Computer Graphics

Homework 2

Due April 17th 2015, Thursday, 23:59

Homework guidelines

- Have fun, learn, be creative, and create something you are proud to show-off!
- No teamwork allowed! It is ok to discuss the projects with other students. However your code must be your own. (You would be surprised how easy it is to detect copied code!).
- Comment your code - you will be glad you did. You will be reusing code throughout the semester.
- You will demo some of your hws. In each demo, you will be asked to demonstrate the functionality of the program and possibly describe how some of the code works. Be prepared for this.
- Late projects will lose points:
    - 50% for up to 24 hours late.
    - 75% for up to 2 * 24 hours late.
    - 90% for up to 3 * 24 hours late
    - 100% after 3 * 24 hours late
- If you have **good reasons** that you cannot complete a project on time **and** you have written documentation, then we can make adjustments to due dates. However, you must notify us before the due date that you would like to discuss such an arrangement. Good reasons would be illness, family emergency, visiting a conference to present a paper, …

Upload your homework

- Please check if your software meets all requirements as follows
- Zip the project directory
- Upload through lms.sejong.ac.kr
- A written report file – this report file should include the following:
    - Instructions about how to navigate/use in your program, including the function of keyboards (e.g. x exits the program)
    - Special functions that you implemented
    - Other things you consider important
    - Specify the libraries or code that you used and did not write yourself
    - Screen shots

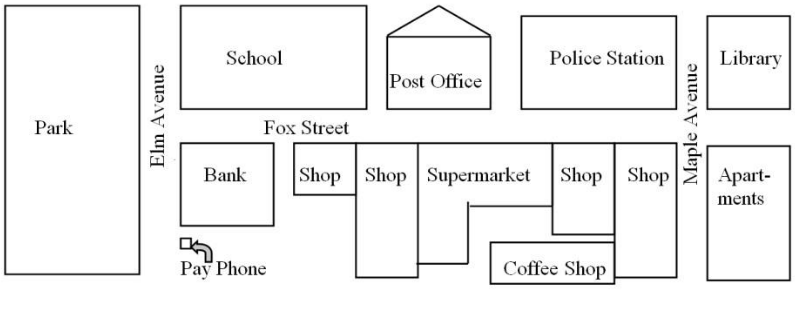
Homework

Purpose

This homework has been designed for you to complete the exercises that you tried in the classes. In this homework, you will learn how to draw 2D objects and control keyboard and mouse.

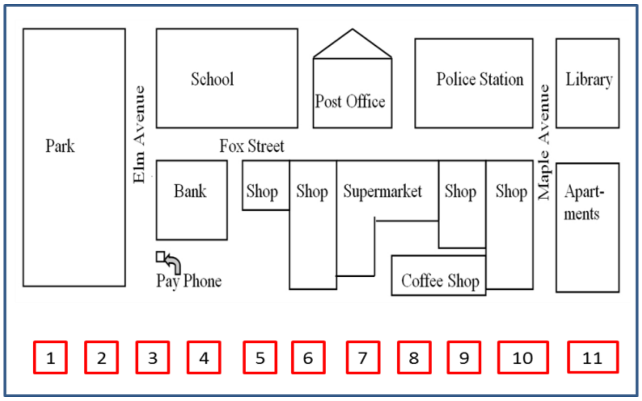
Overview

You should design a **2D scene, which can be extended to 3D** as shown below. You should change the layout but you have to keep the number of objects and shapes. Text is not required. Each object has its own number except Shop. All shops have the same object number. Total numbering is up to 11.



Requirements

- **15 pts – OpenGL program / Double Buffering and RGBA color space & Showing a 2D scene:** You create a program that uses OpenGL and GLUT. The program compiles and runs. A window appears on the screen. You use double buffering. Camera setting (gluOrtho2D) should be (-1, 1, -1, 1).
- **25 pts - Create a 2D scene:** The scene is drawn as a collection of objects (meshes). You should at least implement the scene as provided above and the objects are assembled through a collection of two types of basic primitives: *triangles* and *quadrilaterals only*. Specify the numbering in the written report.
- **35 pts – Mouse position control:** You should implement little red boxes which control the colors of the objects (on the created window, any place is fine but the top right is recommended). If someone clicks one of the red boxes with the mouse left button, the colors for the corresponding object and the box should be changed. The example is shown in the following. You do not have to write the text on the scene.



- **20 pts – Mouse motion control:** When the mouse is moving from top to bottom while the mouse right button is clicked (pushed down), the colors for the even numbered objects will be changed. When the mouse is moving from bottom to top while the mouse right button is clicked (pushed down), the colors for the odd numbered objects will be changed.
- **5 pts – Keyboard control:** When a number (1~0, e) is typed on the keyboard, the color for the corresponding object should be changed. 0 indicates “10” and e indicates “11”.