ABSTRACT

Using an Interactive Narrative App on Object Oriented Programming to Teach and Interest Students in Coding

Micah Dadson

Director: Dr. Matthew Fendt, PhD.

Central to learning is attention, and students are more likely to engage with material if interested in it. Learning games are a popular method to encourage engagement. We designed a narrative learning game that targets middle school girls to teach and interest them in OOP. Because many students report difficulty with first time exposure to OOP, we thought it important to make learning more accessible. The narrative framework of our game serves not only to entertain, but to provide a rich medium via which our subjects can learn. Due to scheduling difficulties, we altered our demographic to the general student population. Nonetheless, we found that most subjects were entertained by Narrate, that many had an increased interest in computer science, and that Narrate was effective in teaching various OOP concepts. This validates the use of narrative games to teach OOP.

APPROVED BY DIRECTOR OF HONORS THESIS:

_____

Dr. Matthew Fendt, Department of Computer Science

APPROVED BY THE HONORS PROGRAM:

_____

Dr. Andrew Wisely, Interim Director

DATE: _____

USING AN INTERACTIVE APP ON OBJECT ORIENTED PROGRAMMING TO TEACH AND INTEREST STUDENTS IN CODING

A Thesis Submitted to the Faculty of

Baylor University

In Partial Fulfillment of the Requirements for the

Honors Program

By

Micah Dadson

Waco, Texas

May 2021

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to thank Dr. Matthew Fendt for serving as my thesis mentor. I am very appreciative of the guidance he gave me and the time he personally invested in my education. I consider it an honor to have been able to work with him for the past 1.5 years.

I would also like to thank Dr. Michael Scullin, as the knowledge I learned in his Cognition course motivated and substantiated my decision to create a learning game structured within a narrative environment.

In addition, I would like to thank Dr. Michael Poor not only for serving on my committee, but because being a student in his courses truly encouraged me to pursue computer science as a career path.

I would also like to thank Dr. Brad Keele for serving on my committee, and also because the neuroscience classes I took as his student were some of my favorite classes that I had the privilege of taking at Baylor University.

Further, I would like to thank Professor Carol A. Macaulay-Jameson from the Anthropology Department for giving her students ten extra credit points on a test for participating in my study. Her generosity and kindness allowed me to collect data for my thesis.

Lastly, I would like to thank my sister Moriah, my brother Matthew, and my friends for taking the time to complete my game and providing me with detailed feedback before execution of the study.

DEDICATION

I would like to dedicate my thesis to my uncle, Jerry.

CHAPTER ONE

Introduction

Historically, more males than females have pursued computer science. In 2015, only 18% of computer science graduates were female (Digest of Education Statistics, 2016). Females may avoid pursuing STEM fields due to STEM stereotypes, or they may prematurely leave the field due to a lack of female representation (Dasgupta and Stout, 2014). However, according to the U.S. Bureau of Labor Statistics, employment of computer and information research scientists is expected to grow much faster than other fields by 2028. Hence, females should be encouraged to partake in this growing and lucrative field. Perhaps by exposing females to computer science early in their educational career, these stereotypes can be eliminated and females will be more likely to pursue computer science.

The goal of our learning game, Narrate, is to teach and interest female students in foundational programming concepts and object oriented programming (OOP) concepts through interactive narrative gameplay. Within this demographic, we particularly aim to target middle school girls. We believe that earlier exposure to computer science may pique greater interest in female students compared to those who are exposed later, like during college in which a student may already have a decided career path. Hence, Narrate was built with this special demographic in mind. However, due to difficulties coordinating with middle school teachers this semester, largely because of the ubiquitous

move to remote learning caused by the COVID-19 pandemic, for this study we have broadened our target demographic to focus on the general student population.

Furthermore, the instruction of Narrate focuses on OOP concepts because OOP languages are widely used in industry, and because most university introductory programming courses teach a language that supports OOP. When learning to code, an individual must learn syntax and programming basics like conditionals, looping and functions. In addition, OOP adds the requirement of understanding abstraction, encapsulation, inheritance, and polymorphism. Not to mention, some languages have their own set of unique requirements, like memory management in C++. To put it simply, learning to program is hard. Narrate does not hope to teach foundational programming concepts and OOP concepts to perfection, but rather to serve as an appropriate primer to future traditional instruction.

In addition, the narrative framework of Narrate is just as essential to its objective as its learning instruction. Central to learning is attention, and students are more likely to engage with material when interested in it. A good narrative aims to capture the attention of its audience throughout the entirety of its duration, facilitating attention and thereby facilitating learning. Moreover, It has been shown that difficult material can be understood more easily and retained in memory for greater time if learned in a rich context and processed deeply (Reisberg, 2016). Narrate hopes that by combining narrative and an engaging interface within the context of OOP, it will mitigate some of the difficulty associated with programming while simultaneously providing entertainment value. We hope to spark long lasting interest in female students and to encourage future learning of OOP.

Our sample population consists of female and male students enrolled in an introduction to anthropology course at Baylor University. We consider this experiment to be the first part of our study, and we are proactively recruiting female middle school students to execute this study again in the near future. Nonetheless, this study revealed interesting results about the use of interactive narrative games to teach OOP.

CHAPTER TWO

Narratives as Agents of Learning

It seems as if humans are rather intrinsically receptive to storytelling. In fact, narratives are such an important aspect to human cognitions that a field within psychology has developed for its explicit study in relation to cognitions, coined cognitive narratology (León, 2016). This chapter briefly examines narratives as agents of learning.

The Modal Memory model is a structural memory model that is widely used to explain how humans process information in cognitive psychology (Reisberg, 2016). It proposes three grand constituents of memory: sensory memory, short term memory, and long term memory. According to this model, we are bombarded with sensory stimuli, but because we do not attend to most of them, sensory memory for these stimuli rapidly decays and we fail to perceive them (Figure 1). To educators, this is alarming; a disinterested student that aimlessly sits in a classroom is barely conscious of the material she should be learning. Thus, step one of learning is to get the student excited about the material, and the narrative framework of Narrate does exactly this by tailoring a storyline specifically to its middle school audience and demanding their active participation through interactive gameplay (Figure 2).

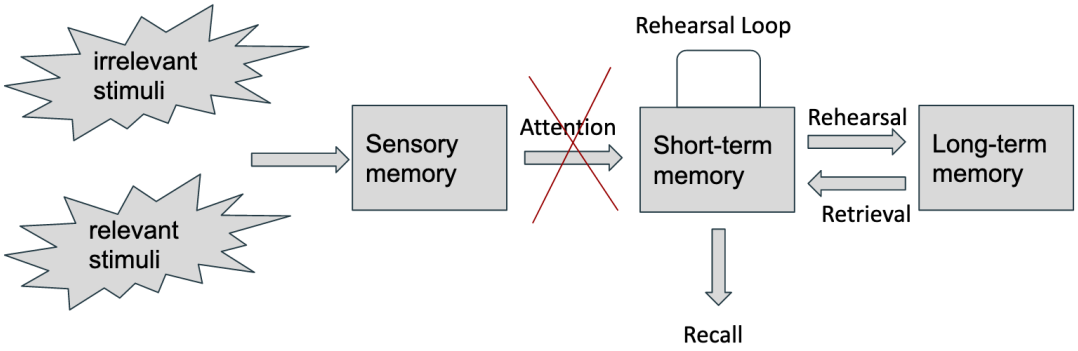

Figure 1: The Modal Memory Model is widely used to explain how humans process information in cognitive psychology (Reisberg, 2016). The constant influx of stimuli makes us largely inattentive to non salient stimuli. Even important information, which quite frankly is all learning instruction, might be ignored if not salient enough.

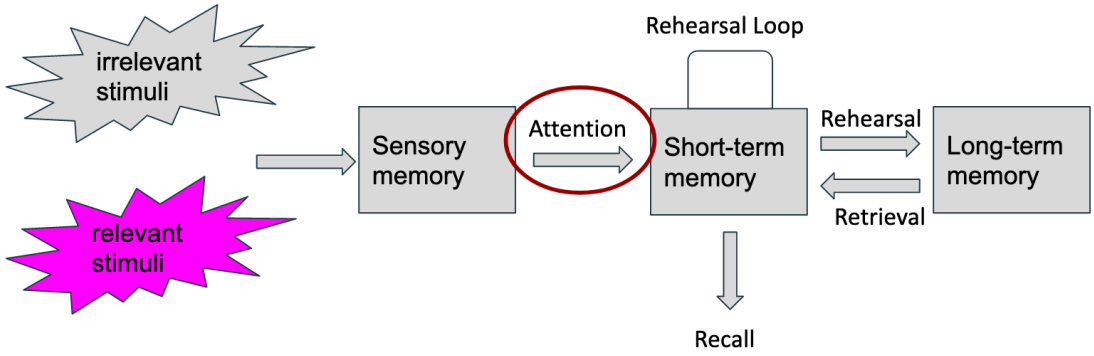

Figure 2: The Modal Memory Model is widely used to explain how humans process information in cognitive psychology (Reisberg, 2016). The narrative framework of Narrate serves in part to make its learning instruction more salient by embedding it within a compelling storyline and within an engaging interface. This allows us to optimize attention, making the user more receptive to learning.

Now that we have engagement, we need to make sure that this information is actually encoded into long term memory. According to this modal memory model, active rehearsal is necessary for this transition from short term memory to long term memory.

Users of Narrate must answer a series of in-game questions via multiple choice questions, drag drop boxes, and keyboard input. Hence, they are actively rehearsing the material that they are learning.

Further, it has been suggested that a stimulus' memorability, or recall ability, is dependent on the richness in which it was elaborated (McDaniel, 1984). The semantic network model has a high degree of explanatory power in cognitive psychology, and it proposes that all of one's knowledge is encoded in a hierarchical memory network (Reisberg, 2016). According to this theory, when information is learned in a rich context, like in the context of a compelling narrative, it allows for an individual to make more associations between the new information and the existing knowledge in her memory network, facilitating learning. This is important because a marker of thorough learning is the persistence of one's knowledge in memory.

Moreover, this presentation of learning instruction within a rich context allows for deep processing of information. The self referential effect is a form of deep level processing in which information is encoded into and recalled from long term memory more strongly if the stimulus is related to the individual than if the stimulus is unrelated to the individual (Reisberg, 2016). Deeper levels of informational processing (i.e. self referential processing) produce greater retention and long term memory traces than more shallow levels of informational processing (i.e. structural processing). To achieve this effect, we aimed to have a nice balance between relatability and drama within the storyline, and so we dramatized middle school scenarios. Also, we believe that the nature of first person narrative allows for self referential processing by encouraging the user to assume the role of the character. As aforementioned, many individuals introduced to OOP

find it difficult due to the abstract nature of OOP. We hope that this rich learning environment provides the appropriate architecture to attenuate some of these innate difficulties of OOP.

In summary, the narrative component of Narrate is central to its educational value; by having a compelling story, it aims to optimize the amount of information that the user proactively attends to, and this active engagement will allow for greater consolidation of the material into long term memory.

CHAPTER THREE

Basic Narrative Structure

Narration has been studied extensively. Vladmir Propp was one of the first individuals to formally study narrative theory. He analyzed the narrative structure of Russian folklore and identified 31 common structural units that sequentially occur in nearly all literature (Propp, 1968). He further identified 7 common character types. Much of contemporary literature is based on these archetypes. As a result, when individuals engage with some sort of narrative entertainment, they have expectations that a) the central story will progress in a familiar and logical manner and that b) character behaviors will resemble prototypical behaviors.

Take Star Wars as an example. It has all 7 of Propp's character types: Luke Skywalker (the hero), Dark Vader (the villain and the father), Obi-Wan Kenobi (the donor), R2D2 (the dispatcher), Leia (the princess), and Han Solo (the helper). These characters may not perfectly align with prototypical behavior, but they do not diverge so radically that they fatally undermine the audience's expectations for the storyline and characters. This example illustrates the persistence of Propp's ideas in contemporary literature.

Of course, not all literature is a manifestation of Propp's formula. However, the story progression of Narrate was implemented with strong reference to Propp's analysis. A user of Narrate should not become so engrossed with its story progression that it distracts from her learning. Instead, the narrative aspect should complement learning. In

fact, Kahneman's Capacity Theory argues that attentional priority is given to comprehension of narrative content over educational content (Resiberg, 2016). Hence, a mystical or exceedingly unrealistic narrative structure would not allow us to achieve the educational results that we aim for. Moreover, a mystical storyline can become unrelatable and therefore take away the learning benefits produced from relatable and familiar content. As a result, an elaborate narrative is not fitting for our purposes.

CHAPTER FOUR

Related Works

Non Narrative Learning Games that Teach Coding Concepts

There are several non-narrative learning games that have been successful in teaching coding concepts, and parts of Narrate's design and learning instruction were inspired by them.

Alice

Alice is a visual programming environment designed to teach introductory programming concepts and to provide first time exposure to OOP. Alice is further capable of teaching concepts like recursion and proper interface design. The interface of Alice initially required keyboard input but developed into a drag and drop system due to the high report of syntactic difficulties among its users (Kelleher, et al., 2002). These findings support our decision to eliminate syntax as a focus within Narrate's teaching instruction. Moreover, college and middle school users of Alice were reported to have significant learning in foundational programming concepts, but not necessarily in OOP concepts (Biju, 2013). Narrate explicitly aims to teach OOP concepts. Furthermore, researchers found that 62% of Alice middle school users created programs that created arbitrary motion, as if these users were without any goals when using the learning environment (Kelleher, et al., 2002). The lack of an explicit objective in Alice may weaken its effectiveness as a teaching tool for middle schoolers. Narrate's design requires

that the user make decisions in order to meet the goals of the game and progress through the storyline.

Star Wars Coding

Star Wars Coding targets kids as young as six years old to teach basic coding concepts. It has been successful in both interesting and teaching young students in programming (code.org). However, it neither specifically targets middle school girls nor does it teach more complex OOP concepts like inheritance and polymorphism. Additionally, in certain-gameplay it requires explicit understanding of syntax, which as previously mentioned, Narrate does not.

Scratch

Scratch is a visual programming environment that allows for interactive game design, similar to Star Wars Coding. It targets children 8 years and older and aims to further understanding in mathematics in addition to coding concepts (scratch.mit.edu). User's of Scratch can share their programs online, and Scratch has a large community of young users. Although it allows for simple storytelling, it lacks an elaborate narrative component. Moreover, Scratch does not explicitly introduce OOP concepts.

Narrative Learning Games

Storytelling is integral to Narrate's learning structure, and the aforesaid softwares lack a rich storytelling component. There are several existing architectures that have

incorporated narrative into their systems to provide an engaging and dynamic experience for their users, and they are discussed below.

Crystal Island–Outbreak

Crystal Island–Outbreak is an immersive inquiry-based software that uses narrative to encourage learning of microbiology science aligned with an 8th-grade science curriculum (John et al., 2014). Crystal Island differs from Narrate in part that its users are immersed in an environment in which its storyline and learning instruction are seamlessly blended. However, the inherent abstract nature of OOP makes it difficult to replicate this idea. Instead, Narrate's storyline serves to complement its learning instruction, and OOP concepts in themselves are not integral to the storyline. Nonetheless, Crystal Island uses elaborate narratives to contextualize investigative learning scenarios with social and ethical dimensions. It further uses narrative to encourage self-regulated learning.

Storytelling Alice

Storytelling Alice is based on Alice and likewise teaches foundational programming concepts and OOP concepts. It provides high-level animations catered towards the storytelling interest of female students by allowing its users to add stories to their programs. 50% of Storytelling Alice users created programs about relationships (Kelleher, et al., 2002), supporting our conviction that a narrative based game with a highly developed social component can be beneficial to learning. Furthermore, middle school users voluntarily spent more time playing Storytelling Alice than regular Alice

(Kelleher, et al., 2002). As mentioned earlier, difficult subject material requires greater practice and greater richness of the learning environment. Further, users were more likely to play Storytelling Alice in the future than the original Alice (Kelleher, et al., 2002). These findings suggest that learning will take place outside of the classroom, indicating that an engaging and narrative based interface can encourage continued and sustained learning. However, like Alice, Storytelling Alice lacks true object oriented inheritance and polymorphism.

Episode

Episode is an interactive narrative mobile app that has a library of over 150,000 interactive stories and 9 billion views (Episodes Interactive). Although it is not a learning game, it has large popularity among our target demographic. As of November 2018, approximately 45% of Episode's users in the United States fall in the age category 10-15 (Episodes Interactive). Further, 8 out of 10 users of Episode are female (Episodes Interactive). The user progresses through the game by selecting a choice in each scene. This allows the user to have some control over the outcome of her story and to establish a personal commitment to fulfill the demands and goals of the game.

Traditionally, narrative implies non participative viewership. Moreover, it has been shown that girls are very keen towards the social aspects of games, especially character development (Inkpen et al., 2014). Perhaps a major contributor to Episode's success is its ability to combine compelling storytelling that heavily emphasizes character development with active gameplay. Given Episode's success among our target

demographic, we hope that by modeling certain aspects of our interface to Episode's, Narrate will have similar entertainment value to Episode.

Summary

There are several learning games that teach introductory programming concepts. However, the major drawback to these existing systems is that they lack cohesive OOP instruction. Because OOP is abstract and difficult in nature, a game that offers repetitive practice in a rich learning environment seems like an appropriate medium to teach these concepts. Narrate hopes to be an innovative software that successfully teaches and interests its users in OOP via interactive narrative gameplay.

CHAPTER FIVE

App Design

Narrate was built using Ren'Py game engine and python. It can be downloaded on Mac, Windows, and Linux operating systems at: https://github.com/MicahDadson/Narrate.

Participant data was stored in the researchers' database and assent was obtained within the app before allowing the user access to the gameplay. Further, each user was given a unique ID and no identifying information was obtained.

The storyline of Narrate involves a girl starting at a new middle school, and the user must pick choices to drive the story.

The layout of Narrate is composed of four mandatory episodes- each of which has its own set of learning outcomes, an optional fifth episode, and an end of game survey. The user progresses through the story by completing a series of questions via multiple choice, drag and drop boxes, and keyboard input (Figures 3-10). The user plays one episode per day for a five day period. We require participation along a five day period to avoid participant fatigue due to the moderate length of each episode.

The learning outcomes of Narrate are based on the C++ Early Objects ninth edition introductory programming textbook by Tony Gaddis, Judy Walters, and Godfrey Muganda.

Design of Episodes 1-4

Each of the mandatory episodes has an identical pretest and posttest based on the specific learning outcome of that episode (Table 1). The first episode has a 9 question pretest/posttest and its learning outcomes are programming basics. The second episode has a 10 question pretest/posttest and its learning outcomes are abstraction and encapsulation. The third episode has a 3 question pretest/posttest and its learning outcome is inheritance. The fourth episode has a 3 question pretest/posttest and its learning outcome is polymorphism.

Moreover, each episode has a series of in-game test questions that aim to measure and reinforce learning as well as non test questions that simply aim to provide entertainment value and drive the story. Episodes 1-4 can each be played once and each take 15-45 minutes to complete.

Episode	Learning Outcome
1	Programming Basics
2	Abstraction and Encapsulation
3	Inheritance
4	Polymorphism
5	Cumulative Review

Table 1: The learning outcomes of the five episodes

Design of the Optional Episode 5

The fifth episode is optional and consolidates information from the first four episodes (Table 1). It presents the user with 5 random questions from a 10 question bank. The outcome of this supplemental episode consists of three possible scenarios; the more accurate the user answers the in-game questions, the more favorable outcome she receives. Episode 5 can be played an unlimited amount of times and takes approximately 5 minutes to complete. There is neither a pretest nor a posttest within this episode.

Survey Design

The end of game survey consists of five questions that are answered on a 5 point Likert scale (Table 2). These questions serve to qualitatively measure Narrate's entertainment value and ability to serve as a primer for future computer science instruction.

Scale	Questions	Hypotheses
Entertainment value 1 = strongly disagree 5 = strongly agree	S-1. I would recommend Narrate to a friend S-2. I enjoyed playing Narrate S-3. I would play this game with a different storyline	H0: An individual who played environment 5 one or more times did not find Narrate entertaining (1, 2, 3) H1: An individual who played environment 5 one or more times did find Narrate entertaining (4, 5)
Increased interest in computer science 1 = strongly disagree 5 = strongly agree	S-4. Narrate increased my interest in computer science S-5. It would be cool to code a game like Narrate	H0: An individual who played environment 5 one or more times did not have an increased interest in computer science (1, 2, 3) H1: An individual who played environment 5 one or more times had an increased interest in computer science (4, 5)

Table 2: Questions from the end of game survey (S) that explicitly measure Narrate's ability to increase interest in computer science and Narrate's entertainment value

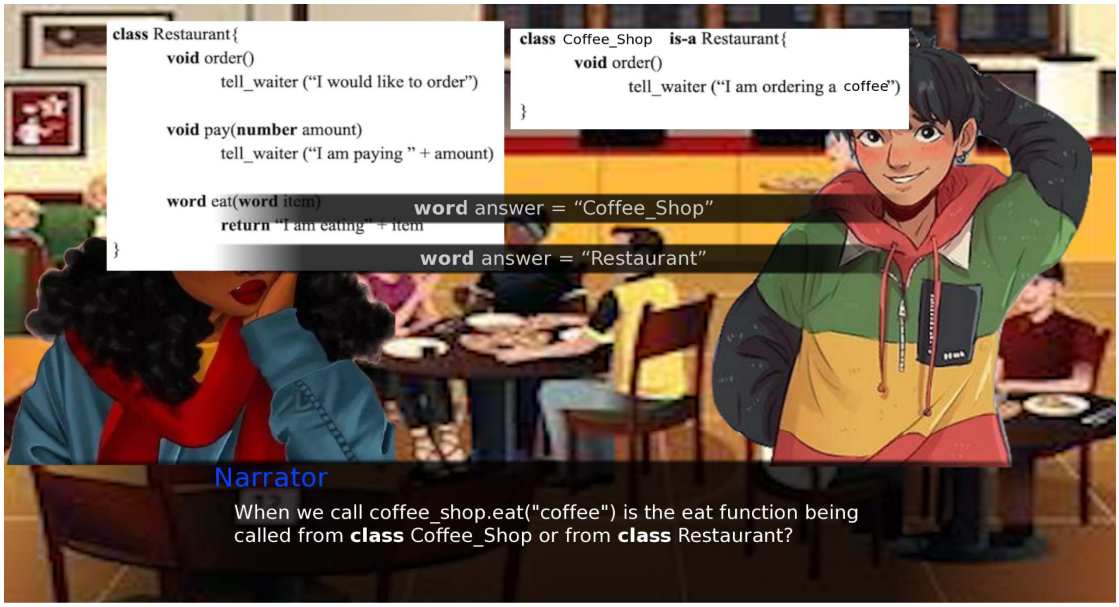

Figure 3: An example of an in-game testing question that utilizes multiple choice to test the user's understanding of inheritance. Shows how the storyline line is catered towards the relationship interests of middle school girls and how testing questions are asked within the context of an immersive environment.

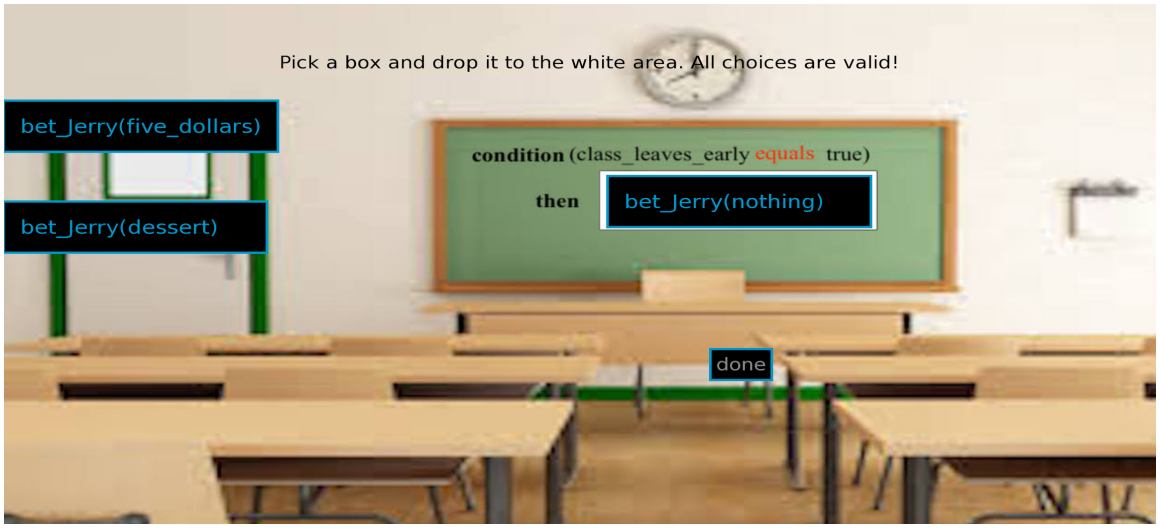

Figure 4: An example of an in-game question that utilizes drag and drop boxes to introduce the user to conditionals and function parameters. Shows how the user is first introduced to topics with easy questions that reward her participation. This is important because it has been shown that constant rewards are beneficial to learning (C. Dweck, 2014).

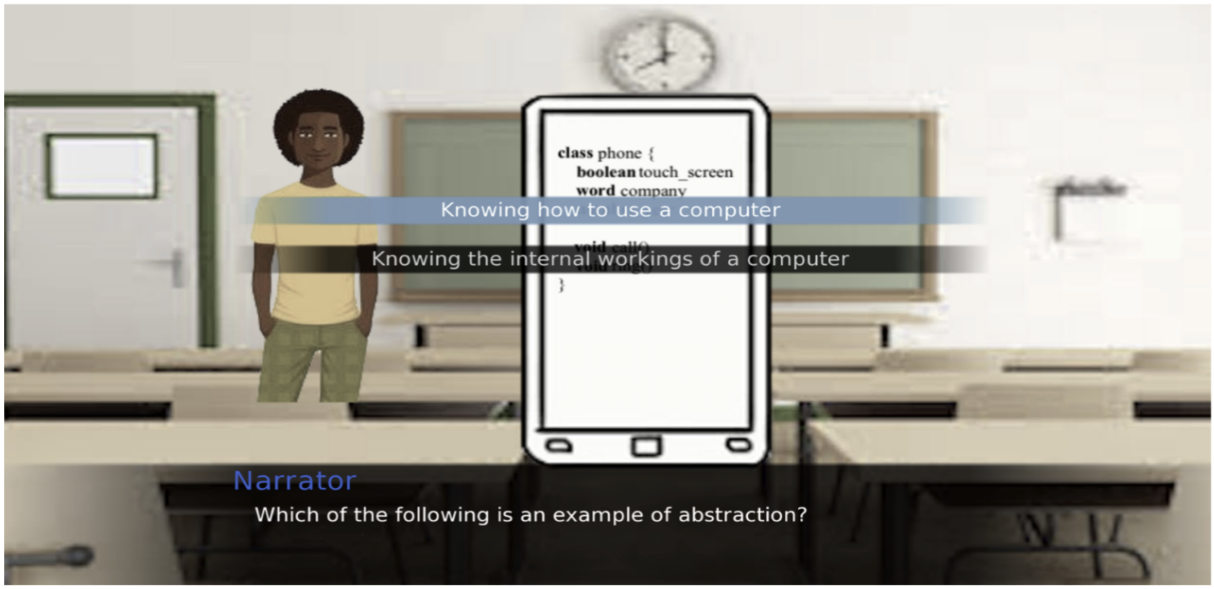

Figure 5: An example of an in-game testing question that utilizes multiple choice to test whether the user understands abstraction. Shows how the user is explicitly tested on her understanding of conceptual topics.

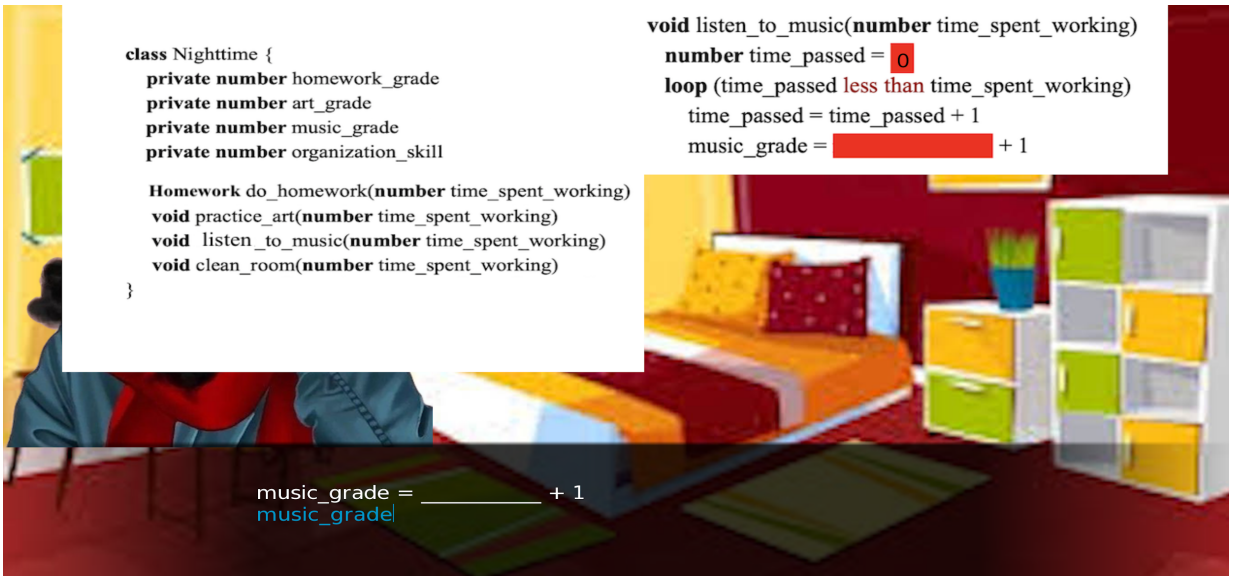

Figure 6: An example of an in-game testing question that utilizes keyboard input to explicitly test the user's understanding on looping. Shows how the user is explicitly tested on her understanding via recall retrieval.

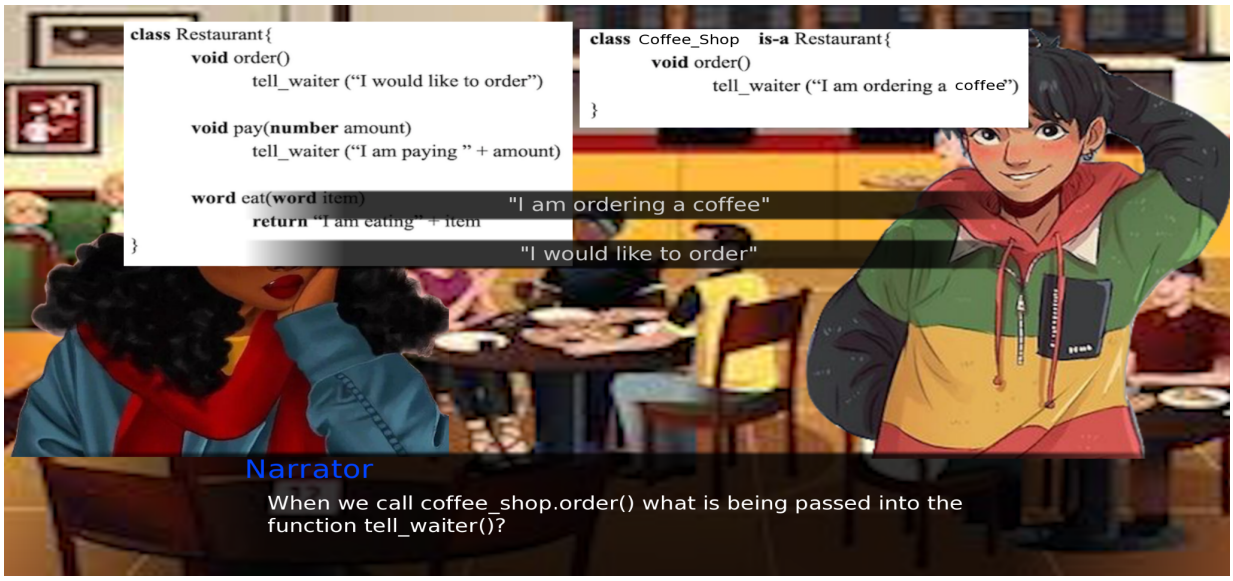

Figure 7: An example of how we test the user over polymorphism in a manner that compliments the storyline.

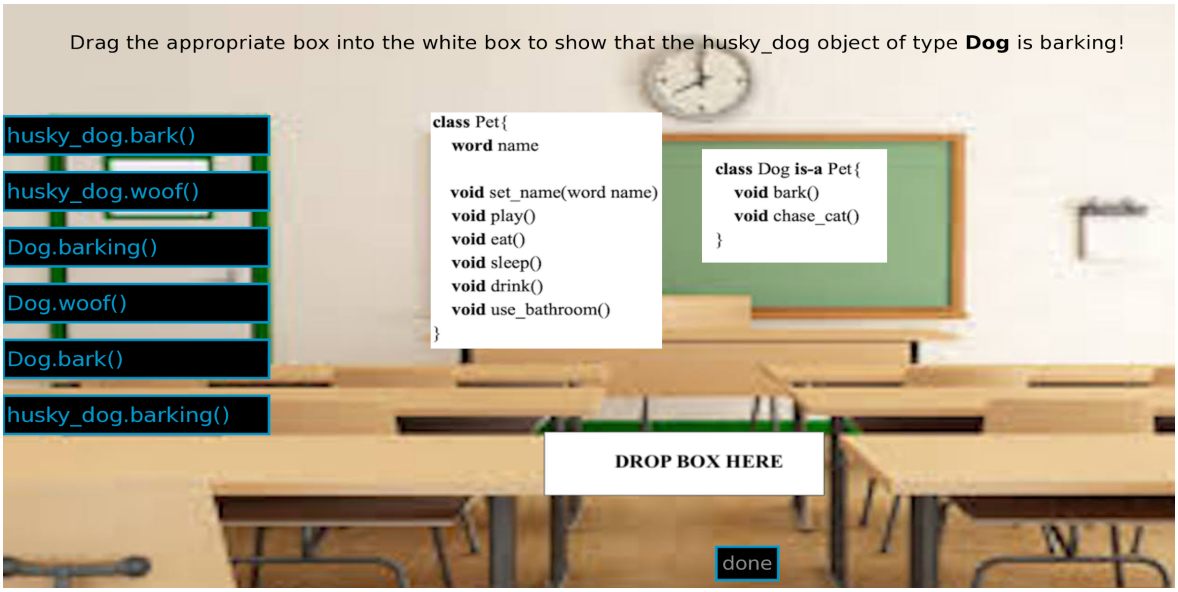

Figure 8: An example of an in-game testing question that utilizes drag and drop boxes to test the user's understanding of inheritance. Shows how explicit testing questions are associated with real life concepts to aid understanding.

Figure 9: An example of how we develop the social structure in a somewhat dramatic way to satisfy the storytelling interests of middle school girls.

Figure 10: An example of a pretest question that a user must answer before she may begin the episode's storyline.

CHAPTER SIX

Experimental Design

Twenty five students, males and females, from an introduction to anthropology course at Baylor University were recruited to participate in our study, and these participants were incentivized with extra credit. The students played each episode at home on their own time. 25 students completed episode 1, 16 students completed episode 2, 13 students completed episode 3, 12 students completed episode 4, 6 students completed episode 5, and 9 students completed the end of game survey. Many of these students experienced technical difficulties, and that may be a partial explanation for the incomplete follow through.

Five hypotheses, based on the experimental design, were tested.

Hypothesis One: Subjects that are intrinsically motivated to play the optional episode five will self-report the overall entertainment value of Narrate higher compared to subjects that do not play the optional episode five.

Hypothesis Two: Subjects that are intrinsically motivated to play the optional episode five will self report having an increased interest in computer science after playing Narrate unlike subjects that do not play the optional episode five.

We want to determine if intrinsic motivation is predictive of self reported entertainment value and/or self reported interest in computer science. Because our subjects are busy college students pursuing education in an unrelated field, we would expect this group to have little intrinsic motivation to play the optional module. Hence,

we think that any voluntary play is revealing of Narrate's ability to create intrinsic motivation in its users. If these two hypotheses are proven true, then if a large proportion of our sample population opted to play the optional module, Narrate might have a generalized ability to entertain while simultaneously serving as a computer science primer. Additionally, if these hypotheses are validated, then perhaps narrative based learning games in general have the ability to effectively engage and provoke sustained interest in their audiences for the subject material at focus. If so, narrative based learning games may be valuable instruments for introducing various abstract or difficult STEM concepts in a positive and enjoyable manner.

Hypothesis Three: Subjects will have greater performance on the posttest compared to the pretest.

We want to effectively teach OOP concepts to our subjects. Higher performance on the posttest versus pretest for each episode is indicative of short term learning. If the results reveal improvement for the learning outcomes, narrative based learning games may be effective tools to teach OOP and possibly other STEM concepts.

Hypothesis Four: Average posttest performance is linearly correlated with average performance on episode five.

Because episode five can be attempted many times and each attempt may present the subject with new questions, we average the subject's performance on episode five, standardizing the data.

Each posttest is taken directly after completing the in-game episode and measures short term learning. Episode five asks questions from episodes 1-4 and is completed on day 5, measuring more "long term" learning. According to Ebbinghaus' Forgetting

Curve, memory for unrehearsed material exponentially declines as a function of time, with only 20% of learned material remembered on Day 3 (Resiberg, 2016). However, if average performance on episode 5 is comparable to average posttest performance, then this may suggest that the narrative framework of Narrate is effective in allowing the learned material to persist in memory. We perform a linear regression test because we suspect that Narrate's design allows for deep informational processing, avoiding the large drop in knowledge that might otherwise be expected.

Hypothesis five: Individuals that correctly answer an in-game question set on their first attempt will perform better on the mapped posttest question(s) compared to individuals that have multiple incorrect attempts for an in-game question set.

We want to assess whether the in-game questions map to the posttest evaluation questions. Of course, it is possible that having multiple attempts for a question set is instructive, allowing individuals to have high performance on the corresponding posttest question. However, we believe that subjects that correctly answer a question set on the first attempt have thorough understanding of the target concept, and therefore that they will have strong performance on the related posttest question(s) if there is a proper mapping.

There are a few exceptions to this hypothesis. In environment 2, instead of allowing only one attempt for the question set mapped to the "functions" concept, we allow two attempts. Likewise, in environment 4, we allow for three attempts for the question set mapped to the "polymorphism" topic. We allow for these exceptions because some of the questions within these question sets are instructive, meaning we required

input from the keyboard, expecting the user to answer incorrectly on the first attempt. Through their wrong response, they were explained how to correctly answer the question.

CHAPTER SEVEN

Results and Discussion

Hypothesis One: Subjects that are intrinsically motivated to play the optional episode five will self report the overall entertainment value of Narrate higher compared to subjects that do not play the optional episode five

Hypothesis Two: Subjects that are intrinsically motivated to play the the optional episode five will self report having an increased interest in computer science after playing Narrate unlike subjects that do not play the optional episode five

To measure hypothesis one, Fisher's exact test was performed to determine whether a subject's intrinsic motivation to play the optional episode 5 was contingent on her ratings of Narrate's entertainment value (S-2, S-3)(Table 3). To measure hypothesis two, Fisher's exact test was performed to determine whether a subject's intrinsic motivation to play the optional episode five was contingent on her ratings of Narrate's ability to increase her interest in computer science (S-4, S-5) (Table 3). The test was run separately on survey questions S-2 through S-5.

Question	% that Agreed or Strongly Agreed with the Question Statement	Corresponding Hypothesis	P-value
S-1. I would recommend Narrate to a friend	66.7%	-	-
S-2. I enjoyed playing Narrate	88.9%	Two	0.56
S-3. I would play this game with a different storyline	88.9%	Two	1.0
S-4. Narrate increased my interest in computer science	**55.6%**	**Three**	**0.04**
S-5. It would be cool to code a game like Narrate	77.8%	Three	0.83

Table 3: Summary statistics for hypotheses one and two

We were able to validate hypothesis three, finding a correlation between individuals that were intrinsically motivated to voluntarily play episode five and self-reported increased interest in computer science, suggesting that non compulsory learning will take place if an individual is interested in the subject material. The ability to encourage non compulsory learning is significant because we want to sustain interest in our subjects throughout their educational careers. Moreover, it is not unsurprising that some subjects did not have an increased interest in computer science; Narrate was designed specifically for middle school girls, not college aged men and women. The fact that half of the participants did not have an increased interest in computer science could be attributed to the fact that our college anthropology subjects already had decided career

paths. These findings motivate us to target middle school students so that we can interest them in computer science before they decide on a career path.

Although we could not correlate intrinsic motivation with self-reported entertainment ratings, most subjects self-reported finding Narrate entertaining and self-reported that it would be "cool" to code a game like Narrate. We designed Narrate to provide entertainment value and to spark interest in computer science, and our results reveal that interactive narrative games can be effective tools to instill interest in students while simultaneously entertaining them. Furthermore, it must be emphasized that Narrate was designed for a different demographic, and the fact that our subjects found Narrate entertaining and coding "cool" highlights the value that a compelling and well developed narrative can add to teaching regardless of its intended audience. Although we cannot accept our hypotheses that those that played the optional module 5 did so due to the fact they found Narrate entertaining and/or coding interesting, it is clear that Narrate has entertainment value with ability to pique interest in computer science.

Furthermore, it should be noted that many of our subjects had technical difficulties. As a result, one's decision not to play this optional episode may have been due to external reasons unrelated to Narrate's entertainment or computer science primer value. Lastly, our sample size for this hypothesis (N=9) is quite small. We need a larger sample size to determine whether we should reject or accept these two hypotheses.

Hypothesis Three: Subjects will have greater performance on the posttest compared to the pretest

To measure hypothesis three, a one-tailed t-test was performed on the pretest and posttest data of each episode (Tables 4-7). Although episode 2 is the only episode with a significant p-value, each episode had improvement between the pretest and posttest. The anthropology students were given extra credit for simply attempting each of the mandatory episodes, so it is promising that we observed improvement for each episode when effort was not a required condition. These results imply that Narrate is an effective tool in teaching OOP concepts, and that narrative based games in general may be effective tools to teach difficult and abstract STEM concepts.

-	Pretest	Posttest
Mean	4.44 (49.3%)	4.72 (52.4%)
Standard Deviation	1.30	1.61
Standard Error of the Mean	0.265	0.329
N	25	25

P Value	0.2556
pretest/posttest	9 questions

Table 4: T-test for measuring the difference in means between the pretest and posttest scores for instruction about programming basics

-	Pretest	Posttest
Mean	3.81 (38.1%)	5 (50%)
Standard Deviation	1.94	1.83
Standard Error of the Mean	0.502	0.474
N	16	16

P Value	**0.048**
pretest/posttest	10 questions

Table 5: T-test for measuring the difference in means between the pretest and posttest scores for instruction about abstraction and encapsulation

-	Pretest	Posttest
Mean	1.692 (56.4%)	2 (67.7%)
Standard Deviation	0.910	1.038
Standard Error of the Mean	0.263	0.3
N	13	13

P Value	0.223
pretest/posttest	3 questions

Table 6: T-test for measuring the difference in means between the pretest and posttest scores for instruction about inheritance

-	Pretest	Posttest
Mean	1.083 (36.1%)	1.5 (50%)
Standard Deviation	0.759	1.041
Standard Error of the Mean	0.229	0.314
N	12	12

P Value	0.148
pretest/posttest	3 questions

Table 7: T-test for measuring the difference in means between the pretest and posttest scores for instruction about polymorphism

Hypothesis Four: Average posttest performance is linearly correlated with average performance on episode five

To measure hypothesis 4, that average posttest performance is linearly correlated with average performance on episode 5, a linear regression test was performed (Figure 18). We processed the data so that only individuals that completed all five episodes were considered in the analysis, and we have a sample size of 6 subjects. Because this sample size is too small to draw conclusions, we neither reject nor accept this hypothesis and hope to get better data with our middle school subjects.

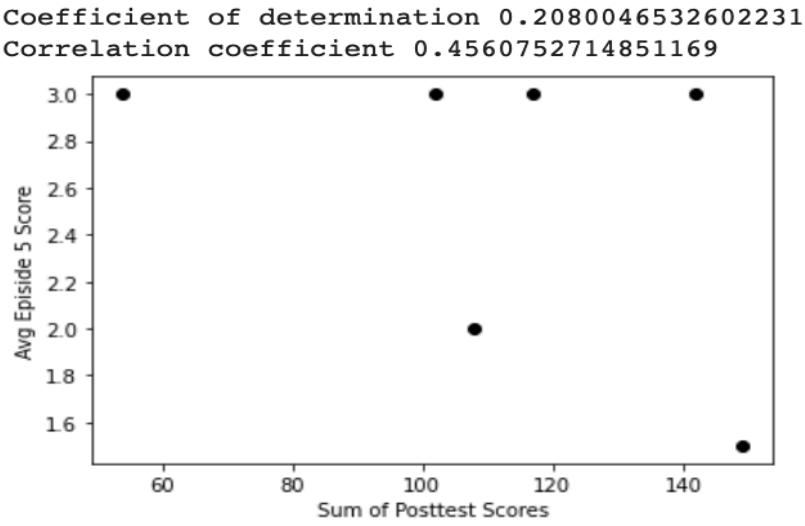

Figure 11: Linear Regression to test the relationship between Average posttest performance and average performance on episode five

Hypothesis five: Individuals that correctly answer an in-game question set on their first attempt will perform better on the mapped posttest questions compared to individuals that have multiple incorrect attempts for an in-game question set

To measure hypothesis five, a one tailed t test was performed between the posttest performance of the group that averaged one attempt on a question set and the posttest performance of the group that averaged multiple attempts on a question set (Tables 7-13). As previously explained, there are two exceptions to this hypothesis; question sets consisting of instructive questions allowed for more than one attempt (Tables 9 and 13). Overall, individuals that correctly answered an in-game question set on their first attempt performed better on the posttest than individuals that did not correctly answer an in-game question set on their first attempt. These results suggest that if an individual knew the correct answer during gameplay, then she knew the correct answer during the posttest, suggesting that the in-game question sets properly mapped to the posttest evaluation

questions. This is important because we show that the posttest questions are appropriate evaluators of the in-game question sets. Because evaluation is an effective teaching tool, we show that we are further able to reinforce the in-game learning material at the time of posttest evaluation.

Moreover, to reference the modal memory model, these results imply that individuals that correctly answered an in-game question set on their first attempt attended to the learning instruction and were able to appropriately recall this information at the time of in-game testing. The active rehearsal of this material within the episode allowed them to appropriately recall this information at the time of posttest testing.

Interestingly, some concepts had quite high p-values, and this could be attributed to a fluke in our sample population or an error in our experimental design. Subsequent testing with middle school students will provide more information.

	Posttest performance for group with avg(attempts) = 1	Posttest performance for group with avg(attempts) > 1
Mean	1.5	1.056
Standard deviation	0.5	0.705
Standard error of the mean	0.224	0.171
Population size	6	18

P Value	**0.071**

Table 7: T-test score for variables and datatypes concepts

	Posttest performance for group with avg(attempts) = 1	Posttest performance for group with avg(attempts) > 1
Mean	1.308	0.833
Standard deviation	0.821	0.799
Standard error of the mean	0.237	0.241
Population size	13	12

P Value	**.087**

Table 8: T-test score for branching and looping concepts

	Posttest performance for group with avg(attempts) = 3	Posttest performance for group with avg(attempts) > 3
Mean	0.667	1.154
Standard deviation	0.471	0.984
Standard error of the mean	0.333	0.274
Population size	3	13

P Value	0.846

Table 9: T-test score for functions concept

	Posttest performance for group with avg(attempts) = 1	Posttest performance for group with avg(attempts) > 1
Mean	2.333	1.333
Standard deviation	0.471	0.624
Standard error of the mean	0.333	0.188
Population size	3	12

P Value	**0.035**

Table 10: T-test score for classes and objects concepts

	Posttest performance for group with avg(attempts) = 1	Posttest performance for group with avg(attempts) > 1
Mean	2	1.455
Standard deviation	0	0.655
Standard error of the mean	0	0.207
Population size	5	11

P Value	**0.012**

Table 11: T-test score for encapsulation and abstraction concepts

	Posttest performance for group with avg(attempts) = 1	Posttest performance for group with avg(attempts) > 1
Mean	2	2
Standard deviation	1	1.044
Standard error of the mean	1	0.330
Population size	2	11

P Value	0.5

Table 12: T-test score for inheritance concept

	Posttest performance for group with avg(attempts) = 3	Posttest performance for group with avg(attempts) > 3
Mean	2.25	1.125
Standard deviation	1.299	0.599
Standard error of the mean	0.75	0.227
Population size	4	8

P Value	**0.116**

Table 13: T-test score for polymorphism concept

CHAPTER EIGHT

Conclusion

Interactive narrative games require active participation and have the powerful ability to captivate the attention of their users. By designing an interactive narrative game that caters to middle school female students, we hoped that our learning game, Narrate, could entertain as well as teach and interest female students in OOP programming. The results indicate that interactive narrative gameplay is effective in teaching various coding concepts, in interesting users in computer science, and in entertaining users. Although not all of our learning outcomes had statistically significant improvement, we had improvement for each learning outcome, showing that interactive narrative can effectively teach difficult subjects. Moreover, we were able to correlate intrinsic motivation with increased interest in computer science, showing that Narrate has the ability to encourage non compulsory learning. This is significant because it suggests that a positive and an engaging introduction to a difficult subject via interactive narrative can encourage self initiated future learning. Although we could not strongly correlate intrinsic motivation with self reported entertainment ratings, we did show that Narrate has strong entertainment value among our college subjects and that most subjects thought it would be “cool” to code a game like Narrate. These findings suggest that interactive narrative may inherently make subject material interesting and engaging regardless of the intended audience. We could not show that Narrate allowed for long term retention of learning material due to an insufficient sample size, but we will get more data for this hypothesis

once we execute this study again with our middle school subjects. Further, we showed that the in-game questions are mapped to the posttest questions. Since testing is an effective teaching tool, we indicated that Narrate is effective in both teaching and assessing knowledge at the time of posttest evaluation. Despite the fact we did not have the ideal demographic, these results are quite promising. We believe that our results will at the very least be comparable when we conduct our study again with middle schoolers as our subjects.

WORKS CITED

About Us. Code.org. https://code.org/about.

About. Scratch. https://scratch.mit.edu/about.

Biju, S. M. (2013). *Taking Advantage of Alice to Teach Programming Concepts*. https://journals.sagepub.com/doi/pdf/10.2304/elea.2013.10.1.22.

Dasgupta, Nilanjana, and Jane G. Stout. "Girls and Women in Science, Technology, Engineering, and Mathematics." *Policy Insights from the Behavioral and Brain Sciences*, vol. 1, no. 1, 2014, pp. 21–29., doi:10.1177/2372732214549471.

Dweck, C. (2014, November). *The power of believing that you can improve* [Video]. TED Conferences. https://www.ted.com/talks/carol_dweck_the_power_of_believing_that_you_can_improv?language=en

Digest of Education Statistics, 2016. National Center for Education Statistics (NCES 2017-094) Home Page, a part of the U.S. Department of Education. (2016). https://nces.ed.gov/programs/digest/d16/tables/dt16_322.50.asp?current=yes.

Episode Interactive. Episode. https://www.episodeinteractive.com/.

Kelleher, C., Pausch, R., & Kiesler, S. (2007). *Storytelling Alice Motivates Middle School Girls to Learn ...* https://www.cse.wustl.edu/~ckelleher/StorytellingCHI.pdf.

Inkpen, K. A. O. (1993, November 30). *"We Have Never-Forgetful Flowers in Our Garden": Girls' Responses to Electronic Games.* Journal of Computers in Mathematics and Science Teaching. https://eric.ed.gov/?id=EJ498308.

James, R., Wardrip-Fruin, N., Mateas, M., Isbister, K., Hong, T., & Kaltman, E. (2016).*GameNet and GameSage: Videogame Discovery as Design Insight*. ResearchGate. https://www.researchgate.net/publication/303329458_GameNet_and_GameSage_Videogame_Discovery_as_Design_Insight.

John, N. L., Shores, L. R., & Hoffmann, K. F. (2014). *Self-Regulation and Gender Within a Game-Based Learning ...*https://pdfs.semanticscholar.org/59d5/350d7c2efc9586b3d236031aca2d9df90b00.pdf.

León, C. (2016). An architecture of narrative memory. *Biologically Inspired Cognitive Architectures*, *16*, 19–33. https://doi.org/10.1016/j.bica.2016.04.002

Mateas, M., & Stern, A (2003). *Façade: An Experiment in Building a Fully-Realized ...* https://www.cc.gatech.edu/fac/Charles.Isbell/classes/reading/papers/MateasSternGDC03.pdf.

McDaniel, M. A. (1984). The role of elaborative and schema processes in story memory. *Memory & Cognition*, *12*(1), 46–51. https://doi.org/10.3758/bf03196996

Prins, R., Avraamidou, L., & Goedhart, M. (2017). Tell me a Story: the use of narrative as a learning tool for natural selection. *Educational Media International*, *54*(1), 20–33. https://doi.org/10.1080/09523987.2017.1324361

Propp, V. (1968). Morphology of the Folktale. https://doi.org/10.7560/783911

Reisberg, D. (2016). *Cognition: exploring the science of the mind*. W.W. Norton.

Richards, W., Winston, P. H., & Finlayson, M. A. (2009, December 17). Advancing Computational Models of Narrative. https://dspace.mit.edu/handle/1721.1/50232.

U.S. Bureau of Labor Statistics. (2019, September 1). *Industries with the fastest growing and most rapidly declining wage and salary employment*. U.S. Bureau of Labor Statistics. https://www.bls.gov/emp/tables/industries-fast-grow-decline-employment.htm.

Vu, Shana. “Cracking the Code: Why Aren't More Women Majoring in Computer Science?” *UCLA*, UCLA, 27 June 2017, newsroom.ucla.edu/stories/cracking-the-code:-why-aren-t-more-women-majoring-in-computer-science.

INDEX

Five Environment Layout

The index includes the learning outcomes and pretest/posttest questions of each episode. To view the learning outcomes, pretest/posttest questions, in-game questions, and storyline of each episode please go to: https://github.com/MicahDadson/Narrate.

The learning outcomes of Narrate are based on the C++ Early Objects ninth edition introductory programming textbook by Tony Gaddis, Judy Walters, and Godfrey Muganda. Narrate is composed of five different environments, each of which has its own set of learning outcomes.

Environment 1

In episode one, the user arrives at a new school and makes new friends. One of her new friends pulls a stunt that gets her whole class in trouble.

Learning Outcomes: Programming Basics

After completing the first environment the user should be able to :

1. Understand what a variable is
2. Understand that there are different data types
3. Understand that variables can be assigned values
4. Understand variable declaration
5. Recognize relational operators
6. Understand conditional statements

7. Understand branching
8. Understand looping
9. Recognize functions

Pretest question #1: What is a variable?

Something that stores information. It can be changed.

Something that stores information. It cannot be changed.

An object defined by the programmer

Pretest question #2: Which of the following is the data type: word draw ()

word

draw

()

Pretest question #3: What does the following piece of code do: age = 7

it checks to see if age is equal to 7

it assigns age the value of 7

it assigns 7 the value of age

Pretest question #4: Which of the following is a variable declaration without a variable initialization?

number age = 7

age = 7

number age

none of the above

Pretest question #5: What does the relational operator do?

It creates a relationship between two things

It tests the relationship between two things

It eliminates the relationship between two things

Pretest question #6: What is a conditional statement?

It allows a piece of code to execute only if a certain condition is met

It allows a piece of code to execute regardless if a certain condition is met

It causes the computer to turn off if a certain condition is met

Pretest question #7: What is branching?

It allows certain pieces of code to execute depending on if the condition is met

It allows every piece of code to execute regardless if the condition met

It allows certain pieces of code to execute while a condition is met

Pretest question #8: What is looping?

It allows every piece of code to execute regardless if the condition met

It allows code to execute while a condition is met

It allows code to execute once if a condition is met

Pretest question #9: Which of the following is a function?

number age = 8

word name = “Sarah”

void set_name ()

number get_age ()

two of the above

Environment 2

In episode two, the user decides to audition for the “High School Musical” production with her friends. She is given the role of “Gabriella’s understudy” and can only hope that the lead for “Gabriella” gets sick so that she can star with her crush Jerry, who receives the role of “Troy.”

Learning Outcomes: Abstraction and Encapsulation

After completing the second environment the user should be able to :

1. Understand function implementation
2. Understand function parameters
3. Know what abstraction is
4. Know what a class is
5. Know what encapsulation is
6. Know what an object is
7. Understand the dot operator
8. Understand function return types
9. Understand that a parameter list can contain zero or many values

10. Understand variable manipulation

Pretest question #1: Which of the following contains the function implementation?

word get_name()

word get name()

    return my_name

void set_name (word a_name)

none of the above

Pretest question #2: Which of the following is the function parameter: void draw (word object)

void

draw

object

Pretest question #3: Which of the following is an example of abstraction?

Knowing how to use a computer

Knowing the inner workings of a computer

The procedure of how a computer is built

Pretest question #4: What is a class?

A data type that can be defined by the programmer

A data type that cannot be defined by the programmer

A variable defined by the programmer

A variable that cannot be defined by the programmer

Pretest question #5: What is encapsulation?

The separating of an object's data

The bundling of an object's data into a single unit

Allowing an object's data to be accessed by anyone

Pretest question #6: What is an object?

An instance of a class

Another name for a class

Another name for data type

Pretest question #7: Why do we use the dot operator?

To access the data of an object

To access only the functions of an object

To access only the functions of an variable

Pretest question #8: Which of the following is the return type of the following: Person set_name (word name)

Person

set_name

word name

Pretest question #9: Which of the following is an appropriate number of parameters that a parameter list can have?

0

0 and 1

0, 1 and 2

0, 1, 2…. 20

all of the above

Pretest question #10: a = 10

a = a + 5

After the second operation is performed, what value will be stored in a?

0

5

10

15

all of the above

Environment 3

In episode three, the user's parents take her and her siblings to a pet shop to adopt a pet.

Learning Outcomes: Inheritance

After completing the third environment the user should be able to :

1. Understand the is-a relationship
2. Understand the child class
3. Understand the parent class

Pretest question #1: Which of the following is not an appropriate is-a relationship

A car is-a vehicle

A tire is-a vehicle

A baby is-a human

All of the above contain an appropriate is-a relationship

Pretest question #2: If class A is-a class B then:

class A inherits the functionality of class B

class B inherits the functionality of class A

Neither class inherits the functionality of the other

Pretest question #3: Does the parent class have access to the functionality of the child class?

Yes

No

Environment 4

In episode four, Valentine's day is approaching and the user finds out that one her best friends, Jerry, has a crush on her.

Learning Outcomes: Polymorphism

After completing the fourth environment the user should be able to :

1. Understand what polymorphism is
2. Understand the relationship between inheritance and polymorphism
3. Know what override means

Pretest question #1: What is polymorphism?

The ability to process objects differently depending on their class

The ability for a variable to be assigned several different values

The ability for a function to be called more than once

Pretest question #2: Can polymorphism exist between class A and class B if they do not have an is-a relationship?

no

yes

yes if we override the classes

Pretest question #3: What does override mean?

It allows the derived class to provide a specific implementation of a method that is provided by parent class

It allows the general class to provide a specific implementation of a method that is provided by the parent class

It allows class A to access the data of class B if there is not an is-a relationship

Supplemental Module

In episode 5, the user goes to the Spring Dance with her crush, Jerry. Five questions will be selected from a bank of ten questions and given to the user for completion. The outcome of the module depends on the user's performance on the quiz with the most optimal outcome resulting if the user answers 5/5 correctly, the moderate outcome resulting if the user answers 3/5 or 4/5 questions correctly, and the least favorable outcome resulting if the user answers 1/5 or 2/5 corrections correctly.

It's finally time for Spring Dance! You and Jerry have been dating for two months and you both can't wait to go to your first dance together! It's going to be such an iconic night! In order to have the most fun night, try to answer the following 5 questions as accurately as possible!

Question #1: You, Moriah, and bestfriend_name have decided to match my wearing red dresses! What is the difference between dress_color = “red” and x equals “red”?

dress_color = 7 checks to see if dress_color and 7 are the same value while dress_color equals 7 assigns to dress_color the value of 7

dress_color equals 7 checks to see if dress_color and 7 are the same value while dress_color = 7 assigns to dress_color the value of 7

There is no difference

Question #2: Your mom told you that if you clean your room the day before the dance, then she will take you to get your nails done. Pick the answer that causes the function nail_salon() to be called if you cleaned your room. If your room is not clean, then do_own_nails() will be called.

a. loop (room_is_clean equals true)

   then nail_salon

   alternative

   then do_own_nails()

b. condition (nail_salon equals true)

   then room_is_clean()

   alternative

   then do_own_nails()

c. condition (room_is_clean equals true)

then nail_salon()

alternative

then do_own_nails()

Question #3: It's time to get ready! Pick the answer that appropriately creates an object of class SpringDance and calls it's getReady() function

a. SpringDance dance

dance.getReady

b. SpringDance dance

dance.getReady()

4. SpringDance.getReady()

Question #4: Jerry has just arrived at your house. Your parents want to take photos for a duration of ten minutes. Select the answer appropriate to demonstrate this.

a. time = 10

loop (time not equals 10)

then take_photos()

time++

b. time = 0

loop (time not equals 10)

then take_photos()

time++

c. time = 0

loop (time equals 10)

then take_photos()

time++

Question #4: After pictures, your parents decided to take you and Jerry out to eat! For dessert you decided to get pie. Observe the following classes:

```
class Food {
        word name
        number amount
        void eat()
                display ("I am eating")
}

class Pie{
        word name
        number amount
}
```

If I create an instance of class Pie called pie, what happens if I do: pie.eat()

a. “I am eating” will be displayed
b. This is an error because an is-a relationship was not declared between class Pie and class Food

Question #6: Time to go shopping for shoes! If class Heels is-a class Shoe, then:

a. an object of class Shoe can access the functions and variables of class Heels
b. an object of class Heels can access the functions and variables of class Shoe
c. neither is correct

Question #7: Is class Heels the derived class or general class?

a. derived class
b. general class

Question #8: Moriah and best_friend name ask you if you want to have a sleep over after the dance and of course you do! Select the answer that makes sense.

a. boolean sleepover = “yes”
b. string sleepover = “yes”
c. boolean sleepover equals “yes”
d. string sleepover equals “yes”

Question #9: You know how to do your spring dance makeup despite not knowing the actual way the makeup is made. This is an example of:

a. abstraction
b. parametrization
c. overriding

Question #10: Your parents got you a gift but they are keeping it a surprise and won't tell you. This hiding of data by making it private is an example of:

a. encapsulation
b. abstraction
c. implementation