Studio K: Tools for game design and computational thinking

David Hatfield, Gabriella Anton, Amanda Ochsner, Kurt Squire, University of WI-Madison, Madison, WI,
Email: dlhatfield@gmail.com, gabby.anton@gmail.com, amanda.ochsner@gmail.com, kurt.squire@gmail.com,
R. Benjamin Shapiro, Tufts University, Medford, MA, ben@cs.tufts.edu
Alex Games, Microsoft, agames@microsoft.com

Abstract: People need to develop good intuitions about how computation operates within domains of practice (from social software to personalized medicine applications) and how to harness the computer's creative potential. *Studio K* provides an opportunity for young people to develop such computational ways of thinking by designing videogames as part of a virtual studio internship. In this demonstration, we will explore the social and performance feedback mechanisms being designed in *Studio K* to support the development of game design expertise and computational thinking. Pilot study data will supplement the discussion.

Introduction

As digital devices become integrated into more and more facets of our lives, it is imperative that people become literate with digital technologies. This requires more than knowing how to use any particular application or operating system, or even how to program. Instead it requires knowing how to abstract from situations and think computationally (Wing, 2006). People need to develop good intuitions about how computation operates within domains of practice (from social software to personalized medicine applications) and how to harness the "protean" power of the computer: creation. If understanding computation is valuable, using computation to build something personally meaningful is quite possibly the best way to get there. Computational thinking (National Academies, 2011; Wing, 2006) describes an ability to answer the question, "What can I build to understand and/or solve this problem?" Rather than promoting a continued consumption of media, a strong understanding of computational thinking creates active participants in media – changing focus to production and encouraging interest-driven use of media (Brennan & Resnick, 2012).

Unfortunately, learning this way of thinking can be difficult, and there are many intimidating challenges that arise for people looking to develop these skills. There is very little instruction in the US that teaches students how to apply computational thinking, especially outside of the context of programming. Computer science is rarely taught in high schools, and, when it is, emphasizes disconnected abstract principles that create powerful barriers to entry for a potentially diverse audience (Camp, 1997). Pulimood and Wolz (2008) suggests such barriers could be addressed through a pedagogical shift that includes 1) authentic inquiry through creative design, 2) collaborative work mirroring modern media practices, and 3) multidisciplinary approaches to content. *Studio K* demonstrates how incorporating the pedagogical shift described above in order to create video games, and then extending it with real-time performance feedback for both designers-in-training and their mentors can make computational thinking more concrete and more accessible,. In this demonstration session, participants will explore *Studio K* and engage in discussions of the social and performance feedback mechanisms designed into the experience. Data from seven pilot implementations will illustrate these discussions with the capabilities and needs of children in afterschool and classroom settings.

Significance of the Tool

While online games generally offer a vast array of literacy practices and reciprocal apprenticeship (Gee, 2003; Steinkuehler, 2004; Black & Steinkuehler, 2009; Black, 2008), becoming a game designer means going beyond technical creation to craft aesthetics, interactions, and stories that motivate users to play. As other programming and design environments, such as Alice (Pausch et al., 1995), Toontalk (Kahn, 1996), AgentSheets (Repenning, 1993), and Scratch (Maloney et al., 2010), have shown, a) providing an ever-growing reference collection of fun games to be deconstructed and built upon, b) enabling peers to help one another to progress in skill from beginner to expert; and c) creating safe spaces to learn to program together without fear of harassment or judgment (Resnick et al., 2009) can be powerful supports for learning to think computationally.

In *Studio K*, young people take on the role of a videogame designer participating in a virtual internship, working individually and collaboratively to design videogames using the 3D game design application *Kodu* (see

Figure 1). Working in *Kodu* (MacLaurin, 2009), designers play, revise, and create games as they navigate a series of design challenge missions targeting fundamental concepts distilled from the rich literature on videogame design (Anton, Ochsner, Squire, 2013; Kane et al., 2012).

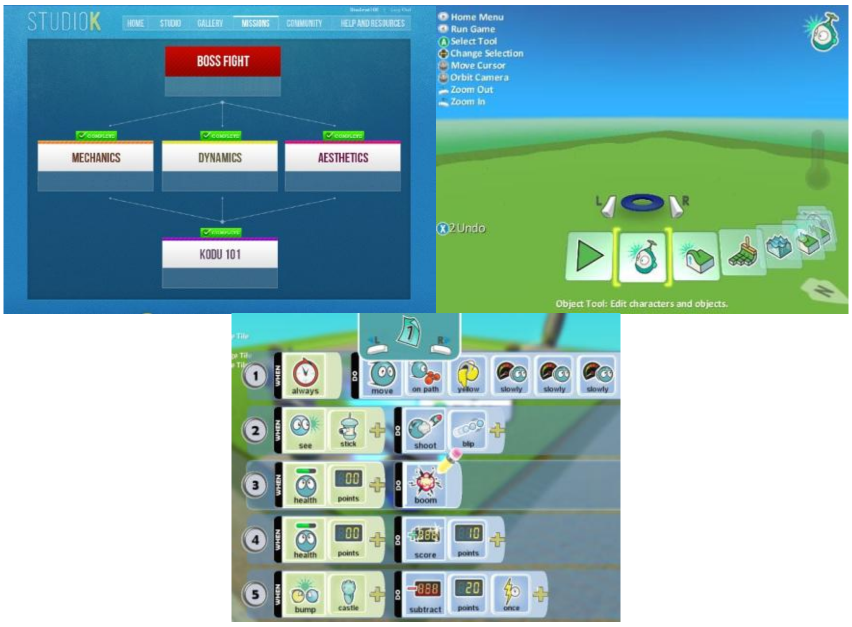

Figure 1. On the left, the Studio K curriculum interface,
In the middle, the initial Kodu interface and design environment.
On the right, programming tiles in Kodu.

While much of the designers' time is spent within *Kodu* designing and creating games, those experiences are contextualized and scaffolded through interactions in the *Studio K* site, where designers engage in professional practices such as playtesting, writing reviews, and iterative design. As they progress through the missions, designers are exposed to increasingly complex models of games and presented with frequent opportunities to develop and share their expertise through community forums and peer reviews.

These social feedback activities are designed to foster a sense of community and mirror the vital practices of successful game developers. As these interactions occur, designers-in-training can develop a richer use of specialist videogame language, an emerging culture of critique, and academic literacy in reading and writing (Peppler, Diazgranados, & Warschauer, 2010; Peppler & Kafai, 2007). These practices promote the designer's agency in design work; designers learn to express opinions, provide justifications, and maintain autonomy in their views even after discussions with their peers (Peppler, Diazgranados, & Warschauer, 2010). These activities also support design processes rich in debugging, iterative design, and feedback, skills central to computational thinking practices (Denning, 2003) which have been observed in use during game play (Berland & Lee, 2011).

In addition to social feedback, designers-in-training can develop their expertise through the choices they make during the design process itself with the aid of learning analytics. As designers create games within *Kodu*, all of their activities are gathered and sent to a database in the site. Research on *Studio K* is investigating the computational thinking practices occurring during game design, as seen in clickstream data, and how learning analytics can be used for mentor- and designer-in-training self-assessment in the site to provide richer experiences. While there are myriad computational thinking practices (Denning, 2003), our initial focus is on debugging, logic, and feedback as previous work suggests their prevalence within game use and design (Berland & Lee, 2011; Stolee & Fristoe, 2011; Brennan & Resnick, 2012). The analyses of these activities are through the lens of analytics on professional programming (Johnson, 2007; Kou, Johnson, & Erdogmus, 2009) to better understand how modifications to these methods can be applied in an educational setting with live embedded feedback.

For example, using a preliminary examination of pilot clickstream data and qualitative observation, we are investigating *debugging* as a computational thinking construct mappable to particular patterns of choices within *Kodu*. Debugging is the act of determining the cause of a malfunction or error (Berland & Lee, 2011). In *Studio K*, we operationalize *debugging* as sequences of *design* or *programming* types of player actions, followed by player actions categorized as *play*, succeeded by additional *design* and *programming* actions. Figure 2 shows a prototype visualization for monitoring *debugging* cycles.

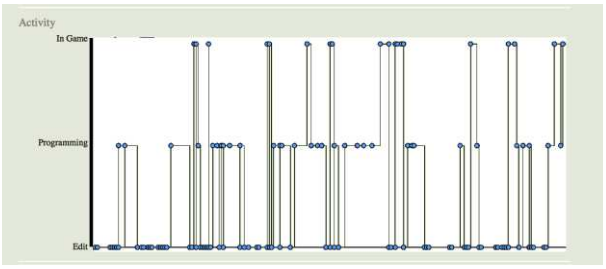

Figure 2. Studio K visualization of Kodu activity.

Pilot studies we are conducting are investigating this mapping of raw clickstream data to higher level constructs to try to better understand such questions as whether the period and frequency of such cycles might provide a good indication of emerging design skill − do they vary with the difficulty of different design challenges? Do designers-in-training demonstrate significant changes in the rates of these cycles from early to later challenges? We are also interested in whether and how such cycles might relate to other performances during the virtual internship, such as designers' reflections on their own designs and on their peers' designs, as well as to learning outcomes measured before and after the experience. And more generally, we are interested in examining (and discussing during the demonstration session) questions like What are the social mechanisms that engage youth in advanced gaming practices that might be leveraged more broadly? What kinds of collaborative tools and data reporting tools for designers-in-training and mentors lead to deeper participation?

Our ultimate goal is to use such measures to drive visualizations within the Studio K website to provide powerful feedback for designers and mentors. The integrated performance and social feedback mechanisms should provide for a rich learning experience both in game design and computational thinking. Students learn increasingly complex programming concepts that are applied within a range of game design situations. The program offers a unique framework for learning programming and computational thinking practices through the lens of game design learning goals. Rather than restricting students to focus on the abstract principles conventionally taught in computer science courses, opportunities for learning are opened to allow interest and creativity to guide activities through curricular participation. As a result the learning focus more strongly emphasizes understanding and application of game design principles through which other skills will be gained. Through these experiences, students should gain autonomy and agency as videogame designers and fluency with valuable computational thinking skills. As *Studio K* develops, it offers a unique opportunity for youth to observe their development and track their progress in interactive, meaningful ways. By further understanding and modifying the methods of analysis applied to professional programming activities, we gain further understanding of how creative technology uses are beneficial to youth and how these practices can encourage larger computational thinking practices. Though these tools are still works in progress, their use has interesting implications for the use of self assessments and usability of learning analytics to support the development of computational thinking for a broader audience than is currently the case.

References

Anton, G., Ochsner, A., & Squire, K. (2013). Interest-driven learning in game design environments. Presented at Digital Media Learning Conference, IL, March 14-16.

Berland, M., & Lee, V. R. (2011). Collaborative strategic board games as a site for distributed computational thinking. *International Journal of Game-Based Learning*, *1*(2), 65.

Black, R.W. (2008). Covergence and divergence: Online fanfiction communities and literacy pedagogy. In Z. Bekerman, N. Burbules, H. Giroux, & D. Silberman-Keller (Eds.), *Mirror images: Popular culture and education.* New York: Peter Lang.

Black, R.W. & Steinkuehler, C.A. (2009). Literacy in virtual worlds. In L. Christenbury, R. Bomer, & P. Smagorinsky (Eds.), *Handbook of adolescent literacy research (*pp. 271-286). New York: Guilford Press.

Brennan, K., & Resnick, M. (2012) New frameworks for studying and assessing the development of computational thinking.

Camp, T. (1997). The incredible shrinking pipeline. *Communications of the ACM*, *40*(10), 103-110.

Denning, P. J. (2003). Great principles of computing. *Communications of the ACM*, *46*(11), 15-20.

Gee, J. (2003). *What Videogames have to teach us about learning and literacy.* New York: Palgrave Macmillan.

Johnson, P. M. (2007, September). Requirement and Design Trade-offs in Hackystat: An in-process software engineering measurement and analysis system. In *Proc. of 1st Int. Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society Press*.

Kahn, K. 1996. ToonTalk™ – An animated programming environment for children. Journal of Visual Languages and Computing. (An abbreviated version appeared in Proceedings of the National Educational Computing Conference. Baltimore, MD, USA, 7 (June): 197-217, 1995.)

Kane, L., Berger, W., Anton, G., Shapiro, R.B., & Squire, K. (2012). *Studio K: A game design curriculum for computational thinking.* Presented at Games+Learning+Society Conference, Madison, WI, June 13-15.

Kou, H., Johnson, P. M., & Erdogmus, H. (2010). Operational definition and automated inference of test-driven development with Zorro. *Automated Software Engineering*, *17*(1), 57-85.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education, 10*(4).

MacLaurin, M. (2009). Kodu: end-user programming and design for games. In *Proceedings of the 4th International Conference on Foundations of Digital Games* (p. 2). ACM.

The National Academies. (2011). *Report of a workshop on the pedagogical aspects of computational thinking.* Available from: http://www.nap.edu/catalog.php?record_id=13170

Pausch, R., Burnette, A.C., Conway, M., Cosgrove, D., DeLine, R., Durbin, J., Gossweiler, R., Koga, S., & Whie, J. (1995). A brief architectural overview of Alice, a rapid prototyping system for virtual reality. In *IEEE Computer Graphics and Applications*

Peppler, K., Diazgranados, A., & Warschauer, M. (2010). Game Critics: exploring the role of critique in game-design literacies. *E-learning and Digital Media*, *7*(1), 35-48.

Peppler, K. A., & Kafai, Y. B. (2007). From SuperGoo to Scratch: exploring creative digital media production in informal learning. *Learning, Media and Technology*, *32*(2), 149-166.

Pulimood, S. M., & Wolz, U. (2008, March). Problem solving in community: a necessary shift in cs pedagogy. In *ACM SIGCSE Bulletin* (Vol. 40, No. 1, pp. 210-214). ACM.

Repenning, A. (1993). *Agentsheets: A tool for building domain-oriented dynamic, visual environments.*(Doctoral Dissertation) University of Colorado at Boulder, Dept. of Computer Science

Squire, K., & Jenkins, H. (2003). Harnessing the power of games in education. *Insight*, *3*(1), 5-33.

Steinkuehler, C. A. (2004). Learning in massively multiplayer online games. In Y. B. Kafai, W. A. Sandoval, N. Enyedy, A. S. Nixon, & F. Herrera (Eds.), *Proceedings of the Sixth International Conference of Learning Sciences* (pp. 521-528). Mahwah, NJ: Erlbaum.

Stolee, K. T., & Fristoe, T. (2011, March). Expressing computer science concepts through Kodu game lab. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 99-104). ACM.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

Acknowledgements
This work was supported by a gift from Microsoft. Any opinions, findings, or conclusions expressed in this paper are those of the authors and do not necessarily reflect the view of the cooperating institutions.