Project Report on implementation of semi-automated Minesweeper

by

Arpit Toshniwal(J08026) and Abhash Kumar Singh (J08001)

under the guidance of

Dr. Gaurav Harit

in the course

Artifical Intelligence

INDIAN INSTITUTE OF TECHNOLOGY
RAJASTHAN
JODHPUR-342005, INDIA

April 17, 2011

Introduction

Minesweeper is a single-player video game. The object of the game is to clear an abstract minefield without detonating a mine.Contrary to the beliefs of some, Minesweeper cannot always be solved with 100% certainty, and may require the occasional use of probability to flag the square most likely to have a mine. In other words, one must sometimes guess to solve a minesweeper puzzle.

Project Proposal

Minesweeper is essentially the knowledge base entailment with the exception at first step.In this program we have implemented a **semi-automated *Minesweeper***.

It is semi-automated in the sense that, it accepts input from user whether to automate the game or to play it manually.User can also switch to *heuristic mode* where automation is not available but it can be used effectively when a chance game is there.This game is being played on konsole.

Another important aspect of this game is that it is fully customized as it asks user for the height and width of the board as well as number of mines.

Implementation Details

Some Important Definitions

We are begining with the ***status types*** of cells.

closed : Initial state, the contents of this cell are unknown.

opened : User has opened this cell

marked : User has marked this cell as containing a mine. The analyzer only uses this state when looking for mines.

invalid : Cell is invalid and not part of the playfield. We place a 1 cell border of invalid cells around the playfield to safe on a number of array bound checks.

There are ***set types*** regarding it in being a *superset* or *subset* of a neighbouring open cell.

super : If it is in super set of some neighbouring open cell.

sub : If it is in sub set of some neighbouring open cell.

both : If it is in super and sub set of some neighbouring open cell.

undetermined : This is the initial state at the begining of game.It leads to the updatation into other states.

Now there are two modes opening and marking whose uses can be understood from their name.We will discuss the rules later in the document.

Common Varialbles

gcells : It contains the number of valid cells left to open in the playboard.

gmines : It contains the number of mines present in the playboard.

Class play_cell : This class contains all the information regarding a single cell including both game information and analysis data.Game information is not visible to the user but user can get all the analysis data if he is carefull enough.The contents of this class are:

mine : If it is true, the cell contains mine.

neighbour_mine : Number of mines in the direct neighbour cells applicable only when the cell doesnot contain mine.

neighbour_mine_found : The number of mine found in the neighbourhood of the cell and visible to user.

unc : It is the set of closed neighbours of this open cell a subset,superset,or both of that of other open cells.

super_set : The sets which are a superset of the UNC of this open cell. These are 8 in number.

sub_set : The sets which are a subset of the UNC of this open cell. These are 21 in number.

sub_set_cntr : It counts subsets.

heur_numerator : numerators of all chances a mine is not in this closed cell.

heur_denominator : denominator of all chances a mine is not in this closed cell.

rating : Rating of this cell by the heuristic function.

constructor : It puts mines as false, neighbour mines as 0, rating as 9, neighbour_closed as 8, all the membership counters as 0 and set type as undetermined.

Important Functions

Neighbour : Helper function to walk through every neighbour of a given cell

1 0 7
2 * 6
3 4 5

AddSetType : Small helper function to add a set type to a playcell.

AddSet : Small helper function to add sets of a specific kind to a cell of the opposite kind .i.e, when a subset set is found, it will mark the corresponding cell 'super' and add itself to that super's sub_set array provided it doesn't already exist.

info : This function prints the details of the program.

help : This function prints help required by user to run this program.

rules : This functon prints hints required by user to play the game.

Clearscreen : This function clears the screen by inserting 40 new lines.

CreateBoard : This function creates the playboard as per user input and updates the cells information to the correct value.

PlaceMines : This function places mines randomly over the playboard.

PrintBoards : This prints the play board and the analysis board next to eachother on the Konsole.

PrintBoardsShowAll : This prints the play board showing the positions of all mines and neighbours.Used for testing the program.

UpdateNeighboursMarked : This function updates all neighbours for a given cell in the situation that this cell becomes 'marked'. Each Neighbour has one less closed neighbour now, and one extra mine has been found.

UpdateNeighboursClosed : This function updates all neighbours for a given cell in the situation that this cell becomes closed. This happens when a previous mine marking is removed. Each Neighbour has on more closed neighbour now, and one mine less has been found.

UpdateNeighboursOpened : This function updates all neighbours for a given cell in the situation that this cell is opened.

CheckSet : Checks the area around the current cell to see whether new sub- or super sets have been created.

CheckCell : This function checks if a rule applies to the given cell.It's only called to check opened cells, either when just opened or when something in their environment changes. It also takes care of mechanism when rules apply.

CheckNeighbours : This function is called after a cell has been changed in some way.We now check if the change has affected the environment such that a rule can be applied to the new situation.

OpenCell : Updates all effected cells and checks if rules apply whenever a cell is opened.

MakeCell : Updates all effected cells and checks if rules apply whenever a cell is marked.

CloseCell : Updates all effected cells and checks if rules apply whenever a cell is closed.

Rules

Following are the rules used for hint or automation.

ruleA :X neighbour mines, X found, so rest is safe.It has two conditions:

- all mines found,so all the closed neighbours are safe.
- if no. of mines is 0 then automatically open all the neighbouring cells.

The second point is always run while the first is run when autoA is true.

ruleB : Mines left equals unknown cells.So,all closed neighbours are mines.Based on this rule they are marked if autoB is true.

ruleC :All unknown neighbours of cell A are a subset of the unknown neighbours of cell B,outside subset is safe.Based on this rule the mines are opened if autoC is true.

Heuristics

In this mode we calculate rating of each cell.Higher the ratings higher is the proability of the cell to contain mine.Initailly all the cells have rating 9(the highest one) but it is updated everytime a cell is opened or marked in the CheckCell function.

Rating is calculate in following manner:

- for each closed cell in the neighbour-hood of an open cell calculate proability of having a mine as($\frac{um - mm}{mm}$) where um is number of unknown mines left and mm is number of mines marked.
- multiply these values obtained from every neighbourhood.
- multiply it with 9 to get the rating and cast it to nearest integer.