# Game of Nim

Professor Minhthong Nguyen

Students: Umar Khan, LeslieMarie Soliven, Anthony Diaz, Bianette Soriano, Gabriel Rodriguez

CECS 174

7th December 2016

**Introduction to the Program:**

This is the program that has designed the game of nim. The game of nim has been played between the user and the computer. There are two modes that the computer can play this game. The mode that the computer play smart or with the strategy is 1 and the dumb or without the strategy is 0. The game has a basic idea of picking up the cards from the piles which varies from 1 to half of the size of the pile. Anything beyond that limit shoots out an error to the user, where the computer never crosses this limit as the computer take has the strategy set to it in smart mode and random pick in dumb mode. In the end, whoever has to pick up the last card from the pile loses the game. The game in the end decides the winner and displays the winner on the screen.

**Pseudocode for the Program:**

**Methods.py Pseudocode**

```
Import random class

Function pileSize ()
    Declare integer a
    Set integer a = random.randint (10,100)
    Return integer a
End function


Function firstPlayer ()
    Declare integer b
    Set integer b = random.randint (0,1)
    Return integer b
End function


Function difficulty ()
    Declare integer c
    Set integer c = random.randint (0,1)
    Return integer c
End function


Function computer_take(pileSize)
    Set integer comp_take = 0
```

```
        If difficulty = 1
                If pileSize > 63 and pileSize <= 100
                        Set integer comp_take = pileSize – 63
                End if
                Else if pileSize > 31 and pileSize <= 62
                        Set integer comp_take = pileSize – 31
                End else if
                Else if pileSize > 15 and pileSize <= 30
                        Set integer comp_take = pileSize -15
                End else if
                Else if pileSize > 7 and pileSize <= 14
                        Set integer comp_take = pileSize – 7
                End else if
                Else if pileSize > 3 and pileSize <= 6
                        Set integer comp_take = pileSize – 3
                End else if
                Else if pileSize > 1 and pileSize <= 2
                        Set integer comp_take = pileSize – 1
                End else if
                Else
                        Set integer comp_take = random.randint (1, pileSize//2)
                End else
        End if
        Else
                Set integer comp_take = random.randint (1, pileSize//2)
        End else
        Return integer comp_take
End function
```

## Game.py Pseudocode

```
From methods importing pileSize, firstPlayer, difficulty, computer_take
Declare integer pileSize
Set integer pileSize = function pileSize ()
Declare integer firstPlayer
Set integer firstPlayer = function firstPlayer ()
Declare integer difficulty
Set integer difficulty = function difficulty ()
Print (‘Pile size:’, pileSize, ‘First player:’, firstPlayer, ‘Difficulty:’, difficulty)
If firstPlayer == 0
    Print (‘Computer goes first’)
    While True
        Print (‘Computers turn’)
        Set integer pileSize = pileSize – function comp_take(pileSize)
        Print (‘New Pile size:’, pileSize)
        If pileSize == 1
            Print (‘Computer Wins’)
            Break the condition
        Set integer user_take = eval (input (‘Enter how many you will take:’)
        While (user_take > pileSize//2) or user_take <= 0
            Set integer user_take = eval (input (‘Please enter a valid enter how many you will take’)
        End while
        Set integer pileSize = pileSize – user_take
        Print (‘New pile size:’, pileSize)
        If pileSize == 1
            Print (‘You Win!’)
            Break the condition
    End while
End if
```

```
Else if firstPlayer == 1
    Print ('You go first')
    While True
        Set integer user_take = eval (input ('Enter how many you will take')
        While (user_take > pileSize //2) or user_take <= 0
            Set integer user_take = eval (input ('Please enter a valid enter how many
            you will take:'))
        End while
        Set integer pileSize = pileSize – user_take
        Print ('New pile size:', pileSize)
        If pileSize == 1
            Print ('You Win!')
            Break the condition

        Set integer pileSize = pileSize – function comp_take(pileSize)
        Print ('New pile size:', pileSize)
        If pileSize == 1
            Print ('Computer Wins')
            Break the condition
    End while
End else if
```

**Methods:**

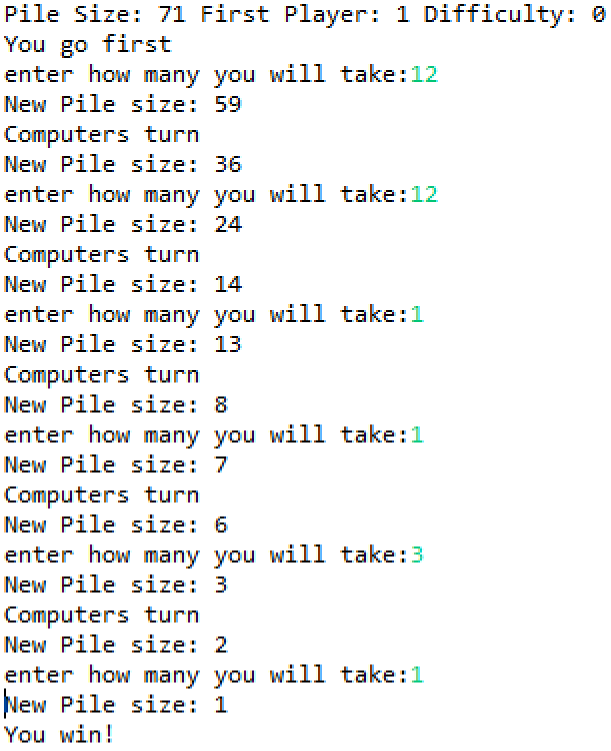
- **Def pileSize ():** This method generates a random number between 10 and 100, this number determines the size of the pile.
- **Def firstPlayer ():** This method generates a random integer between 0 and 1 to decide whether the computer or the user takes the first turn.
- **Def difficulty ():** This method generates a random integer between 0 and 1 to whether the computer plays smart or stupid. 1 is for the smart mode and 0 is for the dumb mode.
- **Def computer_take (pileSize):** This method determines the number of cards taken by the computer depending on the level of difficulty generated by the difficulty function. If the game is in smart mode and the pile size is greater than 63 and less than or equal to 100 it will take away at most 63. If the pile size is between 31 and 62 the computer can take 31

at most. If the pile size is between 14 and 7 the computer can take 7 at most. If the pile size is between 7 and 3 you can take 3 at most. If the pile size is between 1 and 2, computer can take only 1 from the pile and the it will be the last else statement where the computer will determine randomly since there will be only one card left, and the program will decide the winner. If the game is played in dumb mode, the computer takes a random number between 1 and the pile size divided by 2. It follows the same strategy throughout the code.

- **Def main ():** The main for this function has been created as a tester for the game. The methods have been used in the game tester. The methods were imported into the game.py where we used the combination of the methods, loops, and if-else statements to test the game.

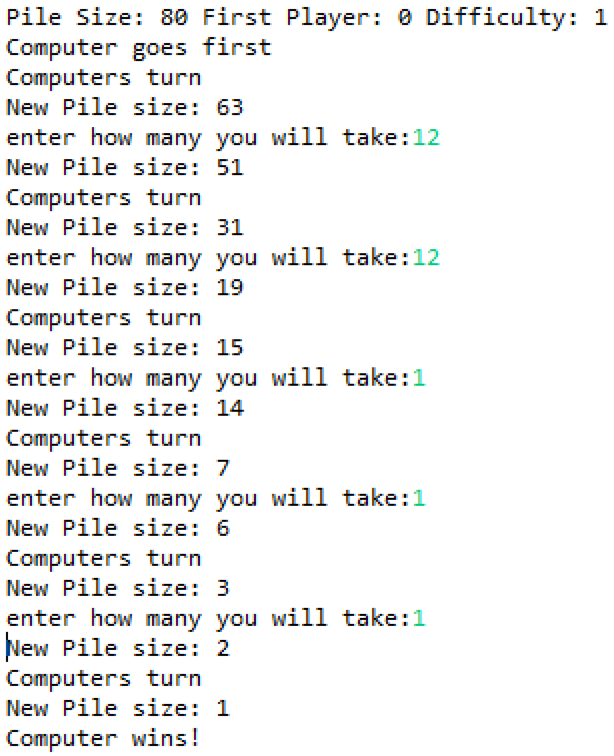
**Test Cases:**

- Test Case 1: When the computer is playing in dumb mode, and the user goes first



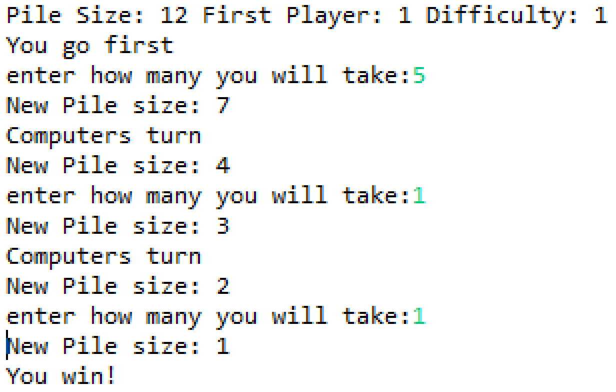
```
Pile Size: 71 First Player: 1 Difficulty: 0
You go first
enter how many you will take:12
New Pile size: 59
Computers turn
New Pile size: 36
enter how many you will take:12
New Pile size: 24
Computers turn
New Pile size: 14
enter how many you will take:1
New Pile size: 13
Computers turn
New Pile size: 8
enter how many you will take:1
New Pile size: 7
Computers turn
New Pile size: 6
enter how many you will take:3
New Pile size: 3
Computers turn
New Pile size: 2
enter how many you will take:1
New Pile size: 1
You win!
```

- Test Case 2: When the computer is playing in smart mode, and it goes first.



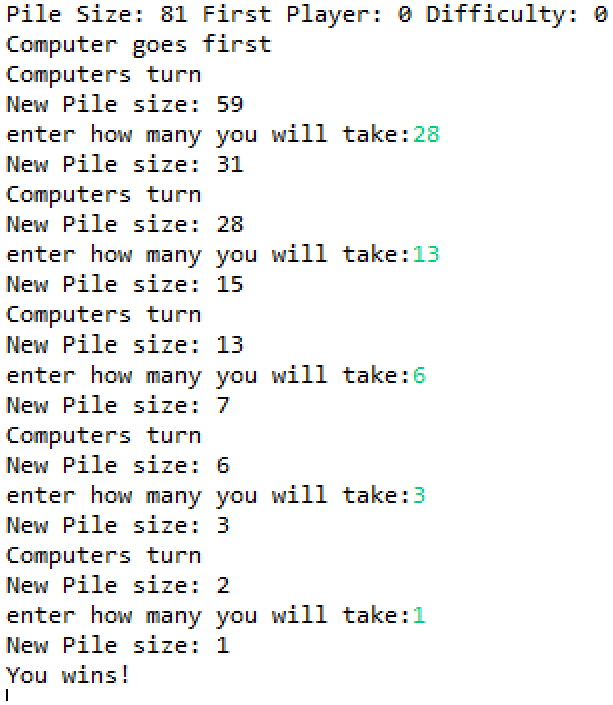
```
Pile Size: 80 First Player: 0 Difficulty: 1
Computer goes first
Computers turn
New Pile size: 63
enter how many you will take:12
New Pile size: 51
Computers turn
New Pile size: 31
enter how many you will take:12
New Pile size: 19
Computers turn
New Pile size: 15
enter how many you will take:1
New Pile size: 14
Computers turn
New Pile size: 7
enter how many you will take:1
New Pile size: 6
Computers turn
New Pile size: 3
enter how many you will take:1
New Pile size: 2
Computers turn
New Pile size: 1
Computer wins!
```

- Test Case 3: When the computer is playing in smart mode, and the user goes first so if the user plays with smart strategy, the user canwin the game.



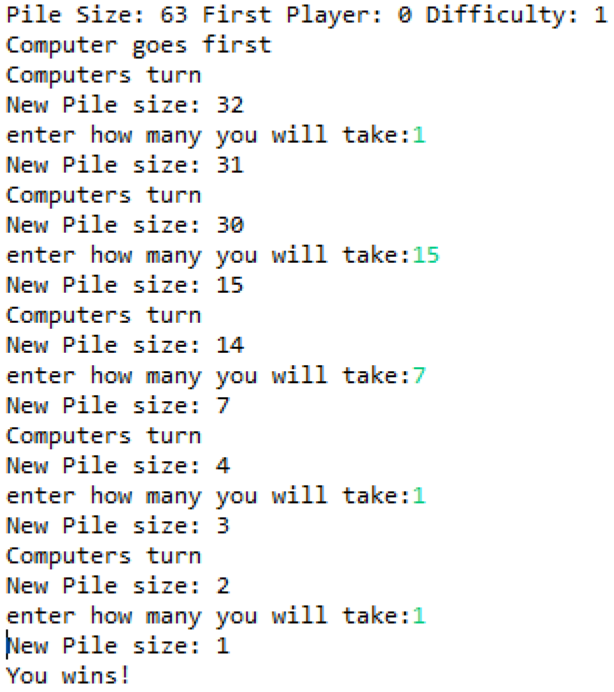
```
Pile Size: 12 First Player: 1 Difficulty: 1
You go first
enter how many you will take:5
New Pile size: 7
Computers turn
New Pile size: 4
enter how many you will take:1
New Pile size: 3
Computers turn
New Pile size: 2
enter how many you will take:1
New Pile size: 1
You win!
```

- Test Case 4: When the computer is playing in dumb mode, and the user goes first. For this case, there is a high chance that the user can win if he knows the strategy.



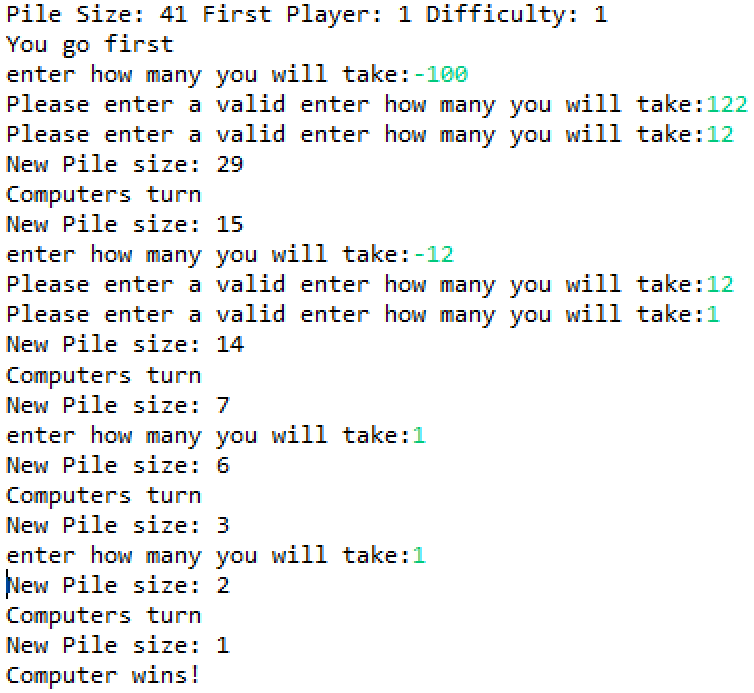
```
Pile Size: 81 First Player: 0 Difficulty: 0
Computer goes first
Computers turn
New Pile size: 59
enter how many you will take:28
New Pile size: 31
Computers turn
New Pile size: 28
enter how many you will take:13
New Pile size: 15
Computers turn
New Pile size: 13
enter how many you will take:6
New Pile size: 7
Computers turn
New Pile size: 6
enter how many you will take:3
New Pile size: 3
Computers turn
New Pile size: 2
enter how many you will take:1
New Pile size: 1
You wins!
```

- Test Case 5: When the computer is playing in smart mode, and the pileSize lands on a strategic number so the user has the chance to win unless he doesn't plays smart.



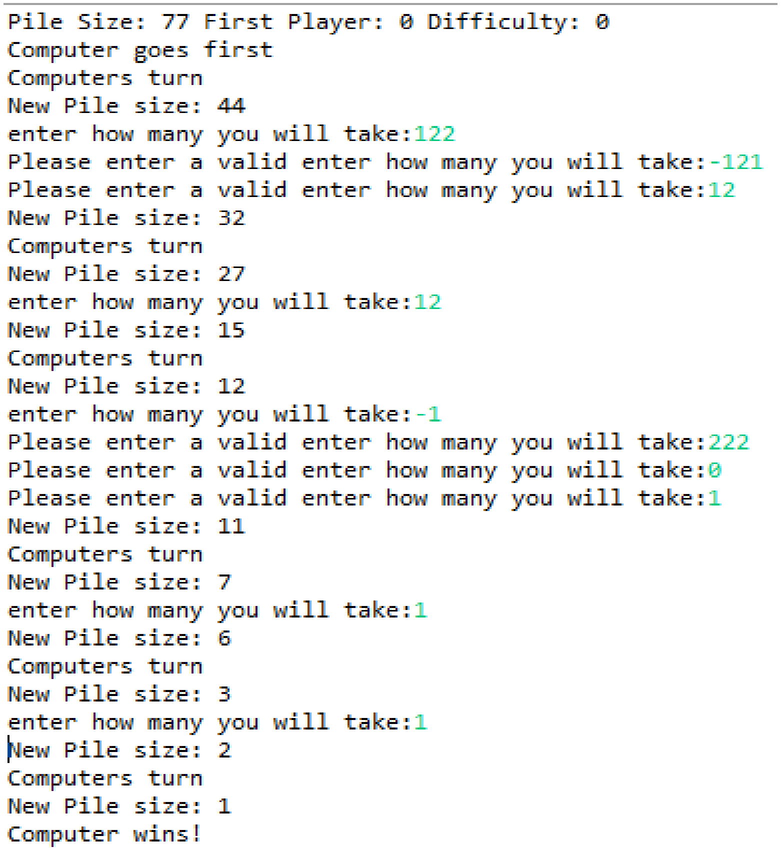
```
Pile Size: 63 First Player: 0 Difficulty: 1
Computer goes first
Computers turn
New Pile size: 32
enter how many you will take:1
New Pile size: 31
Computers turn
New Pile size: 30
enter how many you will take:15
New Pile size: 15
Computers turn
New Pile size: 14
enter how many you will take:7
New Pile size: 7
Computers turn
New Pile size: 4
enter how many you will take:1
New Pile size: 3
Computers turn
New Pile size: 2
enter how many you will take:1
New Pile size: 1
You wins!
```

- Test Case 6: When the user goes first, testing for the input validation.Input validation works throughout the code.



```
Pile Size: 41 First Player: 1 Difficulty: 1
You go first
enter how many you will take:-100
Please enter a valid enter how many you will take:122
Please enter a valid enter how many you will take:12
New Pile size: 29
Computers turn
New Pile size: 15
enter how many you will take:-12
Please enter a valid enter how many you will take:12
Please enter a valid enter how many you will take:1
New Pile size: 14
Computers turn
New Pile size: 7
enter how many you will take:1
New Pile size: 6
Computers turn
New Pile size: 3
enter how many you will take:1
New Pile size: 2
Computers turn
New Pile size: 1
Computer wins!
```

- Test Case 7: When the computer goes first, testing for input validation. Input validation works throughout the code.



```
Pile Size: 77 First Player: 0 Difficulty: 0
Computer goes first
Computers turn
New Pile size: 44
enter how many you will take:122
Please enter a valid enter how many you will take:-121
Please enter a valid enter how many you will take:12
New Pile size: 32
Computers turn
New Pile size: 27
enter how many you will take:12
New Pile size: 15
Computers turn
New Pile size: 12
enter how many you will take:-1
Please enter a valid enter how many you will take:222
Please enter a valid enter how many you will take:0
Please enter a valid enter how many you will take:1
New Pile size: 11
Computers turn
New Pile size: 7
enter how many you will take:1
New Pile size: 6
Computers turn
New Pile size: 3
enter how many you will take:1
New Pile size: 2
Computers turn
New Pile size: 1
Computer wins!
```

## Code of the Method.py:

```python
import random
#Generate a random integer between 10 and 100 to denote the initial size of the pile.
def pileSize():
    a = random.randint(10,100)
    return a

#Generate a random integer between 0 and 1 to decide whether the computer or the
human takes the first turn.
#1 for player going first and 0 for computer going first
def firstPlayer():
    b = random.randint(0,1)
    return b

#Generate a random integer between 0 and 1 to decide whether the computer plays smart
or stupid.
#1 for it being playing smart and 0 for playing dumb
def difficulty():
    c = random.randint(0,1)
    return c

#Computes how much computer has to take
def computer_take(pileSize):
    comp_take=0
    if difficulty == 1: #smart mode
        if pileSize > 63 and pileSize <= 100:
            comp_take= pileSize- 63
        elif pileSize > 31 and pileSize <= 62:
            comp_take= pileSize- 31
        elif pileSize > 15 and pileSize <= 30:
            comp_take= pileSize- 15
        elif pileSize > 7 and pileSize <= 14:
            comp_take= pileSize- 7
        elif pileSize > 3 and pileSize <= 6:
            comp_take= pileSize- 3
        elif pileSize > 1 and pileSize <=2:
            comp_take= pileSize - 1
        else:
            comp_take = random.randint(1,pileSize//2)
    else:# Stupid mode
        comp_take = random.randint(1,pileSize//2)
    return comp_take
```

## Code of the Game.py

```python
from methods import pileSize, firstPlayer, difficulty, computer_take
pileSize = pileSize()
firstPlayer = firstPlayer()
difficulty = difficulty()
# 1 = smart -- 0 = dumb
print("Pile Size:", pileSize, "First Player:", firstPlayer, "Difficulty:",
difficulty)

#When computer goes first
```

```
if firstPlayer == 0:
    print("Computer goes first")
    while True: #making sure pile is not less than 1
        print("Computers turn")
        pileSize = pileSize - computer_take(pileSize)
        print("New Pile size:", pileSize)
        if pileSize == 1:
            print('Computer wins!')
            break
        user_take = int(input("enter how many you will take:"))
        while (user_take > pileSize // 2) or user_take <= 0:
            user_take = int(input("Please enter a valid enter how many you will 
take:"))
        pileSize = pileSize - user_take
        print("New Pile size:", pileSize)
        if pileSize == 1:
            print('You wins!')
            break


#When the user goes first
elif firstPlayer == 1:
    print("You go first")
    while True: #making sure pile is not less than 1
        user_take = int(input("enter how many you will take:"))
        while (user_take > pileSize // 2) or user_take <= 0:
            user_take = int(input("Please enter a valid enter how many you will 
take:"))
        pileSize = pileSize - user_take
        print("New Pile size:", pileSize)
        if pileSize == 1:
                print('You win!')
                break
        print("Computers turn")
        #print(computer_take(pileSize))
        pileSize = pileSize - computer_take(pileSize)
        print("New Pile size:", pileSize)
        if pileSize == 1:
            print('Computer wins!')
            break
```