**Free Apples: An Intelligent Tutoring System Integrating Kolb’s Experiential Learning Model as Pedagogical Module embedded with Facial Expression Recognition**

Borais, Alexavier B.

B.S Computer Science

University of Santo Tomas

Espana, Manila, Philippines

Guazon, John Robert. B.

B.S Computer Science

University of Santo Tomas

Espana, Manila, Philippines

Francisco, Yves Rupert C.

B.S Computer Science

University of Santo Tomas

Espana, Manila, Philippines

Mabasa, Ryan Christopher G.

B.S Computer Science

University of Santo Tomas

Espana, Manila, Philippines

Seño, Jose L.

M.S Computer Science

De La Salle University

Taft Ave., Manila, Philippines

**Abstract**

*Free Apples* is an Intelligent Tutoring System developed by the researchers to teach the basics of Java programming to those willing to learn. Free Apples utilized Kolb’s Experiential Learning Theory as the pedagogy or approach of teaching.

The researchers decided to integrate Kolb’s Experiential Learning Cycle within the system to simulate the four-way process while teaching Java. Each topic is separated into modules. Each module contains lessons, exercises and an exam. Practical exercises are accommodated with a video tutorial; on the other hand, the agent provides a feedback whether the submitted code is correct or incorrect.

At the end of each module, the student will be required to take an exam which will unlock the next module. If the student fails, he or she can retake the exam albeit a different set. During the exam, the student will be assisted by the Agent within the system. The Agent will provide hints to the student depending on the facial expression.

Kolb’s Experiential Learning Cycle is integrated in the learning process of the student being immersed in various exercises they gain new experience. From these experiences, the students reflect on their experiences. From their reflection, they conceptualize new theories. Furthermore, the students are given the opportunity to apply these new theories in their next exercises, exam or in the challenges presented to the students.

**Keywords**

Agent, Experiential Learning, Intelligent Tutoring System (ITS), Facial Expression Recognition (FER), Java

**1. Introduction**

Programming is one of the most useful technical skills in the modern age. It leads to a great deal of opportunities in the digital world. As a result, learning to program can make an individual more productive, efficient and effective. There is vast amount of programming languages to learn, each having its own use, syntax, advantages, and disadvantages.

Learning a programming language can be intimidating as it may seem difficult, but with proper guidance and approach of learning, it can be enjoyable and convenient. This can be done through an innovative teaching-learning style.

One of the emerging approaches to teach students efficiently is the Intelligent Tutor System. According to the Valerie J. Shute and Joseph Psotka [27], “*Our working definition of computer-tutor intelligence is that the system must behave intelligently, not actually be intelligent, like a human. More specifically, we believe that an intelligent system must be able to (a) accurately diagnose students' knowledge structure, skills, and/or styles using principles, rather than pre-programmed responses, to decide what to do next, and then (b) adapt instruction accordingly*.”

An abundance of research exists on the different implementations of the Intelligent Tutoring System on different subjects in the field of education. For instance, AutoTutor an Intelligent Tutoring System that assists college students in learning Newtonian physics, computer literacy and critical thinking skills by simulating the discourse patterns and pedagogical strategies of a human tutor.

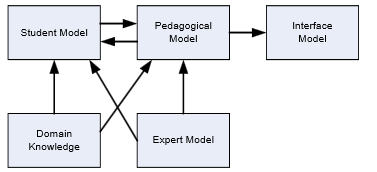
Inspired from this idea, the authors of this study will use the Intelligent Tutoring System to teach students the Java programming language. An existing research authored by Lim-Rañola et. al [16] which is “An Agent-Based Intelligent Tutoring System for Java Using the Experiential Gaming Model as the Pedagogical Module” is similar to this proposal, the difference between the aforementioned study and this study is the Pedagogical Module and the approach on enhancing the learning process of the student. Lim-Rañola et. al used Experiential Gaming Model as the Pedagogical Module and also integrated the concept of a game to motivate the students to learn.

The approach of the authors of this study is to use Experiential Learning as the Pedagogical Module. The agent will provide feedbacks in the exercise, and will be integrated with Facial Expression Recognition which will be used to assess the affective state of the student in the exam and serves as a basis on how the agent will provide hints.

**2. Related Studies**

**2.1 Components of an ITS**

According to Joseph et. al cited in the paper “Design of an Intelligent Tutoring System that Comprises Individual Learning and Collaborative Problem-Solving Modules” by Tuaksubun et. al [24], there are five modules or components of the system where the expert model is an additional module to their system as shown in Figure 1.



**Figure 1: Components of ITS [24]**

**Student Model –** stores the information of the student as well as the feedback from the student.

**Domain Model –** stores the content, lesson and teaching pattern.

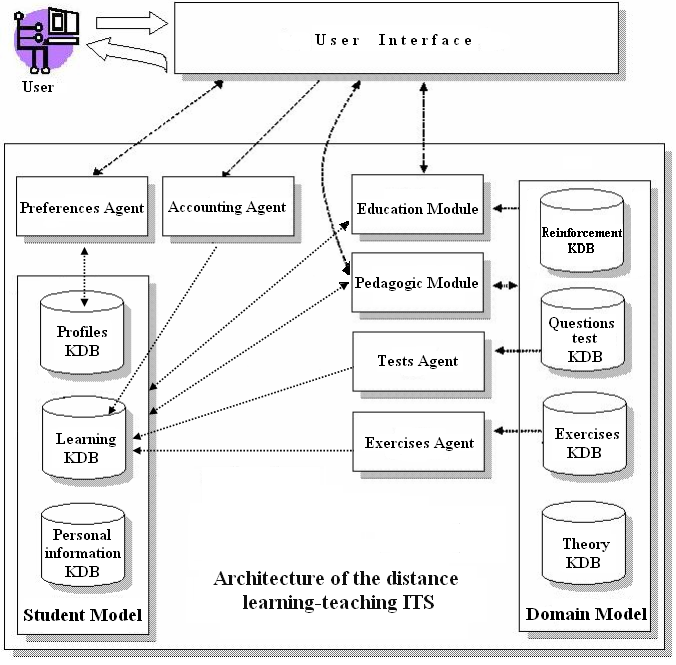
**Expert Model –** stores rule base in solving problems.

**Pedagogical Agent –** responsible for retrieving the information from student model and expert model and process the appropriate content and activities.

**Mentor Agent –** resembles the facilitator of ITS that is responsible for the transformation and storage of information as any parts of the system may require.

**User Interface –** It is the bridge for communication between the student and the system and the system as well as reviewing the behaviors of the student. The information is sent back to the student module.

Figure 1 illustrates the architecture of the distance-learning Intelligent Tutoring System and its main components.



**Figure 2: Architecture of the Distance Learning-Teaching Intelligent Tutoring System [7]**

Firstly, the student reads lectures for the current topic, organized and presented by the Pedagogic Module through the interface. After reading the lectures, the student has to solve the proposed exercises prepared by the Exercises Agent. Afterwards, if the student finishes answering the exercise, the agent stores the exercise taken by the student in the learning KDB. After solving the exercises, the student has to solve the test questionnaire provided by the Tests Agent. Additionally, the Tests Agent stores the provided test questionnaire in the Learning KDB. If there are more topics to be discussed, the system goes back to presenting the lectures for the topics to be discussed. Otherwise, the student has finished studying the subject matter.

The three main modules in the Figure 2 are the Student Model, Domain Model, and Pedagogical Model. The Student Model is composed of three knowledge database (KDB):

**Personal Information KDB** – stores personal data of user to control access to the system.

**Profiles KDB** – store the student’s level and learning style.

**Learning KDB –** stores the exercise data such as tests and activities which have been taken so far by the learner and the metadata about the exercises i.e. the duration of the test etc.

The Domain Model is composed of four knowledge database (KDB):

**Theory KDB –** incorporates the study material or pages of theory that have been prepared for teaching on subject matter.

**Questions test KDB** - stores the set of test questions related to the subject matter.

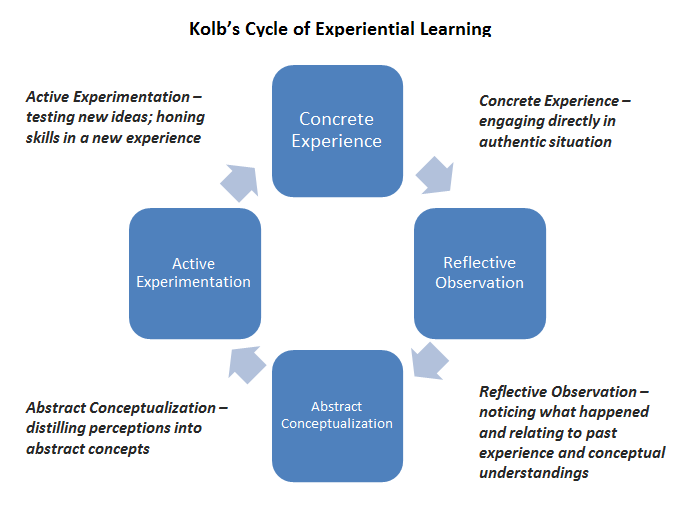
**Exercise KDB –** stores the set of exercises on the subject matter.

**Reinforcement KDB –** comprises the data used by the Pedagogical Module to prepare the material to be presented when a student needs reinforcement.

The Pedagogical Model provides the functionalities to efficiently present the concepts to the student. This module or model has three main tasks: Provide learning guidelines and reinforcement or student. Update the statistics of the exercises and tests presented in the Domain Model. Store the reinforcement data into the Learning KDB and the responses or feedback given by the students.

Other components are Preferences Agent, Accounting Agent, Education Module, Test Agent, and Exercises Agent based from the architecture of the distance learning-teaching ITS [7].

**2.2 Experiential Learning**



**Figure 3: Kolb’s Cycle of Experiential Learning [6]**

Figure 3 illustrates the process of experiential learning. The four steps in the cycle are:

**Concrete Experience (Do)** - a new experience of the situation is encountered, or a reinterpretation of existing experience.

**Reflective Observation (Think) –**perceives the happening and relating it with the past.

**Abstract Conceptualization (Conclude)** - reflection gives rise to a new idea, or a modification of an existing abstract concept.

**Active Experimentation (Adapt/Test)-** the learner applies them to the world around them to see what results.

According to Kolb [3], effective learning is witnessed when a person progresses through this cycle of four stages:

1. Having a concrete experience
2. Observation and reflection on that experience
3. Formation of abstract concepts (analysis) and generalizations (conclusions)
4. Which are then used to test the hypothesis in future situations, resulting in new experiences.

For this reason, effective learning only occurs when a learner is able to execute all four stages of the model. Hence, no one stage of the cycle is an effective as a learning procedure on its own. Learning is an integrated process with each stage being mutually supportive of each other. Kolb [19] also adds that it is possible to enter the cycle at any stage and follow it through its logical sequence.

**3. System Architecture**

conceptual-final.png

**Figure 4: System Architecture**

The student creates a new profile on his or her first use of the program. Then, the student is prompted to input his or her credentials such as username, password, name, age, sex, school, and year level. When the student completes the registration, the Agent will insert all information to the User KDB which is in the student model.

After a successful registration, the student is given access to the system. The student provides his or her username and password in order to load his profile. The Agent checks if the credentials provided exist in the user database, and thereafter, it will retrieve the data of the student when found. Upon successful retrieval of data, the Agent will load all of the following:

Student Grades from the User KDB, Exercises from the Exercises KDB, Modules, Text Lessons and Video Lessons from the Lesson KDB, Practical Exercises, Exam Exercises, Challenges, from the Exercise KDB, and Exams from the Exam KDB.

The student is redirected to the home page once the loading is finished. Also, the student has an option to change the theme and feel of the system.

Initially, the student will only be able to access the first module and its contents. Upon opening the first module, the Agent will retrieve all the Text Lessons related to the module, eventually, it will show the first Text Lesson which is the introduction. Afterwards, the student can proceed to the next Text Lesson which is Lesson 1. Subsequently, the student is required to finish Practical Exercises for each lesson, in order for the student to experience hands-on coding in Java. Succeeding lessons will be unlocked by clearing the exercise for that lesson.

The exercises are composed of two parts, the first one is the video lesson and the second is the practical exercise itself. The student is given instructions on what he or she is supposed to do for the sake of clearing the exercise. The video lesson acts as a guide on how to code in Java and the practical exercise is the area wherein the student does the coding.

In the exercise interface, the student is given a text area, where he or she types the code. After submitting the exercise, the code can be checked by the compiler and the output can be viewed in the console. Finally, a response is given by the Agent if the exercise is correct or the Agent provides a tip or an error report whenever he or she makes a mistake in doing the assigned task.

These work together in forming the dynamic tutoring system module in the system. If the student completes the assigned task, the exercise then locks and saves the code into the User KDB and unlocks the succeeding lesson in the module. This is done in order to give the student an option to review the past exercises completed.

After completing all lessons and exercises in the module, the Agent will unlock the exam for that module deeming that the student is already equipped with the knowledge to move on to the next module. The exam is made up of three sets. Each exam has a specific number of items composed of multiple choices about concepts tackled in the module and a coding exercise to go along with it. The time limit of the exam is 25 minutes. Each item is scored one point and five point for the coding exercise. The passing grade for the exam is 75%.

Before the start of the exam, the browser is opened by the Agent to initialize the Facial Expression Recognition (FER). This is used to capture the expression of the student in a span of time. As the student takes the exam, it writes the emotion in a text file and the Agent in return continuously reads this text file to determine whether the user is having difficulty in the exam. When the Agent detects the student’s emotion as anxious, it will respond by temporarily allowing the student to view the hint for that question. The hint only serves as a guide, but will never give the answer itself.

Once the exam is submitted or finished, the answers will be evaluated, graded and saved. If the student passed the exam, the next module will be unlocked. However, if the student failed, he or she would take the exam albeit a different set in order to proceed to the next module.

All of these are repeated until the all modules are completed. Furthermore, there is an opportunity for the student to take on a whole new module, the Challenges module. In this module, there will be a list of problems where the student has to solve each one of them by using known and existing algorithms in the programming world such as search algorithms, sorting algorithms, and etc. These problems are not graded and not required and is purely optional but serves as extra or additional knowledge for future purposes if ever the user decides to truly enter the world of programming.

**4. Presentation and Analysis of Results**

The researchers conducted the testing with 20 samples consisting of 1st year level college students. The samples studied five modules of the system.

**Figure 5: Module 1 (Basic Syntax)**

**Figure 6: Module 2 (Variables)**

**Figure 7: Module 3 (Basic Operators)**

**Figure 8: Module 4 (Conditionals)**

**Figure 9: Module 5 (Loops)**

**Figure 10: Average Exam Grades**

The graphs display the scores and the trend of the average exam scores of students who had multiple attempts. The trends displayed in the graphs can be interpreted as the learning progress of the students. Majority of the students displayed an upward trend in their scores when taking the exam multiple times. This can be interpreted as the learning process of the student that occurs as the student goes through Kolb’s Experiential Learning Cycle as the student review the module/s.

After the testing, the researchers asked the samples to answer the survey that will provide some insights that will contribute to the research.

**Figure 11: Survey Question 1**

This question is one of the basis of the researcher in their conclusion. The current setup is the integration of Kolb’s Experiential Learning within the system.

The setup consists of two key components, Overview/Introduction of lesson and Practical Exercise with a video guide. The researchers asked which components is the most effective in teaching the concepts.

**Figure 12: Survey Question 2**

The result shows that 16 out of 20 samples responded that both components were effective regardless of their learning style whether they are practical or conceptual learner or any other type of learner.

**Figure 13: Survey Question 3**

Another feature of the Free Apples is the Facial Expression Recognition (FER). It is integrated in the system as a means of assistance to the agent that will provide the hint to the student. The agent will read the facial expression, which is limited to neutral and anxious, of the student through the FER program and will enable the access of the student to the hint if it sees that the student is having difficulties based on his expression. This is to ensure that the student will not abuse the hints if they are given free access to it. If the students are given such freedom, they may likely lose interest and motivation hindering the learning process.

**Figure 13: Survey Question 4**

Another way to determine the effectiveness of the system or specifically Kolb’s Experiential Learning Model is that if the students express interest after using the system and studies the subject. This shows that the students were immersed and motivated in using the system and their interest can be interpreted as their willingness to study more about Java.

These results are factors that the researchers took into consideration in making their conclusions. These also provided them insights on what the system lacks and helped the researchers come up with recommendations that will help future researchers of Intelligent Tutoring Systems.

**5. Conclusions**

As shown in the results, the system has been proven effective in teaching Java in terms of how the teaching material was presented and also proven effective in terms of interest after learning it. All of the students interviewed and selected have no prior knowledge in Java, but expressed intent in learning. They were immersed with the course provided by the ITS, and still found Java to be interesting after learning it.

The researchers considered three factors to conclude that the Kolb’s Experiential Learning is an effective pedagogy for an ITS: the learning trend, the response on how effective the system in terms of after using it, and their interest as shown in the test results.

The researchers were able to achieve their primary objectives: to improve the knowledge of students with regards to Java, to provide a feedback from the Agent, and to provide reinforcement from the agent if needed using Facial Expression Recognition. Although the students found the exercises and exams difficult, the researchers concluded that Experiential Learning is an effective approach in teaching Java.

**6. Recommendations**

* Add more sets of exam to avoid students exhausting all the exams and retaking them. This will also provide students more opportunities.
* Add visual aid in presenting lectures, there are some student that prefer to watch someone explain to them concepts rather than read it themselves.
* Provide a visual agent to increase motivation.
* Provide collaboration exercises to also build up soft skills of individuals such as teamwork and communication.
* Add audio in video tutorials to provide better instructions.
* Add more feedbacks provided by the agents.
* Improve accuracy of feedbacks of agents.
* Improve accuracy of reinforcement/hints by the agent provided it is not abused.
* Add the functionality of tracking the strength and weakness of a student to the agent.
* Add a diagnostic test to determine the logical ability of the student.

**7. References**

1. Albert D., and Schrepp M., (1999). Structure and Design of an Intelligent Tutorial System Based on Skill Assignments in Knowledge Spaces: Theories, Empirical Research and Applications. Knowledge Spaces: Theories, Empirical Research and Applications, 179-196. Retrieved fromhttp://css-kti.tugraz.at/research/cssarchive/publicdocs/publications/file1260877867.pdf
2. Arroyo et al., (2001). Analyzing students’ response to help provision in an elementary mathematics Intelligent Tutoring System. Retrieved from http://users.sussex.ac.uk/~bend/aied2001/arroyo.pdf
3. Dewangan M., and Satao K., (2011). Design and Implementation of English Language Tutorial System with Intelligent Feedback System. International Journal of Scientific & Engineering Research, 2(8),Retrieved from http://www.ijser.org/researchpaper%5CDesign-and-Implementation-of-English-Language-Tutorial-System-with-Intelligent-Feedback-System.pdf
4. D’mello et al., (2008). AutoTutor Detects and Responds to Learners Affective and Cognitive States. Retrieved from http://affect.media.mit.edu/pdfs/08.dmello-etal-autotutor.pdf
5. El-Sheikh, E., and Sticklen, J. (1998). A Framework for Developing Intelligent Tutoring Systems Incorporating Reusability. IEA-98-AIE: 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Benicassim, Castellon, Spain, Springer-Verlag (Lecture Notes in ArtificialIntelligence, vol. 1415).
6. Experiential Learning Defined. (2015). In the The University of Texas at Austin Learning Sciences. Retrieved from https://learningsciences.utexas.edu/teaching/engagement/experiential-learning/defined
7. Gascueña J., and Fernández-Caballero., (2005). An Agent-Based Intelligent Tutoring System for Enhancing E-Learning/E-Teaching. Retrieved from http://www.itdl.org/journal/nov\_05/article02.htm
8. Chan Lane H., (2006). Intelligent Tutoring Systems: Prospects for Guided Practice and Efficient Learning. Retrieved fromhttp://www.iaeng.org/publication/IMECS2012/IMECS2012\_pp63-67.pdf
9. Ghadirli H., and Rastgarpour M., (2012). A Model for an Intelligent and Adaptive Tutor based on Web by Jackson’s Learning Styles Profiler and Expert Systems. Proceedings of the International Multi Conference of Engineers and Computer Science, 1, Retrieved from http://www.iaeng.org/publication/IMECS2012/IMECS2012\_pp63-67.pdf
10. Hill, Robin., (1998). What Sample Size is "Enough" in Internet Survey Research (pp. 3 – 4). New Zealand: Interpersonal Computing and Technology: An Electronic Journal for the21st Century
11. Java Tutorial. (2016). Retrieved from https://www.tutorialspoint.com/java/
12. Kaufer, M.J, Emoodji, (2015), GitHub repository, https://github.com/mjkaufer/Emoodji
13. Landowska A., (2013). Affect-awareness Framework for Intelligent Tutoring Systems. Retrieved from http://kio.eti.pg.gda.pl/publications/files/2013\_Landowska\_Affect-awareness\_Framework\_HSI.pdf
14. Lesson: Language Basics (2015). Retrieved from http://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html
15. Lewis, L.H. & Williams, C.J. (1994). In Jackson, L. & Caffarella, R.S. (Eds.). Experiential Learning: A New Approach (pp. 5-16). San Francisco: Jossey-Bass.
16. Lim-Rañola et al., (2015). An Agent-Based Intelligent Tutoring System for Java Using the Experiential Gamming Model as the Pedagogical Module(Unpublished undergraduate dissertation). University of Santo Tomas, Manila, Philippines.
17. Malik, D.S. (2012). Java Programming: From Problem Analysis to Program Design, Fifth Edition. Boston, Massachusetts: Course Technology
18. Moon, J.A. (2004). A Handbook of Reflective and Experiential Learning: Theory and Practice. New York: RoutledgeFalmer.
19. Mcleod S., (2013).Kolb – Learning Styles. Retrieved from http://www.simplypsychology.org/learning-kolb.html
20. Mcleod S., (2012). Experimental Method. Retrieved from http://www.simplypsychology.org/experimental-method.html
21. Melis et al., (2001). ActiveMath: A Generic and Adaptive Web-Based Learning Environment. International Journal of Artificial Intelligence in Education, 12, 385-407.
22. Sampathkumar et al. (2014). Designing an Adaptive Distributed Tutoring System based on Students’ Learning Style and Collaborative Learning using Intelligent Agents. International Journal of Computer Applications, 87(17), Retrieved from http://research.ijcaonline.org/volume87/number17/pxc3893972.pdf
23. Schildt, H (2007). Java: The Complete Reference. Retrieved from http://iiti.ac.in/people/~tanimad/JavaTheCompleteReference.pdf
24. Tuaksubun C., and Mungsing S., (2007). Design of an Intelligent Tutoring System that Comprises Individual Learning and Collaborative Problem-Solving Modules. Retrieved from http://ijcim.th.org/SpecialEditions/v15nSP3/P17eLearningAP\_DesignOfAnIntelligent.pdf
25. Whiteley W., (2008). Artificially Intelligent Adaptive Tutoring System. Retrieved from http://cs229.stanford.edu/proj2008/Whiteley-ModernIntelligentTutoringSystem.pdf
26. Whitehill et al., (2008). Automatic Facial Expression Recognition for Intelligent Tutoring Systems. Retrieved from http://mplab.ucsd.edu/~jake/its08.pdf
27. Valerie S., and Psotka J., (1994). Intelligent Tutoring Systems: Past Present and Future. Retrieved from http://www.dtic.mil/dtic/tr/fulltext/u2/a280011.pdf