COMPUTING LEARNING ACQUISITION?

Vicki Bennett
Department of Communication
University of Colorado, Boulder
Boulder, CO 80309
Vicki.bennett@colorado.edu

KyuHan Koh
Department of Computer Science
University of Colorado, Boulder
Boulder, CO 80309
Kohkh@colorado.edu

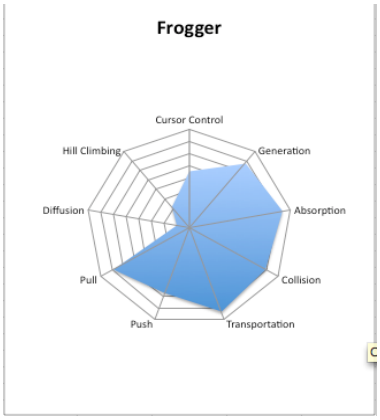
Alexander Repenning
Department of Computer Science
University of Colorado, Boulder
Boulder, CO 80309
ralex@cs.colorado.edu

1. INTRODUCTION. How can learning be computed? Curriculum, using visual language as the motivational context with embedded computer science content was utilized in one college computer science class and two middle school technology classes. From the data collected in these three classes over the course of a semester, associated learning progressions were computed from several computational thinking patterns. By comparing the results (learning progressions), some obvious and some not so obvious indications emerged. We believe that the more obvious indications give credence to the less obvious, and hence the measurement tool. This comparison between middle school students and college students demonstrated that middle school students' learning progressions are slower than college level students (obvious), but middle school students' learning skill scores reached/surpassed the entry level scores of students in the college class after designing only two or more games. Consequently, these results tell us that this measurement tool although not validated yet, could measure accumulated learning in certain contexts. Although this is just the first "outing" of this tool these results strongly indicate that the tool provides an accurate measure of the concept, in this case, learning accumulation or transfer.

2. METHODOLOGY**.** The Scalable Game Design (SGD) project [1] was funded by the National Science Foundation (NSF) to improve computer science interest in middle schools. The goal of SGD is to use visual language programming software (Agentsheets) [2] to introduce a more positive image of computer science through a game design course. The SGD curriculum specifies the design of several computer games and the subsequent uploading of those games to an online cyber-learning infrastructure called the Scalable Game Design Arcade (SGDA) [1]. To better understand the difference between standard educational presentation and the presentation of the SGD curriculum, analyzing the product of the curriculum (games) was undertaken. A new type of measurement tool was needed to facilitate these analyses, the Computational Thinking Pattern Analysis Graph (CTPAG). After uploading, submissions are evaluated using the CTPAG [3]. The graph visually depicts the analysis of each game or set of games and this analysis becomes part of the collected project data.

Computational thinking [4] and computational thinking patterns are the basis for the CTPAG. These are agent interaction sequences that show up in other programming contexts and other disciplines (i.e. science and mathematics), such as "collision" "generate" "absorb". The CTPAG analyzes and visualizes the semantic meaning and computational thinking patterns of the submitted games by semantically showing how close the specific game is to the tutorial and whether or not those patterns are successfully implemented. Each computational thinking pattern that is implemented in the given game is then depicted in the CTPAG (Fig. 1). Figure 1 is the CTPAG depiction of a Frogger game. The computational thinking patterns represented in this game are cursor control, generation, absorption, collision, transportation, push and pull, and a small amount of hill climbing. This is how a standard Frogger game should look as represented with the CTPAG.

Figure 1- CTPAG of the game, Frogger.



From the CTPAG, a learning skill score is computed which represents the student's ability to use the acquired knowledge from previous games to solve programming problems in the present game design project.

Learning Skill Score is calculated by the following equation:

$$skill(m) = \frac{\sqrt{\sum_{n}^{i=1} \left[\max\left(p_{i,j}\right)\right]^2}}{\sqrt{m}}$$

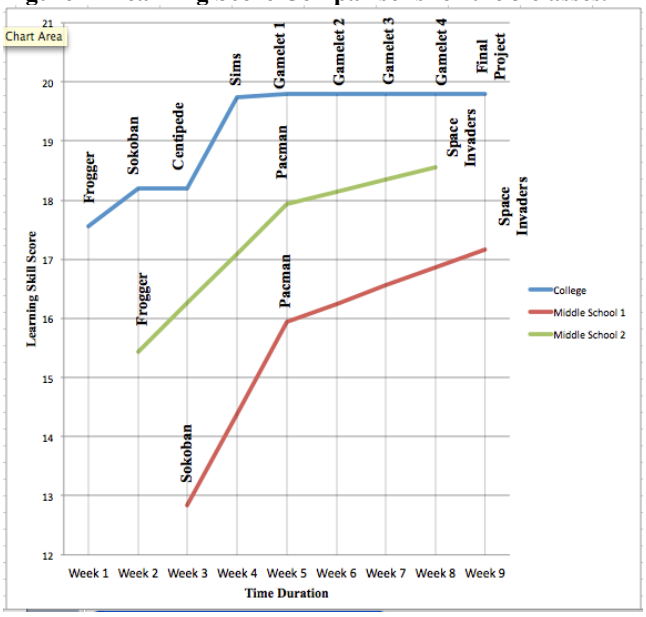
In this equation, P is a computational thinking pattern, n is the number of computational thinking patterns on the CTPAG, which is equal to 9 in this case, and m is the number of submissions. Using the learning skill scores from our 3 classes, a comparison graph was created (Fig. 2).

3. LEARNING SKILL SCORES COMPARISON. We chose three sample classes, one undergrad/grad level class and two

middle school (MS) classes for this study. In one academic semester, we collected and analyzed 268 games and simulations from 30 college students and 73 games from 33 middle school students. From these we computed the average learning skill scores of each game/simulation for the entire class submission to compute the learning progressions for each class. The comparison in learning-progressions is presented in Fig 2.

Both the MS and college students are exposed to the curriculum content and have the opportunity to strengthen their learning by doing it themselves. Consequently, these students are more likely to increase their knowledge and understanding of programming and game design with each new project. Which further increases the retention potential for increasing their learning skill scores over time (Fig 2). Since each of the games have patterns specifically needed to complete it, the learning skill scores can increase with the newly learned games. MS and college students continue to learn each new game's computational thinking patterns as they progress through the curriculum. All the students' learning skill score progressions are illustrated in Fig 2. The top blue line from Frogger to Final Project represents the college class, the middle green line from Frogger to Space invaders represents middle school 1, and the bottom red line from Sokoban to Space Invaders represents middle school 2. In figure 2, students from both middle schools have reached the entry level learning skills of the college class. The learning skill score of the students from middle school 1 after their second game submission reached 0.6, which is over 0.59, the entry-level score of the college class. The learning skill score of the students from middle school 2 did not overlap the college class entry level, but after the third game submission was 0.58, within .01 of the college entry score.

Figure 2- Learning Score Comparisons for the 3 classes.



Since one would expect that the middle school students' progression in understanding and utilizing the game authoring software would be slower than the college students, and this is exactly what the above graph, based on our calculations show. Instead of stressing the obvious, the analyzed data, points to the legitimacy of this way of calculating learning acquisition and hence the learning progressions, shown here. The obvious is represented here to offset other aspects shown in this graph, that middle school students can attain a level of understanding that can prepare them for entering a college level course, intellectually. But since college classes usually progress at an accelerated rate in comparison to MS classes, the average MS student would not normally be able to keep up, especially after only one programming class. The indication of the MS students achieving a college level of understanding after only one programming class is important to the research on this measurement tool, as these results could be substantiated in other ways. For instance, an exam could be given to both the beginning college level students and the MS students after the conclusion of the SGD curriculum to validate this outcome.

4. We believe that this computing tool could be used to determine knowledge acquisition or transfer of educational content in some contexts. The learning progression graph (Fig. 2) offers some interesting visual comparisons, but the most significant result is the most obvious. The timeframe for the middle school students is over three times as long as that for the college students as one would expect to find. So, since this result is obviously accurate, this tool at least measured this aspect accurately. Although this does not in any way determine the total accuracy of the tool for other aspects, it does indicate that the possibility of that accuracy, exists.

More research to determine validity and reliability must be done before any claims can legitimately be made for the results collected from the use of this tool. However, having the use of a tool like this one could shed some new and possibly important light on previous assumptions about the value and scope of enjoyable/visual language curriculum lessons, especially in those areas that are linked to computer science education. Plus, the ability to compute learning progressions with the CTPA graph could be a considerable step forward in showing the worth of game/ simulation-programming courses and the associated visual language. We believe that continuing our research on this tool could further the knowledge on a potentially important addition to the assessment of learning acquisition in computer science and possibly other fields.

5. ACKNOWLEDGMENTS

This material is based in part upon work supported by the National Science Foundation under Grant Numbers No. 0833612 and DMI-0712571. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

6. REFERENCES

[1] V. Bennett, K.H. Koh, & A. Repenning, "CS Education Re-Kindles Creativity in Public Schools," To be presented at ITiCSE'11, Darmstadt, Germany, June 27-29, 2011.

[2] A. Repenning, "AgentSheets®: an Interactive SimJulation Environment with End-User Programmable Agents," *In Proc. Interaction 2000*, Tokyo, Japan, 2000.

[3] K. Koh, A. Basawapatna, V. Bennett, A. Repenning, "Towards the Automatic Recognition of Computational Thinking for Adaptive Visual Language Learning", IEEE International Symposium on Visual