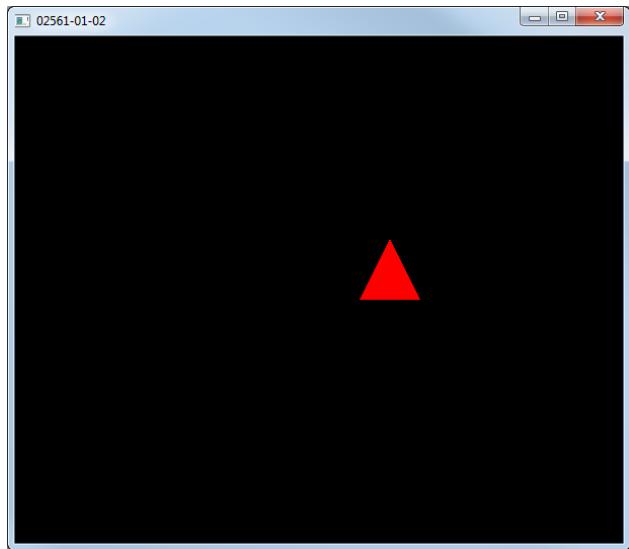
The function `main` is the one which is run when executing the program. It sets up OpenGL/glut.

1.1.2 B

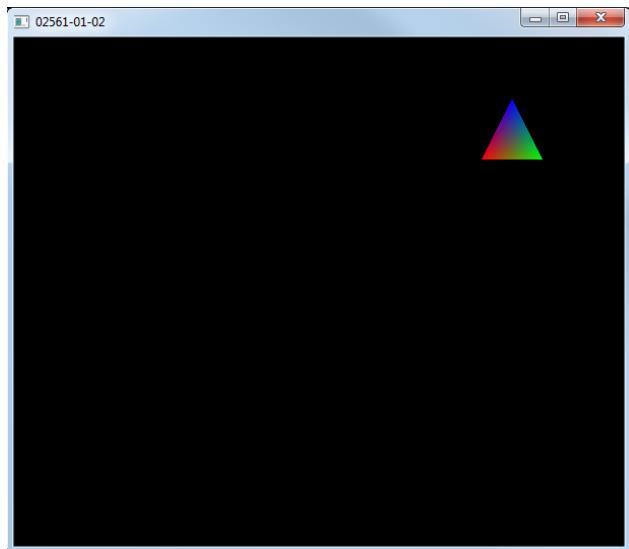


1.2 Part 2

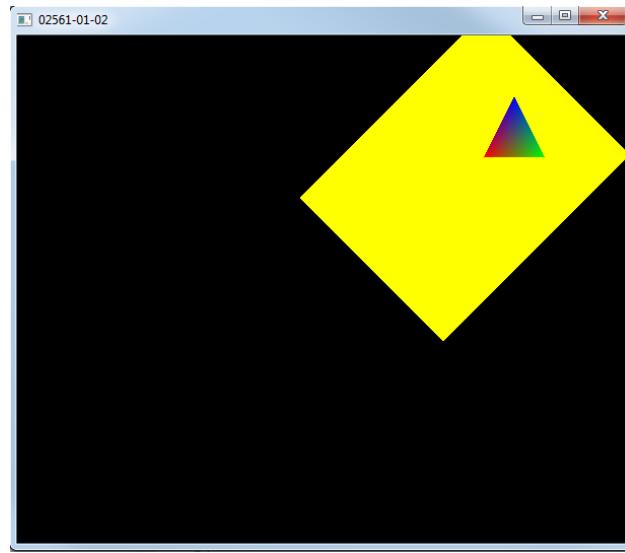
1.2.1 B



1.2.2 C



1.2.3 D



1.3 Part 3

1.3.1 A

`glDrawElements` uses the given indices within to specify in which order the points should be used, instead of using them from 0 to end as `glDrawArrays` does.

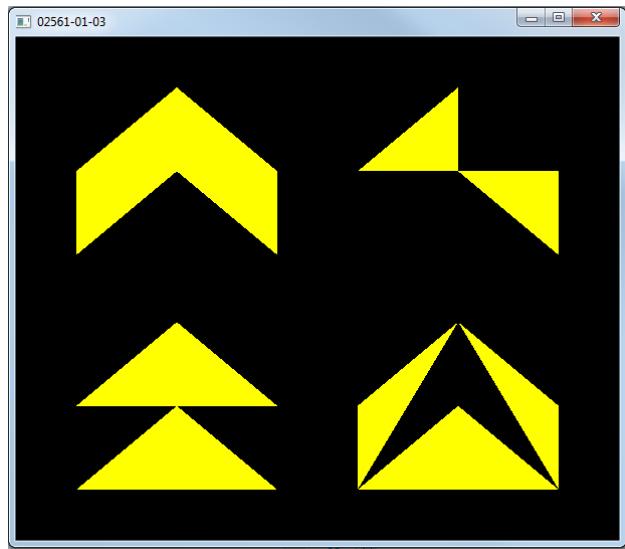
1.3.2 B

GL_TRIANGLES simply forms triangles from three points. Given 6 points it would create 2 triangles.

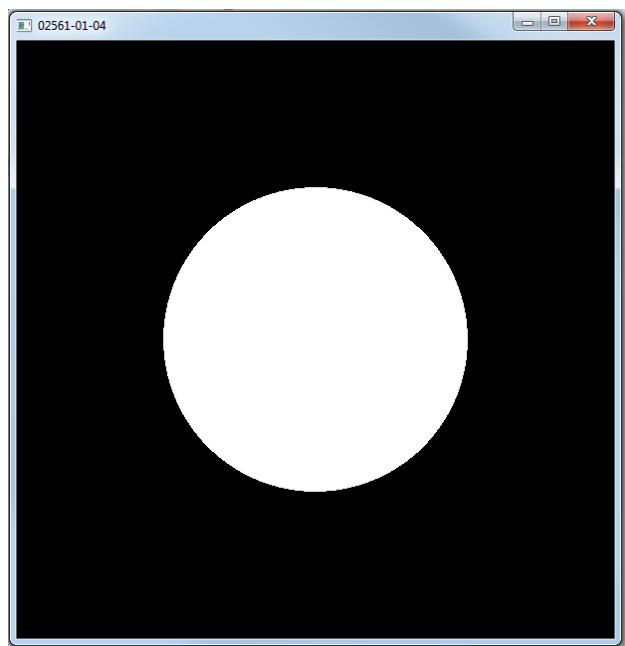
GL_TRIANGLE_STRIP creates connected triangles, in which the two last points of the previous triangle are used as the two first point of the next triangle.

GL_TRIANGLE_FAN uses the first point given for every triangle it draws. Given 3 points it draws a triangle. For each new point given it creates adds a new connected triangle, using the newly added point, the first point given, and the previously added point.

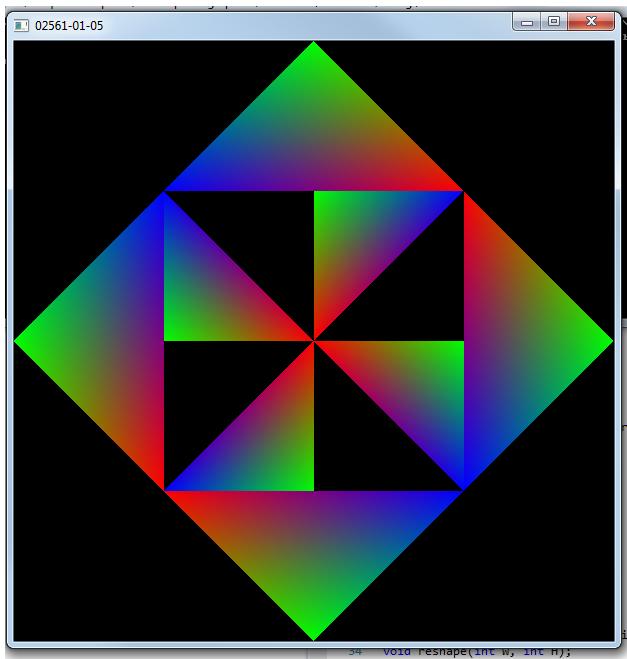
1.3.3 C



1.4 Part 4



1.5 Part 5



2 Exercise 2

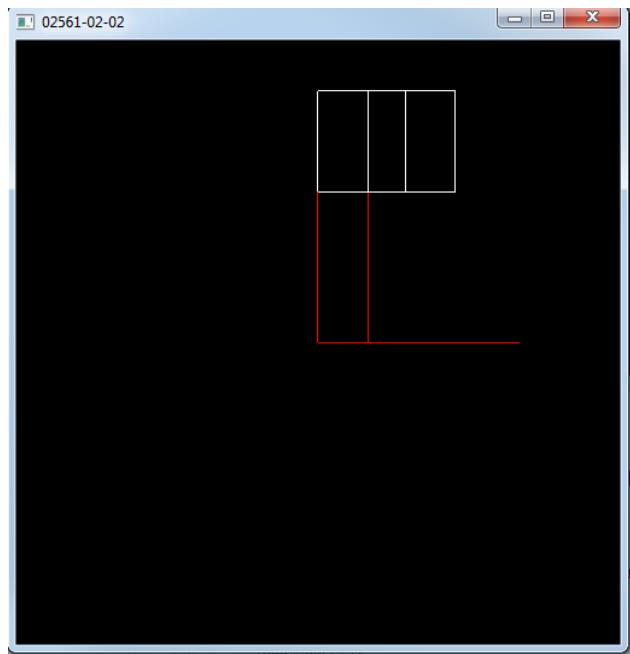
2.1 Part 1

The main difference between the two `display` functions is that this one defines "the camera" by creating the modelview.

The difference in the `vertex` struct is that the old struct handles both color and 2D position, whereas the one in exercise 2 handles position in 3D and not color.

2.2 Part 2

2.2.1 A



2.2.2 B

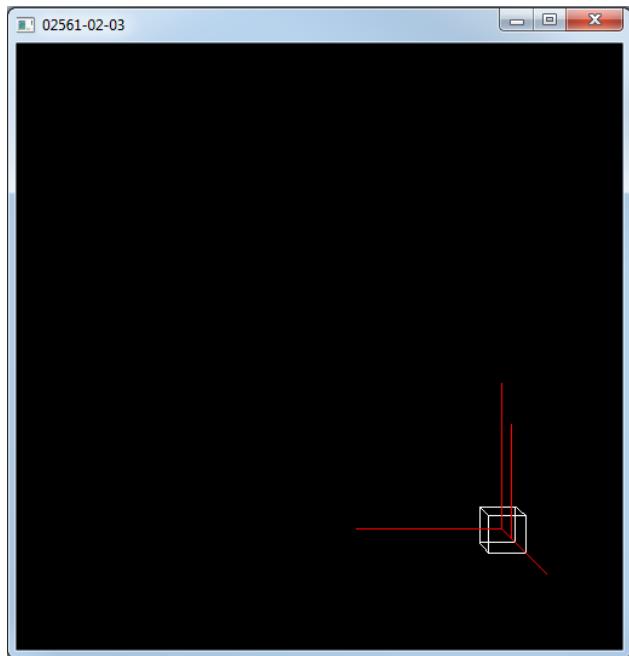
Translate(0,3,0), Scale(2) and RotateY(30) was used.

2.2.3 C

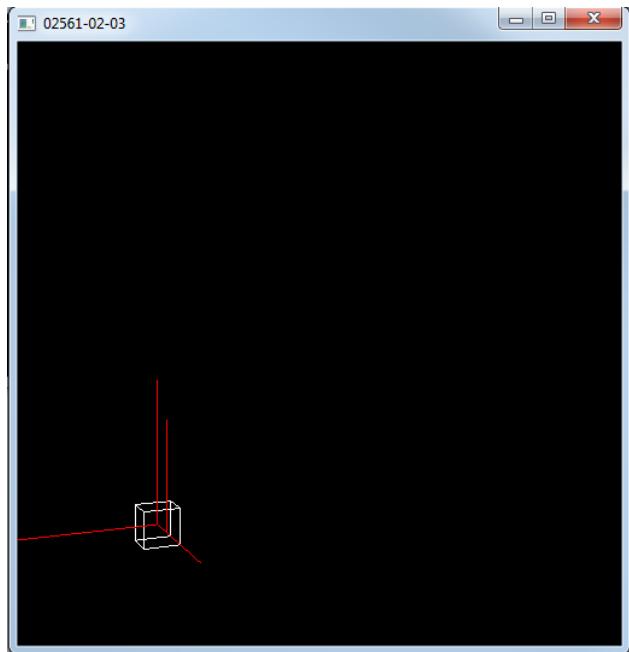
First Translate, then Scale and finally RotateY.

2.3 Part 3

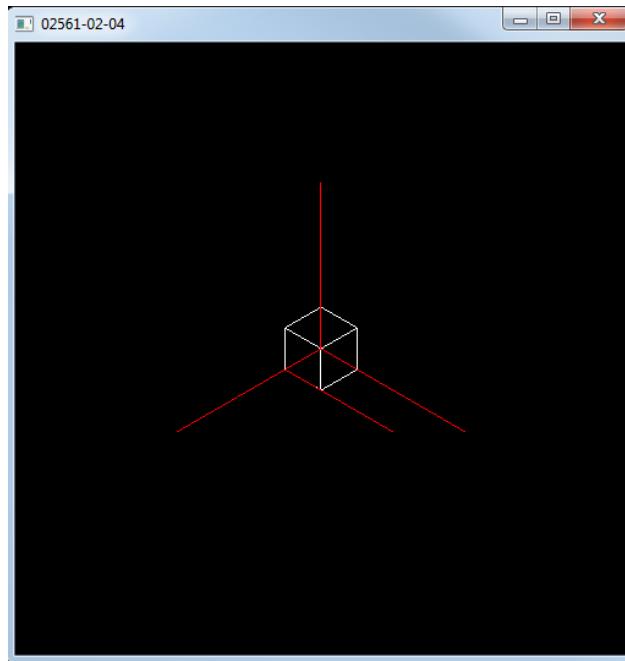
2.3.1 A



2.3.2 B

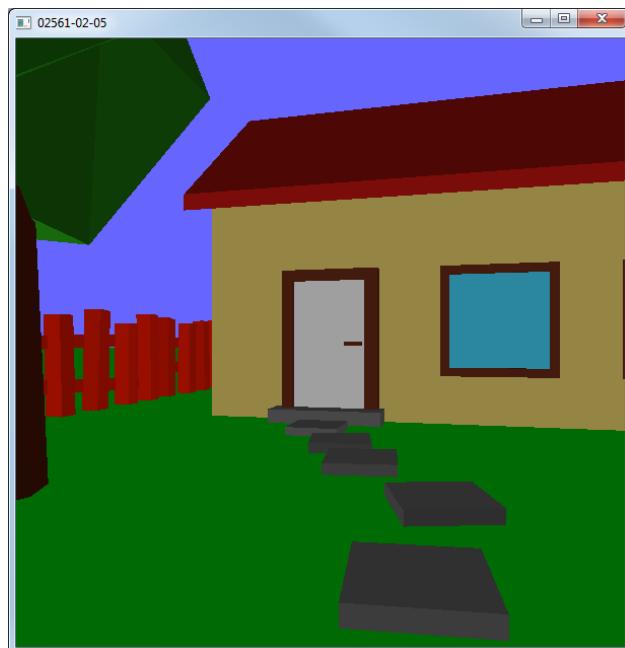


2.4 Part 4



2.5 Part 5

2.5.1 key 2



2.5.2 key 4



2.5.3 key 6



2.6 Part 6

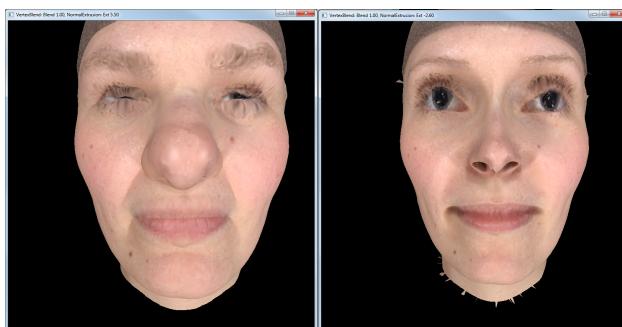
!!!! INDSÆT FORKLARING HER !!!!

3 Exercise 3

3.1 Part 1



3.2 Part 2

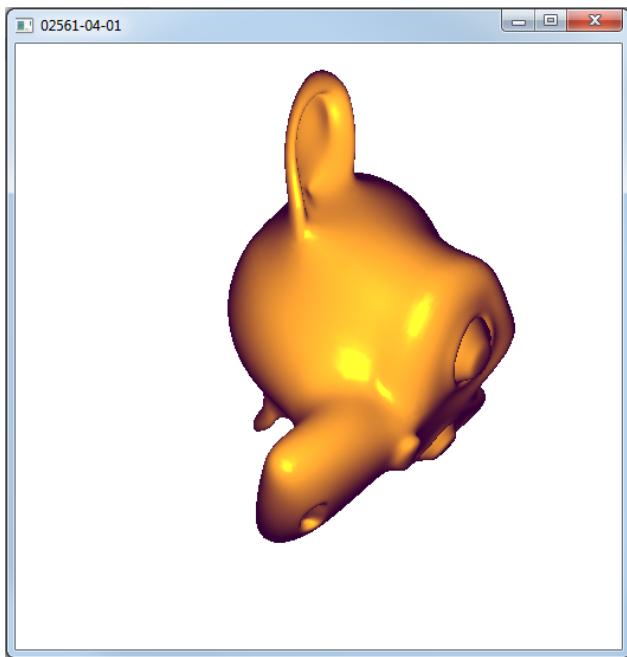


3.3 Part 3

!!!! INDSÆT FORKLARING HER !!!!

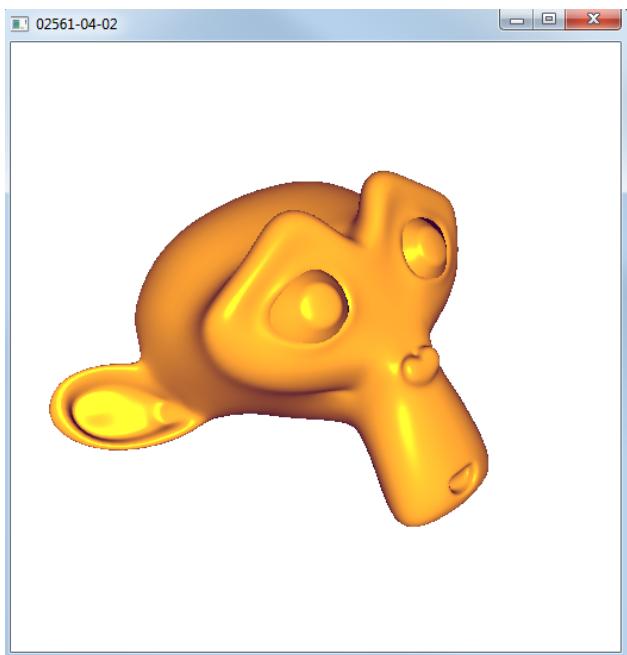
4 Exercise 4

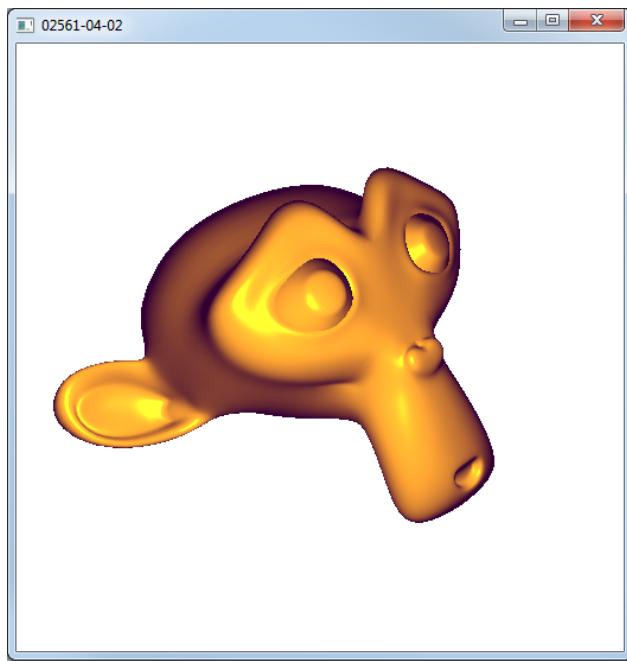
4.1 Part 1



4.2 Part 2

First is point light, second is directional.





4.3 Part 3

4.3.1 A

!!!! INDSÆT FORKLARING HER !!!!

4.3.2 B

Gouraud shading is applied on a vertex level, while Phong shading is applied on a per-fragment level. Gouraud is cheaper but makes the model look more rough, and phong is more expensive but makes it look more smooth.

4.3.3 C

Using point light, the light originates from a single point with a given direction, while directional lighting does not have a specific origin. In directional light all the light rays are parallel, while using point light the light rays starts from the same origin and spread out.

4.3.4 D

Yes.

4.3.5 E

It simply removes the specular component.

4.3.6 F

It makes it look more shiny, like polished metal, by increasing the intensity of specular lighting.

4.3.7 G

We made no simplifications.

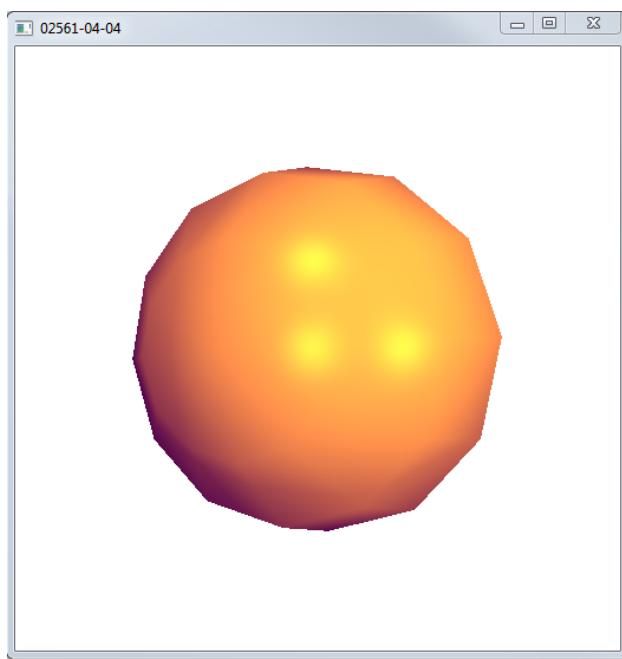
4.3.8 H

It is used to convert to normal vector into eye space. You obtain the normal matrix by taking the transpose, inverse of the modelview matrix.

4.3.9 I

Eye-space.

4.4 Part 4



5 5

5.1 Part 1

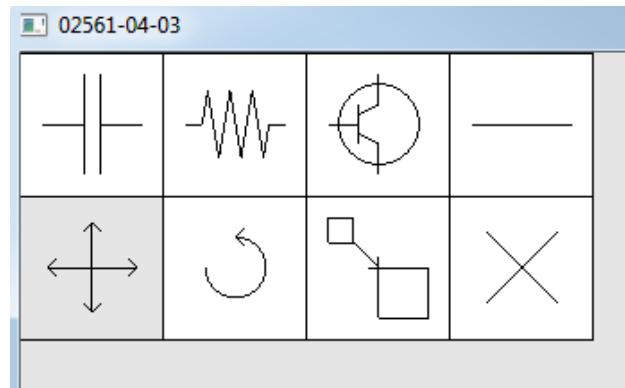
Pick select which object to draw once you click outside the colored boxes.

5.2 Part 2

When the user clicks on a square the uniform variable `colorUniform` is set and the `frameBufferObject` is bound. `renderScene(true)` generates the color and stores, and we then retrieve it in `getId` decoded from a color. The value of the selected index on the board is updated to one of the three possible colors, and we ask to have the `display` function run again, so that `renderScene(false)` can draw our changes

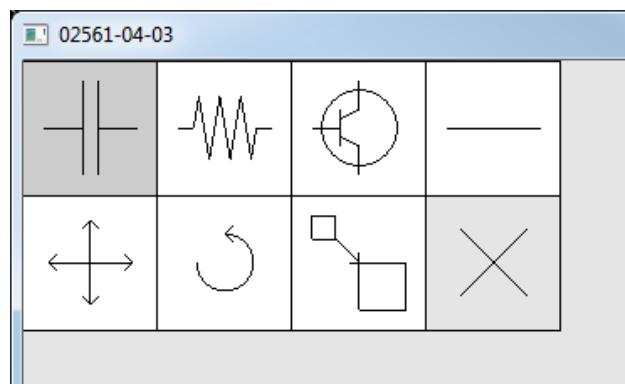
5.3 Part 3

5.3.1 A

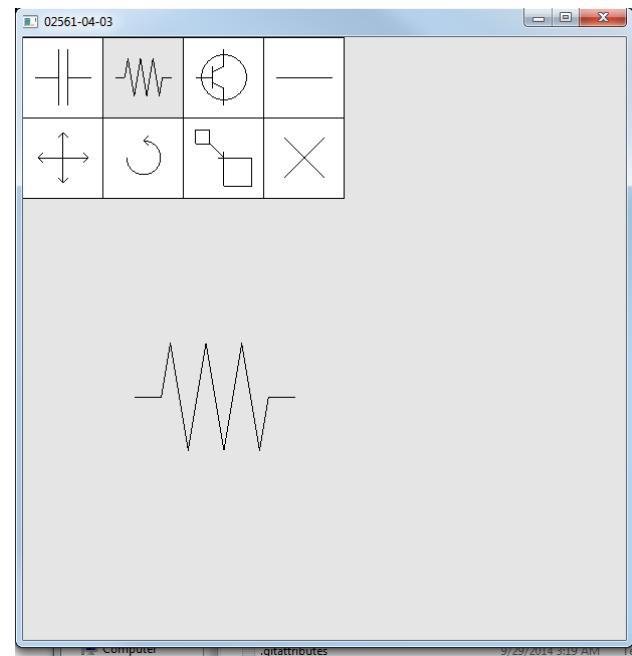


5.3.2 B

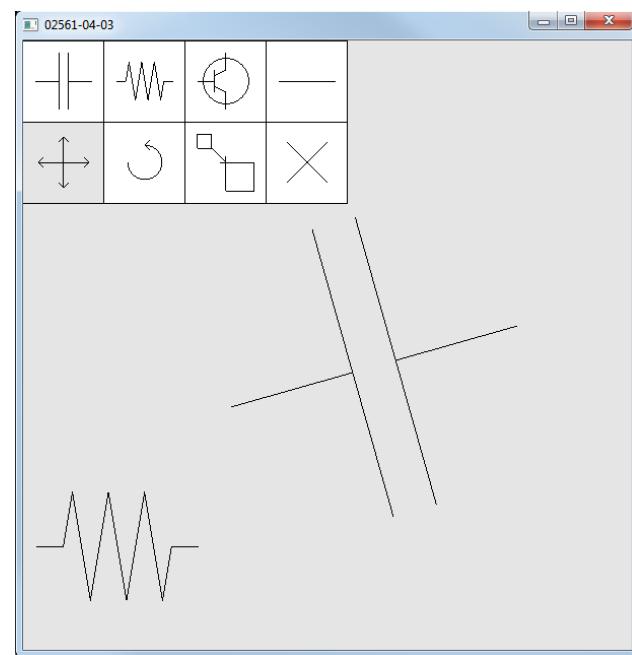
When doing screenshots, the cursor is not captured in the image so it is not shown on the image below.



5.3.3 C



5.3.4 D



5.3.5 E

