**Focus Group for Requirement Gathering of Sorting Algorithms Animation Software**

**Position: Project Room 14, Library**

**Time: November 12, 2020**

**Year 3 group**

Year 2 students’ discussion focuses more on correctness and provides other suggestions as well.

**The difficult part of learning sorting algorithms**

Our participants are confused about recursive steps of some sorting algorithms. Loop is also a difficult part, but demonstrations of each step would help. Participants all agree that animation would be a good way of understanding the sorting process.

**Problems in learning correctness**

Correctness is hard to understand at the very beginning. Participants explained that they did not expect that correctness would have a connection to mathematics and predicate logic. They hoped us to show animations of correctness logically and provide sufficient examples. However, it is difficult for year one student to understand in a logical way with predicate logic.

**Way to understand correctness easier**

It is suggested that providing stages letting users get used to the concept step by step would help. For example, termination and correct output for any legal input are two basic ideas of correctness. These two steps can be animated separately to illustrate fundamental ideas. One of the participants raised an idea that truth table could be included to show the mapping relationship between output and legal input. Moreover, people can understand a concept easier in an opposite way, i.e. providing counterexample of incorrect algorithms. It could be another idea showing users what correctness is.

**How to make it interesting**

We then talked about a module that allows users to build their own algorithms by dragging pseudo code blocks provided and the software will generate corresponding code later. One of the participances suggested that people can be attracted by cute and unexpected things. He suggested that an animated yellow duck character can be displayed at a corner of the software to denote whether the pseudo-code users build is correct. If it is incorrect, the duck would die. This duck can also be used as a listener to record users’ understanding of algorithms when users speak to it.