
This space is reserved for the EPiC Series header, do not use it

Chaotic Creations

Natalie Arnold, Leah Kramer, Christopher Lewis, Terra Williams,
Sergiu M. Dascalu, and Frederick C. Harris, Jr.

Department of Computer Science and Engineering
University of Nevada, Reno

{narnold,lkramer37,christopher_le1}@nevada.unr.edu, terraw9@gmail.com
{dascalus,fred.harris}@cse.unr.edu

Abstract

Chaotic Creations is a Windows application aimed toward Dungeon Masters of the fifth edition of Dungeons and Dragons. It includes a dice roller visual, a database of random-selection tables, as well as tools for generating non-player characters, encounters, and terrains. The application is built using Windows Presentation Foundation (WPF) with C# being used for the back end and XAML used for the front end. SQLite was used for database creation and management. The motivation for this project is to help Dungeon Masters be able to improvise during gameplay more effectively.

Keywords: Database, Software, D&D, WPF, XAML, SQLite, Windows Application

1 Introduction

Chaotic Creations has a place in the current D&D software market as a useful tool for new and experienced dungeon masters. The market potential will be analyzed for the proposed product and evidence of user interest will be presented. Competitive products such as Auto Roll Tables will be examined as well as other sources of indirect competition. Finally, the differences between Chaotic Creations and these other products will be outlined as well as the features that make the Chaotic Creations proposal concept better than the competition.

Dungeon and game masters of RPG games are saddled with the monumental task of creating fully immersive environments rich with detail. One of the largest issues with this task is being able to think on your feet when players do something unexpected. Whether during game creation or in the middle of a session, a game master always needs resources for details in their games. Chaotic Creations would give a game master the ability to come up with these details on the spot with ease. Geek & Sundry, a site famous for its involvement in the D&D community, outlines how to create a successful gaming experience and describes the necessity for game masters to come up with game characteristics on the spot [3]. In another guide to creating a D&D session, gamers are encouraged to find tools that work for them to create scenarios when their players deviate from the main storyline, improvising places, characters, and creatures as necessary [4].

The rest of this paper is structured as follows: Section 2 presents the background and literature review. The software engineering including Functional and Non-Functional Requirements

along with Use Cases is covered in Section 3. Section 4 describes how we implemented Chaotic Creations and our Conclusions and Future Work are presented in Section 5.

2 Background

There exist dozens of tools to help D&D players create and organize their games. From map generators to prewritten storylines, it seems that something exists to make every aspect of the game easier. Since Chaotic Creations focuses on the ability to randomly generate components instantly, comparisons will be made to both D&D Beyond and Auto Roll Tables.

2.1 D&D Beyond

D&D Beyond is a very robust application for running a D&D game, but it is very specific to the 5th edition of D&D and focuses mostly on the players' organization of their characters [2]. D&D Beyond would be considered a source of indirect competition since it focuses primarily on the players but still serves a substantial number of needs for the game. They have a player mobile pp to help keep track of players in the game in addition to their assistance to the DM.

2.2 Auto Roll Tables

Auto Roll Tables is a very similar concept to Chaotic Creations and would be considered a form of direct competition. Auto Roll Tables generates random characteristics of the game in a similar method, but Chaotic Creations aims to add more flexibility and additional tools to the dungeon masters such as a virtual dice roller and logic-driven tables [1]. Chaotic Creation's uniqueness and innovative features stem from the virtual dice roller, a modern UI, logically driven random generation, and terrain generation with the possibility of artistic renderings. With these benchmarks the success of the project will only increase depending on the extra amount of time team members involved can devote to the project.

3 Software Engineering

3.1 Requirements Specification

The requirements detailed in the following subsections include descriptions of the functional and nonfunctional requirements used by this project [7]. The functional requirements detail how different pieces of the project should act technically. The non-functional requirements describe any specifications and limitations that may exist in pieces of the project. The elicitation process was given to us at the conception of the project by other stakeholders that had a hand in the project.

3.1.1 Functional

1. Chaotic Creations will allow the user to interface with random tables saved in the database.
2. Chaotic Creations will allow the user to roll on random tables using a graphical dice roller.
3. Chaotic Creations will allow the user to select an entry of a random table without using the graphical dice roller.

4. Chaotic Creations will allow the user to generate a random encounter.
5. Chaotic Creations will allow the user to customize encounters based on difficulty, terrain, party size, and party level.
6. Chaotic Creations will allow the user to save generated encounters.
7. Chaotic Creations will allow the user to generate a description of a non-player character.
8. Chaotic Creations will allow the user to save generated non-player character descriptions.
9. Chaotic Creations will allow the user to generate a description of a location.
10. Chaotic Creations will allow the user to save generated location descriptions.
11. Chaotic Creations will allow the user to edit saved creations.
12. Chaotic Creations will allow the user to delete saved creations.
13. Chaotic Creations will generate an appropriate treasure reward for a generated encounter
14. Chaotic Creations will allow the user to search for a specific Random Table using a search bar.
15. Chaotic Creations will allow the user to search for a specific saved generation using a search bar.

3.1.2 Non-Functional

1. Chaotic Creations must run on Windows 10.
2. Chaotic Creations must return randomly generated results in less than 3 seconds.
3. Chaotic Creations will have a visually pleasing user interface.
4. Chaotic Creations will have a front end implemented with XAML[6].
5. Chaotic Creations will have a back end implemented with C#.

3.2 Detailed Use Cases

The use cases illustrated in Figure 1 are used to describe interactions between users and the system [7]. This is typically illustrated with actors and things they can do. A more descriptive account of these actions is next.

1. **Roll Dice:** The user will be able to choose between several dice to roll. Once the user selects a dice, a visual animation of the dice rolling will appear. The output will be the number that was randomly rolled.
2. **Generate Random Encounter:** The user will select the “Generate Random Encounter” option. The criteria for the encounter will be randomly selected from the corresponding tables in the database.
3. **Customize Random Encounter:** The user will be able to select or manually enter the criteria to create a random encounter. The dice visual will appear for each random table (if enabled). The user can then adjust the criteria from the output, if desired.



Figure 1: Use case diagram for Chaotic Creations showcasing the interactions within the system

4. **Save Generated Encounter:** The user will generate an encounter. The user can save the details of the encounter for later use. The encounter details will be stored on the user’s local machine.
5. **Generate Random NPC Character Description:** The user will select the “Generate NPC” option. The user can choose the criteria selections to be random and a random NPC character will be created.
6. **Generate Specified NPC Character Description:** The user will select the “Generate NPC” option. The criteria for the NPC will be specified by the user to create a customised generation.
7. **Save Generated NPC Character Description:** The user will generate an NPC. The user can save the NPC description for later use. The NPC description will be stored on the user’s local machine.
8. **Generate Location Description:** The user will select the “Generate Location” option.

The criteria for the Location will be selected by the user. The user can choose the criteria selections to be random or manually entered.

9. **Save Generated Location Description:** After the user has generated a location description, they can save it for later use. The location description will be stored on the user's local machine.
10. **Edit and Delete Saved Information:** The user can edit or delete previously saved information
11. **Generate Loot from Encounter:** Loot for the player will be generated based on the encounter that is generated
12. **Search Single Tables:** The user will be able to look up random tables to use in the game
13. **Navigate Saved Creations:** The user can use the search bar to look for specific saved creations

3.3 Traceability Matrix

Traceability policies “define the relationships between each requirement and between the requirements and the system design that should be recorded.” [7] The Traceability Matrix, as seen in Figure 2, is the tool used to track these relationships. Organizing these connections is essential in analyzing proposed changes and the impact they have on other parts of the system.

	UC01	UC02	UC03	UC04	UC05	UC06	UC07	UC08	UC09	UC10	UC11	UC12	UC13
FR01		■	■		■	■		■			■	■	■
FR02	■												
FR03			■					■					
FR04		■											
FR05			■										
FR06				■									
FR07					■	■							
FR08							■						
FR09								■					
FR10									■				
FR11										■			
FR12											■		
FR13												■	
FR14													■
FR15													■

Figure 2: Tracability Matrix where the UC prefix refers to Use Case and the FR prefix refers to Functional Requirements

4 Implementation

4.1 Architectural Design

Figure 3 gives a broad overview of the user’s available actions and how those actions might be accessed on a high level in a site map. The generators are separated into four categories: random tables, encounter generator, NPC generator, and location generators. These categories are differentiated by how many components will be generated as well as where these components come from. These components will be referred to as features. Random table generators will generate one feature, something that is stand alone like a world event or an amount of treasure. The encounter generator will generate several components made up of several features in order to create a combat encounter for the DM’s players. The NPC generator will output one character made up of several features. The location generator will create a description of an environment detailed with location features.

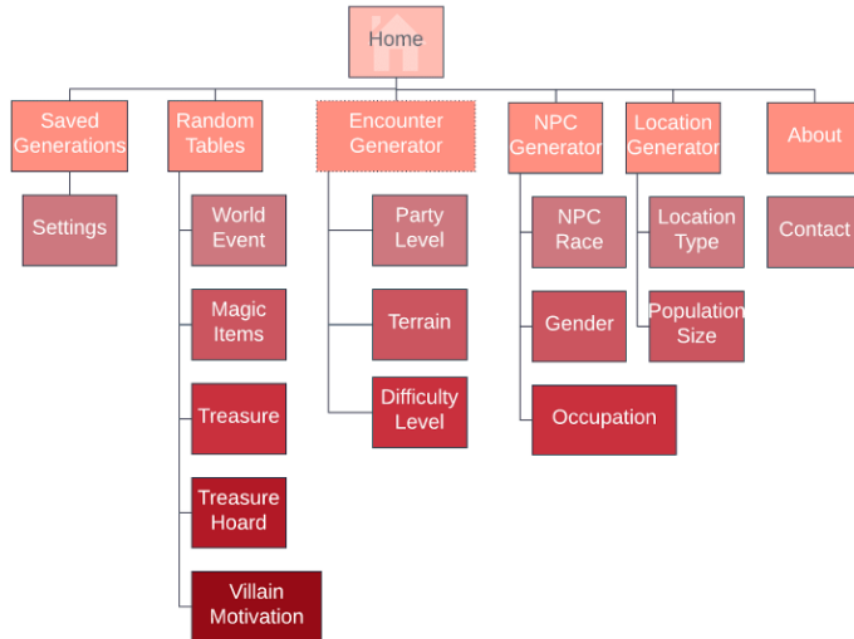


Figure 3: The site-map of Chaotic Creations illustrates the user’s actions in what they would like to generate and what kind of generator category their action would be related to.

Figure 4 represents a component-based high-level system design of the Chaotic Creations application. The Chaotic Creations Generating System has four main components: the GUI, User Account Management and Security, Multiple Feature Generation Management, and Single Feature Generation Management. To the user, generators for multiple featured components will be separated into categories but will be managed the same way in the software. There will also be two databases: one for the user’s account settings and saved generated components and the database containing D&D features to fill generated components.

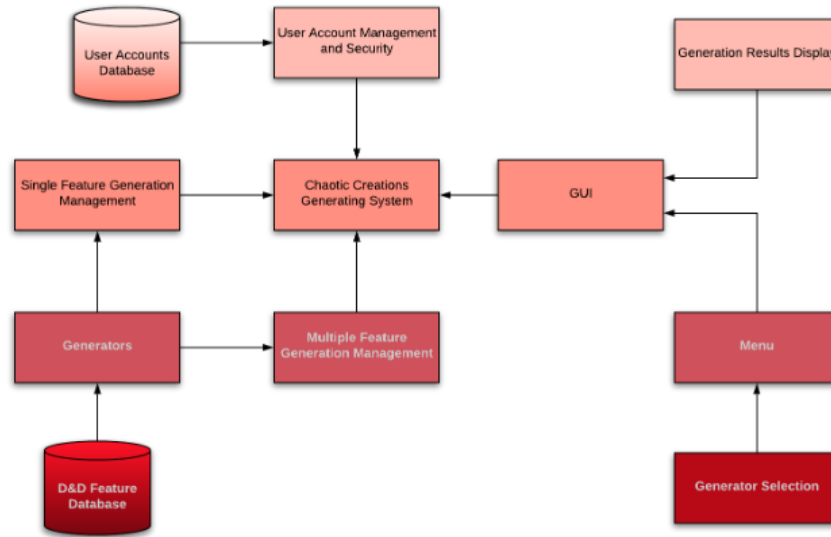


Figure 4: The component-based high-level system design shows how the data and management is divided and maintained.

4.2 Windows Integration

4.2.1 Visual Studio

Visual Studio is one of the technologies used. It is an IDE that provides an interpreter and a great editor for the Chaotic Creations GUI. We used it to interact with the C# programming language. There is also auto-complete and debugging support for C# natively integrated with Visual Studio. Lastly, Visual Studio also allowed quick and easy GUI development using their designer and with its native integration with XAML [6].

4.2.2 Windows Presentation Foundation

Windows Presentation Foundation (WPF) [5] is the connection between Chaotic Creation's front end, XAML, and its back end, Visual Studio or C#. WPF allows for the scripts in the back end to interact and change UI/XAML elements and lets the UI/XAML elements interact with the scripting elements. This needs to be a two way connection so that the front end can tell the back end what the user is trying to interact with and then the back end can deliver data to the front end to be shown to the user.

4.3 Libraries Used

4.3.1 XAML

XAML is used in conjunction with Visual Studio to create Chaotic Creation's GUI. Using these two tools together allows for the creation of a great windows interface through Visual Studio's graphical designer as seen in Figure 5. Visual Studio also allows for automatic text complete between the XAML and scripting parts of the program which makes scripting GUI elements very easy and intuitive.

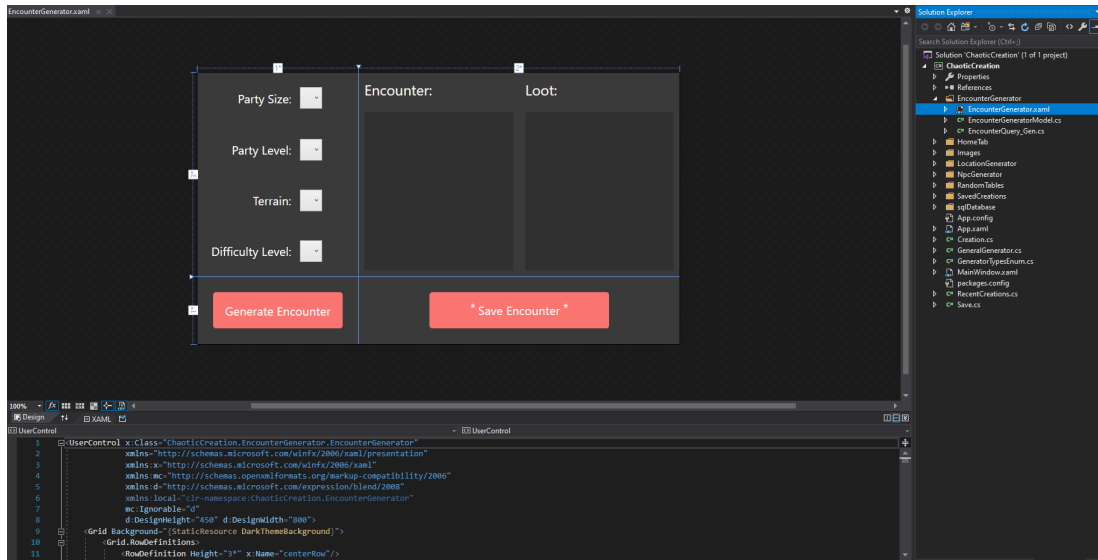


Figure 5: XAML visual interface for Chaotic Creation’s encounter generator tab

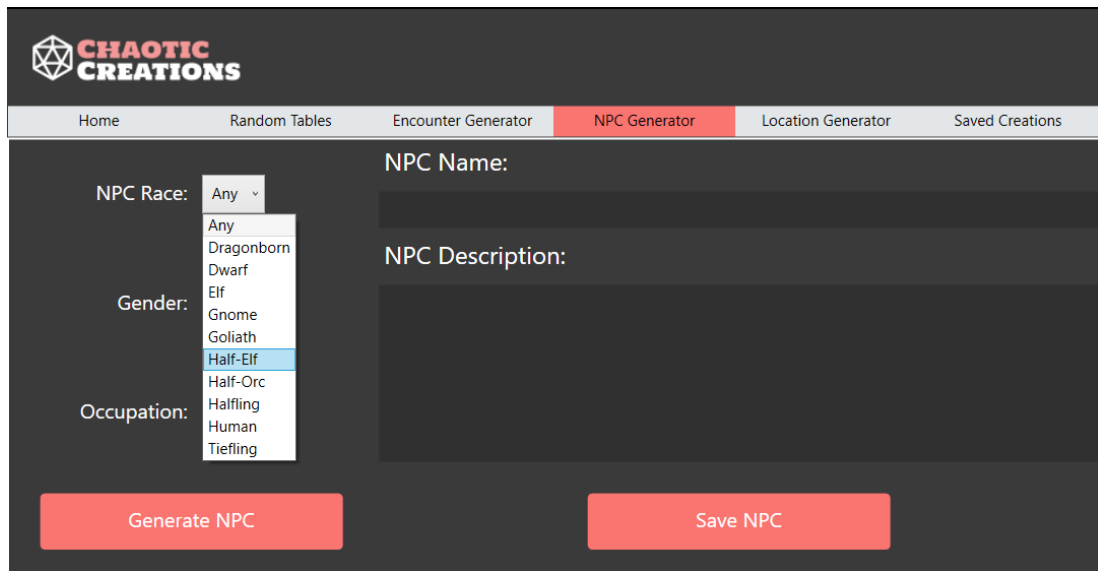


Figure 6: Race drop down menu for NPC generation

4.3.2 SQLite

SQLite was used to manage our database of thousands of generatable objects. C# has libraries that make querying very easy with SQLite. SQLite also allows for intelligent searching, which let us implement drop down menus and user search functionality, as seen in Figure 6 and 7.

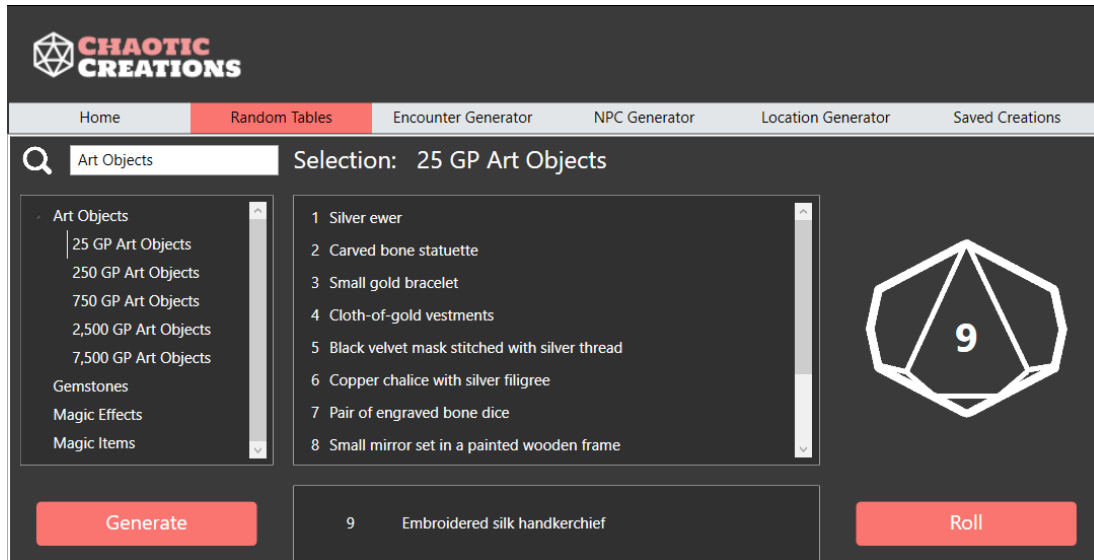


Figure 7: Random tables with search bar implementation and random roll

5 Conclusion and Future Work

5.1 Conclusion

Chaotic Creations is a tool designed for dungeon masters of Dungeons and Dragons fifth addition. This tool provides a variety of different use cases that make it novel. It uses WPF and XAML combined to create a pleasing GUI, Visual Studio and C# for scripting, and SQLite for the back end and database management.

5.2 Future Work

Chaotic Creations could be used in many different works in the future, and has many things that could be added to it or improved on it.

Map Generator: A virtual map generator would work really well in this project. The location generator is very small scale and handles one location at a time, but an entire world map, or even a city map, automates a huge portion of some of the tedium that DMs endure when creating new environments.

User Testing: Chaotic Creations should be much more widely user tested. As it is, only less than 10 people have had access to the tool, and while we are grateful for the feedback, it is very hard to get an accurate assessment of your tool's ability and usability when the reviewers are friends of the group and are therefore biased.

Boss Encounters and Encounter Differentiation: Chaotic Creations currently only pulls from a random list of monsters and only adds monsters to get to the point where the user specified experience values are satisfied. This means you can get encounters with two monster

factions that happen to be enemies fighting with each other, or the amount of monsters you have in the encounter can get really big. Some monsters can even come from completely different terrain and thus, probably wouldn't see each other. Lastly, Chaotic Creations could benefit from a boss encounter section of the encounter generator. Right now, Chaotic Creations only allows the DM to change the difficulty of the encounter, not how many monsters we want to generate or if that monster has additional treasure because it's a boss.

6 Acknowledgments

This material is based in part upon work supported by the National Science Foundation under grant number IIA-1301726. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

- [1] Auto roll tables. <http://autorolltables.github.io/>, August 2019. Accessed: July 6, 2020.
- [2] Patrick Backmann. D&D beyond. <https://www.dndbeyond.com/>, October 2019. Accessed: July 14, 2020.
- [3] James Haeck. How to write the best D&D adventures ever. <https://geekandsundry.com/how-to-write-the-best-dd-adventures-ever>, April 2016. Accessed: July 6, 2020.
- [4] Emily Heller. A beginner's guide to playing dungeons and dragons. <https://www.polygon.com/deals/21294556/dnd-how-to-play-dungeons-dragons-5e-guide-spells-dice-character-sheets-dm>, February 2019. Accessed: July 6, 2020.
- [5] Microsoft. Windows presentation foundation (wpf) documentation. <https://docs.microsoft.com/en-us/dotnet/desktop-wpf/>, 8 2019. Last Accessed: July 14, 2020.
- [6] Microsoft. Xaml overview in wpf. <https://docs.microsoft.com/en-us/dotnet/desktop-wpf/fundamentals/xaml>, 8 2019. Last Accessed: July 14, 2020.
- [7] Ian Sommerville. *Software Engineering*. Pearson, 2016.