Project Proposal

Myles Lefkovitz
mlefkovitz@gatech.edu

Introduction

The problem: How can we develop a gamified/intelligent tutoring system that effectively delivers language education that minimizes time spent learning while maximizing learning effectiveness and learning engagement?

Existing solutions: There are existing gamified ITS implementations for education. There are also existing language learning applications. However, there are no gamified ITS implementations for language learning.

Track: This is a **Development Track** problem.

The project: This project will develop a working prototype of a language learning application that implements best practices in language learning along with an intelligent tutoring system (to maximize learning effectiveness) and gamification (to maximize learning engagement).

Related Work

I focus on three key terms from literature and industry, which are defined below: Intelligent Tutoring Systems (ITS), Game-based learning, Spaced Repetition Systems (SRS)

Intelligent Tutoring Systems:

Intelligent tutoring systems ("any computer program that contains some intelligence and can be used in learning" [1]) apply intelligence to learning by providing prompt and customized feedback to learners [2]. "Research on prototype systems indicate that ITS-taught students generally learn faster and translate the learning into improved performance better than classroom-trained participants." [3]. Another study found

statistically significant evidence of the impact that an intelligent tutoring system can play in the classroom setting [4].

Game-based learning:

"Game-based learning refers to the borrowing of certain gaming principles and applying them to real-life settings to engage users" [5]. 'Gamification' is the application of game-based concepts (like using points, etc) to non-games.

In one study, students who learned with an educational game performed on average 20% better on a post learning test than the control group. Additionally, students respond resoundingly positively to game-based learning [6].

Spaced Repetition Systems:

"The spacing effect is the observation that people tend to remember things more effectively if they use spaced repetition practice (short study periods spread out over time) as opposed to massed practice (i.e., "cramming")" [7]. "Across 254 studies comparing massed versus spaced practice on later memory for verbal information (e.g., words, sentences, facts, passages), overall, spaced practice dominated massed practice in recall performance" [8]. Spaced repetition systems implement the spaced practice concept.

Related Work Across These Categories:

Applications combining Intelligent Tutoring systems with games can keep learners engaged (gaming aspects) and maximize the learning effectiveness (ITS aspects) of these educational materials [9]. Most popular ITS implementations are focused on math and science education. There are few ITS implementations in the language learning space. Further, there are few game-based intelligent tutoring systems that are focused on language learning. One such system developed in an academic setting describes the necessity for frequent design assessments to be performed [10].

There are many popular SRS-based language learning systems. DuoLingo (the most popular language learning application[7]) and Anki (another popular language learning application [11]) both implement concepts from spaced repetition practice.

Little work has been done on combining intelligent tutoring and spaced repetition systems, particularly in the field of language learning. If research consistently shows that spaced repetition and intelligent tutoring systems each improve learning effectiveness, then combining them in a single application should further promote increased learning. Additionally, research consistently shows that game-based learning increases learning engagement and contributes to increased learning. Combining SRS, ITS, and game-based learning methods should increase learning effectiveness and engagement.

Proposed Work

The intended release version will combine spaced repetition, intelligent tutoring, and gamification in web-based and mobile apps.

As a novice software developer with limited development experience, I'm limited in the scope of what I am capable of developing within the time constraints of this semester.

I plan to develop a *prototype* of the full language learning application described above. The prototype will not include all features, will have limited learning material, and will not be ready for live users. The purpose of a prototype is to act as a proof of concept that can be used for further testing against the original assumptions.

The Tool – Interfaces

I will design a web-based application (accessed via a browser) by forking an existing, open source SRS system (like Anki). The web-based application will include:

1. A homepage with a description of the tool (maybe a short video demonstration)
2. User sign-in
3. A status page showing the words learned and their learning levels.
4. A new word memorization page where new words can be added to a user's practice list.

5. A practice page where flash cards are presented to the user.

The Tool – Development

The tool will be developed in the language of the open source package, likely JavaScript/HTML/CSS.

Scope of Work

I plan to modify an existing open source spaced repetition learning platform, limit its scope, add learning material, and add basic ITS features.

Ideally, this project would add more advanced ITS features and add gamified features.

Basic ITS features

The two basic ITS features planned for addition in this tool are a 'problem solving suggestion' component and a 'dependency-based teaching' component.

1. Problem solving suggestion

   An ITS should include customized learning suggestions for specific types of errors.

   For consistently incorrectly answered memorization problems, the ITS should identify that the student has trouble with the memorization of this particular item. Once a problem has been identified, the ITS should attempt to identify the cause. If the item is a word that has a common root, highlighting that root and its definition is a reasonable suggestion. Otherwise, the ITS should suggest a new mnemonic for memorization.
2. Dependency-based teaching

   An ITS should teach simple items first and move on to more complex items later. Teaching a root word before teaching complex words that utilize that root is standard practice. An ITS can ensure that a student has an understanding of the root word (in this example) before allowing the student to move on to the complex word.

By implementing these two features, the hope is that this language app increases learning efficiency greater than an application utilizing SRS or ITS methodologies alone.

Project Plan

Below is a summary of the work to be performed each week. See the Proposal Task List for the full list of work.

Week 8

1. Research ITS and SRS platforms (4 hours)
2. Set up a local development environment as a fork of an open source SRS environment (4 hours)
3. Read through the documentation and add data to the tool (4 hours)

Week 9

1. Add initial learning material to the tool: add material across 3 categories (3 types of right/wrong pairs with corresponding mnemonics) (5 hours)
2. Test the software (2 hours)
3. Document the implementation plan for a selected ITS open source platform (4 hours)

Week 10

1. Implement the ITS within the existing flashcard system (6 hours)
2. Create students (0.5 hours)
3. Create training plans for students (1.5 hours)
4. Prepare Milestone 1 Progress Report with low-fidelity prototypes (drawn diagrams of the intended user interfaces and interactions) (2 hours)

Week 11

1. Implement the students as users in the system (1.5 hours)
2. Test that the student model of the ITS is working properly (1 hour)
3. Design and implement the ITS problem-solving suggestions feature (8 hours)

Week 12

1. Order learning material, develop ITS dependency feature, and test the feature (10 hours)

Week 13

1. Improve the user interface and all user interactions by fully implementing the ITS features developed in previous weeks and deactivating most features not relevant to the current tool while testing and resolving bugs (8 hours)
2. Identify potential beta testers and recruit testing participants (1 hour)
3. Publish beta test-ready application (2 hours)

Week 14

1. Conduct beta tests and review results (2.5 hours)
2. Improve application usability based on beta test findings (2.5 hours)
3. Add additional learning data to prepare for demo (1 hour)
4. Work on final paper (5 hours)

Week 15

1. Improve application usability and fix any outstanding bugs (4 hours)
2. Add additional learning data to prepare for demo (1 hour)
3. Work on final paper (5 hours)

Week 16

1. Improve application usability and fix any outstanding bugs (1 hour)
2. Complete/edit final paper (2 hours)
3. Develop presentation materials (2 hours)
4. Develop presentation script (2 hours)
5. Record presentation (2 hours)
6. Edit presentation (2 hours)

Deliverables

Intermediate Milestone 1

Milestone 1 will include two deliverables for review with my mentor:

1. An in-depth progress report detailing my activities and learnings until that point
2. A series of low-fidelity prototypes (drawings/diagrams) indicating the interactions the UI is intended to support.

I will discuss with my mentor whether it's worth spending time using these interface diagrams to conduct research with a set of potential users.

Intermediate Milestone 2

Milestone 2 will be a beta test-ready application. The plan is for the application to only implement a few features and have a limited set of learning material by milestone 2. A beta test is a great way to receive feedback on the interactions and the interface as well as a way to uncover bugs without spending additional resources on testing.

Final Deliverable

The final deliverable for this project will consist of four components:

1. A paper

   The final paper will describe the basis behind this project (the research reviewed, the problem identified, the lack of existing solutions to this problem, and the consideration behind this project), the work performed, and the author's (my) intended next steps for this project.
2. A presentation

   The final presentation will be a short video demonstrating the developed application.
3. The code used to develop the application
4. A link to the web-based application

References

1. Freedman, R., Ali, S. S., & McRoy, S. (2000). Links: what is an intelligent tutoring system?. *intelligence, 11*(3), 15-16.
2. Psotka, J., Massey, L. D., & Mutter, S. A. (Eds.). (1988). *Intelligent tutoring systems: Lessons learned*. Psychology Press.
3. Ong, J., Ramachandran, S., (2000). Intelligent Tutoring Systems: The What and the How. *American Society for Training & Development.*
4. Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education (IJAIED),* 8, 30-43.
5. Pho, A., & Dinscore, A. (2015). Game-based learning. *Tips and Trends*.
6. Jenkins, H., Klopfer, E., Squire, K., & Tan, P. (2003). Entering the education arcade. *Computers in Entertainment (CIE), 1*(1), 8.
7. Settles, B., & Meeder, B. (2016). A trainable spaced repetition model for language learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Vol. 1, pp. 1848-1858).
8. Kang, S. H. (2016). Spaced repetition promotes efficient and effective learning: policy implications for instruction. *Policy Insights from the Behavioral and Brain Sciences, 3*(1), 12-19.
9. McNAMARA, D. S., JACKSON, G. T., & GRAESSER, A. (2009). Intelligent Tutoring and Games (ITaG). In AIED 2009: 14 th International Conference on Artificial Intelligence in Education Workshops Proceedings (p. 1).
10. Johnson, W. L., & Beal, C. R. (2005, May). Iterative Evaluation of a Large-Scale, Intelligent Game for Language Learning. In *AIED* (pp. 290-297).
11. Godwin-Jones, R. (2011). Mobile apps for language learning. *Language Learning & Technology, 15*(2), 2-11.