**CSCI 1100 – Fall 2015**

**Project Part 1**

***TAs and the project.*** *TAs can advise students on what should be included in a good project and provide suggestions about what features should or should not be included in your Java program. We advise you to stick with the Java constructs and Java classes covered in the course. In past experience students who try to use advanced features or classes often waste too much time debugging and overall this detracts from the quality and focus of the project.*

**Project Part I: Due Friday November 13, 2015 (11:00pm on Moodle)**

**Project Description.** For the project, you will develop and design a program that could be used by elementary (primary) school students to help them learn a basic skill. This program should include game-like or playful elements (to make learning more interactive and fun). Ways you could do this include having levels, questions that increase in difficulty, characters and narrative, scorekeeping, rewards, multiple players, etc.

There are two parts to the project. Part I involves doing some research on current applications and online resources used for teaching students (full details are below). Part II involves designing and developing your own program that combines learning and gaming, while exercising the concepts and skills that you have learned in CSCI 1100. You will hand in a report and do a demo of your program for Part II.

The project is worth 10% of your final grade. You should work on the project in groups of up to 3 members. It is fine if you want to work with someone from the other section (just make sure you indicate this when you hand in your project deliverables).

**Part I** Before you start designing and coding your Java program, you need to do some research. For Part I, you need to search the web and/or use academic sources (e.g., the ACM or IEEE online databases available from Dal's Library page) and find three good examples of programs that children could use in class to help them learn basic skills (e.g., basic arithmetic, spelling and grammar, reading comprehension). For Part 1 you will write a report of 3-4 pages (about 500-800 words plus pictures). You need to submit your report as a pdf on Moodle with the following headings and information (a template has also been included):

**Team**

* List all the members of your team (including their name, Banner ID and Section – note 12:30 class is Section 1 and 9:30 class is Section 2).

**Project Title**

* As a team, decide on the title that you will give your project.

**Applications**

For each program/application you find, include:

* A screen shot of the application with a description of the screen shot
* A short description on what the application does
* At least three features of the application that you think would be good features for your program and **why** you think they are good features for elementary school aged kids
* The URL for the application (or reference)

**Top Features**

Based on these applications and the relevant features, you then need to come up with a ranked list of the 5 features that you would like to include in your program. Make sure you justify why these features are important to your program.

**Game Elements**

Finally, describe how you plan to make your program game-like or playful. You should describe how your program will incorporate at least four game elements (we recommend that you research this: in particular, research *gamification* and *games with a purpose/purposeful games*. Make sure you include any references).

**Sample Output**

In your report, you should provide sample output illustrating generally how your program will run when it is complete. For example, if your program is intended to teach basic arithmetic skills to young children you could supply an example like the following:

|  |
| --- |
| ***Sample Output 1***  *HI - I am your friendly arithmetic tutor.*  *What is your name? Brett*  *What level do you choose? 2*  *OK Brett, here are ten exercises for you at the level 3.*  *Good luck.*  *35 + 18 = 53*  *You are right.*  *0 + 26 = 26*  *You are right.*  *36 + 23 = 59*  *You are right.*  *11 + 3 = 12*  *Oops! wrong. 11 + 3 = 14*  *10 + 33 = 43*  *You are right.*  *20 + 32 = 52*  *You are right.*  *20 + 17 = 37*  *You are right.*  *35 + 39 = 74*  *You are right.*  *10 + 9 = 19*  *You are right.*  *17 + 38 = 55*  *You are right.*  *You got 9 right out of 10*  *Select a higher level of difficulty next time.*  *Bye!* |

**Simplicity.** The focus is on selection of a skill and a method of teaching this basic skill. In particular, the design or use of a windows-based user interface is not required or expected. *Simplicity is important.* You are being asked to design a complete program before you have developed real programming skills. You need to keep it simple so that you will not need to use programming techniques beyond those required in the course.

1. Concentrate on the logic of the teaching system and figure out ways to keep the presentation simple.
2. If you try to build a window interface you are likely to encounter difficulties that take a lot of time and may lower the quality of your project. In particular Applet and Java Swing GUI programming can quickly become a nightmare. The presentation should be kept as simple as possible.