Benjamin Smith, David Olinger

Dr. Dickinson

COMP2020

4/19/23

A Summary of Our Plan for Project 5

An Automated Character Sheet Tool for D&D 5e

- Our project is a digital version of a character sheet for the game Dungeons and Dragons (Fifth Edition).

What does it do?
- The Sheet will track and score everything numerical that a player needs on their character sheet and be able to quickly calculate the outcomes of various rolls using the stats from the sheet (hit points, armor class, skill checks, saving throws, attacks, etc.)
- The project involves several classes, but the most important is the PlayerCharacter class, which stores all the information about the character, as well as an inventory system (a numbered list of InventoryItem objects, which include weapons the player can attack with)
- This inventory system will give the player the ability to add and remove weapons or other items from the list. A player could attack with a specific weapon by including the number of the slot containing the weapon they want to attack with in their prompt.
- The PlayerCharacter class also has the ability to track Spell slots using a HashMap, by mapping slots remaining or the maximum slots to the level of each slot.
- Attacking with a weapon will display the result of the attack roll and the amount of damage the attack causes based on the weapons stats (number of dice to roll and bonus to add)
- Our program will also be able to read and write a "save" of a PlayerCharacter to a text file, in a way that makes it easy for the user to edit (though the user can change many things from the program itself.

What does it not do?
- Our program does not track game specific abilities or mechanics(such as barbarian rage) nor the effects of spells (such as green-flame blade). That is for the player to handle, this program mainly handles the tedious or numerical tasks involved with playing D&D.
- Our program does not simulate battle or interactions between two PlayerCharacters.

What could our program do?
Some ideas we had on how to expand this program once we get the basic character sheet working.

- We would really like to add the ability to generate a random character sheet, starting with random stats, then perhaps a name selected from a list of random syllables (possibly different lists depending on race).
- Making a few files with many different InventoryItem names could allow a randomly generated inventory too, and a separate list of weapons with their stats would let this character have a random weapon too.
- If we really have time, we would like to be able to generate a pseudorandom short backstory for each character too, probably not much more complicated than a game of mad libs.
- We likely won't get around to making a GUI. Although that would be very desireable, it would be a huge undertaking and is likely beyond the scope of this project.

Spell

-String Name
-int damage
-int slotLevel

+Constructor(name, damage, slotLevel)

+Getters and setters

Main

main()

PlayerCharacter

-int Private Arraylist<InventoryItem> Inventory
-int deathSave (Length 2 Int Array where i=1 is successes and i=2 is failures)
-int armorClass
-int maxHealth
-HashMap abilities //ability scores
-int speed
-Hashmap<int, int> maxSpellSlots //level, Max
-Hashmap<int, int> remainingSpellSlots //level, remaining
-int Gold

+Default Constructor (Default stats)
+Constructor from (String containing name, stats)
+int getMod(ability name: String)
+ int [skill/saving throw checks]()
+changeHealth
+void failDS()[death save]
+void succDS()
+void clearDS()
+int attack(Weapon)[returns damage]
+void addItem(InventoryItem)
+void removeItem(index of item to delete)
+int rollAbilityCheck(name of ability: name)
+int rollDie(Number, Sides, Bonus)
+void castSpell(Spell)
+int getGold()
+void changeGold(value)
+String stringSheet()
+void longRest()
+void shortRest(numHitDice)

Inventory Item

-String name
-int quantity

+Constructor(name, quantity)
+Default and copy constructors
+Getters and setters

parent

child

Weapon

-int weaponBonus
-int damageDie
-int numDamageDice
+String weaponType
+String damageType

+Constructor(numDice, damageDie, weaponType, damageType, bonus)
+int toHit()
+int getDamage()