Brainstorm: A More Efficient Mind Controlled Keyboard

Jett Hays
Human Computer Interaction Department
Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA
jehays@andrew.cmu.edu

Languages: C# and Python
Application: Windows form
Repository: https://github.com/jettblu/brainStorm
Architecture: Model View Controller (Event Driven Data Feed)
Package Dependencies: Accord.Net, Math.Net

Is this for you?
This paper is for developers trying to customize Brainstorm or for anyone trying to understand how a complex system can be constructed from simple principles.

Philosophy

Note: <u>Software</u> *is the general architecture of Brainstorm's programs, while* <u>code</u> *is the manifestation of that architecture.*

With a prototype like Brainstorm, it is impossible to separate great research from great software design. While Brainstorm is still a work-in-progress, it *works* and represents *progress*, because it follows the design principles below.

Design Principles

1. *User first, Code Second*

Brainstorm is a machine whose value stems from replacing user burden with technical solutions. The mechanics of *how* Brainstorm transforms brainwaves into language are unimportant for the user. All the user cares about is *if* Brainstorm works and *if* it works well. The software behind Brainstorm ensures both *ifs* are met with consistent performance and clear error handling

2. *Make Assumptions Few and Weak*

Strong assumptions = weak programs. Weak assumptions = strong programs. As a platform for both users and developers, Brainstorm tries to minimize the scope and strength of its assumptions. This allows for a better user experience and enables developers to customize algorithms with ease.

3. *Easy to Understand Code > Slightly More Efficient Code*

Don't make simple things complex. Variable names are descriptive and reusable computations are relegated to helper functions. Breaking a complex chain of logic into understandable steps makes Brainstorm easier to update and maintain.

4. *Leave a trail*

Each section of code has in line documentation that details *what* and *how* it's doing. By leaving a trail through complex code structures, Brainstorm allows developers to locate and understand relevant algorithms.

Implementation

Brainstorm is woven together by the robust seams of Model View Controller architecture. MVC architecture allows for an iterative testing process in which various aspects of the software can be evaluated and subsequently modified based on performance without demolishing the architecture as a whole. By separating concerns, Brainstorm provides developers with a versatile system that can be adapted to fit future virtual keyboard research. An event driven data feed supplies the model with a continuous stream of information that is used to update the view and change system state.