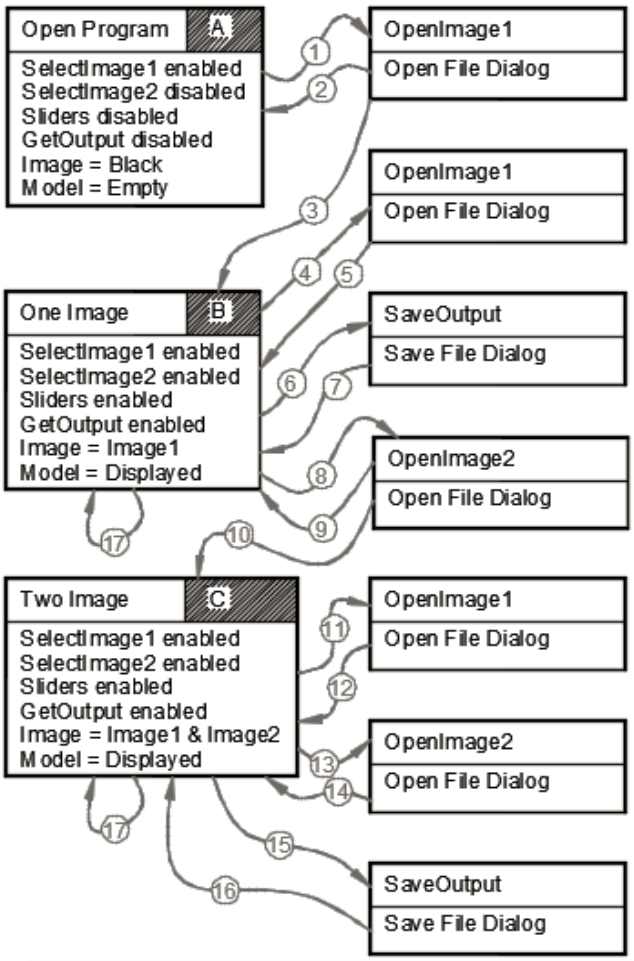
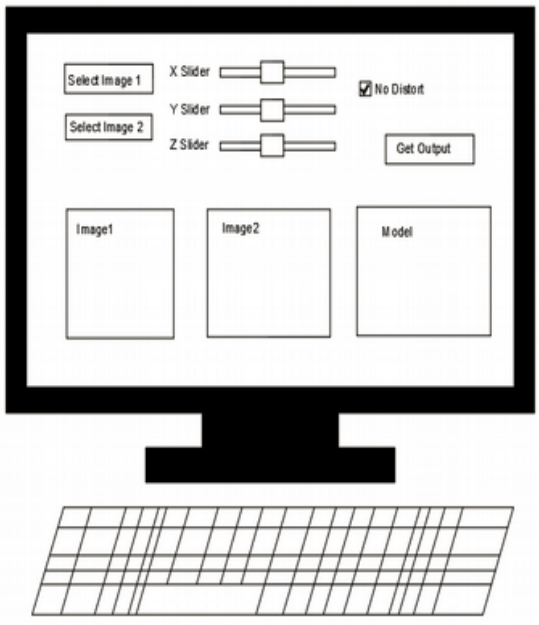
Criterion B: Design

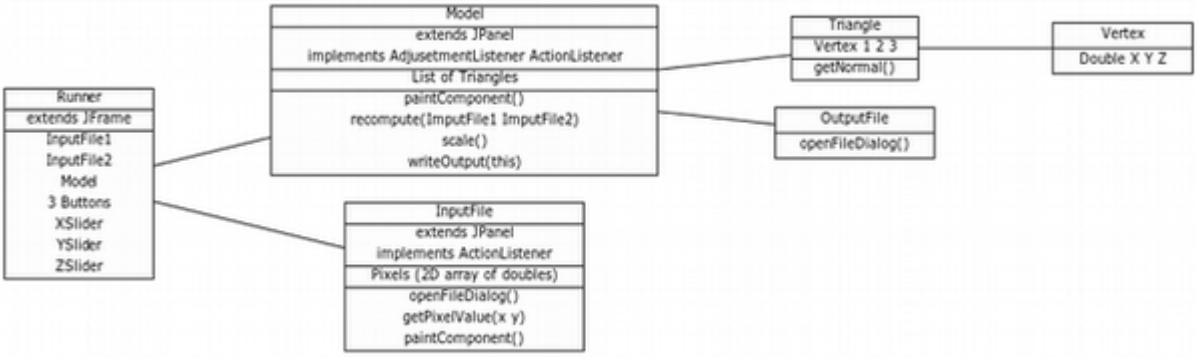




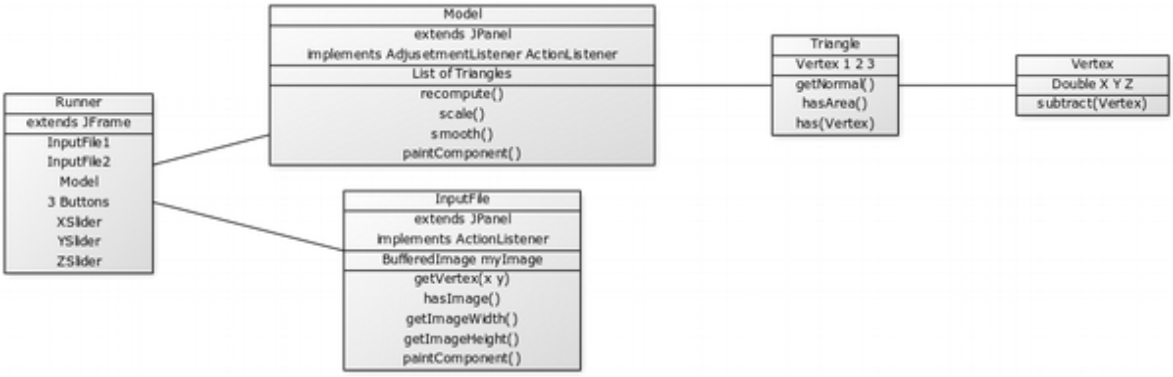
1. User presses SelectImage1.
2. File selected doesn't load correctly or user presses cancel. Return to original state A
3. File selected loads correctly. Transition to state B. Calculate 3D model. Display Image1 and Model.
4. User presses SelectImage1.
5. File selected doesn't load correctly or user presses cancel. Return to original state B.
   File selected loads correctly. Transition to modified state B. Recalculate 3D model. Display new Image1 and new Model.
6. User Presses GetOutput.
7. File saves to location specified and returns to original state B. If there is an error, do not create file.
8. User presses SelectImage2.
9. File selected doesn't load correctly or user presses cancel. Return to original state B.
10. File selected loads correctly. Transition to state C. Recalculate 3D model. Display Image1 & Image2 and new Model. If pixel at [x][y] on Image2 is > than pixel at [x][y] on Image1, adjust pixel value
11. User presses SelectImage1.
12. File selected doesn't load correctly or user presses cancel. Return to original state C.
    File selected loads correctly. Transition to modified state C. Recalculate 3D model. Display new Image1 & Image2 and new Model.
13. User presses SelectImage2.
14. File selected doesn't load correctly or user presses cancel. Return to original state C.
    File selected loads correctly. Transition to modified state C. Recalculate 3D model. Display Image1 & new Image2 and new Model.
15. User presses GetOutput.
16. File saves to location specified and returns to original state C. If there is an error, do not create file.
17. User adjust X, Y, or Z slider. If no distort is checked, maintain aspect ratio (automatically adjust other sliders). Recalculate 3D model. Display Image1 & Image2 and new Model.

Preliminary Overview of Classes and Properties:

Version 1:



Version 2:



Algorithm:

Triangle

Normals of triangle:

P_1, P_2, and P_3 are points of a triangle.

Resulting Vector N is normal.

$$V = P_2 - P_1$$
$$W = P_3 - P_1$$
$$N_x = (V_y * W_z) - (V_z * W_y)$$
$$N_y = (V_z * W_x) - (V_x * W_z)$$
$$N_z = (V_x * W_y) - (V_y * W_x)$$

Divide by length to convert to unit vector.

Example: P_1= 41, 40, 63 P_2= 82, 7, 3 P_3= 63, 43, 55
V=41, -33, -60
W= 22, 3, -8
N= 444,-992,849
Length(N)= 1379
N^I= N/Length(N) = .322, -.719, .616

Constructor:

In triangle class constructor will take three vertex objects as input. It will assign them to P1 P2 and P3. The Triangle class will have one static Vertex for scale factor, and static methods to set the scale.

Output according to ASCII STL:

In Triangle class there will be a toString method that will output in the following format with the scale factor applied.

```
facet normal ni nj nk
        outer loop
                vertex v1x v1y v1z
                vertex v2x v2y v2z
                vertex v3x v3y v3z
        endloop
endfacet
```

Model:
Recompute model:

arr[colCnt][rowCnt][2] where rowCnt and colCnt are equal to the width and height in pixels of the input image. And arr[x][y][0] is the z value determined by the pixel value in column x, row y of inputImage1. And similarly arr[x][y][1] for inputImage2. If inputImage2 has not yet been opened then arr[x][y][1] == 0 for all x and y.

1) Initialize a temporary 3D array of vertexes “varr” from arr:
        such that varr[x][y][i] == (x,y,arr[x][y][i])
2) Create all triangles for the top.
  for col = 0 to colCnt-2
    for row = 0 to rowCnt-2
        Create triangle varr[col][row][0] , varr[col][row+1][0], varr[col+1][row][0]
        Create triangle varr[col][row+1][0] , varr[col+1][row+1][0], varr[col+1][row][0]
3) Create all triangles for the left.
    for row = 0 to rowCnt-2
        Create triangle varr[0][row][1] , varr[0][row+1][1], varr[0][row][0]
        Create triangle varr[0][row+1][1] , varr[0][row+1][0], varr[0][row][0]
4) Create all triangles for the right.
    for row = 0 to rowCnt-2
        Create triangle varr[colCnt-1][row][0] , varr[colCnt-1][row+1][0], varr[colCnt-1][row][1]
        Create triangle varr[colCnt-1][row+1][0] , varr[colCnt-1][row+1][1], varr[colCnt-1][row][1]
5) Create all triangles for the front.
    for col = 0 to colCnt-2
        Create triangle varr[col][rowCnt-1][0] , varr[col][rowCnt-1][1], varr[col+1][rowCnt-1][0]
        Create triangle varr[col][rowCnt-1][1] , varr[col+1][rowCnt-1][1], varr[col+1][rowCnt-1 [0]
6) Create all triangles for the back.
     for col = 0 to colCnt-2
        Create triangle varr[col][0][1] , varr[col][0][0], varr[col+1][0][1]
        Create triangle varr[col][0][0] , varr[col+1][0][0], varr[col+1][0][1]
7) Create all triangles for bottom
        Similar to top but replace varr[x][y][0] with [x][y][1] and negate normals

Test Plan

Function to Test	Method of Testing	Result
Load Images	Load 1 image (different file types) Load 2 images	Works (crashes when image file is corrupt) Works (crashes when bottom image is larger than top)
Remove Image	Remove image 1 when there is only 1 image Remove image 2 when there are 2 images Remove image 1 when there are 2 images	Works Works Works
Update Images	Update image 1 when there is only 1 image Update image 2 when there are 2 images Update image 1 when there are 2 images	Works Works Works
Output Model	Output model with 1 image Output model with 2 images	Works
Overwrite Output	Overwrite an existing file	Works
Scale Model	Scale model with sliders	Works
Smooth Model	Smooth model using button	Works
Reset Orbit	Skew the orbit then reset	Works
STL output	Print an STL	Works

Criterion B Word Count: 175