# MN404 Assignment 2 – Christian Milicevic

**User** **Requirements:**

* Introduce and describe the game, the computer shall introduce itself as the AI player, and prompt the human players for his or her names along with MIT Ids.
* Program is to be interactive to receive and display the response.
* To display a greeting to the human player that incorporates the name provided in response to the prompt: **Welcome <**user’s name**> to the game of Taking Stones**.
* The user’s name must be output in proper name case (i.e., mixed case starting with upper case for the first letter) regardless of the case the user types in.
* The users shall input the number of stones to be picked either by using a standard random number generator function or may input according to their own strategy but the number should be less than or equal to three (3) and more than zero.
* The program must reject and display a valid message if the number of stones to be removed are either less than one (1) or more than three (3) and remind them the game constrains. In case of invalid entries program to prompt that player to re-enter an appropriate number.
* Your program should have the computer use the optimal playing strategy. The optimal strategy is as follows: Divide the remaining number of stones by three. If the remainder is zero, then two stones are removed, or else one stone is removed. For example, if the remaining number of stones is nine or fifteen, then two stones are removed; if the remaining number of stones is eight or ten, then one stone is removed.
* Both the human players shall have their own strategies to play.

* When one of the players has won a game, the program must output a congratulatory message naming the winner along with ID.
* Your program should allow the users to play additional games of NIM as long as he/she enters a “y” or “yes” (lowercase or uppercase) in response to a “Do you want to play again?” prompt.

**Analysis**

The goal is to make a NIM game with 2 players and an AI, the end goal is to have all the user requirements (as stated above) and a functional no error game.

**Design (algorithm):**

* Game launches
* Ask for Player1 Details
* Error check
* Ask for Player2 Details
* Error Check
* Computer Details
* Error Check
* Ready Up Statement
* Set Stone Count between 30-50
* Game starts
* Goes to Player1 Turn
* Then Player2Turn
* Then ComputerTurn
* Computer takes 1-3 stones from pile based off RNG
* Check Win Con for each Turn (player1, 2 and computer)
* Repeat until stoneCount = 0

**Implementation Phase:**

#Taking Stones Game - MN404 Christian Milicevic

#Libaries/Vars

import random

#Game Intro Sequence Function

def GameIntro():

global r\_total\_stones

r\_total\_stones = random.randint(30, 50)

print("\nWelcome to the game of taking of stones!")

print("#############################################################################")

print("The game goal is to take the last stone to be the winner")

print("You can take between 1-3 stones per turn")

print("The game will go in the rotational order of Player1, then Player2 then the computer AI")

print("Player1 will pick the amount of stones in the game which is in the range of [30-50]")

print("The game is a 3 players game consitiing of 2 human players and 1 computer AI")

print("#############################################################################\n")

print("Are you ready to play?")

#User1, User2 and ComputerAI string input and captalization condtions

user1 = input("Please enter in your username Player 1!: ");

global user1\_Id

user1\_Id = input("Please enter in Player1 MIT Id: ");

print("\n------------------------------------------")

print("Welcome Player1:", user1.lower().capitalize(), "with the MIT Id:", user1\_Id, "to game of stones")

print("------------------------------------------\n")

user2 = input("Please enter in your username Player 2!: ");

global user2\_Id

user2\_Id = input("Please enter in Player2 MIT Id: ");

print("\n------------------------------------------")

print("Welcome Player2:", user2.lower().capitalize(), "with the MIT Id:", user2\_Id, "to game of stones")

print("------------------------------------------\n")

user3\_AI = input("Please give the Computer a username: ")

print("\n------------------------------------------")

print("Welcome Computer:", user3\_AI.lower().capitalize(), "to game of stones")

print("This is an AI computer that will take it's turn after Player2")

print("------------------------------------------\n")

userReady = ""

while userReady != "ready":

userReady = input("If you are ready to play type, ready: ")

if userReady == "ready": PlayGame()

else: print("that is not the correct input please try again!\n")

#Game intro is over, set stone and play game function

def PlayGame():

print("\n------------------------------------------")

print("The number of stones in this game has been set to", r\_total\_stones)

print("------------------------------------------\n")

Player1() #GoTo Player1 Function

def Player1():

global r\_total\_stones

global user1\_Id

user = 0

while user != 1 or user != 2 or user != 3:

print("\nPLAYER1 TURN")

print("Player1: Remove [1-3] stones from the total of ", r\_total\_stones, "stones")

user = input("Amount of stones to remove: ")

if user.isdigit():

if int(user) == 1 or int(user) == 2 or int(user) == 3:

r\_total\_stones = r\_total\_stones - int(user)

break;

else: print("Invalid entry please try again!")

#check for win con

if r\_total\_stones <= 0:

print("\nPlayer1 has won the game, has he followinig MIT ID:!", user1\_Id)

PlayAgain() #PlayAgain Condition

else: Player2()

def Player2():

global r\_total\_stones

global user2\_Id

user = 0

while user != 1 or user != 2 or user != 3:

print("\nPLAYER2 TURN")

print("Player2: Remove [1-3] stones from the total of ", r\_total\_stones, "stones")

user = input("Amount of stones to remove: ")

if user.isdigit():

if int(user) == 1 or int(user) == 2 or int(user) == 3:

r\_total\_stones = r\_total\_stones - int(user)

break;

else: print("Invalid entry please try again!")

#check for win con

if r\_total\_stones <= 0:

print("\nPlayer2 has won the game, has he followinig MIT ID:!", user2\_Id)

PlayAgain() #PlayAgain Condition

else: Computer\_AI()

#AI Function

def Computer\_AI():

global r\_total\_stones

print("\nCOMPUTER\_AI TURN")

print("Computer\_AI: Remove [1-3] stones from the total of ", r\_total\_stones, "stones")

#AI Logic Here

stones\_to\_take = random.randint(1, 3)

r\_total\_stones = r\_total\_stones - stones\_to\_take

print("Computer\_AI has removed", stones\_to\_take, "from the pile")

#Win Con

if r\_total\_stones <= 0:

print("\nComputerAI has won the game!")

PlayAgain() #PlayAgain Condition

else: Player1()

#PlayAgain Function

def PlayAgain():

user = ""

while user.lower() != "y" or user != "yes":

user = input("\nPlease enter in y or yes to play again: ")

if user == "y" or user == "Y" or user == "yes" or user == "YES": GameIntro()

else: print("Not a valid entry, please try again!")

#Main

GameIntro()

* **Testing:**

No bugs from what I can see and it works

**Video of Program**

