

LCARS Terminal Starship Commander

David Jarrett (Internet Computing Student)
Supervisor: Prof William Latham

March 17, 2008

1 Abstract

The LCARS Terminal[5] Starship Commander will attempt to occupy a niche in Star Trek[4] gaming, by providing the ability for a player to take control of a starship, from the creation of the ship and it's crew, to taking on missions that aren't just solely based on combat and boldly going out into the wide unknown.

The game fills the niche in two ways. Firstly, it will allow for an infinite number of missions of varied types by providing for the community to create missions that they can then go on to play. Secondly, the actions of the player in these missions will be able to influence the outcome and success of each particular mission they play.

To achieve these aims, the game will first provide the ability for the player to choose a starship and design a crew, and then to provide for these missions to be randomly picked from a list of available missions on the player's computer. Once picked, the player will be able to play the mission through to completion. These missions will be able to be created using an in-built mission editor, which should match the flexibility of these missions with user friendliness in creating them. Finally, the game should allow for two or more players to join together to complete missions together in a multiplayer environment.

To achieve this, there will be two frontend aspects to the game. Firstly, the game will allow for missions to be created by the community of players using the in-built mission editor. Secondly, through the primary game interface, these missions can then be played by players using their pre-constructed ships and crews. Both the mission editor and the game will have to be built from the ground up. There also needs to be a backend database to allow for the data that is used by the game to be stored. The LCARS Terminal community will be used to user-test the features and provide feedback on the functionality of the game.

Once complete, LCARS Terminal Starship Commander will allow for missions to be created and played by the same player, to allow for the freedom and flexibility of those missions and for the player-made-Captain who is playing those missions, and will bring the ability for players to take control of a starship right home onto their computers, providing the same feel as if they were themselves inside the Star Trek universe.

2 Introduction

LCARS Terminal Starship Commander was conceived from experience in playing other games of the same genre, such as the well renowned Star Trek Bridge Commander[1] and Star Trek: Starfleet Command III[2]. These games both allow for the player to take command of a starship, but it is what they lack that has inspired this project. The game will be building on such previous game's failures, in that the previous games are often focused solely on the combat area of Star Trek.

3 Background

LCARS Terminal Starship Commander is based on many other similar games all operating in the same genre. This section will go through the four or five main games where LCARS Terminal Starship Commander draws inspiration on.

3.1 Star Trek Bridge Commander

Star Trek Bridge Commander[1] is a highly graphical game that allows players to take control of a starship. The game provides a single player story which puts the player in one of two provided starships with a very much setup crew, and steers the player through a linear but nonetheless entertaining storyline. At the end of this storyline, the game simply reverts to it's secondary mode of random combat scenarios, either in computer simulations or with other players. This really is the program where potential to make it great was seen, but the failure to allow the player to choose his/her own crew or create additional storylines was missed.

3.2 Star Trek: Starfleet Command 3

Star Trek: Starfleet Command 3[2] is another highly graphical game that allows players to take control of a starship. The game also provides a single player story which puts the player through three independent but interlinked campaigns. Through these campaigns, the player can upgrade and manage their starship and crew. The game also provides a Career Mode, which is extremely similar to the game I am trying to create. Players can choose their Starship and Crew, upgrade the Starship and Crew and go on randomly chosen missions. However, these missions are often focused on Combat, and are completely random in that they don't supply a very good storyline with the missions, and so fail to pull the same effect as in the Star Trek series.

3.3 Traditional RPGs

Traditional RPGs, such as the Obsidian Fleet[3], allow players to take upon any member of the crew of a multiple array of starships. Through the supreme power of the imagination, they allow for anything and everything to happen while the player is controlling that crew member. However, players are relying completely and utterly on the output of other players. LCARS Terminal Starship Commander will take the storytelling of Traditional RPGs and place it in a controlled environment, similar to reading books of other people.

4 Method

4.1 Tables

	Name
Task	A Initial Analysis and Design
	B Written Analysis and Design
	C Implementation of Database
	D Preparation of Frontend Resources
	E Implementation of Mission Editor
	F Implementation of Main Game
	G Integration of Implementations
	H Testing
	I Evaluation
	J Updating Game based on Feedback
	K Writing Up Final Report

Table 1: Task List

		Month											
		Jun 08	Jul 08	Aug 08	Sep 08	Oct 08	Nov 08	Dec 08	Jan 09	Feb 09	Mar 09	Apr 09	
Task	A												
	B												
	C												
	D												
	E												
	F												
	G												
	H												
	I												
	J												
	K												

Table 2: GANTT Chart

4.2 Timeline Description

As can be seen by the two tables presented, the project will take around 11 months to complete, of which 3 of the months will be used to analyse and design the aspects of Star Trek that will be integrated into the game, and 7

of the months will be used for coding and then integrating the three proposed 'main modules' of the project; the database, mission editor and main game.

As stated in the previous paragraph, the first 3 months of this project will be used to analyse and design every aspect of the program, pinpointing on the aspects of the program that are most important, and ensuring that all the necessary data is ready for the implementation of the database in September. This will be made up of three main sub-tasks; firstly, of communicating with the community at LCARS Terminal forums to ensure that their desires are understood; secondly, by looking at other games in the genre to analyse their strengths and weaknesses in depth, as well as looking over the data available that can be utilised in the game; and thirdly, by analysing Java, to ensure that the requirements of the design can be equally met with the capabilities of the programming language.

Once this initial stage is complete, the findings will be written up and the programme properly designed from stem to stern in order to produce an accurate and time-effective design plan. This can then be used to build the system, which will begin in September. The first task of the actual implementation will be the database. During this, the data will be analysed from the previous three months and the database constructed to support such data. At the same time as this, the resources for the frontend of the game and mission editor will be brought together, which will include the necessary graphics, sound effects and animations for the game.

Once this stage is complete, around October/November, the Mission Editor will be built, utilising the frontend resources and the database that will have been put together. The Mission Editor will allow for the construction of missions to be used in the main game, and is therefore required for the main game to be constructed to ensure that the main game has access to the same structure of missions produced by the editor. Around December, the Editor should have been put together enough for the mission data holder to be understood, and construction on the main game can begin. Once again, the game will utilise the database, and eventually the two 'frontend' segments will be joined together into a single application framework as well; this will all be part of the integration of the implementation.

The programming should be complete by March, where the program will be evaluated and beta tested through the release of private betas to the community at LCARS Terminal. This will provide an evaluation that will allow for mistakes in the programming and small amounts of missing but important functionality to be corrected. Following this, the main project report will be written up, and the program is expected to be released as a public beta in early May, followed by a formal release near the end of May.

5 Evaluation

The actual testing and evaluation tasks have not been expressed in great detail but are nonetheless a vital part of any project, and failure to perform them

effectively could lead to failure in the project. Therefore, the two frontend modules (the mission editor and the main game) will be tested thoroughly, and the integrated game will also be tested and evaluated.

The game will be tested constantly throughout the implementation phase, as shown by Task H in the GANTT Chart on page 4. This will involve three types of testing. Firstly, the code of the game will be tested in very small segments (in Java, these are called ‘methods’), which will ensure that each method does exactly what it is supposed to do. This will utilise White Box testing (the testing of the internal code) at a very deep level, to view the passage of data through the code and ensure that for every available input there is an appropriately produced output.

Secondly, each of the two frontend modules of the game will undergo segment or unit testing. This will mainly involve Black Box testing (that is the testing of the system without following it through in the code). This will ensure that each segment of each module does exactly what it is supposed to do in relation to the entire module.

Finally, the entire program will be involved in system tests. This will involve checking the statistical output of the game, such as how much RAM or CPU the game utilises, to bring up a checklist of hardware requirements. It is one of my objectives to ensure that the game is able to run on low-end machines, and this test will test this.

At the end of the testing phase, the game will be provided to the LCARS Terminal community to be evaluated. Not only will this provide a wider range of input for the Black Box testing (to catch any missed bugs) it will allow for the user to report on the frontend interface as well as the inherent functionality of the game and provide for changes to be made based on that feedback.

6 Conclusion

LCARS Terminal Starship Commander aims to rectify the failure of other games in the genre by allowing for the player to take control of a starship and go on randomly chosen missions. These missions will be of varied types, other than just combat, that the community can add to while allowing these missions to be played out dynamically by players in the game. The missions will be dynamic in that their outcomes will be able to be influenced by the player as they play, and they will each have the possibility of providing a well thought out story. Finally, although the game will be primarily single-player, it will allow for multiplayer interactivity as well.

7 Bibliography

References

- [1] Activision. Star trek bridge commander (<http://www.bridgecommander.com/>), 2002.
- [2] Activision. Star trek: Starfleet command 3 (<http://starfleetcommand3.filefront.com/>), 2002.
- [3] Obsidian Fleet. Obsidian fleet (<http://www.obsidianfleet.org/>), 2001.
- [4] Paramount. Star trek (<http://www.startrek.com/>), 2008.
- [5] Piet van Stralen. Lcars terminal (<http://www.lcars-terminal.net/>), 1996.