The Brahma Project Challenge

Design Documentation

Michael Eaton, Victoria Zheng

## Module Decomposition Diagram & Dependency Diagram

# 

The design for the Brahma Project consists of three distinct layers:  The UI layer, the Control layer, and the Foundation layer.  The UI layer is self-explanatory -- it contains the logic for defining the behavior of the UI.  It speaks with the Control layer, which dictates the underlying logic of the UI layer, and manages the lifecycle of Brahma plugins.  The Control layer interacts with the Foundation layer, which details the system’s interactions with files stored on the system, such as detecting and accessing available plugins, and reading configuration data from the available configuration file.  
  
Inside the Control layer there is the Controller and the Launcher.  The Controller contains the high level logic, which interprets input from the UI, and translates foundational data for display on the UI and usage in the Launcher.  The Launcher receives commands from the controller and launches the appropriate plugins, managing their individual threads and lifecycles.  
  
The Foundational layer contains logic for detecting and accessing plugins -- the Plugin Manager. It also contains logic for reading and writing to the program configuration, the Configuration module.

## Class Diagram

Methods:

File Manager (static):  
            getPluginList();  
            getPlugin();  
Directory Watcher:  
            createFileEvent();  
IPlugin:  
 getPluginUI();  
Launcher:  
 launchPlugin();  
 terminatePlugin();  
 getActivePlugins();  
ConfigurationManager (static):  
 getConfiguration();  
 saveConfiguration();  
IConfiguration:  
 /\* Various needed get-set X methods \*/  
DependencyHandler:  
 readDependencies();

areDependenciesResolved();

addPluginToDependencyTree();



## Interaction Diagram

Update the list of plugin



Launch Plugin

