Python & Perl: Final Project Proposal 5 points

Learning Objectives

- Technical Writing
- Project Specification, Scheduling, and Planning

This class is great in that you get to work with two languages: Python and Perl. This class is bad, because you have to work with two languages. Time limits the scope of the assignments, and just when we get to a point where we can do something complex with one, we switch back to the starting point with the other. This project will give you a chance to do something a little more sophisticated with one of the languages. The project's concept, design, and implementation are all up to you. This is an individual project meaning all work is to be your own. You can do your project in either Python or Perl.

The goal of this proposal is for both you and I to have a clear understanding of what you are going to develop. This will also help us identify potential problems now instead of one day before the project deadline. I also want to make sure that you neither do too much or too little. In the end, I want you to have a successful project and have some fun with it.

The best project idea is one you find interesting. Scratch your itch. Here are some suggestions:

- A game. This is always the most popular option.
- Extend an existing application. GIMP, an open-source image editor, allows you to write plug-ins in both Perl and Python.
- Implement something which interests you from another class. For example, maybe you have taken an AI class and would like to use that knowledge to implement the AI of a robot.
- Implement something needed at your job. A two-for-one deal. Get class credit, and get paid.
- Write a web-based application. Python, for example, has 3rd party tools such as Django which lets you develop database-driven websites.

Your proposal should describe your application in as much detail as possible. This will force you to plan and design your system ahead of when you are actually coding it. Ideally, your proposal should be written in such a way that you could give it to another developer with an appropriate skill level and they could build your application using your proposal as a guide to the design.

Your proposal should contain, at a minimum, the following information:

1. **Basic Description (2 paragraphs):** A basic description of what the application will do. This should be a short description describing the basic features and a typical use case.

2. **Target User (1 paragraph):** Describe your target user who would normally be using this application.

3. **Target Operating System (at most 1 paragraph):** Are you writing this for a specific platform, or do you want to make your code portable across different operating systems.

4. **3rd Party Libraries, API's, data sources (2-3 sentences per resource)**: Describe any 3rd party tools or resources you want to use. If a tool is available online, give a working URL.

5. **Risks:** List the parts of the project with which you have concerns and that may cause you potential coding or scheduling problems. All projects have risks associated with them. The trick is to manage the risks and address them earlier rather than later.
6. **Skills:** Brief description of your skills relevant to the project.

7. **Schedule:** A weekly schedule of achievable targets. Design a schedule for the project denoting what you plan to complete at the end of each week for 4 weeks.

For example, you want to develop a 3D game. But you mention that a) you have never developed a 3D game in any language and b) will be using Panda3D, a 3d API for Python, but need to learn it. Those are two big risks, and I, depending on your skill level (item 6 above), may recommend that you decide on a different project.

Another example would be you want to develop a GUI-Based FTP client. In your risks, you mention you have done some user interface work in C# and know the FTP protocol. You say that you would like to learn PyGtk, a GUI API, and you think, based on your experience, that you will be able to do it. That is a reasonable risk, and, as long as you don't get too fancy with your design, I would probably say go for it. If you do not mention any risks, then you have not thought deeply enough about the project and your proposal may be returned for further modification. I may also suggest that some issues be resolved in a brief face-to-face meeting.