

PINEHILL: A NOVEL APPROACH TO COMPUTER
AIDED LANGUAGE
LEARNING

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Kyu Han Koh

Certificate of Approval:

Kai H. Chang
Professor
Computer Science and
Software Engineering

Cheryl D. Seals, Chair
Assistant Professor
Computer Science and
Software Engineering

Juan E. Gilbert
Associate Professor
Computer Science and
Software Engineering

William Buskist
Distinguished Professor
Psychology

George T. Flowers
Interim Dean
Graduate School

PINEHILL: A NOVEL APPROACH TO COMPUTER
AIDED LANGUAGE
LEARNING

Kyu Han Koh

A Thesis
Submitted to
the Graduate Faculty of
Auburn University
in Partial Fulfillment of the
Requirements for the
Degree of
Master of Science

Auburn, Alabama
May 10, 2007

PINEHILL: A NOVEL APPROACH TO COMPUTER
AIDED LANGUAGE
LEARNING

Kyu Han Koh

Permission is granted to Auburn University to make copies of this thesis at its discretion,
upon the request of individuals or institutions and at their expense.
The author reserves all publication rights.

Signature of Author

Date of Graduation

THESIS ABSTRACT

PINEHILL: A NOVEL APPROACH TO COMPUTER

AIDED LANGUAGE

LEARNING

Kyu Han Koh

Master of Science, May 10, 2007
(B.S., Soongsil University, 2004)

60 Typed pages

Directed by Cheryl D. Seals

Teaching language involves understanding the way people perceive words and utilizing methods of instruction that they can easily comprehend. Computer assisted instruction is of great benefit and more interesting to learners in that it adds a multimedia approach to learning (e.g. audio, video, graphics). Incorporating a multimedia approach to instruction generally requires teachers to learn new skills like programming. This thesis investigates tools that would provide a support system where teachers new to programming can create educational modules without the necessity of extensive programming training. The author has identified a need for tools that are visual and intuitive for users, and that also provide a wide range of capabilities. In this paper, AgentSheets was evaluated as a tool to create language learning simulations for novices.

In addition, the current common approach in language learning software does not embody computer technology and language learning theory in one simulation. PineHill introduced in this paper suggests a new design to create the best conditions for learners to acquire communication skill in a new language with a multimedia approach.

Style manual or journal used: IEEE style guide

Computer Software used: Microsoft word 2003

TABLE OF CONTENTS

LIST OF FIGURES	ix
LIST OF TABLES	x
1 INTRODUCTION	1
2 LITERATURE REVIEW	2
2.1 Educational Psychology	2
2.2 Computer Assisted Learning	3
2.3 AgentSheets	7
2.4 Interface Design for Children	9
3 SIMULATION DESIGN	11
3.1 Interface and Approach of Rosetta Stone	11
3.2 Interface and Approach of PineHill	15
3.3 PineHill vs. Rosetta Stone	20
4 PILOT STUDY	24
4.1 PineHill vs. Conventional Class Material	24
4.1.1 Counting Numbers in Japanese	24
4.1.2 Learning Vocabularies	25
4.2 Usability Testing for AgentSheets	25
4.3 Data from Pilot Study with Students	26
4.4 Data from Pilot Study with Teachers	26
5 EXPERIMENTS	28

5.1	PineHill vs. Rosetta Stone.....	28
5.2	Usability Testing for AgentSheets.....	29
6	DATA ANALYSIS.....	30
6.1	PineHill vs. Rosetta Stone.....	30
6.2	Data from the Usability Testing of AgentSheets.....	32
7	SUGGESTIONS.....	35
8	CONCLUSIONS.....	39
8.1	PineHill as an Aspect of Computer Aided Language Learning Simulation.....	39
8.2	AgentSheets as an End User Programming Tool for Language Class.....	40
9	FUTURE WORK.....	42
9.1	Further Development for PineHill.....	42
9.2	Further Research of AgentSheets.....	42
	REFERENCES.....	43
	APPENDIX A.....	46
	APPENDIX B.....	48

LIST OF FIGURES

Figure 1 Rosetta Stone Main Menu	12
Figure 2 Listening & Reading Preview	13
Figure 3 Listening Preview	13
Figure 4 Reading Preview.....	14
Figure 5 Exercise 1 for Reading	14
Figure 6 Exercise 1 for Listening.....	15
Figure 7 PineHill Study Section in Stage 1	17
Figure 8 PineHill Test Section in Stage 1	17
Figure 9 PineHill Study Section in Stage 2	18
Figure 10 PineHill Test Section in Stage 2.....	19
Figure 11 PineHill Study section in Stage 3	20
Figure 12 Behaviors of an agent in PineHill stage 2	21
Figure 13 Behaviors of agent in PineHill stage 1	21
Figure 14 Behaviors of agent in PineHill stage 2	22

LIST OF TABLES

Table 1 Comparison of Teaching Methods [7].....	5
Table 2 Comparison of PineHill and Rosetta Stone	23
Table 3 PineHill vs. Rosetta Stone (1).....	30
Table 4 PineHill vs. Rosetta Stone(2).....	31
Table 5 The Usability Testing of AgentSheets (1)	32
Table 6 The Usability Testing of AgentSheets (2)	33
Table 7 The Usability Testing of AgentSheets (3)	33
Table 8 The Usability Testing of AgentSheets (4)	34

1 INTRODUCTION

The market of general educational software is slowing down [1], but the market for language learning software is growing so fast that it is at a higher percentage than other titles. Barnes and Nobel indicates “the percentage sales growth for language-related items exceeded that for overall sales in the 2005 year” [2]. However, many language learning simulations have poor interactions and no fun factor for students. Some programs are mere extensions of conventional teaching methods. Thus, this thesis suggests a novel approach for building effective language learning software in this paper.

In addition, there are not many types of software that can be used in language classes with teachers. Usually, those kinds of software are developed for self-learning, and software provided for class has poor interactions and no fun factor for students. The author suggests that one very effective method of instruction is a collaborative learning with teachers and computers both aiding in the learning process. Computers allow different aspects to be represented in colorful as well as meaningful ways that can be very appealing. Computers also allow people to participate interactively in the learning process rather than just being passive receivers of knowledge. However, most teachers are not computer programming experts and need systems to support their creation and management of educational content easily. One such interface is AgentSheets, which allows even novices to build educational simulations, and the experiment of this thesis is based on this application.

2 LITERATURE REVIEW

2.1 Educational Psychology

Theories that are helpful in educational psychology have been identified. Those theories involve increasing teaching efficiency, learning ability, and understanding, all of which are useful for certain educational programs and to identify how to support teachers in building educational programs.

Melchiori demonstrated how to increase learning efficiency with “Recombination Technique”[3]. Dr. Melchiori’s experimental result showed improved reading skills with recombination. For example, after the pronunciation of teaching bolo, vaca, mala, and pato, the children were subsequently able to read the words boca and mapa. Children learned how to read ‘bo’ from bolo and ‘ca’ from vaca and they read ‘boca’ even when they weren’t taught that word directly. This method of teaching is the recombination technique. Thus, it is postulated that children construct new information from old information they already know.

Rose and DeSouza reported that showing written words of objects and pictures could increase teaching efficiency in teaching, reading and spelling [4]. For example, when teachers teach students some vocabularies, they can get better results by showing the students a picture of each object, and asking them to name it and then writing the corresponding word. This theory was applied to this thesis in the following way. When a programmer writes a program for teaching Japanese, teaching efficiency can be boosted

if the program uses sounds and pictures of Hiragana (Japanese Alphabet) and the target objects that students are supposed to learn. As Rose and DeSouza said, using two or more methods as equivalent stimuli would increase students' learning efficiency.

In their work, Ip and Morrison presented pedagogical designs and the concepts of learning resources and learning objects [5]. They also suggested a framework for the utilization of learning resources in different pedagogical paradigms in a large scale collaborative environment. In addition, they showed characteristics required in different learning objects to match the requirements of different pedagogical paradigms. From this work, many methods that would be suitable for this thesis were identified.

2.2 Computer Assisted Learning

The traditional teaching environment is usually that of a classroom: a single teacher giving lectures to a group of students who are expected to use their notes and textbook to prepare for periodic examinations and demonstrate their mastery of the subject [21]. But this environment does not allow a teacher to put into practice many of the surprising facts unearthed by modern educational psychologists. Technology provides an alternative to this scenario. The use of computers to enable and enhance the learning process presents a quantum change in the way people pursue education. The role of teacher has changed, from a teacher to a facilitator of learning process, which takes place inside (and outside) of a student (Table 1). The use of computers in education also shifts the focus away from the teacher to the students themselves who learn through experimentation on the computer with the teacher acting only as a guide. It is, thus, important to understand the basic concepts and theories behind learning with the help of computers.

Computer Assisted Learning (CAL) or Computer Assisted Instruction (CAI) is defined as the use of computers to provide course content instruction in the form of simulations, games, tutorials, and drill and practice. This is a very generic definition and it is necessary to distinguish the types of CAL. There are two ways by which CAL can be classified [6]. The first classification is based on the extent to which the computer assists learning and the second classification deals with the simplicity-complexity level of the CAL. If CAL is classified based on the extent to which the computer is used in the learning process then there are two categories. Adjunct CAL is one where computers are used to supplement knowledge imparted through more traditional means. This is the most common use of computers in education. Primary CAL is one where the computer becomes a substitute to a classroom environment. These stand-alone varieties of CAL are usually of longer duration and are generally less well known and understood in the educational world. Classification based on simplicity-complexity of the CAL yields two divisions as well. Employing an easy-to-learn programming language as well as minimal hardware to support the use of the programs, epitomizes the simplistic approach. However, such simplistic CAL produces limited results; graphics capabilities, large-scale calculations, and the like are not components of such programs. Conversely, complex CAI, which permits extensive use of graphics, large-scale calculations, authoring aids, etc., requires complex author languages (necessitating extensive time for authors to acquire proficiency in use) and large-scale computing capability to support such use.

Paradigm:	Conventional	Simulation Games
Teacher's Role	Agent	Facilitator
Student's Role	Passive	Active
Contents	Predominantly Theoretical	Virtual Reality
Motivation to learn	Contents and its Presentation	Curiosity, desire to solve a problem

Table 1 Comparison of Teaching Methods [7]

Computer Assisted Learning has many advantages over traditional classroom learning. For example: [6]

- It involves any student actively in the learning process. It is difficult for the student to be a totally passive member of the situation, and this very activity and involvement facilitate learning.
- It allows the learner to proceed at his own pace, which has strong implications for both the slow learner and the gifted person.

- Reinforcement of learning in such situations is immediate and systematized, which should result in more effective learning, according to established theories of instruction.
- The use of computers in this manner frees faculty members or training coordinators to devote more time to the personal and human considerations of their students.
- CAL is very useful in the realm of remedial education.
- One disadvantage is the cost of hardware, CAI course materials, and individuals to help implement the process.

Education can be integrated with computing in various ways (e.g., through the use of simulations, games, tutorials, etc). In this paper, the author discusses AgentSheets which is an easy-to-use simulation environment. It is appropriate to look at educational simulations in a little more detail. Any education simulation must [8]:

- Create (or recreate) phenomena, environments, or experiences.
- Provide an opportunity for understanding.
- Be interactive (i.e. the user's inputs must have some effect on the course of the simulation).
- Follow consistent models of a theory.
- Be unpredictable in their behavior, either because of in-built randomness or due to extreme sensitivity to user inputs.

These conditions demand a lot from the person who wishes to design educational simulations. It is, therefore, necessary that the environment in which a designer works be extremely user-friendly, so that the designer who is usually an instructor or a student, will be able to create simulations in a relatively short time and without having to be experts in that environment. If the simulation building tool is not user-friendly then it will be rejected by both the designers as well as the students who will learn from the simulation. It is, therefore, necessary to evaluate the usability of a simulation tool before it is actually employed. Evaluation will be explored in detail in the next section.

2.3 AgentSheets

AgentSheets is an agent-based simulation-authoring tool that allows end-users to build interactive simulations and publish them as Java applets on the web [9]. It is based on a grid structure and so is similar to spreadsheets. The whole work area of the AgentSheets represents the virtual world the user tries to represent. The elements of the grid are called agents [10]. An agent [11] is a thing (or person) empowered to act for a client. The client, in turn, can be another agent or the user of the AgentSheets. Every agent consists of:

- Sensors – Methods of the agent that are either actively triggered by the user or that are used to poll other agent's state.
- Effectors – A mechanism to communicate with other agents by sending messages to agents using relative or absolute grid coordinates.
- State – Describes the condition the agent is in.

- Depiction – The graphical representation of the state, i.e., the look of the agent.
- Instance-of – Link to the class of the agent.

Thus AgentSheets provides an object-oriented paradigm with agents being the objects.

Conditions and actions allow agents to do a variety of operations including computing spreadsheet-like formulas, reacting to mouse clicks and key strokes, playing sampled sounds and MIDI instruments, speaking and gathering information from web pages. These operations are possible with the tool in the AgentSheets called Gallery [10], which allows the incremental composition of depictions as well. The gallery serves the following functions:

- Clone depictions – A new depiction in the gallery is created by cloning an existing one. In the simplest case, cloning involves only copying. However, cloning might include an additional transformation called the cloning operation.
- Re-clone depictions – Modification of a depiction can be propagated to the dependent depictions by re-cloning them.
- Palette – Instantiation of agents. The gallery acts as a palette from which depictions can be chosen and dragged into an AgentSheets.
- Edit depictions – A depiction consists of a bitmap and a name which can be edited with a depiction editor. The depiction editor is just another

AgentSheets in which each agent represents a single pixel of the selected agent's bitmap.

- Save and load depictions – The gallery is a database containing depictions and relations. The depictions can be stored to files and retrieved from files.
- Link depictions to classes – Every depiction is associated with an agent class.

This link is used when instantiating agents

AgentSheets provides object-oriented paradigm, which is a natural and close to real world paradigm. This aspect makes it easy for a novice to build simulation models in AgentSheets. It also provides abstraction at sufficient high level so that the novice programmer need not be concerned with details about the implementation. For example, an agent's depiction can be easily drawn using the gallery which has a wide range of samples and tools for manipulation. These agents can be placed on the AgentSheet in specific grid locations and the programmer has to think only about the triggering events and the way agents would react to them and enter them in the form of “if-then” format in the AgenTalk editor. The AgenTalk editor contains a wide range of triggers (both global and local) and a variety of reactions. AgentSheets provides all the concepts of a programming language such as global variables, local variables, method, and event with simple visual techniques and so is suitable for novice programming.

2.4 Interface Design for Children

Africano et al. [12] discussed the development of a design concept —Ely the Explorer —for an interactive play system and learning tool for children. The traditional computer system is not suited for children. Children learn new knowledge through others,

imitating each other. Children can learn more by working together than by working alone. Africano et al. also mentioned children can acquire knowledge through video and computer games: “Media content plays an important role in the development of children’s cultural practices and play behavior. Young children’s play is often related to media characters and in the form of role-play, which derives from media content conversations as children grow up.” Most collaborative interaction techniques of current educational products are limited by traditional input devices. Africano et al. reported that the traditional PC stations used were not suited for children’s physical and cognitive ergonomic requirements. To solve this problem, Africano presented a design concept—Ely the Explorer. With this, children can work independently or in groups. The children can gather different cultures and geography independently and share the information with other children.

Brouwer-Janse et al. [13] discussed that contrasting traditional and computer play reveals several opportunities to enhance computer interactions. First, it provides an environment that supports social play. Opportunities exist for input devices or linked systems that allow several children to play. Another opportunity is to provide richer sensory and motor interactions. Second, computer play gives the child greater control. Getting started could be easier if computers and computer software were more robust and accessible, physically and procedurally. Rather than simply responding, children could take a more active role by using interactive elements (e.g. blocks, cars, or dolls) that could be manipulated to create events and stories.

3 SIMULATION DESIGN

Most of current computer aided language learning software share common interfaces and methodologies such as showing photos for target words, pronouncing words, and providing simple tests. From research of Valerian Postovsky, speaking fluency is developed most naturally after sufficient listening comprehension has been established [14]. Most of language learning simulations focus on this “listening comprehension first” strategy.

3.1 Interface and Approach of Rosetta Stone

Rosetta Stone is one of the best-selling language learning simulations. Its interface provides five different skill menus such as Listening & Reading, Listening, Reading, Speaking and Writing. Besides the Speaking and Writing menu, Rosetta Stone supplies four sub-menus: Preview, Guided Exercise, Exercise 1 and Exercise 2 (Figure 1).

The Preview menu under the Listening & Reading menu teaches the user the target vocabulary with the pronunciation, the photo and Hiragana (Figure 2). In the preview menu under the Listening menu, users will learn the target vocabulary with the pronunciation and the photo (Figure 3). Through the preview menu under the Reading menu, users will learn Hiragana of the target word with the pronunciation and the photo (Figure 4).

The Guided Exercise menu lets users review what they learned from the Preview section. With Exercise 1 and Exercise 2, users can take a simple test with the vocabularies that they learned (Figure 5 and 6).

Rosetta Stone keeps this interface and methodology from the basic level that teaches Hiragana to the most advanced level that teaches long sentences.

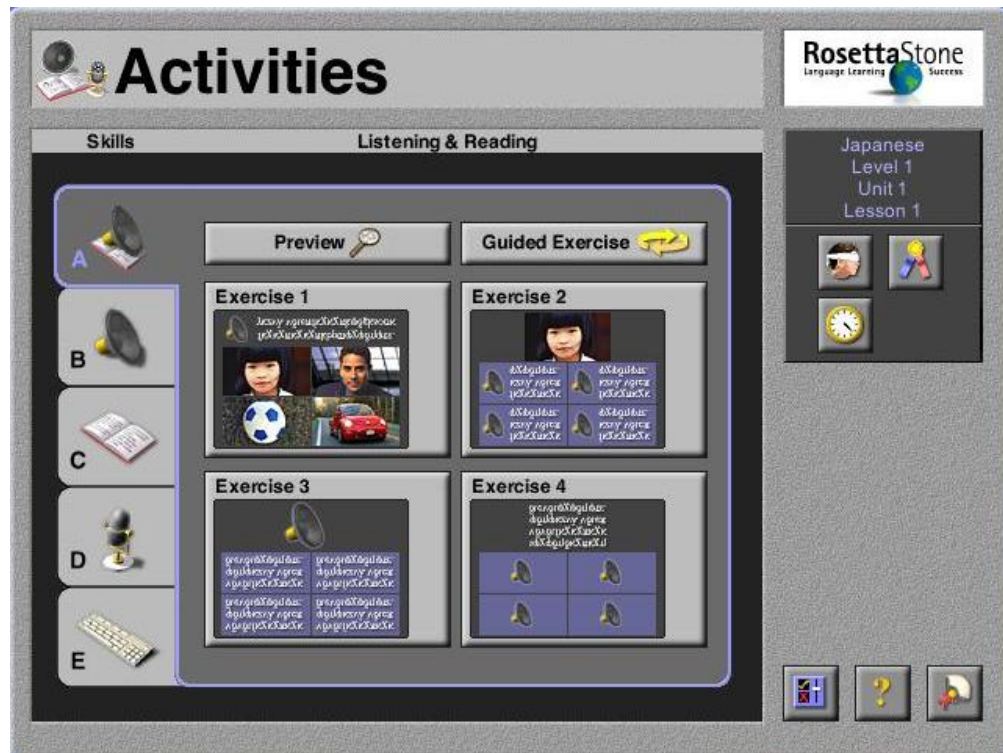


Figure 1 Rosetta Stone Main Menu



Figure 2 Listening & Reading Preview

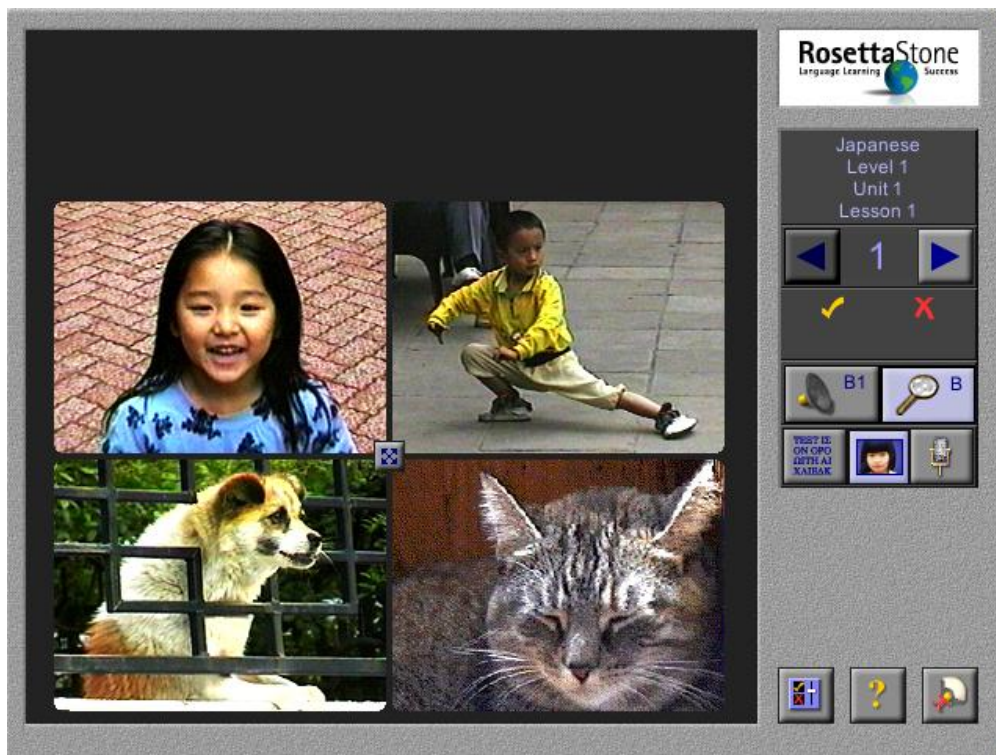


Figure 3 Listening Preview

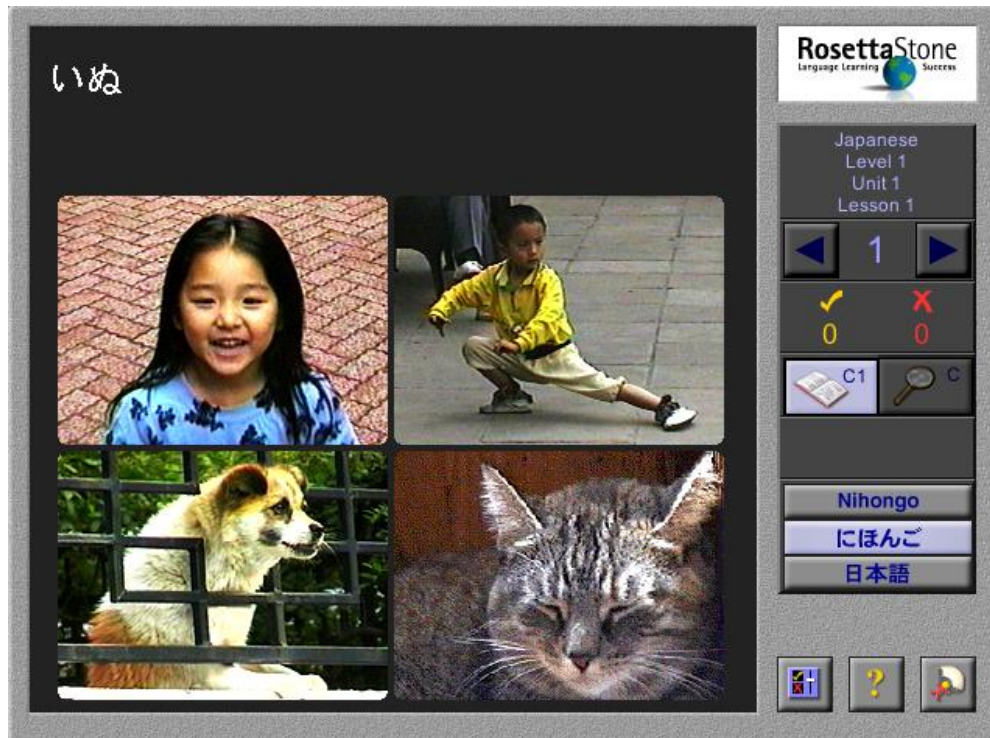


Figure 4 Reading Preview



Figure 5 Exercise 1 for Reading

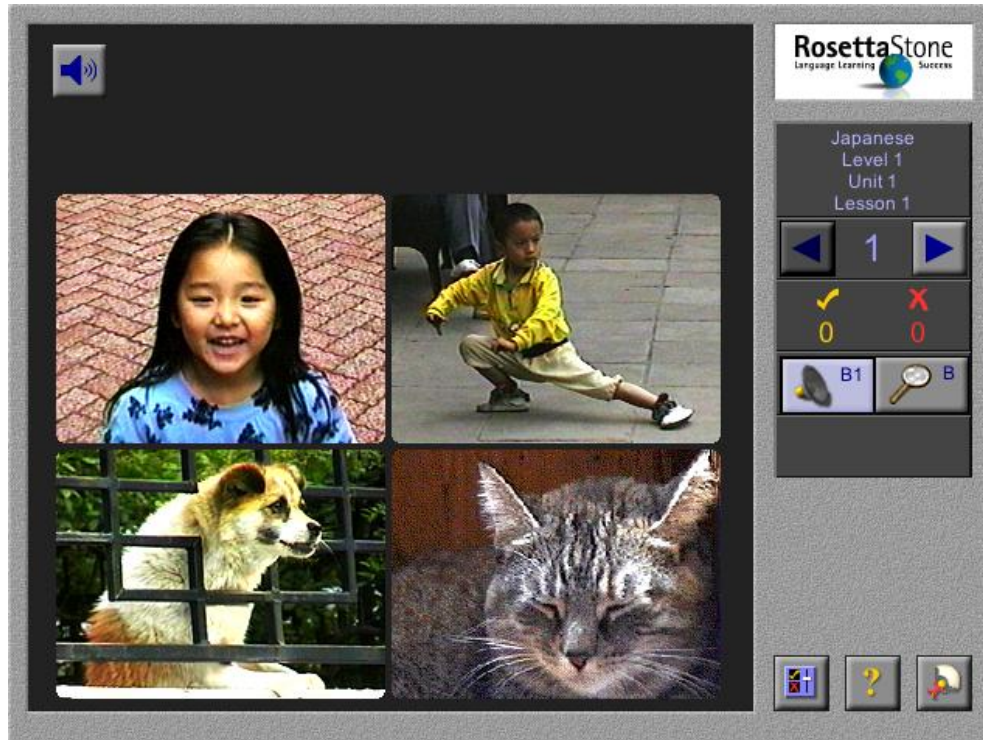


Figure 6 Exercise 1 for Listening

3.2 Interface and Approach of PineHill

The problem of current market selling language learning software is that they have only one interface and methodology for beginners and advanced learners. Additionally, these programs try to teach reading, listening, and speaking at the same time even though most beginners can be traumatized by multiple directions and processes [15]. Listening first strategy might be a good approach for speaking, but it does not seem enough for syllabic writing and reading learning.

In an attempt to solve these problems, the author has designed a multimedia simulation called PineHill to teach the Japanese language. PineHill is composed of three different stages. Users will be guided through the fundamental stages of Japanese

language acquisition, beginning with Hiragana, the Japanese alphabet, and advancing to basic grammar.

Also, the development of this simulation focused on the learnability of software. PineHill is able to be used with minimal help, and students can use this simulation independently after the first try [15].

In the first stage, the user will learn the target vocabularies with photos and pronunciation in the study section (Figure 7). It adopts the same methodology as other software, “listening comprehension first”. Also, there is a test section for target words. The test section has three questions and each question is made of three icons (Figure 8). The top icon works as a question; it pronounces a certain word. The second icon is a choice that contains multiple photos. Users select a matched photo corresponding to the sound when they clicked the top icon. The third one is an answer that lets users know if they are right or not. If their answer is correct, the bulb will be lit and will sound a positive earcon “Good”. Users proceed to the second stage after they finish the study section and the test section. Additionally, the author categorized all target words such as a word list for kitchen, study room, bathroom and so on. Word categorization is helpful to build users’ word learning.

PineHill1

1. Scroll down and click [Run] to start PineHill. 2. Click [Speaker] icon to play sounds. 3. Click [Arrow] icon to forward or backward [More Information](#)

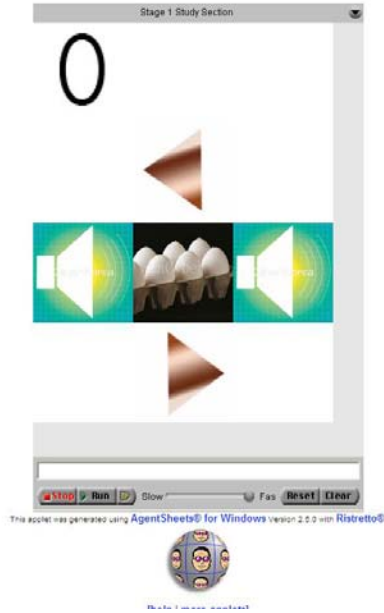


Figure 7 PineHill Study Section in Stage 1

PineHill1 Test

1. Scroll down and click [Run] to start PineHill. 2. Click [?] icon on the top to play sounds. 3. Click [Picture] icon to match the sound and the picture. 4. Click [bulb] icon to check your answer. [More Information](#)

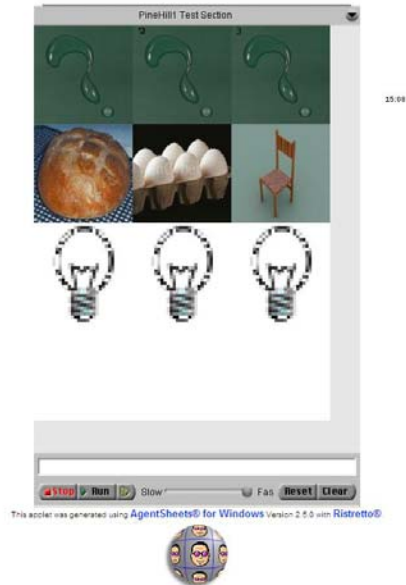


Figure 8 PineHill Test Section in Stage 1

In the second stage, the user will find a similar interface to the first stage, but there is Hiragana (Figure 9). During this stage, users will learn Hiragana with words they have learned through the first stage. Each Hiragana icon that is clicked sounds the pronunciation of itself. Thus, users can learn matched pronunciation for each Hiragana [17]. The test section has the same interface as the test section in the first stage. The top icon works as a question as it pronounces a certain Hiragana. The second icon is a choice which contains multiple Hiraganas. Users select matched Hiragana corresponding to the pronunciation when they clicked the top icon (Figure 10). This interface gives users low anxiety environments that inspire self-confidence in the learner by supporting interface consistency and the word list they have learned in the first stage [18].



Figure 9 PineHill Study Section in Stage 2

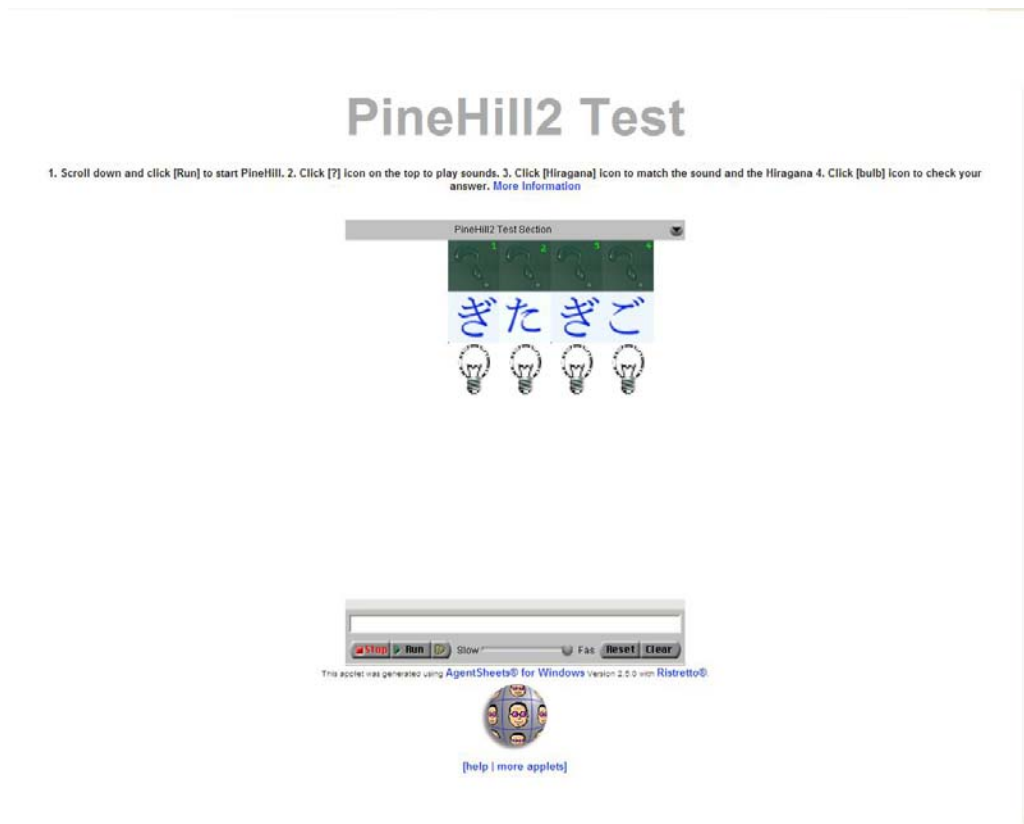


Figure 10 PineHill Test Section in Stage 2

After the first and second stages, users move on to the final stage. In the third stage, users will learn basic grammar with sample sentences that are the combination of photos and Hiragana. These sample sentences contain vocabularies users learned in the first and second stages and new vocabularies they will learn on the third stage. Like the second stage, users will learn grammar with a sound function (Figure 11). This environment supports no translation to let users learn Japanese grammar more effectively than in the conventional way because a language taught by the grammar-translation methods lacks a direct link to a network of meaning [19].



Figure 11 PineHill Study section in Stage 3

3.3 PineHill vs. Rosetta Stone

Rosetta Stone and PineHill share some common methodologies. Both of them are based on the listening comprehension first strategy. The same multimedia materials such as photos, sounds and letters are used for supporting the listening comprehension first strategy. The study session and test session are adopted for both simulations.

However, there are some unique features of PineHill. The interface of PineHill progresses with its stage process. Unlike Rosetta Stone, PineHill adopts phonics theory for syllabic writing and reading learning [20]. Thus, users will learn the relationship between each Hiragana and its sound. Also, PineHill categorizes the word list while Rosetta Stone randomly assigned its words (Table 2).

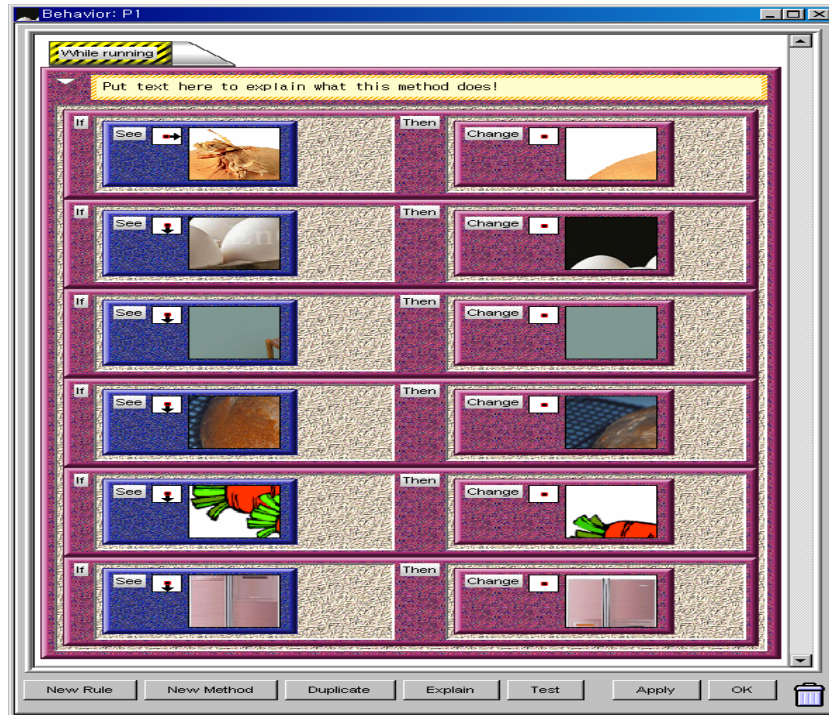


Figure 12 Behaviors of an agent in PineHill stage 2

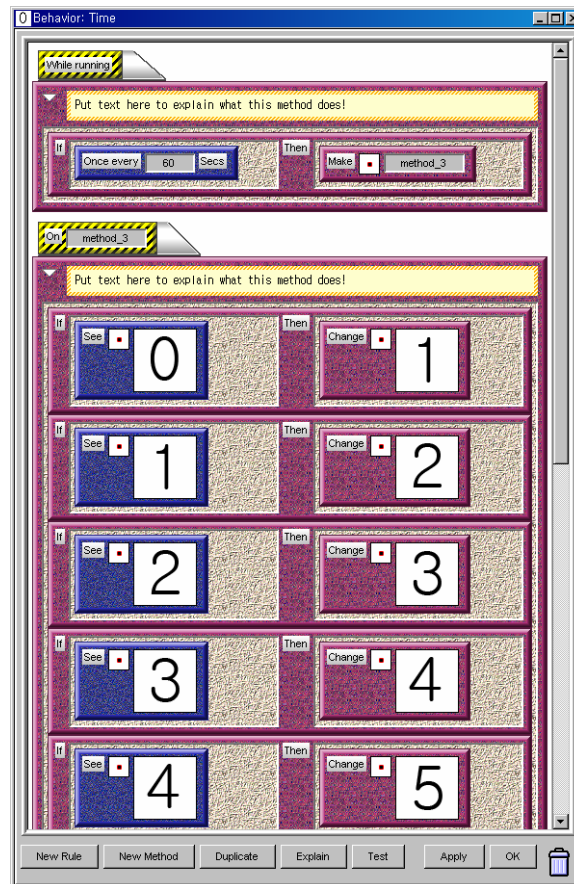


Figure 13 Behaviors of agent in PineHill stage 1

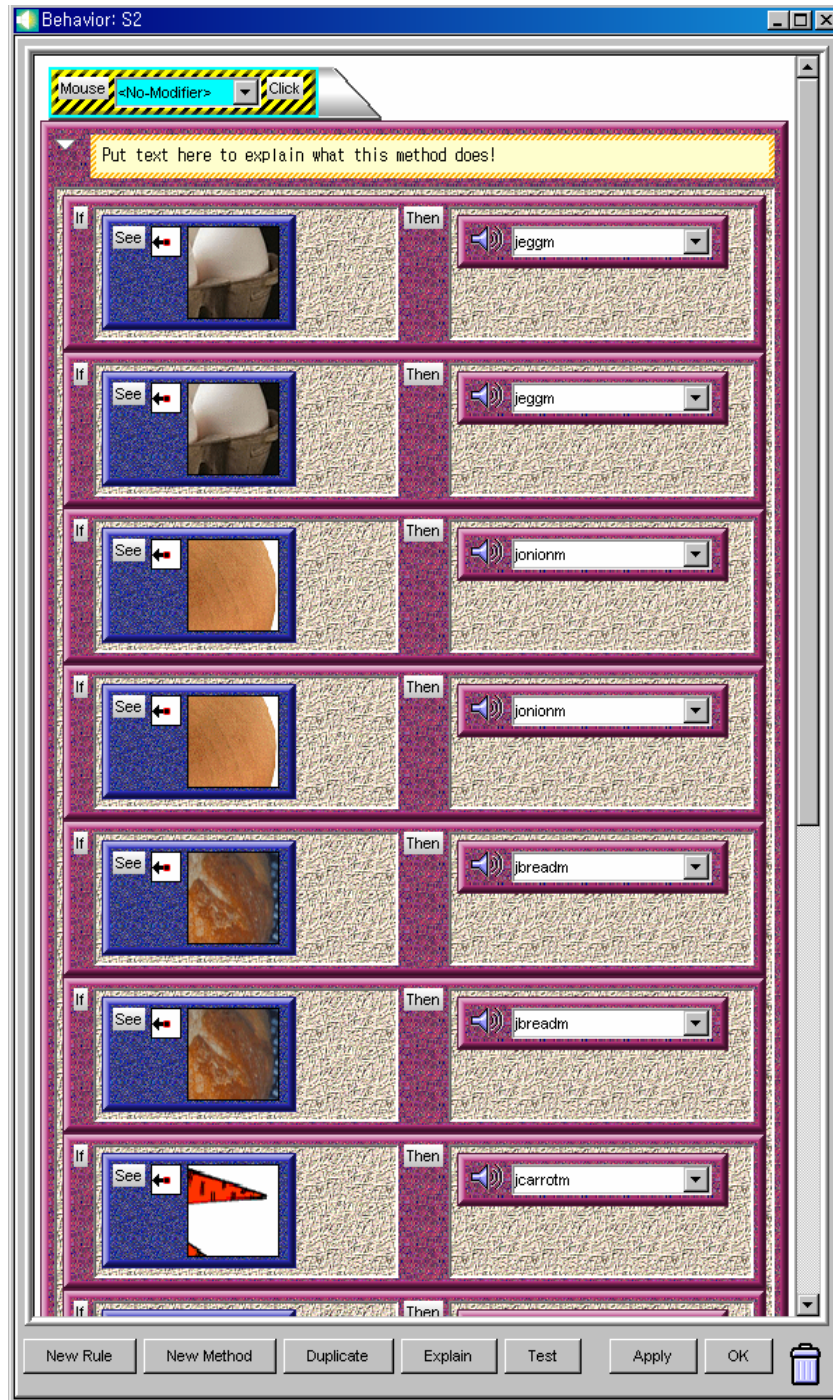


Figure 14 Behaviors of agent in PineHill stage 2

	Rosetta Stone	PineHill
Word List	Randomly assigned	Categorized
Interface	Remains uniform	Progresses by stage
Methodology	Listening comprehension first	Listening comprehension first Phonics
Voice	Female voice	Male voice Female voice
Syllabic Sounds Function	No	Yes

Table 2 Comparison of PineHill and Rosetta Stone

4 PILOT STUDY

Two independent pilot studies were conducted. The first pilot study compares conventional class materials and PineHill. The second one is the usability test of AgentSheets for Japanese language teachers.

4.1 PineHill vs. Conventional Class Material

Twenty people participated for this study. People who had never been instructed in Japanese before were recruited to prevent participants' experience from interrupting the result of this study. Two sets of experiments were conducted. First, participants were separated into two groups (P, H) of ten people. Each group was almost equally distributed by average ages, levels of education, and diversity of nationalities. The P group had two Americans, three Koreans, two Indians, and three Chinese participants. The H group had two Americans, three Koreans, three Indians, and two Chinese participants. There were four college students and six graduate school students in the P group and five college students and five graduate school students in the H group.

4.1.1 Counting Numbers in Japanese

The first experiment dealt with teaching how to count from one to ten in Japanese. The P group was assigned to learn numbers from one to ten with PineHill as an individualized tutorial. H group was allowed to learn numbers with teachers and conventional class material like papers and a whiteboard.

Three minutes were given to both groups to learn ten numbers. After three minutes, participants took an identified test. There was a total of five multiple choice questions in the test. The only difference between the P and H groups was that the P group was administered their test with PineHill and the H group was administered their test in their traditional method of papers and pens.

4.1.2 Learning Vocabularies

The second study was about learning vocabularies of ten objects in the kitchen. The instructional method of the P and H group was changed. In this experiment, the P group worked with teachers, and the H group worked with PineHill.

As in the previous study, every participant was given three minutes for learning 10 vocabulary words. After three minutes, they were tested in the same manner as they had been before.

4.2 Usability Testing for AgentSheets

This testing was conducted to validate the usability of AgentSheets for creating a language learning simulation with novice programmers. The author performed this usability testing with ten Japanese teachers of the Japanese Student Organization at Auburn University. None of the participants had prior experience with programming.

Participants were asked to create a simulation, which can act like PineHill with a given tutorial. The simulation was about learning vocabularies of fruits. The tutorial includes all instructional information about AgentSheets. Some of the tasks that participants were instructed to do the following: how to add agents, how to change depictions and so on. In addition, the participants were provided sound and image files

needed for this study. The participants were allowed to manage their time because the rationale for this study was not how fast they learn AgentSheets but the usability of AgentSheets (i.e. how easy they can use AgentSheets).

4.3 Data from Pilot Study with Students

It was found that PineHill actually helped to increase the learnability of students. Also, it made students active and interested by giving more interactions and increased the fun factor more than conventional class material.

In the first study, eight students in the P group using PineHill recorded sixty percent accuracy, and five students in the H group recorded sixty percent accuracy. There was also an interesting result in the second study. In the second experiment, only four students in the P group recorded sixty percent accuracy while six students in the H group recorded sixty percent accuracy with PineHill.

Eighteen of the participants agreed PineHill is more interesting than paper based material, and sixteen of them reported that PineHill provides sufficient and interesting interactions.

4.4 Data from Pilot Study with Teachers

Every participant finished his or her given task within 30 minutes. Their user satisfaction with AgentSheets was high. Eight participants admitted creating a simulation like PineHill was easy, and seven of them agreed that using PineHill, which is created with AgentSheets, was more effective than traditional teaching with papers and tapes.

Six of the ten teachers reported they wanted to use AgentSheets for their class. The author asked four teachers the reason why they did not want to use AgentSheets for their

class. Two teachers said that they believed teaching by person is more educational than computer supported teaching. One of them thought that he could teach quicker and more effectively with paper based material than AgentSheets. The last teacher stated that he felt comfortable with conventional methods of teaching.

5 EXPERIMENTS

Two independent experiments were performed. The first experiment evaluated the usability and learnability of PineHill and Rosetta Stone. The second one is the usability test of AgentSheets for building a language learning simulation.

5.1 PineHill vs. Rosetta Stone

Forty-five people who were recruited for this experiment were randomly assigned to PineHill and Rosetta Stone. The group of participants who were assigned to PineHill was named the P group, and the group of participants who were assigned to Rosetta Stone was named the R group. A total of twenty-two participants for the P group and twenty-three participants for the R group took part in this test.

The P group was asked to take the test section after they took the study section in each stage. The R group was asked to take Exercise 1 after the Preview in the reading menu & Listening menu. Each group had up to ten minutes for the study section and the preview section, and five minutes were assigned to the test section and the exercise.

After participants finished their assigned tasks, the survey that assessed the learnability and the usability of each simulation was given to them.

5.2 Usability Testing for AgentSheets

Unlike the pilot study, twenty people including experienced programmers and novice programmers were recruited for the usability testing of AgentSheets. A total of ten experienced programmers and ten novice programmers participated in this test.

Participants were asked to create a simulation just like PineHill stage 1. The tutorial that shows the necessary steps to create the simulation and resources such as the image files and the sound files were provided like in the pilot study.

Thirty minutes were assigned to complete this usability test. After participants' usability testing, they were asked to complete the questionnaire on the usability of AgentSheets as a language learning simulation building tool.

6 DATA ANALYSIS

6.1 PineHill vs. Rosetta Stone

From the usability and learnability test of PineHill and Rosetta Stone, interesting results were derived. Rosetta Stone was evaluated as more wonderful and more satisfying than PineHill. However, PineHill was graded higher than Rosetta Stone in easiness, flexibility and fun (Table 3). The fact that Rosetta Stone provides fancy and complex interfaces while PineHill gives simple and direct interfaces may account for those results.

	Average of PineHill (SD)	Average of Rosetta Stone (SD)	P-Value
Terrible ----- Wonderful	4.05 (0.67)	4.22 (0.60)	0.19
Frustrating ----- Satisfying	3.95 (0.84)	4.00 (0.74)	0.42
Dull ----- Stimulating	3.95 (0.79)	4.04 (0.77)	0.35
Difficult ----- Easy	4.05 (0.95)	3.78 (0.95)	0.17
Rigid ----- Flexible	3.55 (0.86)	3.26 (0.75)	0.12
Boring ----- Fun	4.05 (0.89)	3.96 (0.98)	0.37
Strongly agree = 5, Agree = 4, Neutral = 3, Disagree = 2, Strongly disagree = 1			

Table 3 PineHill vs. Rosetta Stone (1)

From question five which asked if simulation was easy to learn and use, eighty-two percent of the P group agreed whereas sixty-five percent of the R group agreed.

Additionally, only nine percent of the P group checked that PineHill was difficult to remember where some of the tools and commands were located while thirty-five percent of the R group did. Also, eighty-six percent of the P group answered that they had a good understanding of how to use the simulation albeit sixty-five percent of the R group did. A more interesting result was from question eight. Even though Rosetta Stone rated higher than PineHill in “Terrible—Wonderful” and “Frustrating—Satisfying”, PineHill was evaluated higher than Rosetta Stone in question eight which asked if the system was fun for learning Japanese.

As these data show, it is believed that PineHill supports simpler and easier interfaces for users than Rosetta Stone, and the interface affects the fun factor in the learning process.

	Average of PineHill (SD)	Average of Rosetta Stone (SD)	P-Value
5. This simulation, PineHill, was easy for me to learn and use. *+	4.00 (0.76)	3.70 (0.82)	0.1
6. It was easy to get started. *+	4.00 (0.76)	3.57 (1.16)	0.07
7. It was difficult to remember where some of the tools and commands were located. *	2.14 (0.94)	3.04 (1.07)	0.00
8. This system would be fun for learning Japanese. *+	3.95 (0.79)	3.60 (0.99)	0.1
9. I have a good understanding of how to use this simulation, PineHill, to learn Japanese. *	4.18 (0.80)	3.67 (1.15)	0.05
Strongly agree = 5, Agree = 4, Neutral = 3, Disagree = 2, Strongly disagree = 1			
* indicates significance p<.05, *+ indicates p<.10 approaching significance			

Table 4 PineHill vs. Rosetta Stone(2)

6.2 Data from the Usability Testing of AgentSheets

The questionnaire used for the usability testing of AgentSheets contained twenty-five questions that focused on the usability and learnability.

Even though there were ten participants who did not have any programming experience, the result showed that most participants had a good understanding of how to use AgentSheets for creating an educational simulation.

	Average General User Satisfaction(SD)
Terrible ----- Wonderful	4.15 (0.49)
Frustrating ----- Satisfying	4.00 (0.65)
Dull ----- Stimulating	4.15 (0.81)
Rigid ----- Flexible	3.55 (0.51)
Boring ----- Fun	4.30 (0.66)
Strongly agree = 5, Agree = 4, Neutral = 3, Disagree = 2, Strongly disagree = 1	

Table 5 The Usability Testing of AgentSheets (1)

Ninety percent of participants answered that AgentSheets was fun and satisfying to use. More than eighty percent of participants agreed that AgentSheets was easy to learn and use, and AgentSheets would be fun for building educational games. Also, seventy-five percent of participants agreed that AgentSheets would be easy to use by people who don't know much about computers.

As these results illustrate, it is believed that AgentSheets is a useful tool for creating language-learning simulations for novice programmers.

Likert-Type Scale Item	Average of General Ease of Use (SD)
The system is easy to use	4.10 (0.64)
The directions were hard to follow.	2.35 (0.93)
It was easy to get started.	4.05 (0.94)
I understand the system well.	4.15 (0.67)
This system was easy for me to learn and use.	4.10 (0.85)
I have a good understanding of how to use this system to build educational programs.	4.30 (0.92)
This system would be easy to use by folks who don't know much about computers.	4.20 (0.95)
Strongly agree = 5, Agree = 4, Neutral = 3, Disagree = 2, Strongly disagree = 1	

Table 6 The Usability Testing of AgentSheets (2)

Likert-Type Scale Item	Average of Assessing Motivation (SD)
The system was boring.	2.20 (0.70)
I would like AgentSheet to be used in my classes.	4.15 (0.93)
This system would be fun for building educational games.	4.30 (0.86)
Strongly agree = 5, Agree = 4, Neutral = 3, Disagree = 2, Strongly disagree = 1	

Table 7 The Usability Testing of AgentSheets (3)

Likert-Type Scale Item	Average of Programming Style Reactions (SD)
It was difficult to remember where some of the tools and commands were located.	2.00 (0.79)
Rule ordering was confusing.	2.35 (0.93)
Creating visual rules by dragging and dropping the desired parts to create behavior was complicated.	1.85 (0.99)
The rules I created for agents' behaviors were simple and natural.	4.15 (0.75)
I found the creation of rules for agent behaviors confusing.	1.80 (0.95)
Strongly agree = 5, Agree = 4, Neutral = 3, Disagree = 2, Strongly disagree = 1	

Table 8 The Usability Testing of AgentSheets (4)

7 SUGGESTIONS

During the course of this experiment and usability study, many areas in which AgentSheets can be improved were identified. It is believed that these changes will help make AgentSheets more usable and enhance the experience of using this software.

First, the one restrictive issue was that an agent cannot be depicted as a text. The text in agents had to be created as images. This proved to be very time consuming and laborious. This constraint will deter the use of AgentSheets in creating text-heavy simulations because there may be many scenarios where it would be more accurate to depict agents as a text rather than images.

Second, agents in the worksheet should be flexible. AgentSheets can be enhanced by supporting for the variable agent sizes. There were many situations where depicting agents of the same size may rob the simulation of its closeness to reality. The agent size is so strict. An agent in the gallery is the same size of an agent in worksheets, making the size of the gallery unnecessarily bigger and longer, and making users uncomfortable with manipulating the behavior of agents. For example, if users want to represent an elephant and an ant, using the same agent sizes will adversely affect the feel of the simulation as the agents involved seem to be out of proportion. This may be solved by merging adjacent grid cells in order to create larger agent sizes. Actually, advanced users can change the shape of an agent with AgentTalk, but the above suggestion is concerning more variations of shape not size.

Third, problems were also encountered while using the interface. These were trivial problems but had a big impact on the understanding and speed of the development of the simulation. The action “Change” actually changes only the depiction of the neighboring grid but not the object itself. To change the object, one had to use “New” and “Erase”. This is not evident and so was discovered after experimentation. Such ambiguities could be removed by using appropriate title for the actions. Also, many commonly used keyboard shortcuts are not supported by AgentSheets. Multiple actions cannot be replicated simultaneously. Thus, copying actions one-by-one tends to be very tedious. Another important point is that it is not possible to rename agents once they are created.

Fourth, there should be a function to import sound files. AgentSheets provides sound functions, but it can support only audio files in a predefined folder. If users want to use their own sound files, they should copy or move files to a designated folder (under the resource folder of the folder AgentSheets installed). It is not an easy step for novice computer users. Also, it would be good if there were a recording function. Participants, Japanese language teachers, wanted to record their own voice and use them, but AgentSheets does not support that option. Therefore, they had to record their voice with recording software and copy those files to the designated folder. The recording function can be used not only for language class but also for general K-12 classes.

Fifth, AgentSheets cannot convert the project that contains more than two worksheets to the Java applet correctly. When trying to convert simulations including two or three worksheets, all worksheets were converted to one Java applet. Because of that, it was necessary to break down a simulation into several simulations including a single worksheet. To allow users to convert the simulation to the Java applet is a huge attraction

of AgentSheets. It can be used for on-line learning by being posted to the web, and can save time in developing other on-line teaching materials. To support this point, AgentSheets should have a function which can convert multiple worksheets to multiple Java applets.

Sixth, another problem exists when AgentSheets converts a project to the Java applet. Even though users create wave files under the resource folder to use their own sound files for their simulations, they cannot use those files after the simulation is converted to the Java applet. The Java applet, which is converted from AgentSheets simulation, does not include any users' sound files.

AgentSheets has two sound folders. One is a WAVE folder that contains wave files, and the other is an AU folder that contains au files. Wave files in the WAVE folder are used for AgentSheets simulations. When users use sound functions during the building of their simulations with AgentSheets, they use these wave files. Au files in the AU folder are used for Ristretto that turns AgentSheets simulations into complete web pages with Java applets. When users launch Ristretto to convert their projects into Java applets, au files under the AU folder are moved to the Java applet folder.

This situation means that au files only under the AU folder can be used for the Java applet. Thus, users should have wave files and au files that have the same names under the AU folder and the WAVE folder in order to use them for AgentSheets simulations and the Java applets. Also, the name of au file under the Java applet folder should be written in lower case to be used for on-line teaching. The capital in the file name is not a problem with AgentSheets simulation or the Java applet, but the Java applet on the web

cannot recognize the au files with the capital. The fact that the Java applet posted on the web cannot play sound files even if it works perfectly off line causes a serious problem.

These steps are very complicated for novice programmers and even experienced programmers. To prevent this confusing error, AgentSheets needs to have a function that can convert wave files under the resource folder to au files under the Java applet folder when Ristretto is launched to convert AgentSheets simulations to the Java applets.

8 CONCLUSIONS

8.1 PineHill as an Aspect of Computer Aided Language Learning Simulation

The results of the pilot testing and formal experiment illustrate that PineHill is an effective method for foreign language training. This type of language learning software intrigues students' curiosity and provides them with a foundation for interactive self-paced language learning.

By merging interactive technology with listening comprehension and problem solving and language acquisition theory, PineHill enables learners of any age, ability or language background to acquire the confidence to master foreign languages.

Also, teachers can post the Java applet of their simulation to a web site for class preview and review. This means students finish their study for class with a selection and mouse clicking only. There is no more need to play a tape or CD-Rom. It would be especially helpful for children who spend most of their days with computers.

Most of all, teachers can keep the class in their control unlike other educational software. Usually, teachers cannot lead the class when they use current market selling educational software; instead the software leads. However, with AgentSheets, teachers can create, recreate and manipulate the content of simulation whenever they want, and it takes less time than paper based class material.

8.2 AgentSheets as an End User Programming Tool for Language Class

The results of the experiments illustrate that AgentSheets is good software for creating a language learning simulation. Most novice programmers found it easy to understand the rule-based programming methodology employed by AgentSheets and were able to easily navigate the various toolbars, menus and options in order to create simple simulations. Another advantage of AgentSheets is that all the rules are presented in simple If-Then statements, which makes it easy for relatively inexperienced users to program using the system. AgentSheets allows the use of colorful images, sounds and visual effects to depict the agents, and this is an ideal application for creating educational simulations. It was found that users attained a greater level of understanding and better appreciated the simulation when feedback was given using sound, text and visual means. The fact that AgentSheets supports both audio and video helps in making highly interactive simulations, which are ideal for children. Another feature that is very useful is that AgentSheets allows a simulation to be converted into a Java applet. This means that simulations can be posted on web sites, thus being made available to a very large audience.

Interestingly, anecdotal evidence was found that some experienced programmers actually had trouble creating simulations using AgentSheets. They seem to become confused by some of the rules and frustrated when their expectations are not met. This may have something to do with the fact that experienced programmers are used to the level of control provided by traditional programming languages such as C++, Java, Visual Basic etc., and that they find it difficult to adapt to the more restrictive environment of AgentSheets. This high level of regulation, however, seems to help a

novice programmer, as it provides them with a framework within which they can work to create simulations by following relatively simple rules. Thus, AgentSheets is a very useful tool that allows even inexperienced users to create interactive simulations for educational purposes.

9 FUTURE WORK

9.1 Further Development for PineHill

In this thesis, only the basic level of Japanese learning was covered. More work is needed in advanced levels such as reading and writing nested sentences, reading Kanji, and speaking in formal ways and casual ways. These levels of learning will be possible if AgentSheets is revised and updated with suggested features.

Currently, PineHill is limited to teaching only Japanese but can be modified and transposed into any other language. If phonics theory is adopted, it seems possible to teach the basic level of other foreign languages.

9.2 Further Research of AgentSheets

During the experiments, AgentSheets supports enough functions to create PineHill for Japanese teaching even though there are some features which should be revised or added. However, if AgentSheets is used for creating other foreign languages, it might need other functions or features.

REFERENCES

- [1] M. Richtel, "Once a Booming Market, Educational Software for the PC Takes a Nose Dive", *The NY Times* November 30, 2006
- [2] J. Levere, "SPENDING: As Many Software Choices as Languages to Learn", *The NY Times*, November 26, 2006
- [3] L. E. Melchiori, D. G. de Souza, & J. C. de Rose, "Reading, equivalence, and recombination of units: A replication with students with different learning histories", *Journal of Applied Behavior Analysis*, 33, 97-100, 2000.
- [4] J. C. De Rose, D. G. De Souza, and E. S. Hanna, "Teaching reading and spelling: Exclusion and stimulus equivalence", *Journal of Applied Behavior Analysis*. 29, 451-469, 1996.
- [5] A. Ip, I. Morrison, "Learning objects in different pedagogical paradigms", *In Proceedings of ASCILITE 2001 conference*. 289 -298, 2001.
- [6] J.A. Chambers, J.W. Sprecher, "Computer Assisted Instruction: Current Trends and Critical Issues", *Communication of the ACM*, 332-333, 1980.
- [7] L. Chwif, M.R.P. Barretto, "Simulation Models as an Aid for the Teaching and Learning Process in Operations Management", *In Proceedings of the 2003 Winter Simulation Conference*, 1994-1995, 2003.

- [8] A.L. Aitkin, "Playing at Reality", The Australian National University. 189 -191, 2004.
- [9] A. Repenning, "AgentSheets®: an Interactive Simulation Environment with End-User Programmable Agents", *In Proceedings of the IFIP Conference on Human Computer Interaction (INTERACT '2000, Tokyo, Japan)*, 2000.
- [10] A. Repenning, "Creating User Interfaces with AgentSheets", *In Proceedings of the 1991 IEEE Symposium on Applied Computing*, Kansas City, Missouri, 191-196, 1991.
- [11] M.R. Genesereth, and N.J. Nilson, "Logical Foundations of Artificial Intelligence", Los Altos: Morgan Kaufman Publishers, Inc, 1987.
- [12] D. Africano, S. Berg, K. Lindbergh, P. Lundholm, F. Nilbrink, A. Persson, "Designing Tangible Interfaces for children's Collaboration", *CHI '04 extended abstracts on Human factors in computing systems*, 853-868, 2004.
- [13] M. D. Brouwer-Janse, J. F. Suri, M. Yawitz, G. de Vries, J.L. Fozard, and R. Coleman, "User interfaces for young and old", *Interactions(March-April 1997)*, 3446, 1997.
- [14] V. Postovsky, "The priority of aural comprehension in the language acquisition process", *The comprehension approach to foreign language teaching* (pp. 170-187), Newbury House Publishers, 1981.
- [15] J. J. Asher, "Fear of Foreign Languages", *Psychology Today*, 52-59, 1981.
- [16] S. Khalifa, C. Bloor, W. Middleton, C. Jones, "Educational computer software, technical, criteria, and Quality", *Proceedings of the Information Systems Education Conference*, 2000.

- [17] J.F. Lee, B. VanPatten, "Making communicative language teaching happen", McGraw Hill. 1995.
- [18] S. D. Krashen, T. D. Terrell, "The natural approach", Pergamon, 1993
- [19] K. C. Diller, "Neurolinguistic clues to the essentials of a good language teaching methodology: comprehension, problem solving and meaningful practice", *The comprehension approach to foreign language teaching* (pp. 141-153), Newbury House Publishers, 1981.
- [20] M. J. Adams, "Beginning to Read: Thinking and Learning About Print", Cambridge, Mass. MIT Press, 1990
- [21] P.M. Regan, B.M. Slator, "Case-based Tutoring in Virtual Education Environments", In Proc CVE 2002, ACM Press, 1-3, 2002.

APPENDIX A

PineHill / Rosetta Stone Questionnaire

1. Please respond by circling the reaction that best reflects your reaction to the system:

Terrible ----- Wonderful

1 2 3 4 5

Frustrating ----- Satisfying

1 2 3 4 5

Dull ----- Stimulating

1 2 3 4 5

Difficult ----- Easy

1 2 3 4 5

Rigid ----- Flexible

1 2 3 4 5

Boring ----- Fun

1 2 3 4 5

2. How many years have you used a computer?

- a. Less than one year
- b. 1 - 3 years
- c. 4 - 6 years
- d. 7 - 10 years
- e. More than 10 years

3. On average how many times a week do you use a computer?

- a. 1 - 2 times
- b. 3 - 5 times
- c. 6 - 10 times
- d. 11 – 15 times
- e. More than 15 times

4. On average how many hours do you spend on your computer per week?

- a. 1 - 2 hours
- b. 3 - 5 hours
- c. 6 - 10 hours
- d. 11 – 15 hours
- e. More than 15 hours

5. This system was easy for me to learn and use.

1	2	3	4	5
Strongly Agree	Agree	Neutral	Disagree	Strongly disagree

6. It was easy to get started.

1	2	3	4	5
Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

7. It was difficult to remember where some of the tools and commands were located.

1	2	3	4	5
Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

8. This system would be fun for learning Japanese.

1	2	3	4	5
Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

9. I have a good understanding of how to use this system to learn Japanese.

1	2	3	4	5
Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

10. What was your score in the test section?

11. What 1-2 things would you change if you were asked to revise the system?

APPENDIX B

AgentSheet

The application usability test questionnaire

1. **Do you have any programming experience?**
 Yes No
2. **How many years have you used a computer?**
 0 1~10 11~20 21~30 more than 30
3. **On average how many times a week do you use a computer?**
 0~1 2~3 4~5 6 or more than 6
4. **On average how many hours do you spend on your computer per week?**
 0~5 6~10 11~15 15~20 more than 20
5. **Have you heard about visual programming?**
 Yes No
6. **The system is easy to use.**
 strongly agree agree neutral disagree strongly disagree
7. **The directions were hard to follow.**
 strongly agree agree neutral disagree strongly disagree
8. **The system was boring.**
 strongly agree agree neutral disagree strongly disagree
9. **I understand the system well.**
 strongly agree agree neutral disagree strongly disagree
10. **Please respond by circling the reaction that best reflects your reaction to the system:**
Terrible Wonderful
1 2 3 4 5
11. **Please respond by circling the reaction that best reflects your reaction to the**

system:

Frustrating Satisfying

1 2 3 4 5

12. Please respond by circling the reaction that best reflects your reaction to the system:

Dull Stimulating

1 2 3 4 5

13. Please respond by circling the reaction that best reflects your reaction to the system:

Rigid Flexible

1 2 3 4 5

14. Please respond by circling the reaction that best reflects your reaction to the system:

Boring Fun

1 2 3 4 5

15. This system was easy for me to learn and use.

strongly agree agree neutral disagree strongly disagree

16. It was easy to get started.

strongly agree agree neutral disagree strongly disagree

17. It was difficult to remember where some of the tools and commands were located.

strongly agree agree neutral disagree strongly disagree

18. This system would be easy to use by folks who don't know much about computers.

strongly agree agree neutral disagree strongly disagree

19. Rule ordering was confusing.

strongly agree agree neutral disagree strongly disagree

20. Creating visual rules by dragging and dropping the desired parts to create behavior was complicated.

strongly agree agree neutral disagree strongly disagree

21. The rules I created for agents' behaviors were simple and natural.

strongly agree agree neutral disagree strongly disagree

22. I found the creation of rules for agent behaviors confusing.

strongly agree agree neutral disagree strongly disagree

23. This system would be fun for building educational games.

strongly agree agree neutral disagree strongly disagree

24. I have a good understanding of how to use this system to build educational programs.

strongly agree agree neutral disagree strongly disagree

25. I would like AgentSheet to be used in my classes.

strongly agree agree neutral disagree strongly disagree