Computer City: Sewers: An educational computer game for teaching digital logic

Aaron Weeden

December 20, 2009

Contents

Abstract

In recent years, computer science education has seen a decrease in both retention of majors and enrollment in introductory computer science classes. Research has shown that incorporating computer games in the CS curriculum improves interest and motivation of students, which can help improve retention rates. Digital logic is one topic in particular that responds particularly well to game-based teaching. To that end, the author of this paper has developed an educational game to teach digital logic to students in introductory computer science classes.

Background

Educational Games in CS Education

Since 2000, there has been a general decrease in enrollment and retention of students in computer science departments [Sung, 2009], [Eagle and Barnes, 2008], [Leutenegger and Edgington, 2007]. The diversity of participants in these departments has also decreased in upper-level courses, with fewer minorities and women participating in these classes [Sung, 2009], [Eagle and Barnes, 2008]. There are many reasons to be concerned over these trends, as outlined in [Sung, 2009] and [Beaubouef and Mason, 2005]. However, there has also been much work done to identify the reasons behind poor retention and enrollment rates, chief among which include poor problem solving skills, lack of feedback, and lack of motivation [Beaubouef and Mason, 2005]. One of the more promising solutions to these problems is the incorporation of computer games into the computer science curriculum. Games have been shown on multiple occasions to have a positive effect on motivation and engagement of students; and with included educational components, they also provide increases in learning [Eagle and Barnes, 2008], [Sweedyk et al., 2005], [Bayliss, 2009], [Wolz et al., 2006], [Bayliss and Strout, 2006], [Belfore et al., 2009]. Games appeal to a diverse generation of students who are already familiar with computers and have played games as they enter introductory computer science classrooms [Becker, 2001]. Thus, games are a way to provide a fun and motivating learning experience for students from a wide variety of backgrounds.

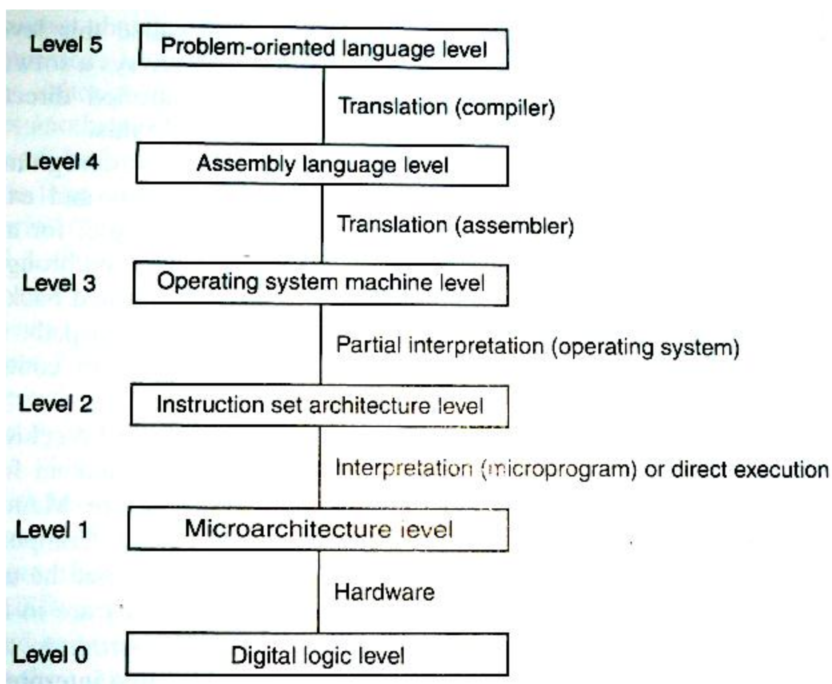
Digital Logic Education

In addition to the decreased retention rate of computer science students, certain fundamental topics in computer science have also seen a decrease in retention. For example, according to Michael B. Gousie, when the ACM and IEEE Joint Task Force published Computing Curricula 2001 [Joi, 2001], the coverage of computer architecture in education decreased dramatically in favor of other emerging technologies [Connelly et al., 2004]. Since this publication, many educators have argued for the retention of digital logic in computer science curriculum. As described by Mark E. Hoffman, "computer hardware is fundamental to all computing, and digital logic is the fundamental theory on which hardware is built" [Connelly et al., 2004]. Understanding digital logic can provide a basis for understanding higher-level concepts and an appreciation for the limitations of the computer [Hoffman, 2004] [Connelly et al., 2004]. A compact understanding of computer architecture is also advantageous for students of engineering [Connelly et al., 2004]. Because digital logic can be implemented in a variety of hardware technologies, digital logic further provides a connection between theory and practice [Hoffman, 2004]. The practical implementations of digital logic are simple and easy to understand, and they link directly to the theory behind them. Furthermore, once the student has learned the fundamental theories, levels of abstraction can be introduced to connect digital logic to higher levels of the computer architecture.

[Belfore et al., 2009] argues that traditional models of teaching do not provide an easy-to-understand connection between the theories of digital logic and their implementations, and that a game-based environment can provide better representations of this connection. Games provide a multimedia experience that help the player understand connections using multiple senses. Thus, games are well-suited for digital logic education, which can greatly benefit from multimedia models of digital logic functionality.

Game Description

Computer City: Sewers is an educational game developed by the author to teach digital logic to students in introductory computer science classes. The game presents a hypothetical city as a metaphor for the computer architecture. In such a city, every aspect of the computer architecture has a corresponding metaphor; 1) buses in a computer correspond to the bus system of the city, 2) addresses in computer memory correspond to the addresses of buildings in the city, and so forth. *Computer City: Sewers* deals specifically with digital logic, which is conceptually the lowest level of the computer architecture [Tanenbaum, 2005]. Thus, the player is placed in the sewers of the city, which are conceptually the lowest levels of the city.



Tanenbaum's Computer Architecture Hierarchy

Storyline

The player takes the role of Bitty, a "bit" within the world of Computer City. Bitty has been placed underground by his/her superiors to collect blueprints of the sewers. In order to collect all of the blueprints, which are manifested as truth tables, Bitty must solve puzzles to open doors and traverse the sewer.

Design

In developing *Computer City: Sewers*, numerous design strategies were employed. These strategies were obtained from research on the common design practices of educational computer game developers. The designs and goals that educators and game designers found to be most important were given the highest consideration in the development of the game.

One of the most fundamental goals discovered in the research is that games should flow concurrently with the learning process [Hoffman, 2004] [Sum, 2006]. The player should be unable to progress through the game without using the knowledge gained from the game's underlying lesson. Thus, game design should be primarily centered around the underlying lesson rather than the game-play or storyline.

In order to achieve the goal of connecting the game to the learning material, the player should be given clear indications of the connections of the game to the real-world applications of the learning material in the game [Hoffman, 2004] [Sum, 2006]. Objects in the game should closely resemble their real-world counterparts, and the vocabulary of the game should closely match that of the real world. Actions in the game should also produce consequences applicable to the real world [Liu and Lin, 2009] [Etu, 2008] [Sum, 2006] [Squire et al., 2008]. Such connections allow the player to understand the learning material outside the context of the game and to use this material in the real world.

To increase player engagement and motivation, games should provide multiple opportunities for players to make decisions. Such decision-making allows the player to both have fun playing the game and feel a greater sense of control [Hoffmann, 2009]. Giving the player flexibility to make decisions also encourages exploration, which increases both motivation and the amount of material learned [Liu and Lin, 2009]. The player will also feel more comfortable confronting more difficult challenges presented later in the game [Hoffmann, 2009]. The player thus takes on a more active role, becoming a producer of information rather than a mere consumer [Etu, 2008]. This in turn helps the player feel motivated to continue playing, which increases the amount learned from the game.

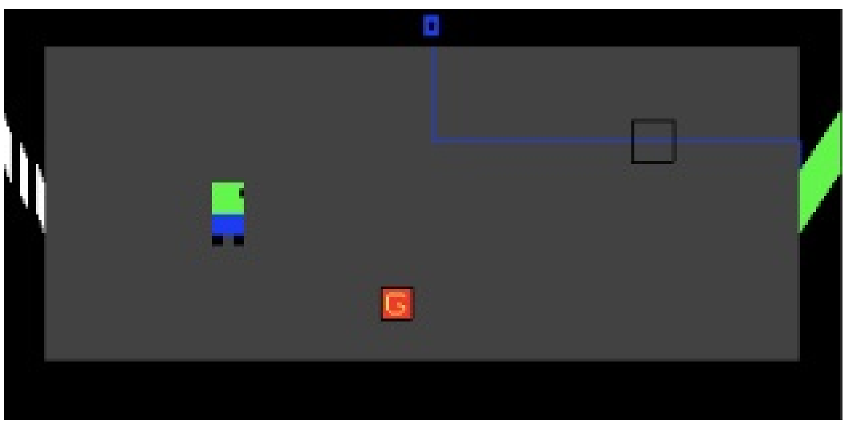
In order to solidify the player's comfort with making decisions, the game should also give constant feedback to the player. When the player makes a decision, he/she should be informed of the results immediately. In this way, the player is able to understand the connection between decisions and their consequences. Understanding of learning material is greatly increased if the player is able to experience such connections rather than imagining them [Squire et al., 2008]. Multiple levels of sensory feedback through multimedia can help the player understand the pattern of decisions and consequences on multiple levels [Guzdial and Soloway, 2002] [Sum, 2006]. Also, the player will feel more comfortable making decisions later if he/she understands the way in which the game presents the consequences. Thus, clear and consistent feedback is important to facilitate the player's learning of the game and its underlying lesson.

As shown, the structure of the game is important because it helps solidify the player's understanding of the game, comfort making decisions, and motivation to continue playing. It is also important that the player be given opportunity to explore the game beyond this structure. An environment that does not allow variation decreases player motivation [Hoffman, 2004]. The player should be able to not only interact with his/her environment, but also to change this environment and experience the results. The ability to manipulate the game's environment increases the player's desire to return to the game and explore it more thoroughly[Etu, 2008]. When the game's learning content is inherent to its structure, this exploration also increases an understanding of the underlying learning material. Thus, the game should allow the player to feel comfortable making changes to the game's structure and experience the outcomes of these changes.

In developing *Computer City: Sewers*, the strategies listed above were all into account. The game strives to have a simple interface that uses real-world representations of objects and to give the player plenty of decision-making opportunities and constant feedback, as well as the ability to customize game settings to fit the learning experience. The player must use what he/she learns about digital logic to progress through the game, and new knowledge builds upon earlier knowledge and can be applied to the real world.

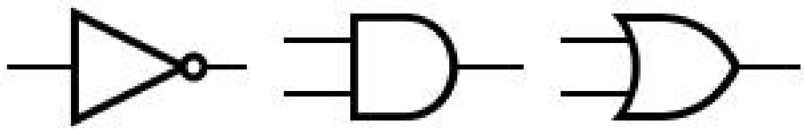
Gameplay

Computer City: Sewers is broken up into rooms, each of which has a puzzle that must be solved in order to open the room's door.



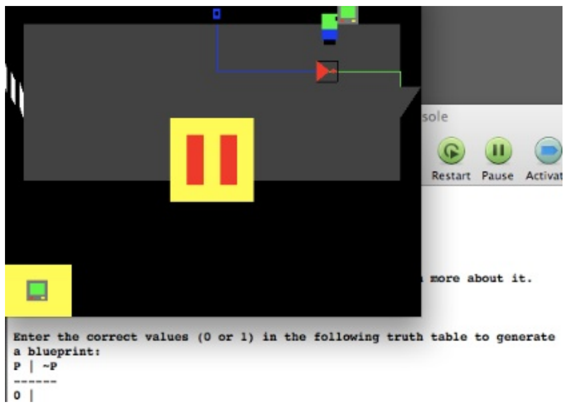
A Room in *Computer City: Sewers*

In some rooms, the puzzle involves running the correct charge into a circuit of gates in order to power the door with the correct charge. In other rooms, the puzzle is to place gates on the ground in order build the circuit that will produce the desired output given an input. In some rooms, the player must assemble a gate out of other gates in order to open the door. In this case a gate assembler screen appears when the player walks next to the assembler object. All gates are pictorially represented according to the IEEE standard [Tex, 1996], so the player is exposed to real-world representations of digital logic concepts.



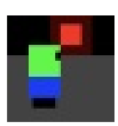
The IEEE Standard Representation of an Inverter, AND, and OR Gate

In other rooms, the player must enter the correct blueprint of the room into a computer in the room. These blueprints are written as truth tables. Thus, the tasks of the game are are associated with real-world applications of digital logic.



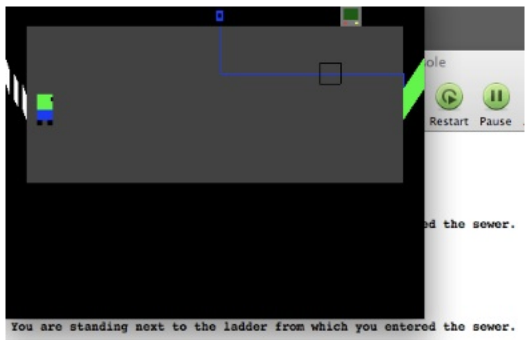
The Truth Table Interface

The player is given control of the learning experience through the avatar, whose exploration of the world is directly linked to the player's keyboard input. The exploration is customizable by allowing the player to change which keys perform which actions and the movement speed of the avatar. Constant feedback of the player's is actions is provided via graphical and textual indications of game events. For example, when the avatar moves next to certain objects in the environment, a yellow screen may appear, at which point the player can click on items in the screen to produce different results. For example, when the player walks next to a "switch button," a button appears in the bottom left corner of the screen. If the player clicks on the button, the charge of the switch next to the button will be flipped, which is indicated to the player graphically.



A Switch Button

The player is shown events as they occur, which eliminates any guessing the player would have to do to determine his/her current situation. In addition to the main game screen, the player is also provided with a text-based terminal to receive information. When game events happen, the player is notified via this terminal. At any time, the player can also press a help button to display information relevant to the current game situation and any objects the player is standing next to. Thus, the player is given constant feedback.



Output of the Help Button on the Text-Based Terminal

Proposed Evaluation

There are numerous methods that can be employed to evaluate the effectiveness of an educational video game. Many game evaluations use human play-testing coupled with pre- and post-tests to evaluate retention of game information [Barnes et al., 2007]. Pinelle, Wong, and Stach provide a heuristic approach with a set of criteria for evaluating a game's effectiveness [Pinelle et al., 2008]. Eric Zhi Feng Liu and Chun Hung Lin employ a Delphi technique [Liu and Lin, 2009]. Michael Eagle uses statistical techniques and data mining to obtain evaluative information [Eagle, 2009].

Computer City: Sewers is evaluated based on the approaches of [Pinelle et al., 2008], [Liu and Lin, 2009], and [Eagle, 2009]. In the future, human play-testing will also be used to evaluate the effectiveness of the game. A pre-test will be administered before each play-testing session to establish player demographics and evaluate each player's experience with digital logic, computer science, and computer games. The players will play the game, which will record statistics such as time spent performing puzzles, number of puzzles and blueprints completed, and number of times the 'help' button is clicked. After the session is complete, a post-test will be administered that will evaluate what the players learned about digital logic and how they found the game in terms of factors such as playability, fun, and motivation. A week after the play-testing session, a followup test will be administered on the same players that will test retention of the information in the game.

The game's ordering of puzzles and rooms can also be tested through play-testing. Different groups of players can be given different orderings of rooms to play through, and the results of data collection and pre- and post-tests can be used to evaluate the best ordering of rooms to produce good flow of the game and the learning material.

The Heuristic Approach

The heuristic evaluation of [Pinelle et al., 2008] takes ten evaluative criteria into account. These are: 1) the game provides consistent responses to the player's actions. 2) the player is able to customize game settings 3) the game provides predictable behavior for computer-controlled characters 4) the game provides unobtrusive views appropriate to the player's current actions 5) the player can skip non-playable content, such as cut-scenes 6) inputs are intuitive and customizable 7) controls have an appropriate level of sensitivity and responsiveness 8) the player is provided information on game status 9) the game provides instructions, training, and help 10) the game's visuals are easy to interpret

Computer City: Sewers strives to fit the relevant criteria in this list. For example, the game is consistent in its response to players' actions. The player always moves in the same predictable and intuitive ways. However, the player can also customize the buttons used in the game as well as the player walking speed. Rooms in the game are shown via a bird's eye view, so the player is able to see the entire range of possible places to move. Whenever a screen comes up within the main screen, it does not obstruct the relevant part of the main screen. The objects' behaviors and visual representations are simple and easy to interpret. Only those objects that are significant are given graphical representations, and they have the same basic functionality each time they are used. For example, switch buttons always flip the values of switches, which change the values of their corresponding circuits. All important game events are reported to the text-based terminal, and any additional information desired by the player can be obtained at any time via the help button. *Computer City: Sewers* also comes with a README that provides a tutorial of game mechanics and game object descriptions. This README can be referenced if the player desires additional help beyond the in-game help terminal. Thus, the player is given plenty of game information and feedback.

The Delphi Method

In another method, Eric Zhi Feng Liu and Chun Hung Lin use a set of evaluative criteria developed by a Delphi technique, wherein a group of educational game developers, educational psychologists, teachers, and students are repeatedly given questionnaires about what criteria are important for evaluating a game [Liu and Lin, 2009]. The answers are read by everyone who answered a questionnaire. The questionnaire is then re-administered based on this analysis, repeating until a consensus is reached. Forty-two experts were given the questionnaire based on analyzing 192 games, and forty-three indicators were produced, broken into five categories: game information, multimedia, interface design and structure, content, and feedback. *Computer City: Sewers* looks to match these evaluative indicators as closely as possible.

Game information refers to the documentation of the game. The game should provide descriptions of hardware and software requirements, a description of the target audience and the lessons that are to be taught. As shown above, *Computer City: Sewers* provides documentation of the game in its README.

The multimedia indicator evaluates whether the game uses picture, animations, and sound to relate to the content of the game and increase the interest of players. *Computer*

City: Sewers uses pictures for gates that match their common representations. The game's graphical appeal is in its simplicity; the avatar is meant to be cute and simple, and the objects are simple representations of digital logic devices. Animation is used only in the avatar's walking graphics. The game also provides text-based explanations of game events, but these are kept short and simple to allow the graphics to be the primary experience of the player. *Computer City: Sewers* does not currently use sound effects, but it could conceivably use ones that correspond to important game events.

The interface design and structure indicator evaluates the clarity and ease of use of the interface. The game should be ordered in a reasonable way, provide an easy-to-understand screen, and show a clear presentation of information. *Computer City: Sewers* provides a simple 8-button interface to navigate the game. The player uses four buttons to move, and there are buttons to display help, settings, and game statistics, as well as a mouse button to interact with certain objects. The screen is kept simple by providing a bird's eye view where only the relevant objects are defined. Important information is communicated via the accompanying text-based terminal, and this information is kept simple, concise, and consistent. The interface is also able to be changed and customized via the settings menu. Thus, the game provides a clear and easy-to-use interface.

The content indicator evaluates whether the learning content matches the game's storyline and goal structure. The player's goals in *Computer City: Sewers* are directly linked to the learning content, as the player must correctly choose inputs, construct gates, and fill out truth tables to advance in the game. The player is able to interact more closely with the content by controlling the in-game avatar.

Finally, the feedback indicator evaluates the game's response to player actions. The game should provide timely, easy to understand feedback. In *Computer City: Sewers*, the player receives feedback during each important game event via the text-based terminal. The player can also request additional feedback at any time via the help button. The current status of the game can also be obtained via the statistics menu. Because a terminal is used to provide feedback, the player is also able to browse back through the feedback history.

The Level Up Approach

In "Level Up: A Frame work for the Design and Evaluation of Educational Games," Michael Eagle advocates the use of statistical techniques and data mining to evaluate the effectiveness of an educational video game [Eagle, 2009]. *Computer City: Sewers* will use this method in the future to evaluate its effectiveness.

One of the measurements in Eagle's procedure is amount of time spent on a given puzzle. This would involve implementing a timer in *Computer City: Sewers* that begins when the player enters a room and ends when the player is able to open the room's door. As Eagle describes, the amount of time the player spends in a given room can be a good indicator of how well the player grasps an understanding of the problem in that particular room. The trend of timings over the course of the game can also be studied to determine the overall understanding of the game and its underlying lessons.

Eagle also considers the number of attempts for a given puzzle a relevant piece of data. Fewer attempts can indicate a better understanding of the puzzle. For *Computer City: Sewers*, this data would be best collected in the rooms in which the player must place gates on the ground and the rooms in which the player must fill out truth tables. In these rooms, the game could count the number of gates mistakenly placed or the number of incorrect values entered into the truth table.

As Eagle shows, performing the evaluation on a puzzle-by-puzzle basis can also determine the best ordering of puzzles within the game. The player should encounter puzzles in such a way that he/she is consistently challenged, but also such that he/she feels confident in his/her knowledge. Such an arrangement of puzzles can be better achieved through the analysis of data from different arrangements of the puzzles.

Future Work

There are many feasible extensions that can be made to *Computer City: Sewers*. These changes will be made with the intent of increasing the quality and quantity of the learning material in the game, the player's interest, and the playability of the game. For example, the game currently lacks any sound. As has been shown by [Belfore et al., 2009], sounds can be especially effective in games about digital logic; the simple binary values can each be given a distinct sound to distinguish them from each other. Sound can provide additional clues to the player as to what behavior he/she are witnessing, and it can help the player understand digital logic in a more immersive setting. Because *Computer City: Sewers* uses simple and consistent graphics to distinguish important behaviors of circuits, simple sounds could also be tied to those behaviors. This would provide a more immersive experience for the player, and it would likely result in better retention of the learning material.

Other areas of improvement will become more apparent when the game is tested. The methods of evaluation described in this paper will be used to determine what and how much is retained by players of the game. Data gathered from play-testing will be used to influence the ordering of game events and guide further development of the game. Areas of poor player retention will be examined, and the game will be adjusted to provide better emphasis on these topics. Successive changes to the game and play-tests of these changes will provide an empirical method for determining which adjustments are best for improving the learning obtained from the game.

Computer City: Sewers can improved by extending the range of topics it teaches. Because digital logic is the foundation of the computer architecture, the game can be extended to evaluate higher-level applications of digital logic. Complex circuits such as flip-flops and adders could be considered in their applications, and the player would be given an opportunity to explore the abstractions that can be made upon the digital logic level.

References

Question and answer with ntiedo etuk, educational video game entrepreneur. *American School Board Journal*, 195(7):11, July 2008.

Summit on educational games: Harnessing the power of video games for learning. Federation of American Scientists, October 2006. URL `http://fas.org/gamesummit/index.html`.

Tiffany Barnes, Heather Richter, Eve Powell, Amanda Chaffin, and Alex Godwin. Game2learn: Building cs1 learning games for retention. In *ITiCSE '07: Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, pages 121–125, New York, NY, USA, 2007. ACM.

Jessica D. Bayliss. Using games in introductory courses: tips from the trenches. In *SIGCSE '09: Proceedings of the 40th ACM technical symposium on Computer science education*, pages 337–341, New York, NY, USA, 2009. ACM.

Jessica D. Bayliss and Sean Strout. Games as a "flavor" of cs1. In *SIGCSE '06: Proceedings of the 37th SIGCSE technical symposium on Computer science education*, pages 500–504, New York, NY, USA, 2006. ACM.

Theresa Beaubouef and John Mason. Why the high attrition rate for computer science students: some thoughts and observations. *SIGCSE Bull.*, 37(2):103–106, 2005.

Katrin Becker. Teaching with games: the minesweeper and asteroids experience. *J. Comput. Small Coll.*, 17(2):23–33, 2001.

Lee A. II Belfore, Amy B. Adcock, and Ginger S. Watson. Introducing digital logic and electronics concepts in a game-like environment. In *SpringSim '09: Proceedings of the 2009 Spring Simulation Multiconference*, pages 1–7, San Diego, CA, USA, 2009. Society for Computer International.

Richard Connelly, Michael B. Gousie, Haldun Hadimioglu, Lubomir Ivanov, and Mark Hoffman. The role of digital logic in the computer science curriculum. *Journal of Computing Sciences in Colleges*, 19(5):5–8, May 2004.

Michael Eagle. Level up: A frame work for the design and evaluation of educational games. In *FDG '09: Proceedings of the 4th International Conference on Foundations of Digital Games*, pages 339–341, New York, NY, USA, 2009. ACM.

Michael Eagle and Tiffany Barnes. Wu's castle: teaching arrays and loops in a game. In *ITiCSE '08: Proceedings of the 13th annual conference on Innovation and technology in computer science education*, pages 245–249, New York, NY, USA, 2008. ACM.

Mark Guzdial and Elliot Soloway. Teaching the nintendo generation to program. *Communications of the ACM*, 45(4):17–21, April 2002.

Mark E. Hoffman. The case for more digital logic in computer architecture. In *ACE '04: Proceedings of the Sixth Conference on Australasian Computing Education*, pages 137–143, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

Leah Hoffmann. Learning through games. *Communcations of the ACM*, 52(8):21–22, August 2009.

Computing Curricula 2001: Computer Science. The Joint Task Force on Computing Curricula, IEEE Computer Society, and Association for Computing Machinery, Dec 2001.

Scott Leutenegger and Jeffrey Edgington. A games first approach to teaching introductory programming. In *SIGCSE '07: Proceedings of the 38th SIGCSE technical symposium on Computer science education*, pages 115–118, New York, NY, USA, 2007. ACM.

Eric Zhi Feng Liu and Chun Hung Lin. Developing evaluative indicators for educational computer games. *British Journal of Educational Technology*, 40(1):174–178, January 2009.

David Pinelle, Nelson Wong, and Tadeusz Stach. Heuristic evaluation for games: Usability principles for video game design. In *CHI '08: Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems*, pages 1453–1462, New York, NY, USA, 2008. ACM.

Kurt D. Squire, Ben DeVane, and Shree Durga. Designing centers of expertise for academic learning through video games. *Theory Into Practice*, 47(3):240–251, Summer 2008.

Kelvin Sung. Computer games and traditional cs courses. *Communications of the ACM*, 52 (12):74–78, 2009.

Elizabeth Sweedyk, Marianne deLaet, Michael C. Slattery, and James Kuffner. Computer games and cs education: why and how. In *SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education*, pages 256–257, New York, NY, USA, 2005. ACM.

Andrew S. Tanenbaum. *Structured Computer Organization*, volume 5. Prentice Hall, 2005.

Overview of IEEE Standard 91-1984: Explanation of Logic Symbols. Texas Instruments Incorporated, 1996.

Ursula Wolz, Tiffany Barnes, Ian Parberry, and Michael Wick. Digital gaming as a vehicle for learning. In *SIGCSE '06: Proceedings of the 37th SIGCSE technical symposium on Computer science education*, pages 394–395, New York, NY, USA, 2006. ACM.