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 neccessary 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.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Add item to cart	Shopper	Med	1
2	Checkout	Shopper	Med	1

TABLE 1. SAMPLE USE CASE TABLE

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 aforementioned army organization and point system 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

As part of fleshing out the scope of your requirements, you'll also need to keep in mind both your functional and non-functional requirements. These should be listed, and explained in detail as necessary. Use this area to explain how you gathered these requirements.

2.1.1. Functional.

- User needs to have a private shopping cart – this cannot be shared between users, and needs to maintain state across subsequent visits to the site
- Users need to have website accounts – this will help track recent purchases, keep shopping cart records, etc.
- You'll need more than 2 of these...

2.1.2. Non-Functional.

- Security – user credentials must be encrypted on disk, users should be able to reset their passwords if forgotten
- you'll typically have fewer non-functional than functional requirements

2.2. Use Cases

This subsection is arguably part of how you define your project scope (why it is in the Scope section...). In a traditional Waterfall approach, as part of your requirements gathering phase (what does the product actually *need* to do?), you will typically sit down with a user to develop use cases.

You should have a table listing all use cases discussed in the document, the ID is just the order it is listed in, the name should be indicative of what should happen, the primary actor is typically most important in an application where you may have different levels of users (think admin vs normal user), complexity is a best-guess on your part as to how hard it should be. A lower number in priority indicates that it needs to happen sooner rather than later. A sample table, or Use Case Index can be seen in Table 1.

Use Case Number: 1

Use Case Name: Add item to cart

Description: A shopper on our site has identified an item they wish to buy. They will click on a "Add to Cart" button. This will kick off a process to add one instance of the item to their cart.

You will then go on to (minimally) discuss a basic flow for the process:

1) User navigates to page listing desired item
2) User left-clicks on "Add to Cart" button.
3) User cart is updated to reflect the new item, this also updates the current total.

Termination Outcome: The user now has a single instance of the item in their cart.

You may need to also add in any alternative flows:

Alternative: Item already exists in the cart

1) User navigates to page listing desired item
2) User left-clicks on "Add to Cart" button.
3) User cart is updated to reflect the new item, showing that one more instance of the existing item has been added. This also updates the current total.

Termination Outcome: The user now has multiple instances of the item in their cart.

You will often also need to include pictures or diagrams. It is quite common to see use-case diagrams in such write-ups. To properly reference an image, you will need to use the `figure` environment and will need to reference it in your text (via the `ref` command) (see Figure 1). NOTE: this is not a use case diagram, but a kitten.

After fully describing a use case, it is time to move on to the next use case:

Use Case Number: 2

Use Case Name: Checkout

Description: A shopper on our site has finished shopping. They will click on a "Checkout" button. This will kick off a process to calculate cart total, any taxes, shipping rates, and collect payment from the shopper.

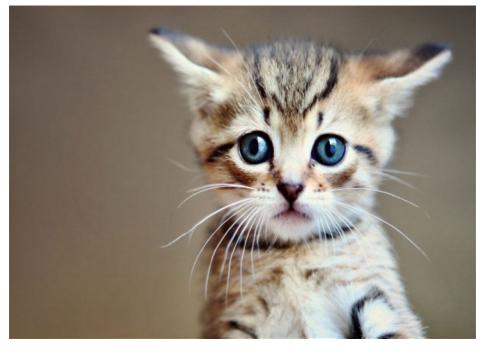You will then need to continue to flesh out all use cases you have identified for your project.

Figure 1. First picture, this is a kitten, not a use case diagram

2.3. Interface Mockups

At first, this will largely be completely made up, as you get further along in your project, and closer to a final product, this will typically become simple screenshots of your running application.

In this subsection, you will be showing what the screen should look like as the user moves through various use cases (make sure to tie the interface mockups back to the specific use cases they illustrate).

3. Project Timeline

Go back to your notes and look up a typical project development life cycle for the Waterfall approach. How will you follow this life cycle over the remainder of this semester? This will usually involve a chart showing your proposed timeline, with specific milestones plotted out. Make sure you have deliverable dates from the course schedule listed, with a plan to meet them (NOTE: these are generally optimistic deadlines).

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

Show the full structure of your program. Make sure to keep on updating this section as your project evolves (you often start out with one plan, but end up modifying things as you move along). As a note, while Dia fails miserably at generating pdfs (probably my fault), I have had much success with png files. Make sure to wrap your images in a `figure` environment, and to reference with the `ref` command. For example, see Figure 2.

Figure 2. Your figures should be in the *figure* environment, and have captions. Should also be of diagrams pertaining to your project, not random internet kittens

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 LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.