Educational Accelerometer Games for Computer Science

Ville Karavirta
Aalto University School of Science
P.O.Box 15400
FI-00076 Aalto, FINLAND
ville.karavirta@aalto.fi

Lasse Hakulinen
Aalto University School of Science
P.O.Box 15400
FI-00076 Aalto, FINLAND
lasse.hakulinen@aalto.fi

ABSTRACT
Algorithm visualizations have been used a lot in computer science to help students understand abstract concepts such as data structures and algorithms (DSA). The most widely used algorithm visualization systems are all based on Java and, alas, are not usable on mobile devices. We tackled this problem by designing a set of assignments for mobile learning of an important concept in DSA, namely binary trees. To make the assignments more motivational and gameful, they use the accelerometer data for simulating the behavior of the algorithm in question. This paper introduces the prototype we built as well as discusses the pros and cons of such assignments.

Author Keywords
Accelerometer, mobile, engagement, mobile learning, algorithm visualization, serious games, gamification

INTRODUCTION

In computer science, learning abstract concepts such as data structures and algorithms has been generally considered difficult for students. This has led to the development of a multitude of algorithm visualization (AV) systems designed to aid students in understanding such abstract concepts. The most well known systems include Animal (Rößling and Freisleben, 2002), JHAVÉ (Naps, 2005), ViLLE (Laakso et al., 2008), and TRAKLA2 (Malmi et al., 2004). The extensive research on educational use of these and other AV systems has come to the conclusion that to be educationally effective, learning materials need to be interactive (Hundhausen et al., 2002; Naps et al., 2003).

The current breed of systems has problems, though. Most of the systems are implemented in Java (Shaffer et al., 2010). With the recent rise of use of mobile devices for web usage in general and for online learning in specific, the existing solutions are not applicable. This has led to research into bringing algorithm visualization to mobile devices (Karavirta, 2011).

Modern mobile devices include growing number of accurate sensors enabling applications to take advantage of information such as location and movement of the device (Lane et al., 2010). This progress has been also a significant boost for mobile serious games, where the aim is to learn useful things while playing. Good examples of uses of mobile sensors for games are accelerometer in MotionMath[1], geolocation in MobileGame (Schwabe and Göth, 2005), and cameras in Art of Defence (Huynh et al., 2009)

The mentioned problems with existing AV systems on mobiles, advance of mobile sensing as well as the rise of serious games is the motivation for us to pursue the use of accelerometer information. In this work in progress paper, we present a prototype of assignments for data structures and algorithms that take advantage of the accelerometers in mobile devices. The prototype presented in Section 3 includes assignments for tree structures where students simulate the behavior of an algorithm by guiding a ball through the tree data structure.

BACKGROUND

Interaction has been hypothesized to be the key to educational effectiveness of algorithm visualization. The Engagement Taxonomy summarizes this to five active levels of engagement of the learner with a visualization (Naps et al., 2003). The engagement levels range from passive *viewing* of a movie-like visualization to student *creating* and *presenting* visualizations to others.

Our previous work, a system called TRAKLA2 (Malmi et al., 2004), includes assignments where students simulate actual algorithms by manipulating data structure visualizations (thus, engagement level *changing* since they are changing the existing visualization). The system has a growing set of assignments, currently around 50. However, the assignments are implemented as Java applets, and do not work on mobile devices. Our main motivation was to improve these and explore new possibilities of mobile sensor data at the same time.

[1] http://motionmathgames.com/motion-math/

[2] http://www.bythemark.com/products/quiz-learn-python/

Mobile learning has become a hugely popular field of research. According to a recent literature review, computer science is one of the disciplines where mobile learning is used the most (Wu et al., 2012). One such example is Quiz&Learn Python[2], which is a mobile quiz game for learning Python programming. It also includes visualizations of program code behavior. We are not aware of research in CS that would use accelerometer in learning. In general, accelerometer data has seen different uses since supported by mobile devices. Imaginative examples include using it for text input in chat on a wristwatch (Partridge et al., 2002) and enabling vision-impaired to play and interact with others by moving the device to give commands (Mehigan, 2009). A popular category for application has been acceleration based games (Chehimi and Coulton, 2008; Baek and Yun, 2008). In educational games, acceleration is used in MotionMath for learning fractions.

In general, playing computer games is often considered to be engaging and motivating. Shabanah et al. (2010) propose that playing computer games can combine all five active levels of the Engagement Taxonomy. The vast popularity of games combined with the increasing number of mobile devices offer a huge potential that could be harvested with well-designed games that also teach the players. Serious games tackle that challenge. They are games whose primary goal is education, rather than entertainment (Michael and Chen, 2006). Optimally, serious games make use of the engaging features of recreational games, while including educational goals in the gameplay. Another approach for making assignments more motivating is to use methods of gamification. Deterding et al. (2011) define gamification as: *"the use of game design elements in non-game contexts"*. However, the line between serious games and gamified applications is not always clear.

ACCELEROMETER FOR INTERACTIVE BINARY TREE ASSIGNMENTS

Our interactive assignments[3] deal with binary trees and binary search trees (BST), both of which are important concepts in data structures and algorithms used in, for example, in many built-in libraries of programming languages for storing and retrieving key-value pairs as well as for sorting data. They also provide a good opportunity to use accelerometer data since the algorithms have the concept of "visiting" the nodes. That is, the algorithm does something to the current node, left child, and right child in some order decided by the algorithm (typically based on the input).

Figure 1 shows an example of one such assignment in progress. The main idea is to guide the ball by tilting the device. The ball needs to be guided through the nodes in the order that the actual algorithm would visit them. In the case of in-order tree traversal shown in the figure, the algorithm will first visit the left child, then the root, and lastly the right child.

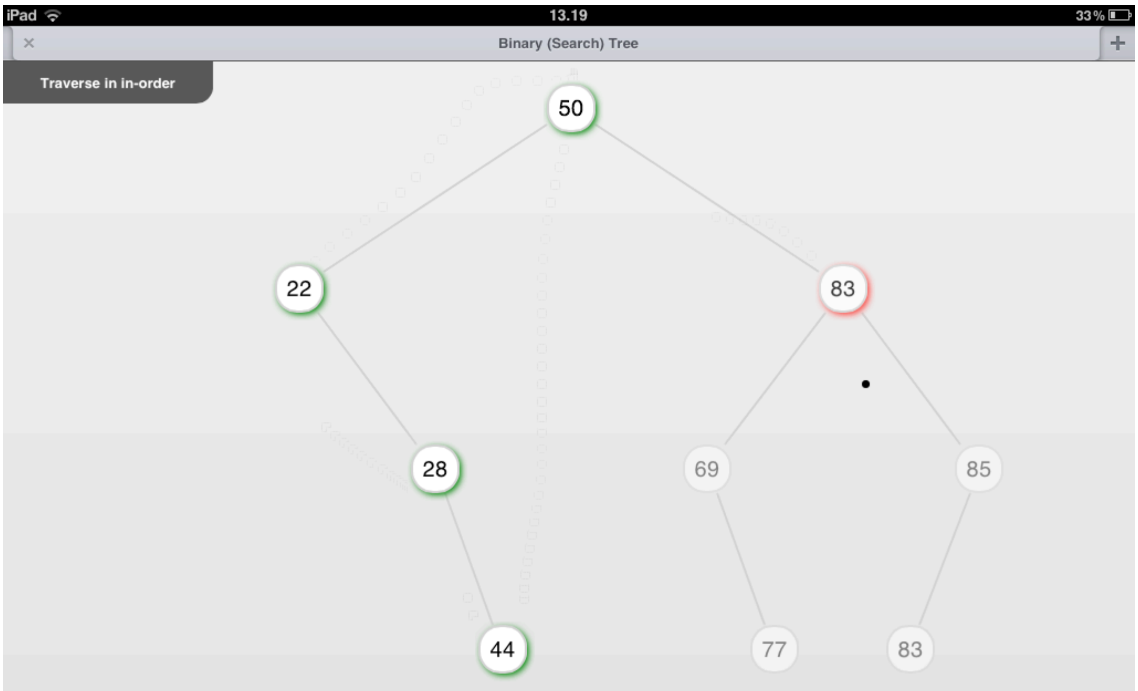

Figure 1: The assignment is solved by guiding the ball (below node 83) in the tree. Green highlight (nodes 50, 22, 28, and 44) is used to indicate correct steps whereas red (node 83) indicates incorrect step.

The exercises have some features that make them pedagogically more useful. First, the assignment can be tried as many times as the student likes. On each try, a new randomized input (that is, randomized tree) is generated to make solving it

---

[2] http://www.bythemark.com/products/quiz-learn-python/

[3] Currently a set of five assignments.

again more meaningful. The student also gets immediate feedback on each step. This is indicated by the green and red highlights of correct and incorrect steps, respectively. Solving of the assignment can always be continued after an incorrect step. To track student progress, each assignment keeps a record of how many times the student has tried to solve it and how many times it was correctly solved. While assignments in our existing TRAKLA2 environment give points to the course fulfillment, the accelerometer assignments record the fastest time the student solved the assignment. A design choice was not to have the chance to pause the assignment. We argue that this makes it more hectic and does not give time to think.

While we see the concept important, we want to briefly touch on the technology used. The assignments are implemented using HTML5 and JavaScript, using the W3C Device Orientation Event specification[4] and SVG (Scalable Vector Graphics) for drawing the graphics. This makes it platform independent instead of choosing some native application platform. However, to be able to use the assignments without a network connection, we have also made a native application for Apple iOS.

DISCUSSION

One could argue that using accelerometer to select the nodes of a binary tree is just a more difficult way to complete the task than clicking them. Also, using accelerometer instead of clicking is likely to increase the time needed to complete the task. This can be demotivating especially if the assignment is a compulsory part of a course. However, the challenge of traversing the tree by tilting the device could also increase the motivation and engagement towards the assignment. McGonigal (2011) states that rules that limit players' allowed actions are one of the four defining traits that all games share, along with: a goal, a feedback system, and voluntary participation. The use of accelerometer to navigate instead of clicking the nodes can be seen as a rule that limits players' allowed actions. In a similar way, for example, golfers have agreed to follow a set of rules that limit their actions of moving the ball to move it only by using a golf club. In our opinion, using accelerometer could be one way to make the exercises more gameful and engaging by adding the limitation of navigating the binary tree only by tilting the device.

Using accelerometer is one step towards game-like assignments, but there are also several other possibilities that could be done in order to make these assignments more gameful. Garris et al. (2002) provide a categorization of game elements that differs slightly from McGonigal's traits of games. They say that the six key dimensions that characterize games are: fantasy, rules/goals, sensory stimuli, challenge, mystery, and control. By taking these elements into account, we could gamify our assignments further. More appealing graphics could be introduced to improve the sensory stimuli of the assignments and a background story could be used to add fantasy element in order to draw the attention of the players. For example, in case of the binary search tree search, the goal of the game could be to guide a parachutist from the top of the screen to the correct landing pad following the rules of the BST search algorithm. Also, increasingly difficult levels could be used to offer suitable challenges for different players.

In addition to thinking the gameful aspects, the scope of game-like assignments using accelerometer could be widened. Binary search tree and tree traversals are rather simple topics, meaning that mental processing required for one step in the algorithm is not complex. Thus, the continuous control of the device does not make solving of the problem too hard. However, in more complex algorithms, this should be addressed in the design of the assignment.

As for the role of such assignments on a course, we like the approach of having them as optional material supporting the learning of course content. Naturally, the other option to use these assignments would be to use them as a required or alternate way to get points for a part of a course. However, as we believe mobility and the use of accelerometer is more motivational for students, we like to keep these as voluntary learning objects. This approach is also supported by one of McGonigal's traits of games. It should be noted, that this does not mean that they could not be used to get credits on a course.

Finally, when compared with the existing AV systems, the engagement in our approach is different and does not directly fit any of the existing categories of the engagement taxonomy. Thus, the educational effectiveness of these assignments is something we will research in the future evaluation of the system.

SUMMARY

In this paper, we have introduced a set of assignments for data structures and algorithms that use the accelerometer data for student engagement. We have also discussed pros and cons of such assignments as well as provided possible future directions. Concrete future steps we want to take include usage with students on a course. This will be done as an additional way for students to learn instead of giving points for it, for the reasons discussed above. Moreover, we will diversify the selection of assignments. Finally, we are encouraged by the good examples of educational games using different mobile sensors. Thus, we will explore new ways to bring these sensors into CS education in more game-like ways.

[4] http://dev.w3.org/geo/api/spec-source-orientation.html

REFERENCES

Baek, J. and B.-J. Yun, 2008. A sequence-action recognition applying state machine for user interface. Consumer Electronics, IEEE Transactions on, 54(2):719–726.

Chehimi, F. and P. Coulton, 2008. Motion controlled mobile 3D multiplayer gaming. Advances in Computer Entertainment Technology, Proceedings of the 2008 International Conference on, pages 267–270.

Deterding, S., Dixon, D., Khaled, R., and Nacke, L. (2011). From game design elements to gamefulness: defining "gamification". In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek '11, pages 9–15, New York, NY, USA. ACM.

Garris, R., Ahlers, R., and Driskell, J. E. (2002). Games, motivation, and learning: A research and practice model. Simulation & Gaming, 33(4):441–467.

Hundhausen, C. D., S. A. Douglas, and J. T. Stasko, 2002. A meta-study of algorithm visualization effectiveness. Journal of Visual Languages and Computing, 13(3):259–290.

Huynh, D.-N. T., K. Raveendran, Y. Xu, K. Spreen, and B. MacIntyre, 2009. Art of defense: a collaborative handheld augmented reality board game. In Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games, Sandbox '09, pages 135–142, New York, NY, USA. ACM.

Karavirta, V., 2011. Perspectives on algorithm visualization on mobile devices. In Proceedings of the Sixth Program Visualization Workshop, pages 59–65, Darmstadt, Germany.

Laakso, M.-J., T. Rajala, E. Kaila, and T. Salakoski, 2008. The impact of prior experience in using a visualization tool on learning to program. In Proceedings of Cognition and Exploratory Learning in Digital Age (CELDA 2008).

Lane, N., E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, 2010. A survey of mobile phone sensing. Communications Magazine, IEEE, 48(9):140–150.

Malmi, L., V. Karavirta, A. Korhonen, J. Nikander, O. Seppälä, and P. Silvasti, 2004. Visual algorithm simulation exercise system with automatic assessment: TRAKLA2. Informatics in Education, 3(2):267–288.

McGonigal, J., 2011. Reality is broken: Why games make us better and how they can change the world. Jonathan Cape Ltd.

Mehigan, T. J., 2009. Harnessing accelerometer technology for inclusive mobile learning. In Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI'09, pages 100:1–100:2, New York, NY, USA. ACM.

Michael, D. and S. Chen, 2006. Serious games: games that educate, train, and inform. Thomson Course Technology PTR, Boston, MA.

Naps, T. L., 2005. JHAVÉ: Supporting Algorithm Visualization. IEEE Computer Graphics and Applications, 25(5):49–55.

Naps, T. L., G. Rößling, J. Anderson, S. Cooper, W. Dann, R. Fleischer, B. Koldehofe, A. Korhonen, M. Kuittinen, C. Leska, L. Malmi, M. McNally, J. Rantakokko, and R. J. Ross, 2003. Evaluating the educational impact of visualization. SIGCSE Bulletin, 35(4):124–136.

Partridge, K., S. Chatterjee, V. Sazawal, G. Borriello, and R. Want, 2002. Tilttype: accelerometer-supported text entry for very small devices. In Proceedings of the 15th annual ACM symposium on User interface software and technology, UIST '02, pages 201–204, New York, NY, USA. ACM.

Rößling, G. and B. Freisleben, 2002. ANIMAL: A system for supporting multiple roles in algorithm animation. Journal of Visual Languages and Computing, 13(3):341–354.

Schwabe, G. and C. Göth, 2005. Mobile learning with a mobile game: design and motivational effects. Journal of Computer Assisted Learning, 21(3):204–216.

Shabanah, S., J. Chen, H. Wechsler, D. Carr, and E. Wegman, 2010. Designing computer games to teach algorithms. In 2010 Seventh International Conference on Information Technology, pages 1119–1126. IEEE.

Shaffer, C. A., M. L. Cooper, A. J. D. Alon, M. Akbar, M. Stewart, S. Ponce, and S. H. Edwards, 2010. Algorithm visualization: The state of the field. Transactions in Computing Education, 10:9:1–9:22.

Wu, W.-H., Y.-C. J. Wu, C.-Y. Chen, H.-Y. Kao, C.-H. Lin, and S.-H. Huang, 2012. Review of trends from mobile learning studies: A meta-analysis. Computers & Education, 59(2):819–827.