

Capella's Song: A Game2Learn Mini-Game Launcher

Michael John Eagle
Advisor: Dr. Tiffany Barnes
mjeagle@uncc.edu
tbarnes@uncc.edu



Michael John Eagle
UNCC

Introduction

Enrollments in computer science programs are declining, especially for women and minorities, while demand for IT jobs is increasing. *Game2Learn* targets novice students by presenting computing concepts in a less intimidating setting, while leveraging the interactivity and motivation of games (Garris, 2002).

Several games have already been developed; however, the games were disjointed and there was no formal delivery system. *Capella's Song* is an online meta-game that will provide a common link for *Game2Learn* games while also maintaining detailed statistics on the student's overall progress. Team-based motivation will encourage students to do extra homework.

Background

The *PUMP Algebra Tutor (PAT)* is an intelligent tutoring system that teaches algebra. *PAT* offers constant feedback and help; and tracks student progress using skill-meters. *Capella's Song* allows the student to reflect on their progress with use of *PAT*-style skill advancement.

Whyville is a web-based cooperative learning game that teaches a variety of subjects. Players complete mini-games to earn currency that can be used to customize their character's appearance. *Capella's Song* will allow character customization by using in-game currency in a way similar to *Whyville*.

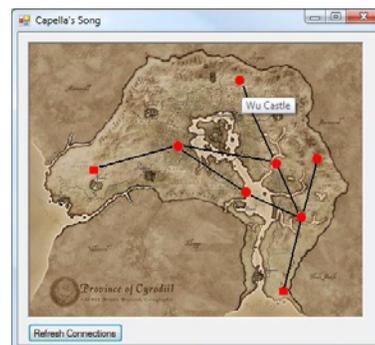
Collaborative learning has been shown to improve performance in critical-thinking and allows students to learn from each other (Gokhale, 1995). *Capella's Song* will incorporate collaborative learning techniques and place the students into groups. Players can communicate through an embedded chat program.

Research

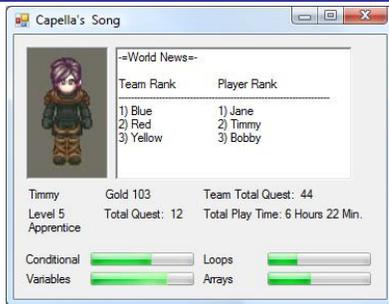
Capella's Song needed a way to communicate with all of the *Game2Learn* mini-games; previous games logged students by writing to a flat text file. These files were difficult to interpret and required a specialized text parser for each game's logs.

To remedy this, I developed a standardized XML log format. *Game2Learn* games can now record each activity, what topic it covers, and its cognitive domain level; based on Bloom's Taxonomy of educational objectives. The logs also provide useful information like interaction timestamps and position information.

The *Capella's Song* game world is represented as a 2D map. Small pictures on the map represent missions and towns. Selecting a mission will launch a mini-game in a separate window. Addition of new content is made easy with a *Capella's Song* map editing tool.



Capella's Song World Map



Capella's Song player statistics

After a student completes a mini-game, *Capella's Song* will read the XML log and update the player's statistics. The student can then see their progress and compare it with their whole class.

To facilitate traditional computer science homework assignments I developed a small IDE using Microsoft .NET's support for run-time compilation in both the C# *Capella's Song* game and other XNA *Game2Learn* games. Teachers can require their students to design normal programs in any .NET language. I designed a system that will automatically check the player's code.

Team communication is done with use of an embedded IRC client.

Impact

Future *Game2Learn* games will share the same logging format; this will help not only the developers, as there are many examples, but also allow bootstrapping of the data. We will be able to see common mistakes and identify problems.

Incorporation of Bloom's Taxonomy will encourage *Game2Learn* developers to think about the pedagogy used in their games. It will also allow us to compare the game's tasks to real world problems.

The *Capella's Song* meta-game serves as a content delivery system. Players can download and play all of the games online. By making the *Game2Learn* games more readily available we can increase the student's exposure to valuable learning tools.

Run-time compilation has limitless opportunities for *Game2Learn* games and introduction of traditional assignments will insure the students get real coding practice.

Conclusions

Microsoft's gaming development IDE, XNA, and Visual C# are well suited for creating educational video games.

From past projects we have learned that educational video games need to be designed with a solid pedagogical foundation in order to be effective; and that game players are motivated by feedback and skill progression. Judging the success or failure of an educational game requires detailed logs and data mining. XML proves to be the ideal format because of its standardization and ease of parsing.

An online meta-game is a good delivery system for educational games. Collaborative learning is a powerful educational tool and the online nature of *Capella's Song* will permit group work. We hope that students will not mind doing more homework because they will be more engaged.

To increase exposure, educational games must be readily available to students and educators.

Including traditional homework assignments as part of the *Capella's Song* meta-game will increase the acceptance of *Game2Learn* as part of a computer science curriculum.

Future Work

Future *Game2Learn* developers should consider what topic and Bloom level their games will incorporate before development. These games should take advantage of run-time compilation to provide students with interactive problems that use real programming languages.

An intelligent tutoring system should be created to parse student game logs and suggest help. This system should also suggest what games the student should play based on what Bloom level the student is on for each topic.

A study should be performed on a CS1 course. The students in this class should play *Capella's Song* instead of traditional homework.

I predict that students will be more engaged in the coursework and learn the same, if not more, than in a traditional CS1 class; the retention rates of the students in these classes should also be monitored.