ENG06 Project #2

Collaboration Policy:

You need to form a team of three to complete one of the project options. You are only allowed to talk and collaborate with members within your team. Team members are expected to equally participate, and collaboratively work towards the completion of the project. ***Other than contacting the teaching assistants for clarification, you may not seek the assistance of other persons.***

Logistics

For this project, you will form a team of three. There are five project options to select from. Each team will select one topic to work on. **The team and topic cannot be changed after 2/24/2012.**

Item 1: Before Thursday 2/23/2012, you must:

1) Form a team of three, (cannot be team of two, or four)
2) Select a team name, and
3) Select three topics (rank the three topics according to your preference).

Submit the above information using the form at SmartSite -> Resources -> "Final Project Team and Topic Sign-up Form". Only one person from each team needs to submit this information. Final team and topic selections will be announced on Friday 2/24/2012.

Item 2: One preliminary (each student will submit one) report will be required. It will be due on Friday 3/2/3012 on SmartSite. This is a 1 to 2 page report where information on you and your team's plan to complete the project is outlined. This report must contain two things, as numbered below:

1) Team's plan: Consider using a table similar to the one below (add more rows as needed). However, your team may use any other method, as long as the information is clear.

Milestone description	Tasks that need to be completed	Responsible team members	Date of completion
GUI and IO interface	Draw grid, user input interface, GUI page transitions	Alice	3/8/2012
…	…	…	…
Integrate all parts	Integrate all components together into a single program	Alice and Bob	3/13/2012
Test and debug program	Test program using different test cases. Fix bugs.	Alice, Bob, and Carl	3/18/2012

2) Your plan (not your team mate's plan): Write a few sentences detailing how you will achieve your assigned milestones and the associated tasks. Consider using tables to help you explain.

Item 3: The completed project will be due on 3/20/2012 at 10AM. Submit the following items onto smartsite:

- A ZIP package containing:
  - All files required to run your program,
  - README file containing Instruction on how to run your program (which file does what)
  - The report (.PDF) – don't forget to include the youtube link in your report.

- E-mail the youtube video link to the TA responsible for your team. E-mail title "ENG6: Final Project Youtube Link - <team name>"

It is you and your team's responsibility to provide a complete set of files to run and test your final program. It will also be important to give clear instructions on how to run your program. This could be done in various ways, depending on the problem that you choose. One good way to document how your program executes is to give an example case and the results.

Grading Criteria:

The projects are open ended. As long as your program can perform the assigned tasks, there will be no correct or incorrect approaches. Certainly there will be more acceptable and attractive solutions, and that will be judged in comparison with competing solutions submitted in the same category. Projects will be judged against the others project in the same group that have been submitted.

The expectation is that each team member must take responsibility for a specific aspect of the project and being able to explain what their contributions to the project have been.

The grade of each team member will be adjusted according to how the project tasks were delegated and who was responsible for what aspects of the project. Each project will allocate at least 10% of the grade to a section that must be included as Appendix A. **Appendix A must contain:**

- A table with the breakdown of the tasks to complete the project, and who was responsible for what part of the project. The intent here is to determine who did what to implement the project. While it is perfectly reasonable that some tasks can be completed jointly, it is unrealistic to claim that that everyone worked together *on all aspects* of the project equally.
- Each member must provide a brief personal summary of what the person's involvement and contributions. Before the project is submitted, the summaries must be provided to all members for review and comment. Appendix A must conclude with the following statement:

"All team members have read the task summaries contained in this report and have been given an opportunity to comment. "

Project Report Requirements:

Each project submission must have a project report that contains any relevant material deemed essential, and it must contain an Appendix A as described previously. The first page of the report must contain the team name and the names of all members. This file must be saved a PDF document.

The report must provide clear explanations on how your program works, and how problems encountered in the project are solved. Use flowcharts or diagrams to help illustrate the program's functionality.

The length of your report is what your team feel is required to fully document and explain your team's work. It is good to keep the report to within a reasonable length, while being concise.

In the event that you have used external resources, you must provide appropriate credit in the project report or in your program code. If it is discovered that you have borrowed or used material from a source that is not credited, it will be consider plagiarism and the case will be turned over to Student Judicial Affairs.

Youtube Video Requirements:

The format of the video is entirely up to your team as long as the following criteria are met:

- Maximum length of the video is 10 minutes
- Each team member must be seen in the video to present their work and contributions
- A clear and easy to follow demonstration that shows the correct functionality of your program.

Use visual aides to help explain your steps. (white board, markers, poster, etc)

Project Option #1: Renewable Energy Planning and Design Tool

For this project, you will be making a program that helps engineers and architects to develop and deploy renewable energy for a small area.

Select an area, or a city from one of the cities in Project 1. Go onto Google maps, and take a properly sized snapshot of the aerial image of that area/city.

- The size of the area/city you choose should be similar to the size of the figure below. The figure shows the length of the North-South street block to be approximately 400 feet, giving the size of each box to be around 20 feet (~ 20 boxes).
- Example: Using a box size of 20 feet x 20 feet, 10,000 boxes in a square grid correspond to an area of 4,000,000 square feet, which is approx 0.14 square miles, or 91.8 acres.

It is up to your team to select the size of the grid and the size of the geographical area. But it is recommended that you select a geographical area around 100 acres, which is 0.15625 square miles, or 4,356,000 square feet, or 2087 feet x 2087 feet, and use a grid size from 10,000 boxes to 40,000 boxes to keep things manageable.

Task 1: Using a GUI to complete the following items:

1) Allow user to select from three different area/cities, and select from two grid sizes. After the user has selected, display the image in MatLab and draw the grid on top of the image.
2) Now they should be able to select the type of solar panel or wind turbine to place (each item here has a color. Assign the color/shading at your own choosing) All types of solar panels and wind turbines are listed in two tables below.
    a. The user first selects an item, then clicks anywhere inside the box on the grid to place the item. Your program should automatically fill in the box with the appropriate color.
3) Continue to perform step 2) as long as the user wishes, while allowing the user to zoom in and out.
4) When the user finishes, your program should allow the user to either:
    a. Save as a picture (a .png picture of the result)
    b. Save as text output (a N by M text file containing the content of each box in the grid. - N and M are the dimension of your grid)
    c. Create BOM: (“Bill Of Materials” contains the quantity of solar panels and wind turbines. Also information in the two tables below)

Task 2: Add a feature into your program that allows the text file specified in 4b) to be loaded into your program (with the proper image). In a way, this feature allows users to save their work and continue working on it later.

Task 3: Using the database from Project 1, estimate the total amount of solar energy available. (don’t need to estimate wind energy)

- The solar data given in Project 1 is kilowatts per square meter. Calculate the total surface area of the solar panels and use the data from Project 1 to calculate the total solar power.

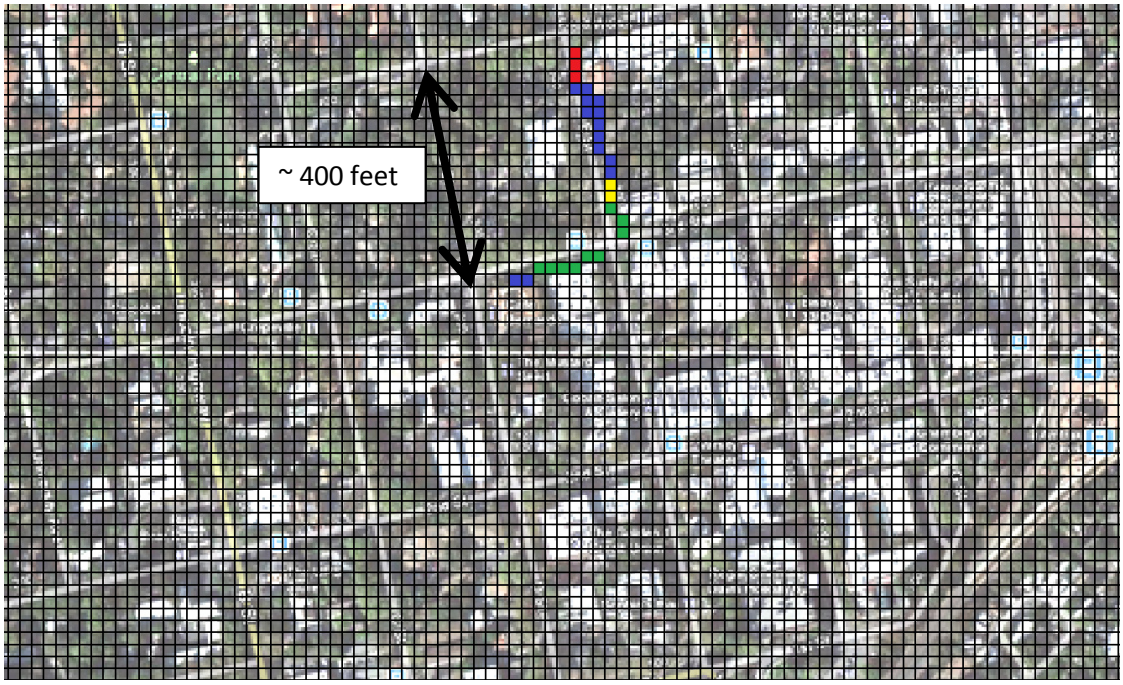
Table of Solar Panels. (Data from www.amazon.com)

Brand	Model	Type	Power (watts)	Dimension (inch x inch x inch)	Weight (lbs)	Price ($)
Sunforce	50048	Amorphous Silicon	15	42.5 x 1.5 x 16	11	279.95
Sunforce	39810	Polycristalline	80	21 x 48 x 2	22	499.95
Instapark	SPCC-5W	Mono-crystalline	5	11 x 8 x 1	2.8	34.95
Instapark	SP-100W	Mono-crystalline	100	45 x 1.5 x 26	21	319.99
Instapark	SPCC-30W	Mono-crystalline	30	21.5 x 1.1 x 17.2	7.2	114.70
Instapark	SP-10W	Mono-crystalline	10	14 x 11 x 1	2.8	39.95
Ramsond	100SP	Mono-crystalline	100	47 x 1.5 x 21.8	12	245.99
Epcom	WK50-12	Polycrystalline	50	32 x 22 x 1.4	12	99.99
Sun Power	E18	Mono-crystalline	400	41.18 x 81.36 x 2.13	56	249.5
Sun Power	T5	Mono-crystalline	320	43.06 x 75.13 x 8.37	47	199.99

Table of Wind Generators/Turbines. (Data from www.amazon.com)

Brand	Model	Type	Power (watts)	Diameter (feet)	Price ($)
Windmax	HY 1000-5	Wind Generator	1000	15	999.99
Windmax	HY400	Wind Generator	500	13	686.40
GudCraft	WG400	Wind Generator	400	13	399.00
GudCraft	WG700	Wind Generator	700	13	449.00
All Power America	APWT400A	Wind Generator	400	10	476.93
Sunforce	45444	Wind Turbine	600	10	749.99
Sunforce	44444	Wind Generator	400	10	474.34
WindyNation	WCK-750	Wind Turbine	750	15	999.98

Figure of Davis zoomed in, overlay with grid. Filled boxes indicate solar panels. Colors indicate solar panel/wind generator type.



Project Option #2: Solar Survey

Use the Arduino hardware module to perform a solar survey of an area/city of your choosing.

Your team will select the area/city to survey by taking enough data to fully cover the area you've selected. Later you will use this data to create the solar profile of the area you have surveyed.

Task 1: To help the survey process, create a GUI that:

- Displays a map of the area (screenshot from Google maps is OK)
- Allow the user to indicate their location, and the time/date. Store this information.
- After the user has finished recording the location, and time/date, the Arduino hardware should automatically start to perform the trials. Allow the user to reposition the hardware module in between trials. See below for what should be recorded in each trial.

Survey Procedure:

- The number of Arduino hardware units you use will depend on the number of teams signed-up for this project option. This will be announced by the TA.

- Minimum of 10 locations per team member (you will need more if the area is large), and 10 trials per location.
  - Each location should be fairly far apart
    - Example: 10 evenly spaced locations in-between highway 113 (WEST), Mace Blvd (EAST), Covell Blvd (NORTH), and highway 80 (SOUTH) is a good example.
  - 250 data points per trial.
  - Watch out for the orientation of the hardware module.
  - Survey the location at three different times of the day – on the same day.
    - Preferably once in the morning, once in afternoon, and once near sunset.

  - For each trial, perform:
    - Load sweep: sweep the variable resistor 0 from resistance code 0 to 255, while keeping variable resistor 1 at resistance code of 128. Record the voltage at each resistance code.
    - Open circuit voltage sweep: Remove the open circuit jumper. Set both variable resistors to resistance code of 255. Record 256 data points.

Task 2: Finally, after all the data has been collected, write a GUI that displays a color plot on top of the map of the area your team has surveyed. Your GUI should allow the user to:

- Select whether to display the plot using
  - Maximum Power Point: calculate using the data from "Load sweep" trial. For each data value, calculate the power by squaring the data value, then dividing the result by the corresponding resistance value. Calculate the power of all data points. The maximum power point is simply the largest power value.

    Maximum power point = max(data point^2/load resistance)
  - Open Circuit Voltage

Note: For both plot methods, you will need to interpolate the data at locations where data is not available.

Project Option #3: Carbon Dioxide Emission from Traffic

In this project, you will be creating an object oriented traffic simulator, and calculate the CO2 (Carbon Dioxide) emission, and the rate of fuel consumption from the traffic. You should base your approach on the bouncing ball simulator discussed during the object oriented lecture. Edit the greenball.m and greenball2.m files to instead simulate traffic on a highway with five lanes:

Task: Create a GUI that shows the traffic simulator, together with controls 'STOP' and 'START', which allows the user to stop and start the simulator. Display the instantaneous CO2 emission (pounds per hour) and fuel consumption (gallons per hour) from the traffic. Follow the guidelines below.

1. There should be 20 vehicles at a time, always moving horizontally from left to right on the screen. (The 20 vehicles may include cars that are "off screen" (see below). The vehicles should be displayed as rectangles, with colors of your choosing. If you can, make the rectangles look a bit more like cars.

2. When a vehicle disappears off the right side of the screen, it should wait a random time and then reappear on the left side of the screen.

3. There should be 5 lanes on the screen; cars should randomly consider changing lanes if it is safe to do so. You should use yellow lines to denote the lanes.

4. The vehicles should not crash into the cars in front of them; they must either slow down or change lanes to avoid collisions.

5. The CO2 emission and fuel consumption from the traffic can be calculated according to information below:

Assume there are only two types of vehicles as shown on the table below on the left. Randomly assign each vehicle a type at the start of the simulation. The table at the right shows how the fuel economy degrades with increasing speed. Use these two tables to calculate and display:

1) Instantaneous carbon emission (unit: pounds per hour) from the 20 cars, and
    a. How much CO2 is being emitted per hour (by the 20 cars)?
2) Instantaneous total rate of fuel consumption (unit: gallons per hour) from the 20 cars.
    a. How many gallons of gasoline are being used per hour (by the 20 cars)?

Vehicle Type	**2002 HONDA NSX**	**2010 VOLKSWAGEN Golf Plus**
Carbon Dioxide Emission	291 grams per kilometer	152 grams per kilometer
Combined Fuel Economy	18.98 miles per gallon	36.22 miles per gallon

Speed (mph)	**Fuel Economy Decrease**
<= 59	0% less efficient
60	3% less efficient
65	8% less efficient
70	17% less efficient
75	23% less efficient
80	28% less efficient
>=85	35% less efficient

Project Option #4: Ngram Viewer

In this project, you will create an Ngram Viewer, similar to the Google Book's Ngram Viewer here:
http://books.google.com/ngrams/

N-gram refers to the number of words in a phrase. The word "speech" is a 1-gram. The phrase "President Polk" is a 2-gram. The question "is anybody there?" is a 3-gram (ignoring the question mark).

The database of text you will be operating on is the State of the Union Addresses from various presidents. The text to the addresses can be found at the link below.
http://www.presidency.ucsb.edu/sou.php

Task: Create a GUI that allows a user to:
- Enter a word, a phrase, or a regular expression
- Specify to search the speech(es) of only one president, or all presidents,

Based on the user inputs above, your GUI will then search and display the total number of times the search term appears in the speech. If the user specified to search all presidents, your program should display a histogram, where each bin of the histogram is a president, and the y axis is the number of times the search word/phrase/reg. exp. is used in the speech(es) of the corresponding president. The histogram should be chronological.

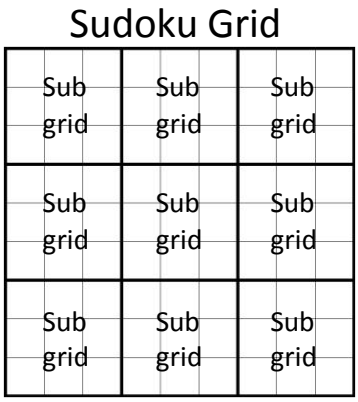
Hint: use the **urlread** function along with regular expressions, to read website content efficiently.

Things to consider:

- Should your program be able to work without internet?
    - Requires internet: your program don't download the entire database. Your program simply queries the web page for information after the user submits the search term.
    - Does not require internet: your program should download all of the data from the web pages prior to processing the user's search term. Be careful querying the web page too many times in a short period. This might overload the server.

Project Option #5: Sudoku

In this project option, you will create a Sudoku (9x9) game GUI. To learn about Sudoku, see: http://en.wikipedia.org/wiki/Sudoku



Note: The Sudoku grid contains 9 sub grids. There are 9 sub grids and each sub grid contains 9 boxes.

Task: Create a GUI that

1) randomly generates a Sudoku puzzle, and
2) allows the user to solve the puzzle.

One way to generate a Sudoku puzzle is:

1) Generate a solved puzzle (where all boxes are filled in) by
   a. Go through an empty grid, randomly filling each box with a number, while adhering to the rules of the game.
      Example: starting with a corner, place a random number (between 1 to 9) there. Move onto the adjacent box. Randomly place a number, without violating the rules. Repeat.
   b. Randomly place the same number into each of the sub grids, while adhering to the rules of the game. Example: place all the 8's into the 9 sub grids, without violating the rules. Then move onto another number. Repeat.
2) Now you should have a completed Soudoku puzzle. You can now obtain an unsolved puzzle by randomly taking away numbers, one by one.
   a. The number of numbers that you take away is arbitrary. The more numbers you remove, the more difficult the resulting puzzle becomes. Each sub grid should have at least one of its numbers removed.
3) After taking away a number, you need to make sure the puzzle is solvable.
   Here is a Sudoku solver available online:
   http://www.mathworks.com/matlabcentral/fileexchange/8083
4) You now have a puzzle to solve!