Teacher Configurable Coding Challenges for Block Languages

Nath Tumlin
University of Alabama
Department of Computer Science
Tuscaloosa, AL 35487, USA
ngtumlin@crimson.ua.edu

ABSTRACT

COPPER (CustOmizable Puzzle Programming EnviRonment) is a meta-configurable tool for creating coding puzzles on a grid using a blocks-based programming language, similar to puzzles in Code.org's Hour of Code. COPPER has the potential to increase student interest and engagement by allowing a teacher to customize levels for individual classes or students. Teachers can create characters for specialized puzzles by uploading pictures to customize their appearance and using the block-language to design the character's behavior. They can then place these characters onto a grid, and arrange them into a puzzle for their students to solve. A teacher can specify the goal of each coding puzzle, as well as restrict which blocks a student may use, allowing a teacher to gradually introduce programming concepts. For example, an elementary school teacher could highlight concepts from a history lesson by building a customized grid where characters from a historical context navigate around objects relevant to the topic being studied. COPPER uses Google's Blockly framework to eliminate the mental overhead of memorizing textual syntax, allowing students to focus on building computational thinking skills. Block-based languages have been shown to be more effective than text-based languages when teaching programming to first-learners. Combined with customization, COPPER has the potential to lead to higher student interest and comprehension of programming concepts in a customized context. This poster will also summarize results obtained through initial experimentation through collaboration with K-8 teachers and their students.

Keywords

Blockly; Block-Based Programming; Customizable Puzzles

1. INTRODUCTION

A popular way of encouraging students to practice computational thinking is to embed tasks that improve these skills inside of a game-like environment, usually having students create programs that dictate the actions of a character as it advances through a coding puzzle level. This approach is used by Code.org in puzzles that are used in their Hour of Code and Code Studio environments [1]. These puzzles do not allow any sort of customization. This project seeks to increase the usefulness of these types of puzzles by allowing teachers to customize the visuals and mechanics of levels that address the specific needs of their own classroom. For example, a coding puzzle could be based on a history lesson about Christopher Columbus, where the student would write code to control a ship, which must navigate across the "ocean" and around waves to reach land. History questions could be mapped to specific events that the navigation encounters within the puzzle context. The following levels could involve the player programmatically planting crops, and so on, through various stories in American history. In this way, the student can both practice computational thinking and reinforce their lessons in other subjects, thus integrating computer science as a fun puzzle solving activity across other classes.

SIGCSE '17, March 08-11, 2017, Seattle, WA, USA
ACM 978-1-4503-4698-6/17/03.
http://dx.doi.org/10.1145/3017680.3022467

2. BLOCKLY AND BLOCK LANGUAGES

Blockly [2] is a library created by Google for implementing block-based programming languages, which can then be compiled into a traditional text-based language such as JavaScript or Python for execution (the user of the block language is not aware of the translation). Blockly provides a library of common blocks, such as control statements and mathematical operations, and allows for the creation of custom blocks for application-specific purposes.

Block-based languages have many advantages over text-based languages for students learning how to program. Many students find that the syntax of text-based languages is confusing, and prefer block-based alternatives [4]. Three barriers that beginning programmers encounter are selection, use, and coordination [3]. Block languages help students overcome these barriers by allowing them to recognize blocks instead of recalling syntax, breaking code into logically separate chunks, and preventing blocks from being assembled into invalid configurations.

3. COPPER

COPPER goes beyond existing coding puzzles by allowing in-depth customization of coding puzzle levels. All levels consist of a grid filled with multiple characters, and the main character. There are three main aspects to COPPER: a level creator, a character creator, and a separate generated environment for students to solve the created levels. The level creator allows teachers to define things such as the dimensions of a level, its appearance, which characters appear in the level, the goal of the level, and which blocks are available to students solving the level. The character creator allows teachers to design characters that can be placed in levels. The teacher can program the behavior of a character using the same block-based language students use to control their player, and teachers can set the appearance of a character by uploading an image. After creating a new character, a teacher can choose to upload that character for use by other teachers. COPPER also has some preexisting characters. Between the assets provided with COPPER and the ability to share among

teachers, an individual teacher does not need to know how to program to allow their students to use COPPER. The final part of COPPER is an environment for students to solve the levels created by their teacher, which is similar in appearance to the Hour of Code puzzles offered by Code.org. These levels can be assigned to student accounts from within a teacher account, and the teacher can see which levels each student has completed.

4. CURRENT STATUS

Currently, COPPER exists as a web application, though it is not publically hosted (official release is planned for late Spring 2017). COPPER's character creator (Figure 1) allows the teacher to use the block language to design and test the behavior of the character, set its appearance either through color or by uploading an image, and set other properties such as whether the player can pass through it, or what sounds should be played when characters interact with obstacles or other characters. After the character's design in complete, the JavaScript code for the character is generated, and the teacher is given the opportunity to save and upload the code for use in their levels. If uploaded publically, other teachers may find, view, modify, and use the character themselves.

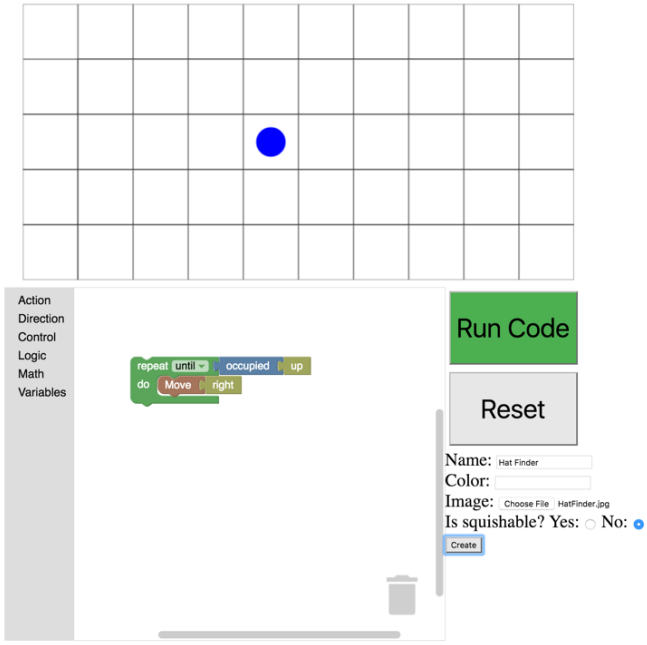

Figure 1. COPPER's character creator

After a teacher has all the characters they need, they can design levels by setting their dimensions and placing characters on the grid in their desired configuration. Finally, teachers are also able to set which blocks the student has access to. This allows the teacher to gradually introduce concepts over time, or to constrain the student and force them to think about the various ways to solve a problem. When a teacher is satisfied with the level, they can save it and assign it to student accounts under their own account, or simply leave it on their account, and open the level from their account on a school computer for a student to use.

Experimental evaluation of the approach is being conducted at present, and initial results will be reported on the poster.

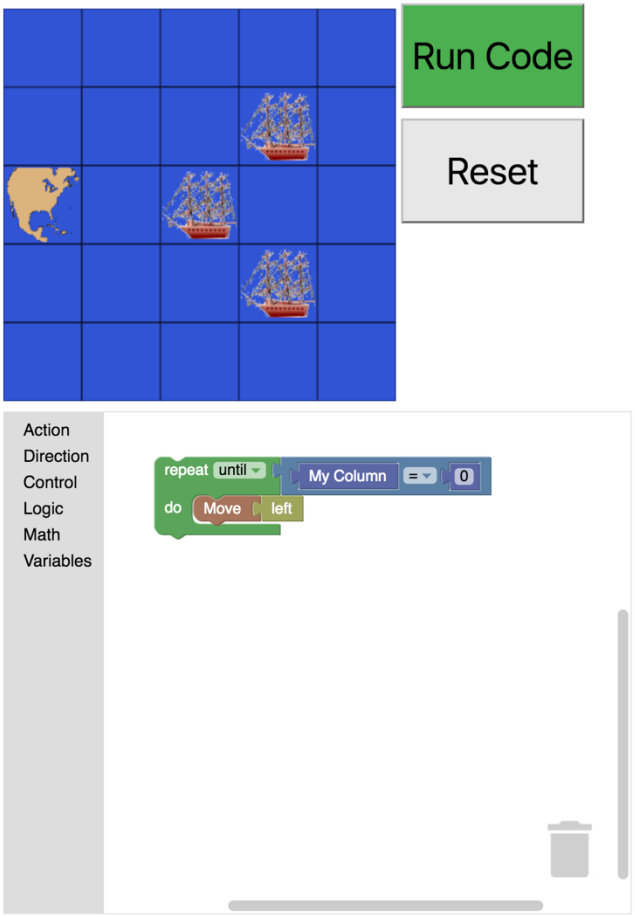

Figure 2. A simple example configuration, representing ships crossing the Atlantic Ocean

5. REFERENCES

[1] Code.org - https://studio.code.org/

[2] Blockly Overview - https://developers.google.com/blockly/guides/overview

[3] A. J. Ko, B. A. Myers, and H. H. Aung. Six learning barriers in end-user programming systems. In IEEE Symposium on Visual Languages and Human Centric Computing, VL/HCC '04, pages 199–206, 2004.

[4] D. Weintrop and U. Wilensky. To block or not to block, that is the question: students' perceptions of blocks-based programming. In 14th International Conference on Interaction Design and Children, pages 199–208, 2015.