



Introduction to Programming

Pass Task 2.1: SwinGame: Hello World

Overview

As a first step before using SwinGame, we will create a simple classic “Hello World” program. This will help ensure that you have all of the software installed correctly, and are ready to move on with creating other programs.

- Purpose:** Install and test the tools needed to get started programming.
- Task:** Create your own Hello World program using the command line compiler.
- Time:** This task should be completed in your third lab class and submitted for feedback before the start of week 4.
- Resources:**
- Chapter 1 of the Programming Arcana
 - Swinburne CodeCasts ([YouTube Channel](#), [iTunesU](#))
 - [Series Introduction](#)
 - Install videos for [Linux](#), [Mac OS X](#), and [Windows](#).
 - [Compiling and Using the Terminal](#)
 - Syntax Videos
 - [Introduction](#), [Getting Started](#), [Calling Procedures](#), and [Creating Your Own Procedures](#)

Submission Details

You must submit the following files to Doubtfire:

- Hello World source code (HelloWorld.pas)
- Screenshot of the Terminal showing use of **cd**, **ls**, and **fpc** commands as well as execution of your Hello World program.

Make sure that your task has the following in your submission:

- Code layout - match the example for indentation and use of case (UPPERCASE, lowercase, and PascalCase) for the different aspects of the code.
- The code must compile and the screenshot show it working on your computer.

Preparation.

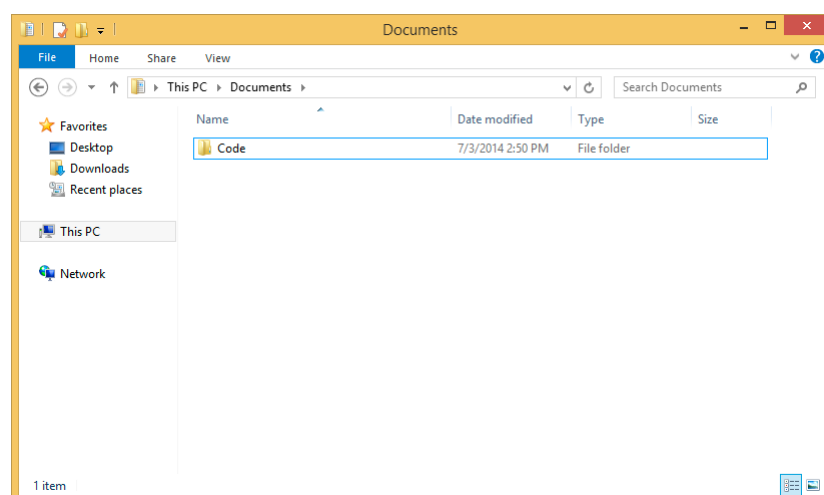
If you have not already done it then you will need to follow the steps to install the tools you will need to use SwinGame in this unit. We will first set these tools up and check they are working by creating, compiling and running the classic 'Hello World' program.

1. Install the tools you need to get started. Watch the installation video for your Operating System and *Read Chapter 1 of the Programming Arcana* for full instructions. Ensure that you have:
 - Installed MinGW for Windows, Xcode for Mac, and devtools for Linux
 - Installed the **Free Pascal Compiler** (the compiler)
 - Installed **SublimeText** (the text editor)

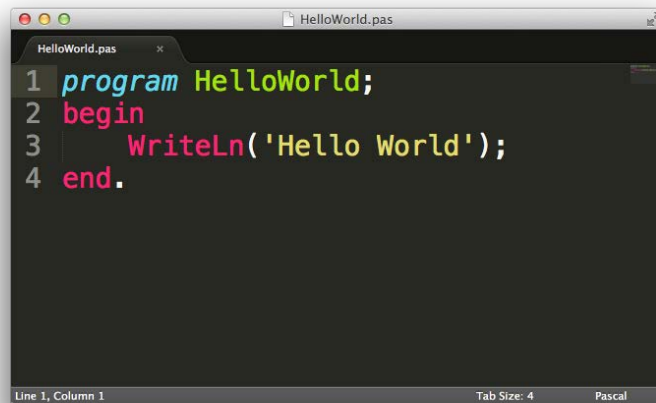
Note: You can skip this step on the computers in the Swinburne lab as this will already be setup.

Hint: If you are using your own Windows machine remember to install MinGW to give you access to the Terminal. See Chapter 1 of the Programming Arcana and the install video for extra help on how to do this.

2. If you don't already have one, make a directory (i.e., a 'folder') to store your code (e.g., *Documents/Code/Lab1*). On a Swinburne computer you may wish to use a directory on your student drive or a USB storage device.
 - Navigate to your *Documents* directory in Finder or File Explorer
 - Right click in the *Documents* directory and select **New Folder**, name it **Code**



3. Open **Sublime Text**, and create a new file.
4. Enter the text for a basic *Hello World* program. It should appear as shown here:

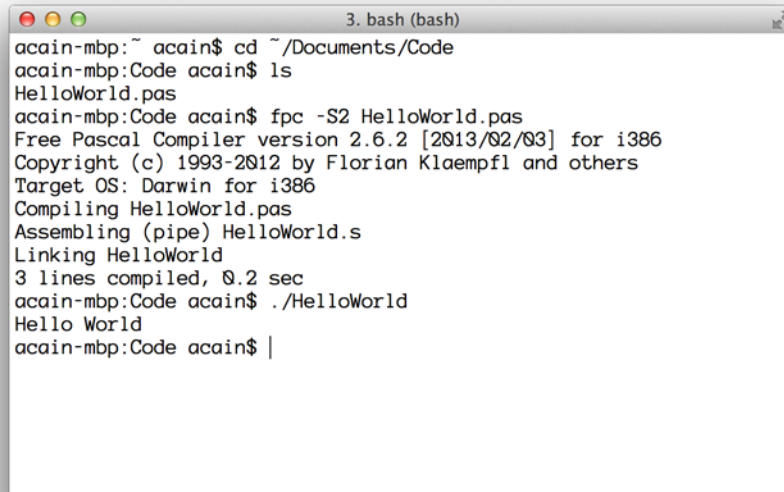


Note: Sublime Text has a number of different colour schemes. Your colours may be different, but Sublime Text should highlight different parts of your code in different colours once you save it as a **.pas** file.

5. Save the file as **HelloWorld.pas** in your code directory.

Note: The pas file contains the *source code* for your program. To make this a *real* program you need to compile it.

6. Open a **Terminal** (MSYS or MinGW shell in Windows), then perform the following commands:
 - Change into the directory containing your code using the **cd** command.
`cd /c/Users/your_user/Documents/Code` on Windows¹ or
`cd ~/Documents/Code` on MacOS or Linux
 - List the files in this directory using the **ls** command
 - Print the working directory using the **pwd** command
 - Compile your program using **fpc -S2 HelloWorld.pas**
 - List the files in this directory using the **ls** command to see the files created by the compile process
 - Run your program using **./HelloWorld**

A terminal window titled "3. bash (bash)" showing a series of commands and their outputs. The user navigates to the directory ~/Documents/Code, lists the contents (HelloWorld.pas), and then compiles the program using the Free Pascal Compiler (fpc). The compiler output shows the version (2.6.2), target OS (Darwin for i386), and compilation details (3 lines compiled, 0.2 sec). Finally, the user runs the compiled program (./HelloWorld), which outputs "Hello World".

```
acain-mbp:~ acain$ cd ~/Documents/Code
acain-mbp:Code acain$ ls
HelloWorld.pas
acain-mbp:Code acain$ fpc -S2 HelloWorld.pas
Free Pascal Compiler version 2.6.2 [2013/02/03] for i386
Copyright (c) 1993-2012 by Florian Klaempfl and others
Target OS: Darwin for i386
Compiling HelloWorld.pas
Assembling (pipe) HelloWorld.s
Linking HelloWorld
3 lines compiled, 0.2 sec
acain-mbp:Code acain$ ./HelloWorld
Hello World
acain-mbp:Code acain$ |
```

Tip: Bash commands (e.g., `cd`, `ls`, `pwd`, `fpc`) do not like spaces in directory or file names (e.g., `My Documents`, or `Hello World.pas`). If you have a space in the name of something you need to add in a reverse slash:

`My\ Documents` and `Hello\ World.pas`

Avoid spaces in the names of your files and folders!

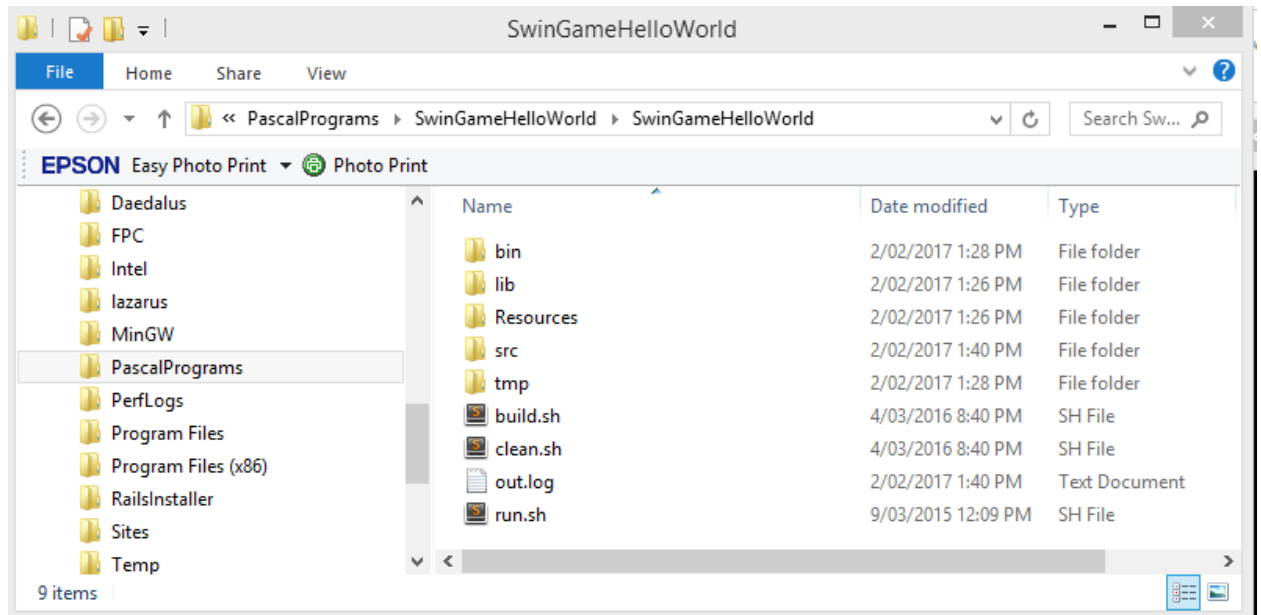
¹ Replace `your_user` with your computer user name

Instructions

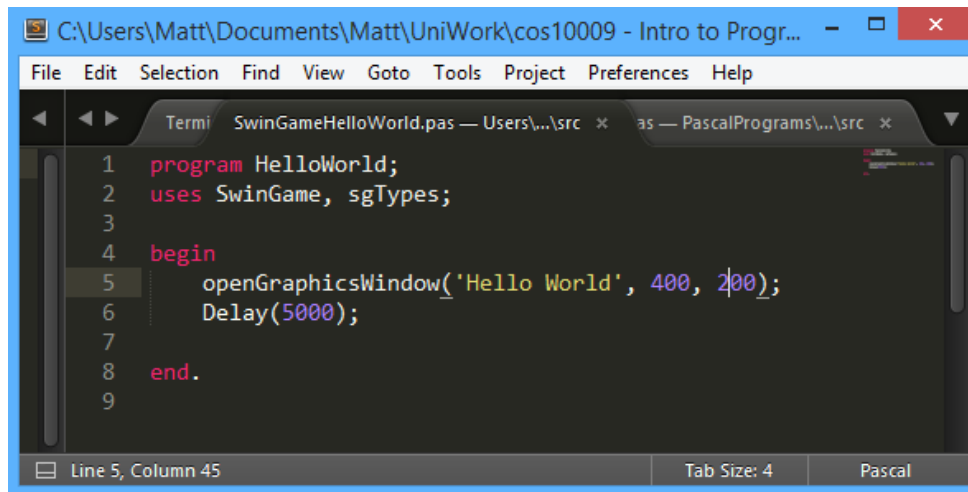
Now that you are sure you have the tools set up and working properly, we can run our first SwinGame program.

When building a program using *SwinGame* you need to start with a *SwinGame Project Template*. This template contains the necessary development environment and code that you can make use of.

1. Download the code from this task's Doubtfire resources. This contains everything you need to create your own programs that use the SwinGame SDK.
2. Extract the zip file to your code directory (e.g. Documents/Code)
3. Open the SwinGameHelloWorld folder and you should see the following files:
 - The **clean.sh**, **build.sh** and **run.sh** scripts. These contain bash commands you will use to clean up the project, compile (or build) the project, and run your program.
 - The **src** folder contains your program's source code
 - The **lib** folder contains the *SwinGame* code
 - The **Resources** folder contains images, sounds, and other resources you want to use.



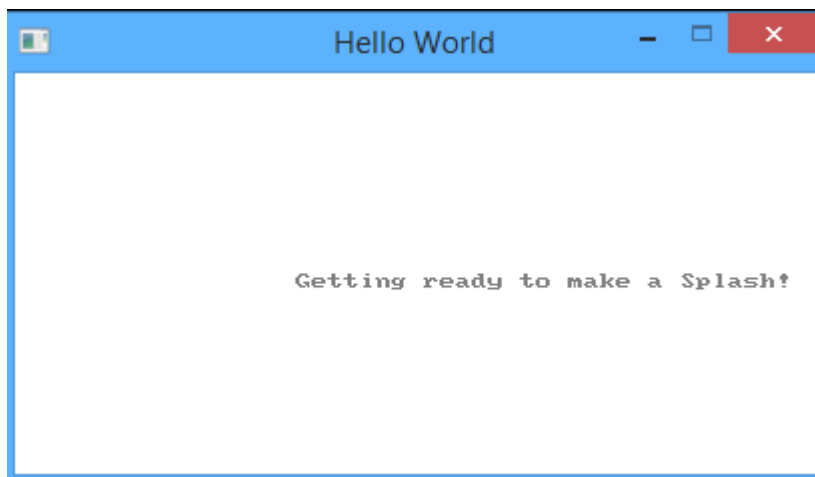
Open the **SwinGameHelloWorld.pas** file in the **src** folder using Sublime Text. It should look like this:



```
1 program HelloWorld;
2 uses SwinGame, sgTypes;
3
4 begin
5   openGraphicsWindow('Hello World', 400, 200);
6   Delay(5000);
7
8 end.
9
```

The **uses SwinGame** line of code gives you access to the SwinGame library and its functions and procedures.

4. Compile and run **SwinGameHelloWorld.pas** by doing the following:
- Open a **Terminal** window (or use one you already have open)
 - Change into the *SwinGameHelloWorld* directory using the **cd** command
 - Compile the template program using **./build.sh**
 - Run your program using **./run.sh** you should see the following screen appear for a few seconds:



Note: The **./build.sh** is a Bash script, it is a saved list of commands you could type at the terminal yourself - this just saves you time. This Bash script was written especially for compiling *SwinGame*. It calls the compiler for you, as well as copying across the files you need for the program to run.

Tip: Although the code is stored in the **src** directory, you need to compile and run *SwinGame* programs from the directory containing the **build.sh** and **run.sh** commands.

Now that the Task is complete you can submit it for assessment, which will help prepare it for your portfolio.

1. Use [Sketch](#) (or your preferred screenshot program) to take a screenshot of the SwinGame window and the Terminal as these are the things you will need to submit.
2. Login to Doubtfire, and locate Pass Task 2.1
3. Change the status of the task to **Ready To Mark**
4. Upload your completed Hello World code and the screenshot.
5. If you check back later Doubtfire will have prepared these as PDFs for your tutor to assess.

You now have another portfolio piece and you are ready to go with your SwinGame programming.

Note: This is one of the tasks you need to **submit to Doubtfire**. Check the assessment criteria for the important aspect your tutor will check.