

EEM076 Lab1

Albin Boklund, Samuel Runmark Thunell

April 24, 2023

1 Forking where? and why?

We decided to fork at every point of the map that the road branches of into at least two roads. Using these points as the forking points enabled generalisation for our code. This is because we fork to every road on the map until we find the goal.

2 Joining and path calculation

We are going to have a lot of threads that have taken different paths but they will slowly but surely meet a end, either a dead-end or the goal. When a thread does this; it needs to join with the rest of the threads that are done.

To do this: Each time that we fork a thread, we will add the other threads to a shared ArrayList. When a forked thread reaches a dead-end or the goal. It will try to join with the other threads. Therefore we loop through the threads and join each one, if a path was returned: add it to a summed up path. Thus, the thread that reaches the goal will trace back to the beginning and we will return that complete path in the end.

Figure 1: *Red bomberman moved to the branching cell from the top, they created the sub-thread to the path under and the main thread will continue right*

3 Problems that occurred

We experienced a problem where the threads could fork to already visited cells. This was caused because we looped through the list of neighbours and not a list of unvisited neighbours. This was fixed by first checking if the neighbour is a valid unvisited neighbour. If it is we fork to it, if not we skip it.

We also had a problem where multiple threads were planning to move to the same cell and they would all succeed. The root to this problem was the non-atomic operation between checking the visited cells and adding to it. We fixed this by adding the cell directly in the if statement. The add operation will return a boolean if the cell already was in the list.

A small problem that did not have too much of an impact was that the first player we create does not add the start cell to the visited list. This means that we will have a extra player that just stands at the start cell. We fixed this by always marking the start cell as visited.