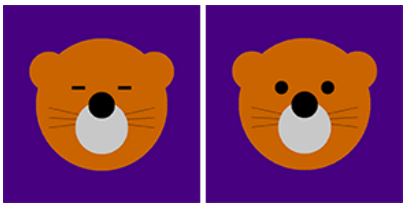
CPSC 1070
Lab Project
Sept. 2 and 4, 2019

In the lab this week, you are to familiarize yourself with the EZ Draw drawing API, and complete the following programming project by Wednesday's lab period.

Write a C program using EZ Draw to draw the face of a tiger on the screen, with its eyes closed. This can be a simple cartoon face, we are not testing your artistic ability. Use at least three different drawing colors for the tiger, in addition to a background color. When the user clicks the left mouse button, the tiger's eyes should open wide, and when the button is released they should close again. Here is an example drawn with only circles, rectangles, and lines. Of course, you could make yours much fancier with a little imagination by simplifying from examples you can find on the web.



Before you begin, download the `ezdraw.zip` file into your home directory, and unzip it using the command `unzip ezdraw.zip` You should now have a directory entitled `ezdraw`. Use the `cd` command to go into the directory, and use `ls` to list its contents. There should be two example programs, `helloworld` and `events`. Use `cd` to go into the `events` directory and study this example. You can compile it using the command `make`, and you can run it using the command `./events` You should get a blue window with an orange rectangle drawn in it. See what happens when you left-click in the window, when you press a key on the keyboard, and when you hold down the left mouse button and move the mouse. Now, examine the code in `events.c` using the `less` command and make sure that you understand what is happening.

Now you are ready to create your program. Use `cd` to go back to your home directory. Use `mkdir` to create a directory called `tiger`, and use `cp` to copy the `ezdraw.h`, `libezdraw.a`, and `Makefile` files into the `tiger` directory from the `ezdraw` directory. Now use `cd` to go into your `tiger` directory, and use either the `vim` or `emacs` editor to edit the `Makefile`'s `NAME` line to be `NAME = tiger` Save the updated `Makefile`, and create your program file in the editor, naming it `tiger.c` Follow the example that you saw in `events.c` to build your program.

When you think you have a complete program, compile and link it using the `make` command. If it compiles and links without errors, you can run it with the command `./tiger` If it has compiler errors, you need to go back to the editor, fix the errors, and recompile until you have a correctly compiling program.

Finally, test the program and repeat until you have it working correctly, and demonstrate it to your lab TA.