

Warhammer 40000 Legiones Astartes Age of Darkness Army Selector/ List Printer

Ian Nevills
James Laubach
Nicholas Reid

Abstract

To condense the army selection book (hitherto: Legiones Astartes Book) and the rules necessary for the play of that faction specifically from the Core Rules Book into an easily navigable format to allow for the quick construction of army lists/builds for veteran and novice players of Warhammer 40000 alike.

1. Introduction

Warhammer 40k is an expansive and very complex tabletop game and its rules can be daunting even for veterans of the game. For novices, navigating thick tomes of factions upon subfactions of units and variants to forge their chosen army list can be nearly impossible. Our goal is to condense the faction book of one faction (Legiones Astartes, "Space Marines") within the setting of the "Horus Heresy" expansion to base 7th edition Warhammer 40k to streamline army creation and hopefully streamline game play by providing an army-list document that provides reference to particular rules. With that in mind, however, we are not seeking to make it so that any person could use our program in lieu of buying the particular books in question. Our plan is for our program to: Allow the user to search/page through a catalogue of units pertinent to the faction in question, and select units (and eventually upgrades) within the bounds of standard army templates in the Legiones Astartes Book and point limits imposed on normal game play, and to output a formatted and easily navigated document that provides information (basic stat lines, a list of upgrades, list of rules (but no detailed text), and an index indicating where further information can be found to clarify each) to streamline game play.

1.1. Plan Goal 1.

None.

1.1.1. Plan Goal 2.. None.

1.2. Background

Warhammer 40000 Legiones Astartes Age of Darkness Army List
Warhammer 40000 Core Rules Book 7th Edition

Primer:

Warhammer 40k is a tabletop science fiction strategy game set in a dark and distant future. Within 40k there are many different factions with equally different game-play fact and origin stories. Warhammer 40k is a game literally buried in lore and reasons for the how and why its factions behave and interact the way they do. In fact, a more accurate description may be a tabletop science FANTASY game given that almost all the standard fantasy races anyone familiar with Dungeons and Dragons or Tolkien's novels would recognize. Though, with that recognition comes the next problem, each of these races is at first familiar but also very different than their traditional archetype. A further description though is beyond the scope of what I intend to cover at this time. For now, I will cover the subject of this project.

The Space Marines:

For our part, we have chosen the Space marines as our subject for this project, they are probably the most widely recognized, if not known to be a part of the setting, and are the likely poster children to introduce new players into the setting via their use in novels and games based on the setting. Space marines in the context of Warhammer 40k, one part warrior monk, one part Roman legion, one part crusading host, and one part Arthurian legend, wrapped in a ton and a half of powered armor and genetic engineering. In the lore of Warhammer 40k, they are the ultimate soldier and the peerless defenders of Humanity, sent into the harshest fighting and the most important conflicts.

The actual game-play is much less....stellar. At their core, the Space marines are a force dedicated to somewhat durable heavy infantry formations with a wide selection of weapons and upgrades and a fairly well rounded stat-line. Everything else in their lineup are either walkers intended for direct support of those infantry or various vehicles and tanks that almost universally double as battle taxis to ensure the infantry reach where they need to be and provide a large enough wall of guns to ensure they can do their job when they get there.

The Horus Heresy:

The Horus Heresy is one part expansion and one part prequel to the main Warhammer 40k setting, and is the scope of our project. The Horus Heresy is a civil war taking place in the 30th Millennium, ten thousand years before.

1.3. Challenges

Our challenges are that the more information we wish to add up to our full goal, the potential complexity and required time to complete the program grows exponentially. There is a lot of information in Warhammer 40K that must be processed for basic game play such as building an army and recognizing factions, unit types, and unit upgrades. Each part has different rules for game play as how factions interact, strength and weaknesses of unit types, and how each unit is benefited by upgrades. So with each new piece of information the complexity of the program increases.

2. Scope

Our bare bones level of functionality that we would call "complete" is the ability to choose an army organization, choose units, and to force users to select army composition within the confines of the army organization. When choosing units our bare functionality would have the system display the unit name, the unit index number, the unit base stats and point value, a text box with stat definitions, and the unit composition.

A near term stretch goal would be to recognize the point limit imposed by the game and to take the point value of units and to tell the user that they have exceeded the point limit imposed by game play.

A stretch goal would be to allow the selection of upgrades, show how they alter the point count, and provide an index to the appropriate page for each upgrade to allow the user to find more detailed information needed for game play. A further goal down that road would be to list appropriate rules to each upgrade (and the base units) and allow another index to provide a similar ability to quickly find needed information for those as well.

The second main stretch goal would be to make the program be able to read back in its output list (or a code that would generate an identical list) to allow a user to edit a list after the fact rather than rebuilding it from scratch.

2.1. Requirements

2.1.1. Functional.

- User need to have container to hold information.
- User needs to be able to add and delete information as needed.
- User needs to be able to create, identify, and access information file.

2.1.2. Non-Functional.

- UI that the user interacts with is easily navigated and simple to learn.
- Make the program adaptable so that a different database (say after an update to the game) could be used instead of the one provided, as long as the format stays the same.
- Ensure that the database is accurate to the source material.

2.2. Use Cases

Use Case Number: 1

Use Case Name: Add Units

Description: User wishes to add a unit to their army. Look the list of units and Click add unit to list.

- 1) Navigate with button to list with the desired units.
- 2) User clicks add unit to list button.
- 3) The list is updated to reflect the new unit in the army.

Termination Outcome: The user has a new unit in the army list.

Use Case Number: 2

Use Case Name: Delete unit

Description: User wishes to delete a unit from their army. Look at users list and click Delete button.

- 1) Navigate to the users list through the interface
- 2) User clicks a delete button.
- 3) The list is updated to reflect the new army.

Termination Outcome: The user has lost a unit from their army.

Use Case Number: 3

Use Case Name: Save user list

Description: The user wants to save their unit list.

- 1) Navigate to the users list through the interface
- 2) Click the save list button.
- 3) The current list is saved.

Termination Outcome: The current user list is saved.

2.3. Interface Mockups

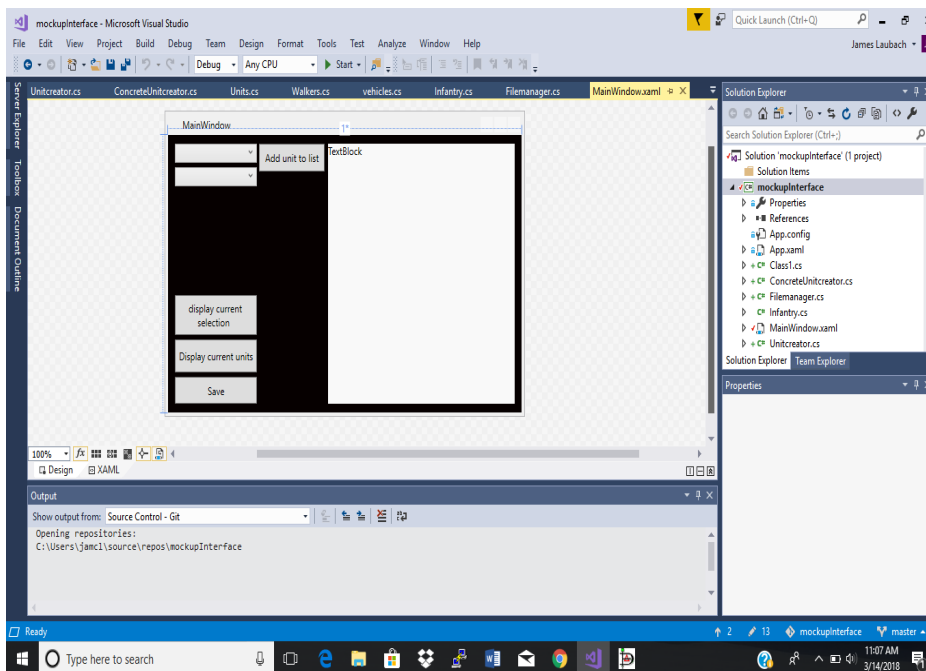


Figure 1. First picture, This is an Interface Mockup

3. Project Timeline

2/15/18 — Presented proposal to the class

2/28/18 — Initial Database (Unit names, index, point values) finished and undergoing conversion to Access format.

3/2/18 — Initial database fully converted to Access, UML finished, timeline finalized, design patterns chosen.

3/7/18 — Text describing stat-line figures and copyright information established, sample list format 1 ready for review.

Prototyped classes,

3/15/18 — Next check in ready, UI demonstrating partial functionality, at least able to access initial database

3/22/18 — One or two main classes fully implemented and functional, with list builder fully functional with initial database

3/29/18 — Adding information to initial database to form final database (Unit names, index, point values, composition, statlines), begin altering three complete classes to suit

4. Project Structure

At first, this will be a little empty (it will need to be filled in by the time you turn in your final report). This is your chance to discuss all of your design decisions (consider this the README's big brother).

4.1. UML Outline

The UML outline for our program

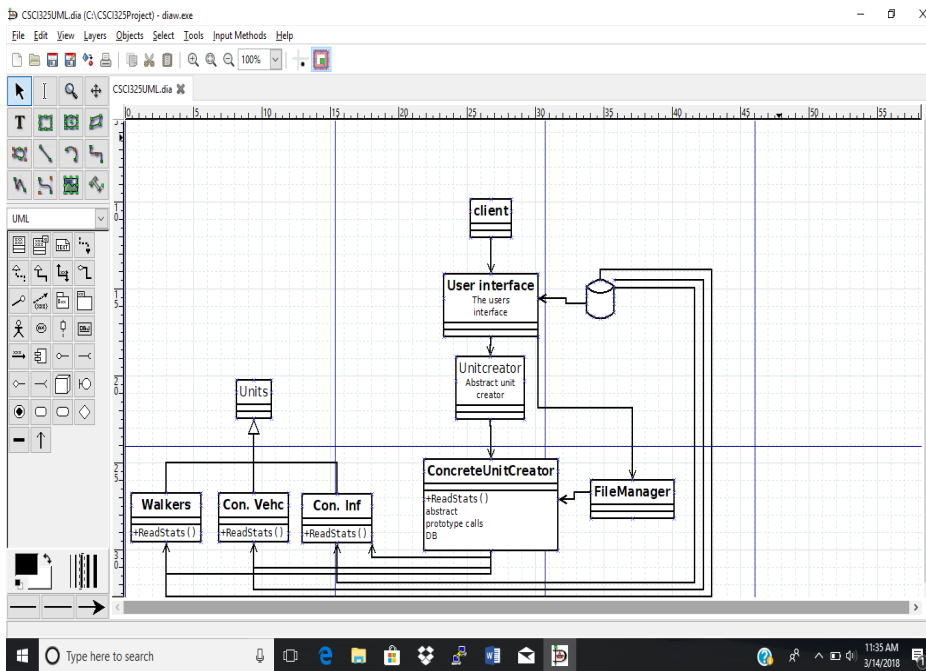


Figure 2. Our most current UML design

4.2. Design Patterns Used

Make sure to actually use at least 2 design patterns from this class. This is not normally part of such documentation, but largely just specific to this class – I want to see you use the patterns!

5. Results

This section will start out a little vague, but it should grow as your project evolves. With each deliverable you hand in, give me a final summary of where your project stands. By the end, this should be a reflective section discussing how many of your original goals you managed to attain/how many desired use cases you implemented/how many extra features you added.

5.1. Future Work

Where are you going next with your project? For early deliverables, what are your next steps? (HINT: you will typically want to look back at your timeline and evaluate: did you meet your expected goals? Are you ahead of schedule? Did you decide to shift gears and implement a new feature?) By the end, what do you plan on doing with this project? Will you try to sell it? Set it on fire? Link to it on your resume and forget it exists?

References

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.