

Graphic User Interface Report

This document intends to give an explanation of the process we used for designing the Graphic User Interface, and justification for our visual choices and interaction mechanisms. A larger version of each screenshot is provided in appendix B at the end of this report.

Objectives

Before beginning the planning and then implementation of the GUI, we believed it important to set out our objectives for it. The objectives of the GUI were already partly defined by the functional and non-functional requirements (available in Appendix C) that we had created for the game. However, we developed and further refined these into basic objectives through a combination of team discussions and client consultation. We will regularly check that the GUI designs and implementation meet these objectives, as well as the functional and non-functional requirements.

The GUI must:

- Be easily expandable
 - Space should be provided in the GUI for new features, whilst not looking incomplete.
- Be intuitive for the user
 - Although a user manual will be available, the user should be able to intuitively play the game and use all controls.
- Not be so complex that it intimidates the user
 - Although realism is important, this should not make the experience intimidating for the user.
- Provide all information and controls required by the user to achieve the goal of safely navigating planes to their exit points.
- Be designed to be easily adaptable to touch input
 - Where keyboard or special mouse controls (e.g. scroll wheel) are used an on screen button alternative should be provided.

Prototyping

Before we started to program the game or implement the GUI, we discussed in length the requirements of our user interface and the mechanisms we intended to use for interacting with the airspace. Throughout this process, we used a series of computerised sketches to illustrate individual ideas to the rest of the team. An example of one of these sketches is provided below

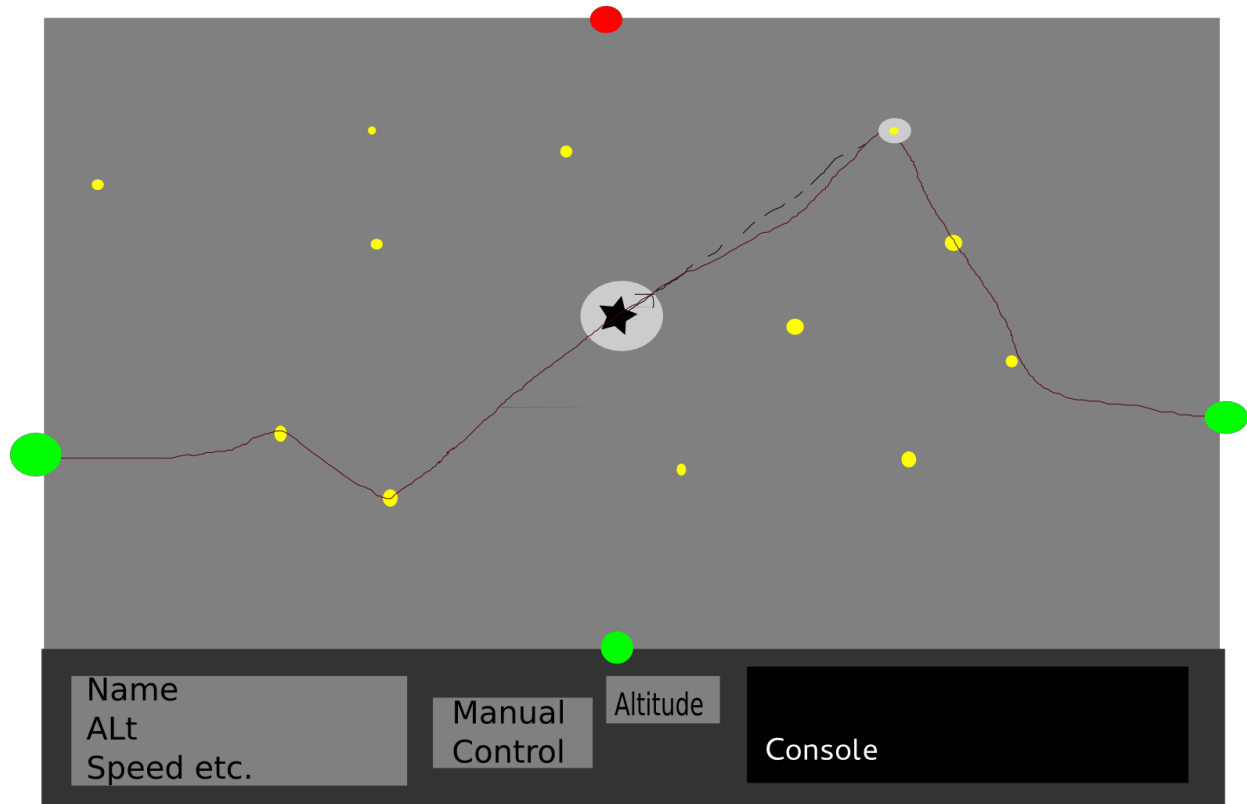
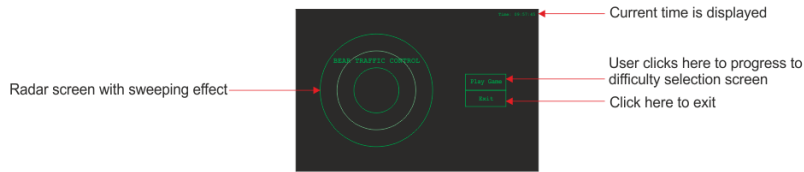


Figure 1: Rough sketches from first GUI meeting. These were used to assist with the development of ideas and to illustrate concepts to all team members.

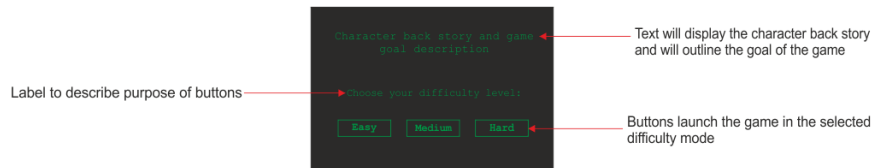
After our initial meeting, a number of initial GUI concept prototypes were produced. Each of the GUI prototypes were checked against the relevant functional and non-functional requirements to ensure that they were achievable using the given design (examples of this are provided throughout this report in relation to our final designs). The GUI designs were also examined by our programming/implementation teams to make sure that they believed they were achievable in the given time frame and that the architecture design would support them. Once we had created a basic prototype of the game, we presented it to our client to gather feedback on the gameplay, and the GUI design. Throughout these reports, client comments that influenced particular features of the final designs are included. Our final mockup of the GUI design is shown on the next page (a full page version is included in Appendix A).

GUI Designs

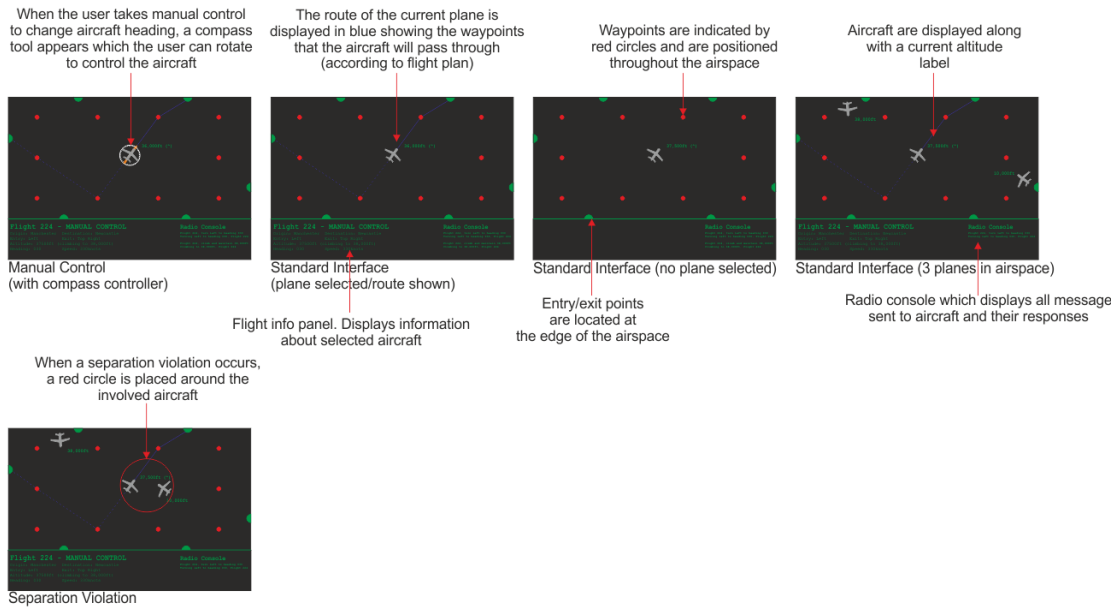
Main Menu



Difficulty Selection Screen



Game Play



Game over screen

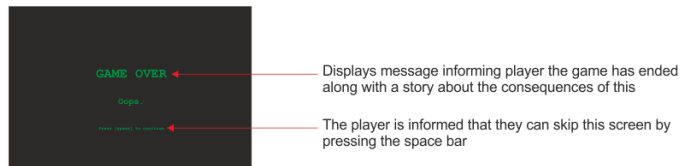


Figure 2: Final mockup of GUI design. This was used as reference material by the implementation team with improvements/alterations made where necessary.

General GUI Features

Throughout the game, all of the text used is in the same font. This use of a consistent font should provide a comfortable and familiar experience to users. The font chosen (and the way in which it appears on the screen) is designed to emulate that of an old computer terminal. While this may not be a realistic choice, NFR (Non-functional Requirement) 4 specified that a fun and engaging experience should not be sacrificed due to realism.

We used a green and black colour scheme similar to that of an older air traffic controller's interface which users may be familiar with from film and television. We felt that this would help us with the overall feel of realism (NFR 1) whilst also ensuring that the game was engaging to the player (NFR 2 & NFR 4). The game features background music which extends some of our darker themes which helps the user connect more emotionally with the struggling air traffic controller he is controlling aircraft through the airspace.

NFR 3.1 stated that "the game will be appropriate for the market in terms of the graphics and language used". We feel that the use of this colour scheme and the general "theme" of the game is appropriate for the market of students.

NFR 5 states that the GUI should be updated regularly and sets a target of 30ms for this update time. (FR 9 and FR 9.1 specify the hardware that this target should be achievable on). Throughout the design (and implementation) of this GUI, the programming teams were consulted to make sure they felt that the current designs would allow us to achieve our 30ms target. To aid this, the implementation of the GUI includes a fps (frames per second) display in the title bar which allows us to identify if our targets are being met. We implemented a cap of 60 frames per second as it is unlikely any higher will make a noticeable improvement. The current version of the game was tested on the Computer Science Lab PCs (the specified hardware) and the game was observed to run consistently at the maximum possible rate of 60fps, exceeding our 30ms target.

Main Menu

The main menu for our game features the buttons for starting the game demo and exiting the game and is the first screen that the user is met with upon launching the game. We believed it was important to provide a clear and understandable menu as NFR 6 made it clear that the game should be easy to play.

To help with further development of the game, we left plenty of space which could be used to display further information such as high scores and statistics.

The main menu also displays the current date and time in the upper right corner. Although this information is not necessary to the user, we believe that it adds to the overall realism of the

game.

NFR 1 described incorporating realism where possible and appropriate, and to this end we added a radar-like graphic to the menu. This radar also produces a realistic “beep” sound on each oscillation as during the demonstration of one of our prototypes to our client, he advised us to think about adding sound to the game. (Note: This was a non required feature, but we were informed that it could be added if time allowed.) The current date and time were added to the corner of the main menu to further increase realism.

We also felt it was necessary to have the name of the game present in the centre as it is important to be identifiable to the user. The layout of the menu was chosen to make the interface again feel technical and real to the player.

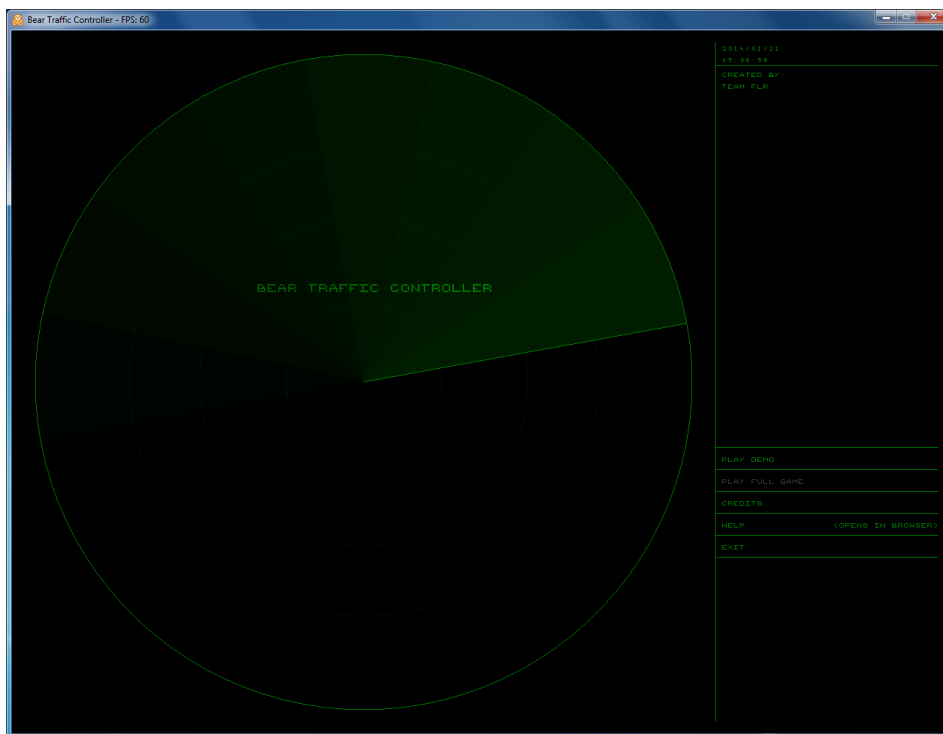


Figure 3: Screenshot of the main menu

Difficulty Selection Screen

FR 11 states that the user must be able to alter the separation rules of the game. We chose to use three discrete difficulty settings to implement this, and it is from this screen that the user can select these. Named difficulty values (Easy, Medium and Hard) were used as users should be familiar with these from other games they have played (given that the target audience is specified in NFR 3 as students it is likely that users will have played other games before).

Each difficulty is chosen using labelled buttons positioned under an introductory message allowing for easy selection of the desired difficulty level. This easy selection means that we are again satisfying NFR 6 as the user should understand how to progress from this screen. The positioning of the buttons should mean that the users read the message before clicking a button and selecting their difficulty.

The buttons used are also designed to be large enough that they could be interacted with using a touch screen, and the ability to adapt for touch input was stated in NFR 7.

The introductory message provides an entertaining backstory and sets the scene for the game, this was implemented because the most important requirement to our client was enjoyment (as specified in NFR 1 and NFR 4). Although the backstory may not be realistic, NFR 4 clearly states we should not be restricted by reality.

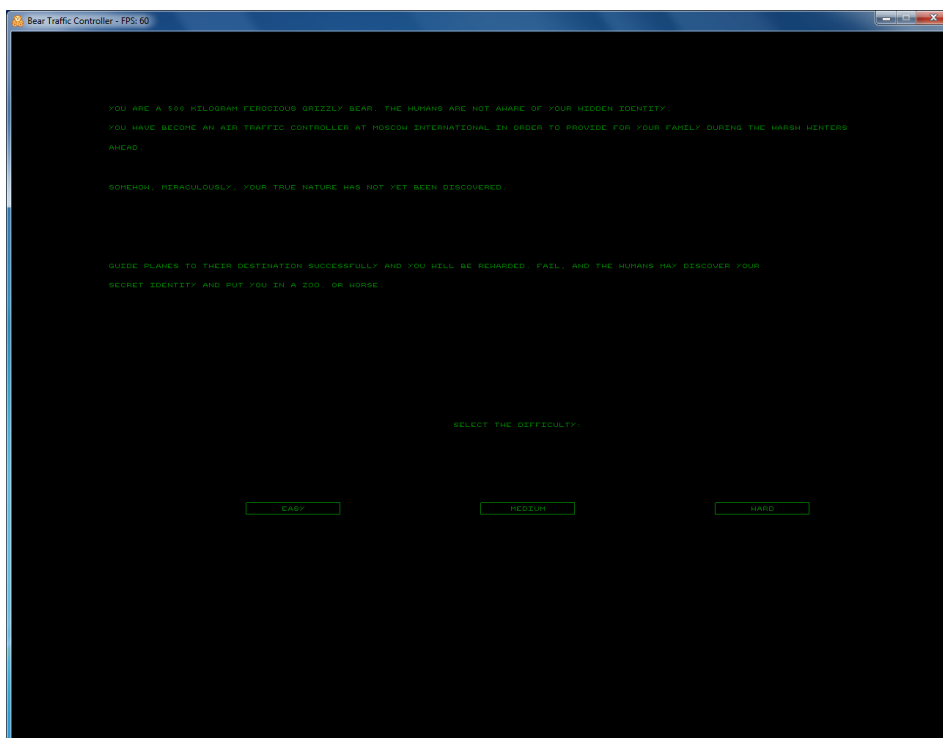


Figure 4: Screenshot from difficulty selection screen

Main Game Screen

After selecting a difficulty the user is placed directly into the game screen, we use the same colour scheme and font as before to ensure consistency and make the game easy to play (NFR 6). FR 8 states that we must track the user's score, and NFR 2 is that the game must be engaging. Early versions of the game displayed the score in the top left of the screen. However, the client asked us to remove this functionality until a later date. As such, it is no longer

implemented at the client's request. Our requirements (FR2 & FR 7) state that each aircraft must have different attributes such as velocity, position, destination and origin, these statistics (for each plane) are displayed in the bottom left corner.

The statistics box is filled when a plane is selected and the box is titled with the flight name to show which flight it relates to. The technical information displayed is similar to that of a real air traffic control terminal so this adds to the realism of the game (NFR 1). The units displayed here are not necessarily those that the game architecture uses, but they are converted to realistic units as NFR 9 rules that units which fit the air traffic control domain should be used (NFR 9.1, NFR 9.2 & NFR 9.3 specify these units explicitly)

In the bottom right corner we have a console which displays messages to the player about incoming planes. This element was implemented as it's difficult for the player to predict where and when a plane is going to arrive from an entry point (of which there are 4 according to FR 4), we implemented this in a format which is both realistic and useful (NFR 1.)

In the centre of the screen is the game panel itself, the corners of this inner panel are labelled with entertaining text naming the nearby exit/entry waypoint, hopefully improving engagement (NFR 2) and enjoyment (NFR 1). Waypoints are situated around the virtual airspace in fixed positions, we have the four corner exit/entry points and the rest are scattered around. Solid red circles with larger red rings around are used to represent these waypoints, the red contrasts enough to be easily identifiable but is less noticeable than more important information such as planes and flight paths. This should make the game easy to understand, and therefore easier to play, meeting NFR 6.

Basic plane sprites are used to represent the aircraft in the airspace, these were chosen as the player can relate to sprites easier than a more realistic dot/line representation. Although this does sacrifice realism somewhat, we believed that this is a prime example of providing a more fun and engaging experience (NFR 4) which also makes the game easier to play (NFR 6). We used white plane sprites to give the best contrast with the dark background, hopefully making the game easier to play and improving playability for those with vision problems (e.g. colour blindness). The background that we chose is also designed to improve player enjoyment, even though it once again sacrifices realism. This background helps to put the game into context much better than a plain black background.

FR 6 states that "the game shall end when the distance between one aircraft and another is lower than the specified separation rules". After trying one of our prototypes that met this requirement but did not display separation violations to the user we decided to make it clear to the user when separation violations are occurring as this would aid playability and although making the game a little easier, would make it more engaging (NFR 2). Now, when flying close to another aircraft, a plane will display a red circle around itself and this circle represents the separation radius, the distance two planes must be apart from each other.

The user has the ability to toggle which altitude is displayed by scrolling with the mousewheel, when a plane is on a different altitude to the selected height, the planes will be shaded grey to show the height difference. There are only two altitudes planes can fly at in our game, so altitude labels were not required, however, adding these labels adds to the realism.

Clicking on a plane with the mouse will select the plane, this will highlight the selected aircraft's flight path. The flight path is displayed as a light blue path between the waypoints showing the planned motion of the pane. When a plane is selected a button appears near the top of the screen, this button allows the player to take manual control of the selected plane, this allows the user to control the plane using their arrow keys. The use of the arrow keys is something which was discussed at length, but as FR 10 specifies the use of the keyboard and mouse for control, we felt this was acceptable. The spacebar can also be used to enter manual control mode, a compass is then displayed around the plane, which gives the user control of the direction of the plane with using the mouse, this was done to meet requirement NF7. The user can order a selected aircraft to ascend or descend to a different altitude level by clicking the corresponding arrows on the aircraft diagram on the bottom of the user's display.



Figure 5: Main Game Screen

Figure 6 (See appendix): Screenshot of the game with a plain black background for comparison with Figure 5. This design was more realistic, but didn't provide as enjoyable experience for the player.

Game Over

When two planes crash, the game ends and the user is presented with a message and the opportunity to continue. The text is typed out, letter by letter to imitate an old computer terminal for consistency with our realistic theme (as mentioned in the General GUI features section). This consistency should ensure NFR 1 and NFR 6 is met. The purpose of the message, like before, is to add entertainment value to the game (NFR 1 & NFR 4).

On showing the prototype game over screen to our client, we were told to make text appear faster, or provide a way for the user to skip reading the text. To accommodate this, the user can press any key at any time to skip the game over screen and return to the main menu.

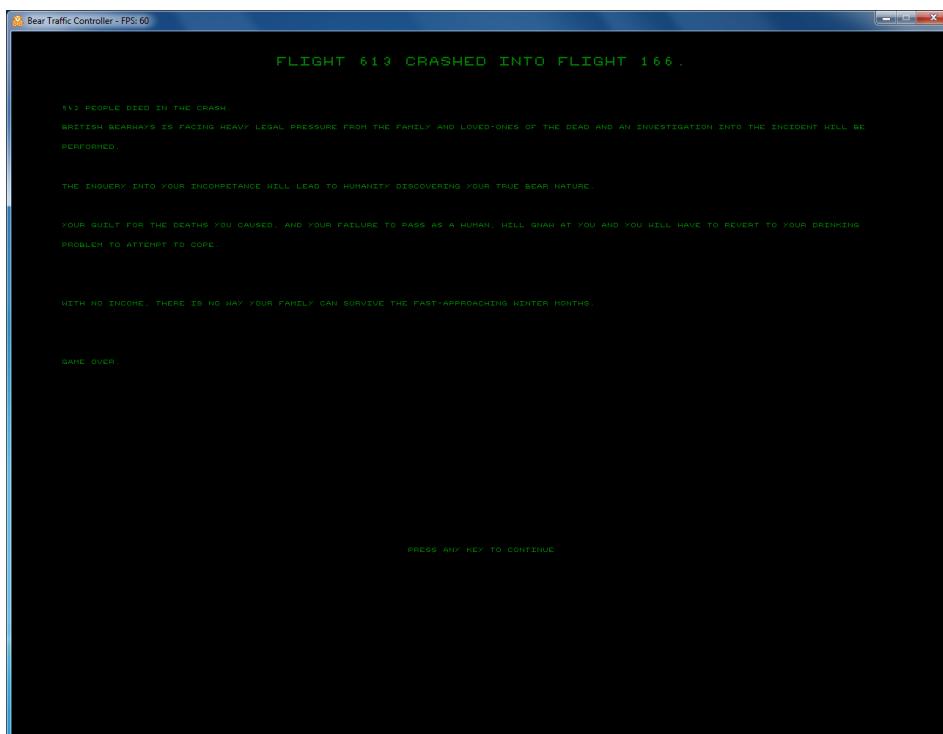


Figure 7: Game over screen

Credits Screen

Selecting the “Credits” option on the main menu will open up the credits window. This window displays the names of the developers, and gives credit to external libraries and produces of sound effects and music that we used. A simple loop of music is played to lighten the mood and contrast with the more gloomy feel to the rest of the game. The credits can be accelerated by holding down the mouse button and the credits screen can be exited by pressing the escape button.



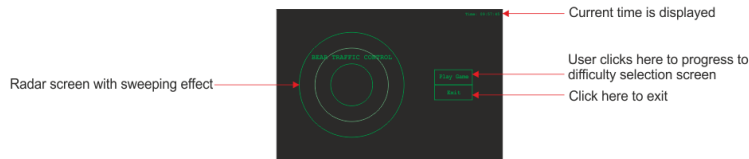
Figure 8: Credits screen

Appendix

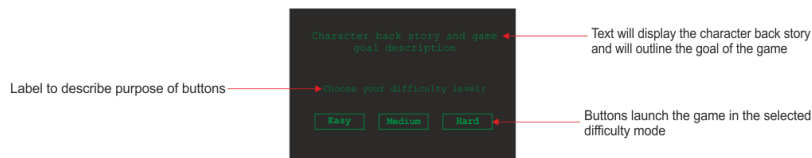
Appendix A - Final GUI Designs

GUI Designs

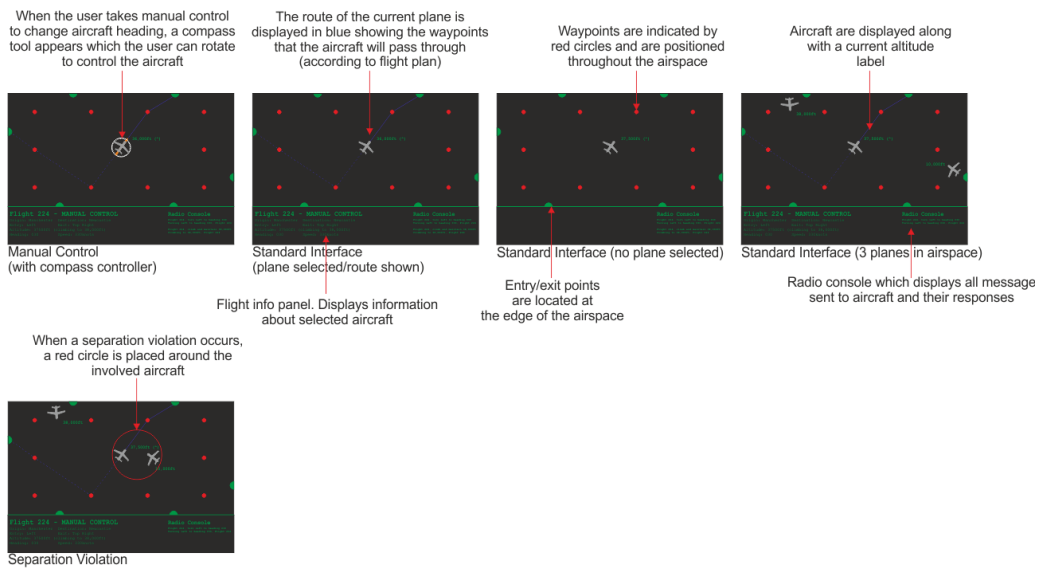
Main Menu



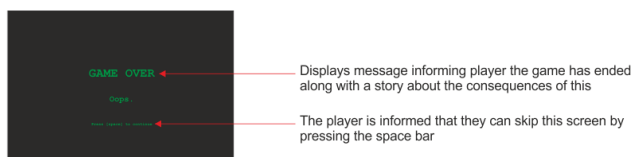
Difficulty Selection Screen



Game Play



Game over screen



Appendix B - Screenshots

The next few pages contain full page screenshots from Bear Traffic Controller. Each of these screenshots has been taken with the game running at it's native resolution on the Computer Science Lab PCs. The screenshot also displays the games title bar, showing you how it would appear to the user, and also allowing the fps counter to be seen.

Figure 3: Main Menu

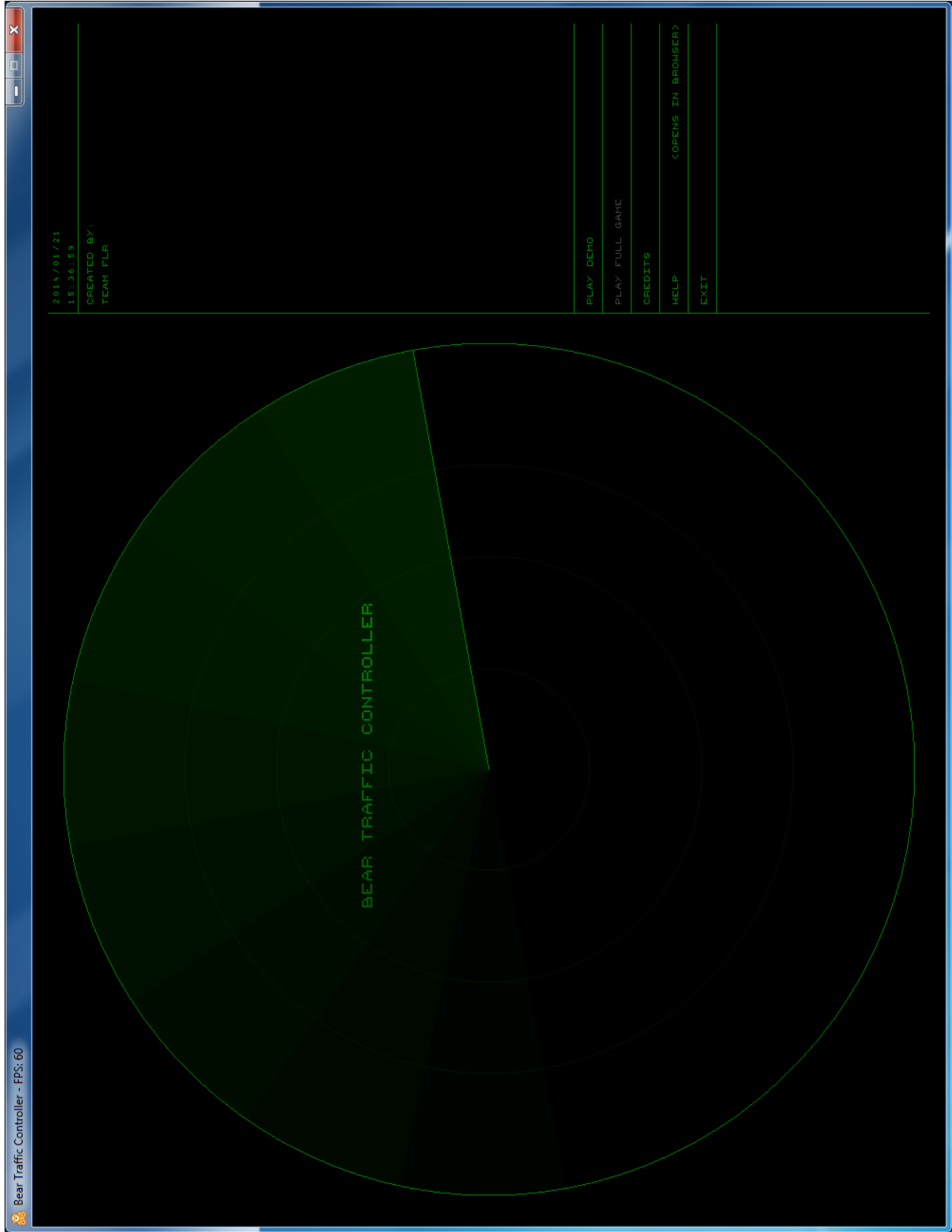


Figure 4: Difficulty Selection Screen

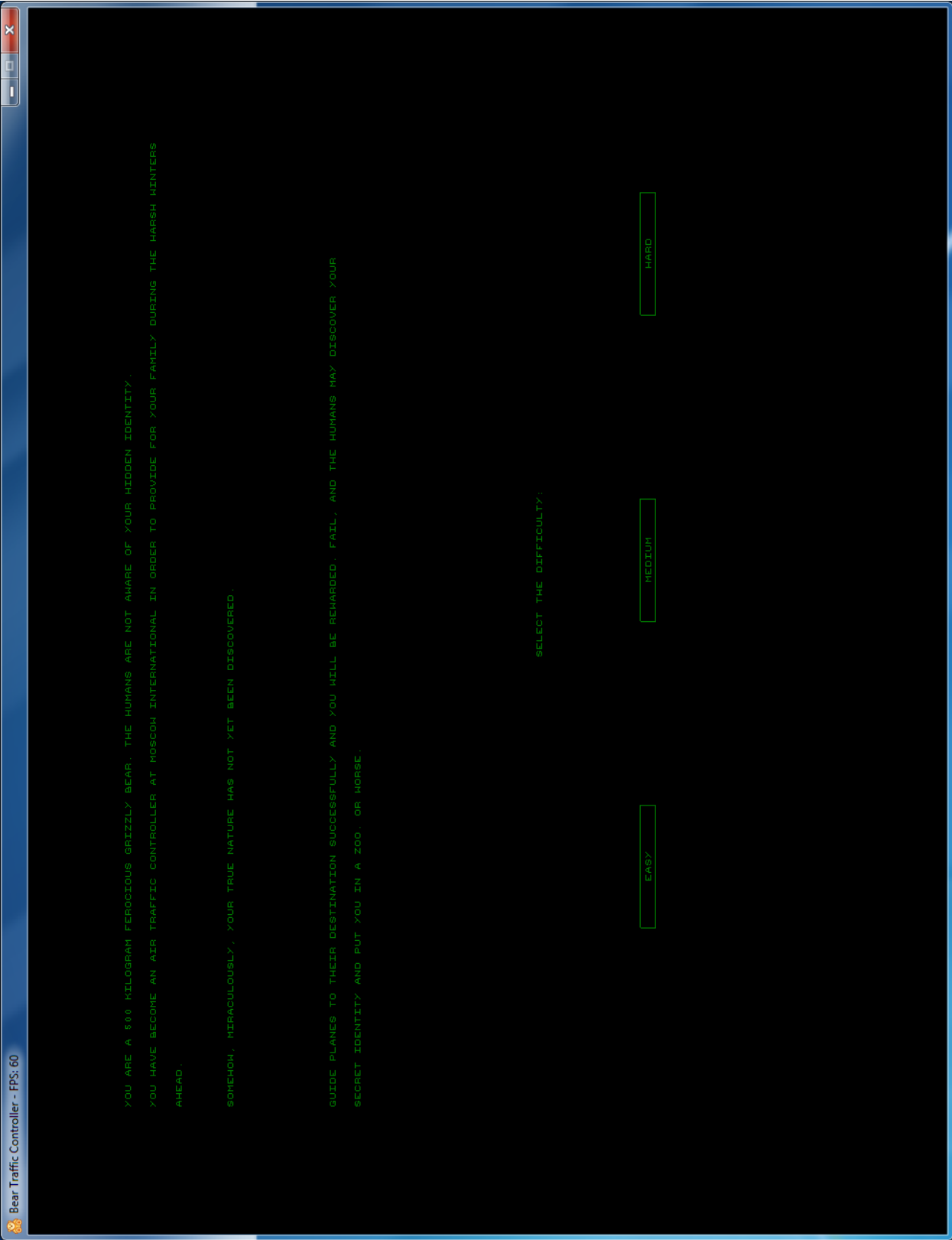


Figure 5: Main Game Screen

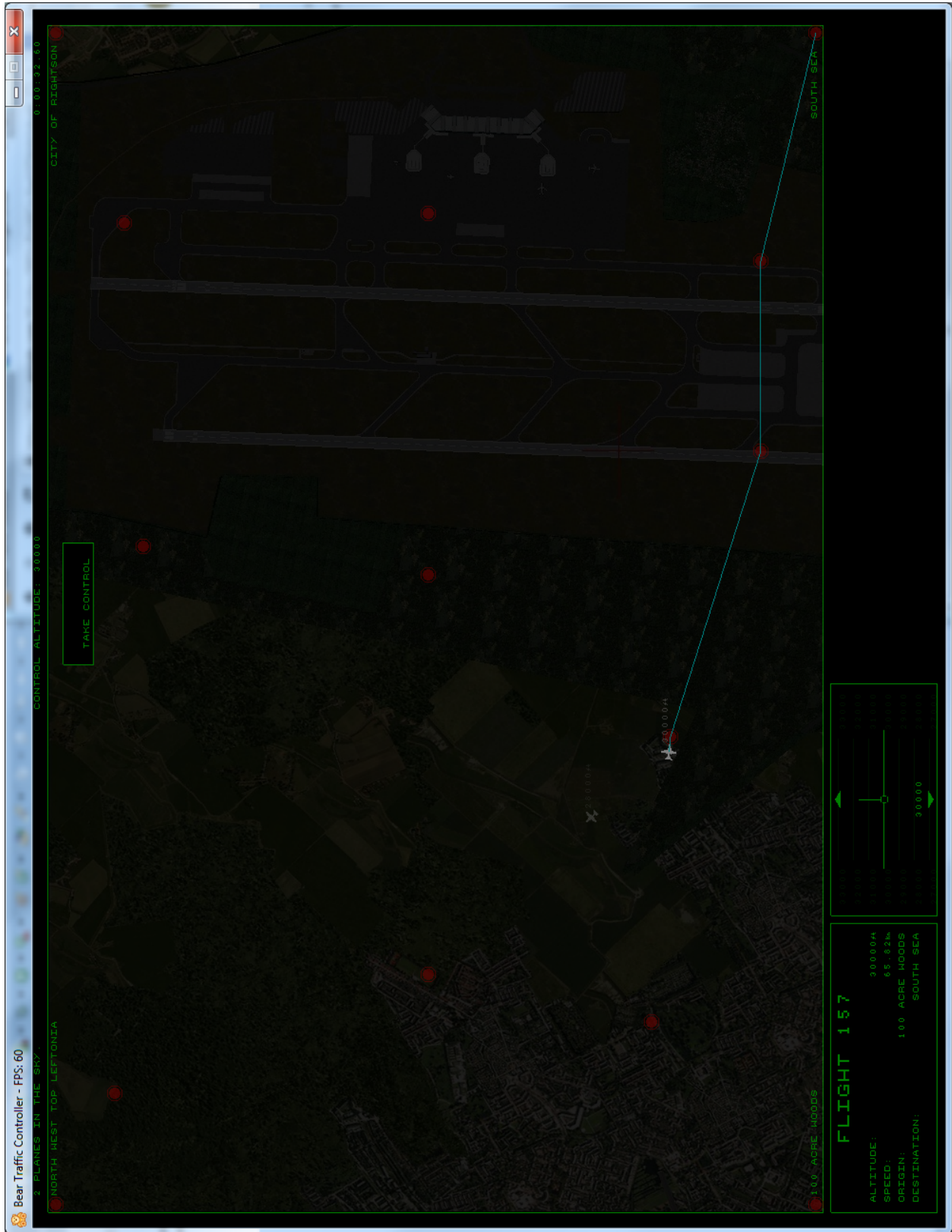


Figure 6: Main Game Screen (black background)

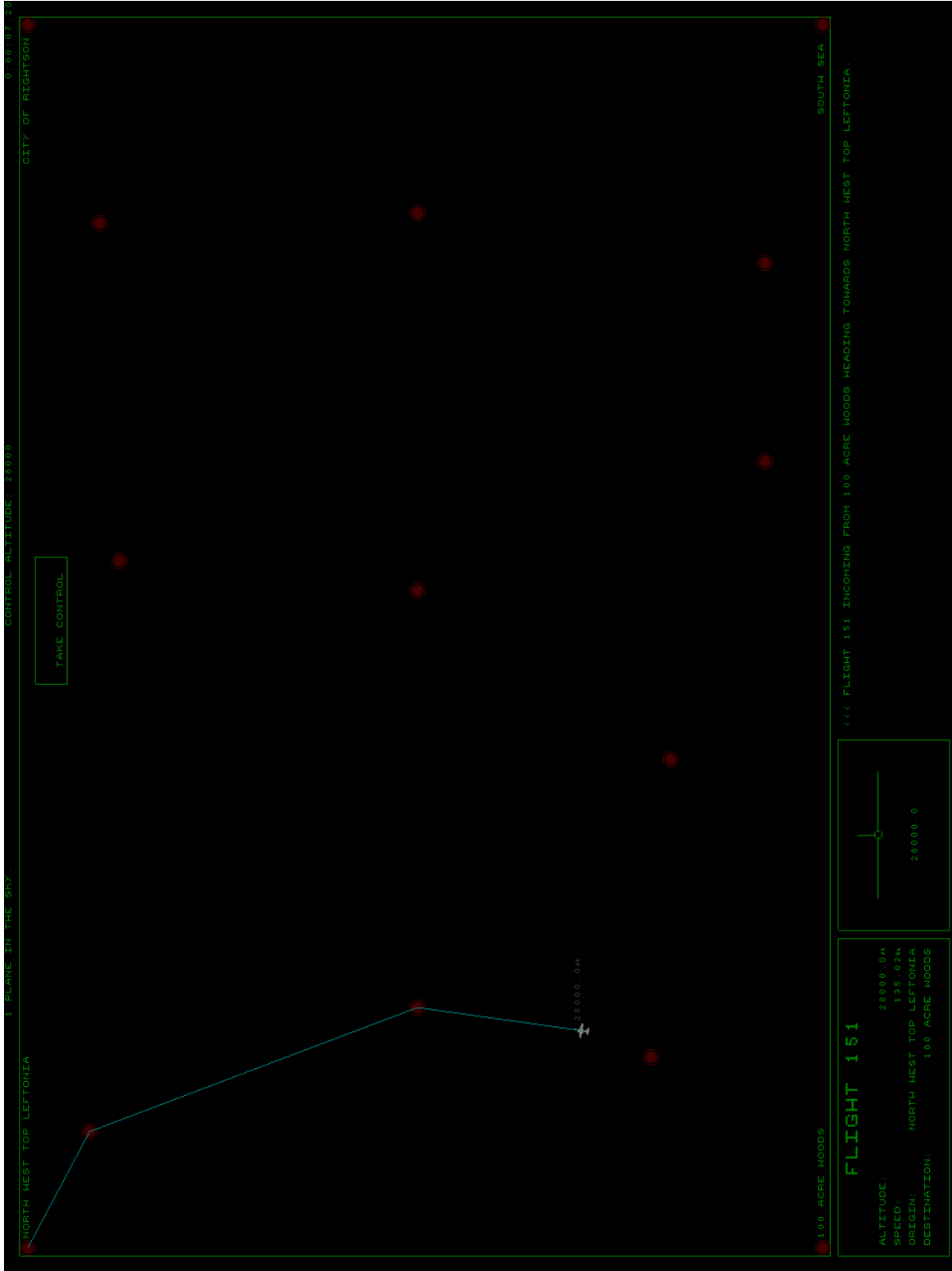


Figure 7: Game Over Screen

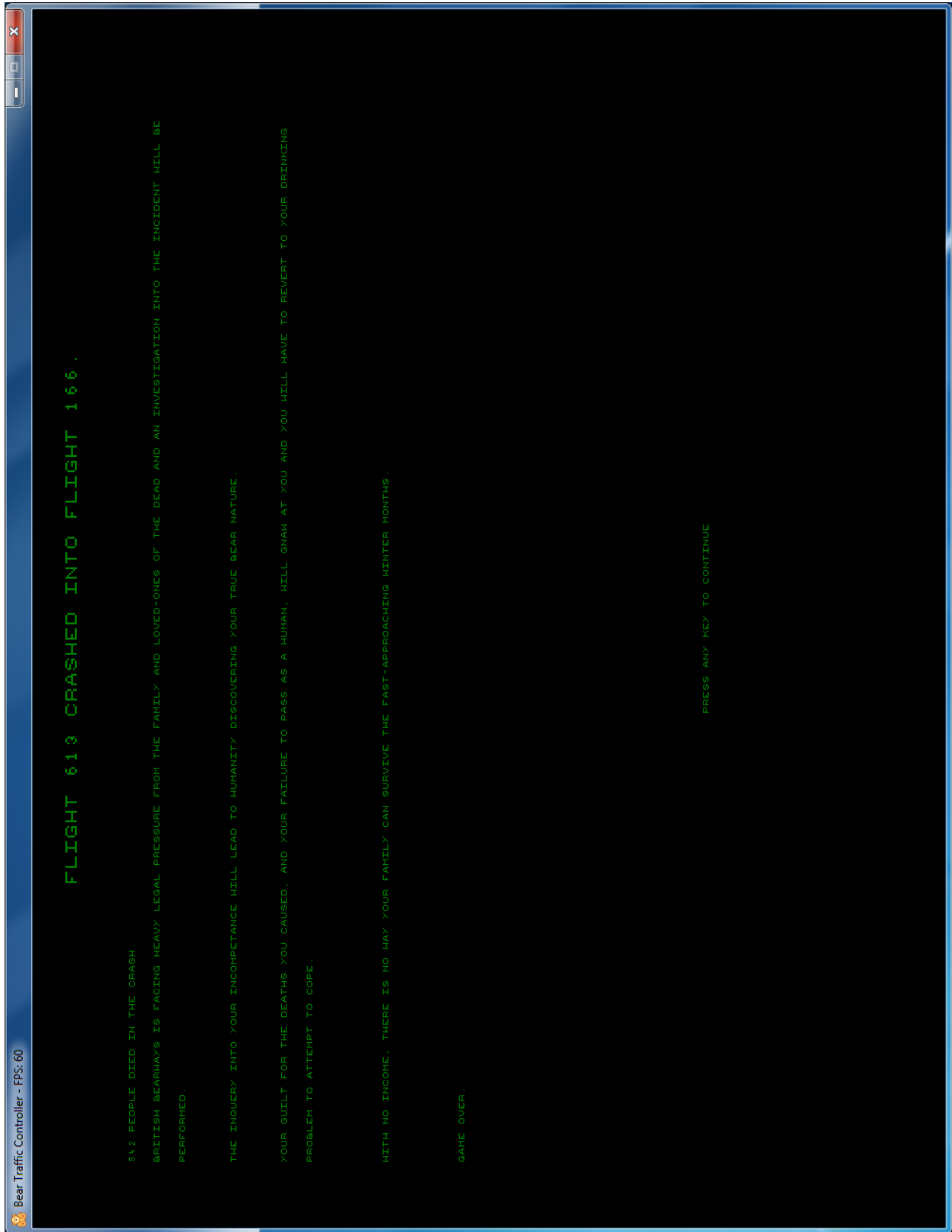
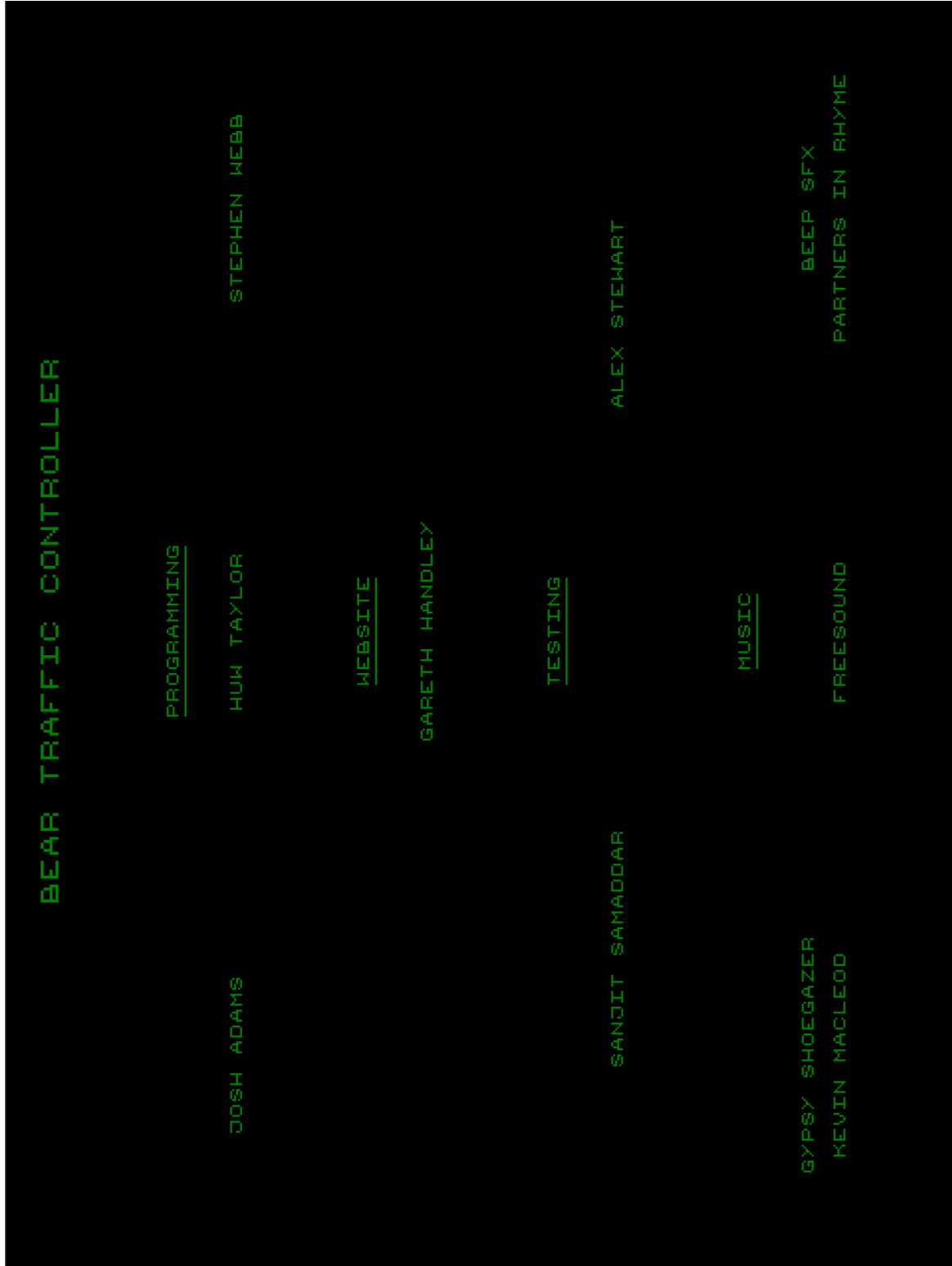


Figure 8: Credits Screen



Appendix C - Original Requirements

Functional Requirements

1. The game GUI shall display the game airspace and active flights.
2. The game shall initialise aircraft entering the airspace with a static flight plan. The static flight plan will be a route passing through waypoints from the planes entry to it's exit point.
3. The game shall allow the player to send orders to aircraft to immediately alter their flight plan.
4. There shall be at least 3 entry and 3 exit points and the exit points will correspond to given destinations.
5. The game shall check aircraft separation regularly.
6. The game shall end when the distance between one aircraft and another is lower than the specified separation rules.
7. The game shall track an aircraft's position and speed while it is within the airspace.
8. The game shall calculate the score as a function of time played and successful flights.
9. The game shall run on PCs provided in the Computer Science labs.
 - 9.1. The game shall run on a PC that meets the Windows 7 minimum system requirements^[1] (1GHz processor, 2GB of RAM).
10. The game shall allow the use of the keyboard and mouse as input devices.
11. The game shall allow for separation rules of aircraft to scale against difficulty.
12. The game shall provide at least ten fixed waypoints within the airspace.
13. The game shall allow aircraft to vary the rate at which they climb / descend.

Non-functional Requirements

1. The game shall incorporate realism where possible/appropriate, but its primary goal is to provide a fun and enjoyable experience to the user.
2. The game shall be engaging.
3. The target audience of the game will be students (not necessarily Computer Science students).
 - 3.1. The game will be appropriate for this market in terms of the graphics and language used.
4. The graphics of the game shall not be restricted by reality, but will instead aim to provide a fun and engaging experience for the user.
5. The game GUI shall be updated regularly. The target for this is every 30ms.
6. The game shall be easy to play and clear instructions will be available when needed.
7. The game shall be adaptable for touch input.
8. The game shall use semi-realistic separation rules.
9. The game shall use units which fit the air traffic control domain, but which are not so obscure as to confuse the user. These shall be:
 - 9.1. Knots for speed (As this relates closely to MPH and is what is used in real life).

- 9.2. Feet for altitude (People have an understanding of feet and it is used in the real world).
- 9.3. Feet for separation.