Computer Graphics - 2016/2017

Exercise 1 - Part A

This exercise has two parts.

In part 1 (this part), you will use the basic drawing functionality of OpenGL, by drawing a circle and changing its properties.

You have to submit part 1, but it will be graded together with part B.

Note: You must submit this exercise (and all other exercises) in pairs.

General guidelines:

(You might lose points for not following these guidelines)

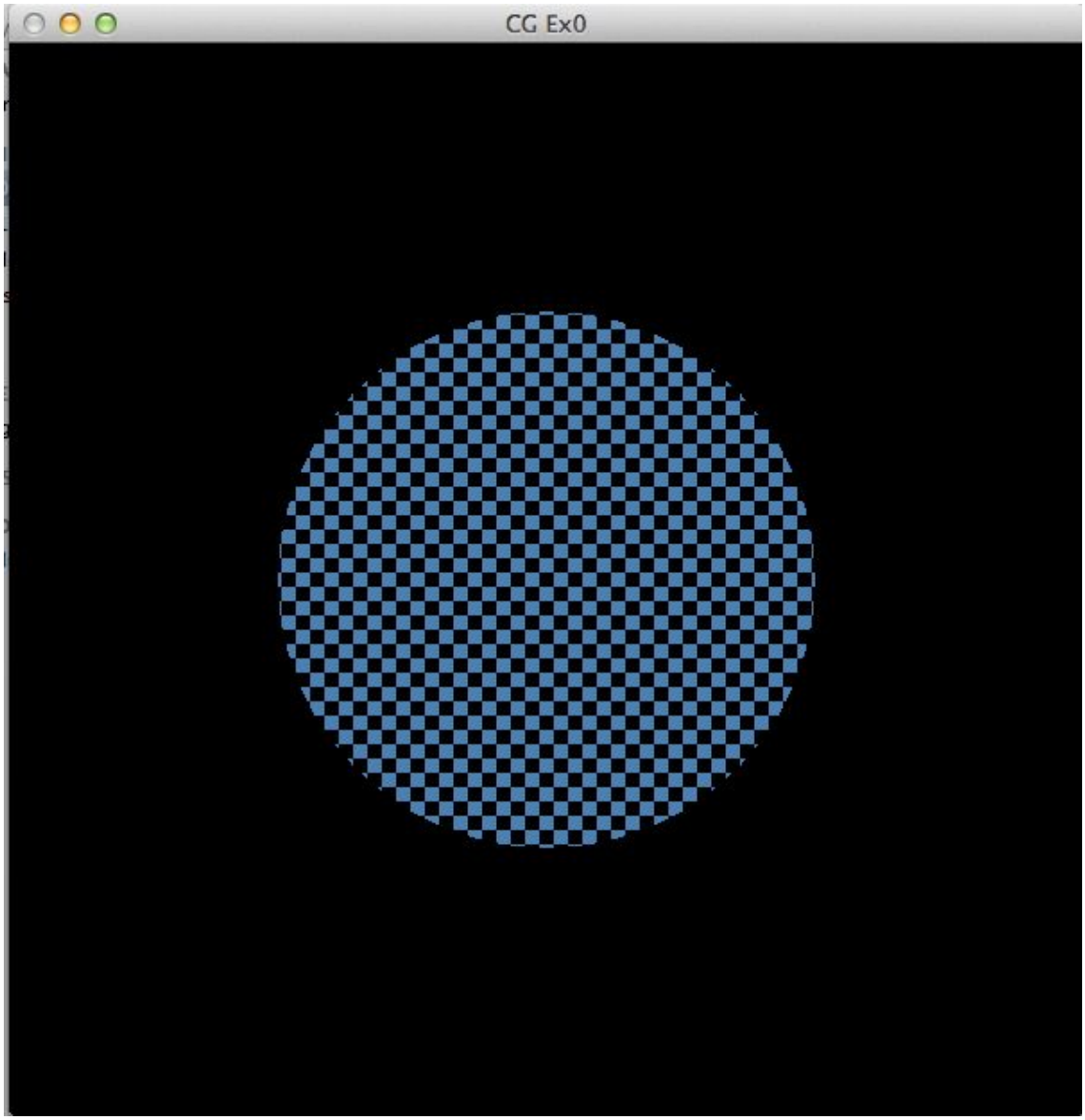
- Make sure that you understand the effect of every character that you write in your code.
- Make sure that your code does what it's supposed to do.
- Do not unnecessarily change existing code.
- Keep your code readable and clean!
    - Avoid code duplication.
    - Comment non-trivial code regions.
    - Block together (using '{' and '}') logically related lines of code.
    - Preserve coding conventions when changing existing code.
- Keep your code efficient. In particular:
    - Do not unnecessarily transfer data to OpenGL (e.g. share vertices between triangles and similar shapes).
    - Do not unnecessarily allocate memory.
    - Efficiently divide the work between OpenGL and the hosting program.
    - Minimize the number of vertices being processed by OpenGL while preserving the visual quality your drawing.
- Add to your Readme.txt a list of all web-pages URLs that you used in order to complete this exercise.
- Add to your Readme.txt a list of all students' usernames that you discussed this exercise with.

0. Initial Compilation

Download the example code from here, unzip, compile (use `make`) and run it. If all went well, you should see a window with a triangle drawn in it.

1. Drawing a circle

- Change the example code to draw a filled circle instead of a triangle.
- You are expected to figure out by yourself how to compose a circle from triangles.
- The circle should be located at the center of the window with a fixed radius w.r.t. the window size. At this part of the exercise you don't have to handle the case of window resizing, but in the next part you do.
- Write a new fragment shader and use it to fill the circle with a checkerboard pattern as shown below:



FAQ

Q: Should we preserve the proportion of the circle when the window aspect ratio is changed?
A: In this part of the exercise you don't have to, but it will be required in the next part (ex1b).

Q: Should the size of the pattern should be preserved when the window size is changed?
A: It's up to you.

Q: Is there a school solution?
A: There is no school solution for this part of the exercise. It should be easy and straightforward. We will publish a school solution for the next part.

Submission

Include the following in your submission:

1. A **Readme.txt** file (not README or Readme as in some other courses), that includes:
   - your id and login
   - your partner's id and login
   - A brief description of your implementation and the changes you made in the example code
   - A brief description of each file of the other files that you submit.
2. All files that are required for compilation of your solution with a single 'make' command, and the shaders necessary to run it.

Pack all files as a single zip file named by the following pattern: ex0_<your 9 digits id>_<your_username>_<your partner's 9 digits id>_<your partner's 9 digis login>.zip (e.g. '`ex0_123456789_mylogin_987654321_myfriendlogin.zip`').
The first login should be the one of the user who submit the file.

Deadline:
You have to submit your solution (via the course's moodle webpage) no later than Tuesday 22/11/2016 at 23:55.

Late submission will result in 2^(N+1) points deduction where N is the number of days between the deadline and your submission (rounded up, the minimum grade is 0, friday and saturday are excluded).