Lecture 3.
GUI Programming – part 1: GTK

Guoyong Shi, PhD
shiguoyong@ic.sjtu.edu.cn
School of Microelectronics
Shanghai Jiao Tong University
Fall 2010

Outline

- **Introduce basic GUI programming in Gtk.**
- **Learn the concept of widget, event and signal, and callback, etc.**
- **Learn to create menu, open file, edit text, and display figures, etc.**

GTK vs Qt

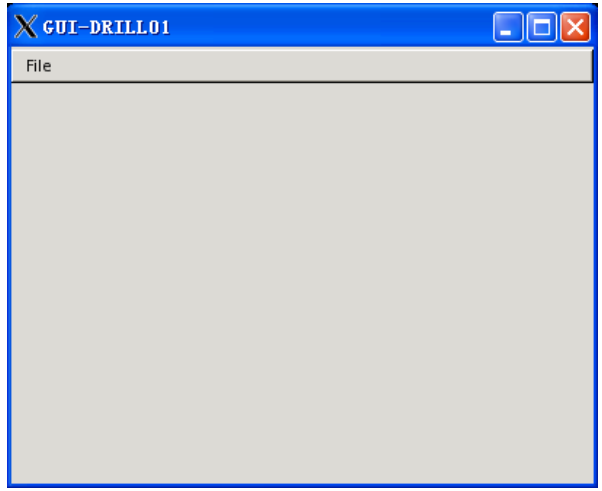
- **GTK is a toolkit for C programming.**
  - **Also possible for C++, but requiring programming skills.**
- **Qt is mainly for C++ programming.**

Make a Simple Window (1)

- **gui_drill01.c**

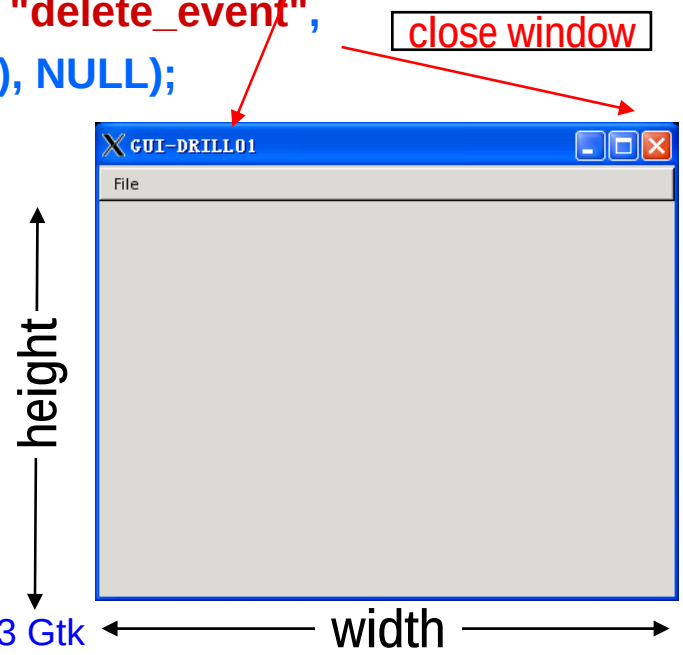
```
#include <stdio.h>
#include <gtk/gtk.h>

int main( int argc, char *argv[] )
{
    int     win_W0 = 400, win_H0 = 300;  /* initial window size */
```

(for GTK lib)



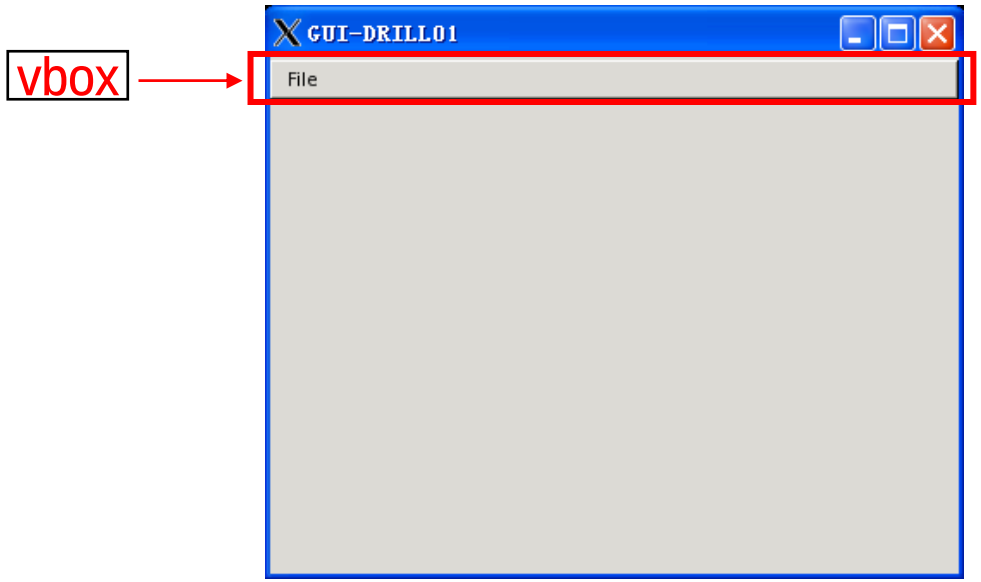
Make a Simple Window (2)

```
gtk_init (&argc, &argv);      /* initialized GTK */
/* Create an initial window */
GtkWidget *window = gtk_window_new (GTK_WINDOW_TOPLEVEL);
gtk_widget_set_size_request (GTK_WIDGET (window), \
                             win_W0, win_H0);
gtk_window_set_title (GTK_WINDOW (window), "GUI-DRILL01");

g_signal_connect (G_OBJECT (window), "delete_event",
                  G_CALLBACK (gtk_main_quit), NULL);
```



“vbox” for holding a menu

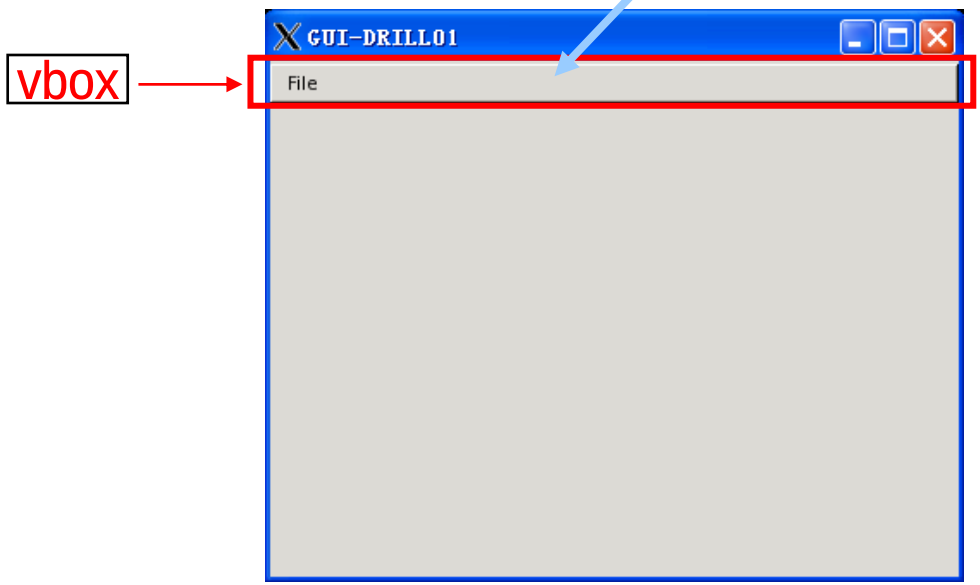
```
/* Make a vbox to hold a menu-bar and other gui layouts. */
GtkWidget *vbox = gtk_vbox_new (FALSE, 0);
  /* <gboolean homogeneous = FALSE>; controls whether each object in the box
   * has the same size */
gtk_container_add (GTK_CONTAINER (window), vbox);
gtk_widget_show (vbox);
```



Make a menu bar

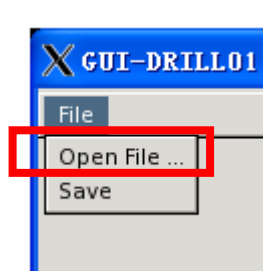
```
/* Create a menu-bar for showing all menu titles. */
 GtkWidget *menu_bar = gtk_menu_bar_new ();
 gtk_box_pack_start (GTK_BOX (vbox), menu_bar, FALSE, FALSE, 2);
 gtk_widget_show (menu_bar);

 GtkWidget *menu = gtk_menu_new ();
```

put the menu_bar in vbox



Make the 1st menu item

```
/* (1) Create the 1st menu-item with a name. */
GtkWidget *menu_item = gtk_menu_item_new_with_label("Open File ...");

/* (1) Append the entry to the menu. */
gtk_menu_shell_append (GTK_MENU_SHELL(menu), menu_item);

 /* (1) Define callback for each menu entry. */
 g_signal_connect_swapped (G_OBJECT(menu_item), "activate",
                           G_CALLBACK(menuitem_response),
                          (gpointer) g_strdup("Open File ...") );

/* (1) Show the 1st menu item. */
gtk_widget_show (menu_item);
```

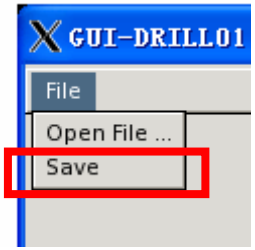


callback function

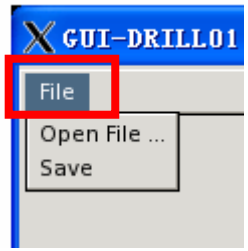
message passed to callback

Make the 2nd menu item

```
/* (2) Create the 2nd menu-item with a name. */
menu_item = gtk_menu_item_new_with_label("Save");

/* (2) Append the entry to the menu. */
gtk_menu_shell_append (GTK_MENU_SHELL(menu), menu_item);

 /* (2) Define callback for each menu entry. */
g_signal_connect_swapped (G_OBJECT(menu_item), "activate",
                          G_CALLBACK(menuitem_response),
                  (gpointer) g_strdup("Save") );

/* (2) Show the 2nd menu item. */
gtk_widget_show (menu_item);
```



Hook up the menu



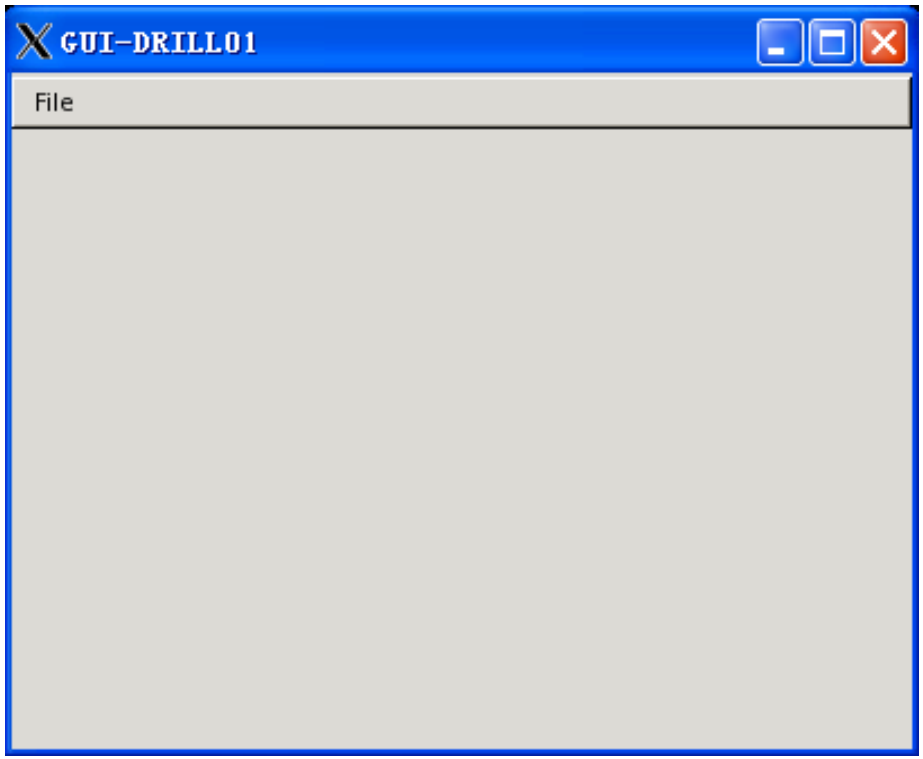

```
/* Define the menu label */
GtkWidget  *menu_head = gtk_menu_item_new_with_label ("File");
gtk_widget_show (menu_head);

/* Hook up the "menu-items" to the "menu_head" */
gtk_menu_item_set_submenu (GTK_MENU_ITEM (menu_head), menu);

/* Append the menu title to the menu_bar. */
gtk_menu_shell_append (GTK_MENU_SHELL (menu_bar), menu_head);

/* Show the window */
gtk_widget_show (window);

gtk_main ();            /* enter the gtk display loop until the window is destroyed */

return 0;
}  /* END OF THE MAIN CODE */
```

The callback function

```
static void menuitem_response( gchar *string )
{
   printf ("%s\n", string);
}
```

The final window

- **The GTK window generated**



* Your SPICE simulator should have a window for functionalities and displaying.

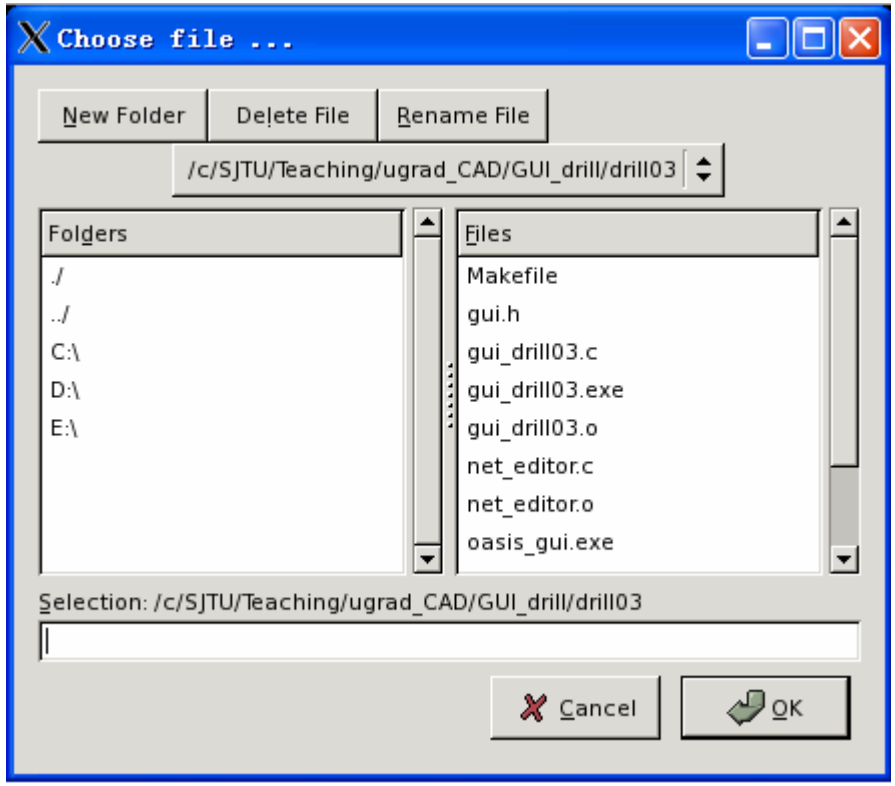
Use “makefile”

```
CC = gcc
PROGRAM = gui_drill01

all: $(PROGRAM)

$(PROGRAM): $(PROGRAM).c
   $(CC) $< -o $@ \
        `pkg-config gtk+-2.0 --cflags --libs`
```

(needed for linking to the GTK libraries)

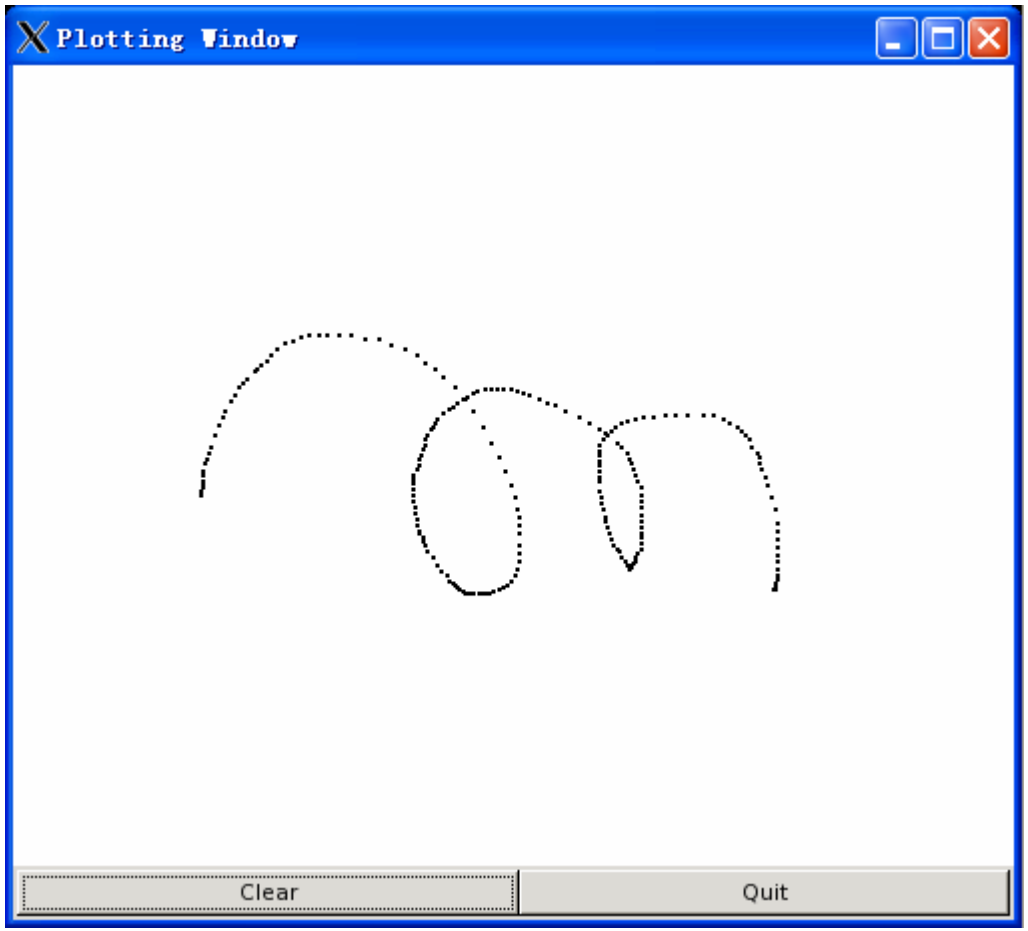
Important: Learn to write “makefile” for compiling your project and for multiple-task programming.

Choose a file to open ...



The “Choose file” widget provided by Gtk.

A scribble window

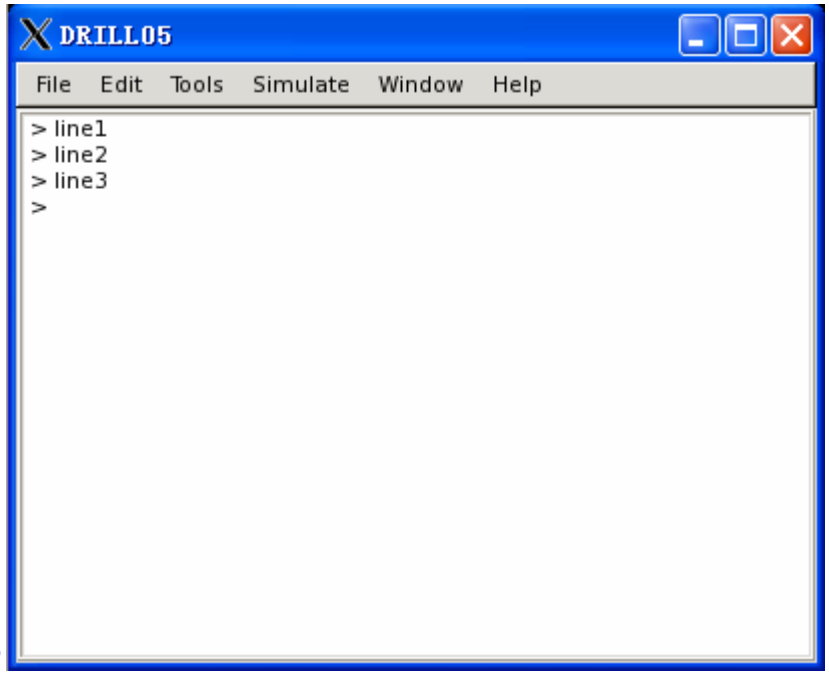


You can program to do hand-drawing.

You also can program to display coordinates and simulation waveforms.

How to make a command window?

- **GTK does not provide a widget for use as a command window.**
- **User can type in text and do text editing in the command window.**
- **You can use the text widget to program a command window.**



Command Window

- **A command window is not purely a text editor.**
- **You need a command parser to recognize the commands.**
- **Programming a text widget into a simple command window is not hard (exercise).**
- **Programming a sophisticated command window (like the one in MATLAB) requires great programming skills:**
  - **symbol table; parser; grammar; etc.**

Summary

- **Introduced 5 programming drills in this lecture:**
  1. **A simple window with a simple menu.**
  2. **A simple window with a menu system.**
  3. **A window with two menu items working: one for opening a text file, the other for popping up a drawing window.**
  4. **Modify 3 so that multiple text files can be opened in tabs.**
  5. **Create a simple command window (but command line not parsed).**

References

- **GTK+ Reference Manual**
- **GTK+ 2.0 Tutorial**
- **GDK Reference Manual**
  - **GIMP Drawing Kit (APIs for drawing and windowing)**

- **All available online**

Programming Kickoff

- **How to write professional programs?**
  - **Divide your project into modules**
  - **Learn multiple-file programming**
  - **Compile using "makefile"**

- **After exercising Gtk/Qt for a while, every team must decide on using one GUI toolkit.**

Assignment 1

This assignment contains the following parts:

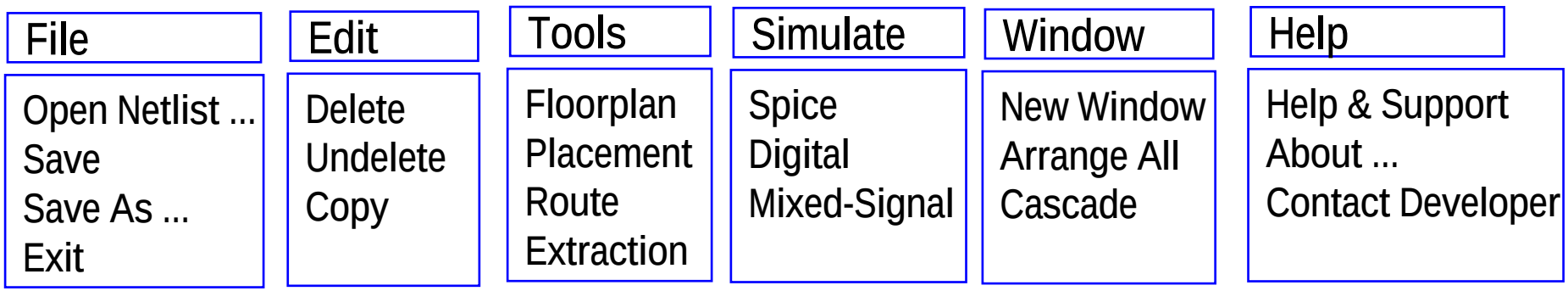
- **Learn GTK and run some examples.**
- **Write a simple GUI window containing**
  1. **a menu system;**
  2. **a text editor;**
  3. **a drawing popup window;**
  4. **a simple command window.**

  (see the requirements next ...)

Requirement 1

On the menu system:

- **The following menu system is for your reference.**
- **Make your menu system easily modifiable.**
- **Your menu system always changes as your project proceeds.**
- **So you should not "hard-code" your menu;**
- **rather, think about "clever" programming to make the menu change easy.**



Requirement 2

- **Add the following two functions to your menu**
  1. **Open a text file editor**
  2. **Pop up a window for hand drawing**

- **You can use the GTK examples in the GTK package you download.**
- **Understand the examples and put the separated code together in one GUI program.**

Assignment Due

The rule for "due" through out the course is:

- **Submit your finished assignment (including code and documents) to Moodle in one week after the lecture is finished.**

- **Your turn-in must include:**
  - **A text report describing the programming details;**
  - **Your source code must be well-commented;**
  - **Must have a makefile.**
  - **Don't use any automatically generated makefile.**