DATA440 Final Project Write-Up

Introduction

This project will make use of the Python libraries PyQt5 and matplotlib to create a Graphical User Interface (GUI) with the functionality of a dice roller, a player sheet, and a graph creator. The idea for the GUI with a dice roller came from my interest in the tabletop roleplaying game Dungeons and Dragons (DND).

I found the idea of creating a dice roller that automatically calculated the results of a roll based on a character's modifier useful and thought it would be a good basis for a project. The additional features of the GUI came from suggestions made to me by peers and a professor that I felt were great ideas.

The dice available to roll include a four-sided die, a six-sided die, an eight-sided die, a ten-sided die, a twelve-sided die, and a twenty-sided die. These die types were specifically chosen because Dungeons and Dragons players use them most frequently for their rolls. I left out the one hundred-sided die, as I felt that while sometimes it is used in DND, it is primarily used by the person running the game. Additionally, one hundred-sided dice are used to generate a random outcome and are not meant for skill checks or attack rolls by characters.

The player sheet allows the user to import a picture of their player and allows users to fill out characteristics like name, class, race, alignment, level, and age in text boxes. There are also places to input the player inventory, the character's features and traits, as well as other proficiencies and languages.

The plot creator allows the user to test different scenarios related to dice rolls to compare the distribution of variances. This is particularly useful when a player wants to compare two weapons that use different dice and modifiers for their rolls. While one die may have a higher maximum damage threshold, the probability of rolling well could possibly be with the second weapon.

Background

A general background for Dungeons and Dragons is that it is a fantasy game where players can create their own characters and roleplay as them going on adventures. The characters can either be based on the set of races or classes provided by the base game or home-brewed (which just means that it was personally created by the player or dungeon master). The dungeon master refers to the individual who runs the games for players.

Depending on a character's race, class, and/or level, the ability scores may vary. Ability Scores (which are Charisma, Constitution, Strength, Wisdom, Dexterity, and

Intelligence) allow the player to add or subtract a value when they roll a die (usually a D20). This value is called a modifier.

When you roll a die for in-game situations, it can be for things like Perception Checks (where your character rolls to see if they may notice something), Deception Throws (to see how well or badly your character can deceive someone or something— which falls under the ability of Charisma), or even a Constitution Saving Throw (to maybe see if your character avoids taking full damage from a certain condition like acid or psychic damage in battle).

Also based on your character's ability scores are Hit Points (which is how much health a character has), Initiative (when a character can enter a battle), and even Armor Class (which is a number rolled against when enemies are trying to land an attack).

Because the ability scores are so intertwined with rolls and have specific modifiers (that in turn can have additional numbers added on to them if you have a proficiency in that skill— which means your character is particularly good at something), rolling and adding up the numbers can sometimes be a hassle or pause the game.

Related to the code itself, a graphical user interface allows a user to interact with the computer through elements like icons, buttons, and menus. PyQt5 is the primary library used in this project, as it is the library that Python has specifically to create GUIs and graphical components within those interfaces.

The library matplotlib in Python is one meant for visualizing data. The visualizations can be static, animated, or even interactive. For the purposes of this interface, the graphs created are static and have no user interaction on the plot itself. The user instead chooses the dice and modifiers that will be used by the distribution of variances calculations and graphs.

Data

The graph creator in the GUI creates bar graphs that plot the distribution of variances. This plot can look at the distribution of one scenario or the distribution of two different scenarios for comparison.

For DND, certain weapons require certain dice (like a six-sided or eight-sided die) and could have their own corresponding modifiers. The player's character may also have a modifier that is related to adding more damage when a weapon is used (like if they have a high Strength attribute). So looking into the variance of an example for two six-sided dice with a modifier of four in comparison to two eight-sided dice without a modifier can tell the user which weapon might be a better choice for maximum damage for an attack. The question arises if it would be better to try the higher modifier with the lower dice, or the higher dice with no modifier. The graph allows users to test random dice and modifiers against one another to see what the variance results would be to aid them in their choices.

Because the rolling of dice is not reliant on what a previous roll's outcome is, they are considered independent. Additionally, the rolls are also considered random variables[1] because they take on different values determined by chance (ex: for a six-sided die, the set of possible results would be {1, 2, 3, 4, 5, 6}, and when rolled, have a ⅙ chance of rolling any of the numbers within that set)[2].

In statistics, the mean (or expected value) of a random variable X can be shown by the formula $E[X] = \mu$. The sum of independent random variables is the sum of their individual means. For example, if you have two independent random variables, X and Y, the mean of their $sum(X + Y)$ is equal to $E[X] + E[Y]$. Here E[X] refers to the expected value of X, while E[Y] refers to the expected value of Y[3]. The expectation is always positive. So if you have a scenario where you are combining the roles of a D6 and a D8, then the expected value of X and Y would be the expected value of the D6 + D8.

Variance is the measure of dispersion, telling us how "spread out" a distribution is. The formula for variance can be represented as:

$$Var(X) = E[(X - \mu)^2] = E[X^2] - \mu^2$$

The equation of the variance of a random variable plus a constant is: $Var(X + k) = Var(X) + Var(k) = Var(X) + 0$. The variance of a constant is zero; therefore when taking the variance of a dice roll like a D6 with a modifier of 4, you would not consider the modifier in the calculation.

An important point to note about the calculations used to create the graph is the fact that the distribution of a sum of independent random variables is the convolution of their distributions. The distribution of the sum of random variables refers to how the probabilities are distributed across the possible values of the sum of those random variables. This tells us the likelihood of getting different values when we add the results of multiple random events.

For example, if you have two D6s and add their values, the distribution of the sum would show the probability of getting a total of 2, 3, 4, 5 and so on up to 12. This is because the minimum value that can be rolled with the two dice is 2, while the maximum value is 12 (if you rolled two ones, that would equal two, while if you rolled two sixes, that would equal twelve). These probabilities are the convolution of their distribution. A convolution refers to the mathematical operation on two functions, f and g, that produces a third function, f*g.[4]

---

[1] https://en.wikipedia.org/wiki/Random_variable
[2] https://online.stat.psu.edu/stat500/lesson/3/3.1
[3] https://online.stat.psu.edu/stat504/lesson/variance
[4] https://en.wikipedia.org/wiki/Convolution

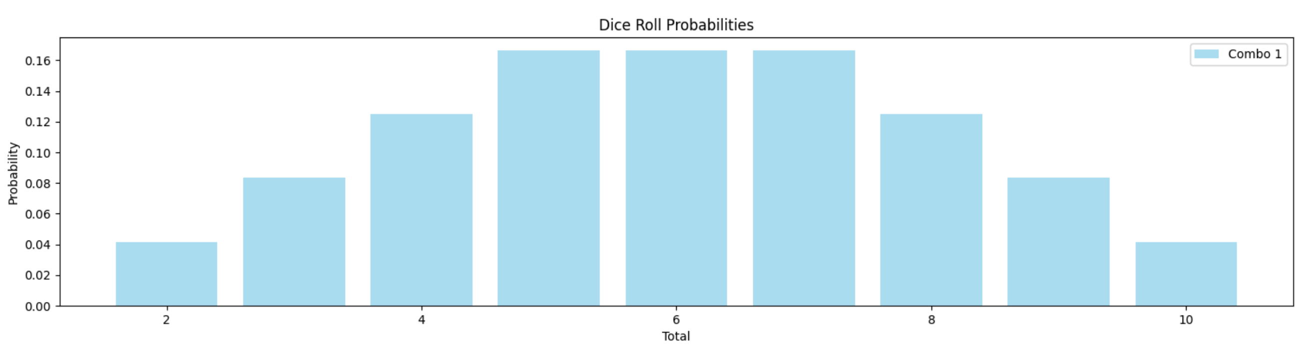
The calculations for the variance with the modifiers would simply be the variance of the independent random variable, but the mean calculations would have to take into account the modifier for the expected value. The equation for the expected value of a random variable plus a constant is: $E[X + k] = [X] + E[k] = \mu + k$. Here you can see that the mean is affected by the constant. For example, if you were to roll a D10 with a modifier of 2, then you would have to add 2 to the expected value (since k=2).

Something else that is interesting is the fact that the covariance of the two independent random variables is equal to 0. This is because the two random events are not connected and have no bearing on the result of the other. Therefore, if you were to try to use the equation for adding two variances together (i.e., $Var(X + Y) = Var(X) + Var(Y) + 2Cov(X, Y)$) the equation would look like: $Var(X + Y) = Var(X) + Var(Y) + 0$.

There is also math behind rolling with advantage. The idea of rolling with advantage in DND refers to being able to roll twice and taking the higher roll. This would look like $max(X, x)$ where X refers to the first random event (or the first roll) and x refers to the second random event (i.e. the second roll). Taking the result of both rolls, you would then be able to choose the result that had the higher value. When rolling with the modifiers, it would look like $max(X + k, x + k)$, where k is the modifier that can be added to the roll. You can also simplify it to be $max(X, x) + k$ since the higher roll will remain higher regardless, since the modifier would be added to both individually.

There is a similar math when rolling with disadvantage in DND. Rolling with disadvantage means that a player rolls two dice and must pick the lower of the two rolls. This would look like $min(X, x)$ where X, like with advantage, refers to the first roll, and x refers to the second roll. The addition of the constant would look similar as well: $min(X + k, x + k) = min(X, x) + k$.

Below is an example done by hand (and a graph created by the code is also featured in the ‘figures’ folder under “plot_2025-05-12-00-32-42-578318.png”):



- Dice rolls are random variables
- They are independent

☆ The distribution of a sum of independent random variables is the convolution of their distributions.

Ex: $E[X] = \sum p_i * x_i$

D4

$F(x) = 1/4$

D6

$F(x) = 1/6$

1/6 1/6 1/6 1/6 1/6 1/6

1 2 3 4 5 6

4 3 2 1

1/4 1/4 1/4 1/4

$1/6 \cdot 1/4 = 1/24 = h(2)$

1/24 1/24

1/6 1/6 1/6 1/6 1/6 1/6

1 2 3 4 5 6

4 3 2 1

1/4 1/4 1/4 1/4

$2/24 = 1/12 = h(3)$

1/6 1/6 1/6 1/6 1/6 1/6

1 2 3 4 5 6

4 3 2 1

1/4 1/4 1/4 1/4

$3/24 = h(4)$

1/6 1/6 1/6 1/6 1/6 1/6

1 2 3 4 5 6

4 3 2 1

1/4 1/4 1/4 1/4

$4/24 = 2/12 = 1/6 = h(5)$

1/6 1/6 1/6 1/6 1/6 1/6

1 2 3 4 5 6

4 3 2 1

1/4 1/4 1/4 1/4

$4/24 = 1/6 = h(6)$

1/6 1/6 1/6 1/6 1/6 1/6

1 2 3 4 5 6

4 3 2 1

1/4 1/4 1/4 1/4

$4/24 = 1/6 = h(7)$

1/6 1/6 1/6 1/6 1/6 1/6

1 2 3 4 5 6

4 3 2 1

1/4 1/4 1/4 1/4

$3/24 = h(8)$

1/6 1/6 1/6 1/6 1/6 1/6

1 2 3 4 5 6

4 3 2 1

1/4 1/4 1/4 1/4

$2/24 = 1/12 = h(9)$

1/6 1/6 1/6 1/6 1/6 1/6

1 2 3 4 5 6

4 3 2 1

1/4 1/4 1/4 1/4

$1/24 = h(10)$

$h(z) =$

1/24	2
1/12	3
3/24	4
1/6	5
1/6	6
1/6	7
3/24	8
1/12	9
1/24	10

$E[Z] =$

$1/24 * 2 + 1/12 * 3 + 3/24 * 4 + 1/6 * 5 + 1/6 * 6 + 1/6 * 7 + 3/24 * 8 + 1/12 * 9 + 1/24 * 10$

$= 6$

Scenario 1: [D6, D4]
- 1 roll of D6
- 1 roll of D4

☆ The distribution of a sum of independent random variables is the convolution of their distributions.

$E[Z] = 1/24 * 2 + 1/12 * 3 + 3/24 * 4 + 1/6 * 5 + 1/6 * 6 + 1/6 * 7 + 3/24 * 8 + 1/12 * 9 + 1/24 * 10$

μ:

$E[X] = \mu$

μ * constant

$E[kX] = kE[X]$

Var * constant:

$Var(kX) = k^2 Var(X)$

μ + constant

$E[X + k] = \mu + k$

$E[\mu]$ $E[k]$

Var + constant

$Var(X + k) = Var(X) + 0$

variance:

$Var(X) = E[X^2] - \mu^2$

convolution can be used as μ

$E[Z] = 6 \quad \therefore \quad E[X] = 6$

$Var(X) = E[X^2] - \mu^2$;

$Var(X) = E[X^2] - 6^2$;

prob. of getting results	possible results
1/24	2
1/12	3
3/24	4
1/6	5
1/6	6
1/6	7
3/24	8
1/12	9
1/24	10

mean:

$\mu = 1/24 * 2 + 1/12 * 3 + 3/24 * 4 + 1/6 * 5 + 1/6 * 6 + 1/6 * 7 + 3/24 * 8 + 1/12 * 9 + 1/24 * 10$

$= 6$

$E[X^2] = \sum X^2 \cdot P(X = x)$

$= (2^2 * \frac{1}{24}) + (3^2 * \frac{1}{12}) + (4^2 * \frac{3}{24}) + (5^2 * \frac{1}{6}) + (6^2 * \frac{1}{6}) + (7^2 * \frac{1}{6}) + (8^2 * \frac{3}{24}) + (9^2 * \frac{1}{12}) + (10^2 * \frac{1}{24})$

$0.1666666667 + 0.75 + 2 + 4.1666666667 + 6 + 8.1666666667 + 8 + 6.75 + 4.1666666667$

$= 40.1666666667$

$Var(X) = E[X^2] - \mu^2$

$\mu^2 = 6^2 = 36$

$E[X^2] = 40.1666666667$

$40.1666666667 - 36$

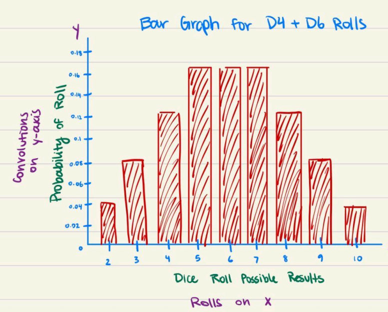
$= 4.1666666667$

$1/24 = \sim 0.0416$

$1/12 = \sim 0.08\overline{333}$

$3/24 = 0.125$

$1/6 = \sim 0.1666\ldots 7$



Conclusions and Future Work

All in all, I feel that this Graphical User Interface accomplished the goals of having a dice roller capability that takes into account modifiers, having an easily accessible character sheet for users, and having a grapher that displays the distribution of variances for different dice roll scenarios generated by the user.

The future course of action for this project would be to make the D20 dice roll twice when the inspiration button is clicked. Another path that this project can take would be to either integrate the PyGame or Panda3D library as the dice roller to show a 3-dimensional die with a velocity for the rolls. Having a rolling dice instead of the GIFs and the updated label would elevate the dice roller as a whole.

The change of the dice roller would require the integration of two different libraries, which could result in some errors, so it would be a more long-term goal for later.

Another possibility for future work could be with the graphs. Right now, they are static, but there are many features in matplotlib that can make a graph animated or interactive, and having a GUI with the capability to interact with the graph may be interesting to see.

As something simple to add, it might be nice to have a QLabel appear when the user rolls a 1 or 20 with a result that either says “Critical Failure!” (for the 1) or “Critical Success!” (for the 20) so that the player knows that they have automatically failed or succeeded in a roll. This is because in DND when a player rolls a D20 and gets a result of 1 for a check, they automatically fail the encounter and face consequences, whereas if they roll a 20, they automatically succeed and get benefits.

Finally, I would also add a way to save the information in the player sheet so that the player doesn’t need to re-input the information each time the GUI is launched (and it may also be useful to be able to upload a character sheet using the format that it can be saved in for easy access and editing by the player).