Intro to Computers for Engineers (14:440:127)
Class Project Information, Spring 2017

You will need to complete a final project in this class. You will complete this project in a group of 5 students (some groups may be groups of 4). You will choose one of the project options below, and be assigned a group (or choose your own group with option 4). It will then be up to you to do the necessary work to meet the project deliverable level you are aiming for (an A, of course) by the project deadline.

The project has the following important dates:

Week of February 20-24: Decide on your project idea and complete the project survey on Sakai by the end of your recitation.

Week of February 27-March 3: You are assigned teams. Set a weekly meeting schedule, and log your progress each meeting. You will need to show your recitation instructor your work log throughout the semester, which should include *individual contributions* by each team member demonstrating an equitable share of work.

Ongoing After Spring Break: Meet with your recitation instructor for 2 minutes as a team each recitation. Tell them what you worked on, what you need help with, and your plan for getting that help (office hours, finding resources, etc.).

Week of April 3-7: Meet with your recitation instructor and either show them C level deliverables, or your honest, reasonable progress towards the C level deliverables. Showing such project will net you **10 bonus points** on your final project score. These points will be at the discretion of your recitation instructor.

May 3rd: Final projects due by 11:59PM (that is one minute before midnight, not one minute before noon). You must submit all materials online. They should include:

- All project code.
- A 2 page write-up describing the development process behind your project
    - What new concepts did you learn, as a team?
    - What were the greatest challenges?
    - What would you like to have added, given more time?
    - What would you have done differently?
- A 1-2 page **manual** for your instructor. ***This document is important, as it tells your recitation instructor how to view the deliverables of the project, thus assign them a grade. Be as clear and simple as possible with these instructions. Give them to a friend first to try out, if you need to.***
- An acknowledgement of external sources used. You can use external sources, so long as they ***do not directly provide you with the code you need to complete a deliverable.***
- An honor statement signed by the entire team.

1. Chess Board Project **(This Project is More Difficult, but you will learn more)**

   This project will require you to create a chess board with moving pieces in MATLAB. In doing so, you will learn about MATLAB’s GUI builder, called GUIDE, and some basic object-oriented programming principles.

   This project has the following deliverable levels:

   C: Opening your program creates a chess board with chess pieces. You can move pieces from one square to another.

   B: When you move a piece to a square with another piece, the other piece disappears. You can press a button on your GUI to reset the board, and undo the previous move.

   A: Pieces can only be moved according to the rules of chess.

   So long as your project meets those requirements, you will get the corresponding grade. That means a great deal about *how* you accomplish this project is up to you. You can, for instance, make each square a button that you press, or make pieces objects that you drag and drop. You can use any representation you want for different chess pieces (characters, images, etc.). All options are viable. Some might be easier.

2. Solar System Project

   This project will require you to create an interactive solar system using MATLAB. This will require you to explore MATLAB's plotting functionality to a greater extent than we do in this course. Alternatively, you can try to implement this project as a GUI, which may add an additional, optional level of challenge.

   This project has the following deliverable levels:

   C: Running your project creates a display of a star and some static images of planets orbiting around, to some scale. It should resemble the Sol system (our star system), though you can make each orbit the same shape, and project everything as a 2D image.

   B: Add motion to your display by showing the planets moving around their orbitals.

   A: Add the ability for a user to add another planet to the solar system by inputting a distance from the star and size of the planet. It need not interact with any other planets (i.e. collide).

   So long as your project meets those requirements, you will get the corresponding grade. That means a great deal about *how* you accomplish this project is up to you.

3. Speed Reader (Adapted from http://nifty.stanford.edu/2015/posera-speed-reader/)

   This project requires you to create a program that takes two inputs:

   - A string of any length.
   - A number of words per minute.

   Your program will take the string, break it up into individual words, and display it one word at a time at the desired rate. For instance, “Hello World!” at 500 words per minute would display “Hello” for 3/25$^{\text{ths}}$ of a second and then “World!” for 3/25$^{\text{ths}}$ of a second. You’ll be surprised to discover that you can read words that are displayed quite quickly!

   This project has the following deliverable levels:

   C: Your project prints words to the command line at the desired rate. While it is printing words, those words are the ONLY thing in the command line.

   B: Your project prints words to another window besides the command line.

   A: Your project has a graphical interface that will let a user input a string, select a rate, and press a button to display the words of that string at the desired rate.

4. A Game of Sticks (adapted from http://nifty.stanford.edu/2014/laaksonen-vihavainen-game-of-sticks/)

   This project will require you to implement the Game of Sticks in MATLAB. The rules of the game are as follows. There are 2 players and 20 sticks. Players take turns sequentially. On each player’s turn, they must remove between 1 and 3 sticks. The object of the game is to **not** be the player to pick up the last stick.

   This project has the following deliverables:

   C: Your project implements a 2-player version of the game of sticks in the command window.

   B: You have a visualization of how many sticks there are, and an interface with buttons for players to interact with the game.

   A: Your game should have a “play the computer” option. The computer should follow all of the rules of the game, and implement some kind of viable strategy (not just make the same move every time).

5. Create your own project

   Choose your team NOW. Come up with a project idea, and propose it to your recitation instructor *in person* no later than your recitation the week of February 20-24. You should come up with goals for the project, and then use those goals to come up with three deliverable levels:

   C: Things that should be very easy to accomplish for this project.

   B: Things that you need to complete the C level deliverables to accomplish that are somewhat challenging, but still doable.

   A: Things that you need to complete the C and B level deliverables to accomplish that are very challenging, **but you think you can reasonably accomplish by the project deadline of May 3rd.**

   You will need to write up these deliverables and present them to your recitation instructor when you propose the project to them in person. The advantages:

   - Choose your own idea
   - Choose your own team

   The disadvantages:

   - More work up front
   - Less class-wide support (other project ideas may get class-wide updates)