

User Manual

Triple TK Image Display

Introduction

This document serves as a manual for the “Triple TK Image Display” program and is directed towards an audience that is assumed to have knowledge of the Insight Tool Kit (ITK) and its components.

The sections of this guide are as follows:

- **Start Up:** What a user needs to get started
- **System Overview:** Structural Diagram
- **Tutorial:** An example run of using the program
- **Reference:** Detailed decomposition of the GUI

1. Start Up: What the user needs to get started

- **Downloads:** Download and install VTK and FLTK.

Website for VTK: www.kitware.com

Website for FLTK: www.fltk.org

*Both downloads are free

- **Images:**

As of right now, you must have a MetaImage. The extension is .mh*. The program will prompt you for the location of this image.

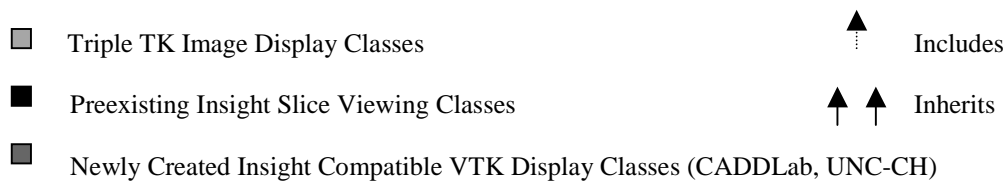
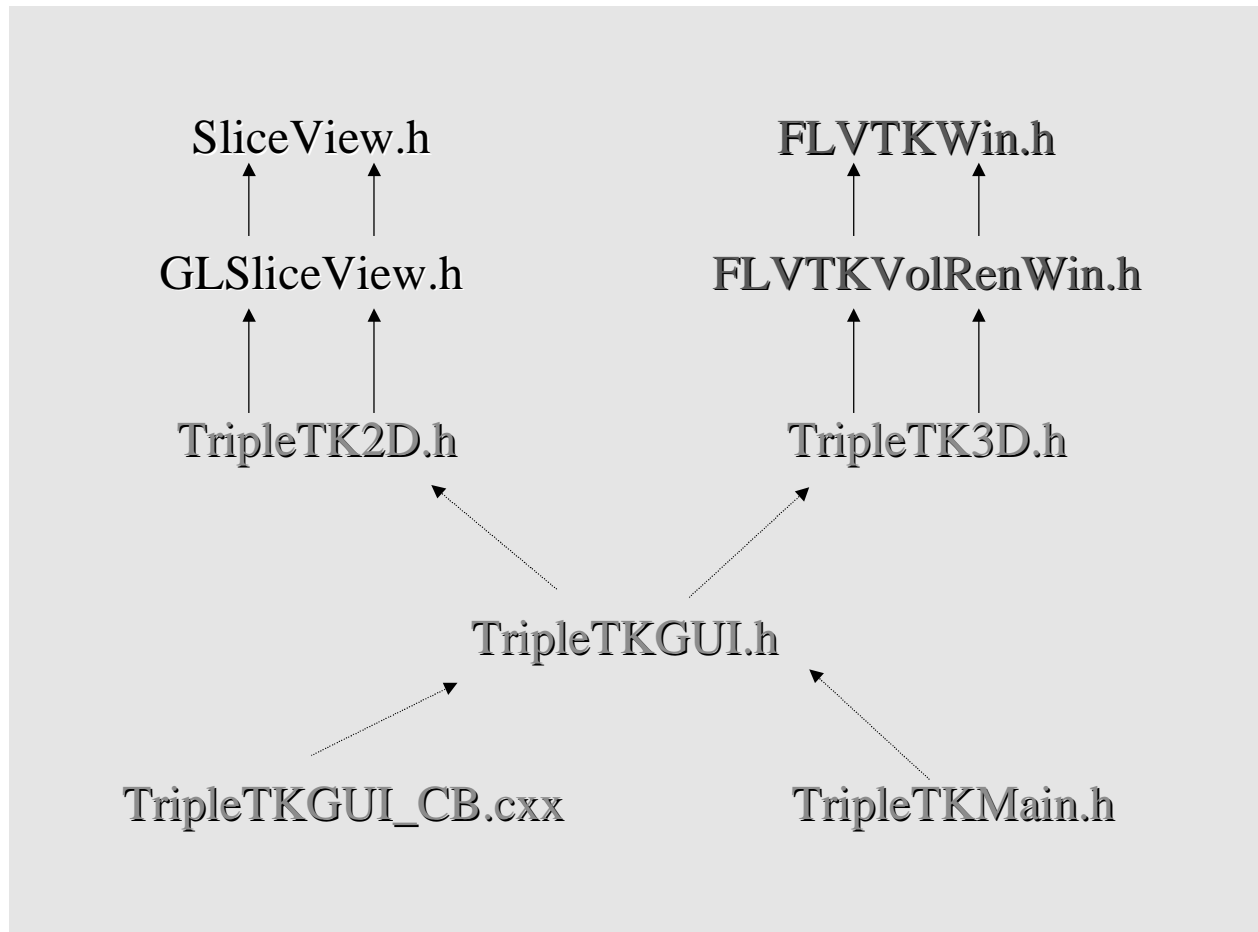
- **Build:** *The directory you installed Insight into is referred to as “Insight” here.
 - 1) Go to your Insight directory and edit the file CMakeList.txt. Look for the line containing the command "SUBDIRS". Add the directory “Examples” at the end of the list between

the parenthesis...the line will look similar to: "SUBDIRS(Utilities Code Examples)"

(without the quotes). That will indicate to the make program CMake that this directory should be included in the building process. *If you want to know more about CmakeList files and the portable maker CMake, please consult: Insight/Documents/CMake.pdf.

- 2) cd to Insight/Examples. Edit the CMakeList.txt file. Add a line containing the command "SUBDIRS(TripleTKImageDisplay)" (without the quotes) above the build options for the example projects.
- 3) Rerun "configure" (on Unix) or CMakeSetup (on windows) as you did the first time you installed the Insight toolkit. Activate the Compilation of "Unsupported". There should be new cache values for fltk and vtk now. Specify the include paths as the top level directories for vtk and fltk. Use the libraries vtkdll.lib and fltkd.lib respectively. Also specify the vtk bin path and the vtk lib path.
- 4) Rerun configure/CMakeSetup. This will generate the Makefiles (on Unix) or .dsp project file (on Windows for Visual C++) for TripleTKImageDisplay.
- 5) On Unix: run make
On Windows: open the itk.dsw workspace file and build it.
- 6) On Unix: The executable should be available at your Insight binary directory, under :
/Examples/TripleTKImageDisplay/...
On Windows: run the program from the Visual C++ environment.
- 7) Enjoy!

System Overview: Structural Diagram

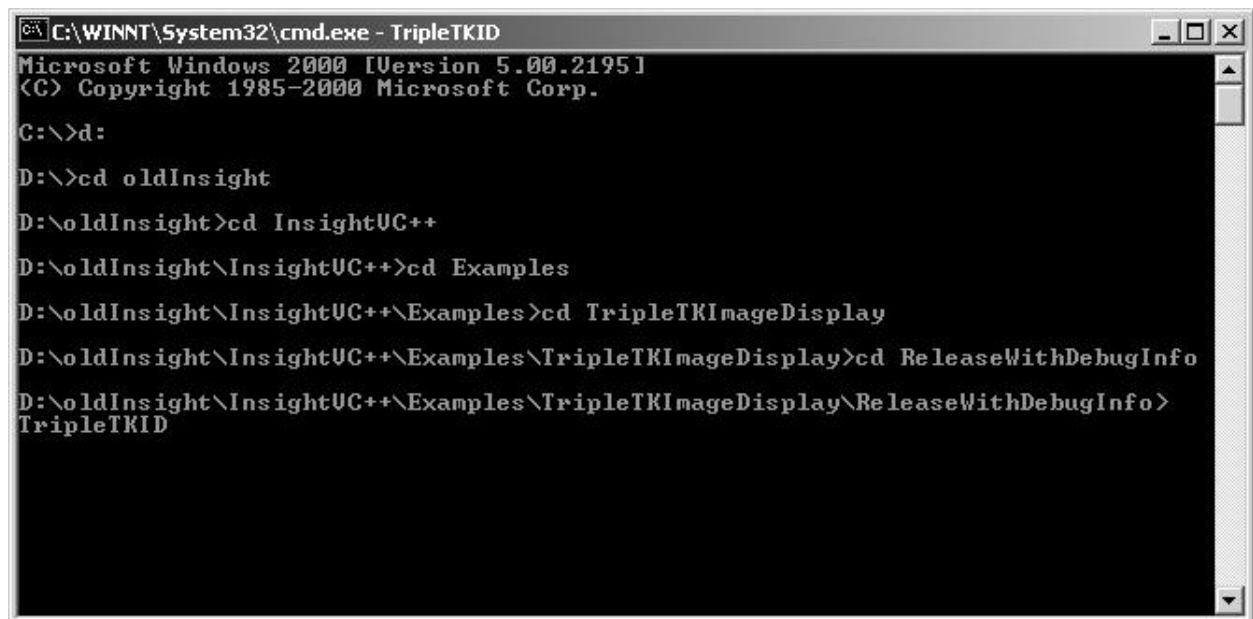


TripleTK2D and TripleTK3D are classes that exist for the purposes of allowing a place where one can make any desirable additions to their respective inherited parents without having to manipulate those parents. TripleTKGUI initializes these classes which are the basis for the 2D and 3D windows of the GUI. TripleTKGUI_CB holds the functionality of the GUI – fltk uses callback functions to define the functionality of its buttons and such. TripleTKMain holds the main function and additional functionality for initializing the program.

3.Tutorial: An example run of using the program.

1) Run the executable.

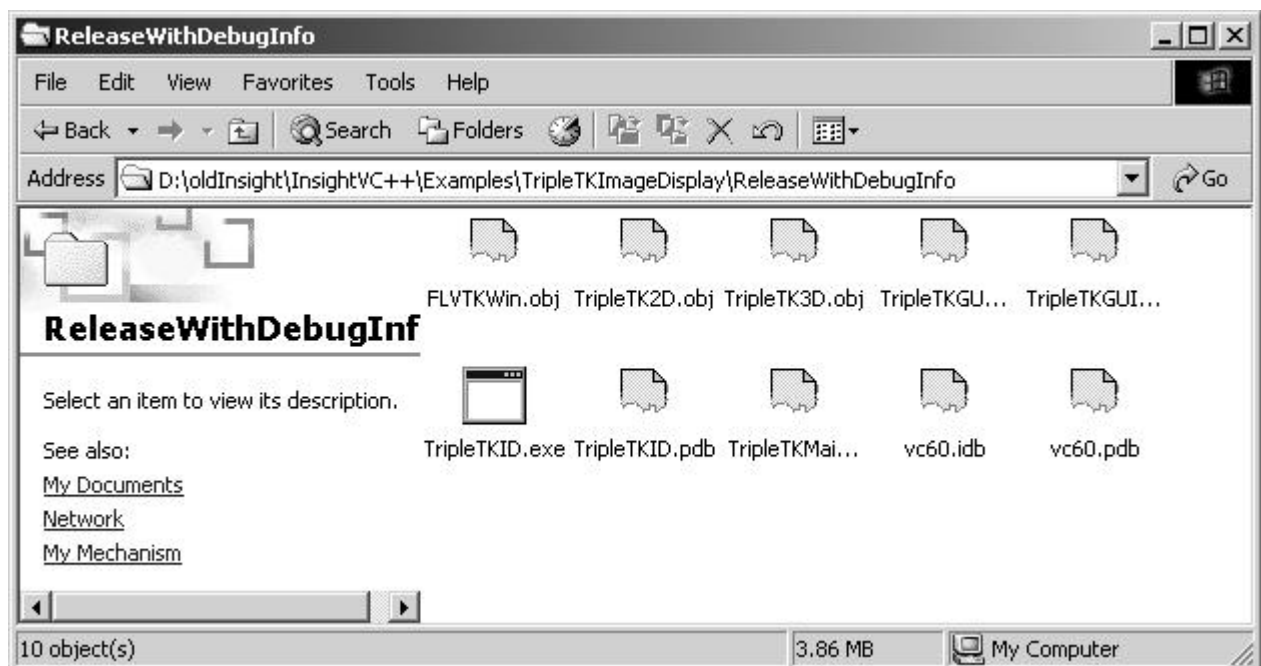
Unix: cd to the directory with executable in it and type TripleTKID.



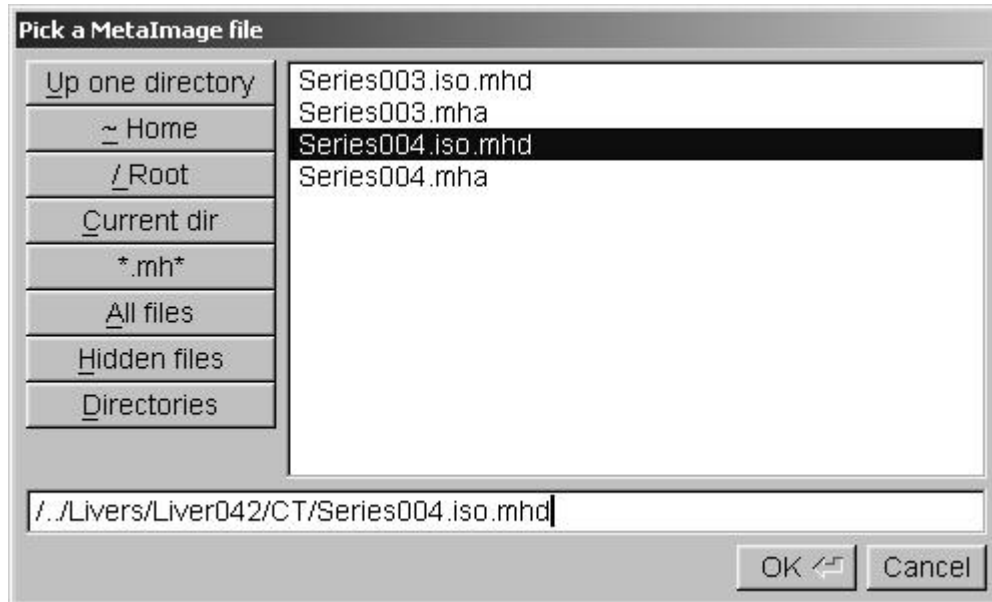
```
C:\WINNT\System32\cmd.exe - TripleTKID
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>d:
D:\>cd oldInsight
D:\oldInsight>cd InsightVC++
D:\oldInsight\InsightVC++>cd Examples
D:\oldInsight\InsightVC++\Examples>cd TripleTKImageDisplay
D:\oldInsight\InsightVC++\Examples\TripleTKImageDisplay>cd ReleaseWithDebugInfo
D:\oldInsight\InsightVC++\Examples\TripleTKImageDisplay\ReleaseWithDebugInfo>
TripleTKID
```

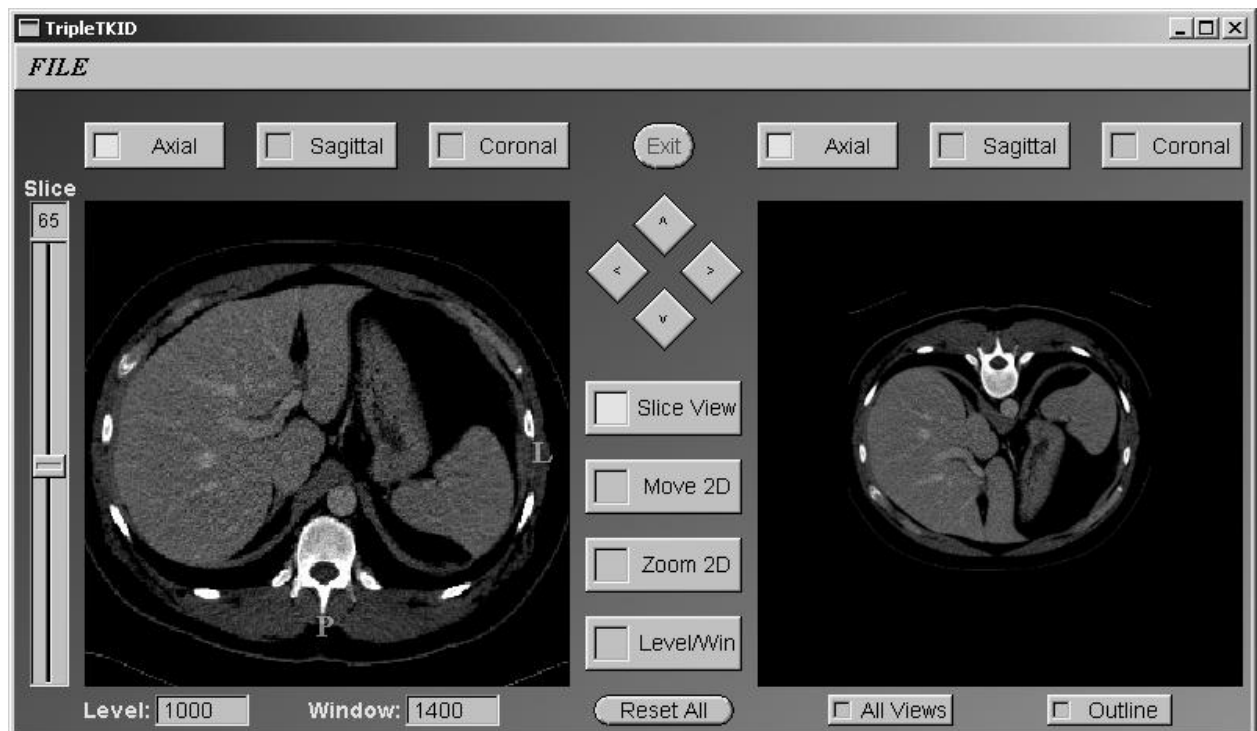
Windows: Double click on TripleTKID.exe



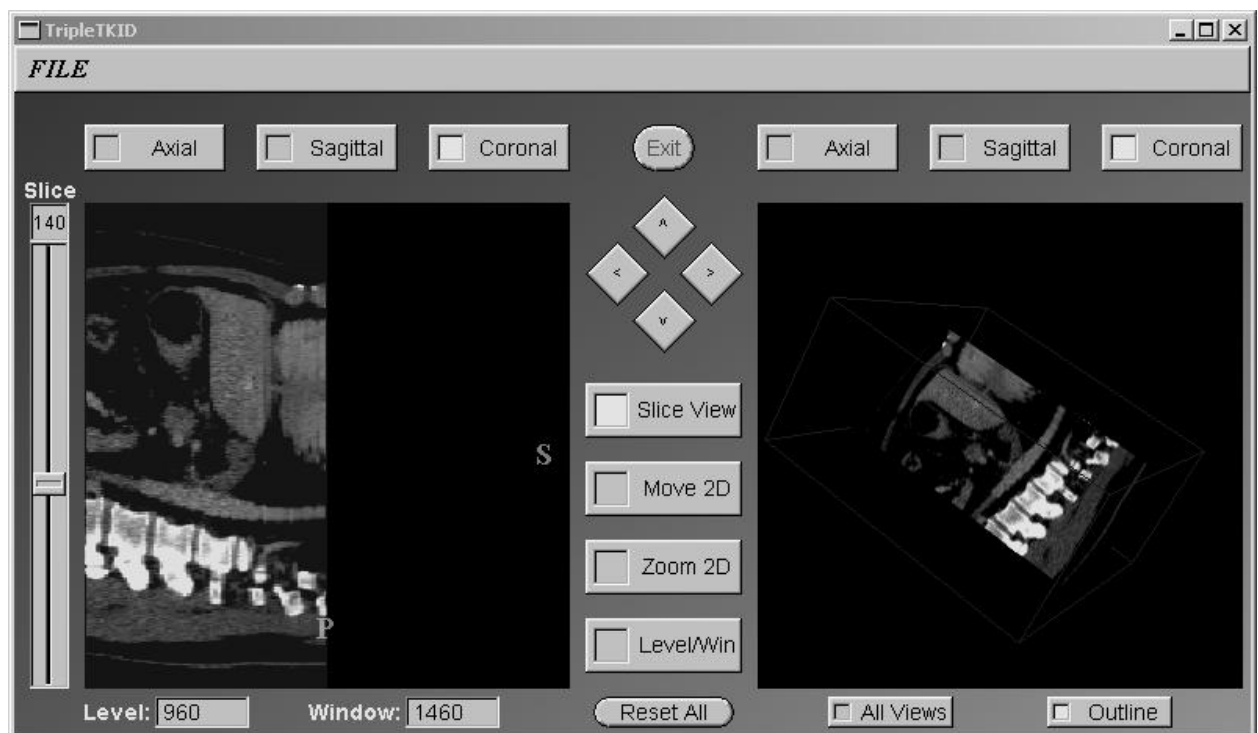
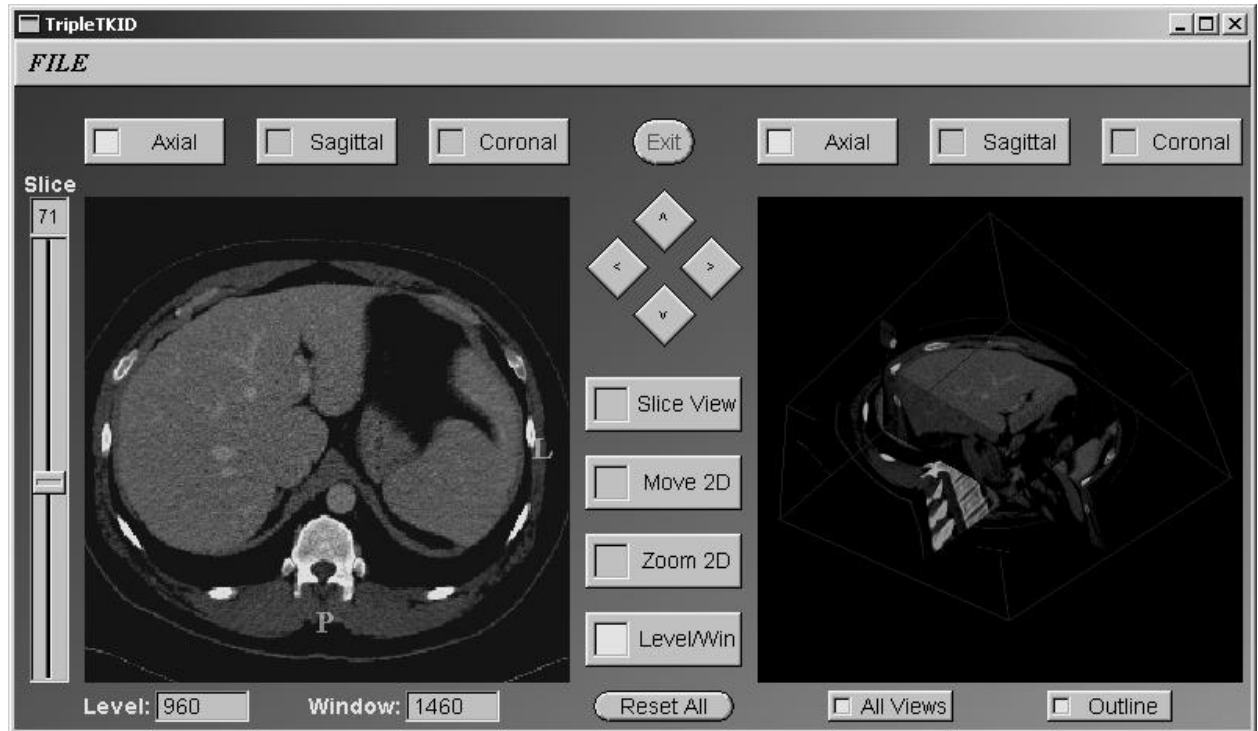
2) Select a MetaImage.



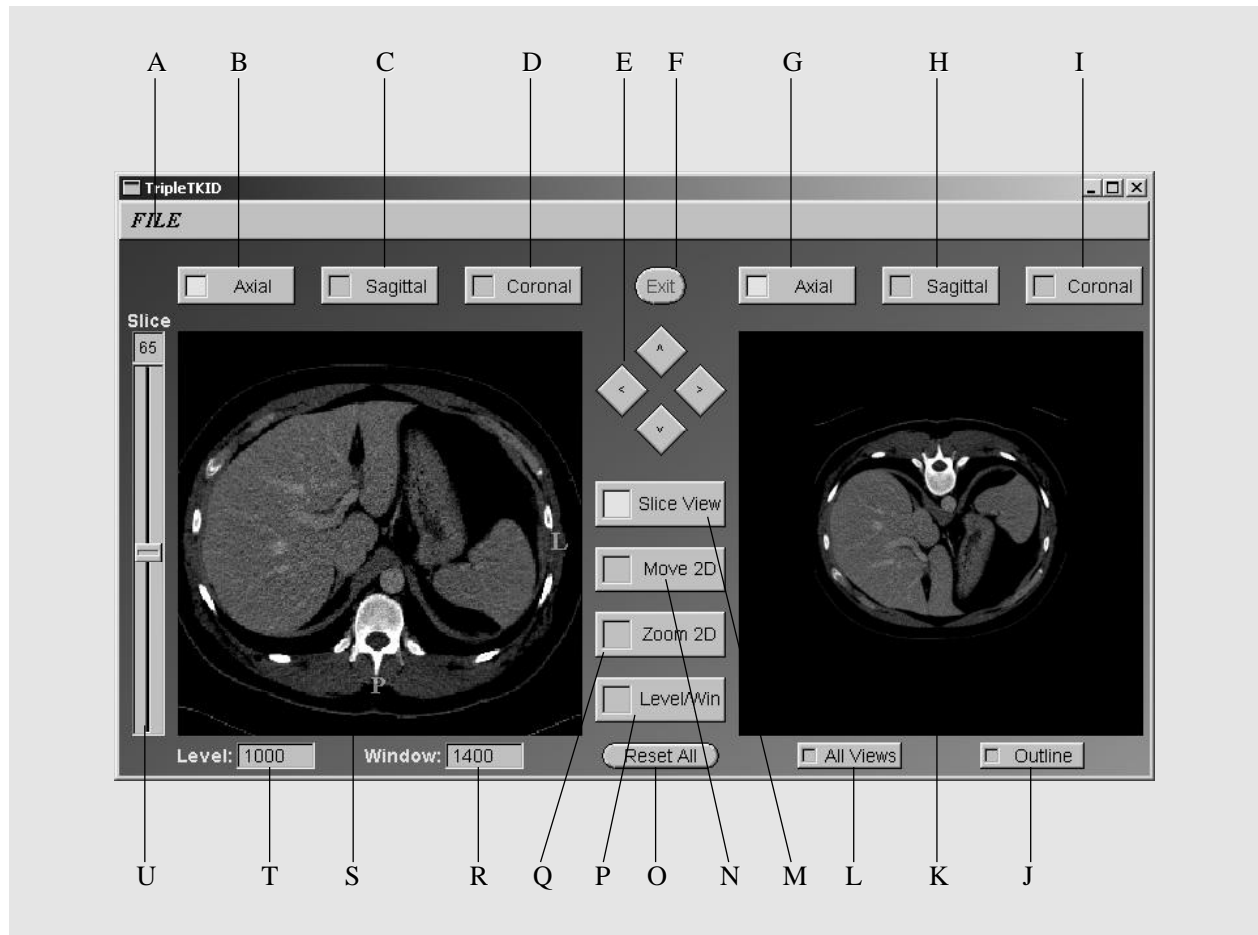
3) GUI at start up:



4) GUI in action:



4. Reference: Detailed Decomposition of the GUI.



A: File menu – solely contains an option to exit the program

B: Axial 2D radio button – is grouped with C & D – when clicked, 2D image shown axially

C: Sagittal 2D radio button – is grouped with B & D – when clicked, 2D image shown sagittally

D: Coronal 2D radio button – is grouped with B & C – when clicked, 2D image shown coronally

E: Arrow keys – functionality is dependent on the radio buttons M, N, P & Q

F: Exit button – exits the program

G: Axial 3D radio button – is grouped with H & I – when clicked, 3D image shown axially

- H:** Sagittal 3D radio button – is grouped with G & I – when clicked, 3D image shown sagittally
- I:** Coronal 3D radio button – is grouped with G & H – when clicked, 3D image shown coronally
- J:** Outline Button – when activated, a blue box shows the boundary of the 3D image
- K:** 3D window – shows the image in 3D
- L:** All Views Button – when activated, 3D image is shown axially, sagittally, & coronally
- M:** Slice View radio Button – is grouped with N, P, & Q – when activated, the up and down arrow keys scroll through the 2D image slices & left, right arrows are deactivated
- N:** Move 2D radio Button – is grouped with M, P, & Q – when activated, the up, down, left, and right arrow keys move the 2D image in their respective directions
- O:** Reset All Button – resets the program to its original state
- P:** Level/Win radio Button – is grouped with M, N, & Q – when activated, the up, down arrows adjust the level value & the right, left arrows adjust the window value of the 2D image
- Q:** Zoom 2D radio Button – is grouped with M, N, & P – when activated, the up and down arrow keys zoom in and out of the 2D image & left, right arrows are deactivated
- R:** Window output box – displays the current window value of the 2D image and is adjusted by the right, left arrow keys when Q is activated
- S:** 2D Window – shows the image in 2D
- T:** Level output box – displays the current level value of the 2D image and is adjusted by the up, down arrow keys when Q is activated
- U:** Slice Slider – can be used to change the slice being viewed and displays the slice number being viewed