Generation of a 3D object from a Digital Elevation Model (DEM)

Supervisor: Prof. Pascal Desbarats

VU Manh Tu
NGUYEN Bao Xuan Truong
PHAM Phu Quoc
BUI The Anh

06-04-2016

I DEM File

1 Project overview

This report outlines the design and development of a computer software to visualize DEM files, which are from different formats, in 3-Dimensions. It allows users using several convenient functionalities such as rotations, translation, zoom along the axises, resizing or cropping using boundary boxes, creating a support at its basis to transform it into 3D object, labeling it on. The result can be exported in OBJ and STL formats.

The solution should be developed on *C++ / Qt framework*[1] and a visualization in *OpenGL*[2].

2 Concepts

2.1 What is digital elevation model (DEM)?

A digital elevation model (DEM) is a digital model or 3D representation of a terrain's surface commonly for a planet (including Earth), moon, or asteroid created from terrain elevation data.

[1] `http://http://www.qt.io/ide/`
[2] Opengl-superbible-comprehensive-tutorial-and-reference-5th-edition-2010

It is 2D map in which each point is associated with its height. DEM can be obtained through techniques such as photogrammetry, lidar, land, surveying, etc.[3]

2.2 DEM file formats

USGS DEM
The USGS DEM format is a standard format for the storage of raster digital elevation data. The USGS has produced five different digital elevation products with the primary differing characteristic being the spacing, or sampling interval, of the data:[4]

- 7.5-Minute DEM 30- x 30-meter data spacing
- 2-Arc-Second DEM 2- x 2-arc-second data spacing
- 15-Minute Alaska DEM 2- x 3-arc-second data spacing
- 7.5-Minute Alaska DEM 1- x 2-arc-second data spacing
- 1-degree DEM 3- x 3-arc-second data spacing

Purpose
DEM file is used in the generation of three-dimensional graphics displaying terrain slope, aspect (direction of slope), and terrain profiles between selected points. At the USGS, DEMs have been used in combination with digital raster graphics (DRG's), digital line graphs (DLG's), and digital orthophoto quadrangles (DOQ's) to both enhance the visual information for data extraction and revision purposes and to create aesthetically pleasing and dramatic hybrid digital images. Non-graphic applications such as modeling terrain and gravity data for use in the search for energy resources, calculating the volume of proposed reservoirs, and determining landslide probability have also been developed.[5]

Structure
DEM file is an ASCII file format consisting of a header record (Type A), data records (Type B) and an accuracy metadata record (Type C)[6]
The physical structure of the DEM distributed to the user is as follows:[4]

[3] https://en.wikipedia.org/wiki/Digital_elevation_model
[4] http://agdc.usgs.gov/data/usgs/geodata/dem/dugdem.pdf
[5] http://www.softree.com/Tips_Techniques/T-004-USGS-DEM/USGS_DEM.pdf
[6] http://www.geobc.gov.bc.ca/base-mapping/atlas/trim/specs/BC-DEM-specifications-2002-12.pdf

- Data recorded in fixed-block format on unlabeled or ANSI-labeled 9-track magnetic tape at 1,600 or 6,250 bpi density.
- Logical record size of 1,024 bytes. No more than one logical record type (A, B, or C) recorded in any 1,024-byte record. However, more than one 1,024-byte record is usually required to store a single record type B. The logical record is padded with blanks if necessary to fill to the end of the logical record. Bytes 1,021-1,024 of each logical record are padded with blanks.
- Physical record size of 4,096 bytes; that is, 4 logical records per physical record.
- Data written as ANSI-standard ASCII characters.

SDTS DEM

SDTS stands for Spatial Data Transfer Standard, is a robust way of transferring earth-referenced spatial data between dissimilar computer systems with the potential for no information loss. It is a transfer standard that embraces the philosophy of self-contained transfers, i.e. spatial data attribute, dereferencing, data quality report, data dictionary, and other supporting metadata all included in the transfer. [7]

SDTS has 7 parts. Parts 1-3 are about SDTS specification which is organized into the base specification, all of them are related, but relatively independent. Parts 4-6 are about multiple profiles, each define specific rules and formats for applying SDTS for the exchange of particular types of data in SDTS: [8]

- Part 1 Logical Specifications:

  It consists of three main sections, which explain the SDTS conceptual model and SDTS spatial object types, components of a data quality report, and the layout of all SDTS modules.

- Part 2 Spatial Features:

  It contains a catalogue of spatial features and associated attributes. This part addresses a need for definition of common spatial feature terms to ensure greater compatibility in data transfers. The current version of Part 2 is limited to small- and medium-scale spatial features commonly used on topographic quadrangle maps and hydrographic charts.

[7] `http://mcmcweb.er.usgs.gov/sdts/whatsdts.html`

[8] `http://mcmcweb.er.usgs.gov/sdts/standard.html`

- Part 3 ISO 8211 Encoding:

  This part explains the use of a general purpose file exchange standard, ISO 8211, to create SDTS file sets (i.e. transfers).

- Part 4 - Topological Vector Profile:
  The Topological Vector Profile (TVP) is the first of a potential series of SDTS profiles, each of which defines how the SDTS base specification (Parts 1, 2, and 3) must be implemented for a particular type of data. The TVP limits options and identifies specific requirements for SDTS transfers of data sets consisting of topologically structured area and linear spatial features.

- Part 5 - Raster Profile and Extensions:

  The Raster Profile is for 2-dimensional image and gridded raster data. It permits alternate image file formats using the ISO Basic Image Interchange Format (BIIF) or Dereferenced Tagged information File Format (Geo TIFF).

- Part 6 - Point Profile:

  The Point Profile contains specifications for use with geographic point data only, with the option to carry high precision coordinates such as those required for geodetic network control points. This profile is a modification of Part 4, the Topological Vector Profile, and follows many of the conventions of that profile.

DTED
DTED is a standard of digital datasets which consists of a matrix of terrain elevation values [9]. It is the oldest digital mapping format that we still use today. This standard was originally developed in the 1970s by NIMA - the National Imagery and Mapping Agency. DTED was originally developed to drive 3D milling machines and provide elevation data needed for cruise missile planning. [10]
DTED format have 3 level:[11]

- Level 0
  - Has a post spacing of approximately 900 meters.
  - Elevation post spacing is 30 arc second

[9] https://en.wikipedia.org/wiki/DTED
[10] http://www.mission-planning.com/DTED_Part1.htm
[11] http://fas.org/irp/program/core/dted.htm

  - Derived from NIMA DTED Level 1 to support a federal agency requirement
  - DTED Level 0 may be of value to scientific, technical, and other communities for and applications that require terrain elevation, slope, and/or surface roughness information. It allows a gross representation of the Earth's surface for general modeling and assessment activities. Such reduced resolution data is not intended and should not be used for automated flight guidance or other precision activity involving the safety of the public.

- Level 1
  - Has a post spacing of approximately 90 meters.
  - A uniform matrix of terrain elevation values with post spacing every 3 arc seconds
  - The information content is approximately equivalent to the contour information represented on a 250,000 scale map.

- Level 2
  - Has a post spacing of approximately 30 meters.
  - He basic high resolution elevation data source for all military activities and systems that require landform, slope, elevation, and/or terrain roughness in a digital format.

DIMAP

The DIMAP stands for Digital Image Map. It is the format for SPOT products introduced for the SPOT 5 launch in May 2002 and developed with CNES.[12]

DIMAP format consists of two parts:[12]

- Image: By default it is described in GeoTIFF format, comprised of
  - A TIFF part, the most widely used image format in the world.
  - A Geo part, adding georeferencing information for the image file (coordinates in the upper left-hand corner of the image and pixel size) to the basic TIFF file and may also describe the map projection used and its corresponding geographic system.

[12] http://www.geo-airbusds.com/en/196-the-dimap-format

- Metadata: this is written in XML, allowing to create customized keywords with corresponding values. It can be linked to an XSL style sheet which sorts and does the HTML layout of the information contained in the XML file.

II Software requirements

1 Functional requirements

1.1 Functional requirement 1

- ID: FR1
- TITLE: Install application
- DESC: A user has downloaded the application, then he should be able to install this application by following the instruction. After install, the application must notification if install success or not.
- RAT: In order for a user to install application.
- DEP:

1.2 Functional requirement 2

- ID: FR2
- TITLE: Select DEM file to open
- DESC: After the installation, the user should be able to run it and open file to import. The application must show 3D image of imported file if its format is acceptable.Otherwise, it shows notification that the file format is incorrect.
- RAT: For a user to open DEM file and visualize it in 3D
- DEP: FR1

1.3 Functional requirement 3

- ID: FR3
- TITLE: Rotate a 3D Image

- DESC: After having a 3D Image, then the user should be able to rotate it by using mouse movement action. The rotation includes move up, down, left, right or any angles
- RAT: For a user to rotate 3D image.
- DEP: FR2

1.4 Functional requirement 4

- ID: FR4
- TITLE: Zoom in/out a 3D Image
- DESC: After having a 3D Image, the user should be able to zoom in the 3D image by double clicking mouse. The mouse clicked point must be in range of 3D image. The user is also able to zoom out the 3D image by hold CRTL key while click.
- RAT:For a user to zoom in/out 3D image.
- DEP: FR2

1.5 Functional requirement 5

- ID: FR5
- TITLE: Select an area in a 3D Image
- DESC: After having a 3D Image, the user should be able to select an area in 3D Image by using mouse to draw a box on the 3D Image
- RAT: For a user to select an area in 3D image.
- DEP: FR2

1.6 Functional requirement 6

- ID: FR6
- TITLE: Resize an area in boundary box in a 3D Image
- DESC: Given that a user has selected an area on 3D Image, then the user should be able to resize this area by clicking right mouse and select resize. A dialog must show in order to select how many percentage will be resized.

- RAT: For a user to resize an area in a 3D image.
- DEP: FR5

1.7 Functional requirement 7

- ID: FR7
- TITLE: Crop a boundary box on 3D Image
- DESC: Given that a user has selected an area on 3D Image, then the user should be able to crop this area by clicking right mouse and select crop. After cropped, only the area within boundary box remains.
- RAT: For a user to crop an area in 3D image.
- DEP: FR5

1.8 Functional requirement 8

- ID: FR8
- TITLE: Create a support at the basis of 3D image
- DESC: After having a 3D Image, the user should be able to create a support at the basis of current 3D image by clicking the "Create support" button. With the support, the 3D object should be ready to be printed by a 3D printer.
- RAT: In order for user to transform image into 3D object.
- DEP: FR2

1.9 Functional requirement 9

- ID: FR9
- TITLE: Label the 3D object
- DESC: Given that a user has created a support at the basis of 3D image. The user should be able to label the 3D object by clicking the "Text" button, then clicking a position on the object or at its support to create textbox where user can write some note as label.
- RAT: For user to label the 3D object
- DEP: FR8

1.10 Functional requirement 10

- ID: FR10
- TITLE: Export to OBJ, STL files
- DESC: Given that a user has created 3D object, and may or may not be has labeled it. The user should be able to export to OBJ or STL file by clicking the "Export" button.
- RAT: For user to export to OBJ, STL files
- DEP: FR8

2 Non-functional requirements

- Compatible operation system: Linux
- Operation fluidity: Rotate, translate, zoom, resize, crop 3D image, create support at its basis or label the object should be executed without delay.

III Architecture Design

1 Introduction

1.1 Purpose

This part will explain the architecture of the application.

1.2 Developing environment

- OS: Linux
- Framework: QT(QT quick QML).
- 3D render library: OpenGL.

1.3 Usability & Maintenance

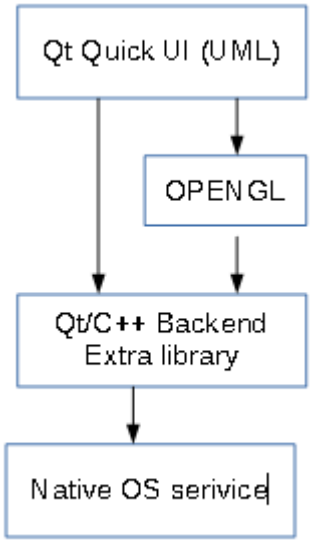
We use design pattern MVC for this project. MVC (ModelViewController) is a software architectural pattern for implementing user interfaces on computers. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that

information is presented to or accepted from the user. In the Model modules we have File Data Object and I/O Layer. The Controller modules we have 3D Process, Modules Process and Control View Data (This part going to contact to View modules, for what you wait to show) all of them for processing DEM files, how to move, cut, and zoom 3D image and print 3D object. Finally is View modules. this is only show the data from the Control View Data, and it has the UI in which user easily contact to the Controller Modules.

Why do we use the design above? First, it is about usability, which means the developers will know where and what they need to look into with specific situation. When a problem happens, developers can detect where the problems are and what to do next. For example, if we have a problem about zoom DEM images, that means we know the problem from Zoom modules and this modules has a problem from Zoom Controller. The developers read the zoom method and fixes the wrong algorithm. They can easily make more features without care about other modules. Finally, it is so easy for maintenance, not only for us, but also other teams if they are going to reuse our code. Because, although it has many modules, we only have to maintain the module that has problems.

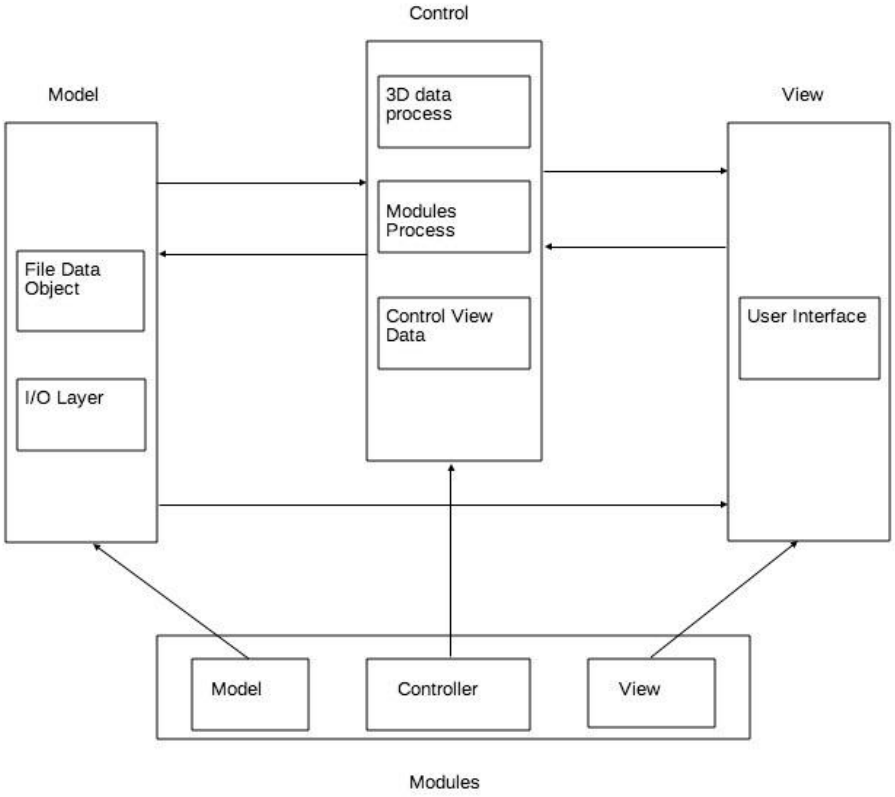
2 Overall Description

2.1 System architecture



- Qt Quick (QML) framework: Build user interface. Handling binding data, view animation.
- OPENGL: Provide 3D process library.
- Qt/C++: Provide database, file access api.
- Extra library: library for processing dem file.

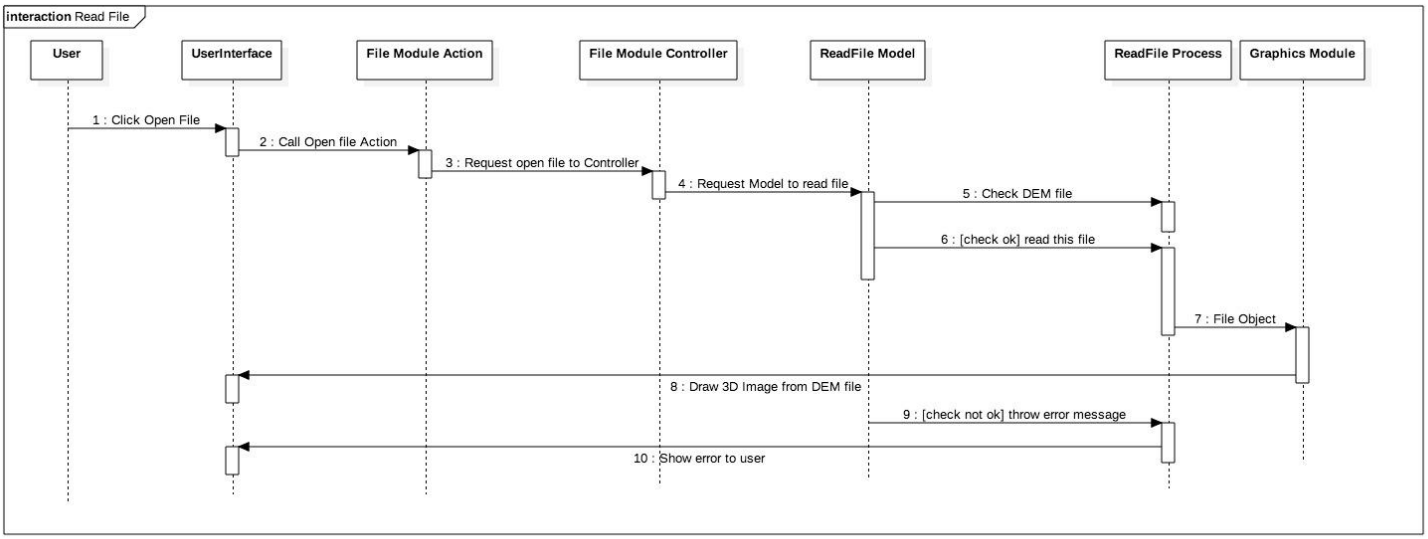
2.2 Software block diagram



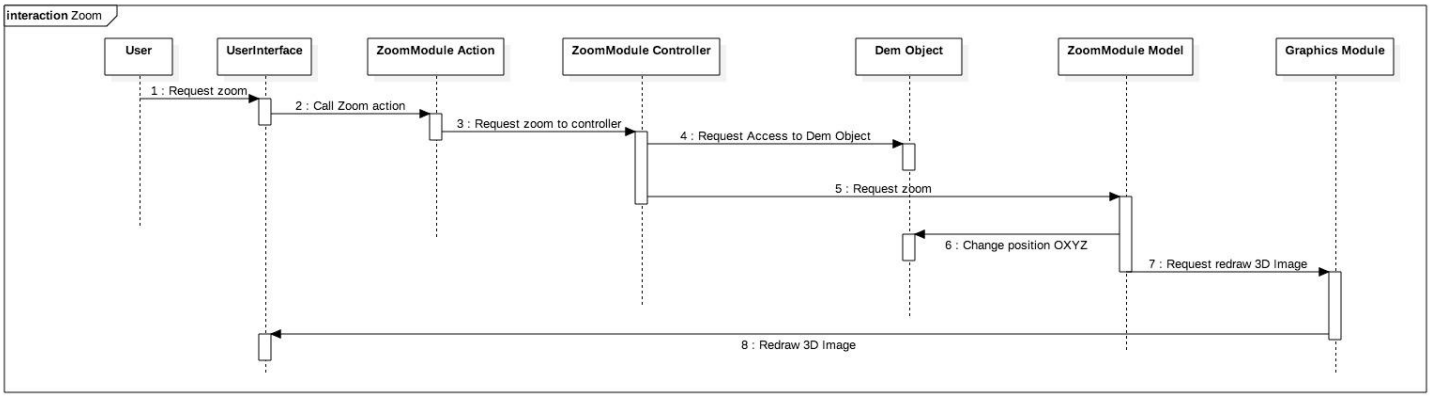
- Model: Handle File In/Out including read Dem file or Export. Model also holds Dem file Object
- Controller: Control data flow, process action request from View
- View: Show user interface
- Modules: A component include Model, Controller & View as part of program such as file module or zoom module

2.3 Data flow

- Open DEM file button



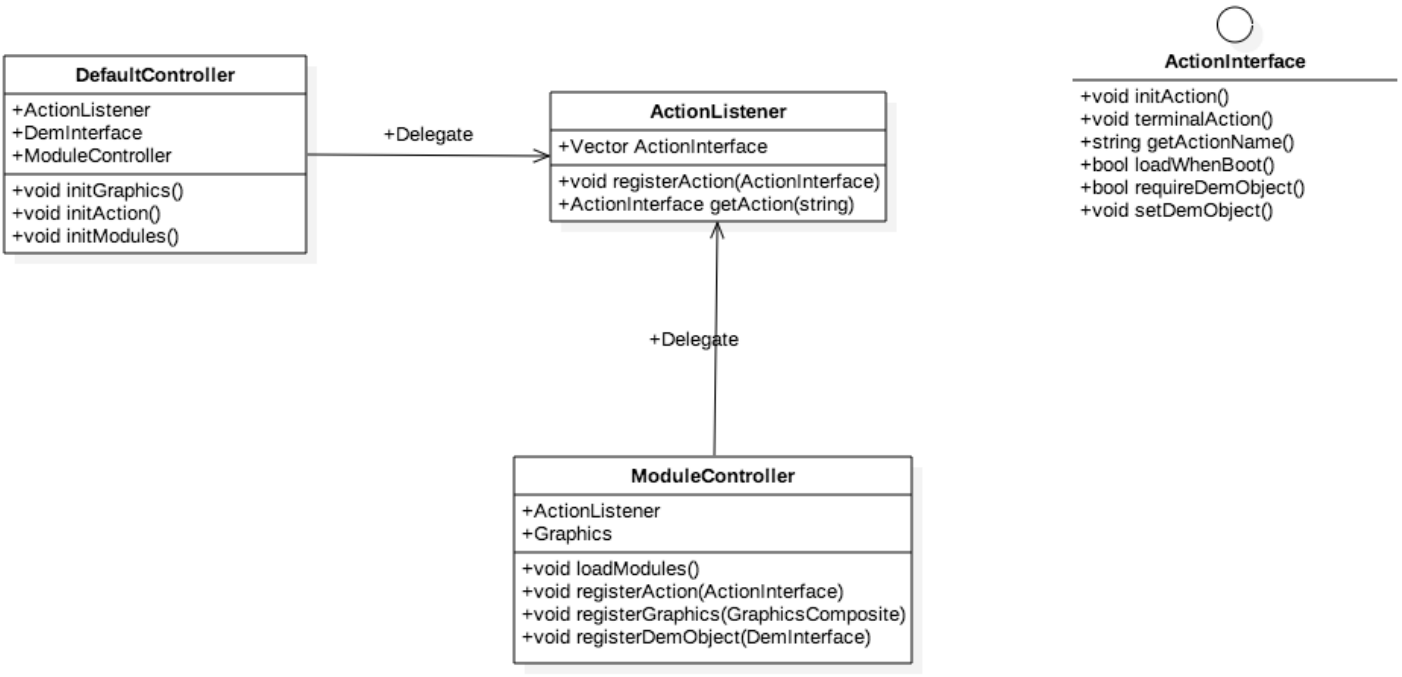
- Zoom



2.4 Class diagram

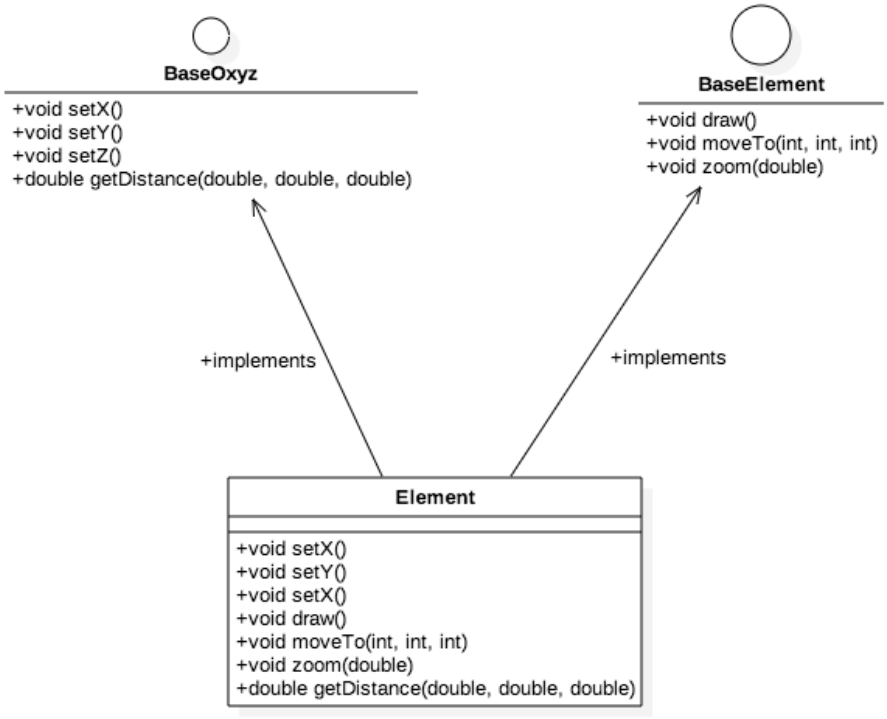
The main idea here is separate the program into multiple modules, each module only take care one task. For example, file module will handle reading Dem file or exporting to OBJ/STL format. To do that, we need to build a basic system with ability of adding more modules. The following class diagram provide a way to do it.

- Controller



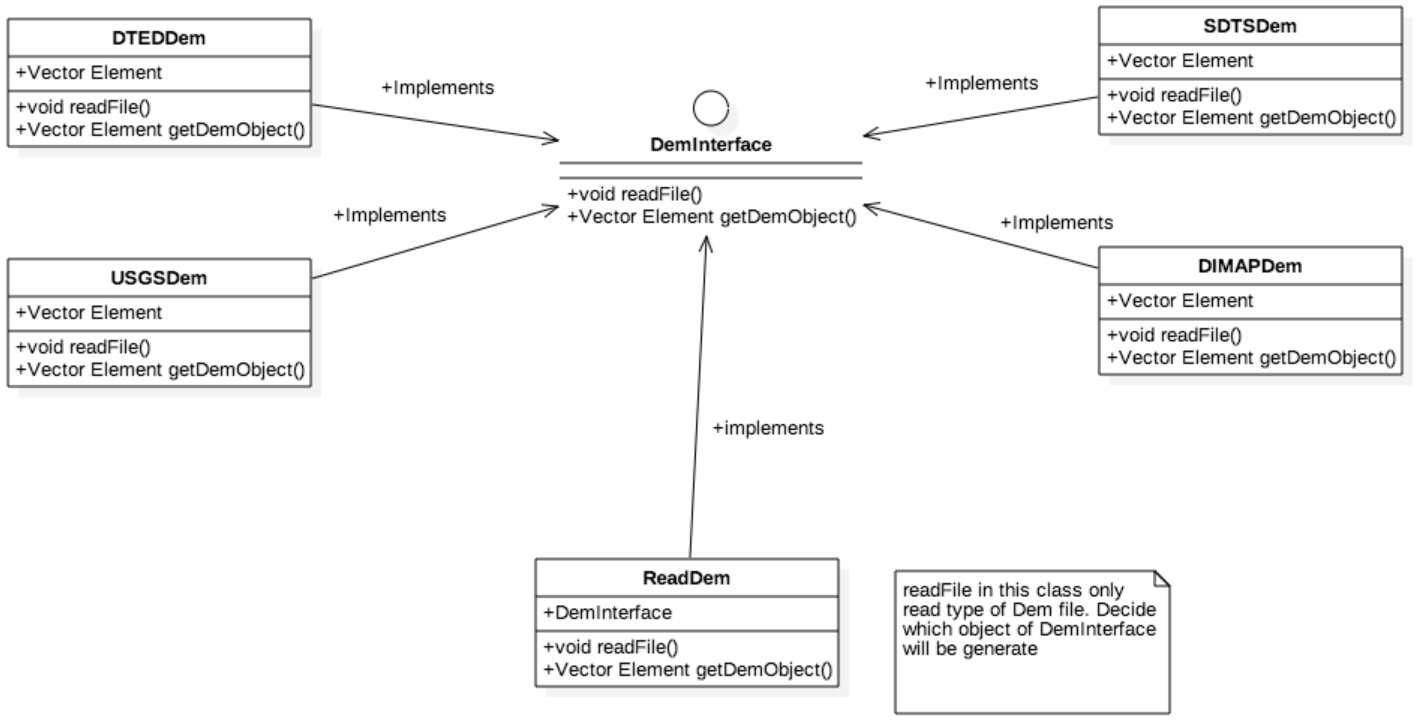
Action listener is the heart of the program. It provides a way for user to interact with the program. ActionListener will collect actions registered from modules & can be called from graphic viewer if needed. ActionListener also allows modules to register to access data from model.

- Base Element of DEM file



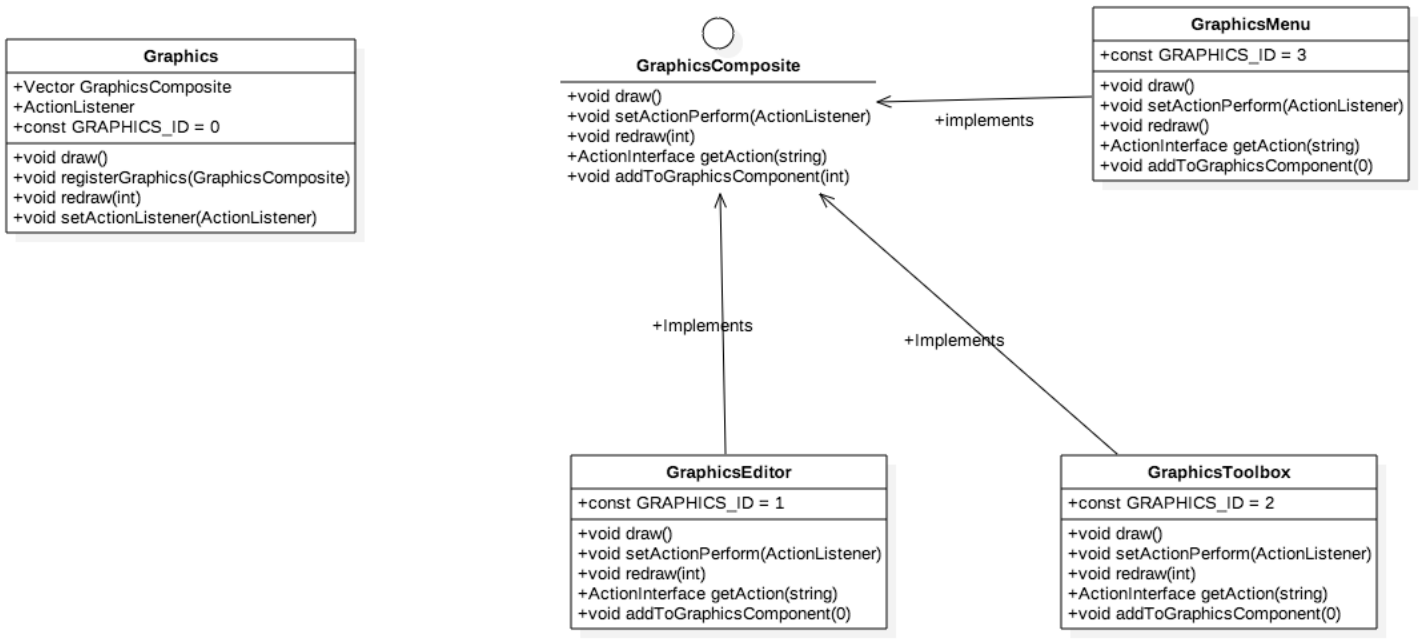
This is the basic element to represent element in Dem file.

- Read DEM file



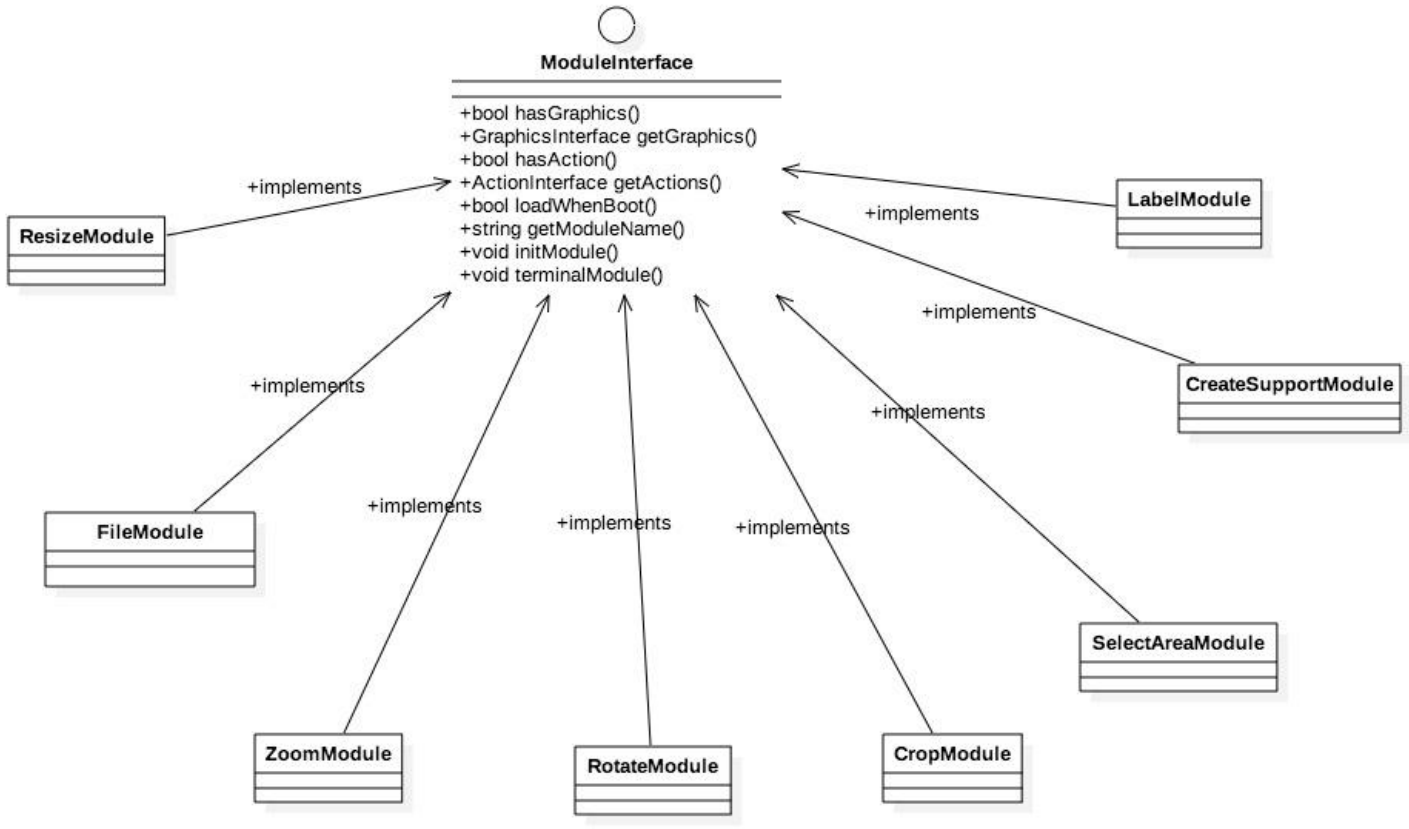
We have 4 types of Dem file. So, we need 4 types of class to read those types. But all of those types will have the same Dem Interface.

- Graphics



In the User Interface, we have 3 basic graphics component includes:
GraphicsMenu: Menu in the top of program. Included a direct way to access functions of program, such as file, view, etc.
GraphicsToolbox: A control panel of program.
GraphicsEditor: An area where 3D Image will be paint.

- Modules



Every module in this program must implements this interface, to register a module.

IV Test cases

Test Case	Test Title	Test Summary	Pre-condition	Test Steps	Expected Result	Actual Result	Post-condition	Status	Notes
TC1	Install application on Linux - Ubuntu	Test on all linux distributions		Install and uninstall app on Linux	Success		The TC1 must be succeeded. The user has to know how to use linux command line to install and run an application.		
TC2	Select a DEM files to open	Try to open USGS, SDTS, DTED and DIMAP files format, show 3D image.	Application run, no DEM file opened yet	Open all DEM files format one by one.	Success - Show 3D image		3D image is drawn and appeared on the main screen		
		Try to open other format files (not DEM file), corrupted DEM file, show error messsage		Open other format files or corrupted DEM file.	Fail - Show error message				
TC2.1	Open a first DEM file			1. Click on "Open" button	The system pops up a file chooser from the user's home directory				
				2. Choose a DEM file, then click OK	The system displays a message "DEM file successfully imported!"		3D image is drawn and appeared on the main screen		
TC2.2	Open second DEM file.			1. Click on "Open" button	The system pops up a file chooser from the user's home directory				
				2. Choose a DEM file, then click OK	The system displays a message "A DEM file already imported. Open another DEM file will overwrite the existing data. Do you want to continue?"				
				3.1 Click "Yes"	The system displays a message "DEM file successfully imported!"		Old 3D image is erased. New 3D image is drawn and appeared on the main screen		
				3.2 Click "No"	The system exits the message		The data remains the same as pre-condition		
TC2.3	Open non DEM file			1. Click on "Open" button	The system pop ups a file chooser from the user's home directory				
				2. Choose a non-DEM file, then click OK	The system exits file chooser and displays a message "Illegal file type. Choose DEM file only"		The data remains the same as pre-condition		
TC3	Rotate and move single point	Rotate or not	A 3D image existed.	The rotation by mouse including move up, down, left, right or angles	Success		The image is rotated as mouse's movement		
TC3.1	Rotate and move multiple point			1. Click any point and rotate and move that point	The application must run smoothly				
				2. Click other point and rotate as the first point, including movement	The second click must be detected as new point				
TC4	Zoom 3D Image	Zoom a 3D image or not	A 3D image existed.	The zoom-in by double clicking on that point and zoom-out by CRTL+Click on that point	The image is zoomed as the mouse's movement.		The image is zoomed-in as mouse's movement		
TC5	Select an area in 3D image	Select any area on 3D Image or not	A 3D image existed.	Using mouse to draw a boundary box on 3D Image to select this area	A boundary box is drawn and the area is selected		A boundary box is drawn and the area is selected		
TC6	Resize a boundary box on 3D Image	Click on a boxes, drag the mouse to Resize the seleced area.	A 3D image existed. At least 1 boudary box was drawn.	Move this 3D Image boundary box	Selected area is resized by mouse's movement		Selected area is resized by mouse's movement		
TC7	Crop a boundary box on 3D Image	Select a boundary box and crop the image (delete all parts of image which is outside of the box	A 3D image existed. At least 1 boudary box was drawn.	Select a boundary box and select crop funtion	Delete all parts of image except inside the box.		Delete all parts of image except inside the box.		

Test Case	Test Title	Test Summary	Pre-condition	Test Steps	Expected Result	Actual Result	Post-condition	Status	Notes
TC8	Create a support at the basis of 3D image	Create a support at the basis of current 3D image by clicking the "Create support" button. With the support, the 3D object should be ready to be printed by a 3D printer.	A 3D image existed.	Use "Create support" function	A support is created.		An object is created and ready to be printed by 3D printer.		
TC9	Make label for 3D object	Use "Add Label" function on popup menu.	A 3D object existed	Click "Add label" function, input text and select position: bottom or top	A label is drawn at bottom or top of the object.		A label is drawn at bottom or top of the object.		
TC10	Export to OBJ or STL	Try to write OBJ or STL format file from the object.	A 3D object existed	Click "Export" function, write file name and select the format: OBJ or STL	A right format file is written		A right format file is written		

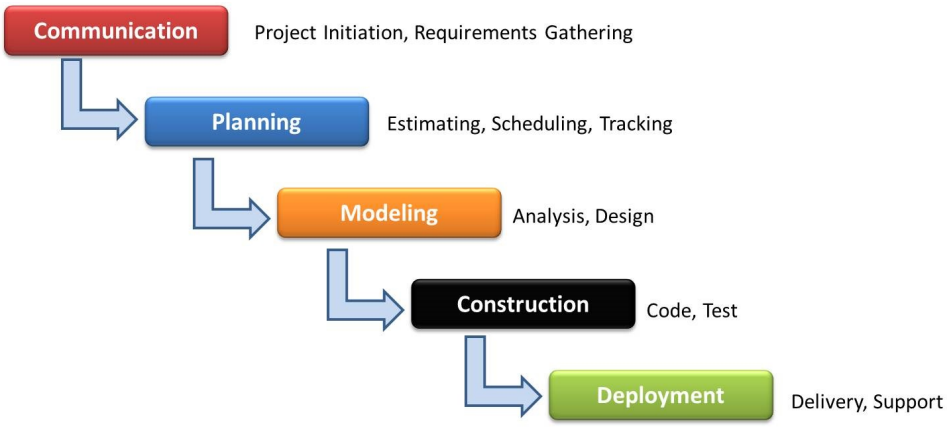
V Tasks assignments

The software engineering project is scheduled with milestones:

1. 15 April: Report on Bibliography, existing analysis
2. 6 May: Report on Requirement analysis and objective
3. 3 June: Architecture and tests: First report completion
4. 26 August: Development: Final report

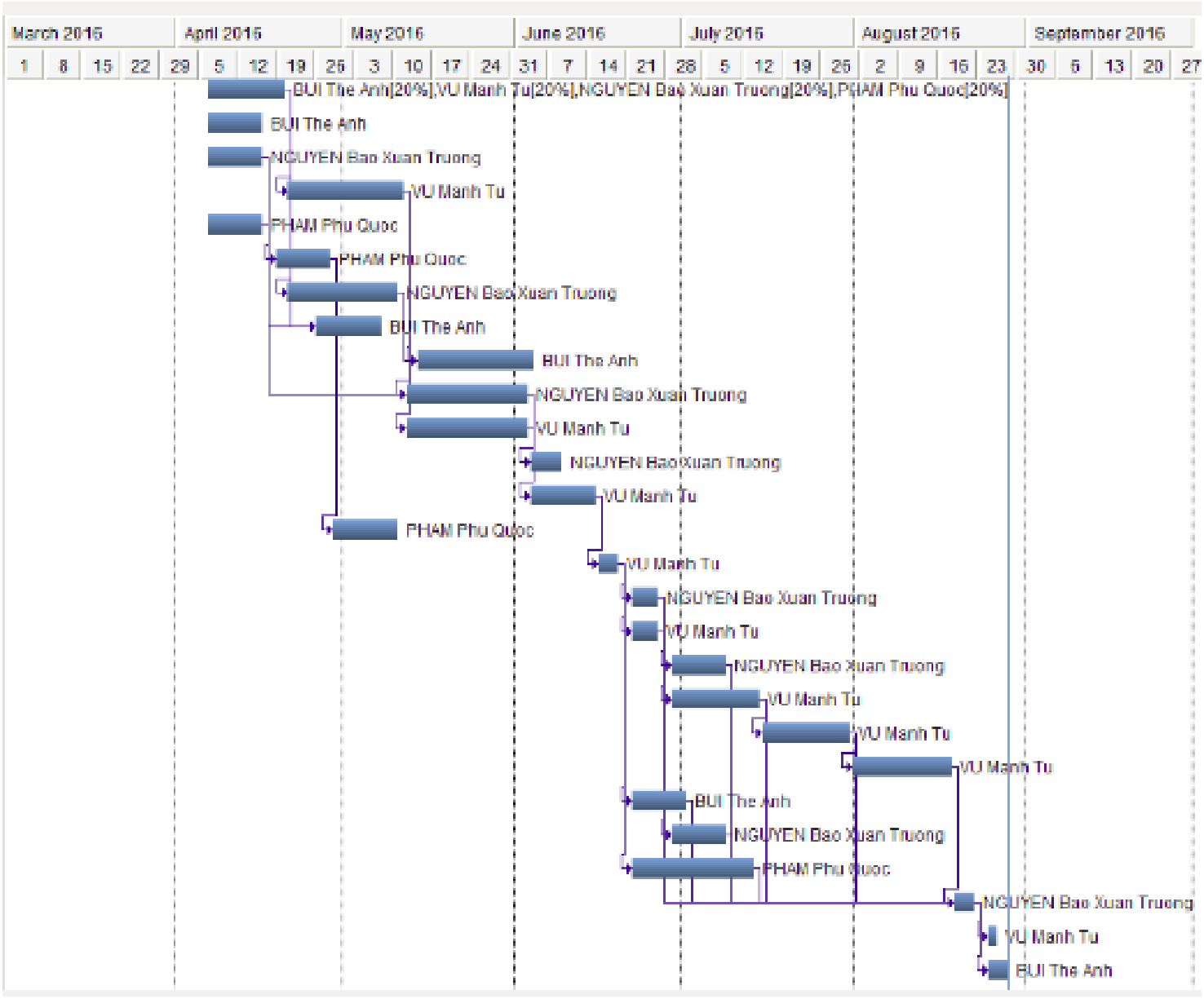
The project duration is 5 months, starting at 6 April and ending at 26 August.
We consider to choose the Waterfall model (see figure below) to manage this project because:

- The time is short: despite the official duration of 5 months, we only have about 2-3 months to work on project due to the overlapping schedules of university and/or companies.
- The requirements are quite clear and fixed.
- The software is not too complex.



The following figures of Task list and Gantt chart shows how the project is controlled and tasks are organized in the team.

	Name	Duration	Start	Finish	Prede	Resources
1	Analyse Requirements	10d	06/04/2016	19/04/2016		BUI The Anh[20%],VU Manh Tu[20%],NGUYEN Bao Xuan Tru
2	1st Report - Bibliography	8d	06/04/2016	15/04/2016		BUI The Anh
3	Research DEM formats	8d	06/04/2016	15/04/2016		NGUYEN Bao Xuan Truong
4	Design Architecture	15d	20/04/2016	10/05/2016	1	VU Manh Tu
5	Research Qt Framework - Hello World	8d	06/04/2016	15/04/2016		PHAM Phu Quoc
6	Research Qt Framework - Draw 3D image using OpenGL	8d	18/04/2016	27/04/2016	5	PHAM Phu Quoc
7	Write Test Cases	14d	20/04/2016	09/05/2016	1	NGUYEN Bao Xuan Truong
8	2nd Report - Software Requirement	10d	25/04/2016	06/05/2016	1,3	BUI The Anh
9	3rd Report - Architecture and Test	15d	13/05/2016	02/06/2016	4,7	BUI The Anh
10	How to read DEM file ? - GDAL library	16d	11/05/2016	01/06/2016	3,4	NGUYEN Bao Xuan Truong
11	Implement Architecture	16d	11/05/2016	01/06/2016	4	VU Manh Tu
12	Test GDAL library	4d	02/06/2016	07/06/2016	10	NGUYEN Bao Xuan Truong
13	Implement File module: Read, write DEM file	8d	02/06/2016	13/06/2016	11	VU Manh Tu
14	Program Interface : Implement QT Windows, Menu, Buttons	8d	28/04/2016	09/05/2016	6	PHAM Phu Quoc
15	Draw 3D image from Raster data	4d	14/06/2016	17/06/2016	13	VU Manh Tu
16	Find other ways to calculate height of points in raster	5d	20/06/2016	24/06/2016	15	NGUYEN Bao Xuan Truong
17	Rotate 3D image	5d	20/06/2016	24/06/2016	15	VU Manh Tu
18	Zoom 3D image	8d	27/06/2016	06/07/2016	16	NGUYEN Bao Xuan Truong
19	Select an area in 3D image	12d	27/06/2016	12/07/2016	17	VU Manh Tu
20	Resize selected area in 3D image	12d	13/07/2016	28/07/2016	19	VU Manh Tu
21	Crop a selected area in 3D image	12d	29/07/2016	15/08/2016	20	VU Manh Tu
22	Create a support in the basis of 3D image	8d	20/06/2016	29/06/2016	15	BUI The Anh
23	Draw label in 3D image	8d	27/06/2016	06/07/2016	16	NGUYEN Bao Xuan Truong
24	Export to OBJ, STL file formats.	16d	20/06/2016	11/07/2016	15	PHAM Phu Quoc
25	Do user acceptance test.	4d	16/08/2016	19/08/2016	17,18,	NGUYEN Bao Xuan Truong
26	Build final project	2d	22/08/2016	23/08/2016	25	VU Manh Tu
27	Final Report	4d	22/08/2016	25/08/2016	25	BUI The Anh


March 2016
April 2016
May 2016
June 2016
July 2015
August 2015
September 2016
1 8 15 22 29 5 12 19 26 3 10 17 24 31 7 14 21 28 5 12 19 25 2 9 16 23 30 5 13 20 27
BUI The Anh[20%],VU Manh Tu[20%],NGUYEN Bao Xuan Truong[20%],PHAM Phu Quoc[20%]
BUI The Anh
NGUYEN Bao Xuan Truong
VU Manh Tu
PHAM Phu Quoc
PHAM Phu Quoc
NGUYEN Bao Xuan Truong
BUI The Anh
BUI The Anh
NGUYEN Bao Xuan Truong
VU Manh Tu
NGUYEN Bao Xuan Truong
VU Manh Tu
PHAM Phu Quoc
VU Manh Tu
NGUYEN Bao Xuan Truong
VU Manh Tu
NGUYEN Bao Xuan Truong
VU Manh Tu
VU Manh Tu
VU Manh Tu
BUI The Anh
NGUYEN Bao Xuan Truong
PHAM Phu Quoc
NGUYEN Bao Xuan Truong
VU Manh Tu
BUI The Anh

VI Implementation

1 Structure Overview

Application is divided into modules. Each module handle a different task. All modules have to be implemented main module file which inherits from Module Interface

```
class ModuleInterface
{
public:
    virtual bool hasGraphics() = 0;
    virtual GraphicsComposite* getGraphic() = 0;
    virtual bool hasAction() = 0;
    virtual ActionInterface* getAction() = 0;
    virtual bool loadOnBoot() = 0;
    virtual QString getModuleName() = 0;
    virtual void initModule() = 0;
    virtual void terminalModule() = 0;
};
```

Module can have Graphics or Action depending on the property of that module. If using Graphics, it is needed to inherit from GraphicsComposite abstract class and declare in the system (method hasGraphics returns true).

```
class GraphicsComposite
{
protected:
    GraphicsComposite* graphicsMain;
    vector<GraphicsComposite*> graphics;
    ActionListener* actions;
    vector<Menu*> menus;
    const int GRAPHICS_ID = -1;
    DemInterface* demObject;

public:
    void setMainGraphics(GraphicsComposite*);
    GraphicsComposite* getMainGraphics();
    void setActionPerform(ActionListener* action);
    ActionInterface* getAction(QString);
    void registerGraphics(GraphicsComposite* graphic);
    int addToGraphicsComponent();
```

```
    bool addMenu(QString menuName, QAction *action);
    bool registerMenu(QMenuBar*);
    virtual void setSize(int width, int height);
    virtual void updateGraphics();
    virtual void updatePaintGL();
    QAction* createQAction( QString name );
    virtual void addVertex(Vertex vertex, int col, int row);
    void setDemObject(DemInterface* dem);
    virtual QSize getSize();
    virtual void mousePressEvent(QMouseEvent *event);
    virtual void mouseMoveEvent(QMouseEvent *event);
    virtual void mouseDoubleClickEvent(QMouseEvent *event);
    virtual void initial() = 0;
    virtual void initializeGL() = 0;
    virtual void paintGL() =0 ;
    virtual void resizeGL(int width, int height) = 0;
};
```

Abstract Graphics Composite provides basic functions for the module to interact with system. For example, to create new menu, the module only calls: bool addMenu(QString menuName, QAction *action);

Similarly, if using Action, it is needed to inherit from Action Interface and declare in the system.

```
class ActionInterface
{
private:
    ActionListener* actionPerform;
public:
    virtual void initAction() = 0;
    virtual void terminalAction() = 0;
    virtual QString getActionName() = 0;
    virtual bool loadOnBoot() = 0;
    virtual bool requireDemObject() = 0;
    virtual void setDemObject(DemInterface*) = 0;
    virtual DemInterface* getDemObject()=0;
    virtual void setGraphics(GraphicsComposite*)=0;
    virtual void setActionPerform(ActionListener*) =0;
};
```

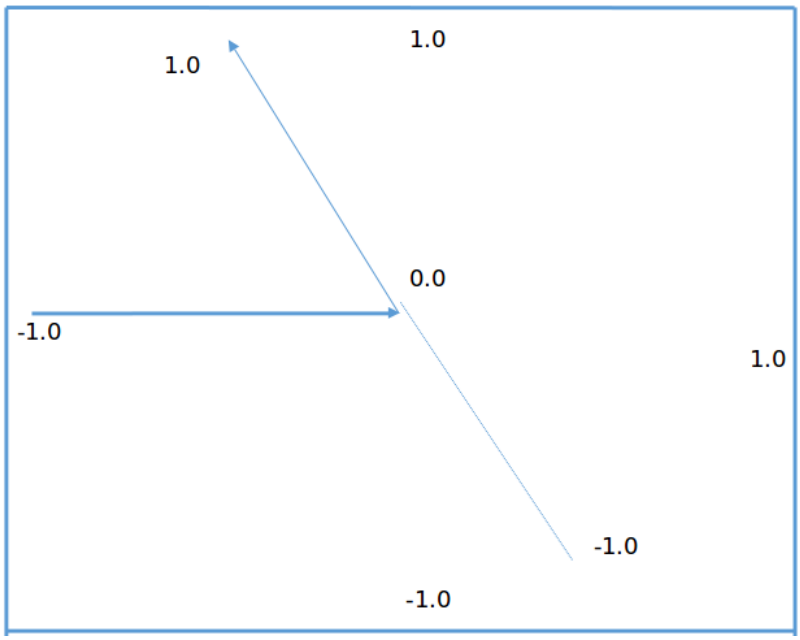
For modules using DEM data, they must have Action. In Action, it must register to use DEM data(requireDemObject function returns true)

and system will automatically provide DEM data via DemInterface.

2 Modules

- a. Module Files: Module Files creates a menu to choose DEM file, read DEM file and generate mesh. It uses GDAL library to read DEM file into Raster data. Raster data is an array of float type which contains height data of each pixel. Based on Raster data and number of cols, rows of DEM file, there is a 2-dimensions array.

  From here, 2-dimensions array's data is changed to Frame of reference of OpenGL with center is base coordinate.



  Calculate x,y coordinates from 2-dimensions array into frame of reference of OpenGL:

  NewX = (X CenterX)/CenterX NewY = (CenterY - Y)/CenterY

  X,Y is coordinates in 2-dimensions array. NewX, NewY : coordinates in frame of reference in OpenGL. CenterX : cols/2 CenterY: rows/2

  Code:

```
for (int k = 0; k < rows; k=k+step) {
```

```
posY = (float)(centerY - k)/centerY;
for (int j =0; j < cols; j=j+step) {
    posX = (float)(j - centerX)/centerX;
```

For Z (height) dimension, the height in 2-dimensions is between -11000 to 9000. However, 9000 is too big to differentiate low/high positions in 3D drawing. Hence, using average = 6000.

NewZ = Data2D[X,Y]/6000;

After identifying coordinates in frame of reference of OpenGL, 3D-interpolating by using triangle:

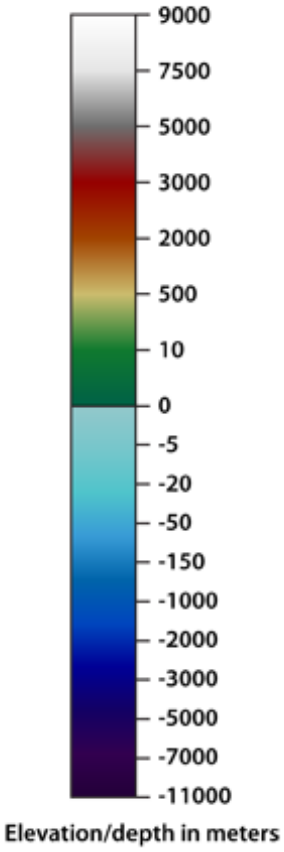
From each point (x,y), find 4 next points: [(y+1, x), (y, x +1)] and [(y-1, x), (y, x-1)] which create 2 triangles, continue for all points in the array.

Identify normal vector: Normal vector is used to identify the direction of light source illuminating the terrain. Based on it, identify dark or light points in the terrain.

It is needed to identify normal vector for each vertex in mesh. When calculate 3 points to draw a triangle, normal vector of that triangle. This vector normal will be assigned to vertices of triangle. When 2 triangles have the same vertex, normal vector of that vertex is the combination of normal vectors of 2 triangles.

```
void DemObject::addVertex(Vertex vertex, int
    position)
{
    int currentPosition = vertexs.size() - 1;
    if (position > currentPosition) {
        vertexs.push_back(vertex);
    } else {
        vertexs[position].setNormal(QVector3D::
            normal(vertex.normal(), vertexs[
            position].normal()));
    }
    ind.push_back(position);
}
```

- b. Color Module: Create color bands based on height of terrain. Files module use these color bands to color for mesh.



Source: http://noaa.maps.arcgis.com/home/item.html?id=feb3c625dc094112bb5281c1767
Each vertex has its height. This Map of height indicates the height by color.

- c. Base Support Module: Create a frame around the mesh. Find all vertices in directions: left, right, bottom & top of mesh. Based on those vertices, draw a triangle to the point having the same coordinate but height = min(height) - 100

- d. Rotate Module: Rotate module catchs right-click and mouse movement, identify from/to points to rotate:

  glRotatef of OpenGL is used to rotate

```
glRotatef(xRot / 16.0, 1.0, 0.0, 0.0);
        glRotatef(yRot / 16.0, 0.0, 1.0, 0.0);
    glRotatef(zRot / 16.0, 0.0, 0.0, 1.0);
```

- e. Zoom Module: Zoom module catchs scroll event of mouse to increase or decrease zoom level. glOtho and glViewport functions are used to zoom

```
int side = qMin(width, height);
glViewport(((width - side) / 2)*zoomByScale,((
    height - side) / 2)*zoomByScale, side*
    zoomByScale, side*zoomByScale);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
float aspectRatio = (float)width/(float)height;
glOrtho(-aspectRatio*zoomByScale, aspectRatio*
    zoomByScale, -1*zoomByScale, 1*zoomByScale,
    1.0, 15.0);
glMatrixMode(GL_MODELVIEW);
```

- f. Move Module: Move module catchs left-click event and movement of mouse to identify the direction and distance.

```
int distanceX = event->x() - lastPos.x();
        int distanceY = event->y() - lastPos.y();
        if (abs(distanceX) < abs(distanceY)) {
            y -= distanceY * 0.0001;
        } else {
            x += distanceX * 0.0001;
        }
```

VII Bibliography

(1) Digital elevation model. Retrieved from `https://en.wikipedia.org/wiki/Digital_elevation_model`. Wikipedia (2016, 9 April).

(2) DEM (Digital Elevation Model) files. Retrieved from `http://vterrain.org/Elevation/dem.html`.Virtual Terrain Project (n.d.)

(3) USGS DEM. Retrieved from `https://en.wikipedia.org/wiki/USGS_DEM`. Wikipedia (2016, March 10).

(4) Digital Elevation Models - Data User Guide 5. Retrieved from http://agdc.usgs.gov/data/usgs/geodata/dem/dugdem.pdf. U.S. Geological Survey (1993).

(5) Gridded Digital Elevation Model Product Specifications, edition 2.0. Retrieved from http://www.geobc.gov.bc.ca/base-mapping/atlas/trim/specs/BC-DEM-specifications-2002-12.pdf. Base Mapping and Geomatics Services Branch - Ministry of Sustainable Resource Management - Province of British Columbia (December 2002).

(6) Terrain Tools - Importing USGS DEM data Example. Retrieved from http://www.softree.com/Tips_Techniques/T-004-USGS-DEM/USGS_DEM.pdf. Softree (n.d.).

(7) Converting and Using SDTS Digital Elevation Model Data. Retrieved from http://www.esri.com/news/arcuser/0799/webdata6.html. Mike Price, ESRI (n.d.).

(8) What is SDTS?. Retrieved from http://mcmcweb.er.usgs.gov/sdts/whatsdts.html. U.S Geological Survey (n.d.).

(9) View the SDTS Document. Retrieved from http://mcmcweb.er.usgs.gov/sdts/standard.html. U.S. Geological Survey (n.d.).

(10) Working with Digital Elevation Models and Digital Terrain Models in Arcmap 9. Retrieved from http://www.lib.uwaterloo.ca/locations/umd/documents/WorkingWithDEM-DTMinArcMap.pdf. David Findlay (June 2005).

(11) DTED. Retrieved from https://en.wikipedia.org/wiki/DTED. Wikipedia (2016, 31 January).

(12) Digital Terrain Elevation Data (DTED) - Part 1. Retrieved from http://www.mission-planning.com/DTED_Part1.htm. Paul, Pablo's Mission Planning Website (n.d.).

(13) Digital Terrain Elevation Data [DTED]. Retrieved from http://fas.org/irp/program/core/dted.htm. Federation of American Scientists (n.d.).

(14) Performance specification Digital terrain elevation data (DTED). Retrieved from https://dds.cr.usgs.gov/srtm/version2_1/Documentation/MIL-PDF-89020B.pdf. U.S. Department of Defense (2000, 23 May).

(15) DTED files (Digital Terrain Elevation Data). Retrieved from http://vterrain.org/Elevation/dted.html. Virtual Terrain Project (n.d.).

(16) The DIMAP format. Retrieved from http://www.geo-airbusds.com/en/196-the-dimap-format. Airbus Defense & Space (n.d.).

(17) Dimap structure. Retrieved from http://www.spotimage.fr/dimap/spec/documentation/3.Dimap_structure.htm. Airbus Defense & Space (n.d.).

(18) The IDE QtCreator. Retrieved from http://www.qt.io/ide/. The QtCompany (2016).

(19) Richard S., Nicholas H., Graham S. and Benjamin Lipchak. OpenGL Superbible comprehensive tutorial and reference. 5th edition. Addison-Wesley, 2010. Print.