**CSCI 552**

**Project #1 README**

Requirements:

* Microsoft Visual Studio 2017
* CMake 3.9.6
* VTK 8.1.0

Submitted Files:

* avanigadda.cxx (source file)
* CMakeLists.txt (source file)
* avanigadda.exe (executable file)

1. Make sure Visual studio, CMake and VTK installed and synchronized each other perfectly.
2. Create a “src” folder and put source files in it. i.e avanigadda.cxx and CMakeLists.txt
3. Create a “build” folder and make sure it is empty.
4. Open CMake-gui desktop app.
5. Give folder path of “src” to “where is the source code” in CMake-gui app.

Example: path = C:/…../Project1/src

1. Give folder path of “build” to “where to build the binaries”
2. Click **configure**
3. After successful configure, you will notice **Configuring done**, then Click **Generate**
4. After successful Generating, you will notice **Generating done**
5. Files are generated in “build” folder. You will find “**avanigadda.sln**” in it. Double Click on it. it will lead you to Visual Studio.
6. After successful installation and scanning of files, it will show **Ready** in bottom-left corner. Then Right click on **ALL\_BUILD** and click on **Build.**
7. After successful build, you will notice **Debug** folder in **“build”** folder containing three files namely avanigadda.exe, avanigadda.dil and avanigadda.pdb. Open Debug folder and put the “Heightmap.PNG” in it.
8. Copy all .**dil** and .**pdb** files in **….\VTK\bin\bin\Debug** (these files are created while installing VTK through CMake)
9. Paste those files in **….\build\Debug** i.e debug folder created in build containing avanigadda.exe and Heightmap.PNG
10. Open command prompt navigate to build folder containing executable file.

Example: cd C:\.....\Project1\build\Debug

1. Run the executable file in command prompt passing image name as arguments.

Example: > avanigadda.exe Heightmap.PNG

**Known bugs:**

While executing avanigadda.exe in command prompt, you will get warnings in vtkOutputWindow. Ignore them and find the visualization result in another window.

**Data representation and visualization methods:**

1. We read the .PNG file through vtkPNGReader object.
2. We can find the pixel intensities of image by vtkImageMagnitude. It will find the magnitudes of each pixel in image.
3. We convert the image data i.e magnitudes into Polydata by using vtkImageDataGeometryFilter. It will extract the geometry from structured points of dataset (contains magnitudes and x,y) and convert it into Polydata.
4. We used vtkWarpScalar to scale the z co-ordinates. It will decide the scaling factor and gives 3D effect to image.
5. vtkMergeFilter merge the image and polydata.
6. vtkPolyDataMapper maps the polydata to graphics.
7. Then generated graphics are pipelined to actor by vtkActor.
8. It is pipelined to vtkRenderer.
9. It is pipelined to vtkRendererWindow.
10. It is pipelined to vtkRenderWindowInteractor to get interaction with graphics.
11. Finally, we add actor to renderer, renderer to renderer\_window. And start the visualization.