Holesum Test Plan

170D WOBC: Module L Practical Exam II (A)

CW2 Kyle Spicer

Due Date: 22 November 2022

1 Purpose

The purpose of this test plan is to create robust unit testing and document manual tests conducted to ensure holesum program performs as intended.

2 Automated Tests

2.1 holesum_driver.c

- the holesum_driver program provides various matrices and tests to verify results reported from the holesum program. An ascii file is ran against the holesum program and tests the run time and results of how the holesum program and then scores it.

2.2 Adding "break" target to Makefile

- When running make break, holesum.c is compiled and ran against /dev/random > /dev/null in a for loop 20 times. This logic will throw random chars into the program in an attempt to verify that the program can handle a lot of invalid noise and performs as expected.

3 Manual Tests

INCORPORATING "ALL" AS COMMAND LINE ARGUMENT:
1. tested logic against all test files provided. Ensuring program exited successfully with improper input.

2. tested every file with valid file path and "all" as command line argument. Successfully printed size of each hole in accordance with extra credit.

TESTING HOLESUM PROGRAM

1. tested holesum program improper file path to ensure errors are handled appropriately. Program exits with informative message and example of correct implementation.

2. tested holesum program with all test files to ensure proper output was reached.

3. compiled holesum.c then ran holesum_driver.c to ensure all of the unit tests were successful. Achieving a 100% output.

MEMORY MANAGEMENT
1. After every test, checked program with valgrind to ensure all memory was allocated and freed properly, prior to program exit.

2. Ensured no memory leaks were possible and reported by valgrind.

RUNNING IMPROPER FILES
1. testnull - is an ascii file that contains no data. returns 0's for the two outputs.
2. test_no_zeros_or_ones - this ascii file is a 5x5 containing no 0's or 1's. The program doesn't crash and returns 0's for the output.

VERIFYING ABC's IN FUNCTIONS AND CODE 1. Revisited every function to verify there were checks in place for error handling and proper exit conditions.