CREATING GAMES AS AUTHENTIC LEARNING IN THE INFORMATION TECHNOLOGY CLASSROOM

Mark Frydenberg
Computer Information Systems Department
Bentley University
Waltham, Massachusetts, USA

ABSTRACT

Teaching students to create computer games has become a common practice in both K-12 and tertiary education to introducing programming concepts, increasing student engagement, and recruiting majors and minors in technology fields. This study describes a project where first-year college students in an introductory technology concepts course use a visual game creation tool to develop original games to play on their computers and mobile devices. The paper argues that the process of making an original computer game develops digital literacy skills and provides an authentic learning experience as students collaborate to create, publish, and deploy interactive games that they can play on their computers and mobile devices.

KEYWORDS

Game Development, Digital Literacy, Authentic Learning.

1. INTRODUCTION

Many educators have turned to game development as a way to engage students with programming and technology concepts. Creating games to play on computers and mobile devices requires specifying rules of play, creating multimedia, posting the files for distribution on multiple channels, and navigating operating system features. This paper contends that completing these tasks allows students to demonstrate many digital literacy proficiencies and provides an authentic learning experience.

1.1 Digital Literacy

The need for today’s students to be digitally literate is growing, and educational organizations worldwide have responded with new curricula to teach digital literacy skills. In the United Kingdom, the Department of Education has teamed up with Microsoft to produce a National Curriculum Guide for K-12 students outlining learning goals. Throughout their education, this curriculum provides opportunities for students to apply principles and concepts of computer science, including abstraction, logic, algorithms, and data representation; to analyze problems computationally; to evaluate and apply information technology, and become responsible, confident, and creators of information and communication technology.(Department of Education, 2014) In the United States, a National Science Foundation Grant has sponsored the development of new curricula for a higher education advanced placement computer science principles course to encourage computational thinking and problem solving skills. (College Board, 2015) The course, to launch in, 2016 will “introduce students to creative aspects of programming, using abstractions and algorithms, working with large data sets, understandings of the Internet and issues of cybersecurity, and impacts of computing that affect different populations.”

The National Computing Curriculum identifies pathways to digital literacy through a student’s understanding of algorithms, programming and development, data and representation, hardware and processing, communication and networks, and information technology. (Department of Education, UK, 2014) Teaching students to create their own computer games has become a popular approach to introducing programming concepts to introductory computer science and technology students. (Mohammed & Mohan, 2010), (Guzdial, 2015), (Bayliss, 2009). (Elliott, 2007). From an instructional perspective, creating games offers opportunities for teaching programming and software development skills on many levels. Games are engaging, allow students to learn by playing, and enable students to turn their ideas into real-world applications. In addition to promoting algorithmic thinking and problem solving, creating their own games presents students with opportunities to be creative, and demonstrate their proficiency as able participants in a world based on digital literacy skills.

1.2 Authentic Learning

Authentic learning pedagogy enables students to engage in realistic tasks using real-life resources and tools, and offers opportunities for students to learn while taking on the roles of professionals dealing with actual problems that arise. (Herrington, 2003). (Herrington, Parker, & Boase-Jelinek, 2014) Authentic learning environments have real-world relevance, are not well-defined, and require investigation of a task over a period of time, and from different perspectives. They encourage collaboration and reflection, and span several subject areas. Authentic activities integrate with each other such that they reflect real world assessment, and allow competing solutions with a diversity of outcomes. They “engage learners in the work of professionals” (Elliott, 2007, p. 34). The instructor fashioned a game creation exercise and contest for students to learn about software development, and required students to publish their games on two different distribution channels, each with its own set of rules and standards. The adaptation of industry-standard procedures as a classroom exercise provides a new way to engage students in authentic learning as they build digital literacy skills and complete the tasks necessary to create and share their games with the world.

1.3 Research Questions

This study shares preliminary results after offering a game development experience to beginning information technology students. Academic goals of the exercise were to provide an authentic learning experience that enabled students to apply digital literacy concepts and skills, and empower t to create computer games that they can play on their own devices.

The study addresses the following research questions:

- Will students with little or no previous experience in developing software succeed in creating games?
- How does creating and deploying games provide for authentic learning as students develop and apply digital literacy skills?

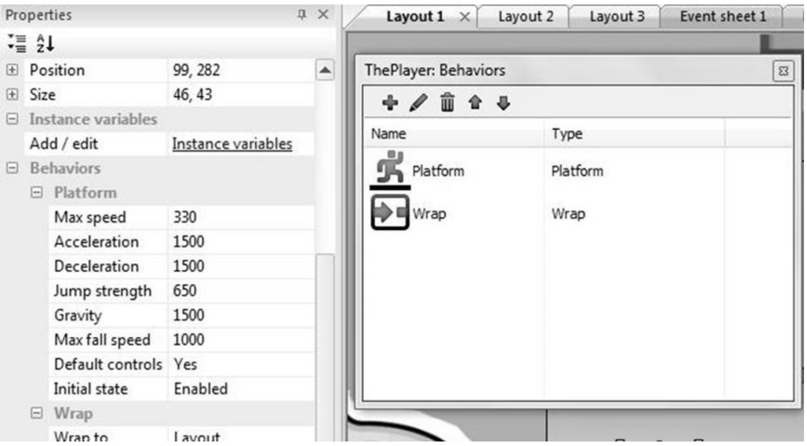
2. CREATING GAMES TO DEVELOP DIGITAL LITERACY SKILLS

This study introduced students to the use of Construct 2 (Scirra.com, 2014), a gaming development tool that enables users to create advanced 2-D games without the use of any programming language. Construct 2's game creator allows users to export their games as web apps by generating HTML5, CSS3, and JavaScript files, and also to generate native app packages for Windows Phone, iOS, and Android, that can be published on the Windows, Apple, and Google Play app stores and marketplaces.

Suitable for beginners who have little or no experience developing software applications, Construct 2 enables students immediately to take on the role of game developers. They design backgrounds, layouts, and objects (characters, platforms, bullets, sounds, and so on) to be used in the game, and specify their properties and interactions during game play. Figure 1 shows the properties of a player object in a 2-D platform game.

Construct 2 uses an event-driven syntax to describe the interactions between these objects. As shown in Figure 2, a rule describes an end-of-game scenario: “When the player collides with the red dot object (Sprite 9), destroy the player and set Lives back to 4 (because the game is over).” The developer would specify similar rules for game play based on input from arrow keys, touch, or tapping buttons on the screen.

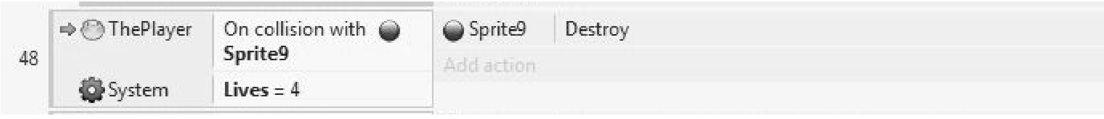
A game must pass several rigorous tests before an app store will accept it for publication. The developer must specify the game’s orientation (portrait or landscape), test the game in an emulator to ensure it functions correctly on a variety of devices and screen sizes, register the game with the app store by finding an available name, provide a series of screenshots and icons at different sizes, package all of the necessary files app, and upload them to the store. Students found this process to be quite complicated, though none of the individual steps involved were complicated on their own.



Position	99, 282
Size	46, 43
Instance variables	
Add / edit	Instance variables
Behaviors	
Platform	
Max speed	330
Acceleration	1500
Deceleration	1500
Jump strength	650
Gravity	1500
Max fall speed	1000
Default controls	Yes
Initial state	Enabled
Wrap	
Wrap to	Layout

Name	Type
Platform	Platform
Wrap	Wrap

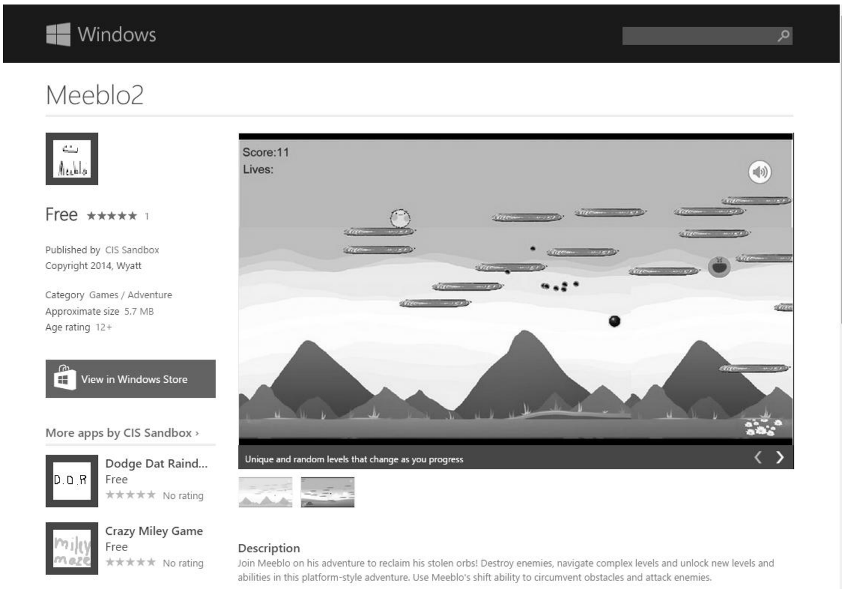
Figure 1. Specifying properties and behaviours of the Player object.



48 ThePlayer On collision with Sprite9 — Sprite9 Destroy
System Lives = 4

Figure 2. Describing a rule in Construct 2.

Figure 3 shows a student's game published to the Windows Store.



Meeblo2

Free ★★★★★ 1

Published by CIS Sandbox
Copyright 2014, Wyatt

Category Games / Adventure
Approximate size 5.7 MB
Age rating 12+

Unique and random levels that change as you progress

Description
Join Meeblo on his adventure to reclaim his stolen orbs! Destroy enemies, navigate complex levels and unlock new levels and abilities in this platform-style adventure. Use Meeblo's shift ability to circumvent obstacles and attack enemies.

Figure 3. A student's game published to the Windows Store.

When preparing an app for the store, the developer may specify pricing information if the app were to be offered for sale (for the sake of this assignment, students could not sell their apps, as doing so would require additional complexities related to setting up merchant accounts); identify an age-appropriate rating, and supply a complete description and game play notes for game testers.

These real-world tasks required to submit a game for publication demonstrate a student's basic digital literacy skills. Students must be adept at interacting with files, creating graphics, converting files from one type to another, and using web-based tools to accomplish a sophisticated task. Completing these steps also gives students additional appreciation of how apps, such as the ones they download routinely to their devices, are published and deployed via an app store.

3. METHODS

This section presents the research context, data, analysis, and discussion.

3.1 Context

The research context was a required course for first year introductory technology students. The study took place in three sections of IT 101 (Introduction to Information Technology and Computing Concepts), an introductory IT course required of all first year students at Bentley University, a business university in the United States. IT 101 teaches digital literacy skills and covers information technology topics, including making use of laptops, productivity and application software, the World Wide Web, computer components and mobile devices, developing web pages with HTML, operating systems, the Internet, image and video formats, and wireless networking. Three academically different sections of the course taught by two different instructors participated in the project.

- An Honors section, for students enrolled in the honors program, chosen because of their high scholastic abilities,
- An Accelerated section, offered to students who self-selected to be this section because of their interest in technology, and
- A standard section open to all students.

Each section met twice a week for 75 minutes each session. Microsoft sponsored the gaming project and provided an academic technology evangelist to lead the training sessions in class. The same Microsoft academic evangelist facilitated the Construct 2 trainings to students in all three sections, and the instructors assisted students during the demonstrations and exercises in their classrooms.

Students installed Construct 2 on their laptops prior to the training. Teaching students to use the software in the classroom required significant support from the Microsoft evangelists who created classroom resources as well as trained tutors to assist students in developing and deploying their games outside of class.

At the first training, students followed a printed tutorial as the presenter demonstrated each step to create a simple 2-D shooter or platform game. Game elements included identifying a player, shooter, layouts, user interface elements for keyboard and touch input, detecting collisions of objects, and creating and destroying objects. These elements are common to many 2-D games.

The second session showed how to include advanced features, such as adding music, keeping score, and transitioning between different layouts for a start screen and a game over screen. Students also learned how to package their games for deployment to the web and to the Windows Store.

For their assignments, students could create their own original game, or adapt the game they worked on in class and add new features such as sound, score keeping, and end-of-game logic. They could work individually or collaborate with a partner in designing and developing their games, but students had to create their own individual games (which could be similar) for this assignment.

Because Microsoft sponsored this gaming training and contest, and because an academic license through Microsoft DreamSpark allows students to publish apps to the Microsoft Windows store without the need to purchase a developers license, this project instructed students to publish their games to the Windows store. (Apple and Android developers must pay an annual license fee to publish apps in those app stores.)

A Microsoft evangelist, tutors in a university computer lab, and students who had taken the course during a previous semester were available to help the current cohort of students with their games during additional "studio time" sessions outside of class. As an additional incentive, Microsoft donated prizes to be awarded to the winners at a games party at the end of the project.

Upon the due date, tutors and former students played and reviewed all of the games from all of the students in the current semester, evaluating them on playability and originality, and whether a game met a checklist of required features. From these reviews, the top 7 games were announced. All students then were invited to a games party where they could play all of the games and vote for their favourite ones. The winners were announced, and prizes awarded.

3.2 Data and Analysis

57 of 85 students (37 male, 15 female, 5 prefer not to say or unanswered) from three sections of IT 101 voluntarily participated in an online survey within two weeks after the gaming assignment ended, to share their feedback about the experience.

Of the 57 responses, 44 were complete and used for this analysis. The remaining 13 students did not answer all of the survey questions used in this study.

Results show that these students use their computers and devices to accomplish daily tasks: 34% of their time online is spent for homework, learning or professional reasons, 14% keeping up with news, 20% of their time is spent communicating or on social networks, 22% of their time they use the web for entertainment including music, videos, and games, and 9% of their time is spent online shopping, banking, or other business tasks.

Despite their being tech savvy, 83% responded that they had not created software applications of any kind prior to the game assignment. 17% of respondents had created software applications in high school, including simple Java, Visual Basic, or C# programs, web pages developed in HTML, CSS, and JavaScript, and games using Flash or other game creation tools. None of the students had used Construct 2 previously. These results suggest that creating games to play on their computers and mobile devices was a new experience for most of the students, and certainly, deploying them to two platforms was new for all of the students.

3.3 Results and Discussion

As shown in Figure 4, survey results suggest that the gaming assignment instilled confidence, captured interest, provided authentic skills in writing programs and apps, and allowed students to reflect on their successful results.

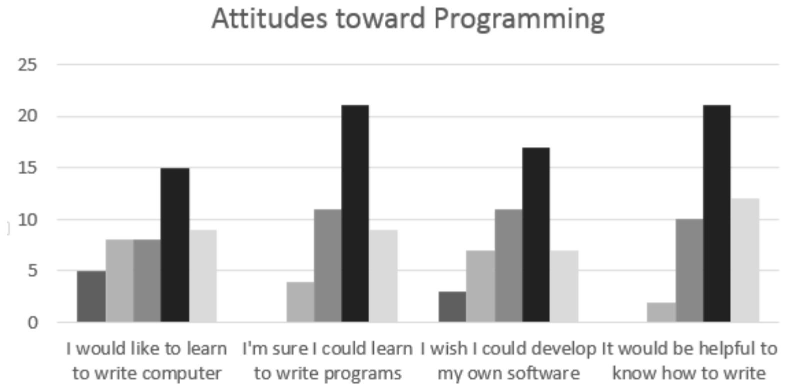


Figure 4. Student attitudes toward programming and creating software.

Said one student, "Games can be easier to make then one thinks." Several student comments reflected the sentiments that "game development is a lot of hard work but it is very rewarding to see a finished project after putting many hours of development into it. I learned to have a final vision but also think step-by-step to make that vision come to reality." Other students got lucky in the process of creating their games: "if you miswrite part of a code, a function you did not expect occurs…You gotta go with the flow if something unexpected works."

3.3.1 Game Development as Authentic Learning

"An authentic, challenging task is the starting point. Authentic tasks are completed for reasons beyond earning a grade. Students also see the activity as worthwhile in its own right." (Means & Olson, 1994, p. 15) The survey asked students their opinions of the gaming assignment as an authentic learning experience.

"Authentic learning requires students to reflect on their experiences and draw conclusions from their findings." (Herrington, 2003). In their reflections, many students remarked that the assignment was "real", gave them a taste of building apps, developed and applied their IT 101 skills, and most were pleased with the games the created, as results suggest in Figure 5.

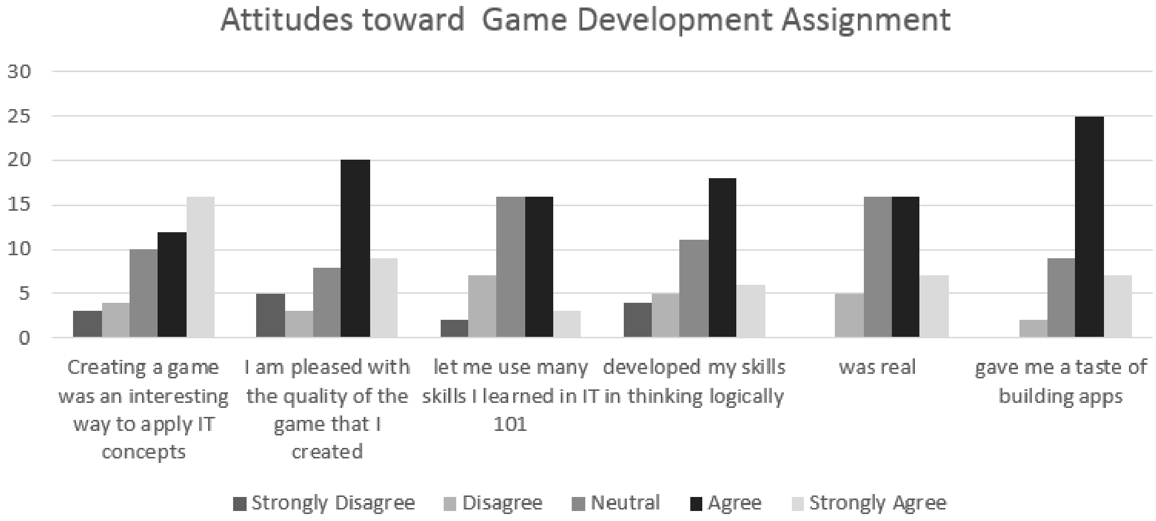


Figure 5. Attitudes toward game development assignment as authentic learning

Results were mixed as to whether students found the experience authentic based on the survey aspects above. Of those who were not neutral in their responses, most agreed or strongly agreed that it gave them a taste of building apps and made them want to learn more about programming. In an open-ended response, students commented that they learned to use Construct 2 and Brackets (an HTML editor), and how to upload their games to the Microsoft store. They learned "to do in depth research on a topic and how to make my own game," that solving problems by walking through them are the best methods to solve computer issues," and one claimed that the assignment allowed the student to "implement higher thinking and creativity into an IT aspect." Another student remarked, "The IT field is a lot harder than it look to be from the outside - but in the end you've always got an opportunity to create something beautiful." For many students, creating a game was the most technically complex task they had ever completed.

Table 1 summarizes ten characteristics of authentic learning environments (Herrington, 2003), and describes how each is manifest in the Gaming Assignment.

Table 1. Authentic Learning Characteristics as Evident in the Gaming Assignment

Authentic Learning Characteristic	As Evident in the Gaming Assignment
Problem has real-world relevance	Students create a game that runs as a web app or a Windows app. Students see the relevance of creating software that runs on their computer or mobile device. The exercise mirrors the requirements of a real-world activity.
Problem is open-ended	Students have the flexibility to use the tools they know in order to design their games, and must apply the concepts learned during the classroom trainings or perform research to find additional tutorials or examples on which to base their own games. Students must ascertain the effectiveness of their choices as they try to implement their own ideas to modify existing games or create new ones.

Task to be investigated over a sustained period of time	The gaming assignment runs over a period of four weeks, so students have time to learn basics, explore alternative solutions, get support from tutors and peers, and realize the trajectory of their progress. The project is divided into smaller, manageable steps that allow students to reach these milestones.
Allow for exploring a task from different perspectives	Students involved in the project are in their first year of college, many have not done any programming or development before. Students with a more technical acumen were able to add complexities; others looked at specific tasks or features such as adding score keeping capabilities, and were able to apply the same steps they did in class example to their own games.
Provide for Collaboration	While students were required to create their own games, they were encouraged to work in pairs to share ideas, and review each other's work. This proved to be effective as students were able to support each other in their learning, show each other how to accomplish various tasks that they remembered that their partners did not. Students used appropriate online collaboration tools to share screens, graphics, or other files with each other.
Provide an opportunity to reflect on the experience	The goal of the exercise was to learn about creating software by developing an actual game, and then deploy it to two different distribution channels. By developing, debugging, and playing their games they reflect on the logic and scenarios they implemented, and how the steps they specified produce the desired game play. By playing their games and showing them to others, students had an opportunity to talk about their work, and reflect on ways that they might want to improve it in future revisions.
Seamlessly integrate with assessment	Not only the instructor determines the quality of a student’s game; an independent evaluation takes place through the process of submitting the game to the Windows store. Each step in the submission process requires that the correct files are in place, and descriptions and documentation follow specifications. Microsoft testers play each game prior to accepting it on the store, testing for touch input and other features. Games accepted to the store received an automatic 'A' because they passed these stringent requirements. Grades for this assignment considered the playability, difficulty, and originality of their games, along with the status of their game's acceptance on the Microsoft store.
Create polished products in their own right	The games that students created were completely functioning software apps that they could run on their computers and mobile devices. Many featured touch input, music, score keeping, animations, and other characteristics found in real games. Not only did students create games, they had to deploy their games correctly so others could access them on their devices.
Allow competing solutions and a diversity of outcomes	While all students completed the same in-class training, the games they created were all different. Even variations on the same in-class example generated a variety of results and outcomes. A games contest gave added incentive to produce a high quality final product.

4. CONCLUSION

This study presents preliminary results based on one semester's implementation of a gaming project.

Future research will aggregate survey results over a period of several semesters to further study how the process of creating and deploying games for play on computers and mobile devices encourages digital literacy skills and impacts student learning.

Initial findings suggest that students found the gaming assignment offers an opportunity for students with no prior programming skills to create software within a controlled and supportive environment. It allows them to demonstrate their understanding of coding principles, including identifying objects and interactions, and that creating software requires a developer to specify exact instructions for the computer to follow.

Students learned to represent different types of data and information (scores, sounds, characters in their games) in a digital context; and to design games for multiple devices and input sources, from touch to keyboard. The assignment provides students an opportunity to reflect upon their games and consider enhancements for future revisions. They also must consider ethical implications (suitability of content, amount of violence in games, and use of open-source sounds and graphics) when building their games. The exercise allowed students to experience the role of a game developer and at the same time, create apps for their computers and mobile devices.

ACKNOWLEDGEMENT

The author acknowledges Michael Cummings and Gavin Baumann, Microsoft Academic Evangelists, for their enthusiasm and continued support of this project.

REFERENCES

Bayliss, J.D., 2009. Using Games in Introductory Courses: Tips from the trenches. *ACM SIGCSE Bulletin - SIGCSE '09*, pp.337-41.

College Board, 2015. *Computer Science Principles*. [Online] Available at: http://apcsprinciples.org/ [Accessed 21 February 2015].

Department of Education, UK, 2014. *Quick Start: A CPD Toolkit for Secondary Teachers*. [Online] Department for Education Available at: http://www.quickstartcomputing.org/ [Accessed 21 February 2015].

Department of Education, 2014. *Computing programmes of study: key stages 1 and 2*. [Online] Available at: http://www.quickstartcomputing.org/secondary/Resources/section2/Computing%20programmes%20of%20study%20Key%20Stages%201%20and%202.pdf [Accessed 21 February 2015].

Elliott, C., 2007. Action Research: Authentic Learning Transorms Student and Teacher Success. *Journal of Authentic Learning*, 4(1), pp.34-42.

Guzdial, M., 2015. What's the best way to teach computer science to beginners? *Communications of the ACM*, 58(2), pp.12-13.

Herrington, J., Oliver, R. & Reeves, T.C., 2003. Patterns of engagement in authentic online learning environments. *Australian Journal of Educational Technology*, 19(1), pp.59-71.

Herrington, J., Parker, J. & Boase-Jelinek, D., 2014. Connected authentic learning: Reflection and intentional learning. *Australian Journal of Education*, pp.1-13.

Herrington, J., Reeves, T., Oliver, R. & Woo, Y., 2004. Designing authentic activities in web-based courses. *Journal of Computing in Higher Education*, 16(1), pp.3-29.

Means, B. & Olson, K., 1994. The link between technology and authentic learning. *Educational Leadership*, 51(7), p.15.

Mohammed, P. & Mohan, P., 2010. Combining Digital Games with Culture: A Novel Approach towards Boosting Student Interest and Skill Development in Computer Science Programming. In *2010 Second International Conference on Mobile, Hybrid, and On-Line Learning*., 2010. IEEE Computer Society.

Scirra.com, 2014. *Construct 2*. [Online] Available at: http://scirra.com/construct2