**Project 3 Proposal**

**The Big Daddies**

Group members:

**Jordan Alexander**, [jordangalexander@utexas.edu](mailto:jordangalexander@utexas.edu)

**Davis Owen**, [sdowen12@gmail.com](mailto:sdowen12@gmail.com)

**Adam Hahn**, [awh635@utexas.edu](mailto:awh635@utexas.edu)

**John Novak,** [john.novak32@gmail.com](mailto:john.novak32@gmail.com)

**Roy Lin**, [roylin7@utexas.edu](mailto:roylin7@utexas.edu)

**Nolan Daniels**, [nolandaniels@sbcglobal.net](mailto:nolandaniels@sbcglobal.net)

**Sheryar Ali**, [sheryar65@gmail.com](mailto:sheryar65@gmail.com)

**General Description of Overall Project:**

The program for this section of the final project will include the basic structure of the overall program. It will allow the creation of students, groups, quests, and all the assignments that they can complete. There will not be a way for students to complete tasks without hard-coding that in this release.

**Purpose and Description of the Freestanding Portion:**

**Parts of Final Project Contained**

* Create (or upload) homework assignments, projects, quizzes, and test content into the system
* Create a series of "Quests" to build certain skills
* Create (or display) a skill list related to a class/assignment/project/quizzes/tests (PARTIALLY)
* Create a level list with corresponding point values
* Keep track of individual student progress
* Allows students to be members of various defined groups (guilds)
* Generate reports in various ways (i.e. individual quests, individual skills, individual students, specific groups, etc)
* Calculate the current level for each student

**User Stories**

Student Class:

Class that represents students; The class will contain attributes name, level, points, skills, groups, and pointsToNextLevel. The student object will also be able to generate a report on the student that includes these attributes.

Student Method (*addSkill*):

Method that will create a new **valid** skill and add the new object to the student's skills list.

Student Method (*addToGroup*):

Method that will add the student object to a specified group, as well as add the group to the player's groups list.

Student Method (*addPoints*):

Method that will add specified point value to student's points attribute and increase the student's level.

Group Class:

Class that represents groups; The class will contain attributes name and members (list of student objects). These group objects will also be able to generate a report on the group that includes these attributes.

Group Method (*addMember*):

Method that adds a specified student to group's member list, as well as the group to the student's group list.

Group Method (*getAverageLevel*):

Method that calculates the average level of all students in the group.

Skill Class:

Class that represents skills that students can possess. The skill will contain the attributes name, level, and points. They will be represented by a string with name and level. Valid skills will be predetermined by the teacher.

Assignment Creation:

Teachers will be able to add .txt files (of a correct format that contains skills, point values, etc.) to create corresponding assignments that will be stored.

Quest Creation:

Teachers will be able to group different kinds of assignments into "quests" and assign skills and point values to the quest.

**Detailed Plan of Work**

* 1st Release
  + Get basic class framework working
  + ~60% of user stories completed
  + Report and Release Notes
  + Unit Tests for those user stories
  + Plan for Acceptance Tests
* Final Release
  + Remaining user stories (Assignment Creation, Quest Creation)
  + Report and Release Notes
  + More unit tests
  + Acceptance Tests
  + Code Refactoring (Plan to only need to do some clean up)

**Plan for utilizing pair programming, unit testing, acceptance testing, and refactoring**

Pair Programming:

Since we have 7 people in a group. We will have two two-people group and one three-people group and also divide tasks for these three groups .

Initial plan: 2 group for the major project forming, 1 group for acceptance testing.

If the time availability is not working for some group member, the group will not strictly assigned, they can meet up anytime but the pair programming log has to be recorded

Drivers need to listen to navigator's idea and navigators provide the best of their knowledge to help drivers

Unit Testing:

We will have one group design the unit tests to begin with, so the other groups can slowly write the code to pass more and more of the unit tests as we progress.

Acceptance Testing:

We will use the group of three people for the acceptance testing. For the first release and second release, this group will not be coding any of that part of the project but will test how the program works and provide feedback on what aspects do not work, or what could change for better use.

Refactoring:

We will have different people to look over the code and refactoring after the code is complete by one person. Also refactoring is kind of relating to coding standard, so we will add new rules to the standard. If there is an algorithm problem/ efficiency, we will calculate the efficiency and the best solution.