Sorting Algorithms in Unity3D

Buchbauer Benjamin, BSc

b.buchbauer@student.tugraz.at

February 18, 2017

Contents

1. Introduction
2. Sorting Algorithms
   1. QuickSort
   2. MergeSort
   3. HeapSort
   4. GnomeSort
   5. RadixSort
3. Visualization in Unity3D
   1. Functionality
   2. Swapping GameObjects
   3. Coding Window
   4. Snippets
4. Conclusion and Future Work
5. **Introduction**

In this study we want to visually outline sorting algorithms either one after another or several at the same time. Furthermore we want to write our own code in runtime to try to sort the elements and compare with different sorting algorithms. The platform we use is Unity3D. It is a cross-platform game engine and used to develop video games for PC, mobile devices, websites and consoles.

TODO: add something from project plan

The remainder of this report is organized as follows. Section 2 discusses several sorting algorithms we have implemented in this project. Visualization of our sorting algorithms is presented in Section3, which also includes some code snippets. Section 5 provides final conclusions and directions for future work.

1. **Sorting Algorithms**

In this section we will discuss the functionality and characteristics of several sorting algorithms which we want to visualize.

**2.1 QuickSort**

QuickSort is a quick, recursive and efficient 'divide and conquer' algorithm. However it is not stable, which means that the relative order of equal elements is not preserved.

In the beginning the list is divided into two sublists. In order to do this quicksort chooses a pivot element from the original list. All elements, which are smaller than the pivot element, are put into the left sublist and all the bigger elements are put into the right sublist. Equal elements can be put into either sublist. After that quicksort can start to recursively sort the sublists and reapply the above steps.