CS460 Computer Graphics

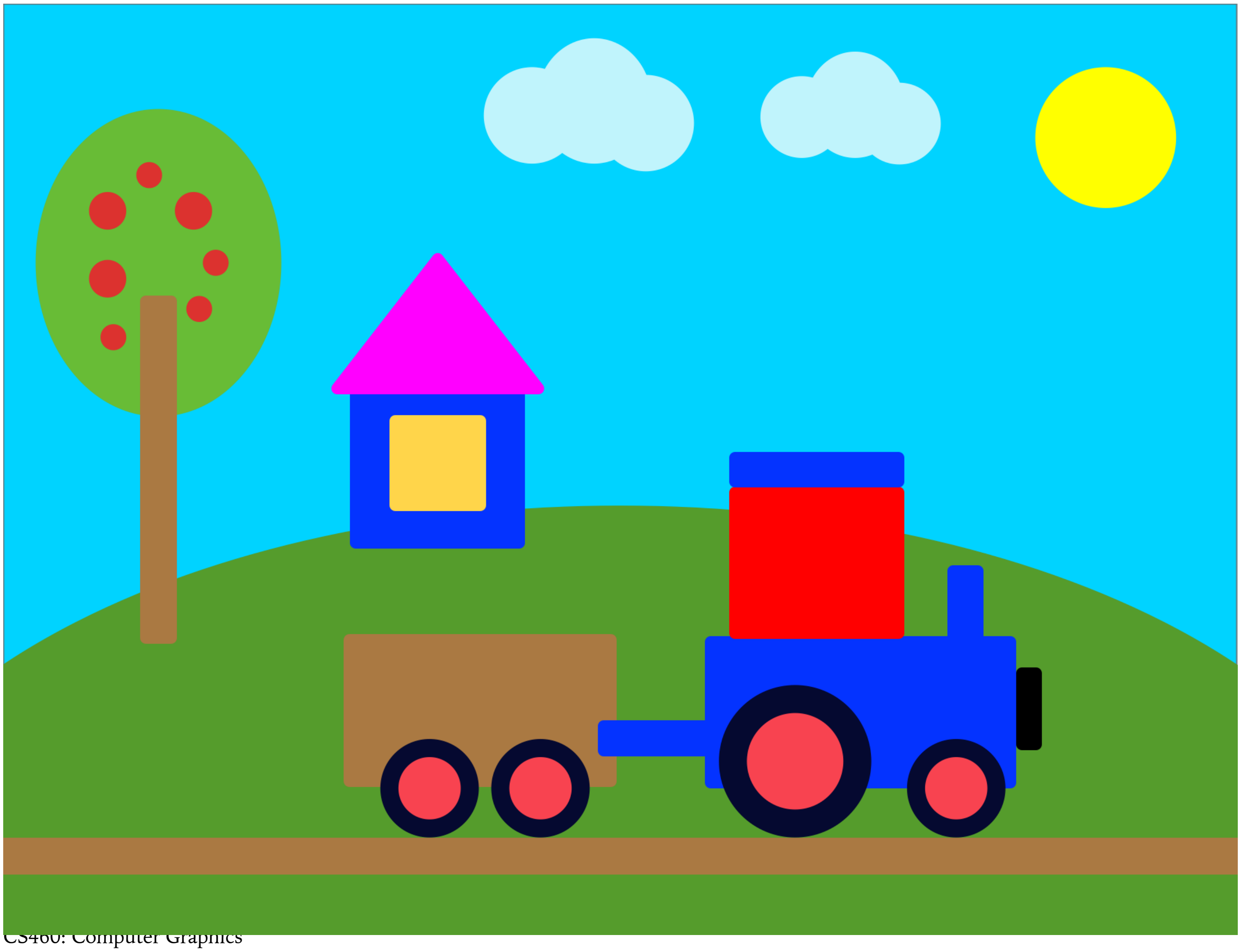
**Assignment 1: Basic Shapes Drawing. Total points 30**
**Deadline: 20 Sep, Monday**

**Description:**

The purpose of this assignment is to give you exercises to use the basic geometric shapes to create a static drawing in an OpenGL canvas. The assignment assumes that you have studied the lectures and completed some of the example problems.

**Instructions:** Drawing a Landscape

You are given the following scene to draw. The dimension of the drawing canvas is 768 pixels width and 768 pixels height. We have provided a skeleton OpenGL project that you can use to jump start the assignment. We can draw the objects (for example) by using layers of basic graphics objects. The cloud is drawn with three circles of same colors placed closed together. The order of the drawing is important. That is, if we draw a black circle first and then draw a red circle, we can then come up with the train wheels as the second circle will overlap the first drawing.



Therefore, we should order them as the following

- Draw the background of the canvas as light blue (glClearColor)
- Draw a big circle to represent the landscape at the bottom. You can use the color green
- Draw other objects, such as the tree, train, sun, and the clouds.
- The following drawing is an approximation. Your drawing does not have to be an exact copy of this drawing. Through your drawing you need to demonstrate that you understand how to use graphics objects and place them on the canvas.
- For each compound drawing object (tree, clouds, home, train engine, and wagon) create separate class that implements the GShape interface. In this way, we can reduce code duplications and provide better code organization.
- The drawing of each object should be done in the respective render methods of that class.
- When initializing the drawing objects with vertex information, the students must use an array of floating points and take advantage of the glVertex2fv function (this has been demonstrated in the initialization, and drawing of the triangle object).
- After the drawing objects have been initialized, they must be added in the GShape ArrayList.

In addition to creating the compound class to draw the mentioned drawing objects, the students can use OpnGL translation functions to animate the drawing objects (for example, moving sun/clouds/train). When creating the scene, use a number of classes instead of adding all the drawing code in one class. For example, you can create a class triangle that would be responsible for drawing triangles. You would then create instances of this class in the initialization steps and put them in a collection (ArrayList). Further, in the display method you can then use the collection to draw all of the triangles. Similarly, you can create classes (data structure) for quads, circles, etc. In order to draw circle, you can use the provided render method in the GCircle class to draw a circle. We have provided you a set of classes that you can complete to use in your assignment.

| **Req Type** | **Marking** |
|---|---|
| **R.x** | Assignment gets 0 if any critical submission requirements (shown in red) are not met. |
| **A.x** | You lose 2 marks for each good practice requirements (shown in amber) not met. |
| **G.x** | You earn 2 marks for each design requirements (green) satisfied and well implemented; 1 mark if it's partly met or met but not well implemented; and 0 if it's not met. |

**Assignment Restrictions:**

Technology Restrictions: You need to use the JOGL library to complete this assignment. You only need to submit the classes (java files) in an archive. You are free to use any IDE (Eclipse, etc) to complete the assignment.

Marking: This assignment is based on 4 design requirements numbered **G.1…G.4** for a total of 30 points. Points are awarded, or deducted, based on itemized requirements as follows:

**Submission and Good Programming Practice Requirements**

The following requirements pertain to all your assignments regardless of what your application is supposed to do (i.e. regardless of the design requirements). These requirements are to ensure that your code is usable, readable, and maintainable.

R.0 UNIQUENESS REQUIREMENT: The solution and code you submit MUST be unique. That is, it cannot be a copy of, or be too similar to, someone else's code, or other code found elsewhere. You are, however, free to use any code posted on our course website as part of our assignment solution. [Assigment mark =0 if this requirement is not met.]

R.1 CODE SUBMISSION ORGANIZATION AND COMPILATION: You should submit all the code files and data files necessary to compile and run your app. The professor will execute your java files by setting up an eclipse project. If you compress your submission on Blackboard, you must use only .zip format (not .rar or .tar or others). However, students are allowed to write code on Windows, Linux, or MacOS. The code should be generic enough to be OS agnostic. Make sure that there is no compiler errors in the submitted project. [Assigment mark =0 if this requirement is not met.]

R.2 README FILE: Your submission MUST include a README.txt file telling the professor how to setup and run your app. The professor should not have to look into your code to figure out how to create the project by using the submitted Java files. Your README.txt MUST contain the following:

- Your name, student number and email address and if you are working with a partner then their name, student number and email address as well.
- Issues: List any issues that you want the instructor to be aware of when they are evaluating your submission. In particular, tell us what requirements you did not implement or that you know are not working correctly in the submitted code. Here you are giving us your own assessment of your submitted project. [Assigment mark =0 if this requirement is not met.]
- Version: JDK version, OS you tested on your code on.

A.3 VARIABLE AND FUNCTION NAMES: All of your variables and functions should have meaningful names that reflect their purpose. Don't follow the convention common in math courses where they say things like: "let x be the number of customers and let y be the number of products." Instead, call your variables **numberOfCustomers** or **numberOfProducts**. Your program should not have any variables called "x" unless there is a good reason for them to be called "x". (One exception: It's OK to call simple for-loop counters i, j and k etc. when the context is clear and VERY localized.) [Minus 5 marks from assignment if this requirement is not met.]

A.4 COMMENTS: Comments in your code must coincide with what the code actually does. It is a very common bug to modify code and forget to modify the comments and so you end up with comments that say one thing and code that actually does another. By the way, try not to over-comment your code but instead choose good variable names and function names that make the code more self-commenting. Don't be afraid to create local variables so that the variable name provides more clarity. [Minus 5 marks from assignment if this requirement is not met.]

A.5 CITATION REQUIREMENT: If you use code from other sources you should cite the source in comments that appear with the code. If the source is an Internet website then put the URL in the comments. You may use bits of code from outside sources but this may not form the complete solution you are handing in. You DON'T have to cite demo code we provide on the course web site or with tutorials and assignments, however that code should not be used for things you post publicly (like on GitHub). [Minus 5 marks from assignment if this requirement is not met.]

VERY IMPORTANT: Any sample code fragments provided may have bugs (although none are put there intentionally). You must be prepared to find errors in the requirements and sample code. Please report errors so they can be fixed and an assignment revision posted.

Here are the requirements of this assignment.

G.1: Scene Drawing: The drawing of your program should match (approximately) the drawing provided in this assignment. You are free to use additional shapes to improve the scene. [15 points]

G.2: Shape Classes: You should complete the data and methods for the classes of GHome, GTree, GCloud, GWagon, GTrain, GRectangle and others that you feel necessary. You can follow the provided classes (GTraingle, GCircle, etc) as a guide. [10 points]

G.3: Shape based Drawing Collection: You would create the drawing elements by using the created classes in G2 and create ArrayList to create collection of the drawing elements. [5 points]

**[Bonus]** G.4: Shape Transformation: Use glTranslate method in your code to move the train drawing object in the scene. Please note, this is not an assignment requirement. [5 points]

Please let me know if you have any questions. Thank you.