Creation of a Hybrid Programming Language

Authors

Dr. Aidan Mooney -Maynooth University

Mr. Mark Noone -Maynooth University

Ms. Emily O'Regan -Maynooth University

This presentation will look at the creation of a hybrid computer programming language combining the power of the traditional textual Java language with the visual features of the Snap! language. Textual programming languages (such as Java, Python and C++) are the main languages taught in Computer Science courses at third level, while visual programming languages (such as Scratch and Alice) are more prominently used in the education of younger students, mainly those students under sixteen. Some visual languages (such as ScratchJr and Snap!) have a reach to children as young as five. It has been well documented that there exists a gap in the education of students in their mid- to late-teenage years where perhaps visual languages are no longer suitable and textual languages may involve too steep of a learning curve.

There is an increasing need for languages that combine the powerfulness of a text-based language with the simplistic design of a visual language. These so-called hybrid programming languages would allow for the introduction of more complex programming concepts to students by having a more welcoming and more suitable interface. A need for a hybrid language is growing alongside the increasing interest among young people in the field of computer programming.

This project attempts to address this need. For the purpose of this project, the platform Snap! is utilised to create a hybrid language. Snap! is a visual programming language which employs 'blocks' to allow users to build programs. Snap! is also considered a platform and runs in the user's browser and presents an interface on which the user can program. Snap! was originally known as BYOB (Build Your Own Blocks) and was heavily influenced by the blocks-based visual language Scratch. Both Scratch and Snap! give the user access to libraries of pre-existing blocks with pre-set functionalities and allow the user to build programs using these blocks. The main additional feature that Snap! offers is the ability to create one's own blocks and extend the functionality of those blocks to create more complex and powerful programs.

The newly created hybrid language presented combines the textual programming language of Java with the visual 'drag-and-drop' programming language of Snap!. Snap!'s ability to allow the user build their own blocks supports the integration of Java into the platform. A 'drag-and-drop' interface is presented to the user, with each 'block' representing a corresponding concept in Java. Samples of concepts developed with this new language will be presented along with some of the main considerations and constraints involved. User experience and feedback was gathered from a subject pool of 174 first year Computer Science students in Maynooth University where these participants were given instructions to work with the hybrid programming language and provide a feedback to evaluate their experience using the language. These evaluations were analysed to understand the impact of the project and the main findings from this analysis will be presented. These along with future improvements to the language will also be presented.