

Homework 6 | CS15, Fall 1999 | Due: 11.29 (in class)

- The game of xrobots is played on a rectangular grid of squares. At any time, each square is either unoccupied, occupied by a robot, occupied by the human player, or occupied by a pile of robotic debris. On each turn, the human player has three choices: (1) stay put; or (2) move to any of its eight adjacent squares (as long as that square is unoccupied); or (3) "teleport" to a random square. If the player is at the top/bottom or left/right part of the board, they can not move up/down or left/right. After the player moves, each robot moves to one of its adjacent squares so as to minimize that robot's horizontal and vertical distance to the player. If two or more robots move to the same square, or if a robot moves to a square containing debris, then all the robots moving to that square are destroyed, leaving behind debris. The player wins when all the robots have turned into debris. The player loses when they move next to a robot, or upon teleporting, are placed next to a robot. Your assignment is to implement the game xrobots. Allow the player to specify the size of the board and the number of robots on the board. A sample game is shown below (player=X, robot=r, debris=o, empty=.), legal moves are (-1=undo, 0=teleport, 1=up left, 2=up, 3=up right, 4=left, 5=stay put, 6=right, 7=down left, 8=down, 9=down right).

With respect to "undo", you should allow the player an unlimited number of undos. That is, you will need to keep track of all moves from the first to current so that the user can undo as many moves as they wish. If you implement the game board as a two-dimensional array, you can not simply store a series of boards (this is too costly in terms of memory). You will need to store a condensed version of the board. For example, the first board shown below can be represented with a collection of triples: (2,2,r), (6,2,X), (2,3,r), (4,6,r), (7,6,r), and (2,7,r).

Board size: 7 7
 Number of robots: 5

```

. . . . . . .
. r . . . X .
. r . . . . .
. . . . . . .
. . . . . . .
. . . r . . r
. r . . . . .
    
```

move1? 4

```

. . . . . . .
. . o . X . .
. . . . . . .
. . . . . . .
. . . . r r .
. . r . . . .
. . . . . . .
    
```

move2? 5

```

. . . . . . .
. . o . X . .
. . . . . . .
. . . . o . .
. . . r . . .
. . . . . . .
. . . . . . .
    
```

move3? 3

```

. . . . . . .
. . o . . . .
. . . . . X .
. . . . o . .
. . . . . . .
. . . . . . .
. . . . . . .
    
```

Win.