BINX: An XNA/XBox 360 Educational Game for Electrical and Computer Engineers

Rasha Morsi[1], Chad Richards[2]

Engineering, Norfolk State University, 700 Park Avenue, Norfolk VA, USA

[1]rmorsi@nsu.edu; [2]dahc03@hotmail.com

***Abstract*- This work focuses on the design and implementation of an educational game for deployment on the Xbox 360 commercial game console. Video games can be extremely powerful and effective learning tools when deployed with methodical precision. Currently, no learning tools are available for commercial consoles that focus on teaching core concepts in Engineering. This work presents an overall design for an educational game to address Number Systems in the scope of the Electrical and Computer Engineering curricula.**

BINX is an educational adventure game designed to address Number Systems and their arithmetic operations in Digital Logic Design, a core course is Electrical and Computer Engineering programs. Object Oriented Design Principles were used in the development of this game. This work presents one sublevel implemented using the XNA framework – a Microsoft Software Development Kit (SDK).The BINX gameplay takes place inside a computer where a malicious virus is threatening the computer's operability. The player has to find the virus and eradicate it by disconnecting the terminals of the graphics processor. The player has the opportunity to practice before taking on the Boss Challenge. In the practice phase, the player practices number conversions without fear of penalties. In the Challenge phase, the player is timed and has to perform the number conversions before the allotted time expires. The combination of practice and challenge provides players with the ability to practice number conversions and, when ready, to take on the challenge task where assessment is conducted to provide the player with the knowledge of how well he/she has performed on the challenge. The number conversion level of BINX was implemented and formally evaluated showing improvement in student scores after using BINX.

***Keywords*- Modeling and Simulation; Game Design and Development; Xbox 360; Educational Gaming; Assessment and Evaluations**

I. INTRODUCTION

Educational gaming is quickly emerging as a powerful learning tool in the educational arena. Currently, there exists a phenomenal interest in the science of gaming, and discovering its powerful application to education.

Many students engage in endless hours of video gameplay. In contrast, it is often a daunting task for students to invest hours in studying for a test. Games are engaging, fun, and provide a sense of accomplishment to the student. Many skills have been documented as being gained through playing video games, these include the ability to: process information quickly, utilize parallel processing and visual processing techniques, enhance critical thinking, improve problem solving skills, and improve data organization [1]. These skills are very important for the success of our future Engineers, therefore, it becomes worthwhile to further investigate how to apply game playing techniques to education.

The purpose of this work is to apply gaming theory to the development of a content specific game for Electrical and Computer Engineering (ECE). After a thorough investigation, it was concluded that no learning tools were available for commercial consoles to teach core concepts in Engineering. This work presents the design of a digital logic game, BINX, and uses the principles of gaming to teach core concepts in ECE. BINX is an educational game that provides the user training and assessment on the arithmetic of various number systems through game play. Number Systems are a core concept in the Digital Logic Course is a core course in most engineering curricula. BINX is deployed on the XNA framework for potential use on the Xbox 360.

II. GAMING AND EDUCATION

In recent years, interest in game based education has skyrocketed. It has become a topic of monumental interest to educators around the world. Video games can be a highly valuable supplement to education in science and engineering for many reasons, for example: there is a huge demand for games; they are already played by students; they are structured according to effective learning patterns; they promote learning; and they are more effective than a lecture[2].

A. Popularity of Games

Games have a mass appeal that is hard to question. According to information assembled by the NPD Group, a provider of retail market research information, computer and video game sales skyrocketed to more than seven billion dollars from 2005 to 2008, of which more than six billion dollars was created by the video game software industry [3]. According to the entertainment software association, the video game software industry has more than tripled in sales since 1996 [4]. It is clear that, "interactive immersive entertainment or video games have emerged as a major entertainment medium and enculturation force for today's youth" [5].

On average, a teenager in America "watches over 3 hours of television, is on the internet 10 minutes to an hour, and plays 1 and a half hours of video games" [6]. This equates to a substantial amount of time being spent on video games. At the college level, the average college graduate "will have spent less than 5,000 hours of their lives reading, but more than 10,000 hours playing video games" [7]. It is clear that when given the option, people will choose to spend their time playing games. If they choose to allocate their time to gaming, why stop them? The "true secret" [7] of their fascination in gaming is that people are learning the skills required for their twenty-first century lives.

B. The Need for Educational Gaming

The skill set required to succeed in today's society has evolved. Today's educational system was devised over a century ago in order to prepare students for the industrial revolution [8]. Times have changed, and so have the teachers and students. In *Games and Simulations in Online Learning*, Galarneau& Zibit state "We need students to learn the skills necessary for the 21st century, yet we teach them with yesterday's tools and measure outcomes using yesterday's assessments" [8]. With the steady progression of technology, students now have the capability of interacting with these technologies to assist in learning very differently than their parents [6].

New educational tools are needed to teach and train today's evolved minds. Students' brains have been programmed to the "speed, interactivity, and other factors in the games" [6] they play. Marc Prensky refers to parents and teachers as *Digital Immigrants*, and students as *Natives* [7]. The problem challenging education today is *Digital Immigrants* from a *pre-digital age*, are unable to transfer knowledge, concepts, ideas, and teach the *Natives*, who speak an entirely different language [7]. Accents prevent the *Natives* from understanding what the *Digital Immigrants* are saying and *Natives* are accustomed to processing data much faster than the *Digital Immigrants* can dispense it [7]. The *Digital Immigrants* are accustomed to linear progression; however, *Natives* prefer multitasking. *Digital Immigrants'* first choice of communication is text rather than graphics; however, *Natives* enjoy more graphics than text. The *Digital Immigrants* prefer to view everything in some kind of order for example, Chapter 1, 2, 3 while *Natives* are accustomed to assimilating information in their own, random way [7]. It is clear that today's student cannot efficiently absorb knowledge from old teaching tactics. Prensky cites, "No less an authority than the late Dr. Albert Shanker, head of both the New York City—based United Federation of Teachers (UFT) union and later the nationally based American Federation of Teachers (AFT) union, claimed, as far back as 1988, that *"only 20 to 25 percent of students currently in school can learn effectively from traditional methods of teaching"* " [6]. If students cannot learn, it is the job of the instructor to implement new learning tools to allow learning to occur. If students are already playing games, why not use them as a means to educate them?

When educators choose not to adapt their teaching styles to the new wave of students they may actually be causing harm to the student. Linear thought processes that are currently implemented in the educational system "will retard learning for brains that have been developed through game and websurfing" processes on the computer [6]. It is clear that today's children, teens, and young adults are not receptive to traditional teaching methods [6]. Often students are blamed, however, Colin Powell at the 2000 Republican Convention appropriately stated that, "Our children are not the problem, the problem is us!" [7]. There is a technological revolution going on right now, and mainstream education is doing nothing to adapt to the learning styles of the new generation of students [6]. The old learning system is obsolete and breaking down. Professors, teachers, and faculty must use innovative techniques to convey information to the students as well as maintain their attention. To meet these requirements, the concepts must be imparted through various methods such as questioning, exploration, engagement, interactivity, and most importantly, fun [6]. These techniques need to be implemented, if not, students will not focus in an educational or training environment. It is imperative that fun and games be injected into education and new learning tools must be developed and implemented to meet this new need.

III. XNA

The XNA framework was used for the implementation of BINX. The XNA framework was selected since, for the first time, a tool exists that allows a nonprofessional game developer to create games that can run on both PCs and the Xbox 360 game console [9]. XNA is not an acronym; it represents and encompasses all of the game development technology Microsoft provides [10]. XNA Game Studio is a Software Development Kit (SDK) with a set of "prebuilt program components" [11] that can be used as part of other programs. Using XNA Game Studio and the underlying language C#.NET (which is the official language of XNA), video games can be developed for Windows. XNA allows one to write games for the Xbox 360, however the console is not required to use XNA. Games can be developed strictly for the PC using XNA with the ability to play the game using the Xbox 360 controller. To develop games that play on the Xbox 360, a subscription to the XNA Creators Club is required.

The XNA framework is a set of managed Dynamic Link Libraries (DLLs) which contain a large set of class libraries that help in the creation of games [10]. A Game Studio project includes references to all of the code and resource files that are implemented by the game application. There are two different models for the development of XNA projects: Windows Game projects and Xbox 360 Game projects. Each project calls a different instruction set. The difference between the two projects is the version of the .NET Framework the two platforms are running. Windows operates on the complete .NET Framework whereas the Xbox 360 runs on the .NET Compact Framework. The reason it is called .NET Compact Framework is because the devices it typically runs on are compact, and also because the framework itself is a subset of what is available on the desktop [11]. There are also minor differences between the projects like mouse support; however, in most cases the code from one project type will still compile in the other [12].

The player uses the gamepad to interact with the system. The gamepad then provides input to the XNA framework. XNA reads the data input provided by the player and creates the output. This output is then presented on the screen for the user.

A. Available XNA Educational Games

Since 2010, a large number of educational games have been developed for the Xbox 360 and the PC using the XNA game engine. Several educational games available through XNA assist in the learning of the vocabulary and grammar of various languages [13-17], other games teach children letters and numbers [18-22]. One game, Math Fighter [22], addresses mathematics from basic math to calculus. So, although educational games exist, they are limited to basic languages, math, basic physics, or astronomy. No games exist in the XNA game environment to enhance engineering education. In general, PC games do exist, mainly for the Kindergarten to High/Secondary School levels, to teach mathematics and science. Time Engineer [23] is a 3D PC-based game that teaches basic concepts in math and science as they relate to engineering. It uses themes like Ancient Egypt to teach these core concepts. Time Engineer has a section intended to cover binary numbers, '1's and '0's, as logic associated with switches turning on and off. It does not address number conversions or any of the other number systems which BINX is intended to address.

IV. NUMBER SYSTEMS

A number system is a group of symbols used to express the foundation of counting, comparing amounts, assessing calculations, and representing values. There are numerous number systems to represent any given number. The four number systems most frequently associated with computers and electronics are Binary, Octal, Decimal, and Hexadecimal. A core concept in Digital Logic Design is the study of these four number systems. Students in Digital Logic Design require a solid foundation in the understanding of these number systems in order to matriculate through the curriculum. The purpose of BINX is to provide practice as well as assessment for these four number systems and their associated arithmetic operations.

Binary is the number system of the computer. All input, output, storage, and internal commands are processed inside the computer using 1's and 0's. Binary is also known as a base-2 number system. A bit is one binary digit. When 8 bits are concatenated, they represent one byte. One byte can have up to 256 different combinations of 1's and 0's. Each digit of a binary number has an associated weight, which is represented as an increasing power of 2. The least significant bit (LSB) represents 2^0.The next binary digit represents 2^1, 2^2, 2^3, respectively, until the most significant bit (MSB) is reached (for a Byte, this is 2^7). For example, the decimal number 6 is represented by 0110.

	MSB			LSB
Weights	2^3	2^2	2^1	2^0
Binary	0	1	1	0

Fig. 1 Basic Binary Number Representation

V. GAME DESIGN OVERVIEW

BINX provides training on number systems and their associated arithmetic operations through a series of levels and sublevels. BINX is designed as an Adventure game, and takes place inside a computer. The game is composed of four main levels, which are in turn composed of sublevels to provide practice and assessment of the material (Figure 2). The first level focuses on number conversion. The second level will address 1's and 2's compliment representation. The third level will cover addition and subtraction. Finally, the fourth level will cover multiplication and division.

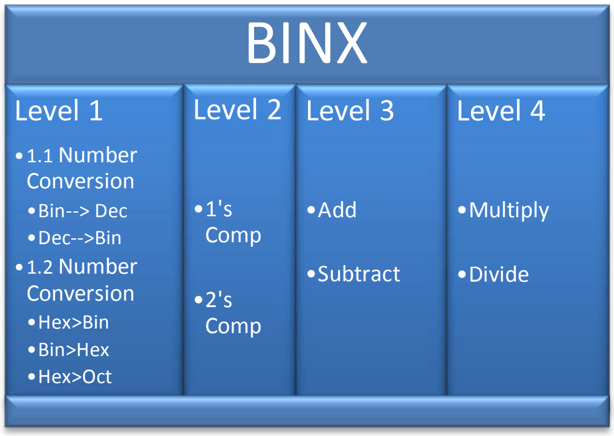


Fig. 2 BINX Level Layout

A. BINX Storyline

The computer is turned on. The CPU lights up, the keyboard activates, and a horrible sound comes from the speakers; however, no video is output to the monitor. A malicious virus is attacking the computer. The main character's name is Chip. Chip is an integrated circuit designed with one purpose, and one purpose only; to wipe all traces of this malicious virus from the computer.

B. BINX Level 1 Design

The learning areas of the first sublevel of level 1 emphasizes decimal to binary, and binary to decimal conversion. This sublevel is aimed at providing the player with practice in number conversion. The objective of the sublevel one Practice task is to collect digits in the correct order to open gates to access the next area of the game. The next sublevel task is composed of a timed Boss Challenge to deliver assessment of the player's mastery of the concept. Each level is composed of various sublevels that have a practice component as well as an assessment portion. This combination is expected to promote learning.

Inside the computer, information is flowing from the motherboard to all the output devices attached to the computer except the monitor. The adventure takes place inside the CPU. The player must navigate a path through the bus to find the viral infection plaguing the graphics processor.

1) Sublevel 1 – Practice:

To matriculate through this sublevel the player must pass through access doors in the maze that lead to the location of the virus. The player must acquire digits in the correct order to gain access through the doors. To reach the challenge sublevel, correct answers must be provided at each door. Each of these doors randomly place digits for the Chip (the character of the game) to collect. The player must then perform the appropriate number conversion and acquire the numbers in the correct order to create the required value to open the door. Once the player assumes the correct digits have been acquired, the player may approach the door and attempt to open it. If the correct answer is provided, the door will open allowing access to the rest of the maze where more doors need to be opened to reach the virus and eradicate it. If the answer is incorrect, the digits will be repositioned and the player will have another attempt to correctly gain access to the next area. Each correct answer was designed to place the player one step closer to the virus. Figure 3 illustrates the design of the first sublevel. Stars indicate where the doors are located in the maze. In this phase of the game, no penalties are associated with incorrect responses. The player is merely allowed to try as many times as is necessary to open each door.

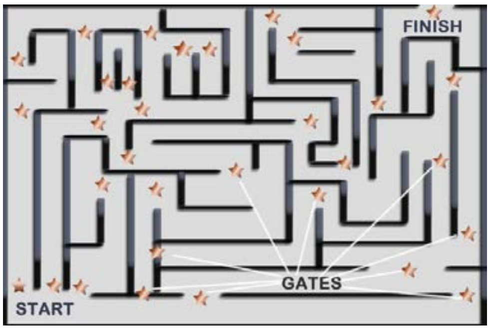


Fig. 3 Sublevel 1 Practice Floor Plan

2) *Sublevel 1 – Boss Challenge:*

Here, Chip has found the viral infection attacking the processor. To stop the spread of the virus, he must locate the infected terminals of the graphics processor chip and sever the connections. This Boss Challenge is designed to provide assessment of the player's ability to conduct the conversions. The Boss Challenge is a timed duel. The player is required to correctly answer 80 percent of the problems, and do so within the given time constraints. If the player was unable to defeat the boss within the time frame or without a passing score, the player would be required to restart the sublevel and do more practice. If the player is able to defeat the challenge, the game will provide an assessment screen to provide the player assessment of their comprehension of the concept. After the assessment screen appears, the player will progress to sublevel 2 only if a passing score was received and the challenge was completed within the time constraints. Fig. 4 illustrates the initial floor plan of the Boss Challenge.

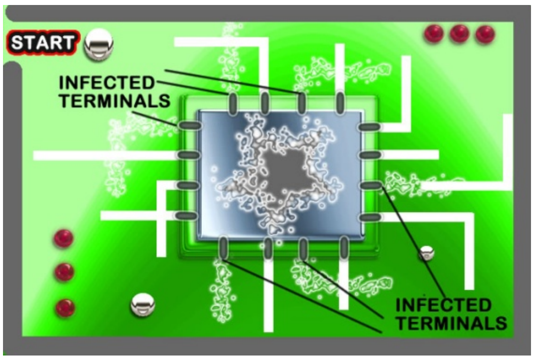


Fig. 4 Sublevel 1 Floor Plan

3) *Camera View:*

The camera position (shown in Fig. 5) was designed to be from the third person perspective. This allows the user to view the main character Chip, and his relative position to the digits which he must acquire.

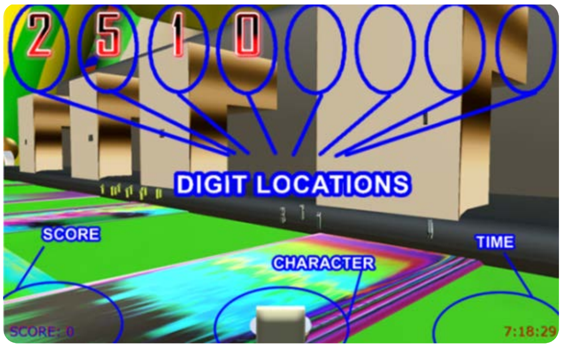


Fig. 5 BINX Level 1 Challenge Screenshot

VI. BINX IMPLEMENTATION

This section discusses the implementation of the first sublevel's practice and challenge tasks. The implementation of these two tasks shows the overall applicability of the game design on the XNA Framework.

A. BINX

The *Binx* class is the main game class. *Initialize()*, *LoadContent()*, *UnloadContent()*, *Update()*, and *Draw()* are five predefined methods of the XNA Framework (Figure 6). The *Initialize()* method is where the game initializes values. The *LoadContent()* method is where graphics, models, and sound are loaded into to the game. The *Unload()* method is used when specific game objects require to be unloaded or disposed of. The *Update()* method updates the games logic and reasoning. For example, the *Update()* method determines if two objects have intersected, it checks if the position of an object has changed, or determines if a specified time allotted has expired. The *Draw()* method renders two and three dimensional objects to the screen. The *Draw()* method is used to draw sprites, environments, and any other visual objects needed in the game world to the screen.

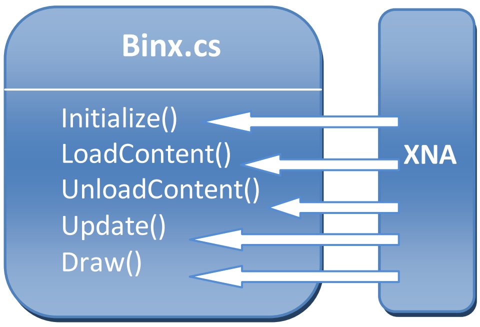


Fig. 6 XNA Predefined Methods

The *Initialize(), LoadContent(), and UnloadContent()* methods are called automatically at the beginning of the game after the *Binx* class constructor is called. Next, the game creates a continual loop between the *Update()* method, and the *Draw()* method (Fig. 7). The game attempts to continually call these methods at a rate of 60 times per second. This switching between the methods is what creates the continual game loop.

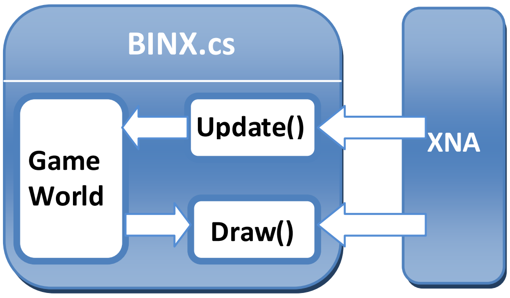


Fig. 7 XNA Game Loop

B. *Game Components*

The implementation of the first sublevel of BINX was designed to be modular so the code could be reused in later additions and versions of the game. To allow reusability and scalability of the code, game components were implemented. The *Game Component* class, and the *Drawable Game Component* class are two classes defined by the XNA framework.

1) *GameComponent:*

Game components are separate classes from the main game class that can be easily called from within the *Binx* class to provide their defined functionality. To do this, each class needs to have its own copy of any of the five predefined methods of the XNA framework, and it also needs to have the methods called to operate. To create a copy of the needed methods and have them called; a class derives from the *GameComponent* class. By deriving from the *GameComponent* class of the XNA framework, the game component can be added to the Components list of the *Binx* class. When a game component is added to the list of Components of the *Binx* class, it's *Initialize()* method is called after the *Initialize()* method of the *Binx* class. Also, the *Update()* method of the game component is called after the *Update()* method of the *Binx* class has finished.

2) *Drawable Game Component:*

The *Game Component* class does not inherently implement the *Draw()* method. If a component needs to render an object to the screen it must inherit from the *Drawable Game Component* class. The *Drawable Game Component* class calls the *Draw()* method of the class after the *Draw()* method of the *Binx* class has been called.

C. *Game Services:*

In BINX, several game components need to be able to access the audio component and the camera component. To provide global access to these objects without a direct reference to them, Game Services were implemented. Game Services (Fig. 8) allow objects to work together without holding on to a direct reference to the object they need. Game Services function by adding their service to the Services collection member of the *Binx* class. Game Services allow objects the ability to search the Service collection member of the *Binx* class and use the members or events of objects that have added their Service to the collection.

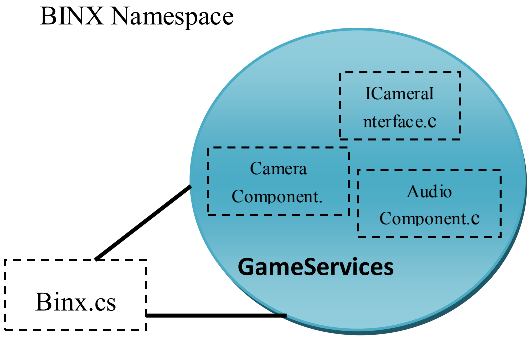


Fig. 8 Game Services Collection

1) *Audio Component:*

The audio component was created to add sound effects to the game. The audio component adds its services to the *Binx* class when it is initialized. Sound is a type of game asset added through the XNA framework. XNA comes with a free tool from Microsoft called XACT, Cross-Platform Audio Creation Tool. The audio creation tool was used to build a sound project. Various sound clips were created in .wav files and then imported into the XACT audio creation tool. XACT was used to create sound banks, wave banks, and cues that are then compiled into a XAP file. The XAP file was then added to the content folder. The audio component accesses these files through the content manager. The *Audio Component* class can then play the audio when a sound cue is called in a specified class using the *Play Cue()* method.

2) *Camera Component:*

The *Camera Component* class does not render anything. This class does not even implement anything in its *Draw()* method. The camera component informs other components of how they should draw themselves based on the camera's position. This information is calculated based upon input from the player. When the player moves the left and right thumb sticks the camera component updates three properties that other game components use to draw themselves. These properties are View, Projection, and World. By updating these properties based on user input, it creates the effect that the player is in the game world. Code Listing 1 shows the View, Projection, and World matrices.

```
world = Matrix.CreateRotationY(avatarYaw) *
Matrix.CreateTranslation(avatarPosition);
view = Matrix.CreateLookAt(cameraPosition,
        avatarPosition + binxOffset, upVector);
projection =
Matrix.CreatePerspectiveFieldOfView(viewAngle,
        viewPort.AspectRatio, nearClip, farClip);
```

Code Listing 1 Camera Matrices

The camera component can be conceptualized as representation of the player in the game world. Imagine the camera component as an object that does nothing but describe the player. The camera defines: what score the player has, what digits the player has acquired, how much time the player has left, how much time the player has used, how many attempts the player has made, how many binary to decimal conversion attempts were correct, and how many decimal to binary conversion attempts were correct. These values are also presented to other components through properties. The properties are called *Score*, *Digits*, *Time*, *Time Used*, *Attempts*, *DtoB Correct*, and *BtoD Correct* respectively. Unlike the *View*, *Projection*, and *World* properties, the camera does not calculate these properties it merely allows other components to get and set them.

An interface is a contract of a class that specifies the basic functionality that the class must have. The Camera component implements the *I Camera Interface*, to ensure the basic functionality of all the methods and properties of a camera component. When the camera component is constructed it adds its services to the game.

D. Game States:

There are nine (9) states that the game transitions through (Figures9 and 10). An Opening Scene State, Title Screen State, Initial Help State, Practice Level Selection State, Challenge Level Selection State, Practice State, Practice Help State, Challenge State, and an Assessment State.

The *Game State* class (Figure 10) is derived from the *Drawable Game Component* class provided by the XNA framework. The *Game State* class creates a list of game components that the state uses called *component list*. The *component list* stores the game components of the state until they are needed. When the *Game State* class calls the *Draw()* method, it calls the *Draw()* method on each item of the *component list*. When the game state calls the *Update()* method, it calls the *Update()* method on each of the components of the *component list*. The Game State class creates two virtual methods called Show and Hide. When the *Hide()* method is called none of the game components in the *component list* have their *Draw()* method or *Update()* method called. This causes the components to become dormant. When the Show method is called, the class sets the *Visible* and *Enable* properties to true which allows the game state to update and draw each of the game components of the *component list* (Code Listing 2). Each one of the nine (9) states in the game inherits from the *Game State* class.

```
for (int i = 0; i < componentlist.Count; i++)
        {
GameComponent component = componentlist[i];
if ((component isDrawableGameComponent) &&

((DrawableGameComponent)component).Visible)
            {

((DrawableGameComponent)component).Draw(gameTi
me);
          }
```

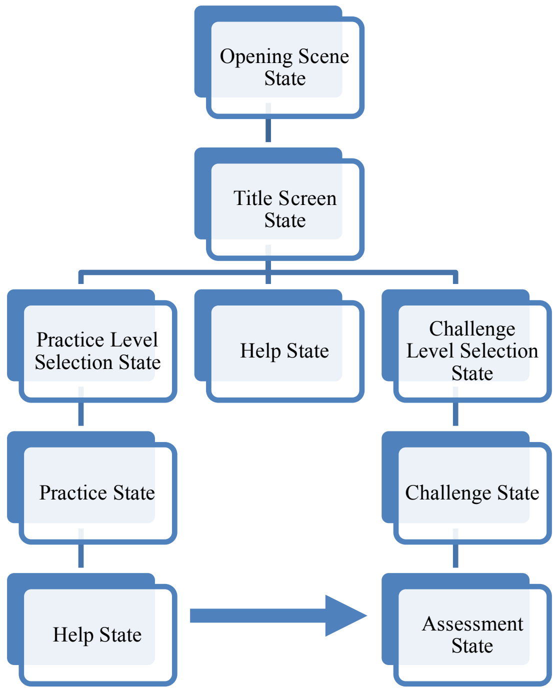
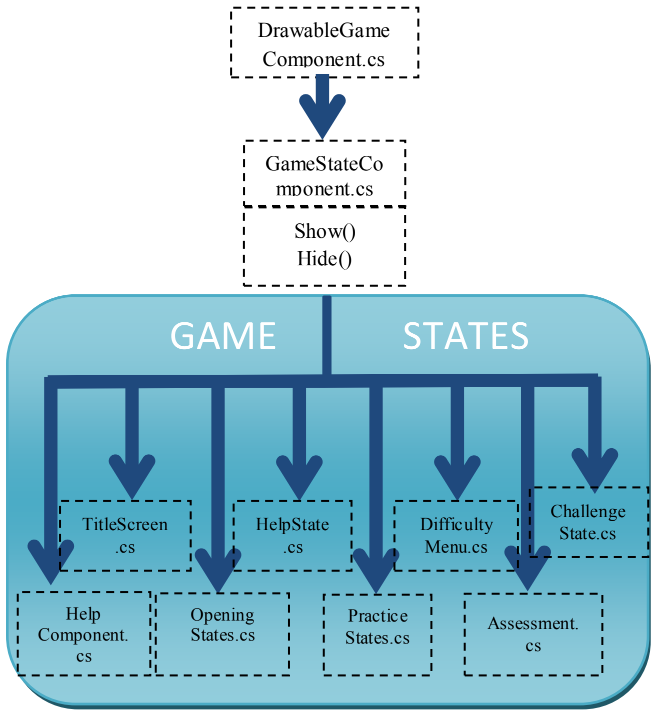


Fig. 9 Game State Flow



Code Listing 2 Game State Component List Draw

E. Assessment State

The *Assessment* class is a game state that retrieves five values from the camera on how the player performed in the challenge state. This component then draws these values to the screen (Fig. 11) in a meaningful way to provide the player feedback on their ability to perform the required conversions in the time allotted. It gets the camera's Score, Attempts, *DtoB Correct*, and *BtoD Correct* properties, and draws these values over a texture2D that is drawn full screen

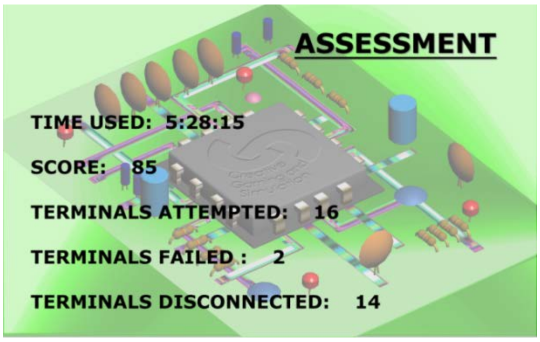


Fig. 11 Assessment State Example

The player's score is calculated when correct answers are provided at the terminal components in the challenge state. When a correct answer is provided at a terminal, the random value displayed on the terminal component is added to the Score property of the camera. This creates the players score in the assessment state.

VII. BINX EVALUATION

The purpose of this evaluation was to assess and evaluate students' use of the BINX game that was designed to allow students to practice converting binary-to-decimal and vice versa as a precursor for designing digital circuits in the Digital Logic Design course at Norfolk State University. To provide evidence of the game's effectiveness, the students committed to (a) completing pre-assessments regarding learning styles, self-efficacy with learning digital circuit design, and frequency of use with technology; (b) using the game for approximately one week; (c) keeping a journal log of their work; and (d) participating in a focus group to share their experience with the BINX game. To guide the assessment activities, the following statements were used:

1. Students will improve in self-monitoring and self-assessment of effective learning strategies.
2. Students will develop a proficiency in converting binary to digital and converting digital to binary.

During the spring 2010 semester, ten (10) students enrolled in the EEN231 (electrical engineering) course at Norfolk State University. However, only nine (9) of the students completed most of the pre-assessments and post-assessments to measure their perceptions of the effectiveness of the BINX practice game for converting binary-to-decimal and vice-versa during one section of the class.

Pre-assessments included measuring learning styles, self-efficacy with learning and demonstrating learning with digital logic design, frequency of technology use, and demographical data. In addition, students were given four (4) homework problems on binary-decimal-binary conversions. The post-assessments included a quiz that included problems on conversion as one section, journal entries, and a focus group.

A. Findings

As a result of the pre and post assessments, findings from the survey, journal entries, and focus group revealed some preliminary information regarding the usefulness, applicability, utility, and practicality of the BINX game.

1) Profile:

A profile of the ten (10) student participants is shown in table I below. The students used the BINX game for a minimum of 1.5 hours each.

TABLE I PROFILE OF STUDENT PARTICIPANTS

#	Ethnicity	Gender	Designation				Major	
			S	J	So	Un	CE	EE
10	AA	Male	1	3	5	1	1	9

Where: **AA**- African American; **S**-Senior; **J**-Junior; **So**-Sophomore; **Un**-Unknown; **CE**-Computer Eng.; and **EE**-Electronics Eng.

With regard to the learning style profiles shown in table II, this reveals that the majority of the students were active, visual, sensing, and sequential learners.

TABLE II PROFILE OF STUDENT LEARNING STYLES

Dimension 1		Dimension 2		Dimension 3		Dimension 4	
Act	Ref	Vis	Ver	Sen	Int	Seq	Glo
75%	25%	100%	0%	75%	25%	68%	32%

Where: **Act**- Active; **Ref**-Reflective; **Vis**-Visual; **Ver**-Verbal; **Sen**-Sensing; **Int**-Intuitive.; **Seq**-Sequential; and **Glo**-Global.

2) Self-Efficacy in Academic Achievement Baseline:

Self-Efficacy is a concept that measures an individual's perception of confidence in their ability to do new things in combination with the ability to cope with the unknown. Taken together with having confidence in academic skills and abilities, the self-efficacy survey provided a baseline of information regarding the students' perceptions of how well they believe they can learn, retain, and engage in conversations about academic work that is new and difficult. As a result, the self-efficacy in academic achievement section included questions that asked the students about their perceptions of how well they can learn new concepts, remember information, identify places to study without distractions, motivate themselves to study difficult work, and participate in class discussions when they are unsure about the topic. Two of the questions on learning were in the context of digital circuit design.

The scale presented on the survey ranged from 1 to 7 representing *'not well at all with help'* (referred to as low efficacy) to *'very well on my own'* (referred to as high efficacy), respectively. Overall, the students indicated high levels of self-efficacy—with the majority scoring at level 5, 6, or 7. Specifically, with regard to the survey items on learning, 78% of the students indicated they could learn new concepts with technology *'pretty well with a little help'*. Specific to learning digital concepts using technology-based applications, 44% could learn the rules *'pretty well with a little help'*, and 33% could learn them '*pretty well with lots of help'*. In addition, when students were asked how well they believed they could design digital circuits, only 1/3 indicated they could do this *'pretty well with a lot of help'*- the remaining students rated their ability to do this at lower levels of efficacy. None of the students indicated they could learn digital circuits on their own. What this suggests is that students appear to have higher efficacy regarding learning new concepts with technology or even learning concepts using technology based applications. However, when asked specifically about digital circuits or designing digital circuits, the students clearly indicated they would need additional help.

With regard to remembering new information, students responded to survey items that asked how well they could remember new information presented in class, textbooks, or via hands-on practice. Using the same efficacy scale, the student responses were equally distributed across the efficacy scale indicating that, as a class, there was no majority that perceived self-efficacy with regard to this statement. This equal distribution across the efficacy scale was also true for learning new materials presented in the textbook. When asked about learning new materials via hands-on experience, more students indicated higher efficacy with 56% indicating they could do this *'pretty well on their own'*. For the most part, students have a lower self-efficacy with learning or designing digital circuits, but have a higher self-efficacy with regard to learning new concepts with technology-based applications and learning new information when given the opportunity for hands-on experience.

In summary, when asked about motivation to do difficult work, the majority of the students indicated higher efficacy with 56% indicating they could motivate themselves very well on their own. Interestingly, only 44% of the students indicated that they could participate in class discussions when they are unsure about the topic *'pretty well, with a little help'*. The remaining students indicated that they could do this *'okay alone'*, and 22% indicated they could *'pretty well alone'*.

3) Baseline: Frequency of Technology Use:

The frequency of use of technology section listed seven activities for which the desktop and/or laptop are used and asked the students to indicate the level of frequency they participate in these activities. The activities included email, instant messaging, MySpace/ Facebook, shopping, researching databases for class work, sending and receiving photos, and uploading or viewing YouTube. The scale used to measure the levels of frequency included (a) Frequently (daily) = 5; Often (2 to 3 times/week) = 4; Sometimes (once a week) = 3; Seldom (2 to 3 times/month) = 2; and Never = 1.

As indicated in the research regarding today's youth and their use of desktops and laptops, all ten of the students indicated that they email every day and eight students use their cell phones for reasons other than to make calls daily. Five of the students use instant messaging and Facebook/MySpace daily, and the majority of the students seldom or never shop online.

In summary, baseline data indicate that the majority of the students were active, visual, sensing, sequential learners who use computers and various other technologies on a daily basis for various purposes including learning. Their efficacy is higher for learning with technology, various software applications, and even new concepts, but not as high when it comes to designing digital circuits or participating in discussions when unsure about the topic. As well, the students' efficacy with remembering information when allowed to engage in hands-on practice was rated high as well as their efficacy with motivating themselves to do work that is difficult to understand. These baseline indicators suggest that the BINX game—designed to help students better understand converting binary-to-decimal and decimal-to-binary, should prove beneficial to a group of students who are visual learners, remember better with hands on practice, learn fairly well with technology, and can continue playing the game even when it gets difficult.

4) Journal Entries: Using the BINX Game:

Given that the BINX game was designed to help students with conversions—which is only one of the concepts critical to successful completion of the course, nine (9) of the ten (10) students provided a recorded time of a 1.5 hour session that was set aside to play the BINX game. During this time, they were asked to keep a journal log of their online experience which included the date; start and end times; whether they were converting binary-to-decimal or vice-versa; the level of difficulty (easy, moderate, or difficult); whether or not they played the Maze or the Assessment; and whether or not they derived at the correct answer. The findings from the logs revealed only the length of time the students played the BINX game but not the level of difficulty, whether or not they played the Maze or the Assessment, or indicated whether they derived at the correct answer.

5) Focus Group After Using the Practice Game:

One week after the students played the BINX game and completed their quiz problems that specifically addressed converting binary-to-decimal and vice versa, four of the ten students participated in a focus group to discuss their experience. The students responded to eight (8) questions that asked about the helpfulness of the game, difficulty of the game (or user-friendlessness), learning opportunities using the games, and recommendations for improving the game.

The results from the focus group revealed that overall the students indicated that the game was very helpful with practicing converting binary-to-decimal and decimal-to-binary digits. The students also indicated that the level of difficulty (on a scale from 1 – 5 with 1 being easy and 5 being difficult) was between 3 and 4. The students indicated that regardless of what challenges they faced with the BINX game, they never stopped practicing—they just figured out how to do what they needed to do. When asked about why they never stopped playing even when the game was difficult, they indicated that they are not used to losing and you want to win. Students learned to work around the "kinks" they believed were in the game and continued to practice both the conversions as well as demonstrating to themselves their gaming skills. Students indicated that after a while "you stop thinking of it as practicing and start seeing it as a game".

Students were also asked whether or not they spent as much time as they should have on the BINX game as a practice tool and they said, "no"—mainly because you can never really get enough practice for anything. Interestingly, students began to create other challenges for themselves such as trying to either beat the allotted time provided in the game or beat their own time when playing the game. In addition, when students were asked if the game helped them with their conversions, the response was "yes, most definitely it helps!". Using a game was something new with studying concepts needed for class, and "you need practice".

In summary, the students enjoyed playing the BINX game, believed it was helpful to practicing conversions of binary-to-decimal and decimal-to-binary, would recommend the game to others with some improvements, and enjoyed playing the game since it eventually did not "feel" like practicing concepts. As well, students would recommend that the scenery change as you move to different levels, allow the levels to change gradually, provide a "clear" button just in case you choose something you do not want, and add more features that are reflective of the other concepts covered in the course.

6) Academic Achievement:

In addition to the above assessments, all ten (10) of the students completed two assessments to measure their understanding and ability to convert binary-to-decimal and decimal-to-binary before and after using the BINX game. The pre-assessment included three (3) homework problems converting binary-to-decimal and one homework problem converting decimal-to-binary. The post-assessment included a quiz that required the students to convert decimal-to-binary for three of the problems and convert binary-to-decimal on two of the problems. These pre- and post-assessments were combined to reveal an average increase or decrease in the students' movement toward correct problem solving. Specifically, of the 10 students 60% increased in their accuracy in converting binary-to-decimal as well as converting decimal-to-binary. This means that 40% of the students realized a decrease in demonstrating an ability to convert from binary-to-decimal and decimal-to-binary. This might be explained by the environment whereas homework problems allow students to work at their own pace in a chosen environment and the quiz is a controlled, timed environment not selected by the student. Either way, the percentage of students who were able to perform the conversions increased from the pre to post assessments.

VIII. CONCLUSION

The aim of this research was to design an educational game (BINX) that can be deployed on an off-the-shelf console as well as a PC. BINX, an educational adventure game, was designed to address Number Systems and their arithmetic operations in Digital Logic Design; a core course is Electrical and Computer Engineering programs. The effectiveness of BINX in the classroom to practice number conversions was also formally evaluated. Although mastering conversions is not the end all for successful achievement in a Digital Logic Design course, they are pivotal and critical. As a result, it was important to measure students' academic achievement before and after playing the BINX game while understanding the context of the students' learning styles and level of comfort with learning via technology and games, as well as frequency of technology use. The results indicated that more students (60%) increased in the number of correct responses they received with class assignments specific to converting binary-to-decimal after using the game.

In addition to achievement in better demonstrating proficiency in conversion of binary and decimals, students also indicated that they enjoyed playing the game to the point that the consciousness of practicing the conversion became secondary to gaming. As a result, the students continued to work at figuring out ways to play even when playing became difficult, created new challenges for themselves such as beating the time allotted in the game or their own time. In addition, students tended to comment on the visual imagery of the game—in that they wanted to see more scenery variations as they moved to different levels of the game. Developing students' proficiency in conversion became "fun" without compromising their learning.

In summary, the overall intent of the BINX game was to enhance students' ability to monitor their learning as well as develop their proficiency in converting binary-to-decimal and decimal-to-binary as one aspect of successful completion of the Digital Logic Design course. The BINX game proved to be helpful with both of these expectations because students were able to determine their learning via the game as well as increase in the percentage of accurate responses to the conversion process. Overall, the BINX game improved the learning of binary-decimal conversion for at least 10 African-American male engineers of the future.

ACKNOWLEDGMENT

The evaluation work performed on BINX was made possible through partial support from the National Science Foundation (NSF) grant # 0737242.Special thanks go to Dr. Gwen Lee Thomas, CEO of Quality Measures LLC who was the external evaluator for the project and conducted all the assessments for the BINX evaluation.

REFERENCES

[1] R. Morsi and E. Jackson, "Playing and Learning? Educational Gaming for Engineering Education", *Frontiers in Education Conference*, 2007.

[2] M. Mayo, "Games For Science and Engineering Education". *Communications of the ACM*, vol. 50, No. 7.

[3] A. Lobao, B. Evangelista, and J. Farias, *Beginning XNA 2.0 Game Programming From Novice to Professional,* New York: Apress, 2008.

[4] (2009) The NPD Group, Inc., [Online] "2008 Video Game Software Sales Across Top Global Markets Experience

[5] Double-Digit Growth". Available: http://www.npd.com/press/releases/press_090202.html.

[6] (2008) The Entertainment Software Association, [Online] "Video Games & Education". http://www.theesa.com/facts/index.asp.

[7] R. Halverson, et al., "Theorizing Games in/and Education". Madison Wisconsin: International Society of the Learning Sciences, June 2006.

[8] M. Prensky, *"Don't Bother Me Mom-- I'm Learning!"*. St. Paul, MN: Paragon House, 2006.

[9] M., Prensky, *Digital Game-Based Learning*. St. Paul, MN: Paragon House, 2001.

[10] D. Gibson, C. Aldrich, and M. Prensky, *Games and Simulations in Online Learning.* Hershey, PA: Idea Group Inc., 2007.

[11] J. Gee, *Good Video Games + Good Learning*. New York, NY: Peter Lang Publishing, 2007.

[12] E. Ferguson, B. Rockhold, and B. Heck, "Video Game Development Using XNA Game Studio and C#.NET", *Journal of Computing Sciences in Colleges*, vol. 23, 2008.

[13] J. Perry and A. Sherrod, *Essential XNA*. Plano, TX: Wordware Publishing, Inc., 2008.

[14] (2012) Phantomgames, *Learn Basic Italian*. *XBox Live.* [Online] Available: http://marketplace.xbox.com/en-US/Product/Learn-Basic-Italian/66acd000-77fe-1000-9115-d80258550b00?cid=search.

[15] (2010) GZStorm Games, *Let's Learn Japanese Beginner*. *XBox Live.* [Online] Available: http://marketplace.xbox.com/en-US/Product/Lets-Learn-Japanese-Beginner/66acd000-77fe-1000-9115-d802585504ac?cid=search.

[16] (2010) J. Glass., *German 101*. *XBox Live.*[Online]Available: http://marketplace.xbox.com/en-US/Product/German-101/66acd000-77fe-1000-9115-d80258550588?cid=search.

[17] (2010) UtopioneerGames, *Spanish 101.XBox Live.*[Online] Available: http://marketplace.xbox.com/en-US/Product/Spanish-101/66acd000-77fe-1000-9115-d80258550752?cid=search.

[18] (2008) Jagglestein.,*Arriba!XBox Live.*[Online] Available: http://marketplace.xbox.com/en-US/Product/Arriba/66acd000-77fe-1000-9115-d80258550126?cid=search.

[19] (2012) Water Bear Interactive LLC, *Alexander Learning Series*. *XBox Live.*[Online] Available: http://marketplace.xbox.com/en-US/Product/Alexander-Learning-Series-360/66acd000-77fe-1000-9115-d80258550a8a?cid=search.

[20] (2010) elettro Deamon., *ABC Fun*. *XBox Live.*[Online] Available: http://marketplace.xbox.com/en-US/Product/ABC-Fun/66acd000-77fe-1000-9115-d80258550575?cid=search.

[21] (2011) Gould Computing., *Avatar Word Wave*. *XBox Live.*[Online] Available: http://marketplace.xbox.com/en-US/Product/Avatar-Word-Wave/66acd000-77fe-1000-9115-d80258550891?cid=search.

[22] (2009) hotshot 10101, *Math Sniper 3D*. *XBox Live.*[Online] Available: http://marketplace.xbox.com/en-US/Product/Math-Sniper-3D/66acd000-77fe-1000-9115-d80258550204?cid=search.

[23] (2011) iBlade., *Math Fighter*. *XBox Live.*[Online] iBlade, Available: http://marketplace.xbox.com/en-US/Product/Math-Fighter/66acd000-77fe-1000-9115-d80258550982?cid=search.

[24] (2009) Software Kids, Time Engineers Video Game. [Online] Available: http://www.software-kids.com/html/time_engineers.html.

[25] C. Carter, *Microsoft XNA Unleashed: Graphics and Game Programming for Xbox 360 and Windows*. Indianapolis, Indiana: Sams Publishing, 2008.

[26] S. Cawood and P. McGee, *Microsoft XNA Game Studio Creator's Guide*. New York, NY: McGraw-Hill, 2007.

[27] R. Miles, *Microsoft XNA Game Studio 2.0: Learn Programming Now!* Redmond, WA: Microsoft Press, 2008.

[28] D. Irish, *The Game Producer's Handbook.* Boston, MA: Thomson Course Technology, 2005.

[29] T. Fullerton, *Game DESIGN Workshop*. Burlington, MA: Elsevier, 2008.

[30] (2009) Microsoft, game details. XNA Creators Club Online.Available: http://creators.xna.com/en-US/.

[31] J. Hall, *XNA Game Studio Express: Developing Games for Windows and The Xbox 360.* Boston, MA: Thomson Course Technology PTR, 2008.

[32] C. Aldrich, *Learning by Doing*. San Francisco, CA: Pfeiffer, 2004.

[33] P. Gestwicki and F. Sun, *Teaching Design Patterns Through Computer Game Development.* New York, NY: Association for Computing Macinery Journal on Educational Resources in Computing, vol. 8 No. 1 Article 2, March 2008.

[34] T. Marsh, K. Yang, and C. Shahabi, *Game development for experience through staying there.* Boston, MA: Association for Computing Machinery Inc., July 2006.

[35] L. Pardew and D. Whittington, *Beginning Game Art In3DS Max 8*. Boston, MA: Thomson Course Technology PTR, 2006.

[36] Y. Rankin et al., *User Centered Game Design: Evaluating Massive Multiplayer Online Role Playing Games for Second Language Acquisition*. Los Angeles, CA: ACM, August 2008.

Morsi, R Ph.D, was born in Egypt and raised in the United Kingdom. She holds a first class honors degree in Computer Systems and Electronics Engineering from King's College, University of London and a Masters and PhD in Electrical and Computer Engineering from Old Dominion University in Norfolk VA. She is currently an Associate Professor, Graduate Program Coordinator and founding director of Creative Gaming and Simulation Lab (CGS) in the Engineering Department at Norfolk State University, Norfolk Virginia, USA.

Richards C. graduated from Norfolk State University's Engineering program with a masters in Electronics Engineering. He is currently working for the US Army Corp of Engineers as a deployed civilian and project manager.