

Exercise 02561-04: Input – mouse, keyboard, menus, windows, selection, pick

Reading	Angel: chapter 2.11 – 2.12, 8.7 – 8.8
Purpose	The purpose of the exercise is to get acquainted with input and window management in Glut and OpenGL. In particular we will work with mouse input, keyboard input, menus, multiple windows, logic operations, selection and picking.
Part 1 Pick	Run the demonstration program 02561-04-01. Understand how the program handles mouse and keyboard input as well as menus and sub-menus. a) What does the function <code>pick</code> actually do?
Part 2 Select	Run the program 02561-04-02 and understand how selection works. a) Outline what steps are needed to perform a selection.
Part 3 Circuit diagram editor	Make a program, which can draw and edit simple circuit diagrams, such as the one shown in Appendix A. The program should support capacitors, resistors, and transistors. A user should be able to insert new components and delete old components. In addition, it should be possible to move, rotate, and scale existing components. a) Implement the function <code>buildTransistor</code> (see how a transistor looks like in Appendix A). The geometry should be defined between $(-1,-1,0)$ and $(1,1,0)$. If implemented correctly the components will appear in the menu. b) Implement the function <code>getSelectedItem</code> which should return what <code>menuItem</code> is related to point (return <code>menu_none</code> if point is outside the menu). If implemented correct, mouse-over effect and menu select should work. c) Implement <code>insertComponent</code> , <code>component_t.render</code> and <code>renderScene</code> . You should now be able to insert components into the scene. d) Implement <code>selectObject</code> and <code>motion</code> . You should now be able to move, rotate and scale e) Implement delete object functionality. (Modify the mouse function). f) Create a simple diagram and take a screenshot.

Exercise 02561-04: Input – mouse, keyboard, menus, windows, selection, pick

Part 4 - Optional	<p>Optional - Implement persistence. At startup the program should load a text file in which each line contains the information about a single component in the diagram.</p> <p>A line could contain the following information:</p> <p style="text-align: center;">comp-name tx ty theta sx sy</p> <p>, where comp-name is component type, (tx,ty) is the translation, theta is the rotation, and (sx,sy) is the scale of the component. When the program is shut down, the current diagram should optionally be saved.</p>
Part 5 – Optional	<p>Optional - Extend the program so you can do more advanced editing of the diagram. This could include zooming, etc.</p>

*Exercise 02561-04: Input – mouse, keyboard, menus, windows,
selection, pick*

Appendix A Circuit diagram

