

Dasher Game Mode

Description

Game mode is a module written for Dasher whose intention is to provide new users with a way to learn how to effectively use the application. Dasher itself can be difficult to understand at first glance. It can be even more difficult to internalize the mechanics behind movement through words. Game Mode, at its most basic level, provides visual guides for entering a target string of text. It is modeled after traditional typing tutorials that give you a passage to type and helpful responses if you type the letters correctly or make a mistake. Please note that this has only been tested on Gtk systems, so far.

Architecture

Game Mode has been designed to be as modular and flexible as possible from the ground-up. It also has been designed to require as little change on the part of the core code-base as possible, keeping in line with the plug-in mentality.

The core class at work here is the *GameModule* class. This is a class that inherits from *DasherModule*, which means it is a *DasherComponent*. Being a Component, it takes advantage of the completely generic event system to interact with Dasher. It subscribes to and responds to Text Draw events and Edit events. It uses the Edit events to know whether a user has typed the correct character or not and respond accordingly. It uses the Text Draw events to acquire the position of the labels on nodes to draw a custom crosshair over the next letter a user must type.

To determine what words a user types, Game Module employs the help of a Word Generator. Word generators define an interface that allows clients to read words from a source defined by the concrete implementation. For example, there exists a *FileWordGenerator* class that inherits from *WordGenerator*. Its purpose is to generate words by simply reading from a file. Game Module calls *NextWord()* to get the next word from the generator.

Game Module defines the block of words that are displayed as targets for the user as “chunks”. The size of these chunks, in # of words, is defined as a constant member. So, an example use case for Game Module would be the generation of a chunk. If the chunk size is three, *FileWordGenerator::NextWord()* would be called three times, appending the results to the member of Game Module that holds the target chunk, *m_sTargetString*. The benefit here is that Game Module no longer cares how the words are obtained. Any implementation of the *WordGenerator* interface can be used without issue.

In terms of how Game Module actually draws to the canvas, this should be kept platform independent and should make use of the Dasher drawing functions that have been implemented on all platforms. The drawing happens in *DecorateView()*. We chose this name because it's very descriptive, but it's important to realize that though Game Module and *InputFilters* share a method in name, they are NOT at all equivalent. Game Mode does not (and should not) modify the way a user interacts with Dasher in terms of input.

Usage

To use Game Mode one currently must enable it in Gconf. Adding an option in the preferences window is a to-do. After Game Mode is enabled, you should see a crosshair pointing to the next letter and a few words that you're supposed to type.

Issues

Currently, there are a few issues to work on:

- If a letter one must type does not appear on screen, the crosshair should disappear. Currently it remains visible, but does not move.
- Target text is NOT styled nicely at all.
- No alerts for completing the target string or target chunk yet.