**PROJECT SIMON**

Warning The following is a project based upon LabVIEW. Your mission, should you choose to accept it, will be fraught with perilous dangers and the requisite thigh-slapping hi-jinks. If you are up to the challenge, please continue.

Objective The purpose of this project is to create an application using LabVIEW that simulates the electronic game named Simon by Milton-Bradley (codename -> SIMON).

Rules Using your brainpower only, limited though it may be, you will implement a program that basically accomplishes the following: generate a series of “lights” (4 total) that a human (heretofore known in this contract as the PLAYER) must reproduce in exact sequence. For example, a green light blinks, the player must select the green light; then a green light blinks followed by a yellow light, the player must hit the green then yellow light in the proper order. Points will be distributed on a partial credit b. That’s it.

Design Criteria (aka point scheme): Each feature must be implemented within the “spirit” of SIMON. For example, you cannot implement a menu that has nothing to do with the game. Nice try, but no cigar. There are 25 possible features to include, read each one before beginning and then choose wisely depending upon the time limit imposed! Remember that if you rename cvi.exe on your competitors’ computers, you can win easily... Ask questions if you don’t understand a particular feature or if you are illiterate.

Basic functionality as described in rules, starting with 1 light (50 points)

Ability for player to vary the degree of difficulty (20 points)

Use of easy tab controls (5 points)

Use of menus (5 points)

Use of run-time popup menus (5 points)

Well-documented source code (10 points)

Incorporate GPIB as random number generator (20 points)

Incorporate DAQ accessory box LEDs to indicate current level achieved (20 points)

Display current level achieved (5 points)

Impose a time limit for each level (20 points)

Incorporate keyboard into the game, not text entry (10 points)

Allow player to end game (5 points)

Allow player to save and resume game for later time (25 points)

Incorporate password controls (10 points)

Implement a top 10 players list MS Word ActiveX (20 points)

Add multimedia effects, such as sound, clip art, video, etc. (10 points)

Make it a multi-player game, with independent tracking of scores (20 points)

Keep a running score of the game (10 points)

Provide rules of the game to the player (5 points)

Indicate to player that they made a mistake (5 points)

User interface design (20 points for best overall; 17 points next; and so on)

*Tiebreaker* : code is well-designed and efficient (25 points)

Player Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| **Scoring Category** | **Maximum Points** | **Player Points** |
| Basic Functionality | 50 points |  |
| Save and Resume | 25 points |  |
| Well-Designed Code | 25 points |  |
| Top 10 using ActiveX | 25 points |  |
| Top 10 Players List | 20 points |  |
| Vary Difficulty | 20 points |  |
| Time Limit for Levels | 20 points |  |
| Incorporate GPIB | 20 points |  |
| Incorporate DAQ LEDs | 20 points |  |
| Multi-player game | 20 points |  |
| Use of keyboard | 10 points |  |
| Use password controls | 10 points |  |
| Multimedia effects | 10 points |  |
| Documentation | 10 points |  |
| Running score | 10 points |  |
| Mistake prompt | 5 points |  |
| Use easy tab controls | 5 points |  |
| Use menus | 5 points |  |
| Use run-time popup menus | 5 points |  |
| Display current level | 5 points |  |
| Allow user to quit | 5 points |  |
| Provide rules | 5 points |  |
| User interface design | 20 pts 1st place; 17 pts 2nd place; 14 pts 3rd place, etc. |  |
| **TOTAL** | 350 |  |

**SKILL LEVELS**

275-350 Exalted Grandmaster (Master of all they survey); 200-275 Peon Designator

100-200 Privileged One (can ignore others in social situations); <100 LabVIEW Programmer