

Monsterismus: Recursively Relevant CS Game Design

Matthew Berland, University of Wisconsin at Madison, mberland@wisc.edu
Timothy T. Yuen, Don Davis, University of Texas at San Antonio
Timothy.Yuen@utsa.edu, Don.Davis@utsa.edu

Abstract: Monsterismus is an educational computer science (CS) game that represents an important step in advancing and developing video games to support CS learning. The development of such games is crucial as significant efforts must be made to promote students' interest in CS and to meaningfully involve students with more authentic programming tasks. Monsterismus provides a two-fold exemplar of best practices in computer science education for both novices and more advanced students. Namely, Monsterismus provides a more relevant and engaging context to engage novices with programming syntax and concepts. Secondly, Monsterismus is the product of students participating in authentic programming tasks, which is important for bolstering students' perception of relevance in order to impede CS attrition and support students' work place readiness. As such, Monsterismus represents the "recursively relevant" design of CS educational game.

The need to bolster CS participation

Many demographic groups are severely underrepresented among CS graduates (NSF, 2011), which negatively impacts the competitiveness and responsiveness of US software manufacturers and represents a loss for consumers (Margolis & Fisher, 2003; Tornatzky, Macias, & Jones, 2002). Therefore, significant attention needs to be given to promoting underrepresented students' interest in CS both before and after matriculation (Beheshti et al., 2008; Margolis, Goode, Holme, & Nao, 2008).

Towards this end, Monsterismus exemplifies "recursively relevant" design of educational games in which both construction of the game and the resulting game provide relevant CS learning experiences. The game itself incorporates design elements to facilitate younger students' interest in and familiarity with CS. Further, the design itself was part of a capstone software engineering class, whereby students engaged in more 'authentic' agile software development whereby they were contracted to complete a software project in consultation with a client (the first author).

For younger novice CS students, the Monsterismus interface provides students with an 'authentic' purpose to engage with programming activities and concepts. Within the game (see Figure 1), students must program an avatar to collect various items (e.g. program a dragon to collect fruit and treasure). This is important because relevance, supported by authentic purpose, is especially important for students frequently underserved in US K-12 education (Ladson-Billings, 1995). Monsterismus serves as an authentic context as many students accept game narratives as a relevant and authentic purpose (Dondlinger, 2007). Therefore, by making programming an integral part of game play, Monsterismus provides students with an authentic context to engage with programming, which is necessary to generating and supporting interest in CS among underrepresented students (Goode, 2008; see also Ladson-Billings, 1995).

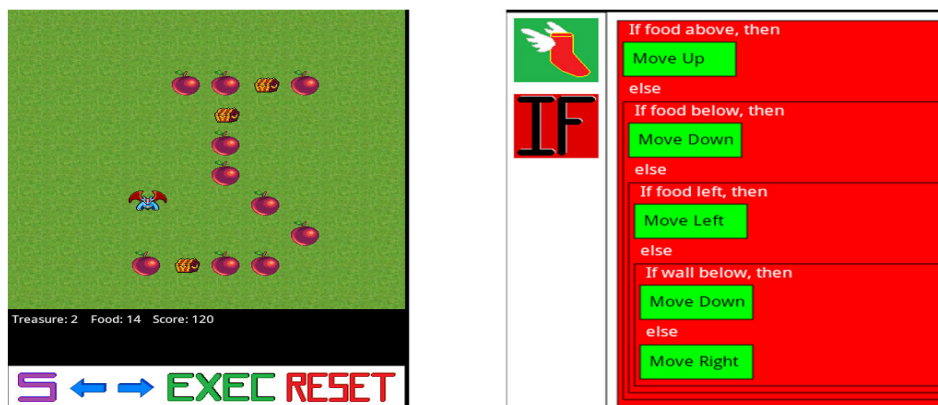


Figure 1: The Monsterismus Action Screen (Left) and Programming Screen (Right)

Moreover, Monsterismus' design reflects the understanding that any attempts to bolster participation in CS should be accessible to and promote the interest of younger learners for two significant reasons: 1) younger learners' interests significantly determine later STEM related career choices (Tai, Liu, Maltese, & Fan, 2006); and, 2) the lack of opportunities to meaningfully interact with CS at younger ages contributes significantly to CS underrepresentation (Margolis et al., 2008). For these reasons, Monsterismus provides an intuitive easy to use 'clickable' interface (see Figure 1), but strives to make significant CS concepts and verbiage, such as conditional statements (Goldman et al., 2008), apparent, else Monsterismus' connection to CS may be lost and so too any potential interest in CS careers it may have garnered (Eisenberg, 2003).

Additionally, the development of Monsterismus represents an effort to increase CS relevance for more advanced CS students. Namely, CS educators have highlighted the need to solidify more advanced students' CS interests and skills by engaging students in more relevant and authentic programming tasks (Fernandez & Tedford, 2006). As such, many CS educators recommend providing authenticity by requiring students to complete tasks in consultation with clients (Ellis, Morelli, Lanerolle, Damon, & Raye, 2007). Monsterismus successfully exemplifies this process as it was commissioned, planned, programmed, and debugged all within a ten week semester.

Monsterismus: Next Steps

Moving forward, Monsterismus' tasks, narratives, and interface will be revised in response to student feedback. Also, researchers will further ascertain students' perceived connections between Monsterismus and CS and the extent to which Monsterismus promotes interest in studying CS or pursuing CS careers.

References

- Beheshti, M., Alo, R. A., Fernandez, J., Gates, A. Q., Ranjan, D., Boadi, A., ... Barker, L. (2008). Work in progress - CS0 course implementation in Computer Science. In *Frontiers in Education Conference, 2008. FIE 2008. 38th Annual* (pp. F4A-7-F4A-9).
- Dondlinger, M. J. (2007). Educational video game design: A review of the literature. *Journal of Applied Educational Technology, 4*(1), 21-31.
- Eisenberg, M. (2003). Mindstuff. *Convergence: The International Journal of Research into New Media Technologies, 9*(2), 29-53.
- Ellis, H. J. C., Morelli, R. A., Lanerolle, T. R. de, Damon, J., & Raye, J. (2007). Can humanitarian open-source software development draw new students to CS? *SIGCSE Bull., 39*(1), 551-555.
- Fernandez, J. D., & Tedford, P. (2006). Evaluating computing education programs against real world needs. *J.Comput.Sci.Coll., 21*(4), 259-265.
- Goldman, K., Gross, P., Heeren, C., Herman, G., Kaczmarczyk, L., Loui, M. C., & Zilles, C. (2008). Identifying important and difficult concepts in introductory computing courses using a Delphi process (pp. 256-260).
- Goode, J. (2008). Increasing diversity in K-12 computer science: Strategies from the field. *SIGCSE Bulletin, 40*(1), 362-366.
- Ladson-Billings, G. (1995). But that's just good teaching! The case for culturally relevant pedagogy. *Theory into Practice, 34*(3), 159-165.
- Margolis, J., & Fisher, A. (2003). *Unlocking the clubhouse: Women in computing*. Cambridge, MA: The MIT Press.
- Margolis, J., Goode, J., Holme, J. J., & Nao, K. (2008). *Stuck in the shallow end: Education, race, and computing*. Cambridge, MA: The MIT Press.
- National Science Foundation, Division of Science Resources Statistics. (2011). *Women, Minorities, and Persons with Disabilities in Science and Engineering: 2011* (Special Report NSF 11-309). Arlington, VA.: National Science Foundation, Division of Science Resources Statistics.
- Tai, R. T., Liu, C. Q., Maltese, A. V., & Fan, X. T. (2006). Planning early for careers in science. *Science, 312*, 1143-1145.
- Tornatzky, L. G., Macias, E. E., & Jones, S. (2002). *Latinos and information technology: The promise and the*

challenge. Claremont, CA: The Tomás Rivera Policy Institute.

Acknowledgments

The authors would like to thank David Kosub, Trevor Vardeman, Jeff Wooldridge, Matthew Johnston, and Carlos Martinez. Also, the authors would like to express their appreciation to Gamestar, Studio K, the Lifelong Kindergarten Group, Nathan Holbert, David Weintraub, Uri Wilensky, and Globaloria for their seminal work that informed the design of Monsterismus.