**Project Specifications**

**DND DECKS**



**Submitted to**

Suncoast Technical College

.NET Application Development and Programming

Instructor: Mr. James Hornberger

**Submitted by**

Joshua Hernandez

**Introduction – Client Specifications and Final Product Description**

DnD-Decks (Dungeons and Dragons Decks), the final product of this project will have the capability to store and create spell cards, and a profiling system to store said cards in decks. Among these features, the spell creator will allow for the creation of spell cards, which is great for Dungeons and Dagon’s Homebrew Players; as they content creators who thrive on creating their own content for Dungeons and Dragons. The Spell-Cards are Json Data sets, which are later formatted by the program and turned into a readable Spell-Card that looks really nice. Among the features, there will also be a filtering system for the spell cards, so it becomes easier to find the cards, as there are 400 pre-installed spells.

Client’s specifications. The targeted clients are Dungeons and Dragons players, which expect from the project proposal that the program will contain a profiling system, calculations system, Homebrew system, Spell cards, and Item Cards; However, DnD-Decks will only currently provide, Homebrew, Profiling, and Spells cards. Calculation and Item cards will be later added on future updates of the software. The clients should, however, still find DnD-Decks useful, as it will provide the Spell-Cards system to help keep track of the player’s spells and add new ones as they go on using it. The Project will be releasing a Beta-Early Access version of the software instead of a full release so that suggestions can be made and bugs can be reported as more features are added towards release.

**Algorithms**

**Form Pre-Load**

Outputs: Pre-Load Form Displays

Inputs: System IO Spell Card Data

Constants: N/A

Formulas:

* Foreach Spell Data create a Spell Card
* Calculate how many files there is and how many files are loaded for each iteration in creating Spell card data, which displays Loading Percentage.

Pseudocode:

Step1: Start

Step2: Display Pre-Loader

Step3: Pre-Loader Load all content (400 JSONs) and make static lists and forms

Step4: Hide Pre-Loader and Show Form1

Step5: If exit button pushed then EnviromentCode(-1) Closes Application

Stop!

**Json Parsers**

Outputs: Json Variable

Inputs: System IO Json Data

Constants: N/A

Formulas: N/A

Pseudocode:

Step1: Start

Step2: read data

Step3: Convert Data

Step4: Display Data

Step5: Stop!

**Spell Form**

Outputs: Spell Cards

Inputs: Spell Class Instances List

Constants: The Form Itself

Formulas: Filter Formula for filtering Spell Cards, and String Combination Formula to combine strings.

Pseudocode:

Step1: Form is Pre-Loaded from Pre-Load

Step2: foreach spell in card list, add to Card Flow (FlowLayoutControl)

Step3: Each Card instantiate with Spell variables and movable to Profile Control

Step4: Stop!

**Save and loading**

Outputs: IO File Json saving

Inputs: Profile Data

Constants: conversion method

Formulas: Json serialization and deserialization

Pseudocode:

Step1: Get files from directory

Step2: load files Json to Class object

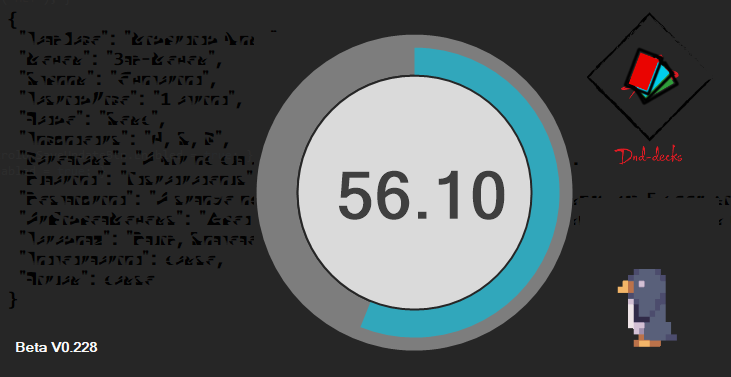
Step3: If user event

Step4: Save all data in directory Class to Json object

Step5: Close and Stop!

**Flowcharts**

**Screenshots**



Spell Form

Displays all Card Controls that contain Spell Data.

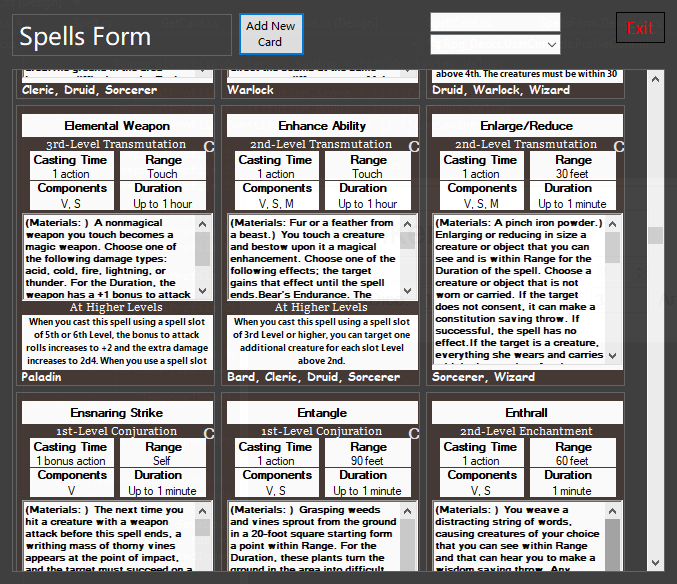
Pre-Loader

Loads before showing main form so all variables are ready.

Penguin is Beta Only.

~~Penguin sold separately!~~

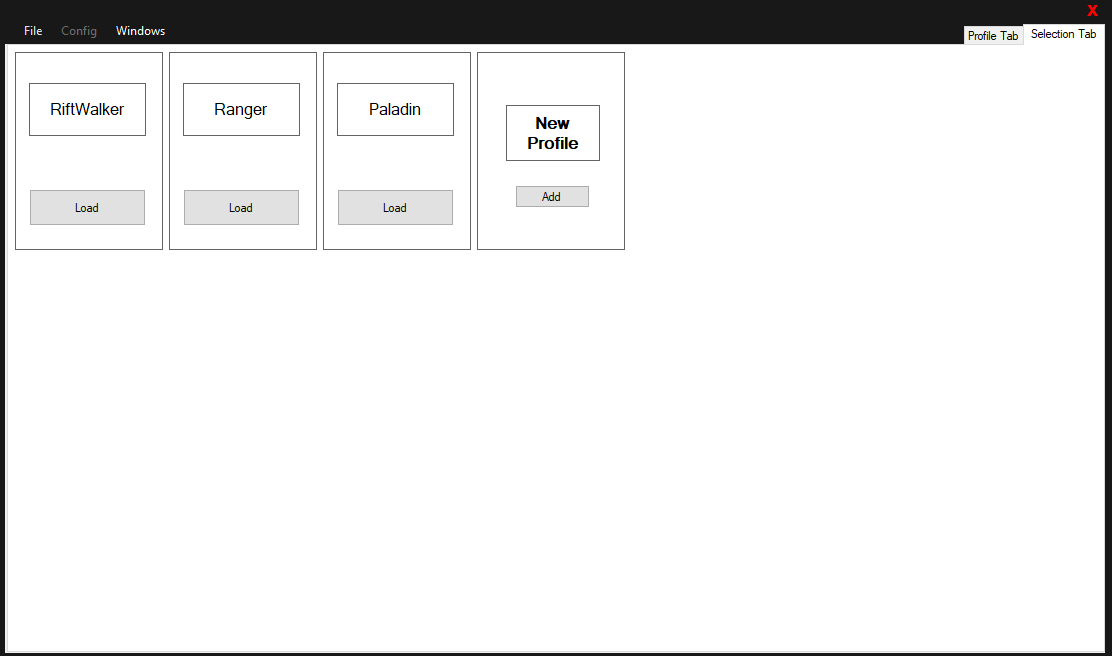
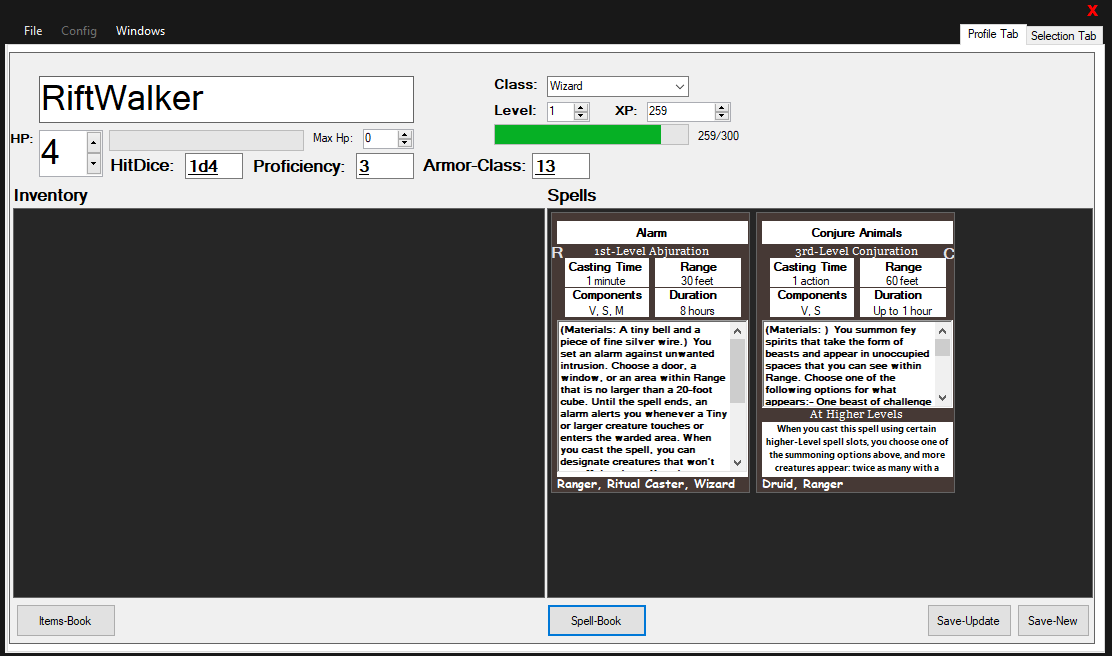
Main form had a tab so profile selection can be made within same window.



Main Form

Select profile and make a new one if you don’t have a profile.

Also Design has been taken note of and made nicer.



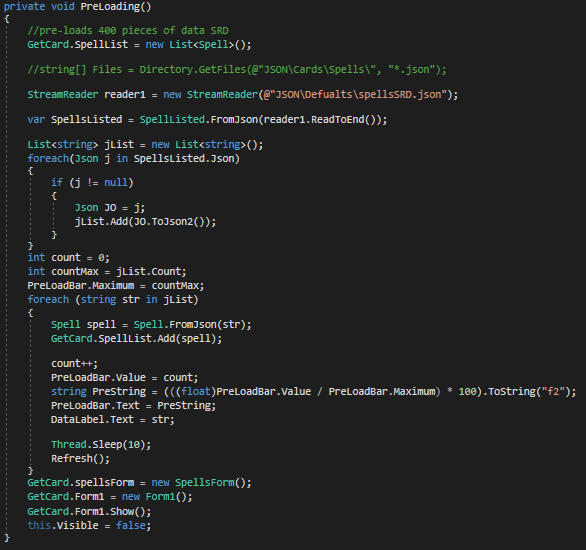
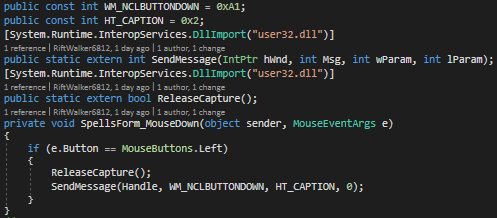
**Sample Code**



Special code allows for forms to be movable if they don’t have a form edge

Json Spell Data

Pre-Loader Code



**Conclusion and Future Development/Enchantment’s**

DND-Decks is my first Dot Net Project. I have tried to implement what I have learned throughout the year, into this project; as well as, research on things such as Json file saving and reading to use in my project. During the course of the project there were lots of errors and bugs that needed fixing and that forced me into days of research and trial and error until I got it. The project has taught me many new programming concepts that I plan to use in future developments, which will help make programs better and more efficient.

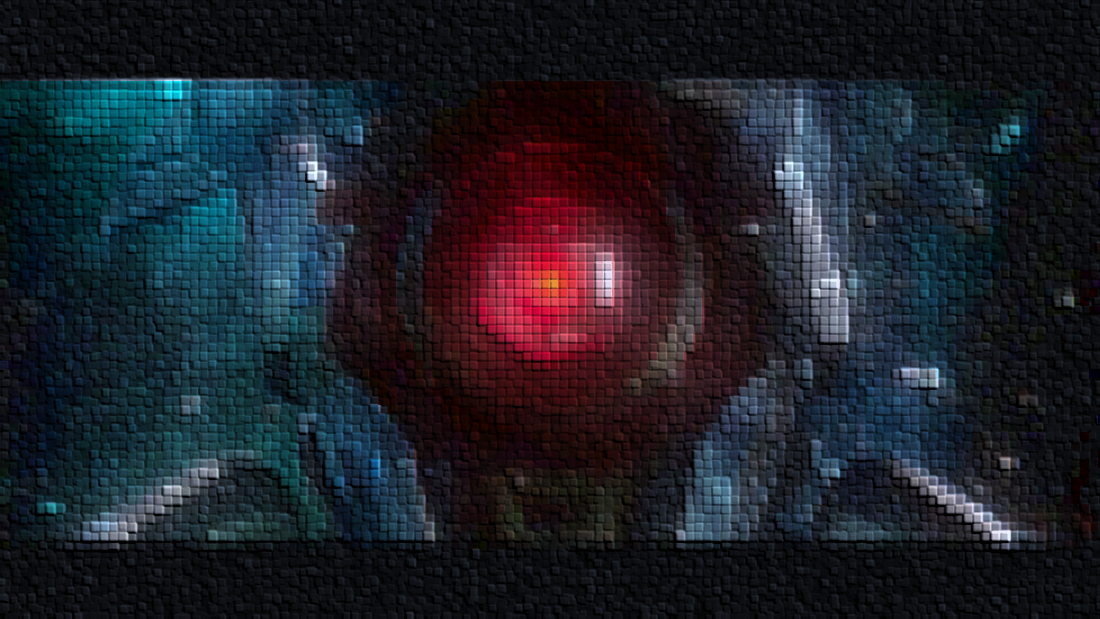
The project development has led me to want to add more features to it later on and bring more updates one the project is released. Once DnD-Decks is done I will add more enchantment’s and later bring version 2 of DnD-Decks, which will be more efficient, faster, better in design, and will be developed on the UNO platform for cross platform support. Future developments of DnD-Decks will also be impacted by user reviews so that the next release version is better for user preference.

**Credits and Acknowledgements**

Thanks to Mr. Hornberger for allowing me to do this project (DnD-Decks).

**References**

* <https://www.newtonsoft.com/json>
* <https://stackoverflow.com>
* <https://docs.microsoft.com/en-us/dotnet/csharp/>
* <https://www.youtube.com>

**Developer Profile**

Name: Joshua Angel Hernandez

Semester: Junior .NET

N-Number: 413517

Developer Email: [RiftWalker0@outlook.com](mailto:RiftWalker0@outlook.com)