[D&D-inator]

# Team Members

* Andrew Emlund
* Spencer Southam
* Gabe Hill
* Alice McClain

# Project Description

DnDinator is a DM utility for Fifth Edition D&D. The application handles the creation and management of characters. Has a built in Dice roller, Loot stat generator, and Combat damage roller. Launch options lead to a version of the app for a DM and a version for the Players.

# Part 1: Functional Requirments

*Below is an example set of functional requirements. Your team must fully define what your system will do. Language and terminology must be appropriate for laypeople, as this document is for the client.*

## Glossary

* DM – Dungeon Master, also reffered to as Game Master/Manager
* Player – Individual participating in the game, has less application functionality than the DM

## Priority

1. DM
   1. -Character [1]
      1. NPC
      2. PC
      3. Block vs sheet [2]
   2. Game options
      1. Player list [1]
         1. Add[1]
            1. Create [2]
            2. Pre-existing [2]
         2. Remove [1]
            1. From game [2]
            2. Death [2]
      2. Manage rules [1]
         1. Level rule [2]
         2. Add rule [2]
         3. Remove rule [2]
         4. Display rules [2]
      3. Save rules [1]
         1. Display path to file [2]
      4. Load rules [1]
         1. Display path to file [2]
      5. NPC [2]
         1. Create recurring [3]
            1. Villain [3]
            2. Master [3]
            3. Other [3]
   3. Player leveling [1]
      1. Add xp [1]
         1. Per player [2]
         2. Party [2]
      2. Add level [1]
         1. Per player [2]
         2. Party [2]
   4. Loot [3]
      1. Distribution [3]
         1. -Prompt for a given table then rolls for x number of loot items on the table
      2. Creation [3]
         1. -Allows the GM to create a new item to the list of items in the game
      3. Table [3]
         1. Create [3]
            1. -Allows gm to create a new loot table from currently created itemEdit [3]
      4. Edit [3]
         1. -add item(s)
            1. -Prompts for given table and shows the list of items that are not already present on current table
      5. Remove item(s)
         1. Prompts for a given table and dispays all the items on that table in numeric order then prompts for an item(s) to delete
      6. Delete Table[3]
         1. -Prompts for a table to delete
   5. Create notes [2]
      1. Save notes [2]
         1. Add to the saved GSON file through GSONIfier
      2. Load notes[2]
         1. Pull the saved content from the GSON file through DEGSONIfier
      3. Create Notes with type specification[2]
         1. Story[2]
            1. Create text file integrated into the overall GSON with Story note type specification
         2. Areas[2]
            1. Create text file integrated into the overall GSON with Area note type specification
         3. Misc. [2]
            1. Create text file integrated into the overall GSON with Custom note type specification
2. PLAYER
   1. Character [1]
      1. PC only
3. BOTH
   1. Roll Dice [1]
      1. Advantage[3]
      2. Disadvantage[3]
         1. Combat[1]
         2. Initiative[2]
         3. Grapple[3]
            1. Athletics check
            2. Acrobatics check
            3. Shove
         4. Hit [2]
            1. Dodge[2]
            2. Weapon[2]

Effects

Crit

* + - * 1. Spell[2]

Saving throws

* + - 1. Checks [2]
         1. Skills [2]
         2. Stats [2]
      2. Damage [2]
         1. Weapon [2]

Primary attack [2]

Damage

Saving throws

Secondary attack[3]

Damage

Saving throws

Bonus Damage [3]

Saving throws

Effects[3]

Crit [2]

Melee attack [2]

* + - * 1. Spell [2]

Saving throws

* + - * 1. Other magic [2]
        2. Environmental [3]
      1. Spells [2]
         1. Healing [2]
         2. Stat buffs [2]
      2. Other actions [3]
      3. Bonus actions [3]
      4. Reactions [3]
    1. General [2]
    2. Saving throws [2]
       1. Enemy [2]
          1. Stat check things
          2. Special NPC
          3. Death
       2. Ally [2]
       3. Self [2]
          1. Death
          2. Other checks
    3. Checks [2]
       1. Skill
       2. Stats
    4. Attribute [2]
       1. Physical
       2. Mental
  1. Create character[1]
     1. Manual entry [2]
     2. Auto-generated
     3. Edit character
        1. applies to creation
        2. CHARACTER NAME
        3. Class
        4. Level
        5. Player name
        6. Race
        7. BACKGROUND
        8. Hit Point Maximum
        9. CURRENT HIT POINTS
        10. TEMPORARY HIT POINTS
        11. Strength
        12. Dexterity
        13. Constitution
        14. Intelligence
        15. Wisdom
        16. Charisma
        17. CP
        18. SP
        19. GP
        20. EP
        21. PP
        22. PASSIVE WISDOM (PERCEPTION)
        23. OTHER PROFICIENCIES
        24. LANGUAGES
        25. EQUIPMENT
        26. ATTACKS
        27. SPELLCASTING
        28. FEATURES
        29. TRAITS
        30. Acrobatics (Dex)
        31. Animal Handling (Wis)
        32. Arcana (Int)
        33. Athletics (Str)
        34. Deception (Cha)
        35. History (Int)
        36. Insight (Wis)
        37. Intimidation (Cha)
        38. Investigation (Int)
        39. Medicine (Wis)
        40. Nature (Int)
        41. Perception (Wis)
        42. Performance (Cha)
        43. Persuasion (Cha)
        44. Religion (Int)
        45. Sleight of Hand (Dex)
        46. Stealth (Dex)
        47. Survival (Wis)
        48. HIT DICE
        49. SUCCESSES DEATH SAVES
        50. FAILURES DEATH SAVES
        51. IDEALS
        52. BONDS
        53. FLAWS
        54. PERSONALITY TRAITS
        55. ARMOR CLASS
        56. PROFICIENCY BONUS
        57. INITIATIVE
        58. SPEED
        59. STRENGTH SAVING THROWS
        60. DEXTERITY SAVING THROWS
        61. CONSTITUTION SAVING THROWS
        62. INTELLIGENCE SAVING THROWS
        63. WISDOM SAVING THROWS
        64. CHARISMA SAVING THROWS
        65. INSPIRATION
  2. -Save character [1]
  3. -Load character [1]

# Part 2: Objective Coverage

* Data persistence via writing files to disk and reading them back
* JavaFX for the application's GUI
* Proper encapsulation and separation of components into their respective MVC categories
* HashMaps will be used alongside enums to properly access and store class and race data
* Proper teamwork, communication, and load distribution will be used to ensure each group member pulls their own weight
* UML will be used to properly structure the application, and provide reference to that structure
* User input will be gathered and checked
* Output will be formatted in an intelligent way
* Overrides will be used for all applicable methods
* Proper optimizations, such as StringBuilder, shall be used where applicable

# Part 3: Basic Design

# 

# Part 4: Project Plan

* Phases Base Level
* Phase one: Continued planning and distributing of jobs for the program
* Phase dos: pretesting and editing
* Phase san: polishing and finalizing
* Phase vier: demonstrating and final execution!