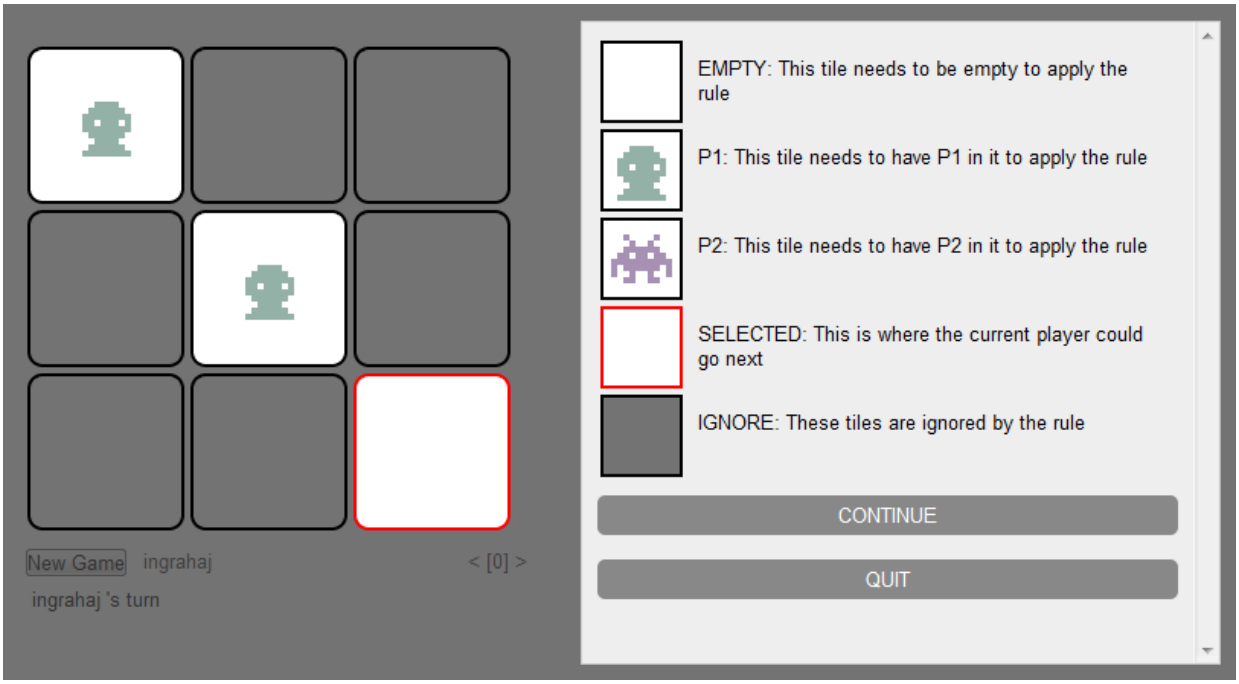
Rule Creation in CTArcade: Teaching Abstract Computational Thinking From Concrete Guidelines

John Ingraham

The work described below has been used in the context of a larger joint project, *CTArcade: Learning Computational Thinking While Training Virtual Characters Through Game Play* . The URL for this project is: http://ahnjune.com/wp-content/uploads/2011/06/ctarcade-chi2012-final.pdf

CTArcade is a web application framework that seeks to engage users through game play and improve their computational thinking (CT) skills. With CTArcade, we seek to provide a tool that addresses the lack of development of computational thinking skills present in our education system. Many students find that introductory programming courses are challenging, leading to high dropout rates [5]. Our approach attempts to teach computational thinking to younger audiences through more appealing media such as games. Specifically, CTArcade currently allows its users to explore the concepts and strategies of Tic Tac Toe. However, in the interest of teaching its users abstract computational thinking, CTArcade attempts to follow a design moving from concrete concepts to abstract ideas as users discover more advanced functions. The hope is that the concrete treatment of Tic Tac Toe will "provide a scaffold on which an understanding of more abstract relations can be built" in game strategy and, more generally, computational thinking [6]. The levels of abstraction that are built into CTArcade are designed with this in mind, utilizing *concreteness fading* [2] to gradually introduce abstract concepts to users. The first level simply entails playing Tic Tac Toe alone or against a basic computer player to let users develop an understanding of the game and its rules. The second involves the user's customization of their virtual character's behavior by selecting and permuting some premade strategies. The third, and final, level of abstraction is rule creation, which allows the user to design new rules through a flexible framework and add them to their character's strategy. By allowing the users to define their own strategy and test by both playing against it and pitting it against other users' characters, the user is employing and improving upon his computational thinking skills.

My contribution revolves around the rule creation functionality included in CTArcade(fig. 1 below). As the interface is aimed at a younger audience, it was designed with several levels of interaction, each more advanced and abstract than the last. Rule creation is the most abstract level of interaction that is provided to the user, allowing them to go beyond merely using and permuting the premade strategies. The goal was to create a graphical representation of rules that was intuitive enough to require no more explanation than a few simple examples (the premade rules). This representation ended up being a gameboard with all but a few spaces grayed out and the rest each with a symbol representing a condition on the board. Symbols in the grid displayed conditions on the spaces that had to be met before the rule could be applied; there was one each for a space being empty, occupied by the player, occupied by the opponent, and selected as the place to play if the rule was applied. The basic motivation behind this representation of a rule in a strategy was the perceived similarity to actually making a play in tic tac toe. The player simply looks at the board and, based on where he and the opponent have played, chooses where to go next. In addition to aiding in the development of an intuitive, graphical representation of rules in Tic Tac Toe, I helped to design and implement two ways to integrate rule creation into the natural workflow of CTArcade. The user could choose to create a rule at any time, but also had the option to review a game they had already played. As the user looked back over his moves, the interface would ask he thought he should have moved differently, in which case he could opt to create a rule based on the current state of the Tic Tac Toe board. Merging rule creation into the normal flow of a game in this way reduces the cognitive load of the user, because ideas for rules come up naturally during play and are much harder to create or conceptualize with no current context.



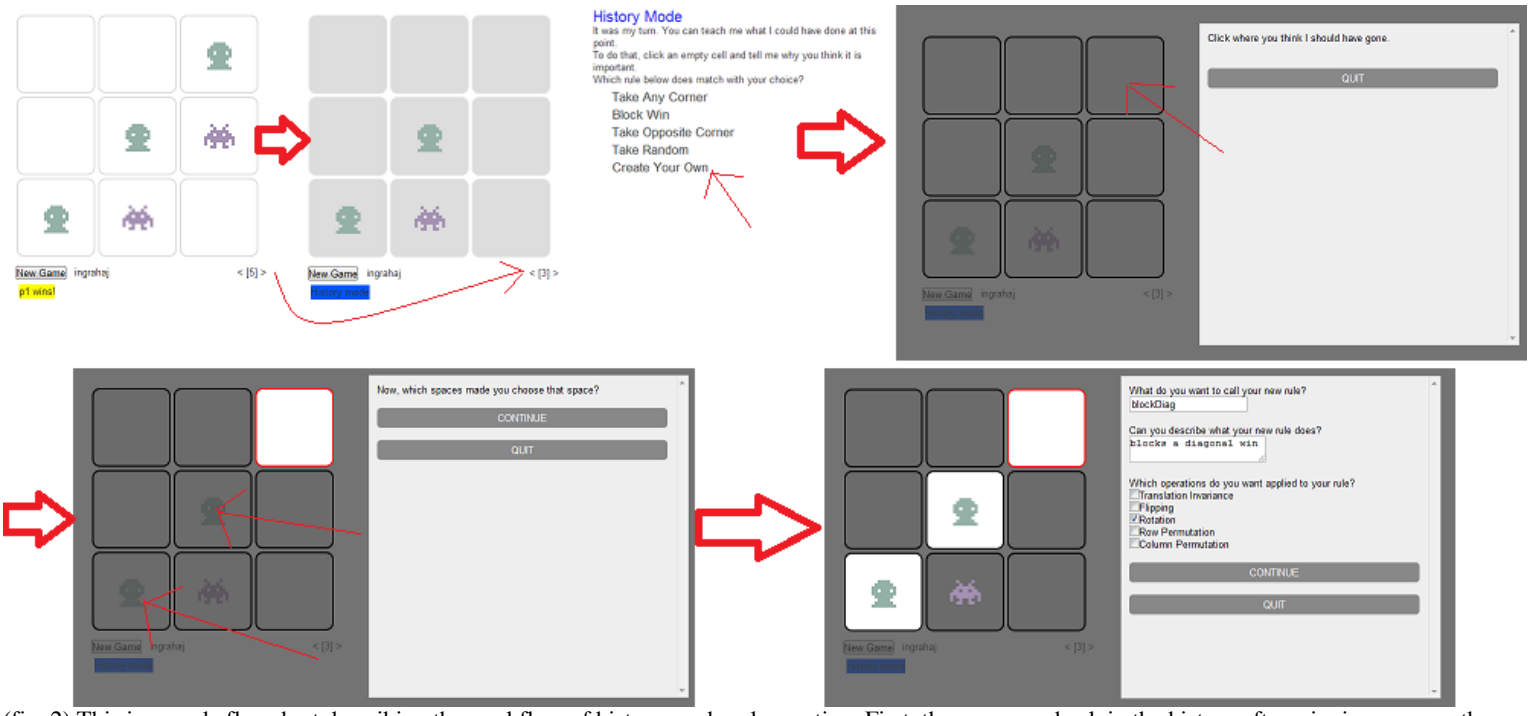
(fig. 1) The standard rule creation interface - this can be invoked at any time while the user trains his/her AI

While it is certainly a large step up in abstraction to step back and ask oneself the motivation behind any particular move, rule creation is built upon the two other, more concrete layers of the interface. After understanding the game itself and the algorithmic approach the computer player takes, the next natural step for the user is to design his own rules. In an experiment with KidsTeam at the Human Computer Interaction Lab here at the University of Maryland using Cooperative Inquiry[1], a group of 7-11 year olds were observed as they were briefly introduced to each successive layer of abstraction

found in CTArcade, except applied to Connect Four instead of Tic Tac Toe. Those that did not know the rules for Connect Four were quickly able to grasp them, but when asked to imitate a computer player and follow a specific strategy (with direction from the leaders of the experiment), some of them began to get confused. We also presented a paper prototype for rule creation to see if any of the participants were able to grasp the idea, and while a couple began designing rules with no issues, most were confused by the prototype and potentially the idea of rule creation itself. It remains to be seen if switching to Tic Tac Toe instead of Connect Four or providing a more malleable computer interface would aid in the understanding of rule creation for users in this age group, but for now it mostly exists in CTArcade as another level of abstraction for users to explore once they understand both the game itself and the algorithmic approach a computer player would take. However, even if rule creation in the way we have implemented it remains conceptually out of reach of the original target audience, it is still valuable for broadening CTArcade's appeal to older or more advanced students.

My Contributions:

- Designed an initial prototype for a Scratch-style[4], generalizable rule creation system
- Aided with a KidsTeam experiment on the feasibility of CTArcade in general, along with testing a rudimentary paper prototype of rule creation for Connect Four
- Refined the initial prototype into the first draft of the less flexible, but more intuitive rule creation interface in CTArcade
- Integrated this page into CTArcade, matching style and implementing workflow to allow smooth transition into and out of the interface when appropriate (fig. 2)
- Implemented various symmetry conditions that allow advanced users to easily generalize their rules with basic transformations



(fig. 2) This is a crude flowchart describing the workflow of history mode rule creation. First, the user goes back in the history after winning a game, then the AI asks the user
where it should have gone and why. The user may then select "Create Your Own" and follow the intuitive steps that will create a rule preventing that mistake in the future.

Summary

Rule creation in CTArcade serves as functionality that adds depth to the application and solidifies the computational skills of those that are able to utilize it. A graphical approach is used to provide an intuitive representation of rules while still remaining flexible enough to allow the creation of any deterministic strategy in Tic Tac Toe. Ideally, the focus on intuitive graphical respresentations of abstract concepts will ultimately aid the user in transitioning from the concrete rules of Tic Tac Toe to the abstract task of designing game strategies, thus developing important computational thinking skills. However, even if most younger users find this task too confusing, the depth that this feature provides will hopefully appeal to other users that may find the first two layers of abstraction trivial. As more content is added to CTArcade, rule creation becomes an incredibly important tool, as more complicated games quickly require larger and more creative sets of rules to create a competent strategy. Since CTArcade is still new and in development, the effect on users' computational thinking abilities from use of the tool is yet to be determined, much less the contribution of each layer of abstraction; it is possible that an intermediate step between manipulating provided rules and creating one's own would improve users' ability to use the rule creation interface successfully. However, rule creation is clearly a crucial step in computational thinking as it applies to games, as it reflects the user's resourcefulness and creativity given very basic tools in computational problem solving.

Bibliography

[1] Druin, A. Cooperative inquiry: Developing new technologies for children with children. In Proc. CHI 1999, ACM Press (1999), 592-599.
[2] Goldstone, R. L., and Son, J. Y. The transfer of scientific principles using concrete and idealized simulations. The Journal of the Learning Sciences 14, 1, (2005), 69-110
[3] Lee, T. Y., Mauriello, M. L., Ingraham, J., Sopan, A., Ahn, J., & Bederson, B. B. (2012). CTArcade: Learning computational thinking while training

virtual characters through game play. In Proceedings of ACM CHI 2012 Conference on Human Factors in Computing Systems, Extended Abstracts. [4] Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., et al. Scratch: Programming for all. Communications of the ACM 52, 11 (2009), 60-67
[5] Robins, A. "Learning and teaching programming: A review and discussion." Computer science education 13.2 (2003):137.
[6] Uttal, D. H., Liu, L. L., and DeLoache, J. S. Taking a hard look at concreteness: Do concrete objects help young children learn symbolic relations? In Child Psychology: A Handbook of Contemporary Issues, Lawrence Balter, Catherine Tamis-Lemonda, Eds., Psychology Press, New York, NY, 177-192. Psychology Press, New York, NY, 2000.