



## Assignment 2

### Overview

This project is designed to give you practice using various layout containers, buttons, the text entry widget, and UTF8-encoded characters. You will create a simple utility that lets a user create strings of characters that include characters not found on the keyboard, within a text entry widget. By making the strings selectable, they can then be copied to the clipboard for use in other text documents.

The application should look something like the one in Figure 1. Details follow.

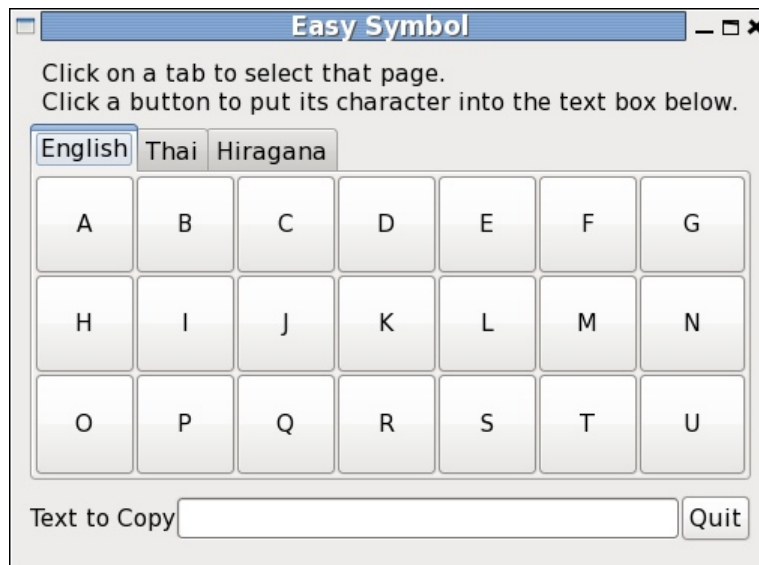


Figure 1: Character Selection Tool

### Functional Requirements

1. The top-level window must contain a GTK+ notebook widget containing *at least three tabs*, a text entry widget that can store at least 200 characters, and a QUIT button. The text entry widget should be labeled, either directly to the left, as illustrated above, or directly above and left-aligned.
2. Each tab of the notebook should be labeled across the top, as shown in Figure 1, and should contain the exact same number of buttons as all other tabs, arranged in the same shape grid. Each tab must contain at least 20 buttons. The grid must have at least 3 rows and at least 3 columns. The buttons must fill the grid completely, so, for example, if you have 24 buttons, they should be arranged in either a 3 x 8, 4 x 6, 6 x 4, or 8 x 3 grid. There should not be any empty cells in the grid.
3. Each tab of the notebook must use characters from a single alphabet, such as Roman, Greek, Cyrillic, Coptic, Hebrew, Arabic, or Hiragana, to name a few, and must contain only characters that are not on a standard U.S. keyboard. One or more tabs can contain symbols such as mathematical symbols. Although the symbols do not have to be arranged in any specific order, your user will be unhappy if they are not in the alphabetical order of the languages using that character set. Each button should be labeled by a unique character.



4. When the user single-clicks a button, the character in its label should be copied to the text entry box at the bottom of the window to the right of the current cursor position. The text box should be editable, so that the user can insert or delete characters by typing on the keyboard or by pasting from another application. The text should also be selectable, so that the user can copy any portion of it to the clipboard using Ctrl-C, in order to paste it into other applications.
5. When the user clicks the QUIT button the application should terminate.
6. The title should be something appropriate for the application.
7. The top-level window *should not be resizable*. It should be a fixed size large enough to make the application easy to use, meaning that the buttons are not too small or too large, or too tall or wide. The text entry box's visible area should be large enough to display at least 20 characters. In Figure 1, the text entry is the minimum acceptable size. Remember that it will scroll as text is entered into it.
8. GTK+ does some things automatically that make this design easy to use. If the user presses an arrow key on the keyboard, the focus will advance to an adjacent widget. If focus is on a button, pressing the Enter key will cause it to emit the "clicked" signal on that button. This means that someone can use this tool without having to move the mouse very much.

## Other Requirements

Your source code must satisfy the requirements stated in the *Programming Rules* document on the course website. Please review those rules if you are not familiar with them. It will be graded using the following rubric:

Component	Percent of Grade
Correctness (meeting the functional requirements)	60
Documentation	15
Structure (modularity and design)	15
Program Style	10

## Submission

If the program consists of more than one source code file, create a directory named `hwk2_username`, with permission `0700`, and put all of your source code files and a makefile into it (and nothing else!) If it is a single source code file, name it `hwk2_username.c` or `hwk2_username.cc`, where *username* is the username you have on our UNIX system, and the extensions are for C and C++ respectively, with permission `0600`. Put the directory or the program into the directory

```
/data/biocs/b/student.accounts/cs493.70/projects/project2
```

on our file system.