# Assignment 1

In [1]:
```
"""
    Name: Pragy Parashar
    Student ID: 31940757
    Date Created: Aug 19, 2022
    Last Modified: Aug 26, 2022
    This Python program will allow a user to play a dice rolling game.
    The aim of Dice Rolling Game is for a player to roll 2-6 dice multiple times in each game.
    Victory is achieved when the median value is the same as the mean value in a result list of the dice rolls
    for all rounds of the game.
"""
```
```
None
```

# 1. Main menu function

In [2]:
```
def main_menu():
    """
    This is the game landing page. This function prints the start menu for the game and asks for a user input.
    The user can choose one of the three listed options else the function displays an error message asking user to
    input valid choice.
    :parameter - This function requires no parameters
    :return: "choice_main_menu" Player choice is type cast as an integer
    """
    print("----------------------Main Menu----------------------")
    print("|                                                   |")
    print("|         Please enter 1 to Start Game              |")
    print("|                                                   |")
    print("|         Please enter 2 to View History            |")
    print("|                                                   |")
    print("|         Please enter 3 to Exit                    |")
    print("|                                                   |")
    print("-----------------------------------------------------\n")

    choice_main_menu = input("Please enter your choice to proceed: ")

    if choice_main_menu.isdigit():
        if int(choice_main_menu) in (1, 2, 3):
            return int(choice_main_menu)
    else:
        print("Please enter a valid option.")
        return main_menu()
```

# Choose Number of Dice Function

In [ ]:
```
def choose_num_of_dice():
    """
    This function is invoked to let the player choose the number of dice the player wants to play the game with.
    The player can choose only dice between 2-6. All other values are handled with error message where user is asked
    to enter an acceptable value
    :parameter - The function requires no parameters
    :return: "number_of_dice". Player choice is type cast as integer
    """
    #  prints warning message for user to select the number within the mentioned range
    print("Note: Please choose a number in range 2-6.")

    # takes user input for number of dies
    number_of_dice = input("Please enter number of dies: ")

    #  Input validation. Checks if the input is a numeric value.
    if number_of_dice.isdigit():
        #  Type cast the user input from a string to integer value
        number_of_dice = int(number_of_dice)
        # Input validation. Checks if the user input is in the mentioned acceptable values
        if number_of_dice not in range(2, 7):
            # prints the error message if the input validation fails
            print("Please enter a value in range 2-6.\n")
            #  invokes the function again to receive a valid input
            return choose_num_of_dice()
        #  returns the valid value for number of dies from user input
        return number_of_dice
    #  if input is not a numeric value
    else:
        # prints the error message if the input validation fails
        print("Please enter a value in range 2-6.\n")
        #  invokes the function again to receive a valid input
        return choose_num_of_dice()
```

# Choose Type of Dice Function

In [3]:
```
def choose_type_of_dice():
    """
    This function is invoked to let the player choose the type of dice to proceed the game with. The input validations
    checks if the dice type if among 6, 8 or 9. The invalid values are handled with an error message asking user
    to provide a valid input value. This function also provides a visual aid as to how each dice type looks like.
```

```
    :parameter: The function requires no parameters
    :return: "type_of_dice". Player choice type cast as integer
    """
    #  print statements to show the user the types of dice and make a valid choice
    print("\n \nPlease choose a dice type: \n \n")
    print("\n---------------------- Enter 6 for this dice ----------------------")
    print(" Your Dice-6 looks like this: \n \n")
    print(" --------", "  --------", "  --------", "  --------", "  --------", "  --------")
    print("|        |", "|       0|", "|0       |", "|0      0|", "|0      0|", "|0      0|")
    print("|    0   |", "|        |", "|    0   |", "|        |", "|    0   |", "|0      0|")
    print("|        |", "| 0      |", "|       0|", "|0      0|", "|0      0|", "|0      0|")
    print(" --------", "  --------", "  --------", "  --------", "  --------", "  --------")

    print("\n---------------------- Enter 8 for this dice ----------------------")
    print(" Your Dice-8 looks like this: \n \n")
    print(" --------", "  --------", "  --------", "  --------", "  --------", "  --------")
    print("0       |", "|       0|", "|0       |", "|       0|", "|0      0|", "|0   0  0|")
    print("|        |", "|        |", "|    0   |", "|    0   |", "|        |", "|0   0  0|")
    print("|       0|", "| 0      |", "|       0|", "|0       |", "|0      0|", "|0      0|")
    print(" --------", "  --------", "  --------", "  --------", "  --------", "  --------")

    print("\n---------------------- Enter 9 for this ----------------------")
    print(" Your Dice-9 looks like this: \n \n")
    print(" --------", "  --------", "  --------", "  --------", "  --------", "  --------")
    print("|        |", "|        |", "|        |", "|        |", "|0      0|", "|0      0|")
    print("|    0   |", "|   0    |", "|    0   |", "|    0   |", "|    0   |", "|0      0|")
    print("|        |", "|        |", "|        |", "|        |", "|0      0|", "|0      0|")
    print(" --------", "  --------", "  --------", "  --------", "  --------", "  --------")
    print("\n \n")

    #  User input for selecting a die type
    type_of_dice = input("Please enter a dice type: \n")
    #  Input Validation. Checks if the user has entered a numeric value
    if type_of_dice.isdigit():
        #  Typecast the user input from string to integer value
        type_of_dice = int(type_of_dice)
        # Input Validation. Checks if the user input is valid
        if type_of_dice in (6, 8, 9):
            # returns the type of dice user has chosen
            return type_of_dice
        else:
            #  Prints error message if input is invalid
            print("Please enter a valid option.")
            #  Invokes the function to choose the dice with a valid option
            return choose_type_of_dice()
    else:
        #  Prints error message if input is invalid
        print("Please enter a valid option")
        #  Invokes the function to choose the dice with a valid option
        return choose_type_of_dice()
```

# 2. Game menu function

In [4]:

```
def game_menu():
    """
    This function is the landing page when the player chooses to start the game. It prints the game options for player
    to select from. The invalid input by player is handled by the displaying an error message asking user to input
    valid value.
    :parameter: The function requires no parameters
    :return: "choice_game_menu". PLayer choice type cast as integer
    """
    print("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Game Menu ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
    print("Press 1 to \" Select number of dies for the role \" \n")
    print("Press 2 to  \" Select the type of the dice you would like to roll \" \n")
    print("Press 3 to \" Roll dice \" \n")
    print("Press 4 to \" Check win or loose \" \n")
    print("Press 5 to \" Exit Game \" \n")

    choice_game_menu = input("Please enter your choice to proceed: ")

    if choice_game_menu.isdigit():
        if int(choice_game_menu) in (1, 2, 3, 4, 5):
            return int(choice_game_menu)
    else:
        print("Please enter a valid option.")
        return game_menu()
```

# 3. Roll dice function

In [5]:

```
def roll_dice(number_of_dice, type_of_dice):
    """
    This function is invoked for user to roll the dice after choosing the type and number of dice. The function also
    provides option to roll the dice more than once.
    :param number_of_dice: Number of dice chosen by the user
    :param type_of_dice: Type of dice chosen by the user
    :return:  "roll_list". List containing the tuple of values generated after the dice have been rolled
    """
    # creates an empty list to store the result of each round
    roll_list = []
    rolled_num = tuple()
    # number of dice given by user
```

```
number_of_dice = number_of_dice
#  type of dice given by user
type_of_dice = type_of_dice

''' checks if the dice type is 6
creates a tuple by randomly selecting values between 1-6 inclusive for all dies that player has entered

else check if the dice type is 8
creates a tuple by randomly selecting values among 2, 3, 4, and 8 inclusive for all dies that player has entered

else checks if the dice type is 9
creates a tuple by randomly selecting values among 1, 5, and 6 inclusive for all dies that player has entered
'''
if int(type_of_dice) == 6:
    rolled_num = tuple(rand.randint(1, 6) for num in range(number_of_dice))
elif int(type_of_dice) == 8:
    rolled_num = tuple(rand.choice([2, 3, 4, 8]) for num in range(number_of_dice))
elif int(type_of_dice) == 9:
    rolled_num = tuple(rand.choice([1, 5, 6]) for num in range(number_of_dice))
else:
    print("Please enter valid dice type!")

#  checks if the player has played any games by checking length of the resulting list
if len(rolled_num) != 0:
    # dice_face = dice_faces(rolled_num)
    # adds the resulting tuples from a roll to a list
    roll_list.append(rolled_num)

    # print(f"\n{dice_face}")
    # asks for user input if they want to go for another round of the game
    replay_option = 'start'
    while replay_option not in ('1', '2'):
        for roll in roll_list:
            print("You Rolled: ", roll)
        if replay_option == 'start':
            replay_option = input("Would you like to play again? Enter 1 for Yes or 2 for No: ")
        else:
            print('Please enter a valid option.')
            replay_option = input("Would you like to play again? Enter 1 for Yes or 2 for No: ")
        #  checks if the user input is an integer
    if replay_option.isdigit():
        latest_rolled_dies = 0
        #  checks if the player wants to play another round
        if int(replay_option) == 1:
            #  invokes functions to start the game
            latest_rolled_dies = choose_num_of_dice()
            new_roll_list = roll_dice(latest_rolled_dies, type_of_dice)
            for roll in new_roll_list:
                roll_list.append(roll)
return roll_list
```

# 4. Check win/lose function

In [6]:

```
def check_win_or_loose(game_roll):
    """
    This function checks whether the user has won or lost the game. The function checks for mean and median values
    for the roll made by the user and returns "Win" if the mean is equal to the median and "Lost" when mean is not
    equal to the median or if the player has played more than 10 round in a game.

    :param game_roll: List of tuples of each roll made by the player
    :return: "win_or_loose". String indicating whether the player has lost or won the game
    """
#  creates an empty list to store all the values of the rolls
sorted_roll_list = []
win_or_loose = ""
#  creates a variable to store the median of the result
median_value = 0
#  creates a variable to store the mean of the result
mean_value = 0
#  checks if the game has been played or not
if len(game_roll) != 0:
    # loops through the list storing the result of all the rolls
    for roll in game_roll:
        #  checks if each roll has valid results
        if len(roll) != 0:
            #  loops each round result
            for num in roll:
                #  adds to the list values generated in each roll on each dice
                sorted_roll_list.append(num)
#  checks if the resulting has values or not
if len(sorted_roll_list) != 0:
    #  function to sort the list in an ascending oder of the rolled number
    sorted_roll_list.sort()
    #  checks if the
    if len(sorted_roll_list) % 2 == 0:

        median_value_1 = sorted_roll_list[len(sorted_roll_list) // 2]
        median_value_2 = sorted_roll_list[(len(sorted_roll_list) // 2) - 1]

        median_value = (median_value_1 + median_value_2) // 2

        mean_value = sum(sorted_roll_list) // len(sorted_roll_list)

    else:
        median_value = sorted_roll_list[len(sorted_roll_list) // 2]
```

```
            mean_value = sum(sorted_roll_list) // len(sorted_roll_list)

    if median_value == mean_value:
        win_or_loose = "Win"

    elif len(game_roll) > 10:
        win_or_loose = "Lost"

    else:
        win_or_loose = "Lost"
    return win_or_loose
```

# 5. Game history function

In [7]:

```
def display_game_history(game_history_list):
    """
    This function prints the game history of the player indicating number of rounds played in a game, number of dice
    used in a game and result of the game
    :param game_history_list: list containing the number of rounds played in a game, number of dice
    used in a game and result of the game
    :return: None
    """
    if len(game_history_list) == 0:
        print("Failed to retrieve the history! You have not played any games yet.")
    else:
        sorted_rolls = []
        sorted_set = set()
        # loop gets the number of dice used in each game and appends the same to a list
        for games in game_history_list:
            sorted_rolls.append(games[1][1])
        # sorts the number of dice used in an ascending order
        sorted_rolls.sort()

        # loop to add the sorted number of dices to a set to remove the duplicates
        for dice in sorted_rolls:
            sorted_set.add(dice)
        # loops through the game history list and sorted set to print the game history in ascending order of dice numbers
        for rolls in sorted_set:
            for games in game_history_list:
                if rolls == games[1][1]:
                    print(f"~~~~~~~~~~~~~~~~~~~~~~{games[0]}~~~~~~~~~~~~~~~~~~~~~~")
                    print(f"Dice Rolling Times: {games[1][0]}\nNumber of Dice: {games[1][1]}\nResult: {games[1][2]}")
```

# 6. Main function

In [8]:

```
def main():
    """
    This is the main control unit of the game. It contains the input validation made by user
    The function navigates to the appropriate functions according to the inputs made by the user.
    :return: None
    """
    game_history_list = []
    game_dict = {}

    main_menu_loop = True
    # loop will keep running the start menu unless the user wants to exit
    while main_menu_loop:
        main_menu_choice = main_menu()

        if main_menu_choice == 1:
game_menu_loop = True
number_of_dice = -9
type_of_dice = -9
win_or_loose = ''
rolled_list = []
# loop will keep running until the player wants to exit the game
while game_menu_loop:

    game_menu_choice = game_menu()
    if game_menu_choice == 1:
        number_of_dice = choose_num_of_dice()
    elif game_menu_choice == 2:
        type_of_dice = choose_type_of_dice()
    elif game_menu_choice == 3:
        if number_of_dice == -9:
            #  prints error message if user has made no selection
            print("Please select the number of dies.\n")
            '''
            Invokes function and navigates back to game manu for user to select number of dies
            '''
            continue
        #  Validation. Checks if user has selected type of dice.
        elif type_of_dice == -9:
            #  prints error message if user has made no selection
            print("Please select the type of dice.\n")
            '''
            Invokes function and navigates back to game manu for user to select type of dice
            '''
            continue
        #  Validation. Checks if user has selected number and type of dice.
        elif type_of_dice == -9 and number_of_dice == -9:
            #  prints error message if user has made no selection
```

```
                print("Please select the type and number of dies.\n")
                '''
                Invokes function and navigates back to game manu for user to select type of dice
                '''
                continue

            else:
                if rolled_list:
                    number_of_dice = choose_num_of_dice()

                rolls = roll_dice(number_of_dice, type_of_dice)

                for rounds in rolls:
                    rolled_list.append(rounds)
                print(rolled_list)
            continue

        elif game_menu_choice == 4:
            if len(rolled_list) == 0:
                print("Sorry you have not played any games right now.")

            else:
                win_or_loose = check_win_or_loose(rolled_list)
                print(f"You {win_or_loose} this game!")
            continue

        elif game_menu_choice == 5:
            if len(rolled_list) == 0:
                game_menu_loop = False
            else:
                dice_in_a_game = 0
                temp_list = rolled_list
                # loops through the rolled_list to get the total number of dice player used in a game
                for rolls in rolled_list:
                    dice_in_a_game += len(rolls)

                # validation to check if the player skipped to check the result of any roll
                if win_or_loose == '' and len(rolled_list) != 0:
                    win_or_loose = check_win_or_loose(rolled_list)

                if len(game_dict) == 0 and len(game_history_list) == 0:
                    game_dict[1] = tuple([len(rolled_list), dice_in_a_game, win_or_loose])
                    game_history_list.append(["Game-1", game_dict[1]])
                else:
                    keys_list = max(game_dict.keys())
                    last_key = keys_list + 1
                    game_dict[last_key] = tuple([len(rolled_list), dice_in_a_game, win_or_loose])
                    game_history_list.append([f"Game-{last_key}", game_dict[last_key]])
                game_menu_loop = False
        else:
            print("Please enter a valid option.")
            continue

    elif main_menu_choice == 2:
        if len(game_history_list) == 0:
            print("Sorry! Failed to retrieve a game history. You have not played any games.")
        else:
            display_game_history(game_history_list)
        continue

    elif main_menu_choice == 3:
        print('Goodbye! Thanks for playing the game.')
        main_menu_loop = False

    else:
        print("Please enter a valid option.")
        continue
```

In [ ]:

```
if __name__ == '__main__':
    main()
```