

Introducing programming to science students with interactive flowchart software and online automatic grading

Kowit Kittiwutthisakdi

School of Science, Walailak University, Nakhon Si Thammarat 80160, Thailand

email: tonjung@gmail.com

Overview

When students are learning to develop algorithms, they often spend more time dealing with issues of syntax than solving the problem. Additionally, the textual nature of most programming environments works against the learning style of the majority of students.[1,2] RAPTOR is a flowchart-based programming environment, designed specifically to help students visualize their algorithms and avoid syntactic baggage. RAPTOR programs are created visually and executed visually by tracing the execution through the flowchart. Required syntax is kept to a minimum. Students preferred using flowcharts to express their algorithms, and were more successful creating algorithms using RAPTOR than using a traditional language or writing flowcharts without RAPTOR. Additionally Thai online automatic grading was also used in the class.

Aims and objectives

- To improve students' understanding in algorithms using the flowchart-based programming and online automatic grading.

Activities

- In lab session, students program both in c language along with RAPTOR version that they can access from their laptops.
- Students do assignments on www.programming.in.th using RAPTOR as a aid tool.

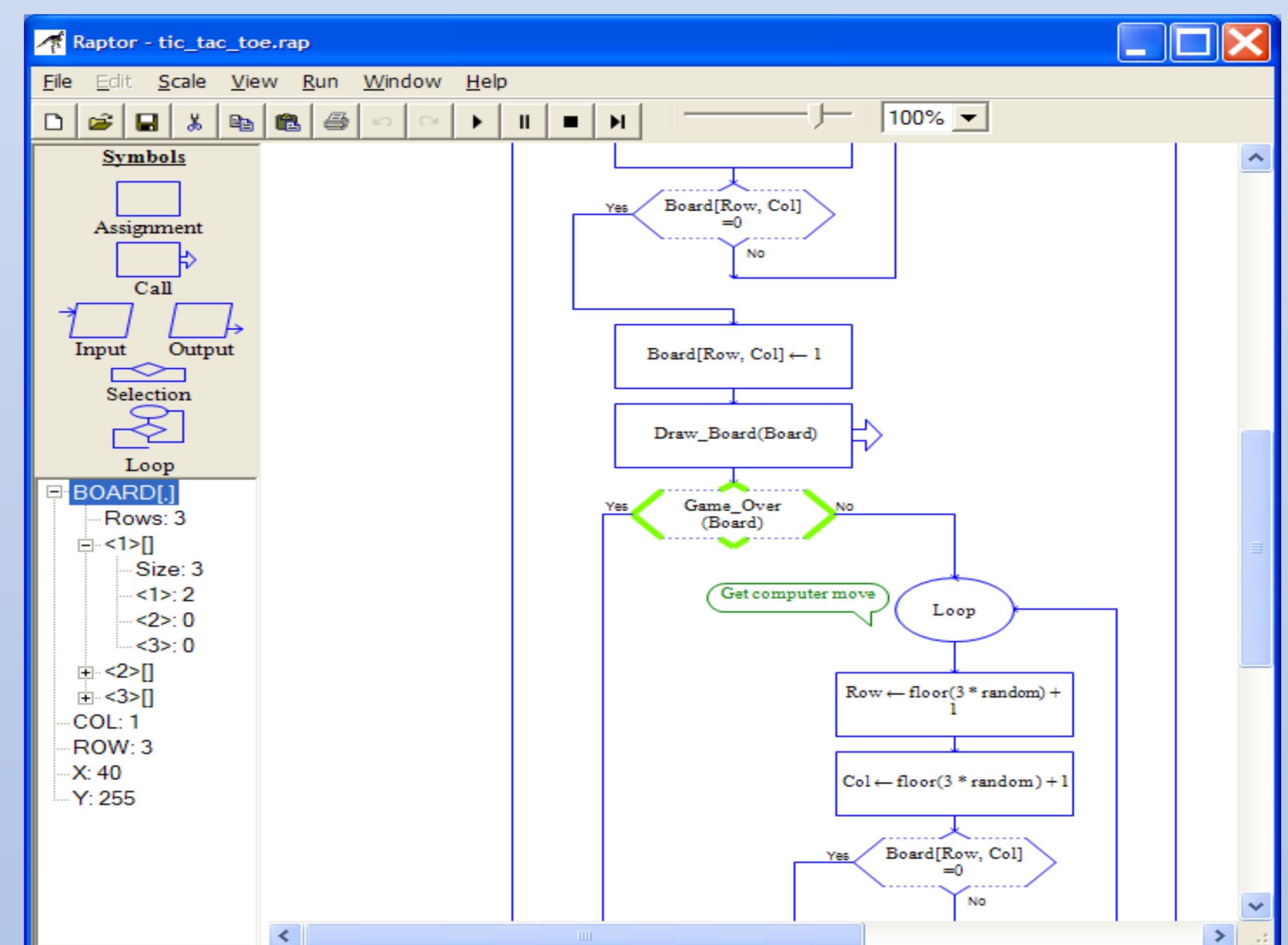


Figure 1: A RAPTOR flowchart in action.

DESCRIPTION OF RAPTOR

RAPTOR stands for Rapid Algorithmic Programming Tool for Ordered Reasoning, which is a free flowcharting software package running in the .NET Framework. When a new RAPTOR file is opened, a blank flowchart is formed with “Begin” and “End” symbols includes. User can drag either one of the six flowchart symbols / structure to the editing area each time:

- Assignment Symbol: For assigning variables
- Call: For calling sub-flowcharts
- Input Symbol
- Output Symbol
- Selection Structure
- Loop Structure (Iteration)



Figure 2: Some simple assignments

Old method: 4.14 New method: 4.83

Figure 3: Evaluation Score from students

References

- Cardellini, L. An Interview with Richard M. Felder. *Journal of Science Education* 3(2), (2002), 62-65.
- Thomas, L., Ratcliffe, M., Woodbury, J. and Jarman, E. Learning Styles and Performance in the Introductory Programming Sequence. *Proceedings of the 33rd SIGCSE Symposium* (March 2002), 33-42.