22/10/2013

Began thinking of ideas for project, came up with UAV flight planner(which I initially discarded as I thought it would be too difficult), an app which helps identify electronics parts by using the camera on a Smartphone (unfortunately I discovered this had been done many times before) and a ecological simulation game where you had to keep a growing village off the grid by investing in green power.

5/11 /2013

Continued with brainstorming ideas for project, came up with several more games and practical applications, most of which were not interesting or not feasible, then revisited feasibility of flight planner program.

6/11/2013

Researched existing solutions to proposed projects found many had already been extensively covered. Chose project that I would use and that I felt I would enjoy the most, the flight planning and mapping program for controlling small unmanned aerial vehicles ( herein referred to as UAV's)

12/11/2013

Finalised idea, began working on documentation and planning and began planning multiple levels of feasible goals

14/11/2013

Wrote identifying the problem section and began writing the needs objectives boundaries section

15/11/2013

Continued writing needs objectives and boundaries. Found that the objectives section was difficult to cover as there were so many different features I could implement in the project however I believe this flexibility will eventually be an advantage.

20/11/2013

Continued work on the documentation without any problems. Created agenda for first progress meeting

9/12 /2013

Created Short term Gantt chart scheduling. wrote about possible alternatives for development of solution and wether existing solutions could be for the purpose of the software i am developing. outlined the social and ethical issues associated with the program, including inclusivity and copyright restrictions. Wrote data types section, recording necessary data types and what they would be used for. Wrote the ergonomics issues section in the User's Perspective. Worked on the choosing the programming language section and settled on C# as I already had experience using it.

10/12/2013

Wrote section on choosing software design approach, settled on the structured approach, but with elements of the prototyping approach to easily implement the multiple stages of final objectives. Wrote documentation about necessary variables.

11/12/2013

Wrote the relevant data formats for display and the appropriate messages and icons section of the documentation. Wrote the developer experience section.

12/12/2013

Created screen design prototype, found this particularly difficult as i do not have much experience in Photoshop, thankfully however, fellow students were able to help me in learning the basics necessary to complete the screen designs. Created the context diagram. created data flow diagram and IPO charts. created structure chart. Wrote flowchart and pseudo code algorithms

13/12/2013

Created more screen designs and began using the screen designs to make the storyboard which, apart from having to use MS paint as I do not have Photoshop, was fairly easy. Fixed structure chart which had data passes missing and was not passing any of the rendered data back to the user/system. Created system flowchart which was more difficult than I expected but I found that some of my other diagrams and the textbook both helped a lot with understanding what the system flowchart was for. Created Pseudo code algorithm for testing if a point is in a polygon. Wrote data dictionary. Annotated story board elements and the interactions between the menus. Submitted Part 1 of the Project.

16/02/2014

Created visual studio project. Realised a windows form application would not be suitable for a full screen application that would need to work on multiple sizes of screens so began researching alternate design structures and started testing with a WPF application

28/02/2014

Watched some tutorials on the features of the wpf designer, created another prototype application to attempt to get the GMap.Net control onto the form, had no such luck. (dll's are not fun)

5/03/2014

created more prototype WPF applications, now mostly understand the XAML design language found the object browser in visual studio which let me begin searching the contents of the dll for the control. I feel as though I'm close but haven't found it yet.

11/03/2014

Found the control within the dll but still at a loss as to how to get it onto the main window of the application. Looked on forums and tutorials but it seemed like implementing the control was assumed knowledge (the documentation on this API is not fabulous).

7/04/2014

now working happily with WPF and have a basic interface design fleshed out. Still no luck getting the control onto the form however.

9/04/2014

Finally found how to get the control out of the GMap.NET dll files (there is an add custom control to toolbox function in visual studio) and got it on to the form of a prototype application. Read through many of the functions available in the GMap.NET class. created a working map interface that could be zoomed in and out with the scroll wheel but that could not pan or be drawn onto.

10/04/2014

continued work on prototype program. Successfully integrated some aspects of the GMap.NET API (changing map provider, moving to lat long locations) but could not access the classes of the library necessary for creating the overlays on which marker points are drawn. Searched through the dll files and example programs but could not resolve issue. Notices that the version of the API made for WPF applications was much smaller than that of the windows forms application. Hence discovered that the WPF version was severely limited and omitted many of the key functions from the Windows forms version such as marker placing and movement, line and route drawing, and polygon drawing and editing. Decided to fall back to a normal windows forms application to regain the main functionality of the program at the cost of easily implementable resizing.

12/04/2014

Figured out creating overlays on the map and drawing points onto these overlays. having some trouble with the drawing though as it seems to put it at the centre of the visible map every time instead of at the coordinates i specified in the calling function.

15/04/2014

sorted out problems with markers in the wrong position (I was creating the overlay again every time I called the marker creating method and so it was clearing all the other values o markers on the map). Can now change map provider (Google maps, bing maps, OpenStreet maps etc), search for location by name and by coordinates and move the view there, draw points and polygons(still cannot place points with mouse click or edit them once there placed). I am having some trouble finding a way to get ground elevation values from points (may have to use Google maps web services) but this is not crucial until later in the project.

22/04/2014

Figured out how to make controls scalable in a windows forms application (with a table layout panel)and created final prototype/final project (the one that will be version controlled through git hub). Sorted out some minor zoom problems and created the main side bar interface based on the plans from initial documentation.

24/04/2014

Had to find a way to change what mouse clicks on the map did depending on what the user was trying to do. settled on a state machine with an enumerated drawing mode variable and a switch statement inside the mouse click event, this would tell the program to either do nothing, create a new point in a path, or a new point in a polygon.

Had some trouble translating the position of the mouse click on the screen to the actual lat long coordinates that that point represented, after some research I found that there was already a function in the API specifically designed to do just this ( FromLocalToLatLng ) which already outputted the point in the correct format for me to use.

Putting in the path and polygon drawing functions I initially had some trouble rendering the path/polygon after each new point was added, unfortunately, updating the list of points the path/polygon was created with did not update the path/polygon itself. This meant that I had to clear the path/polygon and re render the new one after each consecutive point was added.

In order to distinguish which end of the path was the start and which was the end I needed to add different coloured markers at each end. The end marker was simple as i just add a marker to the last point in the list when the end path button click event was called. For the beginning marker however the first point does not exist when the begin path button click is called and I did not want to use a dedicated flag for such a small thing. I solved this by placing an if statement in the create new point method which drew a marker when the number of points in the list of points was 1.

now have fully function path and polygon drawing with appropriate interface for beginning ending and clearing said paths/polygons.

25/04/2014

Tried to get the contents of the list pointsPath (contains the coordinates of each point in a flight path) to display in a data grid view control on the form. Initially I tried setting the list as the data source for the data grid view, but this resulted in the table not updating. Then I tried changing the List to a binding list, but I then realised that displaying it this way meant I had no control over what each column in the table (dgv) was displaying and this meant that there were unwanted empty columns in the table. Eventually I just created an update table method that was called whenever a point was added which set what parameters from the Points in the list went into what columns.

28/04/2014

Implemented exporting Flight paths to text files, had some difficulty getting the text file in an appropriate format that the telemetry radio program(communicates with the helicopter) could read. Eventually found a website that had the format of the file and managed to implement writing to this kind of text file from a normal "browse" dialogue box. Still need to implement exporting to CSV for use in excel and possibly some other formats. Also Created classes for Geometry and Flight Path manipulation, began working on necessary algorithms e.g. point in polygon, intersecting lines, are of polygon, convex hull etc.

30/04/2014

created convex hull, calculate area, and calculate angle between three points algorithms. convex hull does not seem to be working at the moment however, as all I can get it to output is null or the same list of points it took in. I have tried debugging with break points throughout the function but I cannot yet isolate the issue

the calculate area function is, in reality not actually used to output the area of the polygon, it is used to determine the direction of rotation of the points in the polygon. This is to ensure the convex hull function is using the internal angles of the polygon and not the external. If the user creates a polygon the calculate area function will be used to make sure it is in the correct direction (Anti clockwise).

1/05/2014

Have discovered a major issue with the convex hull algorithm. It seems to take out points that should not be taken out in some cases, but work perfectly in others. I managed to narrow the problem down to the function for calculating the angle between the three points. the problem in the Angle function was that I was using Atan2(aka inverse tan) and this will only return angles in the range Pi to -Pi (which is a bit confusing as inverse tan should be between Pi/2 and -Pi/2) and when a turn in the polygon crosses this boundary angle it throws out the calculation completely.

tried normalising the range of angles to 0 to 2Pi but this just shifted the boundary problem around.

2/05/2014

use the area algorithm to determine if 3 points progress cw or ccw

convex hull was creating crossing lines which cause problem

intersect polygon

fixed bug cause by rounding problem

point in line segment

change convex hull algorithm so it never creates crossing lines

tolerance factor

18/05/2014

Implemented the reverse list function which reverses the list of points if they progress clockwise to ensure that the convex hull algorithm works as intended. In order to determine if the points need to be reversed I use the area of a polygon function, which returns positive if the points are anticlockwise and negative if they are clockwise (this is the simplest way to determine the general direction of the whole polygon). To reverse the list i simply swap the 1st and last values , then the 2nd and 2nd last values, and so on until the middle of the list is reached.

19/05/2014

Class functions that were written to pass in and pass out a list, seem to be acting on the list as if it were a reference object. I believe this to be due to the referential nature of the list object. I have now decided to use this to my advantage as it means i do not have to pass in and return my lists all the time.

began work on the survey path generation, currently i just have a stub that returns the polygon (no maths needed yet) so that i can test the rendering of the newly created flight path. I had some problems with the rendering of the flight path as all my rendering functions had been created to work with one point being added at a time, rather than an entire list being rendered at once. by splitting some of my rendering function up into separate more discrete functions (which can then be used inside loops to render the paths easily) i have managed to make the rendering of the basic path work. However the generation of markers denoting each point in the path and the end points does not work as I have created it to depend on the length of the list of points as its counter, and thus again, only works adding points one at a time. I believe a similar fix to the basic rendering will be achievable with this issue.

20/05/2014

solved problem with markers not rendering, created a new marker function that looped through all the points in the list and add the markers at each one, rather than adding them one by one as the points are added to the list.

created bounding box function to find the limits of the drawn polygon and used this to begin writing the algorithm for generating a survey path over the polygon ( with path and point spacing and path angle inputs). just using stubs for the moment to display on the map things like the bounding box and convex hull to see if they are working properly (these will not be seen in the final application).

21/05/2014

created trig algorithms to generate a zig zagging rectangular path over the polygon, can adjust path spacing and angle. I now need to populate these lines with point according to Point spacing, and then cut out any that aren't in the polygon.

I had some minor problems with zoning of trig functions and negatives and positive offsets but i managed to fix these by realising i needed to offset one of the angles by Pi/2 to get the correct offsets. I still have a fairly major problem in that all of my algorithms are being calculated in degrees of latitude and longitude, which appear fine, but do not work if the user tries to input any values in metres. To fix this i will need to write another algorithm to convert the metres input into degrees, which will change depending on the latitude of the line being measured.

24/05/2014

Approximated scale factor from degrees to metres, just so the user interface worked close to measuring in metres (approximately 100km to a degree)

The generate survey function was having index out of range errors when it was being called without a valid survey polygon (i.e. empty/not initialised) but in a simple if test to prevent this.

created function to clip the generated zig zag path to the boundaries of the polygon. initially no intersections were being registered with the polygon at all, but I realised that it was only when the lines were vertical that this happened. It turns out the calculations to test of the intersection points were on the initial lines had some rounding errors in the 6th decimal place and lower, a tolerance factor which gave the test some leeway corrected this. I then had to handle the case of more than 2 intersection points (concave polygons). I stored all of the intersection points in a list and then sorted this list be Y value then picked the lowest and highest to be the end points. This will work for all cases but horizontal lines, so will have to change this function to sort on X in this case.

This algorithm now works very effectively with the full survey polygon (concave or not) and so I no longer need to use the convex hull algorithm.