

Test Plan

Test Cases

Execution

Input: `“./cardwars extra args”`

Expected Output: `“Please run the game without any additional arguments”` ; return from program

Output: matches expected

Number of Players

Input: `“0”` ; `“22”` ; `“”`

Expected Output: `“Invalid number of players, please try again with 2, 3, or 4”` ; return from program

Output: matches expected

Player Names

Input: P1: `“one”` and P2: `“one”` ; P1: `“one”` and P4: `“one”` ; P1: `“one”` and P3: `“one”` Expected Output: `“Players must have unique names”` ; return from program Output: matches expected

Input: P1: `“\n”` and P2: `“\n”`

Expected Output: `“Players must have unique names”` ; return from program Output: matches expected

Test Programs

Usage

- 1) Run `“Make”` from the terminal in the testing directory.
 - 2) From the testing directory, execute `“./memgrind 4 a2 a3 a4 a5”`
 - 3) Run `“make clean”` while in the directory `“testing”` to delete the executables.
-

card_grind_test.c

A modified version of Card Wars for testing purposes. See the Design and Analysis documentation for more information.

card_grind.c

This is the grind program which takes as arguments the information that would typically be user input. Those arguments are given to a function `test4` which calls `fork` and has the child process load the process image of `card_grind_test.c` with `execvp()`. `Test4` is called many times to test for memory error and to get an accurate value for the average amount of time it took per function call. This average is outputted in computer clock cycles.

Note that this only supports testing for four players, so 4 must be given along with 4 player names as arguments to the program.