



10-2011

Rock-Paper-Scissors: Nifty Tools and Assignments

James Huggins

Follow this and additional works at: https://digitalcommons.kettering.edu/computerscience_conference



Part of the [Computer Sciences Commons](#), and the [Higher Education Commons](#)

ROCK-PAPER-SCISSORS*

NIFTY TOOLS AND ASSIGNMENTS

Jim Huggins

Associate Professor of Computer Science

Kettering University, Flint, MI

jhuggins@kettering.edu

In the childhood game of "rock-paper-scissors", two participants select one of three different gestures. The selected gestures are then compared using the following rule set: rock crushes scissors, scissors cuts paper, and paper covers rock. Ties are broken by repeating the game as needed. The game can be played until the first winner is determined, or in repeated rounds (e.g. best two out of three).

With only three gestures to choose from, and against a known opponent, the possibility of a tie on any given throw can be fairly high. In response, Sam Kass (with help from Karen Bryla) developed an extension of this game by adding two new gestures: "lizard" and "Spock". The full rule set may be expressed as follows:

- Rock crushes Lizard and crushes Scissors
- Paper disproves Spock and covers Rock
- Scissors decapitates Lizard and cuts Paper
- Lizard poisons Spock and eats Paper
- Spock smashes Scissors and vaporizes Rock

I used this game for an assignment in a systems programming course. Students implemented a stand-alone server application and a companion client application for playing the game. The server accepted requests from clients, paired them up into two-player games, and refereed each game. The server permitted multiple concurrent games. Students were given complete freedom in defining the network protocol used in exchanging information between server and clients, as well as the client user interface.

The assignment could easily be adapted for other courses. In CS1, students could write either the client or the server. In CS2, students could consider better ways to represent the decision engine (e.g. as a graph). In Discrete Math, students could analyze the expected number of games, or what kind of fair game extensions could be developed (e.g. a 9-choice game).

More detailed information may be viewed at

<http://www.kettering.edu/~jhuggins/Huggins-RPSLS/>

* Copyright is held by the author/owner.