Let’s try to make our first game! You are going to create a variant of the ancient stick removing game called Nim; but, you are going to rig it so the computer wins all the time.

Structure of the game

* There are 13 sticks in the pile at the start of the game. (Actually with this game you can start with any number of sticks that is any multiple of 4 plus 1: i.e.**13** = **4**x3+**1**)
* Each player can pick up to 3 sticks each turn (They can pick 1 or 2 or 3 sticks during their turn)
* The player who picks up the last stick loses

**ASSIGNMENT – 3C level 4 or 3U level 3**  
Create a Nim game where the computer wins every time by:

* Have the User go first
* Next have the computer pick up a number of stick so that the turn total ends up with 4 sticks being taken.
  + For example:
    - The User picks 3 sticks, Computer picks 1 stick (note 3+1 =4)
    - User picks 2, Computer picks 2
    - User picks 1, Computer picks 3
* Be sure to present a running total of remaining sticks for the user
* Ensure User interface of the program is thorough and complete. The purpose, instructions, and program intent is clearly understood as the program is executed.
* Be sure to include descriptive comments in the program code using both single line comments and block quotes for the program header.
* You must put in a good effort and make sure all of the rubric criteria are met.

**Possible Extensions:**

* Add a variable number of sticks
* Allow user to choose to go first or 2nd
* Play again option (you will need to learn loops!)
* Have computer choose random number of sticks
* Allow the user to set a game win limit or be able to continue to play until a sentinel is input.

**Self Assessment:**  
When you are done, complete the self-assessment on the rubric below. Highlight where you think you performed on each task.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | Level 1 | Level 2 | Level 3 | Level 4 |
| **Knowledge:** Program compiles, runs, and calculates and displays appropriate values | Errors in program, does not run or displays inaccurate values | No errors when compiling/running, minor errors in display of values. Extensions not attempted. | Program runs and displays information correctly. Extensions are attempted but not complete | **Program runs and displays information correctly, including extensions** |
| **Application:**  Applies appropriate use of variables and commands to run efficiently. Displays results clearly. | Errors in user input, calculations or output | Minor errors only in use of variables and commands/ calculations. No extensions. User output somewhat clear. | Appropriate use of commands and variables, extensions are attempted but not complete. User output is clear. | **Variables and commands used correctly and efficiently, including extensions. User output very clearly formatted.** |
| **Thinking:**  Uses appropriate coding techniques, code is clearly and efficiently written and runs without errors. | Some errors, code attempts appropriate use of variables and user input/output. | Minimal errors, variables are used with some effectiveness to store user input and perform calculations. Results are displayed. | Code runs properly and uses variables effectively to store user input and make calculations, and displays results appropriately. | **Code is clearly written, understandable and efficiently calculates interest using appropriate variables and calculations.** |
| **Communication:**  Appropriate commenting on code to improve readability, reflection questions are answered in detail. | Limited or unclear commenting in code, reflection questions have little detail. | Some commenting in code, reflection has some thought and detail. | Code is fully commented and understandable. Appropriate reflection completed with details. | **Code is well commented and very understandable. Reflection answers show insight and critical self-evaluation.** |

Reflection Questions:

1. Play the game with a non-computer science person. (parent, friend from another class, sibling,...) and ask them to list at least three things they would like added to your game (list them here) and describe how you meet their conditions (if you were able to)
2. Describe any difficulties your non-programming user had with playing your game.

**Teacher Assessment:**  
Do not modify or write on this page – I will complete my assessment/comments here and return to you.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | Level 1 | Level 2 | Level 3 | Level 4 |
| **Knowledge:** Program compiles, runs, and calculates and displays appropriate values | Errors in program, does not run or displays inaccurate values | No errors when compiling/running, minor errors in display of values. Extensions not attempted. | Program runs and displays information correctly. Extensions are attempted but not complete | Program runs and displays information correctly, including extensions |
| **Application:**  Applies appropriate use of variables and commands to run efficiently. Displays results clearly. | Errors in user input, calculations or output | Minor errors only in use of variables and commands/ calculations. No extensions. User output somewhat clear. | Appropriate use of commands and variables, extensions are attempted but not complete. User output is clear. | Variables and commands used correctly and efficiently, including extensions. User output very clearly formatted. |
| **Thinking:**  Uses appropriate coding techniques, code is clearly and efficiently written and runs without errors. | Some errors, code attempts appropriate use of variables and user input/output. | Minimal errors, variables are used with some effectiveness to store user input and perform calculations. Results are displayed. | Code runs properly and uses variables effectively to store user input and make calculations, and displays results appropriately. | Code is clearly written, understandable and efficiently calculates interest using appropriate variables and calculations. |
| **Communication:**  Appropriate commenting on code to improve readability, reflection questions are answered in detail. | Limited or unclear commenting in code, reflection questions have little detail. | Some commenting in code, reflection has some thought and detail. | Code is fully commented and understandable. Appropriate reflection completed with details. | Code is well commented and very understandable. Reflection answers show insight and critical self-evaluation. |

Teacher comments: