Hello, I'm chengjiao yang, this is my first computer graphic project--HW1

1.What does this project do

answer: this project 100% doing what the HW1 requirement(Homework 1: *PoliBook*, an interactive polyline viewer and editor), and 100% complete it, and 100% can run in Zoolab machines.

2. What are the files in zipfile doing

answer: this project born from the start\_code which the professor provide us, I didn’t change structure in it, what I do is modify the example.cpp at the project. In addition, I make a file called drawing for storing the 9 grs format files, the project will load those files.

3.What functions does this project have

answer: this project has 6 stage:

* **'p' key: (state p) Response:** the program draws all polyline files provided at the top of the viewport in 10 equal-sized thumbnails (9 thumbnails of polylines above + 2 occurrences of vinci.dat) and randomly picks one to draw in large dimensions in the "main drawing window" . In the state p, if a user clicks on any of the thumbnails at the top, the corresponding polyline file in that thumbnail becomes the current drawing. The "main drawing space" is erased and the polyline in that thumbnail is redrawn to fill the entire "main drawing space".
* **'t' key: (state t) Response:** An 6x6 tiling of all polyline files is drawn in the "main drawing space" (polyline thumbnails still at top edge). Repeatedly hitting the 't' key should create a different 6x6 tiling where the polyline picked for any tile is "random".(polyline thumbnails still at the top edge and aspect ratios are maintained within each tile)
* **'e' key (state e) Response:** PoliBook clears the screen and goes into drawing mode (polyline thumbnails still at the top edge). On the first click in the drawing area, a dot is drawn at wherever the user clicked. On subsequent clicks, a line is drawn from the last accepted mouse click position to the current mouse click position. Your program should be set up to accept up to 100 possible mouse clicks in one polyline. You can accept and store user-provided points in an array. If the 'b" key is held down while clicking, the current click point is **NOT** joined to the previous click point and instead a fresh polyline is started and drawn in addition to whatever previous polyline had been drawn. For instance, holding the "b" key down may be useful for beginning to draw the window of a house after drawing other parts. Part a) of the figure below shows how the drawing or addition of points works. The example 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.
* **'m' key (state m) Response:** Move a previously drawn point on the current polyline. Part b) of the figure below shows this move in action. 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.
* **'d' key (state d) Response:** Delete a point from a polyline. Part c) of the figure below 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 segments just erased are connected with a line segment.
* **'g' key (state g) Response:** Clear the screen and draw the gingerbread man described in class.
* esc: **Response**: The program terminates.

In addition, there is a key ‘c’, this key allow u to draw again from preview graphic when you already go into the m stage or d stage.

4.How to run the project

answer: just run GLSLExperiment.sln in vs2010, use Ctrl+F5 to run it, or u can directly run GLSLExperiment.exe at Debug folder.