C2STEM Teacher Dashboard

Caitlin Snyder

1 Introduction

Computation is an important component of 21st century education (Wing, 2006; Wing & Stanzione, 2016). Advancements are leading to the introduction of computational tools into Science, Technology, Engineering, and Mathematics (STEM) curricula and classrooms. It has been shown that computational model building is effective in promoting science learning (Basu, et at., 2017). Although the positive benefits of synergistic learning environments that promote learning in STEM and CT concepts (Basu, et al., 2016) are well documented, it is not well understood how teachers can work with such systems to encourage students' learning gains.

In this paper, a visualization tool is described that takes on the challenge of investigating ways a teacher can view students' progress and identify when intervention is necessary. The paper is organized as follows. A description of the C2STEM system is presented followed by a description of data collected. The visualization components of the dashboard visualization are discussed in detail along with possible user interactions. Finally, the paper concludes with possible future steps.

2 C2STEM

2.1 System Description

Our work focuses on C2STEM, a Collaborative, Computational STEM learning environment. It is a block-based computational modeling environment that incorporates Netsblox [6], an extension of Snap! (http://snap.berkeley.edu/). A domain specific modeling language (DSML) is combined with physics constructs to help students create dynamic (simulation) models. Students initialize variables under a *Green Flag* and then update variables in a *Simulation Step* block to capture dynamic behaviors at each time step. To evaluate this behavior the system allows for the use of animation and data tools (a table and graph) that are updated each simulation step. An example student-facing project for the 2D Velocity Challenge, including example physics DSML blocks (left-column) are pictured in Figure 1.

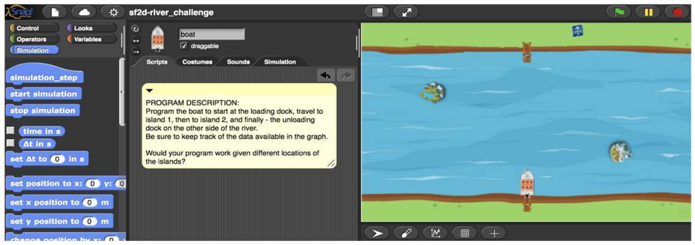

Fig.1. 2D Constant Velocity Task

2.2 Data

The C2STEM environment collects a variety of information about the students' progress as they work through modules to solve problems. Data sources for analysis include logs of student actions in the system, final student projects for each task in the system, post and pre test scores, embedded assessments and video data in some cases. Logs of the student actions included all possible movements the student could make in the system, including moving blocks and switching between tasks. In previous work, a task model was developed and used to abstract higher level actions from these logs (Hutchins, et al. 2018). These higher-level actions are as follows:

1. *Model*. Actions where the students are moving blocks, either onto the stage, off the stage or around the stage (Note that the stage is grey section next to the simulation window and represents which code will be run).
2. *Create*. Actions where the student is creating their own blocks
3. *Play*. Actions where the student is playing the simulation, either by clicking the green flag or double clicking on a set of blocks on the stage
4. *Stop*. An action where the student is stopping the simulation while it is running.
5. *Step*. An action where the student steps through their code line by line
6. *Change*. Actions where students are changing blocks on the stage by editing parameters or conditional statements

Final student projects include the submitted code blocks for each student. Post and pre tests are given to the students after and before modules respectively. Embedded assessments are given to the students while they work through the different tasks for each module. Finally, videos record students dialogue and screen as they build computational models. In this paper we look at scores (test and assessment) and log data.

3 Visualization

In order to support teacher involvement and intervention in student computational model building, a dashboard was created focusing on student success and progress in the system (using test and assessment scores) along with an ability to compare student actions. The dashboard consists of three interactive and linked views as seen in Figure 2 below.

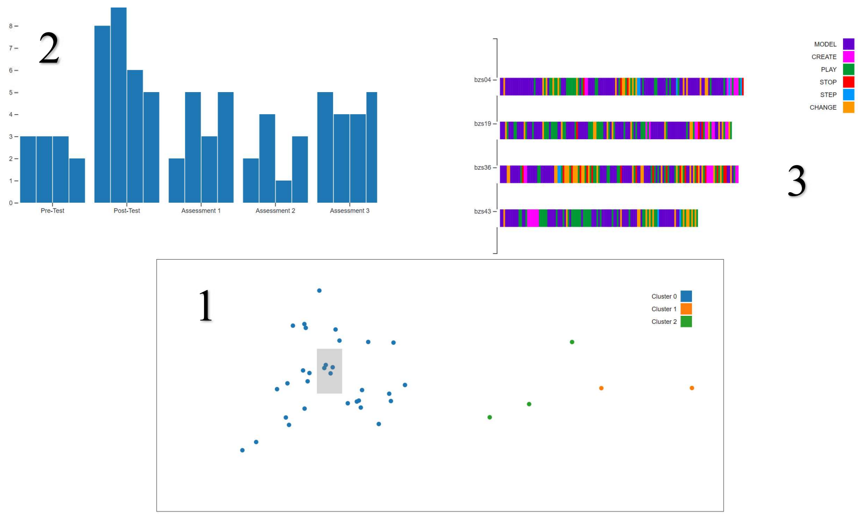

Fig.2. Teacher Dashboard

3.1 Cluster Visualization (1)

One goal of this visualization is to help teachers decide which students to look at as they work on the system. This is achieved using a k-means clustering model that identifies similar students using features based on their actions (specifically the proportion of the different types of actions in the model building environment). The clustering result is visually presented to the teacher using PCA. Each student is represented by a colored dot whose name can be seen by hovering (Note that the names are anonymized in the demo). The color channel represents the individual clusters. The selection of the students through brushing populates the other two views with data.

3.2 Scores Visualization (2)

While the clustering component helps teachers decide which students to compare, the score visualization allows teachers to compare students' progress through test and assessment scores. The final score for each is mapped to the y-channel while each individual bar represents one of the students selected in the cluster visualization. These bars are grouped together on the x-channel by assessment type.

3.3 Action Visualization (3)

Along with viewing student progress, teachers are able to compare actions taken in the system with the final component of the dashboard. Similar to the scores visualization, it is populated by the selection of students in the clustering component. Each type of higher-level action is mapped to a specific color, with the individual students mapped to the y-channel. The order of the sequence is mapped to the x-channel. In order to allow for the reduction of clutter and the narrowing of focus, the teacher is able to select each action (by clicking on the corresponding box in the legend). The visualization is updated by removing the other actions, as seen in Figure 3, so that the teacher can directly compare students based on a single action type.

Fig.3. Updated Action Visualization

4 Conclusion and Future Directions

This dashboard visualization is narrowed down to a focus on actions and student progress in a single model building task. Teachers are able to identify patterns in student actions that may lead to either positive or negative learning gains. Some of these patterns include the continually clicking of play, which may represent frustration on the students part or a hope that their simulation will magically fix itself between play actions, and the pattern of playing, changing and playing again which may represent a type of debugging strategy. Future work on this dashboard should include the inclusion of multiple model building tasks and the ability to compare one students actions throughout the tasks, instead of just comparing against other students,

References

1. Basu, S., Biswas, G., & Kinnebrew, J. S. (2017). Learner modeling for adaptive scaffolding in a computational thinking-based science learning environment. *User Modeling and User-Adapted Interaction*, *27*(1), 5-53.
2. Basu, S., Biswas, G. & Kinnebrew, J.S. (2016). Using multiple representations to simultaneously learn computational thinking and middle school science. In Thirtieth AAAI conference on Artificial Intelligence (pp. 3705-3711). Phoenix, Arizona, USA.

3. Hutchins, N., Biswas, G., Maroti, M., Ledezci, A., & Broll, B. (2018). A Design-Based Approach to a Classroom-Centered OELE. In International Conference on Artificial Intel-ligence in Education (pp. 155-159). Springer, Cham.
4. Wing, J. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.
5. Wing, J. M., & Stanzione, D. (2016). Progress in computational thinking, and expanding the HPC community. *Communications of the ACM*, *59*(7), 10-11.