

Games

Notes prepared for a panel at [CS2 Workshop](#) at Denison University.

Games are an excellent way to teach a number of topics in CS in general, and CS 2 in particular. Most of us have used games in our classes. I'd like to justify this, and to throw out some ideas of projects that will remind you of some of your favorite assignments so we can add to this list.

Many games are well suited to programming problems. Games provide a well-understood set of rules with a rich level of complexity. If they aren't complex and interesting, then no one plays them. Project requirements flow naturally from the subject, rather than being imposed by the instructor. This makes it easier to write and read the requirements.

I will not defend the view that students learn valuable skills when playing video games. Students learn a great deal by bicycling across country, we don't cover that in class.

There is evidence that some students find games frivolous, and would rather be ending world hunger. I love to hear about activities that interest such students, but will continue to assign some game projects.

You may wish to give the student a starting point with many of these projects: sometimes the representation is complex enough to block the student.

Here is a short list of some CS2 topics, and games that give rise to projects.

- Logic
 - R. Smullyan, *What is the Name of this Book?*
Full of interesting engaging puzzles at variety of levels.
- Binary Search
 - Guessing Game - *I'm thinking of a number between 1 and 100*
- Recursion
 - Towers of Hanoi
- Three good manipulatives*
 - Spin Out
 - Chinese Rings
 - Matryoshka Dolls
- Backtracking
 - Chess
 - 8 queens
 - Knight's Tour
 - Suduko - also a great chance to talk about bit operations.
 - Mazes - and Spanning Trees provide a simple way to build mazes.
 - Magic Squares
- Linked Lists
 - Josephus Problem
 - Mastermind
- ADTs Queues, PQueues, and Table
 - Best First and A* Search

- Sliding Block Puzzles
 - Sam Lloyd's 15 puzzle
 - Rush Hour - comes with 40 different puzzles
 - Golf (From "work" to "play" changing one letter at a time)
- Trees
 - Alpha-Beta Pruning
 - Chess
 - Checkers
 - Connect 4 - Small enough to discuss solutions
 - French Military Game: small version of hare and hounds
- Sorting
 - Cards
 - Bridge
 - Poker - (ranking a poker hand demonstrates the importance of representation)
- Hashing
 - Revisit the Searching problems above, which include the ADT Table. Compare how different hashing schemes distribute the board positions.
- Other Excellent Game-based Projects
 - [Scratch](#) can be used by CS1 students in the first weeks to program simple interactive games. Sample games include tag, races, and mini-pacman. Scratch is not rich enough an environment for Data Structures
 - [Greenfoot](#) is a Java-based environment that gives a great starting point for simple animated games. Students extend predefined objects.
 - Game of Life (Is this really a game?)
 - Kuniaki Moribe's addictive *Same Game*. Given a 2-diminsional array of marbles, player can remove groups of two or more contiguous marbles of the same color. Rest of marbles shift down.
 - Simulating a move requires recursive fill and more
 - Fertile ground for ad-hoc strategies as well as backtracking
 - Video Games: Great inducement to learn
 - Graphics
 - Physics
 - Performance
 - Embedded Systems

Good video games are hard to write, and a time sink: The Black Hole of Liberty City.
I try to steer students away from careers in Game Programming.

Sources

- E. R. Berlekamp, J. H. Conway, and R. K. Guy, Winning Ways for Your Mathematical Plays, and other books by these authors.
- W. W. Rouse Ball and H. S. M. Coxeter, Mathematical Recreations and Essays
- R. Smullyan, What is the Name of this Book?