CPS 111 Final Project

Important Dates:

November 14th (Tuesday) 11:59 PM: Proposal due (submit on-line as PDF)
November 20th (Monday): Last day for project meetings
Week of December 4th: In class presentations and Q/A
December 12th (Tuesday) 4:30 PM: Project due, submitted to Canvas (**no late submissions!**)

The final project is an opportunity to spend time on a project that you will enjoy working on, and that has some significance to you. You will work in a team of 2 on a project of your choosing. You will meet with me as a group to discuss your project and make sure it is of the appropriate scope (see details on this meeting below). This project should be different from any of the homework assignments, lab exercises, examples in the book, and other group projects. *The complexity of your program should be roughly on par with 1.5 homework assignments.* At any point during this process I welcome you to consult with me for feedback. Think of topics that make a connection with your other courses and/or topics that particularly interest you. Creativity, whimsy, humor, eccentricity and ingenuity are all greatly encouraged!

Each project should include the following, unless approved otherwise by me: user interaction and/or file input/output; loops and conditionals; use of functions and classes to modularize the program; clear, readable, well-written code; and good comments and documentation. User interaction should be clear, friendly and error-proof to the extent possible. Each project must also be accompanied by a document describing the design (see below for details).

Here are the beginnings of some ideas that might spark your imagination:

- Continuation/elaboration of an idea from class: analyzing texts for linguistics information, graphics using turtle, animation, simulation, etc.
- Image processing application, that provides a menu of image processing options.
- Steganography: Hide a text file or an image *inside* another image!
- Program to test arithmetic for children: program generates math problems, kids are given several tries (with encouragement), and then the correct answer is explained (if necessary). Starts with simple questions and gets harder as the questions are answered.
- Games: card games, board games, simple / classic arcade games etc.
- Inventory: manages information such as for collections of books, CDs, etc.
- Word processing: this could go in many directions such as mad libs, formatting, spell checking, counting words, etc.
- Calendar program: keeps track of appointments, checks for conflicts, etc.
- Banking program or finance program
- Mathematics program: handles functions, does numerical integration, etc.
- Physics problems that can be implemented on the computer (e.g. simulation of free fall)

You may find that learning to use a new module can really expand the possibilities for what you can accomplish. A (non-exhaustive) list of some modules are listed at the end of this document.

The final project is worth 10 points of your final grade, distributed as follows:

Proposal and meeting (1 point): Your proposal document should be uploaded as a PDF to Canvas. Your document should be around 1 page long (no more than 2). It should list the members of your team and carefully describe the problem you will be working on (think about writing the homework assignment whose solution is your project). You should specify what the goals of your project are, what the anticipated challenges are, and what your plan of attack will be. Think about setting key goal posts and how long you expect it will take to reach them (implementing major pieces of functionality, testing and documentation, etc.).

You should come *as a group* to the meeting prepared to discuss your project and your plan of attack. The main purpose of the meeting will be to determine whether the project is of a reasonable scope. If the project is too simple, then I may ask you to elaborate on it. If the project seems too ambitious, I may help you set more modest goals.

Project Writeup (1.5 points): This writeup should be submitted, as a PDF file, along with your project code at the end of the project. It must contain:

- Team members' names
- Clear statement of the project goals
- Description of the major components of the design, particularly the functions, classes, and methods (including parameters and return type)
- Description of how to run/use your project
- Kinds of testing you did and modifications you would make for a next version
- Any sources you may have used for information and any external tools you may have used

Project Code (6 points): Your project code will be graded on the following criteria:

- Functionality: works correctly according to design
- Program design: no redundant code, decomposed into logical classes and functions
- Code style: relevant and meaningful class, variable, and function names
- Interface: user-friendly interaction, robust to unexpected input
- Documentation/Clarity: this involves both commenting your code so it is clear what everything does and writing clear, readable code to begin with

Final Presentation and Q/A (1.5 points): You will briefly demo your project to the class, discuss your design decisions and challenges you faced, and answer questions. Note that the presentations are scheduled *before* the final due date. Your project should be very near to completion by the demo date. The remaining days should be spent on polish, testing, debugging, and documentation, *not* on implementing core functionality.

Bonus (1 point): If your project goes beyond the basic requirements and shows significant additional complexity and/or imagination, you will receive 1 bonus point.

The completed project is due December 12th (Tuesday) at 4:30 PM. NO LATE SUBMISSIONS WILL BE ACCEPTED. You are *highly* encouraged to submit *before* this deadline to ensure that you get credit for your work! Double check your submitted files and make sure that all are complete and the correct versions. Remember that all necessary files, including the project writeup, any non-standard Python modules, and any media and text files your program needs, should be included in your submission.

Python Modules

Module name	Description	Installed on machines
graphics.py	for graphical applications	No
pygame	for games	Yes
visual python	3D graphics/animation	No
tkinter	Creating GUIs	Yes
scipy	Scientific/Math	Yes
numpy	Often used with scipy	Yes
matplotlib	Graphing/Math	Yes
pyaudio	Audio processing	No
pyOpenGL	Computer Graphics	No