

Minesweeper Solver Analysis

Jacob Slagle

January 13, 2019

Introduction

Minesweeper Solver is a personal project I started to practice my coding skills. Specifically I wanted to take a problem, develop an algorithm to solve it, and create a project around it with all of the trappings. The game minesweeper was suitably complex: a good, efficient algorithm would take a bit of thinking, but not so much that I couldn't also focus on code design and increasing my Python knowledge. I also like the poetry of taking something which I've used as a distraction in the past and using it to improve myself.

The Problem

Simply put, the problem is to solve a game of Minesweeper (I assume the reader is familiar with the mechanics of the game). In more detail, I seek an algorithm that, if possible, wins a game of Minesweeper without ever risking losing the game, i.e. never makes a move unless its a sure move. It suffices to look at a freeze frame of the game and determine with certainty which spaces must contain mines, and so should be flagged (right-clicked), and which spaces must not contain mines, and so should be revealed (left-clicked). Its unnecessary to find all such spaces, because an algorithm that can find at least one new space each time can be applied repeatedly.

What if there are no spaces that can be determined to be mined or free of mines (henceforth "free" for short)? Indeed, this is the case at the beginning of the game before any spaces have been revealed. However every game of minesweeper I've played, including the one I've coded, ensures that the first square selected is free as well as all surrounding squares. As it turns out, sometimes it's impossible to make any certain moves in the middle of the game either. In this case all we can do is guess, and I will discuss below how to make a good guess. For now though, we consider this: when possible, how do I determine which moves I can make with certainty.

Algorithms

Here I use the pronoun "we" as in formal mathematics.

As a starting point we consider what exactly it means for a move to be certain. Suppose we're looking at a (freeze) frame of a game in which some squares have been revealed. As a point of fact there is some way to place flags on the unrevealed squares such that each revealed square with a number in it has exactly that number of flags adjacent to it- namely,