CMSC162 Intro to Algorithmic Design II Blaheta

Lab 3
Optional preview

5 September 2023

This week's lab continues our work on class design, and also works a little more on our maze solver project by building a model for the mazes. Because Labor Day is the day before the lab, doing this preview in advance of the lab is optional; if you are reading this file in the lab itself, just go ahead and jump right to the full lab handout.

If you want to get a jump on things, though, read the descriptions below and the code I've provided in `/home/shared/162/lab3/`. One file in that directory just reads in a maze and writes it back out again (with a little extra info); the others collectively illustrate how the same functions, with stream parameters, can read from a string in one context and from cin in another.

There are some things I do in those files that you might not have seen before; if so, make a list of places in `mazerw.cpp`, `strdemo.cpp`, or any other file in that directory, that I've used C++ features that you've never seen or aren't really sure how they work. Include line numbers. We'll talk about them in lab on Tuesday.

I expect to get the lab posted to the usual place (on cs.longwood) sometime Monday afternoon, or earlier.

The file format

Maze files look like this:

```
7 4
#######
#...#o#
#*#...#
#######
```

The first line contains two numbers (the width and height of the maze); subsequent lines contain a map of the maze itself, with each different type of maze content represented by a different character:

walls	`#`	(hash mark)
open spaces	`.`	(period)
start	`o`	(lowercase ‘O’)
finish	`*`	(asterisk)

Each maze will have exactly one start and exactly one finish; though note that not all open spaces need be reachable from the start, and the finish may also be unreachable.