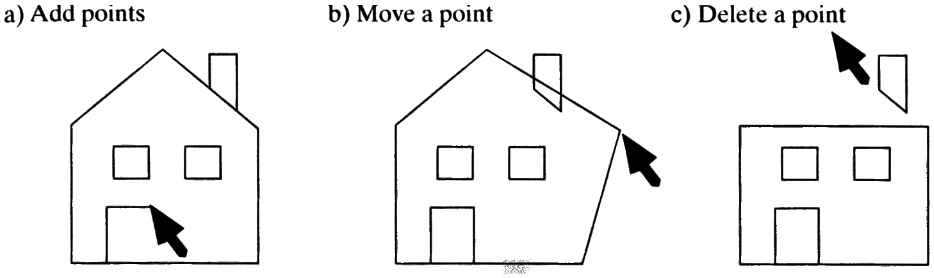
CSCI 4250/5250 Project 2
Due Date: 1pm, Monday, October 1st, 2012

Write a program to implement a polyline editor. Drawing programs often allow one to enter polylines using a mouse and then to edit the polylines until they present the desired picture.

Figure a) below shows a house in the process of being drawn; the user has just clicked at the position shown, and a line has been drawn from the previous point to the one designated by the mouse.

Figure b) shows the effect of moving a point. The user positions the cursor "near" the vertex of some polyline, presses down the mouse button, and "drags" the chosen point to some other location before releasing the button. Upon release of the button, the previous lines connected to this point are erased, and new lines are drawn to it.

Figure c) shows how a point is deleted from a polyline. The user clicks "near" the vertex of some polyline, and the two line segments connected to that vertex are erased. Then the two other endpoints of the segment just erased are connected with a line segment.



a) Add points

b) Move a point

c) Delete a point

Write a program that allows the user to enter and edit pictures made up of polylines. You may assume that the user will not need more than 100 polyline objects. Each object will have no more than 50 polylines.

Create a polyline class that has methods to handle all operations that must be performed on a polyline. The user interacts with the screen by pressing keyboard keys and pointing and clicking the mouse. The functionality of the program should include the following "actions":

- Begin ('b'): (create (**b**egin) a new polyline)
- End ('e'): (**e**nd the creation of a polyline)
- Loop ('c') (end the creation of a polyline and form a **c**losed polygon to be colored in the current color)
- Delete ('d'): (**d**elete – the user should select a point to be deleted)
- Move ('m'): (**m**ove – the user positions the cursor "near" the vertex of some polyline, presses down the mouse button, and "drags" the chosen point to some other location before releasing the button)
- Restart ('r'): (**r**estart **--** erase the screen and start the picture over (i.e. erase all polylines too))
- Quit ('q'): (exit from the program)

The list of polylines can be maintained as an array of polyline objects.

- The verb **Begin**, activated by pressing the 'b' key, permits the user to create a new polyline, which is stored in the first available "slot" in the array.
- The verb **End** stops the creation of a polyline.

- The verb **Loop** stops the creation of a polyline and the makes the last point in the polyline the same as the first. The polyline is then replaced by a polygon drawn in the current drawing color.
- The verb **Delete** requires that the program identify which point of which polyline lies closest to the point currently designated by the mouse. Once that point is identified, the "previous" and next vertices in the chosen polyline are found (if there is a previous and next). The two line segments connecting to the chosen vertex are erased, and the previous and next vertices are joined with a line segment.
- The verb **Move** finds the vertex closest to the currently designated point and waits for the user to drag the mouse and release it at the new position of the point, at which point the vertex is moved to this new point.
- The verb **Restart** clears the screen allows the user to start all over. The list of polylines should be "cleared" too.
- The verb **Quit** causes the program to terminate.

To allow the user some choices concerning the color of various objects, a right click of the mouse should cause a popup menu to appear which allows the user to select a color from a menu containing the colors red, blue, green, yellow, magenta, cyan, and black.

A portion of the screen window at the bottom should be reserved for reminding the user what action is required next. It will be like a "status" bar and will start with a brief explanation of the keystrokes available to the user. Depending on the selected keystroke, the next action required should be reflected in the "status" bar. For example, if the user selected '**b**' for begin, the status bar might have a message saying to select the next point using the mouse or select '**e**' for end or '**c**' for ending and making the polyline into a polygon.

Name your files for the ***polyline*** class: polyline.h and polyline.cpp, and name your main program editor.cpp.

Instruction to turn in the program:

- login to PeerSpace (peerspace.cs.mtsu.edu) using your pipeline username and the class account password;
- click on **Tools|Assignments** to submit your softcopy. Check to make sure the file is submitted by checking the **status** of submission: click on the ***status*** link after your submission.