Background

One of the most powerful and versatile forms of content management system today is the wiki, a collaborative easy-to-edit knowledge base consisting of many interconnected pages. Wikis have been used to great success to store encyclopedic and dictionary knowledge as evidenced by Wikipedia and Wiktionary. For my project I will be applying the concept of a wiki to create a knowledge base for computer programming. This will consist of creating a website where the individual pages (i.e. the building blocks of knowledge base) are functions, classes, or modules, and the relationships between pages (i.e. the associations in the knowledge base) are function calls or references to the classes or modules. Users will be able to easily edit code or create new pages as well as view relationships in the code knowledge base graphically.

WikiCode

There are several wikis on the web currently which are meant to be repositories of computer programming knowledge; however, these websites (e.g. codecodex.com, rosettacode.org, and en.literateprograms.org) are meant primarily as teaching tools. They feature implementations of various algorithms or common programming tasks written in several programming languages along with explanations of how these implementations work. Essentially, these wikis are just domain specific encyclopedias about programming with various code examples. With WikiCode, however, the knowledge contained within the wiki is actual code, with each page corresponding to a single function, class, or module. Users will be able to
easily edit the code in the wiki and traverse it in a novel way by clicking identifiers contained on one page to link to other pages. WikiCode will also be able to present the relationships in the code in a graphical format, similar in spirit to a call graph where all functions are shown as nodes and with caller-callee relationships drawn as directed edges. Since the code stored on the site will not actually be executed on the server, we will be limited to showing only those relationships which can be deduced from static analysis; however, even with this limitation I think such an interface will still provide users with an interesting way to look at source code.

For this project I will restrict the wiki to hosting Python source code because Python is a modern scripting language especially popular in the open source community. Each page of the wiki will also contain editable documentation for that page’s function, class, or module. This will augment the knowledge base so that the code available is easy to navigate and understand.

**Technology**

In order to build WikiCode, I plan to use the Python web framework Django. Django is a “batteries-included” framework which will simplify handling http requests and rendering html templates. For storage of the wiki data, I plan to hold most of the relational/linking information between the pages in an RDBMS. However, rather than hold the code and revisions in an RDBMS, I will use the version control system git. There are several reasons for this (1) git is already equipped to handle file diffs and code revisioning, (2) I hope to enable users of WikiCode to use git to push and pull code and changes as well as being able to use a simple web interface for edits, (3) git is a mature version control system and many programmers are already familiar with using it. To actually create and modify the git repositories used for the backend of WikiCode I will use GitPython which is a well-documented library with a great deal of functionality.
Additionally, to gather data about the relationships between pages in the wiki I will use a Python library called Jedi. Jedi is a smart autocompletion library which also provides an API for finding definitions of functions, classes, and modules in source files. This will enable me to analyze code so that WikiCode can infer relationships and create links for users to click on. Lastly, users will be able to view a graph visualizing these relationships which will be drawn by the Javascript library arbor.js.

**Deliverables**

Here are the goals of the WikiCode project along with a rough time-line of the when I plan to complete these goals:

1. **Mid-February:** Create a website that allows users to add and edit pages containing Python code.

2. **Early March:** Ensure that the code on the site is stored in such a way that it can be analyzed with the Jedi library.

3. **Late March:** Enable the site to be able to find the relationships in the code (e.g. if one function makes a call to another function, create a hyperlink in the wiki). Implement a method of storing these relationships.

4. **Mid-April:** Complete hyperlinking functionality so that users can traverse the wiki. Implement a system so that code can be documented.

5. **Late April:** Add support for drawing a graph representation of the wiki.

6. **April 30th:** Fix remaining issues and submit the final abstract and project report.