CS486C – Senior Capstone Design in Computer Science

Project Description

Project Title: RTX Gamification Classmate System	
Sponsor Information: the RTX Project	Terry E Baxter, Professor Environmental Engineering Civil and Environmental Engineering Northern Arizona University terry.baxter@nau.edu

Project Overview

How does Blackboard Learn or any other online course delivery and grading system – called Learning Management Systems or LMS – work for you? Are you able to easily see and understand your grades and track your progress across the semester? Do the actions you conduct on the LMS make sense; in other words, are they intuitive? At any point in the semester, can you predict what your final grade might be? Can you "game" your grade to determine what you need to earn on future assignments to get the grade you are hoping for? While your Instructor may use different teaching strategies such as blended learning, flipped classrooms, synchronous, asynchronous strategies, or some combination commonly labelled Flex, everything eventually still comes back to having to use an LMS that provides the assignments, deadlines, grades, and other information used in the class.

Course gamification can improve learning experiences through a framework that allows conventional teaching approaches to either be enhanced or altogether changed. Ideally, a gamified course promotes learning through a variety of experiences and challenges that guide the learner through a course in a way that maintains interest and motivation to learn, as well as to ensure that content is covered and understood rather than skipped. Even though redesigning a conventional course learning environment already involves a tremendous amount of time, teachers are more willing to put in this effort (planning, redesign, and new materials) because it is more familiar and predictable. Gamifying a course however, involves an additional level of planning and redesign that is beyond what is already done.

Gamification is still unfamiliar in most educational settings and to most teachers, and has not yet become a serious alternative to the mainstream teaching approach. This is partially because the idea of how to actually gamify a course is neither a trivial concept to grasp or to implement. Additionally, gamifying a course becomes quickly limited and constrained within the context of a commercially available LMS that is built around conventional teaching styles and are simply augmented with online learning hooks such as blended, flipped, Hi Flex, synchronous, and asynchronous delivery modes. Although these hooks are definitely useful, there is a need for an intuitive course gamification platform that can not only allow teachers to organically create a gamified course, and manage its content and delivery, but that can be used by learners in a way that is also intuitive and organic.

As your potential project client you should probably know something about me and how I fit into this project before the details are presented. I have been at NAU since 1993 and teach in the Civil and Environmental Engineering programs. Beginning in 1996, I have strived to integrate and enhance courses by developing web-based and multimedia content. I have managed project teams conducted individual projects involving a variety of talents and skills ranging from audio, video, 3D modeling and animation, programing and application distribution. Three of the more notable projected resulted in a multimedia wastewater operators training program for the City of Phoenix (built in Macromedia (Adobe) Director), a Web-based live video streaming viewer (RTX PlayLive, a Macromedia (Adobe) Flash Web page application), and the Initial RTX Classmate (a C# application with a distributable Windows application installer). All work that I do in these areas is done under a project called the Remote Teaching eXperiment or simply the RTX Project.

The Problem

RTX Classmate was initially built in 2015 to serve as a companion for a computational-heavy freshman course introducing methods for conducting material balance analyses (CENE 150L Environmental Engineering Computations Lab) to allow students direct access to their course assignments' status and overall grade. The reason for developing this app was driven by two functional requirements; 1) to overcome Blackboard's inability to handle a complex grading system and 2) to provide students with a tool that could be also used to triage course assignments by determining what scores they would need on future assignments in order to obtain a particular grade in the class.

Beginning in 2017 the decision to gamify CENE 150L inside of Blackboard was made and a plan for creating "*Material Madness*" was formalized. Since June 2019 there have been 3 stages of major gamification revisions to how the course content delivery is managed for individual players, and to how the course's grading system is linked to the game's strategy and overall gameplay. Managing the organization and control of content, and tracking the status and grade of multiple course players in Blackboard is now significantly more complex and inefficient. These changes, coupled with having reached the limit of what can done for gamification in Blackboard or any other LMS that NAU might use, have led to the following problems.

- RTX Classmate is no longer a usable course companion for CENE 150L, and
- Building a gamified course using an LMS course delivery platform becomes increasingly counter-productive as more gaming features are added.

A recent assessment of RTX Classmate concluded that this application must be redesigned and expanded to not only meet the new requirements of CENE 150L but also to provide a generic course gamification platform that is robust enough to allow most courses to be gamified.

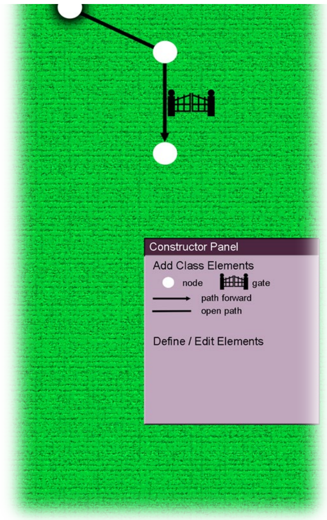
A Vision for the Solution

The vision for a solution to address the problem above is to redevelop RTX Classmate from the ground up as a software system that can easily allow teachers to construct, reconstruct, test, and deploy a course through a map-like game board interface that provides the exact look players would see and use while playing the course. The new RTX Gamification Classmate would include a desktop application used to develop, test, and deploy (publish) a gamified course design to a secure internet location. This content would then be playable in a companion web application either on a PC or a mobile device without having to update a separate mobile applications (*e.g.*, for iOS and Android devices). Data used by the gamified course to deliver course content, define game rule settings, and maintain player data (player accounts, profiles, game status, skill level, and course scores) that are also stored and updated to a secure internet location.

Key Features of Envisioned RTX Gamification Classmate Software System

Some of the main key features of the overall software system envisioned include the following.

Figure 1 Example map-like constructor interface, tool panel, and course node selected.



Desktop Application

- Not dependent on internet access to construct, reconstruct, and test a gamified course
- Not limited to use for just courses that are being gamified
- Is extensible with add-ons or plugins
- Creator does not need to know or use syntax-dependent coding
- Uses an intuitive “looks-like” user interface map to construct or reconstruct a course (see mocked up example at left)
- Connects to a secure internet location to transfer and manage data
- Uses tool panels to construct and define course elements, and their attributes and behaviors

Web Application

- Handles player login and prevents potential conflicts due to multiple players using the system at the same time
- Detects and adapts to the device being used by the player
- Accesses a secure internet location to load data to display course map elements, and obtain their defined attributes and behaviors
- Accesses a secure internet location to load and update player account data such as player profile and game status
- Handles standardized player announcements, warnings or hints.

- Provides players access to different areas and features using a tab-based menu

Integrated System

- Is independent from NAU ITS other than basic services provided for web site, database, and securing the site.
- Can be extended with new features without having to rebuild or modify the software
- Can handle and store “announcements” pushed from desktop app to web app.
- Allows players to select an optional MMS service to receive course announcements
- Desktop and web applications can be linked to respond to one another in real time for a “classroom demonstration” mode and a motoring or “player tutorial” mode
- Can handle multiplayer classroom chat sessions when added as a defined course element
- Can handle launching of URLs from web app when added as a define course element

What’s Really Needed?

The most immediate need to achieve this solution would be, at a minimum, to develop the desktop application to construct, reconstruct, and test a gamified course, and to be able to transfer necessary data to a secured internet location.

Impact of a Successful Project

A course gamification platform that allows course construction and reconstruction to occur regardless of whether for gamification, will have high potential for introducing the concept of course gamification and advance interest in this teaching and learning approach. A successful project would allow the current CENE 150L course to be constructed, reconstructed, tested, and deployed with all of its current features for the purpose of conducting a full beta test (of the desktop application) and possibility for demonstration beyond the capstone course.

Two additional goals that would stretch this project beyond this initial milestone of success would be the following.

a) Compile a ***player’s version*** of the desktop application that consists of only the testing features so that students could play (beta test) the course from an executable file.

b) Extend the overall system build to include the web application and system integration.

Knowledge, Skills, and Expertise Required for this Project

The knowledge, skills, and expertise that are considered required for completing this project successfully included but are not limited to the following.

- Data structures and database systems
- HTML5 and JavaScript, PHP, or some other common scripting language used within dynamic web pages and server side programming
- Java, Python, C++, or other common cross-platform language for desktop application development
- Deploying executable and executable installer files

Equipment, Software, and Cloud Service Requirements

The requirements anticipated for equipment, specialized software, or a cloud service for this project are the following.

- There should be no equipment requirements beyond what is already available through campus resources.
- There should be no specialized software requirements beyond what is considered available through ITS or freely available from online sources.
- There should be no requirement for a specialized secure cloud service beyond what can be provided under the client's existing personal NAU-hosted site and secured through NAUauthZ service.

Software and Other Deliverables

The main deliverables expected at the completion of this project include the following.

- A final report and all ancillary documentation that details the development and software design decision-making process and rationale for selected design, implementation, testing, and a cost estimate for the overall system development, and add-on or plugin software requirements and specifications. This document would provide the basis for future work to modify and extend the product.
- All source code that is professionally documented, packaged, and that will successfully run from an IDE.
- All successfully compiled executable files and any required APIs and installer packages
- A complete set of UML class diagrams.
- User's Manual (written or a video format "how to") for all software features requiring or allowing human interaction.