

# Dungeons and Dragons 5E Character Sheet Creator

Andrew Newbill and Logan Brown

## Abstract

Andrew Newbill and Logan Brown are proposing a D&D 5E character sheet creator. This tool would use C# and WPF to provide an easy to use interface. Its main purpose is to create character sheets for the role-playing game Dungeons and Dragons which can be saved or loaded at will. We are targeting users who want an offline tool that is easy to use. This project aims to help new players through the creation of a character sheet(which defines the character in its entirety)so that they may begin playing without a Dungeon Master manually instructing them through the process. This project will pull from an existing database all of the items, spells, abilities, races, classes and other information necessary to define a character. Interactivity will be minimal although dice rolls for attacks and other actions are currently being targeted.

## 1. Introduction

The idea for a character sheet creator was borne from the need for a free tool for character creation that is fully usable offline. We are designing this tool with the aim of walking the user through the process of creating a character sheet for Dungeons and Dragons. The program would allow the user to fully define their character including: name, avatar, age, race, sex, statistics, origin, abilities, spells. One of the most common issues with playing Dungeons and Dragons is understanding how to properly create a character. Our project would alleviate this issue and allow users to start playing in the same session that they create their character. We are targeting those new to Dungeons and Dragons, since other more advanced tools exist but are often complex to the point of discouraging play. This tool will allow new users to invest nothing monetarily in order to begin playing. Simplicity is our overarching goal for this project in order to draw more players into the hobby. In this tool's final iteration it will need to allow fully creating a new character, and loading an existing character for viewing. We will also include exporting to image files so that the sheets may be printed and used for physical play.

### 1.1. Background

We decided upon this project quite quickly whilst brainstorming. We both have a passion for role-playing games and also recognized the need for this tool. Our goal is to get players into the hobby and we hope that this tool will facilitate that. Our experience playing the game will ease some of the research necessary in order to create this tool.

#### 1.1.1. Terms To Know.

- Dungeons and Dragons - A pen and paper role-playing game featuring fantasy elements.
- Character Sheet - A physical or virtual document which defines your character.
- Abilities - actions a character can perform that are not basic attacks.
- Spells - magic that a character can cast.
- Statistics - a character's traits, such as Intelligence or Strength, that are represented numerically.
- Origin- background information for a character.
- Avatar - a picture which visually represents your character.

### 1.2. Impacts

This program is planned to be free when released, so that it is accessible to everyone. In this way, it will spread quickly, allowing many of people (who may be unfamiliar with Dungeons in Dragons) to have what they need to have fun. The impacts that this will have are:

- Widespread encouragement of problem solving skills that are not exercised often
- Practice with minor mental math that many people frequently struggle with
- A way to make new friends, or connect with existing ones
- A general happiness and relief due to the simplicity of the application

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Character Creation	User	High	1
2	Character Upkeep	User	Med	2
3	Information Addition	User	Med	2

TABLE 1. USE CASES FOR ALL MAJOR CONCERNS

### 1.3. Challenges

The main challenge with this program is the complex rules relating to character creation in DnD. The program must make this a seamless process to the user. It is also important to consider the many ways the user may break the program. They could enter a very large value in a box meant for an integer value, or attempt to load a corrupted save file. The program must be aware of those possibilities.

**1.3.1. The GUI.** - Characters are complex, and need a lot of information in order to properly function in game. There is a lot of information, and the user has to be able to hold all of it in a GUI form. The KISS method will be our saving grace.

**1.3.2. Information.** - As mentioned before, there is a lot that goes into a character. With offline functionality being a basic goal, a database will be necessary.

**1.3.3. User Input Error.** - There will be a big mix of Integers, Strings, Doubles, and even more Strings in this project. Being sure that users are allowed to make mistakes with input *without* crashing the program is a big consideration. A possible error handling function for a few of the common data types might not hurt.

## 2. Scope

This project should meet multiple standards that are required in order to create and view a 5th Edition Dungeon and Dragons Character. A lot of these requirements are big and small, but are all equally important to the feeling of ease when creating your character.

### 2.1. Requirements

Here are the following minimum requirements, and under any of these can exist a stretch goal that under them improves upon the base idea:

- Have a Minimalist GUI for Character creation and viewing  
(Stretch Goal) Have an enhanced GUI that does not bore whoever uses it
- Full offline functionality
- Ability to create a 5th Edition Dungeon and Dragons Character (Handbook only)  
(Stretch Goal) User can add custom information, and use it how they like
- Need no outside tools to create a character (e.x. Dice, Character Sheet, Source Books, etc...)

#### 2.1.1. Functional.

- A simple GUI for ease of use - Users must be able to use this application easily without feeling lost.
- A database that the application can refer to for majority of the information it needs.
- Users must have enough space on the device that they use - There will be a decent amount of information from the data space, therefore needed room to store this data.

#### 2.1.2. Non-Functional.

- Additional Information - The user might want to add more information to the programs already long list, which should be very simple.

### 2.2. Use Cases

Use Case Number: 1

Use Case Name: Character Creation

Description: A user will be able to click create character, and then go through the process of creating a fifth edition character without any outside reference. After the full creation of the character, they can go back and view what they have made.

- Basic Flow:
- 1) User Selects Race
  - 2) User selects Class
  - 3) User Selects Background
  - 4) User Selects Items
  - 5) (Optional) User Types a Back-story

Termination Outcome: User has a ready to play, level 1 character. The User can review any information at any time.

Use Case Number: 2

Use Case Name: Character Upkeep

Description: Users will be able to keep up with their character as they play their game, and add/take away anything as needed (Items, Health Points, Stats, etc...).

- Basic Flow:
- 1) User decides which desired data needs to be changed/added.
  - 2) User clicks on data, then being allowed to modify it.

Termination Outcome: The desired data has been modified/added. If needed, the user can change it again, and again, and again until satisfied.

Use Case Number: 3

Use Case Name: Information Addition

Description: Sometimes the User may want to add custom items or data. They will be able to press a button to add items as they please.

- Basic Flow:
- 1) User Presses the Designated Button
  - 2) User Fill Out all needed Information
  - 3) Item is added to the users personal computer/device.

Termination Outcome: The New Data will have been added, and the user can manipulate it as needed.

## 2.3. Interface Mockups

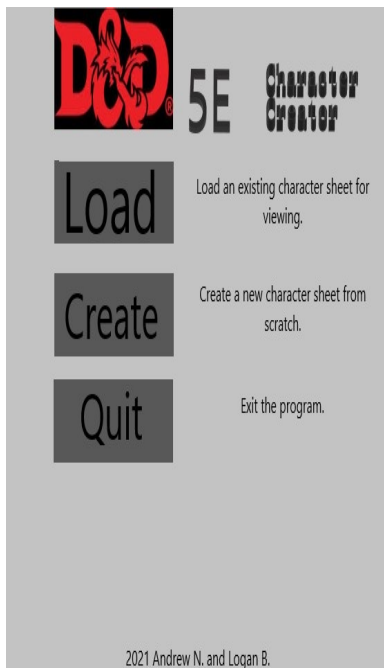


Figure 1. Main menu!

### 3. Project Timeline

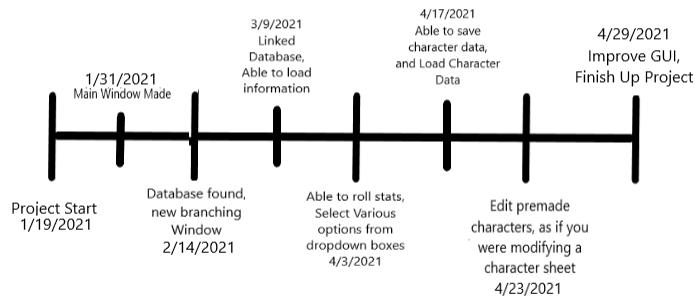


Figure 2. Here is a time-line of our project.

### 4. Project Structure

The project is structured around a CharDataContainer class. This class contains all of the information that the program will need to generate a DnD 5e character. It contains the name of the player, a Race object, a class object, as well as the player's level. At first, we wished to pull all of the information from the database about each race and class. This proved to be difficult, so for now the database is only used for getting the races/classes to fill the listboxes. Each class and race is hard-coded so if a race or class that is not defined is loaded in from the database, the program does nothing.

## 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 3.

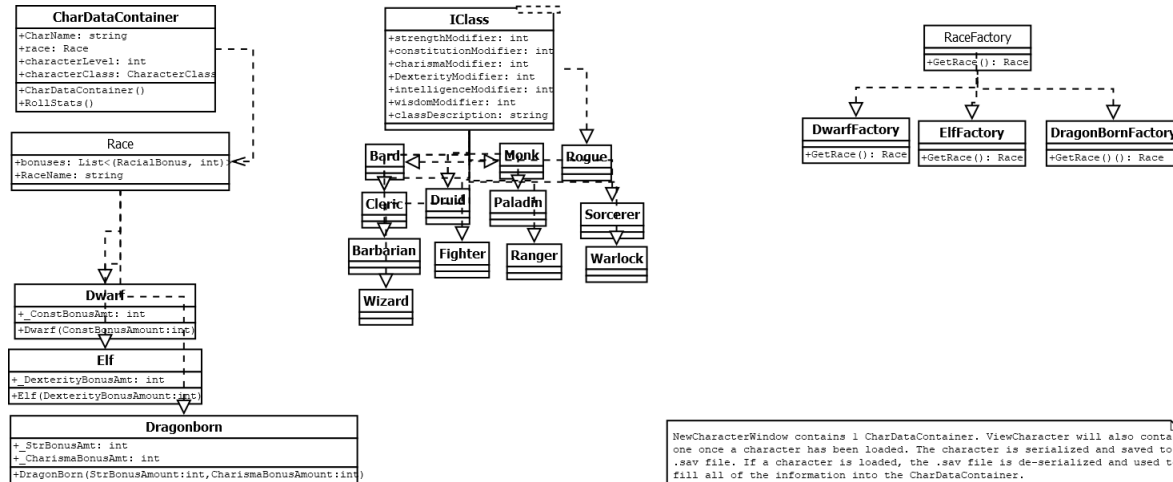


Figure 3. This UML does not represent the WPF side of things. NewCharacterWindow and ViewCharacter both contain a CharDataContainer. UI stuff is handled outside of this context.

## 4.2. Design Patterns Used

The factory method was used for the race of the player's character. Each factory is capable of returning a Race object using the GetRace() function. Each race gets its own factory. Each factory instantiates a concrete Race, such as Dwarf, and returns it with the GetRace() function. Certain factories take input parameters, to handle cases where the racial bonuses are user select-able. The character's (DND)class is handled with a an interface pattern. IClass is the parent class of every (DND) class below it. It is advantageous to use this method since every class rolls for attacks and spellcasts in different ways. These are not yet fully implemented and are subject to change before final submission.

## 5. Results

This project is coming along nicely, we can create an almost full character, as well as save and load them. Many goals have been met, but there are only a few concerns; A fancy GUI is looking lower on the list, even tho it was a minor stretch goal. Editing a character should be done, though it will be a right squeeze to have it implemented. The main time consumer is making sure that all the rules and exceptions are taken care of. With the numerous amounts of aforementioned exceptions, we will busy till the last day of production.

### 5.1. Future Work

We plan to continue updating it and adding features as a way to practice our c# skills and hopefully grow a community. We are not aware of any free offline tools that are open-source so we were inspired to give back to DnD by creating this project. Many challenges arose, primarily in the areas of saving and loading character data. Ultimately, we think the path we chose with regards to using a database and OOP will make this software easy to build upon. We would love for someone to fork this and make a Pathfinder version for example! We hope to keep the program portable so that save files can be backed up and used anywhere without the need for an internet connection. If even one member of the DnD community benefits from our work, then all was not in vain!

## References

- [1] H. Kopka and P. W. Daly, *A Guide to L<sup>A</sup>T<sub>E</sub>X*, 3rd ed. Harlow, England: Addison-Wesley, 1999.