STUD - Jakub Dóka

Abstract

Game about spamming arrow keys quickly, with no mistakes.

DIrections

The Game

Directions

Assignment

Creating the game where player must go from current position to green square using only arrow keys while next move is always intuitive.

Blueprint

The game itself is simple but we still need lot of helping objects and composition. As I want to make range of things, the UI elements like buttons, bars and selections are also implemented. Thus, the project has two packages:

* Game
* Ui

The Game depends on the Ui and defines all game logic. Main objects of the game are Path and Player. Player’s role is to move along the Path. Path needs to expand in a way that does not confuse player.

Diagram

Description automatically generated

User guide

Once you download and unpack the project, you must double click on the ‘run.bat’. This runs the script that builds the game. ‘Main.jar’ should appear in the project directory but game should also be already running.

Once you launched the game, you should explore on your own. I tried to make UI simple and responsive. Good thing to mention is that game will access your file system to save some data like best score and your settings. Talking about settings, here you can change the colour theme of the game and difficulty.

Algorithms

Only interesting algorithm, I use, is split sort (SS). SS utilizes the fact that 1-element list is always sorted and merging of two sorted lists of lengths N and M has complexity O(N + M).

SS first splits the list to smaller lists until there is only one element in each and performs merging of newly created sorted lists. Each time we merge two lists we use them to merge again until we and up with list with original length. In the end, complexity is O(n log2 n) as we are always splitting to halves (log2 n) and we sort on each level (n).

If you look at my implementation of split sort, its rather unusual. First, it’s not recursive. Second, it does only one big allocation that is of same length as the inputted list. Here is some pseudo code that compares the implementations.

Text

Description automatically generated

Finally, why does the game need to sort? Well as you can notice, path is generated in a way that it does not overlap weirdly and does not get confusing. You can say it even creates interesting patterns. Each time the path expands, it must find the right places to do so. The path newer goes straight, it chooses the direction of expansion based of the free space. It sorts all segments by one axis and finds big enough spaces between them. Then it picks one randomly. If there is nothing to pick, path will expand to one of the edges. If there was no randomness in picking the positions, path would create pattern that would move at some direction and that’s not fun at all.

In-code documentation notes

Overrides are not documented as purpose of the method is already documented at its original declaration. One-line getters are also not documented to keep my sanity in takt. On the other hand, all other methods are documented. My comments usually include information that might not be expected when using the method, I refuse to write things obvious from signature of the method unless I have nothing else to write and documenting the actual implementation is also useless for the user, “code speaks for itself and newer lies”.