Assignment HW21
Due date (on or before): Nov-24-2014 (also announced in class)

Assignment

The main goal of this assignment is to get you to have a **working skeleton** of your project program. It doesn't need to have all the fancy graphics, and even not all the logic and cases, but it should be a compiling program, that gets from the beginning to the end of your project.

For example, for a "hangman" game project:

a. The word selected by the computer can be randomly selected between two possibilities (or even hard-coded one option!).
b. The word is printed as all 'dashes' (or underscores).
c. The user guesses a letter, gets a reply (whether the letter is there or not)
    a. No need for graphics.
    b. Counting how many 'tries' is useful.
d. The word is printed accordingly: Guessed letters are visible, and non guessed ones are as 'dashes' (or underscores).
e. After certain amount of guesses, the game ends in a loss.
f. If the user guesses the word, she/he won.

What do I need to submit?
Specifically, for this assignment you will need to submit:

1. **Word document :** The core of your 'final report'.
    a. Include the basics: Title, what is the project, and maybe even copy the flow-chart from your slides!
2. **The important part:** In this word document, include a short 'annotated' flow of the program, using screen shots from the running program. Something like a story-board.

***** Again, the goal is to show you have program running, that implements the whole start-to-finish project.**

=== Reminder===

Final Project

The final project comprises 15% of your final grade: Please pay attention to the below !

More importantly than the grade itself, this is your opportunity to demonstrate independent learning, creativity, and have fun with 'C'. Use this opportunity wisely!

You will have about 6 weeks (in words: SIX weeks) to complete the project. We will also spend time in labs and lectures to discuss related stuff.

Below are the technical details. Keep in mind: The goal is to learn while being engaged. Do a project you would like to see working and have fun doing it !!

Details

1. The project can be done in pairs or individually.
    a. If done as a pair, please specify who is responsible to what part (e.g., graphics, program, file I/O, etc).
2. Every **Monday** we will have a short debrief by each team on Project status.
3. Final results of the project are:
    a. Working program.
    b. Presentation (less than 10 minutes) with slides on the project.
    c. Project report (including the program printed as appendix).
4. Final presentations are on the last week of classes: **Dec-15** , during class times.
5. Project should include:
    a. Console-graphics – Preferably translates as well into file (see below item).
    b. Can demonstrates I/O – Keyboard / Screen ; Input from files, output into files.
    c. Good to have: Demonstrates Arrays and pointers – Storing, handling.
    d. (extra) Demo-mode (Auto run).

6. All project materials (presentation-slides, report (hard copy), programs) should be submitted no later than **Dec-15, Monday, at Midnight**.

Rubric used:

Date: Mar-03-2014
PLO3:
Analyze engineering problems and resolve them using appropriate design steps and processes.

Notes:

1. In this PLO, the student is given an open-ended problem specification, and a large part of the process is finding the right questions to be answered, and methodology of arriving and pursuing those.
2. This PLO refers to BOTH engineering-problems AND art-productions.
3. NOT all items will be relevant for each specific assignment.

	Initial (Comprehension)	Emerging (Application)	Developed –I (Analysis)	Developed-II (Synthesis/Evaluation)
Problem understanding	Discusses the importance of developing a clear understanding of a problem.	Applies best practices to rephrase the problem in a simpler and more direct manner.	Analyses various aspects of the task and draws on previous experience to attain a clear understanding and formulation of the problem.	Synthesizes the components of the task and suggests possible extensions to the task.
Alternative approaches for solution	Describes why approaching a solution from multiple vantage points can yield tradeoffs in solution.	Applies various approaches to finding a solution.	Analyses different approaches to a solution to determine viability and tradeoffs.	Evaluates multiple approaches to a solution and selects the most viable option.
Reduction to specifications	Describes the importance of adhering to specification and constraints for a given project.	Adheres to specified specifications and constraints when working on a specific project.	Analyses specifications and constraints to determine the most effective way to complete a given project.	Creates new specifications that expand the scope of the project.
Iterative process	Able to discuss the importance of applying an iterative (refining) process in solving engineering problems.	Applies iterative process in solving a problem.	Analyzes the progression of the iterative process to determine which elements need more refinement.	Synthesize previous iterations to create the most appropriate final product/solution.

Example project ideas:

- Sudoku: 3x3, (extra credit: 4x4).
- Towers of Hanoi
- Connect-4
- Crossword puzzle
- Checkers
- Knights tour
- 8 Queens
- Sorting (1-dimensional!!) – demonstrate by bar-graph for example.
- Hirogliphs
- CandyCrush
- < Your idea here >

Some example and notes:

Levels of Graphics

Towers of Hanoi:

0th level (Horizontal):

Peg A: 5 4 3
Peg B: 2
Peg C: 1

1st level (Vertical):

```
 3
 4
 5    2    1
Peg   Peg   Peg
 A    B    C
```

2nd level (graphic):

```
 ======
 ========
==========    ====    ==
```

Higher levels:
Better graphics, showing alternatives as the process evolves, counter of 'moves', explaining process, time to finish/to do, etc.

=====

Soduku:

Many more options for graphics: what are possibilities, what is checking now, etc.

=====

Knight Moves

Example of the problem statement (borrowed shamelessly verbatim from RJ Mical "Weekly Tickler")
" The classic chess question about knight moves asks whether you can move a knight around an 8x8 chess board in a way where the knight visits every square once and only once. The answer is Yes, there are many ways to accomplish this.

But it's a long and tedious tour on an 8x8 board.

Here are some easier questions to ponder. For all of these questions, the knight starts in the top-left corner of the board.

* Can the knight visit every square of a 3x3 board? Think about it for a second, maybe do a drawing, then read on.
  - The answer is No, it's not possible to visit every square because there is no way to reach the middle square.

* Can a knight visit every square of a 5x5 board? Yes. Can you find a way? One awesome spiral solution is given below. I've been checking out the 5x5 space quite a bit. Fascinating!

* How about a 4x4 board? There may be a way, but I never found one, and I've mostly convinced myself that a solution doesn't exist. I can solve 4x3, but not 4x4. Can you do 4x3?

Here's today's reachable-but-hard challenge: What if we limit the area to the first three rows of a chess board? Can you find a way to move the knight to every position of the first three rows? In other words, can you visit every position of an 8x3 board?

I don't know how many answers there are to this question. I suspect that there are not too many. The path I found is almost a fractal, with beauty in the way it folds!

Bonus question: Surely the answer set grows exponentially in size as the dimensions increase. What algorithm would give an approximation of the number of possible solutions, based on any starting conditions? Can such an algorithm be derived?

“

=== END ===