Diary

|  |  |  |  |
| --- | --- | --- | --- |
| Week commencing | Entry | To do | Done |
| 20/01/2014 | I had my first meeting with my E.P mentor he explained the basic criteria for EP and where relevant resources are. I started my Rational.  I did some more research into my topic and looking for alternative languages for writing a 3D object editor, such as C#, C++, C and Silverlight. The information is in the file called ‘Gathering Information about different programming languages’.  The main problem with 3D graphics and any computer graphics is that they require sufficient memory so when I create my program I have to consider the performance aspect. |  |  |
| 27/01/2014 | I finished a very basic timeline that indicated when I’ll have free time and when I’ll be able to work on my EP.  I had a discussion with my dad who has previous experience in programming in C, C++ and C#. The information is in the file called ‘Choosing Language’.  I decided what I want to do and which programming language I want to learn and I’ve outlined why in my Rational. |  |  |
| 3/02/2014 | I have not done anything this week because I had my art coursework due |  |  |
| 10/02/2014 | Further research on C# and OpenGl called SharpGL  Installed Visual Studio 2013 on my computer that will enable me to work. |  |  |
| 17/02/2014 | Tutorial on C# completed. in file: MathsQuizTutorial  This tutorial introduced me to the basic C# structure.  <http://www.tutorialspoint.com/csharp/csharp_program_structure.htm>  I did this tutorial to get to know C# a little bit better. It seems like an easier language to learn than C++ and just as powerful although I still don’t have that much experience.  This tutorial was easy to complete, the syntax may be new but the style of programming is similar to what I have done before. I know for my project I will need something a bit more sophisticated, but right now I need to get comfortable with language. |  |  |
| 24/03/2014 | There already existing 3D modelling programs, and this week I’ve been looking at some. Blender is a more professional program which is used to create 3D characters and shapes in movies and games. This program seems to be a lot more complex than what I will try and produce. Google SketchUp seems a lot more like what I will try and create. It is more users friendly and it is very easy to manipulate the shapes.  Both of the programs allow you to add texture to the surface of the shape drawn but I don’t know yet if I will have texture as it is one of the more complex aspects in 3D graphics modelling. |  |  |
| 03/03/2014 | (Reflection) I have been doing something almost every week, but the pace at which I am going seems to be a bit slow for my project. The reality is that I am not crystal clear of the program that I want to design. I need to make a design of the final product by the end of this week.  I will introduce a new column into my diary this week: ‘To do’ column. In it I will write what I need to do that week. When I complete what I had to do, I will write ‘Done’ in the next column.  I have created a new word document in which I have a list of tasks that I think need to be completed. They are smaller tasks of my project to make it more manageable and so that it is easier to approach.  The design of the product is in file: DesignManagement.V1 | Finish DesignManagement.V1 | Done |
| 10/03/2014 | A ‘double buffer’ is a function that allows you to display graphics more smoothly avoiding the image from jumping when the next image is loaded onto the screen. The principle is that it sets all ‘background’ pixels to display an image and then loads them onto the screen. This is a useful feature that I will use in my program and will stop from an image from being displayed pixel by pixel.  I’ve installed Tao libraries and OpenGL libraries onto my computer so I can access the graphics card functions. However I still have not figured out how the Tao library works. I will look for more tutorials and websites that tell me more on how to use it.  I have found a tutorial on YouTube that walks me through C# and OpenGL Library. I think this is a good tutorial it is clear and is presented by a blogger who regularly posts blogs and tutorials on various types of programming languages. | Watch this tutorial from  <https://www.youtube.com/watch?v=2KEHrB82Z2M> | Done |
| 17/03/2014 | I have not completed the task to the right as I still cannot get the Tao Framework to work. For some reason the files are not seen by the Visual Studio (VS). I tried re-downloading the files from the site suggested in the YouTube tutorial but I still cannot attach them to my project.  I have an option to contact the guy from the tutorial and ask him for help or search for other tutorials that explain how to use different 3D graphics libraries. | Build a basic application form.  Add canvas and camera to the application. | Not done  Not done |
| 24/03/2014 | I did not do anything this week because I had too much school work this week in preparation for my Art Exam. | - | - |
| 31/03/2014 | I reflected on my project doing a mid-term project review which can be found in ‘MidtermProjectReview.doc’  I have abundant the list of targets (DesignManagement.V1) that need to be done and because I find it hard to flow them. C# is still a bit new to me. It was useful to list some features that I wanted to be done but it is harder to follow them than I thought. I am still not quite sure how I will manage the mouse interaction with the user for example. I have talked to my dad about this who is an experienced programmer and he said it might be little tricky to introduce a mouse interaction. I need to add the canvas first though so that I draw on it. | Midterm Project Review | Done |
| 7/04/2014 | This week I familiarised myself with C# even further completing a set of programming challenges from the Euler Project site. The projects done from this site are a bit irrelevant to my project in terms of context but they are programming challenges that develop my problem solving skills and most crucially it enables me to familiarise myself with the syntax in C#.  I also found out more about classes and object oriented programming which I think will be vital to know for my project. Object oriented programming enables you to create templates and then create objects from it. It will help with redundancy in my code as I will just have to build an object, say a square, just using that template.  I also built a basic windows application form (WAF). I did not however add a canvas because I run out of time.  It seems like I will not be able to get in time for July because over estimate what I can do during the time provided. | Build a basic application form.  Add canvas and camera to the application. | Done  Not done |
| 14/04/2014 | I did not do anything this week because I had revise for other school subjects and it was Easter. | - | - |
| 21/04/2014 | I’ve added a simple canvas. However the canvas is a canvas that only allows you to draw a 2D object. I will have to research a canvas that I can implement into my software.  The work is attached in a word doc called ‘FirstApp.doc’ | Add a canvas to the application | Done |
| 28/04/2014 | Art Exam | - | - |
| 05/05/2014 | Preparation for exams | - | - |
| 12/05/2014 | Preparation for exams | - | - |
| 19/05/2014 | Exams | - | - |
| 26/05/2014 | Exams | - | - |
| 02/06/2014 | Exams | - | - |
| 09/06/2014 | Exams | - | - |
| 16/06/2014 | After a long break from my project, I have added some extra features and the button events.  There is a class that enables me to control Boolean variables. They will be used for storing the state of mouse on canvas. These variables will change between ‘True’ and ‘False’ and when a mouse is does anything on the canvas an event is selected. The program will look up against the variables and select an appropriate method to make the cursor draw or select or perform another appropriate function on a shape on the canvas.  So far the GUI is very simple. (look at file: GUI.V1) I have a button for drawing simple shapes on the canvas such as rectangle and square and line and a canvas that uses Windows Forms Application (WFA). This is a special canvas that enables you to implement camera lighting and viewpoints. I have yet to link the buttons so that they perform. File “3DGraphicsV1” explains what I have done together with the code.  New timeline: | Get E.P. documentation up to date. | nearly done  diary up to date but  some files are still missing |
| 23/06/2014 | I’ve found out that there is a problem while using the 3D graphics in the Windows Forms Application. I asked my dad while trying to figure out why there was an error coming up when I tried to reference the library and it turned out the Viewport3D library was not appropriate to use with WFA. Therefore I need to create a new project in a WPF Application and rewrite the GUI!  I have looked at the some example code that was put in place on their website to show how to use the library in WPF.  The code that they provided drew a rectangle from a triangle mesh and added light and camera. I’ve used this example to create my own shape which you can rotate with keys on the keyboard.  I’ve tried adding buttons however when you create a 3D object they disappear because the 3D graphics is drawn on the Main Form rather than on Canvas so the button which also sit on the Main Form get blocked. I still have not found a way around this problem. For now I use the keyboard to call events.  I need to look for more resources, find better tutorials on the web.  Screenshots of the program are in the file “WPFGraphicsV1” | Try this tutorial: <http://www.c-sharpcorner.com/UploadFile/mahakgupta/example-of-3d-graphics-in-wpf/> | Completed |
| 30/06/2014 | I have found this really good tutorial that takes you through creating a simple triangle that is drawn in the Viewport3D.  It explains what all the methods do in the library, I will now explain in my own words some of the methods that I will use to show what I learnt from the tutorial:  In the 3D graphics a surface is represented by a ‘mesh’. Every shape is made up of a triangle mesh. I already knew this but it is important to bring it up. A triangle is the simplest representation in the in which you can define a surface there is no smaller shape apart from a point that would define a surface. A rectangle for instance can be represented by two triangles and even a large triangle can be still defined by two triangles. The more complex the shape is the more triangles you have to use to define the shape in a simple way but even that may not be a good enough representation; think of a sphere it has no vertices so it would take an infinite number of triangles to define the surface.  The following stuff is new. Your 3D objects in Viewport are only the surface of the shape that you are trying to make. The triangles that define the surface of the object collected in what is called a GeometryModel3D. This is where you store all the information about the texture and colour and position of the triangles. First you begin by adding some points that are going to be the vertices of your object. Then you define which points make up what triangles. Then you need to define the triangle normals of the points. The Viewport does something clever here because depending on what order you define your triangle normals, i.e. clockwise or anticlockwise, it only renders one side which would be the only one visible for the user, so it actually uses less memory of a computer which makes the application more efficient and quicker.  Then you collect all the triangles and make a group model where you can add texture.  There is something called MyVisual3D. A variable of this type enables you to group the GeometryModel3D models if you have more to add to Viewport3D.  I would like to put an analogy onto this. I like to of it a construction site. First you have steel bar fundaments which are like the Points of the triangle. On top of the steel bars you let you cement set to form a part of the floor, which are like the Triangles. You can now call all the steel bars and concrete a triangle mesh and group it so that it is a GeometryModel3D. Then you can add texture, like brick and windows on the outside. Now you group the whole building and call it MyVisual3D which you can add to the already existing world Viewport3D. Note that you can add the building to a group of buildings on a street just like adding another MyVisual3D; my point being that you can have groups of groups of objects which are then added to Viewport3D.  I have completed the tutorial up to the triangle. Next week I will need to try the next bit where it shows you how to make more complex shapes such as cubes. | Complete this tutorial: <http://kindohm.com/technical/WPF3DTutorial.htm> | Completed up to triangle |
| 07/07/2014 | The July Deadline which is in a few days time looks a bit too optimistic. My application so far is not up to a level I want yet. It does not perform any adding of shapes yet from the user side though I have classes which create the objects.  I want to add edit shapes in my final product. That is the whole point of the project, to build a 3D engine. I am still not entirely sure if I want it to be mouse or keyboard control. Doing it mouse control would be pretty difficult because how can you mark a 3D point on a 2D image. I would need to take account of the perspective and how and where the camera is looking. If I find a solution I will implement it.  I have created a new timeline this week so that I have a better knowledge of how much time I have left. I will try to get as much done as possible before I go on holiday. I have noticed I do much more now that I have finished my exams and things are starting to get shape which is promising.    I have completed the drawing of 3D cubes part of the tutorial. The tutorial is very useful because it gives you tips. For example, there is a piece of code that I will use in my project which generates the normals for a given triangle. This is helpful because I can just call the method to create them. | Complete this tutorial: <http://kindohm.com/technical/WPF3DTutorial.htm> | Completed up to cube |
| 14/07/2014 | I have completed the tutorial up to the camera bit where you specify the camera’s position and direction in text boxes.  The tutorial goes on to show how to add line on each vertex. I don’t think I will find this beneficial in my project. Maybe if I have more time at the end I will add this feature. For example to highlight the shape that has been selected by the user, but for now I will not add this.  This tutorial was very useful because it gave me a good solid background and helped me to program some basic methods that will be constantly repeated when creating a new objects and defining its position.  I need to add more functionality that the program can do, like editing shapes. First of all I will need to think of how the user will select shapes. Mouse clicking on it will be hard because the computer will not know whether the user whetted to select the project in front or behind. The problem with selecting a 3D object in a 2D image is that you are missing a depth dimension. It would be like drawing a line from the tip of the cursor which corresponds to a line in the 3D space of a computer. I would need to have an algorithm that would check what lies on that line and then selecting the object that is the closes to the ‘user’/screen.  An alternative would be to have a list of objects that there are, but then how would the user know which object they have selected. They would have to remember which object is which and that would not be a user friendly environment.  I would need to learn how to implement these solutions. I have too little time to start on anything big now I only have 2 days till I go. I will think of it while I am away. | Complete this tutorial: <http://kindohm.com/technical/WPF3DTutorial.htm> | Completed up to camera |
| 21/07/2014 | Holiday: abroad and no computer so no work on EP |  |  |
| 28/07/2014 | Holiday: abroad and no computer so no work on EP |  |  |
| 04/08/2014 | Holiday: abroad and no computer so no work on EP |  |  |
| 11/08/2014 | Holiday: abroad and no computer so no work on EP |  |  |
| 18/08/2014 | I have written my own function for zooming in and out. I have also changed the textboxes that specify the camera position and direction to a keyboard control which automatically updates the Viewport. This is a much more responsive solution for the operation of the camera.  The ‘zoom’ function was not too hard to program but it made me think. First I had to find out how to the input from the mouse wheel is received which was not too difficult to find on web and then I had to calculate the new camera position. I used Pythagoras Theorem for 3D triangles and the algebra method of triangles to calculate the new positions of the camera.  I have constructed a new timeline (Timeline V.3) that shows what needs to be done before the hand in date. | Complete this when you come back: add zoom and the selection and editing of shapes. | Only zoom was complete |
| 25/08/2014 | When a new shape is added a list created with already created objects. You can add and delete object change their positions and change the size. |  |  |
| 01/09/2014 | I pulled everything together and updated documentation before the start of term.  According to my timeline I should have a majority of my EP completed by now. I have a working solution but it is not as nicely programmed as I hoped. I still need to add some bits and pieces to it. | Update documentation |  |
| 08/09/2014 | --- | Work a bit more on it. |  |
| 15/09/2014 |  | Ask someone to review it |  |
| 22/09/2014 | Peer review + small twiks | Write an evaluation |  |
| 29/09/2014 | Evaluation asking |  |  |
| DEADLINE: | 6th October |  |  |