

## Lab 3 - Nim

Amy Stamile & Richard McCormick

Due 9/27/19

### Lab Section 3

#### 1. Problem Statement

Create a Nim game that takes in a random pile size between 10 and 100. There are four possible scenarios of how the game will be played. First scenario is that the user starts the game and the computer is in “dumb” mode. Second scenario is that the computer starts the game and is also in “dumb” mode. Third scenario is that the user starts the game and the computer is in “smart” mode. The last scenario is that the computer starts the game and is also in “smart” mode.

#### 2. Planning

In our planning process, we laid out the shell of our code. We knew there were going to be two start options: the computer starting and the user starting. Within each scenario we indicated that there will be two modes for the computer set at random: dumb and smart. We initialized our variables at the beginning of our code. We knew we needed a random integer variable of 0 or 1 for the start mode and a random integer variable for the computer mode. We also created a pile variable which was a random integer between 10 and 100. Afterwards we started filling in the shell with the particular criteria for each situation that can possibly be presented.

#### 3. Implementing and Testing

In our testing process, we tested one portion of our code to make sure that part was running properly and that was in the situation that the computer was in dumb mode and that the user starting the game. Then we did the same testing for the computer in dumb mode and the computer starting the game. We used print statement to check that each section was properly being triggered, and to make sure that the random number generator was working correctly. Afterwards, we deleted these statements.

```
Command Prompt
C:\Users\Amy\Desktop>java Nim
Current number of marbles in pile: 94
How many marbles do you want to remove?: 45
Computer removes 17 marble(s)
Current number of marbles in pile: 32
How many marbles do you want to remove?: 15
Computer removes 3 marble(s)
Current number of marbles in pile: 14
How many marbles do you want to remove?: 7
Computer removes 2 marble(s)
Current number of marbles in pile: 5
How many marbles do you want to remove?: 2
Computer removes 1 marble(s)
Current number of marbles in pile: 2
How many marbles do you want to remove?: 1
The computer took the last one. You win!
C:\Users\Amy\Desktop>java Nim
Current number of marbles in pile: 28
How many marbles do you want to remove?: 15
Please enter a valid number: 13
Computer removes 5 marble(s)
Current number of marbles in pile: 10
How many marbles do you want to remove?: 5
Computer removes 2 marble(s)
Current number of marbles in pile: 3
How many marbles do you want to remove?: 1
Computer removes 1 marble(s)
The user took the last marble. Computer Wins!
C:\Users\Amy\Desktop>

//Smart Mode
else if (intelligence == 1){
    if (pile > 1){
        if (pile > 64){
            compChoice = pile - 63;
        }
        else if (pile > 32){
            compChoice = pile - 31;
        }
        else if (pile > 16){
            compChoice = pile - 15;
        }
        else if (pile > 8){
            compChoice = pile - 7;
        }
        else if (pile > 4){
            compChoice = pile - 3;
        }
    }
    else if (pile == 2){
        compChoice = 1;
    }
    else if (pile == 3){
        compChoice = 2;
    }
}
System.out.println("Computer removes " + compChoice + " marbles(s)");
pile -= compChoice;
if (pile == 1){
    System.out.print("The user took the last marble. Computer Wins!");
    pile = 0;
}
}
```

#### 4. Reflection

One major problem we encountered was that we originally used an online IDE to test our code. This was useful at first because it allowed us to rapidly test each iteration of our code, but unfortunately the webpage crashed and we lost a lot of work. This was a significant setback, but fortunately we were able to rewrite the lost code. From this experience, we learned to not write code in an online environment.