Matthew Harding 201220275 – Comp 281 2019 – Assignment 2

1081

The first part of this assignment required me to create a replica of Conway's "Game of Life"[1] in C. The first task for this problem was to create functions that take in the user input through scanf functions. In my initialiseGame function, I use scanf to take in the number of rows, columns and steps to simulate for the game and store them as integer variables. Inside the same function I then initialise two game boards, named board and tempBoard, using malloc to create dynamic '2d arrays'.

Next, I invoke the getBoard function that runs through a loop can casts the user input characters to the board 2D array. This is then followed by a call to my timeStep function which runs through a nested for loop to check the number of neighbours each cell contains, and then uses that number to generate the new cell value and store it in the temporary board. The temporary board was necessary to prevent the generated cells from affecting the outcome for the neighbour cells.

The time step function is then runs through a loop for the amount of times that were originally input. Finally, the program prints out the final board array to the terminal and frees the memory up.

The program was run and accepted by judge.

1084

Highway Lite is a simple highway simulation, program will take in a number of rows, columns and steps to take through command line, followed by a list of arrival times and lanes for the vehicles. To deal with the input I first used a scanf function to take three integers and cast them to their corresponding variables. To handle the pairs of arrival times and lanes I created a struct to simulate tuples, and then stored them as they arrived in a dynamic array using malloc. I then use the initialiseGame function to fill the 2D array with number of lanes X lane length empty character '.'.

After initialising the game, the main function runs for the given amount of timesteps, first dealing with the move phase of the cars currently on the board. To do this I created a nested for loop to move throughout the 2d array, moving from the end of each road backwards towards the 0^{th} index to avoid the situation of two cars next to each other writing over when being moved forward.

After the move phase has been completed the timestep function runs the arrivalPhase function, which takes in the array of tuples and checks the given arrival time against the current timestep – 1, as the problem illustrated the cars are displayed on the board on the timestep after they arrive.

All that is left to do after the program has simulated the given timesteps is to print the final 2D array and free up the memory.

The program was run and accepted by online Judge.

References

[1]Conways game of life - https://en.wikipedia.org/wiki/Conway's_Game_of_Life [6/3/2019][Date Accessed - 06/03/2019]