

## **ENGR131 – Final Project**

*Every team member must submit the following:*

Phase	Points	Due at midnight at end of the day (except *)	What to Submit
Choose partner	0	*Due in lab Nov 4 – 7	Included in Part A of Lab #9
Proposal	10	Tue, Nov 18	Complete online form on website
Alpha Version	20	Tue, Dec 2	M-files, data file, any support files
Presentation	extra: 10	*Optional: Tue/Thu, Dec 2/4	Complete online form on Blackboard
Beta Version	30	Thu, Dec 4	M-files, data file, any support files
Final Version	40	Mon, Dec 8	M-files, data file, any support files

**Introduction:** The final project is worth 10% of your grade for the course. You will write an application of your own choosing that uses various techniques learned throughout the semester. You will create a software development team by choosing one or two partners. The project will follow a professional software life cycle ([http://en.wikipedia.org/wiki/Software\\_release\\_life\\_cycle](http://en.wikipedia.org/wiki/Software_release_life_cycle)). Your team will be assigned a TA as your official grader for the entire project and will not necessarily be your regular TA. You will contact this *grader* TA for all help on the project. Your regular TA will still be responsible for lab and homework assignments.

**General Requirements:** Teams must have 2 – 3 people (no more, no less). A team with 3 people will require a *larger* project than what is required for a team of 2 people. You may request to have a partner assigned to you as explained in Lab #9. All content must adhere to a PG-13 rating. Each student must submit regular reports and should contribute the same amount of effort to the project. The scope and difficulty of the overall project should be large enough that an “average” student would require at least 12 person-hours to complete the project. This will be determined by the ENGR 131 teaching staff based on your proposal.

**Technical Requirements:** Below are the *minimum* requirements. You may use code written by someone else (including ENGR 131 assignments), but your team must create new, original code that meets each of these requirements:

1. Create a MATLAB event-driven GUI. The interface may consist of text and/or graphics. You may use GUI controls that were not covered in class. NOTE: Use of the built-in GUIDE development tool is not allowed.
2. You must use all of the following programming techniques:
  - a. A loop.
  - b. A function that receives at least one argument. Built-in functions and callback functions do not count.
  - c. Either an array of cells or an array of structs.
  - d. A data file that is:
    - a. Read using either “fscanf” or “textscan”.
    - b. Modified by the program and rewritten using “fprintf”.
3. A 3-person team is also required to have an additional feature in the program (beyond the above requirements) that a third person could work on. For example, a game might include an AI-player that plays against the user. The “additional” feature must be approved by the assigned grader.

### **Development Phases** (see due dates above)

1. *Proposal (10 pts):* Fill out the online form (on Blackboard) with your ideas for using the required techniques (loop, function, cell/struct array, file I/O). It will be graded based on clarity and completeness. The design of your application can be changed later, but any significant changes should be re-approved by your grader.  
*Every team member must fill out the form.*
2. *Alpha Version (20 pts):* In the software industry, “alpha” indicates the first version that is available for testing within the company. Yours is allowed to be quite worse than a typical alpha and does not need to be

functional. Every team member should submit the same .m files, the data file, and any other support files (e.g. graphics, audio). The following list is the grading rubric:

- (5 points) Progress report: Each team member should write 3 or more sentences explaining, in her/his own words, the successes and difficulties with the program and what every team member worked on. On Blackboard, click on “Type Submission” and type your report there.
- (3 points) Overall program: The GUI should run without errors, but the controls do not need to do anything.
- (3 points) Loop: The header and “end” should be in the correct place, but the loop does not need to run or contain anything in the body.
- (3 points) Function: There should be a function header and a function call, but the function does not need to contain anything.
- (3 points) Cell/struct array: The array should have some data in it, but you don’t have to use it. The data can be “hardcoded” with temporary values.
- (3 points) Data file: The file should have some data in it, but you don’t have to use it. The data can be “hardcoded” with temporary values.

3. *Beta Version (30 pts)*: In the software industry, “beta” indicates a version that is ready to be tested outside of the company. Yours is allowed to be quite worse than a typical beta. The following list is the grading rubric:

- (5 points) Progress report: Each team member should write 3 or more sentences explaining, in her/his own words, the successes and difficulties with the program and what every team member worked on.
- (5 points) Overall program: Should run without errors. At least *one* control should do something.
- (5 points) Loop: The loop should do something, even if it doesn’t work correctly.
- (5 points) Function: The function should be called and do something, even if it doesn’t work correctly.
- (5 points) Cell/struct array: The array should be used in some way by using the values in it.
- (5 points) Data file: Should be reading and writing the file but don’t have to use the values yet.

4. *Final version (40 pts)*: In real life, the final or “release” version is supposed to be flawless. In your program, only the required parts need to be perfect. Flaws in non-required parts will not be penalized, such as mistakes in computations or advanced algorithms. Be sure to simplify the application if your original ideas are not working. *Note: do not use more than 10 MB of support files because these may be difficult for teaching staff to download.* The following list is the grading rubric:

- (5 points) Progress report: Each team member should write 3 or more sentences explaining, in her/his own words, the successes and difficulties with the program and what every team member worked on.
- (7 points) Overall program: Should run without errors. All controls should work.
- (7 points) Loop: Should work correctly.
- (7 points) Function: Should be called and work correctly.
- (7 points) Cell/struct array: Should be used correctly.
- (7 points) Data file: Should be reading (using) and writing (changing) the values in it.

5. *Presentation (extra credit: 10 pts)*: This part is optional and is equivalent to 1% of your grade for the course. Either demonstrate your program or create an advertisement describing its exciting features and showing at least one screen shot (actual or hypothetical). Presentations should be short (2 – 3 minutes is ideal). Creativity and humor are appreciated!

You must register in advance by 10 AM on Tuesday morning using the online form (link is on Blackboard). You are encouraged to use your own computer, but you can also use the computer at the podium which has Matlab installed (bring a flash drive).

You can choose to present on Tuesday or Thursday. Project members from separate lecture sections are allowed to present together (in either lecture period) or separately. Presentations do not require all project members, but only students who participate will get credit.