Komputer Kingdom: An Online Gaming Environment to Build Foundational Computer Science Skills

Dept. of CIS - Senior Design 2013-2014

Kaycee Anderson
kaycee@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Juan José López Delgado
juanlop@sas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Caroline Ho
hoca@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Johanna Martens
martensj@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

ABSTRACT
Research has shown that a majority of children and young teenagers perceive computer science and computer science related careers to be boring, difficult, and anti-social in nature, despite the fact that this age group spends a large amount of time connected to the internet through social media, hand-held devices and video games. Several approaches and technologies have been developed in attempts to make computer science more accessible to younger ages and easier for them to learn and understand fundamental computer science concepts. These approaches have recently gained success and accreditation throughout the computer science community in teaching students who already have some interest or motivation to pursue the field. www.komputerkingdom.com is a web based game, that aims to bridge the gap between students who have no initial interest or experience with computer science and those who want to learn how to program. The goal of the website is not to teach children how to program or write code. Instead, the website aims to use a series of mini-games to teach basic topics fundamental to computer science through problem solving and experimentation techniques. In a small sample of students given a basic survey, the data supports a proof of concept that by playing the games the students surveyed demonstrated an increased understanding of a certain skill.

1. INTRODUCTION
“Over 87% of youth in America between the ages of 12 and 17 use the Internet and usage spikes from 60% in the sixth grade to 82% by seventh grade”[8]. Nonetheless, this age group also reports little interest and a lack of enthusiasm in pursuing computer science academically or as a career path. [8] Graduate students who were doing work in computer science reported not being formally exposed to the field through high school courses, but that they often developed their interest while playing games online. [8] In the current era of technology the number of computer science careers and jobs available is exceeding the number of students graduation with computer science degrees, and 9 out of 10 schools do not even offer computer science courses. [2] These facts are simple motivation for addressing the need to foster students’ interest in computer science, especially at a young age.

Not many students are exposed to computer science in their early years. Consequently, they are not getting the relevant exposure or developing the thought process and skills that are fundamental to foster a greater interest in the field. Nevertheless, children at a young age are already being exposed to computing in various aspects of their lives through the internet, hand-held devices and personal computers. They may not know the inner workings of such devices, but they are still drawn to using them and have a high interest in them. Some research suggests that presenting educational material through computer games is more effective at motivating students to learn that material than they would be through a non-electronic approach [4].

Komputer Kingdom was created as a web based game that aims to provide a fun setting outside of the standard teacher-student classroom setting where students will be able to build their skills related to basic concepts in computer science through games and activities using patterns of gameplay with which they are already familiar. An attempt is made to address the issue of interest by disguising learning as fun. This was achieved through designing new electronic games and activities. These games were partially inspired by some activities that currently exist as offline educational tools from CS Unplugged [1], but were reimagined and revamped to be fun mini-games that encourages learning through individual problem solving and exploration. The website also provides a section that directs users who like Komputer Kingdom to other websites that will teach them actual programming and give them an in-depth and thorough understanding of more computer science concepts.

Upon first exposure, students frequently find traditional programming languages too difficult due to their specific syntax, complex nature, and overwhelming capabilities [3]. Due to these perceptions they are often deterred from further studies in the field [3]. Reasons like these are why Komputer Kingdom involves no programming, so a user’s first exposure will instead focus on skills and basic concepts neces-

*Advisor: Stephanie Weirich (sweirich@cis.upenn.edu)
sary to build a good foundation for learning a programming language, should he/she choose to pursue that option. Independent from a student’s career path, the problem-solving skills and analytical capabilities that an education in computer science provides is an important asset in multiple fields and across disciplines. Komputer Kingdom is a bridge between students with no interest or with a negative perception of computer science and those who would like to take the next step in the learning process. A short pre-survey and post-survey were administered to a small group of middle school respondents before and after they were asked to play on Komputer Kingdom. The trend in the data collected shows that after playing the mini-games on the website the students’ overall demonstration of the concepts being taught increased after playing around on the website. This shows a proof of concept that middle school students have the capacity to learn basic computer science concepts through games such as the ones present on the website.

2. RELATED WORK

Caitlin Kelleher and Randy Pausch conducted research on a range of Programming Environments designed specifically for novice programmers. Their research was conducted with the intent of presenting a taxonomy of theses languages and environments [5]. The article presents three challenges to learning programming that tend to overwhelm and discourage the novice programmer. These are: forming structured solutions to problems, understanding how programs are executed, and mastering rigid syntax and rigid commands. The formation of structured solutions to problems is similar to understanding how to break down a problem into its simplest components, or how to turn large tasks into smaller individually manageable subtasks. Unlike forming structured solutions, which is not unique to programming challenges, understanding execution concerns, specifically understanding how programming “tools” (e.g. variables, loops, functions...) are used to accomplish goals and understanding how these “tools” work. Finally, mastering rigid syntax and commands is the actual act of writing code, taking ideas and seamlessly translating them into symbols and commands on a page. While the article discusses broadly the range of all environments intended to teach or simplify programming (to all ages), the target audience of Komputer Kingdom is children age 10-15, middle school. Beginning programmers have difficulty translating their intentions into statements. The article mentions two possible avenues for making this process easier. The first is to design learning languages, which factor out the syntax challenge and focus on problem solving. The second is to find alternate ways for beginners to communicate their instructions to a computer.

Some examples of the former avenue (learning languages) were Basic, a relatively successful language in its time designed to support a small set of instructions and remove “unnecessary” syntax; Junior Java, designed to remove a lot of the syntactic complexity of Java, which allowed students to focus on the concepts of programming; and Cornell Program Synthesizer, designed to draw attention away from syntax and toward concepts by adding predefined “templates” for programming statements.

Kelleher also delved into some alternatives to typing programs which were mostly pictorial based and aimed at removing any and all syntactical challenges. Most notable of these was Pict. Pict, created at the University of Washing-

2.1 Pex4Fun

Pex4Fun from Microsoft Research is a “web-based serious gaming environment for teaching computer science”. In addition to providing a learning environment for students through a gaming experience, it also aims to include a “unique social experience, tracking and streaming progress updates in real time” [10]. Some differences between Komputer Kingdom and Pex4fun include reference to the “gaming environment”. Part of the homepage of pexforfun.com has C# code that the user is expected to expand on. Pex4Fun describes this as an interactive Coding Duel puzzle. Though the puzzle is technically an interactive game, the required use of code in the game can be intimidating and off-putting to a young student with no prior coding experience.

Another difference is the intended use of the Pex4fun social environment. Instead of providing an exclusive social experience for the students, Pex4Fun is meant to also connect teachers and curriculum authors with the students. Though the social environment is online, it is still fundamentally built upon the traditional student-teacher interaction. The conceptual worlds of the two generations differ greatly; unlike their “digital immigrant” teachers, today’s young students grew up immersed in digital technology. As such, these students are beginning to believe that games are their best teachers, and they are finding that collaborating with fellow “digital natives” is more intellectually stimulating and rewarding [8].

2.2 Scratch

Scratch, developed by the MIT Media Laboratory in collaboration with Dr. Yasmin Kafai’s group at UCLA, is described as “a visual, block-based programming language designed to facilitate media manipulation for novice programmers” [9]. It aims at teaching fundamental computer science skills including: user interaction, loops, conditionals, boolean logic, communication/synchronization, variables, and random numbers. However, it focuses on keeping kids online. In addition to the actual Scratch program that students download, MIT has also set up a website http://scratch.mit.edu/ that acts as as a creative learning community where students can share their projects with one another [6]. MIT recognized the importance of an online environment for teaching today’s students, and that is arguably one of the main reasons Scratch has become such a popular program among students. In addition, as can be seen from Figure 1, MIT focused on developing a very visual environment. Those key computer science concepts are still there, but there is no written formal code. Instead, there are command blocks that students can drag and drop, and they fit like puzzle pieces. This less intimidating method
of programming is conducive to a learning environment for novice programmers.

Though Scratch’s online social environment acts as a better model than Pex4Fun’s, as it emphasizes student interaction instead of student-teacher interaction, Komputer Kingdom does not have a social aspect. Instead of student collaboration, Komputer Kingdom tracks completion and individual high score, to promote a more individualistic structure whereby students are encouraged to focus on the learning aspect of the games in order to succeed, rather than completing them just to achieve a reward.

2.3 Computer Science Unplugged

Computer Science Unplugged is a “collection of free learning activities that teach Computer Science through engaging games and puzzles that use cards, string, crayons and lots of running around”[1]. Through the collection of activities targeting ages 5 to 12, the aim of Computer Science Unplugged is to engage groups of students in large pre-planned activities. Those pre-planned activities are found in the form of downloadable workbooks from their website. The activities are designed to teach basic fundamental concepts of computer science that are often the first things students need to learn to be successful programmers. Some of the activities involve teaching the concepts of binary numbers, sorting algorithms, minimal spanning trees, public key encryption, finite state machines, searching algorithms, image representation and more[1]. Each of these activities requires an instructor who puts together a group, gathers the often necessary physical supplies (crayons, string, water, crates) and printouts of the workbook and plans a lesson accordingly. Then in one setting, in teacher student format, the instructor runs through the activity with the students. Computer Science Unplugged has found that this approach engages kids in an offline environment and stirs up an interest in the computer science field, while teaching basic problem solving concepts to those who may not be interested in computer science, but still find the activities engaging.

The main focus of Computer Science Unplugged is to keep kids offline while learning the basics. Komputer Kingdom has borrowed the concepts of binary numbers, binary search, finite state machines, and image representation from the well established CS Unplugged, and designed games around those concepts to be played online at moments notice, no set up required. Results from studies indicate that there are learning benefits in video games where the educational elements are embedded within the design of the games. Students are also better able to pace themselves and take advantage of the modular aspects of games; the ability to take charge and have control over navigation is critical to enhancing motivation and learning. [7] Further studies have also reported that computer games were much better at stimulating learning and motivation than their offline counterparts. [4]

3. SYSTEM MODEL

3.1 Game Flow

From a user perspective the website navigation is very simple. As can be seen in Figure 2, the journey into Komputer Kingdom begins on the landing page. This homepage has links to an information page for parents, a sign-up link and a login tab. The information page contains everything parents need to know about the website, including security and privacy concerns, an overview of the website’s mission and a description of the games and concepts they will find within the website. The registration link is where students can create and customize their profiles that they will need to keep track of their progress and login to the website. The main focus of the homepage is to serve as a login screen.

Once the user logs in, he/she is taken to the main welcome screen, where the user can find the gameboard, links to his/her personal profile, the user’s scoreboard and more information on computer science. The main gameboard Figure 3 is an interactive map of Komputer Kingdom, with clickable mini-games lining the main path. When a user clicks on one of the mini-games, the selected mini-game loads on the gameboard frame, replacing the gameboard. Upon completion of the mini-game, the user is prompted whether he/she would like to play again or return to the main screen. Komputer Kingdom is a fantasy land where the user plays the role of an animal character. The medieval fantasy theme with animal characters was chosen because it is gender neu-
Figures 3: Gameboard. The gameboard that is the home-screen for interaction with the mini-games.

The gameboard that is the home-screen for interaction with the mini-games. This also provides further abstraction from comparison with Komputer Kingdom to any specific historical or fictional society, leaving its interpretation to the user’s imagination.

Komputer Kindrom has two main character hosts, a green dragon named Draco, that guides the students through registration, login and sign-up, and a pink anthropomorphized bean, King Poroto, who helps the player throughout his/her adventure in the kingdom by giving him/her instructions, information on computer science skills and motivational advice during the each of the mini-games.

3.2 Mini-Game Description

The website consists of four different mini-games that teach and exercise four different foundational topics in computer science: binary numbers, binary search, finite state machines and image representation. Each mini-game has its own clickable icon on the gameboard with an image that represents the nature of the game. Once a mini-game is chosen, it is loaded on the gameboard frame and detailed instructions are displayed. These instructions teach the player how the game works and the reasoning behind it, and also offers some guiding questions for the users to think about.

3.2.1 Binary Numbers

The binary number mini-game aims to teach the player basic binary number representation. The game itself consists of five cards, each with a different number of dots which represent powers of two (1, 2, 4, 8, 16), the binary representation of the dots displayed, a one minute timer and a decimal number that has to be matched Figure 4. The objective of the game is flip the different cards up or down to match the randomly generated decimal numbers as many times as possible in a one minute time period. Flipping a card corresponds to flipping a bit at that position in the binary number. This changes the binary number representation that the user sees and, in this way, teaches him/her how the number of dots and the binary number corresponds to the decimal representation above.

3.2.2 Binary Search

This mini-game teaches Binary Numbers through counting dots and turning over cards to have the correct number showing.

The binary search mini-game tries to subtly teach the user the efficacy of using binary search when looking for an item in a sorted array. The game consists of an array of thirty sorted bubbles, numbered zero through twenty-nine Figure 5. The objective of the game is to guess the randomly generated number that the game character is thinking of, in as few tries as possible. Every time the user makes a guess, the character will tell the player whether his/her guess is higher or lower than the target, and then all of the bubbles that fall outside of this range will be greyed out, as they are no longer a possible guess.

3.2.3 Finite State Machine

The finite state machine mini-game is supposed to help users reason about patterns as a sequence of characters at a basic level. The game includes seven islands on an ocean, a boat, and an ‘A’ button and ‘A’ button. Six of the islands are labeled 1-6 and the seventh island has a treasure on
Figure 6: FSA. This mini-game teaches how a Finite State Machine works through experimentation trials and memorization.

The object of the game is to reach the island with the treasure. Users can press either ‘A’ or ‘B’ to move from one island to another. However, there is no reference as to which island you will transport to when hitting a specific button. Therefore, users must attempt to remember the sequence of button presses that will take you to a specific island. This is meant to correspond to a finite state machine following an input sequence.

3.2.4 Image Representation

Figure 7: Image. This mini-game teaches Image Representation through a grid coordinated image with large square pixels.

The image representation mini-game tries to teach users the idea behind pixels in a two-dimensional grid and how that corresponds to an image on a screen. The game consists of a 10x10 grid filled with red, blue, yellow, and white squares that represent pixels. Rows and columns are numbered from 0-9 and the coordinate systems follows the standard system for image representation, with column numbers increasing in the rightward direction and row numbers increasing in the downward direction. To the right of the grid is a list of coordinates, which are displayed in different colors corresponding to the different colored squares. Users are supposed to drag the squares to their designated coordinates. If the user puts each square in the correct location, they will see an image of a smiley face.

4. SYSTEM IMPLEMENTATION

Komputer Kingdom can be broken down into three main technological components: the client side, the web server, and the MySQL database. These components and their relationships can be seen in Figure 8. This section will explain the role each component played as well as the specific technologies used within each component.

Figure 8: Main Technological Components

4.1 Client Side

The client-side was developed using HTML, CSS, JavaScript (including JQuery), and AJAX, four core technologies were used in front-end development. HTML, CSS, and JavaScript were used consistently throughout the whole website, and the outside front-end framework Bootstrap was leveraged. It includes design templates for interface components to allow for quicker and easier front-end development.

The four games relied heavily on JavaScript. A couple of outside resources were also used to aid in development. Gridster is a JQuery plugin for drag-and-drop multi-column grids, which was used to help implement the Image Representation game. Lightbox, an outside script for image overlay, was used to display the instructions for each game. The games also included AJAX to facilitate data exchange between JavaScript and PHP. This was necessary to allow for the user’s score at the end of each game to be sent to the database. AJAX was also used in the high score table (accessible through the Level link on the Dashboard) to allow for user’s high scores to be dynamically displayed on the page.

---

1 getbootstrap.com
2 gridster.net
3 http://lokeshdhakar.com/projects/lightbox2/
4.2 Web Server

A LAMP (Linux, Apache, MySQL, PHP) solution stack was installed on top of an Amazon Elastic Compute Cloud (EC2) instance. Linux was the specific operating system used, Apache worked as the HTTP web server, MySQL served as the database management system (explained in the subsequent section), and PHP was the server side scripting language. The ubiquitous LAMP stack was used because it provides a well-established solution for building dynamic websites. It has the capacity to serve many requests simultaneously, and it has been well tested for performance and security. Amazon EC2 Web Services was leveraged because it can be launched quickly and easily, it integrates well with a LAMP stack, and Amazon provides many resources and documentations for their services.

4.3 Database

MySQL was the database management system hosted on the server. It was chosen because it is one of the most widely used open-source relational database management systems and has a plethora of outside resources for developers. It also integrates very nicely with PHP, and phpMyAdmin, a free software tool written in PHP, was used to handle administration of the MySQL database through a web based GUI.

The database contains tables to store user data (including usernames and passwords) as well as each user’s high score for each game. To protect user passwords, sha512 was leveraged to hash the passwords, as well as a salt to act as additional data input for the one-way hash function. A table storing login attempts for each user was created as additional security to help protect against brute force attacks.

5. RESULTS

There are two main components to evaluating Komputer Kingdom’s performance in relation to its goals. The first component is the evaluation of how well it improves participants’ understanding of basic computer science concepts. The second component is the evaluation for the level of user enjoyment and change in interest resulting from interaction with the website.

Two electronic surveys were created to quantitatively capture data about user reaction to Komputer Kingdom. A copy of these surveys can be found in the appendix. Volunteers were asked to first answer all of the questions on the Pre-Survey for KomputerKingdom.com, then log in and play all of the mini-games on the website, then finally they were asked to complete a Post-Survey for KomputerKingdom.com which modeled similar questions to the pre-survey. Core data was collected for the intended audience of 10-15 year olds with the sample size being 13 respondents to fully complete the requested tasks. Additional data for comparison was collected for 16 voluntary respondents of age strictly greater than 15 years old.

An analysis of the data collected in regards to specific skill evaluation improvement in basic computer science concepts for the sample of 10-15 year olds can be seen in Figure 9. When asked a question about binary number recognition before game play only 53.8% of respondents answered correctly, while 76.9% demonstrated that skill after game play. When asked to identify coordinates of pixels in an image 0% were able to do so correctly at first, but after game play 30.8% of respondents were able to identify the correct coordinate location. In response to a question about efficient binary search methodology 53.8% of those surveyed correctly responded to start in the middle, while after game play that number increased to 84.6% getting the answer correct. When confronted with a question about following commands in a finite state machine before game play 38.5% of participants answered correctly, compared to the increased 76.9% that were able to do so correctly after game play. All of the core data that was collected shows an increased average ability to demonstrate understanding of a specific skill after the respondent was able to play an associated mini-game on KomputerKingdom.com. This across the board increase in correct responses related to questions about the 4 basic skills that were taught in Komputer Kingdom effectively show a proof of concept that middle school aged students have the capacity to learn through this environment of playing exploratory problem solving games.

By collecting similar data for a sample of respondents with age greater than 15, it can be quantitatively reinforced that middle school aged children were the correct intended audience for the goals of Komputer Kingdom. As can be seen in Figure 10 the trend for older participants shows that many of these concepts are trivial to them because there is initially a greater ability to demonstrate the skill without game play and their ability to demonstrate the skill correctly after game play was not always shown to improve or was shown to remain constant with their pre-survey responses, with the one exception being the image representation skill. In this specific skill of image representation in both age groups neither was very good at demonstrating this skill before or after game play, but did show improvement overall after game play. While this can be considered improvement and

\[\text{http://aws.amazon.com/ec2/}\]
\[\text{http://www.phpmyadmin.net/home_page/}\]
\[\text{http://pajhome.org.uk/crypt/md5/sha512.html}\]
demonstration that the games on Komputer Kingdom teach specific skills, it should be noted that neither group reached over 50% of respondents answering correctly. The intended target audience of middle school students age 10 to 15 years old showed overall greater improvement than the group of respondents of age greater than 15. This reinforces that Komputer Kingdom is targeted to the right audience for teaching these sorts of basic skills.

The second component that Komputer Kingdom subtly addresses is the level of enjoyment and interest that users take away from playing on the website. In the pre-survey the sample of 13 middle school aged students rating on a scale from 1 to 5 their interest in working with computers in the future, 1 represented low interest and 5 signified high interest. The initial average of all respondents was 2.54. In qualitative terms a rating of 3 would mean that the users in the sample felt neutrally about computer science. Because this initial rating is less than 3, it is evident that the majority of those surveyed were not really interested in working with computers in the future. However, after experiencing Komputer Kingdom, the average interest measured in the survey response increased to a rating of 3.31. This tips into the good side of the scale where the majority of users rated their interest greater than before. While the disparity between these two numbers is small and both values straddle neutral closely, this small increase demonstrates a proof of concept showing that experimenting with computer science through problem solving games has the potential to increase interest in the field.

Lastly the survey was able to collect data on how much the users enjoyed playing on Komputer Kingdom. The rating scale was from 1 to 5 with 1 being not enjoyable and 5 being super enjoyable. The average reaction to Komputer Kingdom produced enjoyment rating of 3.69. This number is slightly above neutral showing that of the small sample most users enjoyed the games, even if they only enjoyed it moderately, the data shows that they did not hate it.

Based on the data collected it can be seen that Komputer Kingdom exists as a weak proof providing insight into the field of using games to teach middle school children basic computer science concepts. The data supports the findings of related work in this field showing that games are a viable medium for education. Using those basic principles of research to design computer science related games was a novel and non-trivial concept. Komputer Kingdom selected 4 basic computer science skills and made games available to teach those skills. The data collected shows that Komputer Kingdom played a role in teaching those skills to participants that played the corresponding skill building games through an enjoyable an interest peaking overall experience.

6. FUTURE WORK

Komputer Kingdom currently exists as a standalone website that demonstrates a proof of concept about teaching computer science to middle schoolers. There is room in this model for expansion and improvement.

Feedback and comments left in the surveys stated that the games might be better if they encouraged gradual learning of the concept instead of the all at once fashion. This could be achieved through breaking up the game design into smaller iterations or levels. The design could begin with a the smallest and simplest goal for just introducing the skill and then progress to more challenging and complex tasks. This also would be good to enforce repetition and practice of a skill. Repetition is missing from the current design. Komputer Kingdom as is behaves in a manner of one and done. It would be nice to find a design that would keep middleschoolers coming back to the site. This might also increase the enjoyability factor for the users. They might better understand what is happening skill wise within the games and not lose enjoyability due to confusion and the perception that something is too hard or will take too long to finish.

As Komputer Kingdom currently only teaches 4 basic computer science skills there is plenty of room to expand the skills that are taught through designing more new and engaging mini-games. Some challenges with this include figuring out what basic computer science concepts are able to be transformed into mini-games and what those educational mini-games could possibly look like. Some concepts are too complicated to easily lend themselves to be transformed into a short and educational game. Others are too simple and a game could be made, but provides no challenge or enjoyment to a user. More research should be done on what makes a mini-game enjoyable and educational specifically related to computer science concepts. Some current hypotheses include being solvable quickly, being simple, and not directly referencing any computer science concept, even though that is what the game really teaches.

Due to the positive results shown in Komputer Kingdom, it is worth pursuing more analysis and further design to enhance the overall concept that has been created in this realm. Good feedback has been given about the use of bright colors and images on the website as well as having the mini-game structure as part of the teaching goals. Any future work should try to fit within this model and expand on what has been demonstrated with Komputer Kingdom.
7. ETHICS

While it is necessary for any website of this nature to recognize and capitalize on the business side of the application in order to grow and improve it, there is still the danger (especially in the case of building a platform for kids) of being too greedy and/or money driven. From its inception, the motivation behind building Komputer Kingdom was to give younger kids exposure to basic, fundamental computer science concepts through a fun and easily accessibly medium. The center point of this motivation is that the site is meant to be easily accessible and fun, and had no intention of making money. And while the site’s business side might be necessary for helping it grow, it should be heeded that the site was never intended to be a business.

The biggest ethical issue that Komputer Kingdom as a platform faces by far stems from it’s role as an educational platform intended for pre and young teenagers. While the whole point of catering to children within this age range is to catch them during a period in their lives when their minds are most malleable and open to absorbing information, it also presents the enormous ethical responsibility of control over what that imparted information is. Under the premise of being an educational platform especially, as the site gains recognition and credibility, advertising companies and the like will hope to capitalize on Komputer Kingdom’s impressionable audience. It will be a challenge, as it is with any platform, product or form of entertainment aimed at children, not to let Komputer Kingdom’s site be used by money driven advertisers to manipulate impressionable kids.

8. CONCLUSION

Educational based games have become popular and effective learning tools for children, but there is still an evident dearth of computer science focused educational games. With such a booming field, there needs to be a parallel boom in education to help fulfill the increasing demand of computer scientists. The few introductory learning tools for computer science that do already exist, such as Scratch and Pex4Fun, focus mainly on teaching coding. However, to children, especially those beneath the high school level, the idea of ‘code’ can be perceived as difficult and off-putting. Komputer Kingdom was developed to help address this issue by bridging the gap between no experience with computer science and an introduction to coding. The website provides a fun, interactive experience with mini-games that were designed to teach fundamental computer science skills. Based on the measured success of these games as effective learning tools, Komputer Kingdom acts as a strong basis for future work and growth in the field of education.

9. REFERENCES

APPENDIX
A. PRE-SURVEY SCREENSHOTS

Figure 11: Pre-survey completed before game play.
B. PRE-TEST SCREENSHOTS

Figure 12: First part of Pre-test completed before game play.
You have 5 sacks. Each sack weighs a different amount. You can only compare two sacks at a time. What is the smallest number of comparisons you need to be sure you find the heaviest sack?

- 1
- 4
- 7
- 10

See Question Below

Using the graph above, if I start at 1 and then make choices in the following order, B A A B B, what number do I end up at?

- 1
- 2
- 3
- 4

Figure 13: Second part of Pre-test completed before game play.

C. POST-SURVEY SCREENSHOTS
Figure 14: First part of Post-survey completed after game play.

D. POST-TEST SCREENSHOTS
Rank the mini games that you played on KomputerKingdom.com from most fun and interesting to least fun and interesting. 1 being your favorite, 5 being your least favorite.

<table>
<thead>
<tr>
<th>Mini Game</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match the Dots Game - Binary Number Learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create the Image Game - Pixels forming Images</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guess the Number Game - Binary Search Pattern</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sorting the Bottles Game - Comparative Unknown Sorting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Get to the Treasure Game - Finite State Automata</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 15: Second part of Post-survey completed after game play.
Figure 16: First part of Post-test completed after game play.
You have 5 sacks. Each sack weighs a different amount. You can only compare two sacks at a time. What is the smallest number of comparisons you need to be sure you find the heaviest sack?

- 1
- 4
- 7
- 13

See Question Below

Using the graph above, if I start at 1 and then make choices in the following order, B A A B B, what number do I end up at?

- 1
- 2
- 3
- 4

Figure 17: Second part of Post-test completed after game play.