

HOW TO SETUP ANGRY MARTLETS IN WINDOWS

By Andrew Herriot
aherriot@uvic.ca

INSTALLING AND SETTING UP ECLIPSE FOR C/C++ DEVELOPMENT

INSTALL MINGW COMPILER

In order to develop C++ programs, we need a compiler. Windows does not come with its own C/C++ compiler. Luckily, there is a free one available called the Minimalist GNU for Windows (MinGW) Compiler.

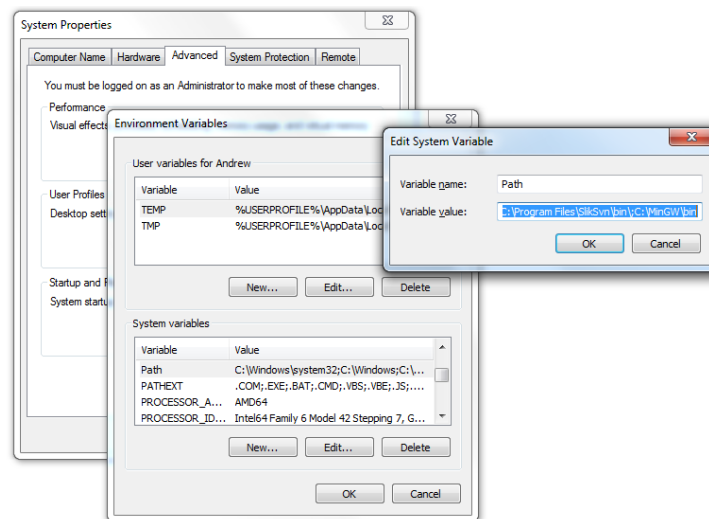
Download and install the latest version of the MinGW Compiler by following the links "**Automated MinGW Installer**" ⇒ "**mingw-get-inst**" at <http://sourceforge.net/projects/mingw/files/>

Once we have installed MinGW, we need to set the "**Path**" system variable to include the MinGW binary directory so that Eclipse will be able to find our compiler. The "**Path**" variable specifies where Windows will look when it is trying to find programs to run that are not in the current directory.

To do this in Windows Vista/7, open the Start menu, right click on "**Computer**" and select "**Properties**". Click "**Advanced System Settings**" ⇒ "**Environment Variables**".

Under "**System Variables**", find the variable named "**Path**" and click "**Edit**".

Append onto the value, a **semi-colon** followed by the location of your "**mingGW\bin**" folder, For example, add "**;%C:\MinGW\bin**"



INSTALL JAVA

Java is required because Eclipse is written in it. You can use either the JDK or the JRE, but it is recommended that you get the JDK so that you can also create your own java programs in the future.

Download and install the JDK.

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

INSTALL ECLIPSE

Eclipse is a free flexible integrated development environment (IDE) that should help with you accomplish many of your software development tasks.

Download and extract Eclipse to a place that you want to keep it (perhaps in your "**Program Files**" folder) from <http://www.eclipse.org/downloads/packages/release/helios/sr2>. Eclipse does not come with an installer, so you can manually create shortcuts to "**eclipse.exe**" for your desktop or start menu.

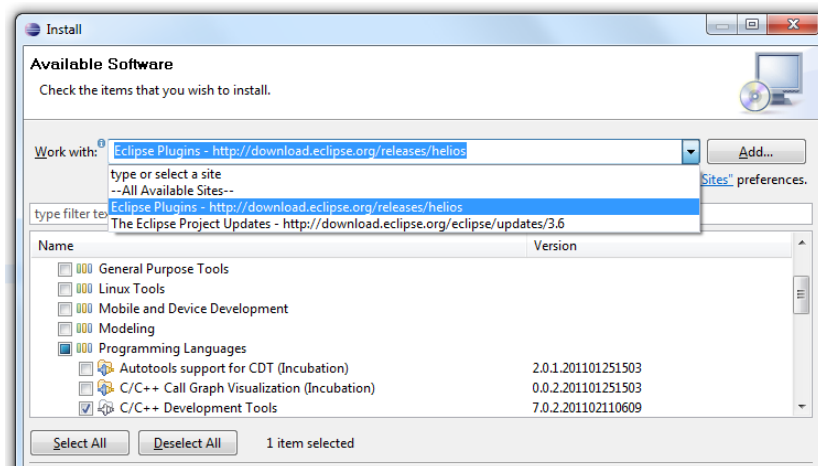
Start Eclipse.

Here it will ask you to specify a "**Workspace**" which is a folder where you can maintain all of your Eclipse projects. Once you have that set up, you are ready to add C/C++ support to Eclipse.

In Eclipse, go to "**Help**" ⇒ "**Install New Software...**".

Click the drop down menu at the top and select the following update site if it exists:

<http://download.eclipse.org/releases/helios>



If it does not exist, click "**Add**" and fill "**Eclipse Update Site**" for the name, and use this as a location: <http://download.eclipse.org/releases/helios>. Select the site you just added from the drop down menu.

Wait for the available software to populate in the list.

Expand the "**Programming Languages**" list.

Check the "**C/C++ Development Tools**" option.

Keep clicking next, and agree to any license agreements until it downloads and installs.

Restart Eclipse.

SETTING UP THE ECLIPSE PROJECT

DOWNLOAD THE PROJECT FILES

A few links to the actual project files.

All of the code can be found at <http://agherriot.wordpress.com> in one large zip file. Inside there is the "**Angry Martlets**" eclipse project folder along with Box2D files and a few manuals.

While the "**Angry Marlets.zip**" file contains all of the Box2D files, you can always get the latest version at:

<http://code.google.com/p/box2d/>

You may want the source code for GLUI and freeglut, instead of just the dll files, which are included above. If you use the source code for these, you will have to figure out how to build them yourself.

Download GLUI at <http://glui.sourceforge.net/>

Download freeglut at <http://sourceforge.net/projects/freetgl/>

MAKE SURE YOU HAVE THE RIGHT PERSPECTIVE OPEN

Eclipse uses a concept called "Perspectives". Different tasks require different menu items and features, and perspectives allow you to select the interface you need for the job at hand.

Click "**Window**" ⇒ "**Open Perspective**" ⇒ "**Other**" ⇒ "**C/C++**".

If you do not see the "**Project Explorer**" on the left, then

Click "**Window**" ⇒ "**Show View**" ⇒ "**Project Explorer**".

SETTING UP THE PROJECT

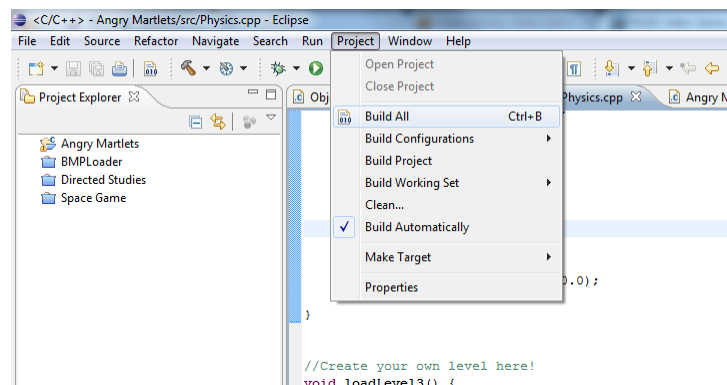
Every time you create a new project, you have to set a few options to make sure the project can find all the right files it needs to compile.

In order for to start using the project, you simply drag and drop the project folder into your workspace. Then right-click in the "**Project Explorer**" and select "**Refresh**".

The "**Angry Martlets**" project should be visible, where you can now right-click it and select "**Open Project**".

Once the project is open, select the "**Project**" menu and then "**Build All**". If everything went according to plan, you should now be able to open up the project folder and find the executable in the "**Debug**" folder.

The challenge now is to see if you can create a third level for Angry Martlets. The next page outlines the project structure. Good luck!



PROJECT OVERVIEW

THE PROJECT STRUCTURE

This section outlines what can be found in each file of the project.

The source code for Angry birds contains the following files:

File	Description
Angry Marlets.cpp	This is the main file that also contains the game loop.
BMPLoader	The functions for uploading bitmaps.
ContactListener	The functions that are called every time two objects collide.
Graphics	For initializing the graphics and drawing, and also for creating and handling the GUI.
ObjectData	Each physical object has a class of this type attached to it to hold extra parameters.
ObjectDefinition	This file defines the specific objects we are using such as birds, bunnies, and wooden planks.
Physics	This file initializes the physics and also holds the level data.
Settings	Retrieves and holds parameters for the graphics and physics.
Utilities	Extra constants and functions for use throughout the program.
Box2D	The Box2D folder contains all the Box2D files.

SOME FINAL NOTES

Here are a few odds and ends.

There are many places to find help with various aspects of the project. Box2D has a fantastic manual and API; and GLUT comes with a manual. There are tonnes of tutorials and an API for OpenGL, GLU, and GLUT on the internet. You can also find a comprehensive C++ tutorial at www.stanford.edu/class/cs106b/materials/CS106BX-Reader.pdf

Make sure to always have a copy of **glut.dll** and **glui.dll** in the same folder as your executable in order for it to run.

On computers that do not have MinGW installed (most computers), you must also include two other files in the same folder as your executable. The **libgcc_s_dw2-1.dll** and **libstdc++-6.dll** files contain the standard C/C++ library implementations. So if you want to share your work with your friends and family, make sure you include these in your distribution.

To create your own texture, it must be a 24-bit bitmap file, and the dimensions must be a power of 2. For example, it can be 64x64, 32x128, 256x64, etc. When the image is loaded by the program, any pixel in the image that is pure Cyan (Red = 0, Green = 255, and Blue = 255) will be turned transparent. It is recommended that you draw your textures with a black background, and then afterwards, use a "Bucket Fill" tool to change only the pure black pixels to cyan. This way, you will not have any ugly pixels that are slightly off cyan bordering along the edges.

I hope that you attempt to setup and build this project for yourself. It might be a bit of a challenge, but once you have it up and running, the possibilities are endless. Good luck and don't hesitate to contact me if you need to.