

'Robots' Game

You will implement an ASCII game called 'robots' that has the same rules as the GUI game robots that came with Unix BSD and is still around in many linux distributions. The game is played on a rectangular field. The player has to escape unharmed a bunch of robots that are programmed to kill the player. The robots move up and down, left and right, and diagonally, always in the direction of the player. The player moves within the rectangle up and down, left and right, and diagonally. The player escapes by manipulating robots to crash into each other. The result is a heap. If a robot walks into the heap, the robot is kaput and adds to the pile of heap. In dire straits, the player can teleport, which places the player at a random location in the rectangle. If that location is a robot or a heap, then the player dies. You can find more details by watching a game on youtube or by reading at wikipedia. Finally, a browser version is at the second link below.

[https://en.wikipedia.org/wiki/Robots_\(computer_game\)](https://en.wikipedia.org/wiki/Robots_(computer_game))
<https://arp242.net/robots/>

Implementation

You need to implement the game as an object. Its fields are the width and the height of the rectangle (30 by 60 makes a good field), the player, a list of robots, and a list of heaps. The game is turn-based. The human player gives a movement to her avatar by using one of the following commands:

<i>Input</i>	<i>Movement</i>
<i>i</i>	Up
<i>j</i>	Right
<i>l</i>	Left
<i>k</i>	down
<i>u</i>	up and left
<i>o</i>	up and right
<i>n</i>	down and left
<i>m</i>	down and right
<i>.</i>	does not move
<i>t</i>	teleport

Player, robots, and heaps are represented by two-dimensional coordinates, i.e. by a tuple of two integer values. In order to represent the playing field, you need to generate a list of 30 lists of 60 characters. The lists contain mostly a single white space, but have an "I" where the player is, an "r" where a robot is, and an "*" where a heap is. You print the playing field by printing out 30 strings obtained by joining a list of 60 characters.

The game starts by randomly generating the position of the player and a given number of robots. The player is guaranteed to not be on top of a robot. The game prints the rectangle with the position to the screen.

The game then waits for the input from the user and executes the command. A player cannot escape from the rectangle. If a keystroke would take the player out of the rectangle, then the player just stays where it is.

The game then calculates the new position of robots, after determining whether robots have crashed into each other or a heap. The robot movement is simple. If the player is above the robot, the robot moves up, if the player is to the left, the robot moves to the left, etc.

COSC 1010 – Group Project 3

The game is round-based. Each round consists of displaying the field with the positions of the characters. The game then asks the player for a move. The game then updates the field according to the rules. If the number of robots is down to zero or if the player has died, then the game ends.

Deliverables

This is a group project, so you will be working in groups of **two or three**. If you are not able to get into a group then let me know by no later than Nov 20, 2017 (Monday). You have to present your work (code) to me during my office hours **by Dec 8, 2017 (Friday)**. Also, you (one of the project partners) have to upload your code on D2L by the same date. Just uploading the code on D2L will not fetch you any credit for the project.