Creating Games in Python

# Introduction

Your task for this week is to see what it is like to work with another programming language based on what you have learned about programming so far.

You will learn to work with the Python programming language along with a collection of Python modules called Pygame that are specifically designed for writing games.

Learning a programming language from scratch is a daunting task and not something we'd look to achieve in a week.

However, by using what you have learned about programming in your course so far, you should be able to understand an existing Python program and modify it to make your own functionality.

You will work together in groups to modify the Snake clone provided with functionality of your own choosing.

# End of Week Presentation

You will meet up with your staff supervisor at the end of the week for an informal presentation in room 4506. You may wish to use a PowerPoint presentation to show the work the group has done. At this point you can discuss the choices you made and receive feedback from your staff supervisor.

# Task One

Python uses a number of similar mechanisms to C++ to create functionality within our programs.

* Programs in both languages consist of a number of statements that are executed sequentially.
* Both languages use if statements in order make decisions within a program.
* Both languages use loops to provide repetition within a program.
* Functions can be created with both languages.

The first part of your task is to find out how to achieve programming mechanisms you are familiar with in C++ in Python.

* What does a for loop look like in Python?
* What does an if statement look like in Python?
* How do I define a function in Python?
* How do I pass parameters in Python?
* How do I return a value from a Python function?
* What else?

## Task Two

There are some significant differences between the languages. Python doesn't have arrays like C++, but it does have the very powerful list data structure.

Lists can be used in a similar way to arrays in C++, but they are a lot more flexible.

Carry out some research to find out the following:

* How do I create a list?
* How do I access elements in a list?
* How do I change elements in a list?
* How do I write a for loop that goes through all elements in a list?

# Task Three

Spend some time looking at the wormy.py script in your groups. You'll notice when you run the script, it has very basic functionality.

Decide in your group what functionality and features you'd like to add to this program to make it a more interesting game.

For example:

* Can you slow the snake down?
* There is always just one apple, can you add more?
* Can you add some sound effects?
* Can you load image files in to improve the graphics?

Try not come up with anything too complex. Remember you only have a few days to implement them.

Choose one feature each to start with and implement that feature in your own version of wormy.py

# Integrating It All Together

Before Friday, (Thursday at the latest) try and pull all the features into one version of the wormy.py so your group has a single game with all your features in it.

This can be tricky, depending on how you have all written your code. Try and use functions as much as possible as it’ll be easier to copy over function declarations and calls from one script to another rather than sections of code that are all in the one function.

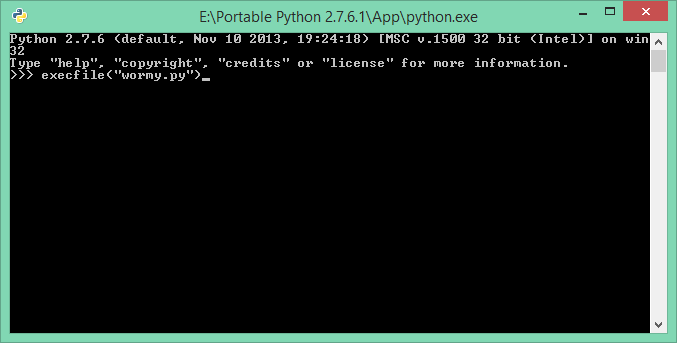
You may wish to start with the unaltered wormy.py script and then each individual team member can add their own changes to the script one after the other.

# Running worm.py

Download the and extract the python\_pygame.zip from Blackboard under the module Programming in C++

Double-click on python.exe.

To execute the wormy.py python script, use the execfile Python function as follows:



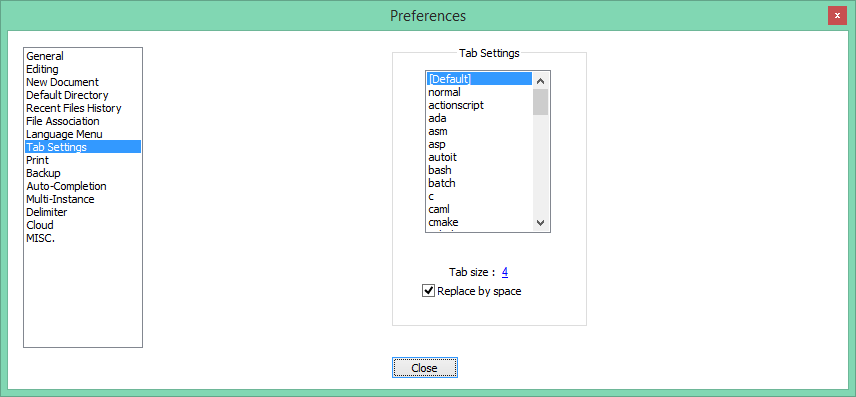
# Editing the Script

You can use any text editor to edit the script. Notepad++ has a couple of features that make it very helpful when it comes to editing Python scripts.

One of the major differences between Python and C++ is the use of white space (spaces, tabs) has meaning in Python and you therefore have to make sure you use spaces and tabs consistently when writing your scripts.

The wormy.py uses spaces throughout for indentation. To get Notepad++ to uses spaces for all indentation, you have to change the Tab settings from the Settings >>> Preferences menu.

Make sure the Tab size is 4 and Replace by space is ticked.



It can also be useful to see tabs and spaces within the editor. You can turn this on from the View menu.

# 

# Links

Python Programming Language

<http://www.python.org>

Pygame

<http://pygame.org>