# CS278 FINAL GROUP PROJECT, Fall 2017

# Due Dates:

Proposal: November 21

Code: December 11

Presentation: December 12, 2017 1-3pm

## Instructions

1. Form groups between **2 and 4 people**.
2. Your group will form a mini-software-development business, so **pick a group name for your group**. Trade contact information, and figure out times when you can meet and work together. I’m planning to give you time in November and December before finals for you to work as a group, but you’ll also need time outside of class.
3. **Discuss ideas for an application that uses assembly language, NOTE: you could make a console application or a windows application. Windows apps tend to be more challenging in assembler. Look ahead in your textbook…**
4. Fill out the proposal form and give it to your instructor.
5. Design the application. Use pseudo code to design the main application flow.
6. Implement the application. (See the assessment rubric for items on which you will be graded)
7. Create a final presentation and present during the final exam (see rubric for the presentation)

## Group Project Ideas

**A Game**

* Console Frogger, Console Snake, Console Blackjack, Pong etc.
* Some other game of your own design / choosing

**A Mathematical Application**

* An encryption program that implements RSA
* A decryption program that tries to *break* an encrypted text
* A program for arbitrarily large integers (128 bits, 256 bits, … ) written in assembler. Include functions to read from a file, do addition, subtraction, multiplication, division
* Some other mathematical application approved by your instructor.

**Highly Optimized Algorithm using Both C++ / Assembler**

* If you pick this option, you may want to investigate how to integrate SIMD instructions into your implementation, like we discussed in class

**Some Other Application Approved by Your Instructor**

# CS278 Project Proposal Form Due Nov 21st

# GROUP NAME: FIGHTING RUTABAGAS

# NAME 1: JUDE BATTISTA

# NAME 2: ANDREW McNEILL

# NAME 3: TERSA MOTBAYNOR ALMAW

## Paragraph description of what your group intends to build using assembler

JTA will be creating a hangman console with asci characters. It will be a two player hangman where one person thinks of a word or a phrase, and the other team member tries to guess the letters one at a time. Every time the guesser inputs a letter it is added into the guessed array. If the guess is correct then the target word is updated, and if the word is complete the guesser wins. If the guess is wrong then the asci hangman is updated, and if the hangman is completed then game over and the guesser loses. All throughout the game, it keeps track of the guesses whether the answer is correct or incorrect. Before the game starts the users are prompted to enter their name, the level complexity and the help information.

## List of features listed in order of increasing complexity

GetNames: prompt users to enter their name

GetMaxWordLength: the level of complexity, the amount of letters in words they will guess

GetWinningScore: prompts the user to play

Help: Description of how to pay the game

GetTargetWord: prompt the user to enter a word

GetGuess: prompts the user to enter a letter

CheckGuess: this compares the guessed letter to one of the letters in the target word or already guessed

StateCheck: checks if the users have completed the word, the round or the game and moves on accordingly.

UpdateRoles: sets the roles of who is the guesser and which player is the asker

UpdateResponses: provides feedback depending on the accuracy of the guess

UpdateScore: Refreshes the score depending on the score at the end pf the previous round

UpdateSuccessesfulGuess: Updates the target guess word as the guesser chooses correct letters

UpdateGuesses: Refreshes the guessed letters box

UpdateHangman: Refreshes the graphical representation of the **failure** state

## CS *278 Final Exam Assessment Rubric*

Group Title: FIGHTING RUTTABAGAS

Group Members: JUDE BATTISTA

ANDREW McNEILL

TERSA MOTBAYNOR ALMAW

|  |  |  |
| --- | --- | --- |
| **CATEGORY** | **POINTS** |  |
| **FINAL PROJECT** |  | 100 |
| **TOTAL** |  | 100 |

(4 pts) Fill out and turn in a project proposal sheet (See the previous page)

(40 pts) The project chosen by the group used **assembly language as an integral part** of the project.

(10 pts) Instructor’s perception of project difficulty and applicability to the class.

(10 pts) Instructor’s perception of individual group member’s contribution to the overall project.

(16 pts) The code in a project was cleanly written and commented appropriately.

Each of the following categories is worth 2 points:

a)      All functions and variables are commented.

b)      Pre-conditions for functions are documented

c)      All files are commented appropriately at the top of each file.

d)      The presentation contains high level description of the architecture of the project.

e)      Named variables appropriately according to purpose.

r)    Coding style is clean and easy to read.

g)      Spaces used appropriately.

h)      Appropriate indenting used for nested loops (both assembler and C++)

(20 pts) The group presentation to the class:

Each of the following categories is worth 4 points:

a)      Used presentation software. Group members maintained eye contact with the audience; spoke clearly and loudly enough for everyone in the room to hear.

b)      A description of the specific goal(s) the group tried to accomplish and a description of the unique responsibilities of each group member. What portion of the project did each of you do?

c)      A description of the “design” of the program. Use a flow chart and (possibly) class diagrams (if your project has any user defined classes in it) to illustrate the design of your program.

d)      A very brief, high level, “code walkthrough” to show the rest of the class the code your group wrote. Describe the general outline of your code, but please don’t walkthrough your code line by line.

e)      A description of the “lessons learned” in the process of doing the group project.