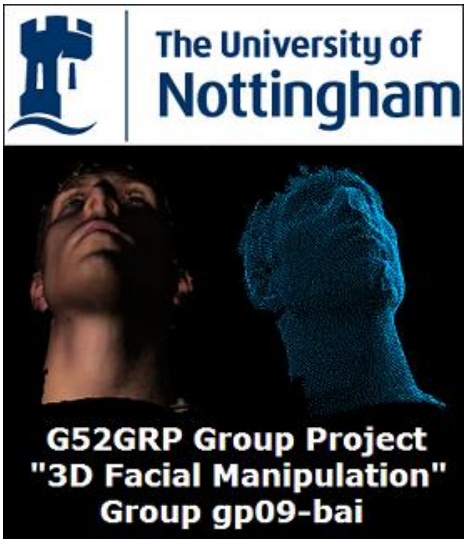
G52GRP Interim Group Report



Project Title: 3D Facial Manipulation

(***Formerly: Creation of Database of Faces)***

Date: 4th December 2009

Group ID: gp09-bai

Name	Computer Science UserID	E-Mail Address
Daniel Hakimi-Nayeri	dxh18u	psycdh@nottingham.ac.uk
Alexander James McNeil	ajm08u	psycrajm@nottingham.ac.uk
Aaron Polley	axp08u	psycrap@nottingham.ac.uk
Danni Wu	dxw09u	psyddw@nottingham.ac.uk

Supervisor: Dr. Bai Li

Contents

Introduction

Aim

The aim of this project is to create a 3D facial manipulation application for matching and animation. With the aid of a 3D scanner, the system should be able to process a scanned face and display it within an application.

Analysis of current system

The current system (3DMD) has state of the art facial scanning software and high quality cameras. With these tools it is able to produce high quality face imagery, scan thousands of point coordinates and manipulate the data i.e. change the structure of a face. These are only some of the many operations 3DMD can perform. It is widely used for medical purposes, such as monitoring facial structures of patients over time. This means being able to predict the facial anatomy after a certain type of treatment, greatly improving any kind of surgery and treatment planning.

Problems to be solved

The new system that is to be implemented has got to be able to do similar actions to the current system and be able to compare and calculate the similarities between two faces. The comparisons between two faces will be implemented in the final version of the system rather than the prototype. A point loader for the face images will also be implemented to calculate the measurements and symmetry of the face.

Solution

The new system will be implemented in C++ and OpenGL. The reason that OpenGL is well suited for this system is because it's fast and where hardware acceleration exists, then high quality graphics can be shown in real-time. Also OpenGL has complex and sophisticated features such as textures, shadows and lighting effects.

Using a 3D scanner provided, the image taken will be stored onto the computer it is connected to. From there the face images will be loaded and displayed within an application, with the ability of rotating the image. In the application the user will be able to switch between to images other than face images, such as a torus or a tetrahedron.

Design

Functional requirements

1) The system shall be able to collect 3D data via 3DMD scanner.
2) The system shall convert the 3D data to various kinds of facial models (point, wireframe and solid).
3) The user shall be able to transform from one kind of facial model to another.
4) The system shall allow user to morph the 3D face model shape by dragging the global control point on the 3D face model.
5) The user shall be able to change the light condition of the face model.
6) The 3D facial model shall be enlarged, reduced, rotated or parallel moved.
7) The system shall be able to match two 3D facial models.
8) The user shall be able to perform the decomposition of a 3D facial model.
9) The system shall provide an undo and redo option for the user to manipulate on the facial model.
10) The system shall save the changes to the original model when required to.

Non-functional requirements

Product requirements

1) The transformation process from the OBJ data file to solid, wireframe or point representation of the 3D facial model shall not exceed 1 second.
2) The system shall be able to work in different operating systems such as Macintosh, Windows and Linux.
3) The size of the whole system shall not overweight 500 MB.
4) There shall be at least 10 help frames.
5) The probability of data corruption shall be lower than 20%.
6) The system shall be constructed modularly in order to facilitate the update of the system.

Organizational requirements

1) The system development process and deliverable document shall confirm to the process and deliverable defined in IEEE standards.

External requirements

1) Personal information related to the 3D data model should be protected according to Data Protection Act
2) Any publication related to the data sample shall not be allowed without the permission of the sample respondent.
3) The system shall have available safeguards from viruses, Trojan Horses, etc.
4) The system shall provide various kinds of language options such as English, Chinese and French.

Domain requirement

1) The parametrical surface in three dimension is defined by three bivariate functions:

$$Q(u,v) = (X(u,v), Y(u,v), Z(u,v)) \quad \text{where } 0 \le u \le 1, 0 \le u \le 1.$$

2) The surface curve (Bézier curves) of degree n defined by n + 1 control points $P_0, P_1, P_2, \ldots$ and P_n is

$$C(u) = \sum_{i=0}^{n} B\,n, i\,(u) Pi$$

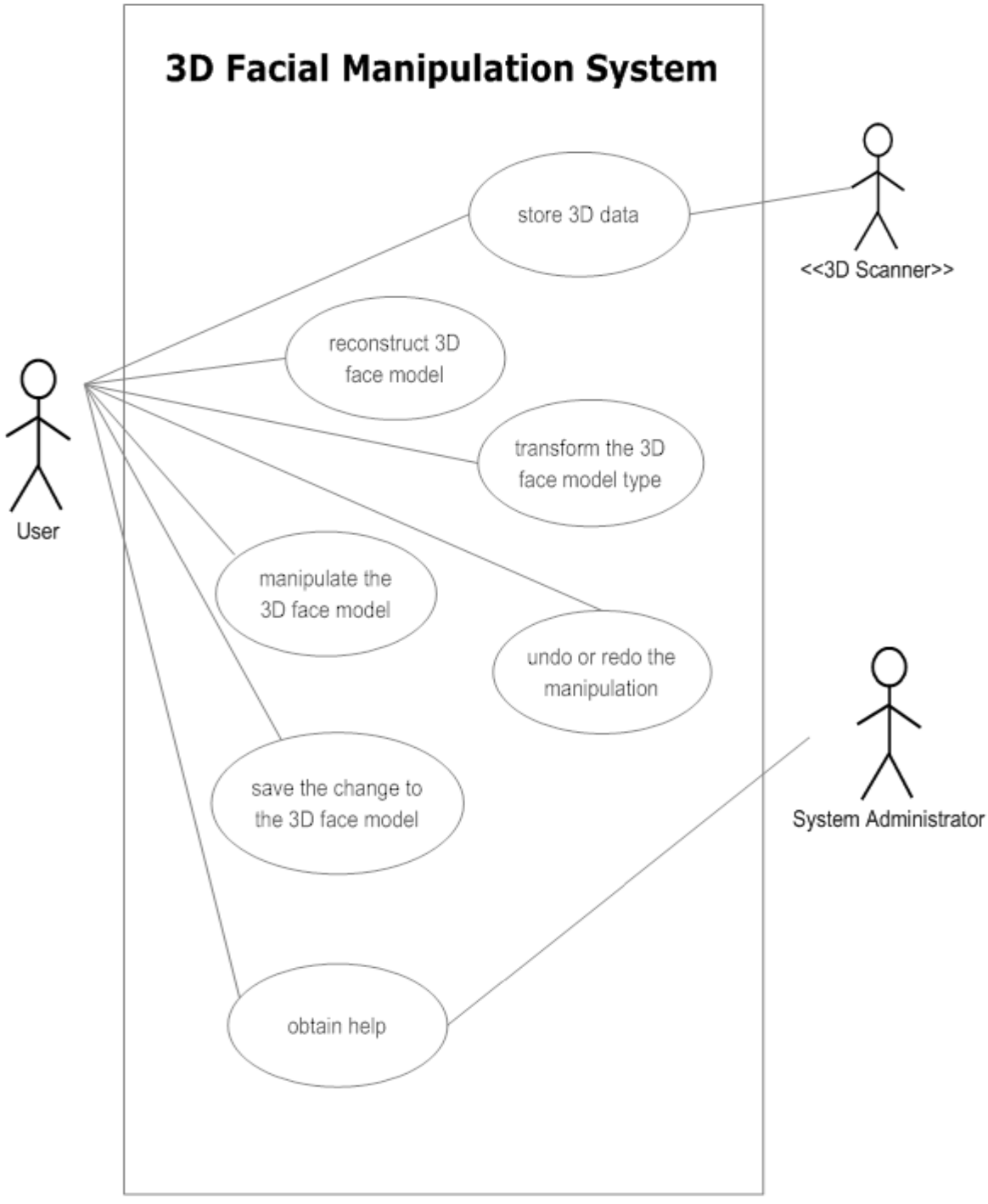
$$\text{Where } B_{n,j}(u) = \frac{n!}{i!(n-i)!}\, u^i (1 - u)^{n-i}$$

User requirements

The user should be able to store the 3D facial model data of the respondents via a 3D scanner. The 3D model data shall be able to export to different kinds of data file such as OBJ, RAW, BMP or WRL. By using OpenGL as rendering engine, the OBJ file is loaded to enable the reconstruction of the 3D face in different representations such as solid, wireframe or point via different rendering modes. Once the face is reconstructed, the user could do various manipulations on the modelling face by pressing different keys. For example, the user could change the light condition of the 3D face model, transform from one presentation model (e.g. solid) to another (e.g. wireframe).

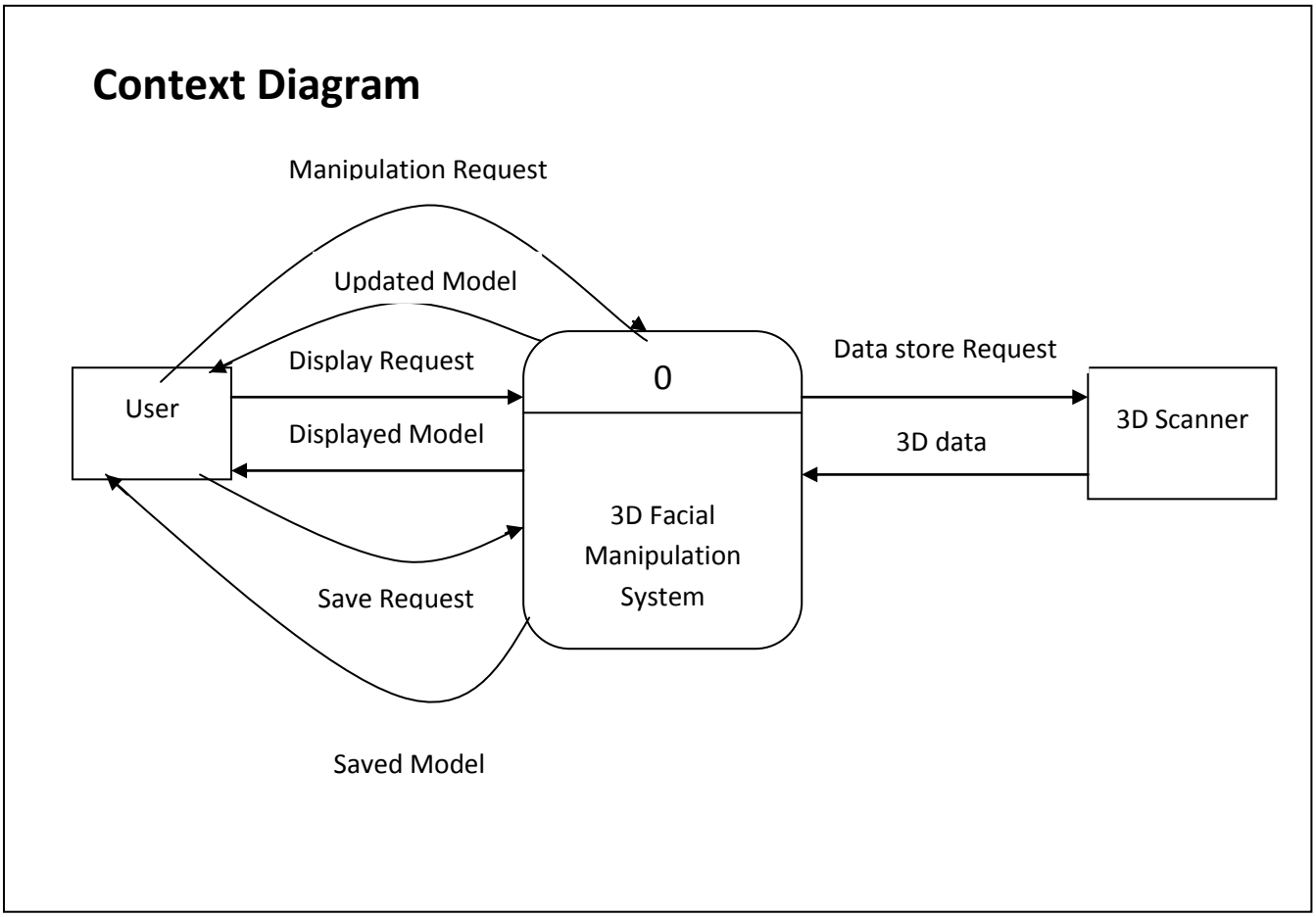
Moreover, the user can do morphing manipulation on the 3D point facial model. Provided with various kinds of options displayed on the screen, the user can rotate the model, move the model up, left, down, right respectively. In addition, the user can zoom in or out the face model by pressing Z(zoom in), Q(zoom out). The colour of the 3D face model could be altered, and the user has the option to make the 3D point model disappear or reappear. The user can save the modification on the 3D facial model data file. (At the final stage, the user could even decompose the 3D facial model and form a new 3D facial model. When encountering any problem, the user could seek help by clicking on the help button on the screen.)

Use Case


3D Facial Manipulation System
store 3D data
reconstruct 3D face model
transform the 3D face model type
manipulate the 3D face model
undo or redo the manipulation
save the change to the 3D face model
obtain help
User
<<3D Scanner>>
System Administrator

DFD

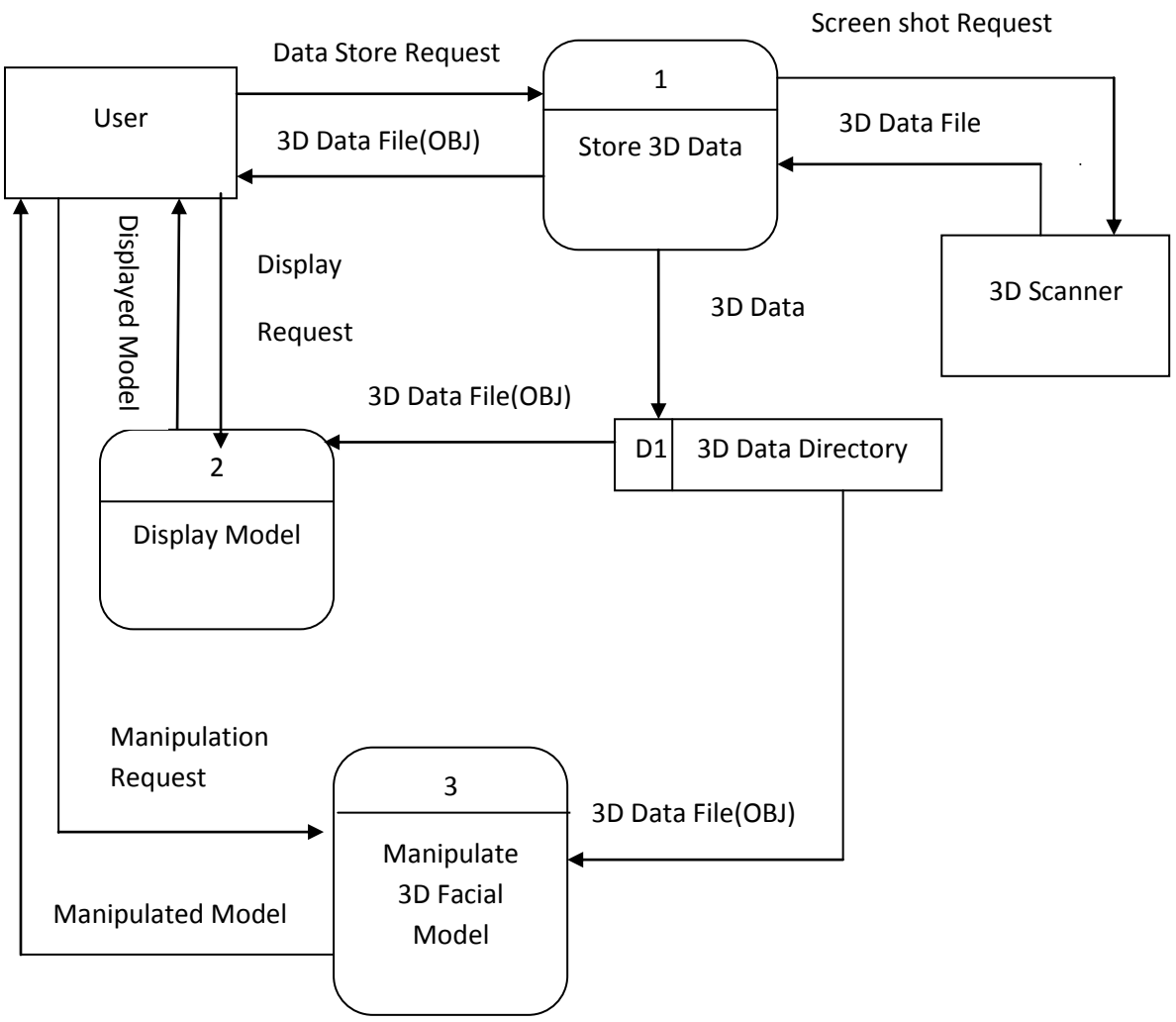
Process modelling



The context diagram shows the main data flow between the external entity (user) and the designed system. Not all use cases are included here such as obtain help, decompose 3D facial model and undo or redo the manipulation because this functions has not been designed yet at this stage. The construction of the context diagram and DFD will be iterated as the project processes.

Level 0 DFD is an elaboration of the context diagram. It shows the main processes the user may interact with.

Level 0 Data Flow Diagram



Architecture Design

Operational Requirements

The technical environment that the application will be presented with is a computer with a minimum of a 1GHz CPU and 512MB of RAM in addition to a graphics card that supports OpenGL 2.0 or above, the application will integrate with the existing operating system software but will however run full screen preventing the use of other applications. It will be possible to run this application on multiple platforms as the code is written in C++ and this is supported on Windows, Mac and Linux operating systems. It will be fairly easy to maintain the application as the application will have only one code base that can just be compiled for different operating systems without modification.

Performance Requirements

Speed

The speed of the application is important to maintain control of the program, there are a number of minimum performance expectations that are outlined below:-

Action: Loading model
Time: This action should take no longer than 5-10 seconds depending on Model size, depending on the machine speed.

Action: Saving model
Time: This action should take no longer than 5-10 seconds depending on Model size, depending on the machine speed.

Action: Menu System
Time: Viewing menu items should take no longer than 2 seconds.

Action: Rotating, Moving, Zooming
Time: These actions should be quick, the model should update in no longer than 0.01 seconds when these actions are performed, depending on the machine speed.

Action: Morphing between models
Time: This action should take no longer than 30 seconds, depending on Model size and depending on the machine speed.

Capacity

The program should preferably have the following capacity requirements:

Hard Disk: Should take no more than 5GB of storage space (depending on amount of face data).

Memory: Should use no more than 256MB of memory.

Availability and Reliability

The program should be available for use at any time that the user would like to use the program. The program should also of course be reliable, the program must not have memory leaks, and all actions must be able to be completed without the stability of the application being affected.

Security Requirements

System Value

The programs value is dependent entirely on the 3D Face Data that is being used, someone's personal data is extremely valuable. Thus, of course data security must be up kept in all circumstances, and security of the system must be paramount.

Access Control

The program will be accessible by only those who need to use the program to manipulate 3D Face Data, all parties whose data is used will be contacted before the program is used, to ensure they are happy with their personal data being used for these purposes.

Encryption/Authentication

The system will not be encrypted, and will not require authentication for usage, however those will access to the program will secure it's use by only allowing certain persons to use the program, and the copying of the program and 3D Data will be strictly prohibited.

Virus Control

The system must be guarded against viruses and other external sources of misuse, regular virus and malware checks will be performed on any machines that the program runs on, and ensuring the security and integrity of the program is upheld.

Cultural and Political Requirements

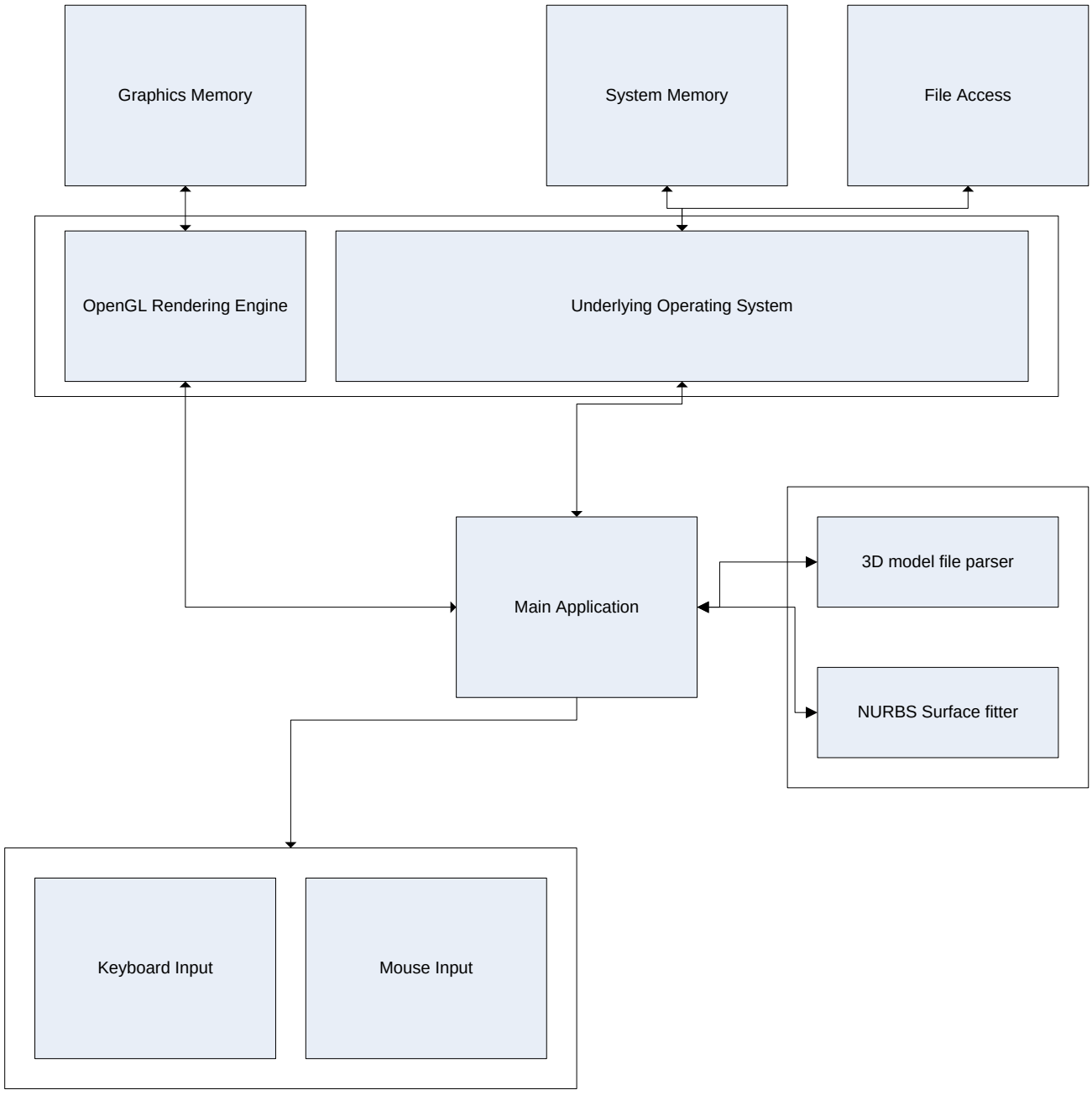
Although it would be an asset to the application if it was able to be used in multiple languages it is currently beyond the costing of this project to hire a interpreter to translate all of the text in the program into multiple different languages, you will be able to customize the faces loaded into the application however you will not be able to customize any of the application itself due to the fact that this would require the over complication of this project. In legal terms this application will comply with data protection laws as required and has not used any copied or reverse engineered code in its construction.

Chosen Architecture

For this project we have chosen to go with a Monolithic Application architecture simply because this allows us to produce one large application that is able to perform the required tasks without having the codebase complicated with extra module loaders and other such checks related to loading additional modules at runtime, it is also hard to see how any other architecture would benefit the project.

Program Design

Structure Chart



Program specification

Module name	Purpose	Deadline	Programmer	Programming language
3D model loader	Load in the 3D model form file	11 DEC	Aaron/Alex	C++
NURBS surface fitter	Create a NURBS surface from 3D file data	28 JAN	Aaron	C++
Application GUI	Allow easy manipulation of NURBS Surface	10 FEB	Aaron	C++
File Open Dialog	Allow users to select the 3D face to be loaded	28 JAN	Alex	C++
Save file interface	Allow the user to manipulate a face and then save the 3D face out to a file	20 FEB	Aaron	C++

The program does not have many events to deal with however it must deal with key presses from the keyboard and also events generated by the mouse. Both of these types of event are handled by the program and are used to manipulate the face on the screen in real time, i.e. you will be able to drag a NURBS surface about using the NURBS control points.

Input	Description
3D file	Inputs the 3D model data
Keyboard	Key presses form the keyboard
Mouse	Mouse movement & mouse clicks

Output	Description
3D file	Saves the 3D model data

Display	Rendered 3D model on screen

Pseudo code

```
Application start{
        Setup OpenGL();
        Add Mouse input();
        Add keyboard Input();
        Show file open dialog();
        Load 3D file();
        Parse 3D file data();
        Setup user interface();
        While(esc !=true){
              Process NURBS();
              Render scene();
              Check Inputs();
              If(save == true)
                    Save3Dmodel();

            }
            Unload 3D data();
            Cleanup();
}
```

User Interface

User Scenario

Below are the user scenarios for the Prototypes, there are two prototypes, one which loads various 3D Objects, included the 3D face and textures it. This program will be referred to as the "3D Face Loader". Next there is a prototype that loads a 3D face and displays points; this will be referred to as the "3D Point Loader". User scenarios for the final program, I.E the ones we will actually implement are listed later.

Prototypes

There are a number of scenarios that a user may perform, including:-

- **Scenario:** Loading the object file

- Action: In the "3D Face Loader" currently the user loads an object by using a right-click menu and selecting "Load obj" from the menu, in the "3D Point Loader" the face is loaded automatically.

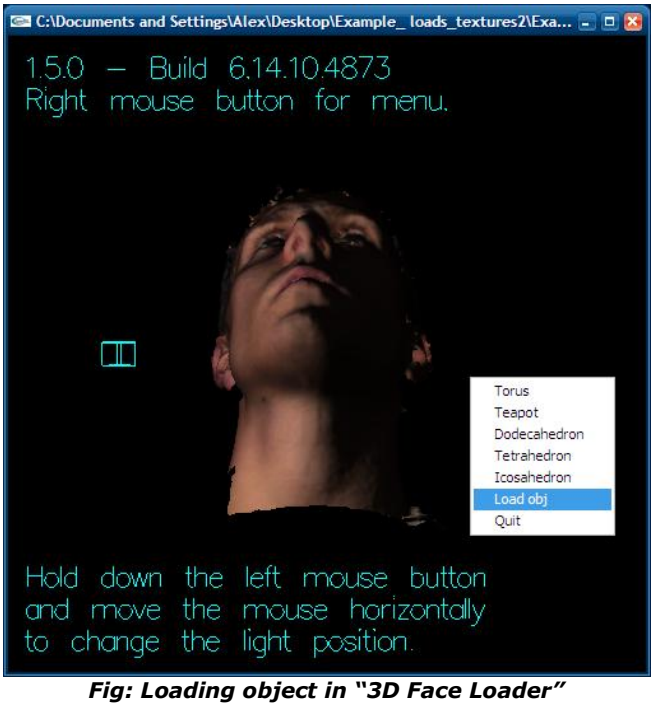


Fig: Loading object in "3D Face Loader"

- Scenario: Manipulating light position

- Action: This only applies to the "3D Face Loader", here the user clicks with the left-mouse button and moves the mouse on the X axis to move the position of the light, when the mouse is not in contact with a control point.

- Scenario: Changing model

- Action: This only applies to the "3D Face Loader", here the user can change which model is displayed by using the right-click menu and selecting a different object.

- Scenario: Rotating Model

- Action: This only applies to the "3D Point Loader", the user can rotate the model on the X axis by using the Left and Right keys, and can rotate the model on the Y axis by using the Up and Down keys.

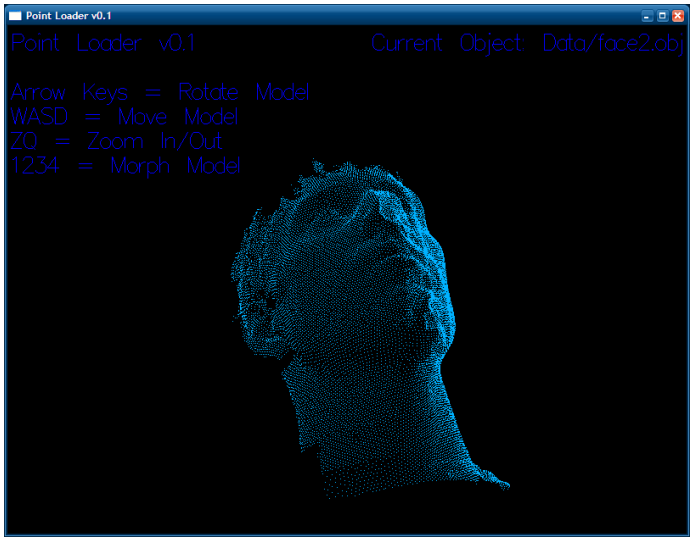


Fig: Rotating object in "3D Point Loader"

- Scenario: Moving Model
- Action: This only applies to the "3D Point Loader", the user can move the model on the X axis Left by pressing the A key and Right using the D key. The user can move the model on the Y axis Up using the W key and Down using the S key.

- Scenario: Model Zoom
- Action: This only applies to the "3D Point Loader", here the user can zoom in on the model to see detail using the Z and Q keys. Z zooms in on the model, and Q zooms out.

- Scenario: Morph Model
- Action: This only applies to the "3D Point Loader", the user can "Morph" between different models using the keys 1, 2, 3, 4 which correspond to different models loaded in the program.

- Scenario: Quitting Application
- Action: In the "3D Face Loader", the user can quit by choosing "Quit" from the right-click menu, or by pressing the [X], a standard component of all Windows. The user quits by pressing the [X] in the "3D Point Loader".

Final Program

There will be no right-click menu in the final program.

<u>These are the user scenarios for the Final Program:-</u>
- Scenario: Loading the object file
- Action: In the final program, loading the object file will be done from an Open File dialog as shown below. The user will browse to the folder they choose, select the file and open it for viewing in the program.

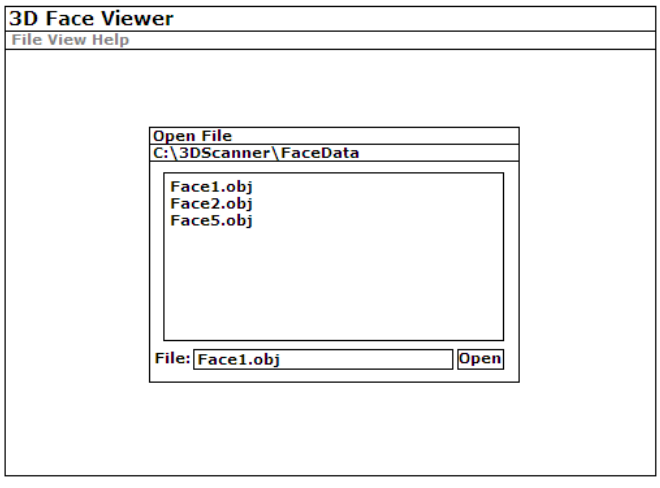


Fig: Final Program "Open File" Dialog

- Scenario: Manipulating light position
- Action: This will work in the same way as the "3D Face Loader".

- Scenario: Changing model
- Action: This will not be in the Final Program. Changing model will mean loading a new model, or "Morphing" between existing models.

- Scenario: Rotating Model
- Action: This will work in the same way as the "3D Face Loader".

- Scenario: Moving Model
- Action: This will work in the same way as the "3D Face Loader".

- Scenario: Model Zoom
- Action: This will work in the same way as the "3D Face Loader".

- Scenario: Morph Model
- Action: This will work in the same way as the "3D Face Loader", however different keys may be used, or the Morphing GUI may be on-screen.

- Scenario: Point Manipulation
- Action: This is an extra scenario; in the final program you will be able to manipulate control points. This will be achieved by left-clicking on points and moving the mouse on the Y and X axis. Movement on the Z axis may be implemented using the mouse-wheel, or by detecting when a point is selected and using the left-mouse button to move the point "Inwards" and the right-mouse button to move the point "Outwards".

- Scenario: Quitting Application
- Action: In the final program the user will be able to Quit by using the File menu and selecting Quit or by pressing the [X] as you can in the prototypes.

Final Program Changes

Of course these scenarios are not final, and may change with the completion of the final program.

Interface Structure

The basic components of the interface are the following:-

Prototype

Mouse

- Component Name: Right-click menu
- Description: This is only used in the "3D Face Loader" prototype. This is a menu-ing system that allows the selecting of different objects for viewing, and also allows the user to quit the application.

- Component Name: Light Movement
- Description: This only applies to the "3D Face Loader", the user can move the light using the left-mouse button, when the mouse is not in contact with a control point.

Keyboard

The following only apply to the "3D Point Loader":-

- Component Name: Arrow Keys
- Description: The user can rotate the model on the X and Y axis using the Arrow Keys (Up, Down, Left, Right)

- Component Name: WASD Keys
- Description: The user can move the model on the X and Y axis using the W,A,S and D keys.

- Component Name: Z and Q Keys
- Description: The user can move the model on the Z axis using Z and Q.

- Component Name: 1, 2, 3 and 4 Keys
- Description: The user can "Morph" between models using these keys.

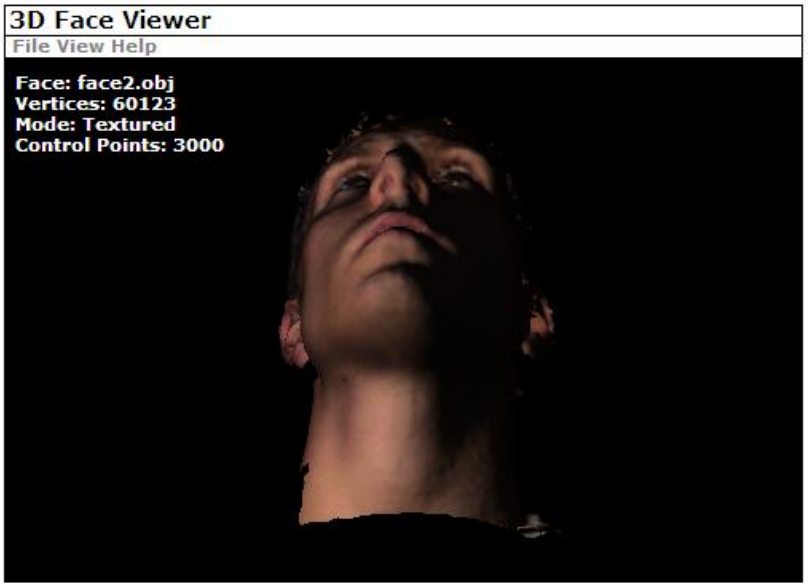
Interface Standards

Metaphor

A metaphor for the user interface, must be that it is WYSIWYG (What You See Is What You Get), since the interface is very visual, instant feedback must be given when an operation is performed. It will use the metaphor of keyboard and mouse interaction, to give the user precise control over the objects manipulation.

Templates

Below is a template of what the user interface is assumed to look like at this point in time:



The file menu will have the following approximate structure:

```
File -> Load Object
      Morph Object
      Save Object
      Quit
View -> Textured
      Wireframe
      Points
      ---------------
      <face-filename>
```

```
      ... to N faces loaded
Help -> Help
      ---------------
      About
```

Note: This is of course subject to change, the final template will be shown in the final report.

Objects

The Main objects will be the following:-

- Main Window
- Open file dialog
- Menu system objects
- Face object
- Text objects
- I/O objects
- Control point objects

Actions

The main actions are detailed in the User Scenario section above.

Icons

The Main icons will be the following:-

- Face icon
- Text icon(s)

Prototyping

As explained previously, there are two prototypes. The language used for the prototypes is C++ and OpenGL, as we feel it is best to prototype in the language and graphics framework that will be used for the final project.

3D Face Loader

The first prototype is what is known as the "3D Face Loader", this prototype loads a number of objects, A Torus, Teapot, Dodecahedron, Tetrahedron, Icosahedron and face Object. It allows the user to select between these objects using a right-click menu and it allows the user to move a light to illuminate an object using the left-mouse button. It also shows the current OpenGL version to the user, the binary filename in the Title, and text on the screen telling the user how to use the program.

A later version allowed the user to switch from a textured mode to a point mode using the W and P keys.

3D Point Loader

The second prototype is what is known as the "3D Point Loader", this prototype loads a Face object, and other objects such as a Cube, Torus, and Sphere. It allows the user to rotate the object using the Arrow Keys, move the model using the WASD keys, Zoom In/Out using the Z and Q keys respectively and to "Morph" from one object to another using 1, 2, 3 and 4 (all points change towards a new location until the first object has "morphed" into the second).

Comment

Elements of each prototype will be used to create a new program, and new features will be added to the new program such as a NURBS surface using control points.

Interface Evaluation

In evaluating the interface, we saw that, some parts of the interface are functional and some are not. For example the right-click menu will not be used in the final version, as it conflicts with other GUI actions. And the user will be able to load Objects from a file by selecting them using an Open File dialog. Also the text output will be different, as for the prototypes having large amounts of text on the screen was important for debugging and ease of use for all group members, however these elements can be delegated to help files in the final version.

Implementation specification

Programming language

The application will be programmed in C++ and OpenGL, these languages were chosen as C++ has a very good interface to OpenGL. OpenGL was chosen as the 3D graphics interface because it is cross platform and allows for very good 3D accelerated graphics on a wide range of hardware.

Operating systems

This application will be able to run on Windows (XP and above), Linux and Apple Mac, however the primary operating system is Windows and as such the vast majority of the testing and use will be on Windows based computers.

Computer

The computer will require any reasonable modern CPU and a reasonable amount of RAM i.e. 1GHz CPU and 512MB RAM.

Additional hardware and software

Additionally the computer hardware will require a graphics card with 3D acceleration capabilities in addition to hardware support for OpenGL 2.0 as a minimum. As most computers come with this type of graphics card we cannot see any significant increase in cost.

Implementation

Methodology

The methodology we utilised for the project was the prototyping methodology. Moving forward developing a prototype was the most efficient way of starting the project. Within a few weeks of the prototype being developed we had some of the aspects of the features that would be in the final prototype program. We used our group supervisor (Dr. Bai Li) as an end-user; this was an important part of the project because we were able to obtain feedback quite early on in the project. From attaining feedback from our supervisor, we were both able to compare the developed program to the final software specification. This was essential to giving us an insight into the correctness of the initial project and made it clear when the deadlines could be met.

We took an evolutionary approach to the prototyping; instead of building many prototypes we just used one well structured one and over the weeks kept refining it. Possibly with more time for the prototype stage we could have used throwaway prototyping. By building on the system we had it was easier to make changes, or add features to it. For example the point loader that was implemented was created from a copy of the prototype and will be merged into the main prototype without causing too many problems. At this part of the project it is useful to have a functional system than no system at all; this is a great advantage with evolutionary prototyping.

By using prototyping we greatly reduced time and improve the quality of the requirements in the final system. There were not many changes in the development which lead to the quick completion time. There are also disadvantages that come with using the prototyping methodology some of which include the confusion of the prototype with the final system. With careful consideration for the two different versions, we will be able to avoid confusing the user by not adding all the features to the prototype. These are features such as security and error–checking that the prototype will not have. Also spending too much time on the prototype will need to be avoided because the whole idea behind prototyping is to finish it quickly, adding too many features will delay a final system being implemented.

Technical and Management Issues

(1) Data:

Due to the limitation of the amount of the experimental 3D data, the prototype may not be tested thoroughly. Also, the data collection could not be one hundred percent successful. The light condition affects the outcome of the data collection a lot. Some parts of the data of the face model may lose due to the reflection of light.

(2) Lack of deliverable and milestones:

Insufficient deliverable and milestones at this stage may lead to the difficulty of tracing back to the initial design when a specific requirement needs to be updated or altered.

(3) Statement of Work:

Since the group member has different culture background, the communication obstacle within the group could result in the different understanding of the project. This may further result in the inconsistency in the separate work accomplishment.

(4) Unbalanced Skill Distribution:

Unbalanced skill distribution is a major concern in the program design phase. Since some team member is proficient in programming, while the others is not so good at it. The main programming task is accomplished by one or two certain person. When it comes to the deliverable stage, the programmer may need to explain his program to the rest group members which is of low efficient.

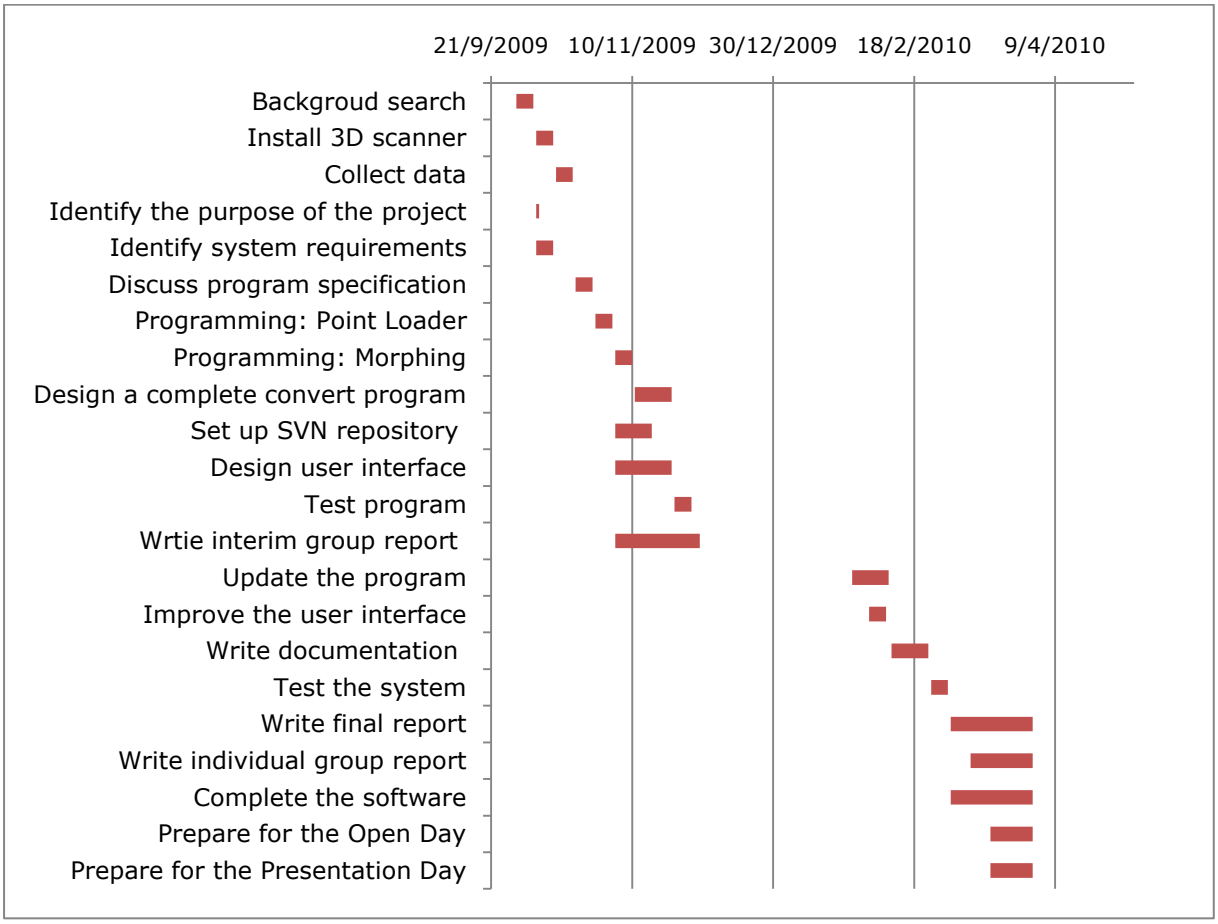
Project scheduling

ID	Task Name	Duration(days)	Start	Finish	Prede.
1	Background search	6	30/9/2009	6/10/2009	
2	Install 3D scanner	6	7/10/2009	13/10/2009	
3	Collect data	6	14/10/2009	20/10/2009	2
4	Identify the purpose of the project	1	7/10/2009	7/10/2009	1
5	Identify system requirements	6	7/10/2009	13/10/2009	1
6	Discuss program specification	6	21/10/2009	27/10/2009	3
7	Programming: Point Loader	6	28/10/2009	3/11/2009	6
8	Programming: Morphing	6	4/11/2009	10/11/2009	7
9	Design a complete convert program	13	11/11/2009	24/11/2009	7,8
10	Set up SVN repository	13	4/11/2009	17/11/2009	
11	Design user interface	20	4/11/2009	24/11/2009	
12	Test program	6	25/11/2009	1/12/2009	10
13	Write interim group report	30	4/11/2009	4/12/2009	4,5,11
14	Update the program	13	27/1/2010	9/2/2010	9
15	Improve the user interface	6	2/2/2010	8/2/2010	10
16	Write documentation	13	10/2/2010	23/2/2010	14
17	Test the system	6	24/2/2010	2/3/2010	15, 16
18	Write final report	29	3/3/2010	1/4/2010	17
19	Write individual group report	22	10/3/2010	1/4/2010	17
20	Complete the software	29	3/3/2010	1/4/2010	17
21	Prepare for the Open Day	15	17/3/2010	1/4/2010	20
22	Prepare for the Presentation Day	15	17/3/2010	1/4/2010	20

The project starts at the beginning of the semester (September 21st, 2009). The group members are required to hold an informal weekly discussion and a formal weekly meeting with the supervisor on Wednesday. The majority of the tasks are displayed on the following Gantt Chart. The members should start from the backgrounds search which facilitates their understanding of their whole project. The tasks do not have to be accomplished strictly according to their task number. However, they do have to start after the precedent task has been finished. Also, it is worthy noticing that there is a Christmas Holiday and assessment period (12/12/09 – 24/01/10). Accordingly, there is a time gap which is shown in the Gantt Chart below. Since the deadline of Open day and Presentation Day is at the end of the Easter holiday, it is better to bring the deadline forward (before the holiday start). Although there appears to be a little bit time clashing for the task (18-22) because they all start at the approximately same time and end on April 1st. The specific

duration of each task may need to be adjusted, whereas, those tasks which are a reflection of all the previous work can be accomplished simultaneously.

Gantt Chart



Appendix A: Meeting Minutes

Meeting No: 1 **Date:** 30th September 2009	**Meeting No:** 2 **Date:** 7th October 2009
Date: Wed 30 September 2009 Time: 1PM Location: Computer Science Atrium Attended: Alex, Aaron, Danni, Daniel Meeting Information: - Introduced ourselves - Talked about the project outline(s) and discussed which project we would like to per sue - Discussed programming experience in the group - Organised formal meeting times Time: 40 Minutes	Date: Wed 07 October 200 Time: 1PM Location: Computer Science Hub & Bai Li's Office C37 Attended: Alex, Aaron, Danni, Daniel Meeting Information: Hub: - Contact Details - Level of programming experience in the group in more detail (we touched on this briefly last week but it was not clear) - Allocation of roles in the group - Revision Control Systems Bai Li's Office C37: - Introducing ourselves to Dr. Bai Li - Talking about the project, and choosing the Database of Faces project - Discussing programming languages, and implementation details - Discussed the use of a Database - Bai Li gave us handouts, and explained how we should start the project Time: Hub: 15 Minutes C37: 35-40 Minutes
Meeting No: 3 **Date:** 14th October 2009	**Meeting No:** 4 **Date:** 21st October 2009
Date: Wed 14 October 2009 Time: 2PM Location: Computer Science Hub Attended: Alex, Aaron, Danni, Daniel Meeting Information: - 3D Model Data - Uses for 3D Model - Please use the mailing list! - Future meeting times - Formal meeting time with Dr. Bai Li. Time: 30 Minutes	Date: Wed 21 October 2009 Time: 12am Location: Computer Science Hub & Bai Li's Office C37 Attended: Alex, Aaron, Danni, Daniel Meeting Information: - Discussed Aarons Obj Loader code - Discussed the need for a Raw to Obj converter (UPDATE: We didn't need this converter eventually) - Use the mailing list! (Again!) - We can't use Git as it won't work correctly on Solaris - We could use SVN instead. - Using FTP for now. Time: 20 Minutes
Meeting No: 5 **Date:** 27th October 2009	**Meeting No:** 6 **Date:** 4th November 2009

Date: Wed 27 October 2009 Time: 12am Location: Computer Science Hub & Bai Li's Office C37 Attended: Alex, Danni, Daniel (Aaron not present - Conference) Meeting Information: - Got Raw data for the Raw to Obj Converter (UPDATE: We didn't need this converter eventually) - Got the software to convert face data - Bezier Surface - Control Points - Mesh - Explained to Dr. Bai Li how the project was progressing so far & what we had done (Aarons Obj Loader) Time: 25 Minutes	Date: Wed 4th Nov 2009 Time: 12am Location: Computer Science Hub Attended: Alex, Aaron, Danni, Daniel Meeting Information: - Discussed: - Told Aaron the details of the previous meeting - The Point Loader code, and what to do with it next - That the Morphing code may need fixing - That we should start the Interim Report now, so that it doesn't have to be rushed in time for the deadline - Asked Danni and Daniel to start work on the Interim Report - Setting up SVN (No git) Time: 35 Minutes
Meeting No: 7 **Date:** 11th November 2009	**Meeting No:** 8 **Date:** 18th November 2009
Date: Wed 11th Nov 2009 Time: 12am Location: Computer Science Hub Attended: Alex, Aaron, Danni, Daniel Meeting Information: - Discussed new changes Aaron made to fix the Point Loader - There are still issues with objects of different sizes - Discussed that the code is designed for objects with the same no. of points, we need to generalise this for any object and any no. of points - Merging Object Loader and Point Loader into 1 program, Aaron said that the Object Loader can be changed to render in Points too - Discussed the need to be able to manipulate points with the mouse Time: 20 Minutes	Date: Wed 18 Nov 2009 Time: 12am Location: Computer Science Hub Attended: Alex, Aaron, Danni, Daniel Meeting Information: - Discussed merging the object loader and the point loader into one program, using the point loader controls but the object loader as a base. - Discussed the use of SVN - An SVN repository has been set up so that we can manage source code more easily - Discussed writing a script to archive the SVN repo to FTP once per day for easy access - Discussed the Interim Report outline Daniel uploaded to the forum - Discussed the need to have a rough 1st Draft of the Interim Report by next week - Looked at Danni's outline of the Interim Report, and commented on how to fill some of the sections Time: 40 Minutes
Meeting No: 9 **Date:** 18th November 2009	**Meeting No:** 10 **Date:** 19th November 2009

Date: Wed 18 November 2009 Time: 12 PM (Noon) Location: Computer Science Hub & Lab (A32) Attended: Alex, Aaron, Danni, Daniel Meeting Information: Discussed: - The Interim Report, and the outline that Danni and Daniel had proposed - The need to create a set of control points for the models (discussed with Bai Li previously) using NURBS - Merging the two programs, ideas on interface - The need for sections of the report to be completed by next Wednesday - New Minutes uploaded for viewing - Need for meeting with Bai Li, for a status update to what we have done so far, and to discuss further how we would like the project to progress from the current point - Looked at the outline Danni had proposed and suggested improvements, and how to split the report into sections (E.g Aaron and Alex would complete sections that are relevant to the programming aspects) Time: 30 Minutes	Date: Thu 19 November 2009 Time: 1 PM Location: Bai Li's Office C37 Attended: Alex, Aaron, Danni Meeting Information: Discussed: - Set up the scanner to take a 3D image of Danni - Discussed new minutes - Showed Dr. Bai Li the status of the project so far * Showed Dr. Bai Li the object loader that Aaron wrote that loads and textures a model * Showed Dr. Bai Li the points loader that Alex wrote that loads a model and shows points, and morphs between models * Discussed how we will merge these two programs, and create control points - Dr. Bai Li explained what control points are, and why they are useful for the project - We discussed possible uses for the project: * E-fit for the police * Matching faces (e.g. for Airport security) * Facial recognition (and the algorithms to perform this) - We thought about the need for (possibly) 2 programs: * one that shows the serious side of the project, and it's uses for further applications * And one other 'fun' program (e.g a simulation of the Nintendo Mii avatar character) which would hopefully mean more people would be interested in our project on the open day - Discussed the Interim report, and the need to work on this as the deadline is approaching quickly Time: 40 Minutes
Meeting No: 11 **Date:** 25th November 2009	**Meeting No:** 12 **Date:** 2nd December 2009

Date: Thu 25 November 2009 Time: 12 PM (Noon) Location: Computer Science Hub & Lab (A32) Attended: Alex, Aaron, Danni (Daniel cannot make the meeting today - email has been sent before the meeting took place explaining his absence) Meeting Information: - (Explained Daniels absence) - Discussed with Aaron the progress of the control points, he explained what a NURBS surface is, and how the implementation is progressing - Discussed with Aaron the interface for the new program, need to mix Win32 with GLUT, and replacing Win32 keyboard control's with GLUT equivalents - Discussed the progress of the Interim Report with Danni, discussed that some parts will need to be completed by Me (Alex) and Aaron - Discussed progress of the merging of the two programs - Danni explained that an email had been sent regarding the progress of the report, but the group had not received it, we then went to the lab to check what the issue was and resolved it - In the lab: Discussed the parts of the Interim Report that have been completed - In the lab: Danni showed us the sections she had completed, and the GANNT chart, we discussed the need for DFD's and ERD's, ERD's are not needed but DFD's are - In the lab: I proposed that I would complete the user interface section and Aaron proposed that he would complete the program design section, as we have both been working on these areas - In the lab: Possible need for a meeting on Friday (emails will be sent) Time: 40 Minutes	Date: Wed 2 December 2009 Time: 12 PM (Noon) Location: Lab (A32) Attended: Alex, Aaron, Danni, Daniel Meeting Information: - Interim Report needs to be completed. - Finding out which sections of Interim Report are not completed and why - Delegating work to each team member to complete for Interim Report deadline - Discussion of deadlines, code completion date etc - Working on Interim Report Time: 50 Minutes