Shawn McClelland

**Project Summary:**

**Project Title:** Shawn’s DnD character Creator and Dice Roller

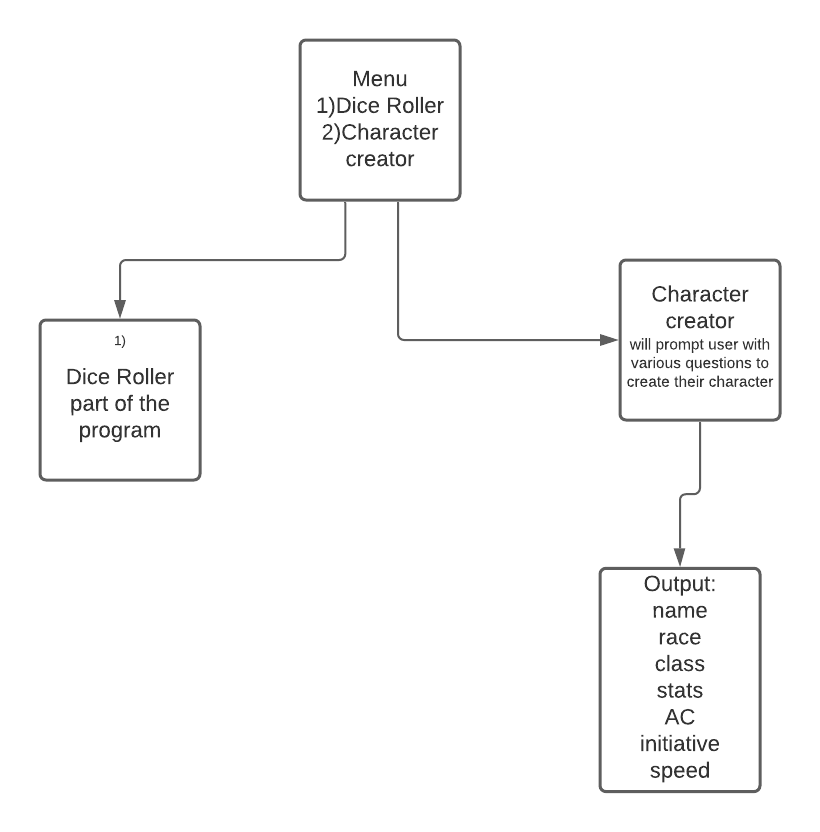
**Description of Project:** This program will assist new DnD players in creating their first character. This will create a level 1 character. The character creator will only contain information contained in basic DnD 5e. It will only support those basic classes/races. The program will also ignore the creative aspects of the character such as alignment, traits,flaws. Spells and skills will also be omitted because those are creative and also at DM discretion. The program will take in your name, race, and class with the optional option of inputting your own stat values. The program will output name,race,class, stats with all race/class bonuses taken into account, your speed, your initiative, and your armor class. These are all of the major aspects of a character that can be intimidating for a new player to accomplish on their own for the first time. The dice roller aspect of this program will let the user choose between a d4,d6,d8,d10,d12, or d20. It will output a random value in the given range per dice type.

**Intended user:** The intended user for this program is a user that is new to DnD. This program takes care of all of the difficult aspects of creating a new character. The dice roller aspect of this program is intended for anyone playing DnD that forgot their dice.

**Problem Being Solved:** This program will assist in lowering the barrier of entry for the game Dungeons and Dragons. Instead of having to research all of the rules on rolling stats, then calculation speed,initiative, armor class, and roll bonuses, the player can instantly start thinking about the creative aspects of their character, such as what spells they will use and their character’s backstory.

**What technologies:** I will be user Tkinter and I will set up the gui aspects in classes. One of the frame objects created by a class will obtain all of the info and then class methods will be used to manipulate the data and then output it back to the user on another gui screen.

**Use Case Analysis:**



**Data Design**

**What data is your program really about?**  The main data my program is about is the race,class, and stats. Which is mainly integers, but the race and class are strings that describe some bigger construct

**What is the best way to represent that data(database,object,arrays):**

An object that keeps it all in one place

**Will the data need to be persistent? How will you make that happen?**

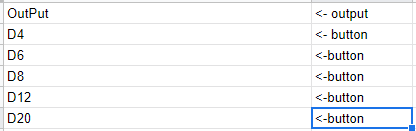
No, it is generated then manipulated and then quickly sent over to the result screen

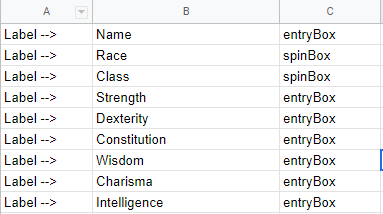
**Will the data need to be aggregated into a larger structure? How**

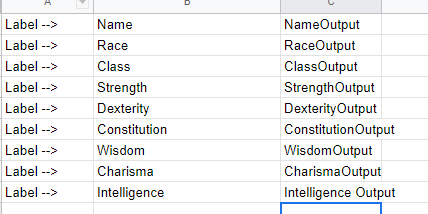
Does Not need to be integrated into something larger, because it is a small scale program meant to just generate the numerical side of a character sheet

**UI Design:**

**Menu:**

**Dice Roller:**

**Character Creator:**has an info box to the left of the labels to tell the users how to roll their stats manually if they wish to do so and has the submit button below,

**OutPut:**(info box below all of the out

**Algorithm**

Goal: make a functional dice roller that can roll multiple types of dice, and to make a character creator that takes in user inputted data and manipulates some of it to output the numerical side of a DnD character.

**Class App(Tk):**

The class App will start up a tkinter gui. It initializes with the term self.This is to take in the information from TKinter and make a functioning gui window.

Set the geometry to 500 x 500 and display two buttons, one that calls the switchRoll method and one that calls the switchCreator method.

**switchRoll method:**

The switchRoll method is put in place to change the window, it removes the other windows from the view and displays the frmDice frame, which is its own class

**switchCreator method:**

The switchCreator method does the same thing as the switchRoll method, except it switches the frame to the frmCharacter frame, which is also its own class

**Class frmDice(Frame):**

The frmDice class is a frame object. It is intilized with self and parent. This is to inherit the attributes of a Frame object while also taking in the same information from the App class.

The frmDice frame will create and then display all of the buttons(d4,d6,d8,d12,d20). Each button will call to its own method to roll the dice and output a value

**Method rollD4-D20**: These methods will roll a value between the range on each of the dice and return that value generated.

**Class frmCharacter(frame)**

The frmCharacter is the location where all of the data is acquired from the user,manipulated, then outputted to the user on a new window. This is definitely the largest part of the program.

The farm Character class is a frame object. It is initialized with self,parent just like frmDice

Frm Character will create and grid a label that says “Character creator”

frmCharacter will then create entry boxes that will take in the user’s data.

frmCharacter will also create two Spinbox menus to assist in the data collection of race and class.

All of these entry and spin boxes will be grided onto the screen, along with corresponding labels next to the boxes so the user knows what they are supposed to be inputting in each data field

Create and grid a button with the label submit on it at the bottom of the screen, this will link to the command/method called submitWindow

**Method submitWindow:**

Submit window will be there to open up the new window, and also call all of the other methods needed to manipulate the data and prepare it for being displayed

Set name,characterClass,race,strength,dexterity,constitution,wisdom,charisma,intelligence equal to get(variable). All of these get methods will be defined in the class and will set all the variables equal to the form submission.

Send the strength,dexterity,constitution,wisdom,charisma,and intelligence through their respective checkStat(stat) methods.

After that set strRoll,DexRoll,conRoll,wisRoll,charismaRoll,and intRoll through all of their checkStatRoll(stat) methods. These methods are to get all of the roll bonuses

Set all of the stats to strings afterwards to make outputting easier.

Set speed = to the checkspeed method.

Initiative = dexRoll

armorClass = dexRoll +10

After this it will output all of the data with proper labels to go along with it onto the new window created by submitWindow

**Method GetName:**

The getName method will use self.entryboxname.get to take the data from the entrybox labeled name and return it

**Method getCharClass**

The getCharClass method will use self.spinboxname.get to take data from the spinbox labeled class and return it

**Method getRace**

The getRace method will use self.spinboxname.get to take data from the spinbox labeled race and return it

**Method getStr**

getStr will take data from the entrybox labeled strength and return it

**Method getDex**

The getDex method will take data from the entrybox labeled dexterity and return it

**Method getCon**

The getCon method will take data from the entrybox labeled dexterity and return it

**Method getWis**

The getWis method will take data from the entrybox labeled wisdom and return it

**Method getCharisma**

The getCharisma method will take data from the entrybox labeled charisma and return it

**Method getInt**

The getInt method will take data from the entrybox labeled intelligence and return it

**Method checkStr**

The checkStr method will take in the strength and race variables

If the user did not input anything for the strength stat, it will simulate rolling 4 d6 and then adding the highest 3 values to give the user a base strength stat. It will then check to see if the user has selected the races Dragonborn(+2),Half-Orc(+2),or Human(+1). Their strength stat will increase if one of those was chosen

Return strength

**Method checkDex**

The checkDex method will take in the dexterity and race variables

If the user did not input anything for the dexterity stat, it will simulate rolling 4 d6 and then adding the highest 3 values to give the user a base dexterity stat. It will then check to see if the user has selected the races Elf(+2),Halfling(+2), or human (+1). Their dexterity stat will increase if one of those was chosen

Return dexterity

**Method checkCon**

The checkCon method will take in the constitution and race variables

If the user did not input anything for the constitution stat, it will simulate rolling 4 d6 and then adding the highest 3 values to give the user a base constitution stat . It will then check to see if the user has selected the races Dwarf(+2), Half-Orc(+1), or Human(+1). Their constitution stat will increase if one of those was chosen

Return constitution

**Method checkWis**

The checkWis method will take in the wisdom and race variables

If the user did not input anything for the wisdom stat, it will simulate rolling 4 d6 and then adding the highest 3 values to give the user a base wisdom stat. It will then check to see if the user has selected the race Human(+1). Their Wisdom stat will increase if one of those was chosen

Return wisdom

**Method checkCharisma**

The checkCharisma method will take in the charisma and race variables

If the user did not input anything for the charisma stat, it will simulate rolling 4 d6 and then adding the highest 3 values to give the user a base charisma stat. It will then check to see if the user has selected the race Dragonborn(+1), Human(+1), or Tiefling(+2). Their charisma stat will increase if one of those was chosen

**Method checkInt**

The checkInt method will take in the intelligence and race variables

If the user did not input anything for the intelligence stat, it will simulate rolling 4 d6 and then adding the highest 3 values to give the user a base intelligence stat. It will then check to see if the user has selected the race Gnome(+2) Human(+1) or Tiefling(+1). Their intelligence stat will increase if one of those was chosen

**Method checkStrRoll**

The checkStrRoll method will take in the variable strength.

It will set strength to an int to ensure it is ready to be mathematically manipulated

Create a variable called strRoll and set it equal to strength - 10.

Divide strRoll by 2

Return strRoll.

**Method checkDexRoll**

The checkDexRoll method will take in the variable dexterity.

It will set dexterity to an int to ensure it is ready to be mathematically manipulated

Create a variable called dexRoll and set it equal to dexterity- 10.

Divide dexRoll by 2

Return dexRoll

**Method checkConRoll**

The checkConRoll method will take in the variable constitution.

It will set constitution to an int to ensure it is ready to be mathematically manipulated

Create a variable called conRoll and set it equal to constitution-10.

Divide conRoll by 2

Return conRoll

**Method checkWisRoll**

The checkWisRoll method will take in the variable wisdom.

It will set wisdom to an int to ensure it is ready to be mathematically manipulated

Create a variable called wisRoll and set it equal to wisdom - 10.

Divide wisRoll by 2

Return wisRoll

**Method checkCharismaRoll**

The checkCharismaRoll method will take in the variable charisma.

It will set charisma to an int to ensure it is ready to be mathematically manipulated

Create a variable called charismaRoll and set it equal to charisma - 10.

Divide charismaRoll by 2

Return charismaRoll

**Method checkIntRoll**

The checkIntRoll method will take in the variable intelligence.

It will set intelligence to an int to ensure it is ready to be mathematically manipulated

Create a variable called intRoll and set it equal to intelligence- 10.

Divide intRoll by 2

Return intRol

**Method checkSpeed**

The checkSpeed method will take in the variable race

If the race is dwarf, gnome, or halfling set the variable speed to 25

Else

Set speed to 30

Return speed