

# Setup your codeblocks environment for OpenGL coding

Ramkumar N

June 29, 2018

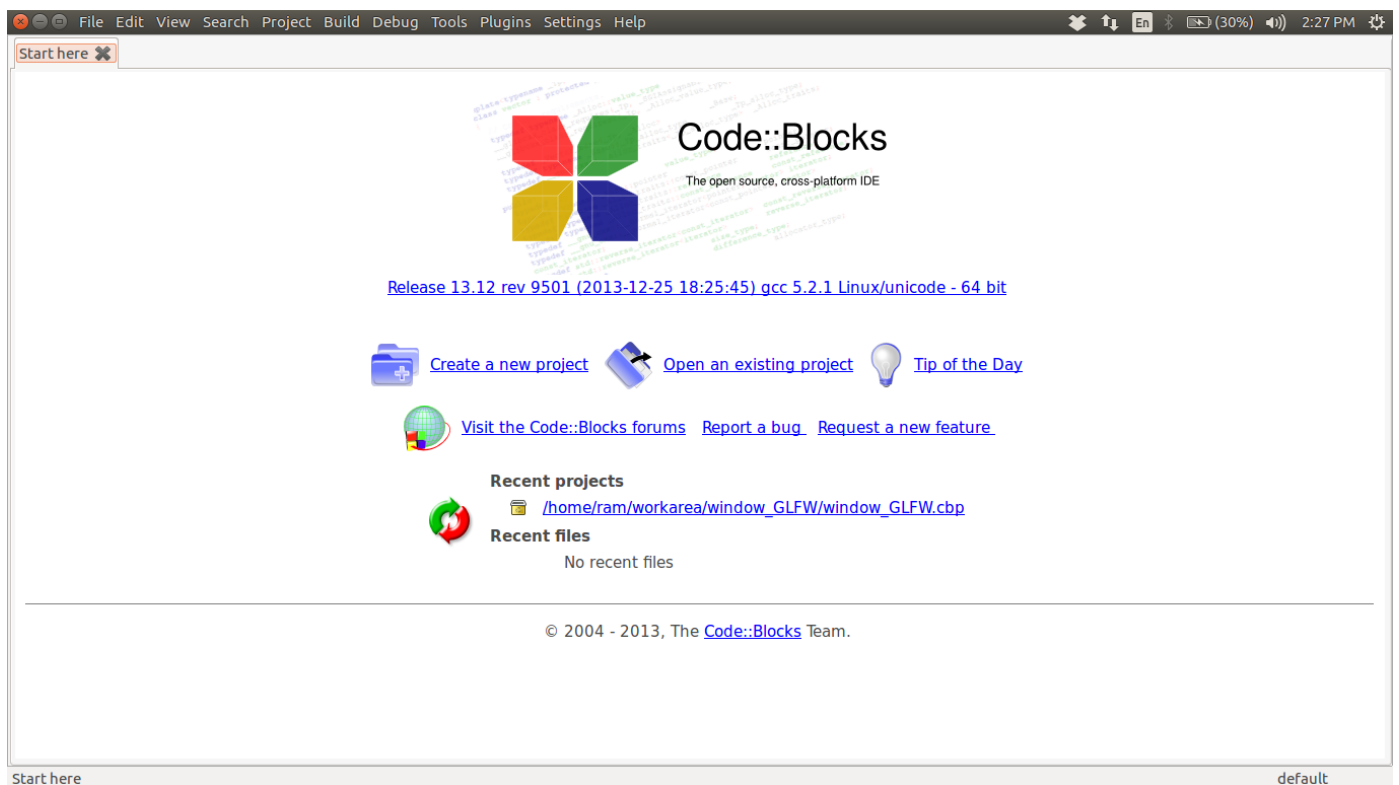
If reading the document is a drag, then follow this setup routine with me by clicking this video link.  
To install codeblocks type the following on your terminal.

```
sudo apt-get install codeblocks
```

Then type

```
codeblocks
```

It will open up a window quite like as shown in the figure below.



Now, let's change the entire theme to make it look more aesthetic. Type the following commands on your terminal.

```
cd $HOME/.codeblocks/  
mv default.conf default_backup.conf  
wget https://raw.githubusercontent.com/ramkalath/config_files/master/cpp/codeblocks/default.conf  
cd ..  
cb_share_config
```

Close the window titled "Welcome to Code::Blocks Share Config". Now open codeblocks IDE again. Click on **Settings** → **Editor**. Then select **Syntax highlighting** → **Color theme**. Select the color theme you like. If you have chosen a dark theme, the cursor might still be set to a black color and it might not be visible. Click on **Margins and caret** button and set

the *Caret* → *Colour* to white by selecting white from the pallet. You can see changes only after you create a new project which we will do next.

The easiest way to create a project is to auto-generate the project file from the **Makefile** using a small *python script* and a default *project template*. The below process sets up an environment for this.

```
cd $HOME/.codeblocks
wget https://raw.githubusercontent.com/ramkalath/config_files/master/cpp/codeblocks/cbproject.py
wget https://raw.githubusercontent.com/ramkalath/config_files/master/cpp/codeblocks/template.cbp
```

Now add the following alias at the end to ur *.bashrc* or *.zshrc* or any other shell config file. To open *.bashrc* using sublime, type

```
sudo subl home/user/.bashrc
```

Then add the following lines at the bottom of your *.bashrc*.

```
alias cb_project_create="python ~/.codeblocks/cbproject.py"
cb(){codeblocks $1 > /dev/null &!}
```

Now close the terminal and open it again.

Let us test this setup. Create a project directory and download a *cpp* file and a *Makefile* using the commands below.

```
wget https://raw.githubusercontent.com/ramkalath/Augmented_Reality_Tutorials/master/ch9/window_Glfw/Makefile
wget https://raw.githubusercontent.com/ramkalath/Augmented_Reality_Tutorials/master/ch9/window_Glfw/
window_Glfw.cpp
```

After that navigate to the directory location on your terminal and type *cb\_project\_create* and you will find a *.cbp* file automatically generated for you. This is the solution/project-file that hosts details about your project so that codeblocks can open and compile it properly. Type the following to open the file.

```
cb /path/name_of_file.cbp
```

On the left you will find the project management window click *Sources* → *your\_filename.cpp*. This will expand and display the *.cpp* file. Click that to open the *.cpp* file. Now press the *View* → *Perspectives* → *Code::Blocks minimal*. Now any command you type will initiate syntatic and semantic autocompletion. The usual convention is to use this as an IDE to compile and execute. However, I still recommend using codeblocks as an editor and typing *make* and *make run* commands on the terminal to compile and execute. You can type *make clean* to clean the unnecessary files after compilation.

**Note:** This setup will only work with the structure of *Makefile* included as a link in this document.

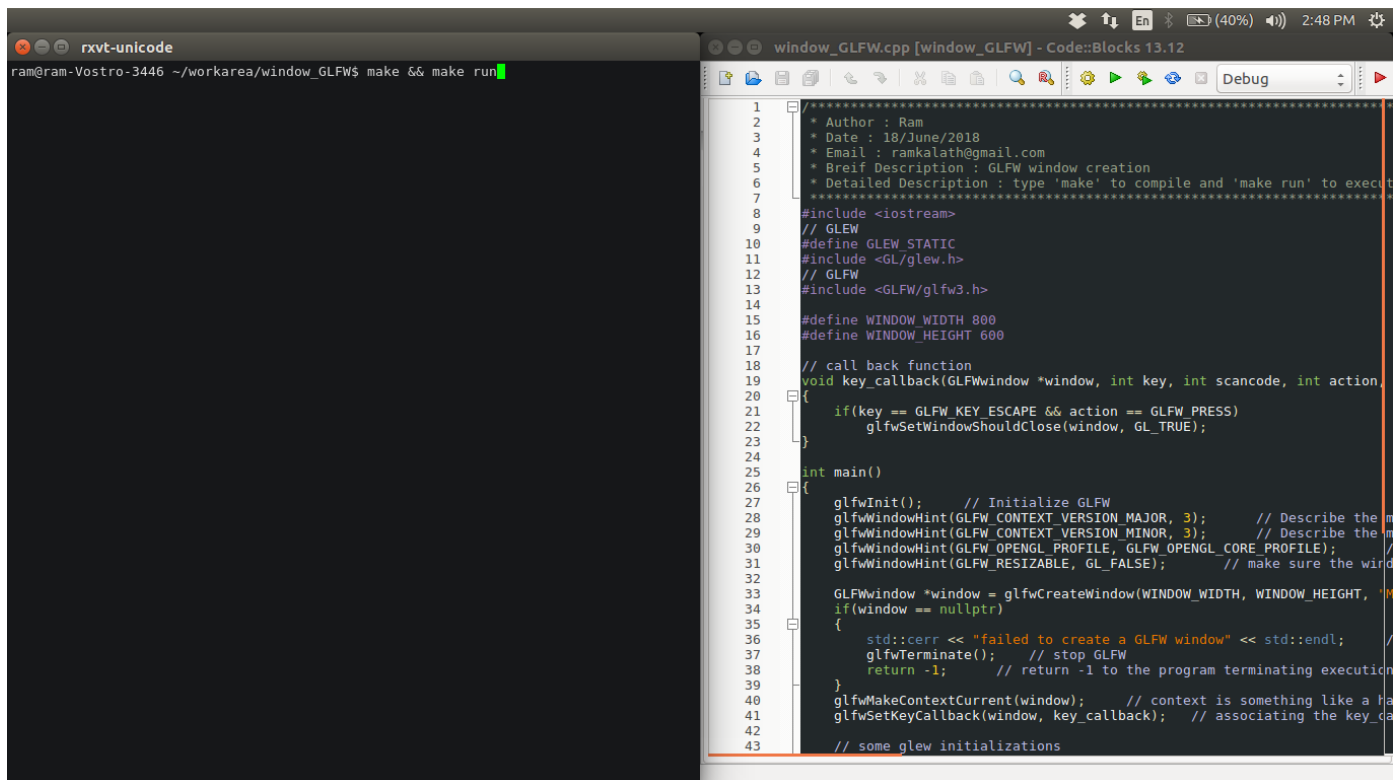


Figure 1: My setup of codeblocks (with son-of-obsidian theme)