macOS

Getting Started in macOS

To work on the Mars Lander exercise, you need to install the Xcode C++ compiler from the App Store. There are two ways you can proceed to use the compiler: by typing instructions into a terminal, or from within the Xcode integrated development environment. Both of these options are described in more detail below.

The integrated development environment option

This is the way to go if you do not want to experiment with the Linux way of doing things, but would prefer to stick with the familiar Apple look and feel.

Here is a quick guide to getting started with the C++ part of the Mars Lander exercise.

Xcode (version 7.0 or newer)

- Unzip the supplied Python and C++ source code into a folder of your choice: remember where this is!
- Start Xcode, for example by searching for it in **Spotlight**. The main Xcode menu should appear at the top of the desktop.
- From the **File** menu, select **New** and then **Project**. A dialog appears. From the menu bar select **macOS** (or, in older versions of Xcode, **OS X Application** from the left pane**)**, from the main pane select **Command Line Tool**, then press **Next**. On the next screen, enter **lander** for **Product Name** and whatever you like for **Organization Identifier**, select **C++** for **Language** and press **Next**. Finally, select where you want to store the project (e.g. **Documents**), untick the **Source Control** box, and press **Create**.The main **lander** project window will now appear.
- In the left pane, right click (or ctrl click if you have a one-button mouse) on **main.cpp**, select **Delete** and then **Move to Trash** (you do not need this file).
- Now select the lander **target** from the left menu of the middle pane, then the **Build Phases** tab in the middle pane, then **Link Binary With Libraries**. Press the **+** button to add a library, then search for and add both **GLUT.framework** and **OpenGL.framework**. Your project window should now show the two frameworks.
- The lander source code calls some functions that are now officially deprecated (though still widely used). If you wish to suppress the consequent warnings, select **Build Settings** in the middle pane, then **All**, then search for **deprecated** and change the **Deprecated Functions** warning setting from **Yes** to **No**.
- Next, right click on the lander **folder** in the left pane, select **Add Files to "lander"**, navigate to where you put the Python and C++ source files, select **lander.cpp**, **lander_graphics.cpp** and **lander.h** (but *not* **spring.cpp)**, tick the **Destination: Copy items if needed** box (you may need to press the **Options** button to see this box), then press **Add**. You can now edit **lander.cpp** by clicking on it in the left pane.
- Unlike Python, all C++ programs must first be *compiled* (i.e. translated into machine code) before they can be run.
- The **Product** menu contains options for compiling (building), debugging and running.

To see the output of any **cout** commands you might insert in the code for debugging, select the **Log/Report navigator** icon (far right icon at the top of the left pane). Then choose your Debug/Run session in the left pane: you will now see the console output in the middle pane.

- To compile and run the **spring.cpp** program for Assignment 3, repeat these instructions except: choose some other **Product Name** (not **lander**); skip the **Link Binary With Libraries** step; and add the **spring.cpp** source code file instead of the three lander source files.
- You may need to enter an administrator password before running/debugging the program from within Xcode.

The terminal option

Your Mac runs a version of the Unix operating system (very similar to Linux) with a proprietary front-end desktop. If you want, you can use it just like any other Unix system, by typing commands directly into a terminal. By doing so, you will learn what goes on "behind the scenes" whenever you press a button on some application's shiny graphical user interface.

You will need to use a text editor to edit the C++ source code. macOS comes with a perfectly adequate editor called TextEdit, though you might prefer to install your favourite alternative editor.

Start a terminal, for example by searching for it in **Spotlight** (you will normally find it in the **Utilities** sub-folder under **Applications**). The terminal runs a standard Unix shell, which is a program that lets you control your computer (run programs, examine the contents of directories, etc) from the keyboard. It typically gives you a prompt, like %, and you can type commands into it. When a program or command has finished, you get the % prompt back.

It would serve you well to learn a few things about the operation of the shell (it's exactly the same as the command line interface to the DPO Linux machines). A good place to start is the first few sections of this excellent tutorial. For example, to launch applications in such a way that you get the % prompt back in the shell straight away and are able to issue new commands, type the application name followed by **&**.

Here is a quick guide to getting started with the C++ part of the Mars Lander exercise:

- Unzip the supplied Python and C++ source code into a folder of your choice. In the shell, use the **cd** command to navigate to this folder. Hint: drag and drop a folder into the terminal to find the folder's location.
- Launch your favourite text editor and open **lander.cpp** (to work on the lander program) or **spring.cpp** (to work on the spring program). You can now edit the C++ source code in your text editor.
- In the shell, type **make** to compile the lander and spring programs. If this does not work, it is probably because the command line tools are not installed. Install them by typing **xcode-select --install** and then follow the installation instructions.
- Type **./lander** to run the lander program, or **./spring** to run the spring program. To quit the lander program, press **Esc** or **q**.
- When you have finished with the shell, type **exit**.

Last modified: Monday, 15 April 2024, 9:29 AM

‹ Windows

Jump to...

Linux ›