# Test Plan - Maze

James Viner

### Sample Files

`basic_maze.txt`

A simple maze involving horizontal and vertical movement.

`empty.txt`

An empty file.

`invalid_symbol.txt`

A maze with a dollar sign, which is an invalid symbol.

`map00.txt`

A small, instructor-supplied sample maze.

`map01.tx`

A small, instructor-supplied sample maze.

`map02.txt`

A large, instructor-supplied sample maze.

`map03.txt`

A large, instructor-supplied sample maze.

**no_solution.txt**

A maze without a valid path from start to end.

**out_of_bound.txt**

A maze with a hole in the outer walls, allowing escape.

## Automated Test Cases

Automated tests may be executed with `make check`, which builds and runs a bash script that performs unit tests for this program. Additional test details can be found in `test/test.bash`; the script may have to be made executable before running.

# Common Test Cases

Prerequisites: `maze/` directory does not already exist.

## TC1: Installation

1. `git clone` git@git.umbc.tc:tdqc/tdqc12/jviner/maze.git

2. `cd maze`

**Expected:** `maze/` directory is created.

## TC2: Correct branch

1. git branch

**Expected:** main is default branch.

## TC3: Build cleaning

1. `make`

2. `make clean`

3. `ls`

**Expected:** No .o files or executable files are left in directory.

# Project-Building Test Cases

Prerequisites: In the project's directory; have run `make clean`.

### TC4: Build Explicit

1. `make maze`

2. `ls maze`

**Expected:** File is listed.

### TC5: Build Implicit

1. `make`

2. `ls maze`

**Expected:** File is listed.

### TC6: Build debugging

1. `make debug`

2. `readelf --debug-dump=decodedline maze`

**Expected:** Debugging info is listed.

### TC7: Build profiling

1. `make profile`

**Expected:** Program build output contains -gp flags.

# File-Based Test Cases

Prerequisites: In the project's directory; have run make; sample files in samp/ folder.

### TC8: Leak Prevention - Large sample file

1. `valgrind ./maze samp/map03.txt > /dev/null`

**Expected:** Program does not enter infinite loop, exits normally, and valgrind shows no leaks.

### TC9: Leak Prevention - Empty file

1. `valgrind ./maze samp/empty.txt`

**Expected:** valgrind shows no leaks. Nothing is printed.

### TC10: Leak Prevention - Nonexistent file

1. `valgrind ./maze samp/nonexistentfile.txt`

**Expected:** valgrind shows no leaks. Appropriate error message printed.