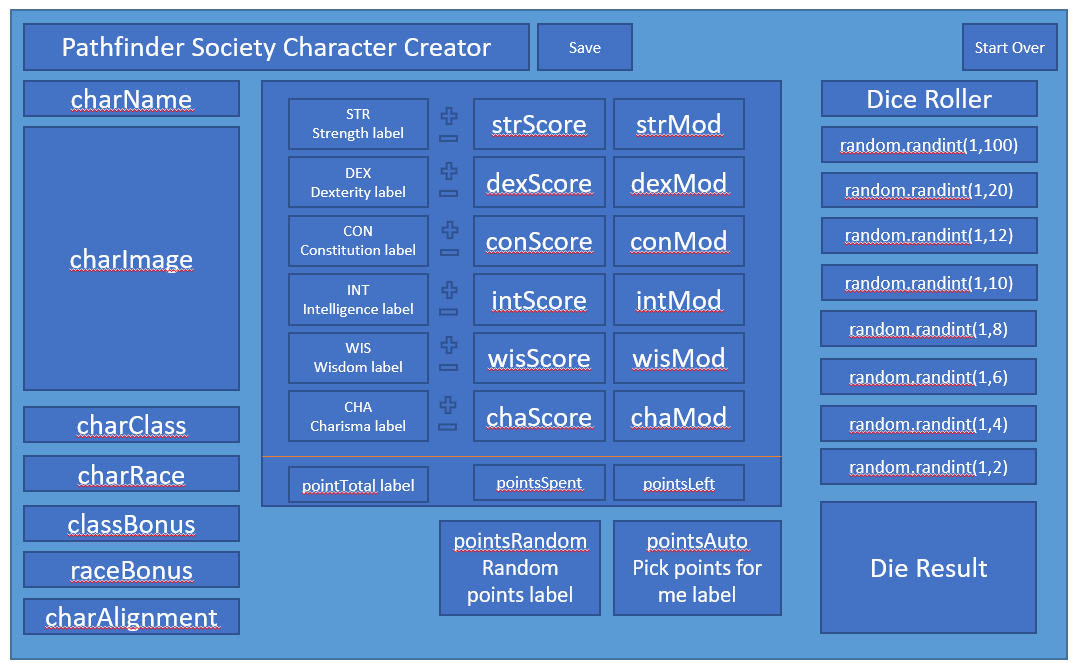
## PathPathfinder Society Character Creator – Original Project Submission

This program helps a user create a new character for the Pathfinder Society (*PFS*) role-playing game system. It will begin with selecting a name, class, race, alignment, then calculate a Point Buy system. This will also feature a randomized die roll for a D100, D20, D12, D10, D8, D6, D4, and D2 and a result display.

I thought of creating this application because, due to the pandemic, my social time playing games with friends has shifted online and virtual. This project is to help me understand the mechanics of a game I enjoy, and to find ways where a program can automate calculations and help guide creating a character.

Goals for the application are to successfully implement a button based point-buy system for stats including a choose for me button based on user selected class, buttons for randomized die rolls, selection of a character class, race, and alignment, and display of a character image. Character Class and Race will open new windows for a button based menu selection. A button to start over and reset variables to initial values should be included, as well as a Save button to allow output to a file.

My target audience would be Pathfinder Society players wanting to tinker with new character builds, explore stats, and to have a randomized die rolling at the click of a button.

GUI Outline:  


## Project Documentation Pseudocode

Enter Character Name  
 charName = str(input(“Please enter a name for your character: “))

Select Character Class  
 Barbarian  
 Bard  
 Cleric  
 Druid  
 Fighter  
 Monk  
 Paladin  
 Ranger (Archer)  
 Ranger (Melee)  
 Rogue  
 Sorcerer  
 Wizard

Character Image  
 If Class = Barbarian = charImage = barbarian.jpg  
 If Class = Bard = charImage = bard.jpg  
 If Class = Cleric = charImage = cleric.jpg  
 If Class = Druid = charImage = druid.jpg  
 If Class = Fighter = charImage = fighter.jpg  
 If Class = Monk = charImage = monk.jpg  
 If Class = Paladin = charImage = paladin.jpg  
 If Class = Ranger (Archer) = charImage = rangerarcher.jpg  
 If Class = Ranger (Melee) = charImage = rangermelee.jpg  
 If Class = Rogue = charImage = rogue.jpg  
 If Class = Sorcerer = charImage = sorcerer.jpg  
 If Class = Wizard = charImage = wizard.jpg

Select Character Race  
 Dwarf Bonus to Con, Wis Minus to Cha  
 Elf Bonus to Dex, Int Minus to Con  
 Gnome Bonus to Con, Cha Minus to Str  
 Half Elf +2 Bonus to one  
 Half Orc Bonus to Dex, Cha Minus to Str  
 Halfling +2 Bonus to one  
 Human +2 Bonus to one

Class Bonus – remove this, class bonus out of scope for point buy

Race Bonus  
 If charRace = Dwarf +2 conScore, +2 wisScore, -2 chaScore  
 If charRace = Elf +2 dexScore, +2 intScore, -2 chaScore  
 If charRace = Gnome +2 conScore, +2 chaScore, -2 strScore  
 If charRace = Halfling +2 dexScore, +2 chaScore, -2 strScore  
 If charRace = Half Elf, Half Orc, Human +2 score of choice

Alignment  
 9 grid display of alignment chart

|  |  |  |
| --- | --- | --- |
| LG | NG | CG |
| LN | N | CG |
| LE | NE | CE |

#Error if any Evil Alignment is chosen

Point Buy  
 20 point pool  
 Base state is 10  
 MOD goes up every 2 points past 10  
 MOD goes down every 2 points below 10, starting at 9  
 If score if less than 7, error  
 If Score is 7-8, points cost -2 per point  
 If Score is 9-10, points cost -1 per point  
 If Score is 10-12, points cost 1 per point  
 If Score is 13-15, points cost 2 per point  
 If Score is 16-17, points cost 3 per point  
 If Score is 18, points cost 4 per point

Random point buy – remove this, random point buy not very useful for players  
Replace with Reset points  
 Set points spent to 0  
 Set points left to 20  
 Set all scores to 10

Auto Point Buy  
 If Class = Barbarian – strScore = 17, dexScore = 14, conScore = 14, intScore = 8, wisScore = 13, chaScore = 7  
 - strMOD = 3, dexMOD = 2, conMOD = 2, intMOD = -1, wisMOD = 1, chaMOD = -2  
  
 If Class = Bard – strScore = 12, dexScore = 14, conScore = 14, intScore = 11, wisScore = 11, chaScore = 15  
 - strMOD = 1, dexMOD = 2, conMOD = 2, intMOD = 0, wisMOD = 0, chaMOD = 2  
  
 If Class = Cleric – strScore = 13, dexScore = 10, conScore = 12, intScore = 8, wisScore = 18, chaScore = 10  
 - strMOD = 2, dexMOD = 0, conMOD = 1, intMOD = -1, wisMOD = 4, chaMOD = 0  
  
 If Class = Druid – strScore = 14, dexScore = 14, conScore = 14, intScore = 9, wisScore = 16, chaScore = 7  
 - strMOD = 2, dexMOD = 2, conMOD = 2, intMOD = -1, wisMOD = 3, chaMOD = -2  
  
 If Class = Fighter – strScore = 15, dexScore = 17, conScore = 14, intScore = 7, wisScore = 13, chaScore = 8  
 - strMOD = 2, dexMOD = 3, conMOD = 2, intMOD = -2, wisMOD = 2, chaMOD = -1  
  
 If Class = Monk – strScore = 17, dexScore = 13, conScore = 14, intScore = 8, wisScore = 16, chaScore = 7  
 - strMOD = 3, dexMOD = 1, conMOD = 2, intMOD = -1, wisMOD = 3, chaMOD = -2  
  
 If Class = Paladin – strScore = 16, dexScore = 12, conScore = 14, intScore = 10, wisScore = 8, chaScore = 14  
 - strMOD = 3, dexMOD = 1, conMOD = 2, intMOD = 0, wisMOD = -1, chaMOD = 2  
  
 If Class = RangerArcher – strScore = 12, dexScore = 18, conScore = 12, intScore = 10, wisScore = 13, chaScore = 7  
 - strMOD = 1, dexMOD = 4, conMOD = 1, intMOD = 0, wisMOD = 2, chaMOD = -2  
  
 If Class = RangerMelee – strScore = 16, dexScore = 14, conScore = 14, intScore = 11, wisScore = 13, chaScore = 7  
 - strMOD = 3, dexMOD = 2, conMOD = 2, intMOD = 0, wisMOD = 2, chaMOD = -2  
  
 If Class = Rogue – strScore = 8, dexScore = 17, conScore = 14, intScore = 12, wisScore = 11, chaScore = 12  
 - strMOD = -1, dexMOD = 3, conMOD = 2, intMOD = 1, wisMOD = 0, chaMOD = 1

If Class = Sorcerer – strScore = 7, dexScore = 12, conScore = 13, intScore = 10, wisScore = 12, chaScore = 18  
 - strMOD = -2, dexMOD = 1, conMOD = 1, intMOD = 0, wisMOD = 1, chaMOD = 4  
  
 If Class = Wizard – strScore = 7, dexScore = 14, conScore = 14, intScore = 18, wisScore = 11, chaScore = 7  
 - strMOD = -2, dexMOD = 2, conMOD = 2, intMOD = 4, wisMOD = 1, chaMOD = -2

Set points spent to 20  
 Set points left to 0  
#Scores from suggested values via <https://rpgbot.net/pathfinder/characters/classes/>

Dice Roller  
 Roll D20 – random(1,20) #Most commonly rolled  
 Roll D100 – random(1,100)  
 Roll D12 – random(1,12)  
 Roll D10 – random(1,10)  
 Roll D8 – random(1,8)  
 Roll D6 – random(1,6)  
 Roll D4 – random(1,4)  
 Roll D2 – random(1,2)  
 print Die Result in box

Start Over  
 Reset all variables to default/blank

Save   
 Output text file – no die roller

## Progress of project – 10/2/2021:

Completed:

Pseudocode and known settings.

Desired Layout of labels, buttons, inputs.

Die Rolling functionality

Exit button

Reset button for all

Reset button for ability scores

Auto Points button is functional

In Progress:

Writing code for GUI application - coding labels, buttons, and inputs.

Die rolling layout on screen – align columns

On-going project documentation

Pop-up window for selecting character class to be radio button selection list

Pass character class, or charClass, to autoPoints module for if statements to pick points based on class

Problem areas:

Layout of rows for conforming to Window size – Rows are way too far apart at this point.  
Changed Window size for better layout

Research for prompt box to include radio buttons, not just a string input  
In progress – will need this fixed for auto-points to work based on selected character class, or charClass

## Project Status Update – 10/10/2021

Completed:

User Guide

Images for character classes and PFS logo

Pop-up Window for Class Selection – text entry at this time

charClass as Global variable for functions – auto points working for each Class

Layout looking better for character images using rowspan

In Progress:

On-going project documentation

Pop-up window for selecting character class to be radio button selection list

Ability modifier manual + / - button functions

Problem areas:

Research for prompt box to include radio buttons, not just a string input  
Still working on this – tried converting entire program from breezypython to only tkinter but ran into problems  
Was able to test a radio button menu and OK button to exit but problems were returning value from tk.Radiobutton

Manual ability scores not yet built

Race and Alignment not yet built

## Validation Testing – 10/10/2021

**Images** – Character images working based on text-entry of class selection for each class, PFS Logo appears in top left main window  
\*fixes include making all images .GIF and using rowspan to not mis-align other fields when images appear

**Pick Abilities for Me** – verified auto-points work based on text-entry of class selection for each class  
\*fixes include using charClass as a Global Variable

**Die Roller** – verified each die function outputs correct results

**Call back and Exit button:  
Reset Ability Scores** – verified point pool and abilities reset to defaults

**Start Over / Reset** – verified that all fields, character image, resets to defaults

**Exit** – button exits program correctly

**Character Class** – opens new window correctly (Monk entry shown below, with Pick Abilities For Me used)  
\*will attempt again to make this a radio button menu for secure coding selection

Graphical user interface, table

Description automatically generated