

Programming 2

Course Information

- CVG: Programming 2
- Website: www.activehelix.co.uk/courses

Recommended Reading

- Virtually any XNA Book!
- XNA Game Studio Creators Guide

Programming Concepts

- Machine languages tell a computer what to do
- Assembly languages tell a computer what to do in readable human-like terms
- High-level programming languages allow you to abstract your programs away from low-level machine language and allow you to describe them in an easier fashion

- Virtual machines can help port programs to many platforms easily
- All programs can be reduced into machine language formats
- NET speeds up the VM process by translating the code only the first time it is run; JIT compilation

.NET Framework

- The Microsoft .NET Framework is a software component that is a part of Microsoft Windows operating systems
- It has a large library of pre-coded solutions to common program requirements
- It manages the execution of programs written specifically for the framework

- Programs written for the .NET Framework execute in a software environment that manages the program's runtime requirements.
- This runtime environment, which is also a part of the .NET Framework, is known as the Common Language Runtime (CLR).
- The CLR provides the appearance of an application virtual machine, so that programmers need not consider the capabilities of the specific CPU that will execute the program.
- The CLR also provides other important services such as security mechanisms, memory management, and exception handling

Principal Design Features

- Interoperability
 - The .NET Framework provides means to access functionality that is implemented in programs that execute outside the .NET environment
 - Access to COM components is provided
- Common Runtime Engine
 - Programming languages on the .NET Framework compile into an intermediate language
 - Common Intermediate Language, or CIL (formerly known as Microsoft Intermediate Language, or MSIL)
 - This intermediate language is not interpreted, but rather compiled in a manner known as just-in-time compilation (JIT) into native code
 - The combination of these concepts is called the Common Language Infrastructure (CLI)

- **Language Independence**

- The .NET Framework introduces a Common Type System, or CTS
- The CTS specification defines all possible datatypes and programming constructs supported by the CLR and how they may or may not interact with each other
- The .NET Framework supports the exchange of instances of types between programs written in any of the .NET languages.

- **Portability**

- The design of the .NET Framework allows for it to be cross platform compatible
- A program written to use the framework should run without change on any type of system for which the framework is implemented
- Microsoft's commercial implementations of the framework cover Windows, Windows CE, and the Xbox 360
- Microsoft submits the specifications for the Common Language Infrastructure to both ECMA and the ISO, making them available as open standards

- **Base Class Library**

- A library of functionality available to all languages using the .NET Framework
- Provides classes which encapsulate a number of common functions

- **Simplified Deployment**

- Managed installation tools

- **Security**

- The design is meant to address some of the vulnerabilities, such as buffer overflows, that have been exploited by malicious software

C# Programming Concepts

- C# is Object Oriented and therefore uses classes
- C# is an object-oriented programming language developed by Microsoft as part of the .NET initiative and later approved as a standard by ECMA and ISO.
- Anders Hejlsberg leads development of the C# language, which has a procedural, object-oriented syntax based on C++
- Includes influences from aspects of several other programming languages (most notably Delphi and Java) with a particular emphasis on simplification.

DirectX/XNA Programming Information

- Managed DirectX came out in 2002
- Managed code is computer program code that executes under the management of a virtual machine
- Managed tends to be slower but benefits include programmer convenience

XNA Concepts

- XNA is actually a generic term but is mainly used to refer to the XNA Framework (the API we use to make games)
- XNA is a set of tools, complete with a managed runtime environment, provided by Microsoft, that facilitates computer game design, development, and management
- It is supposed to do this by reducing the amount of boiler plate code needed

- The XNA Framework is based on the native implementation of .NET Compact Framework 2.0 for Xbox 360 development and .NET Framework 2.0 on Windows
- XNA Game Studio is an integrated development environment (IDE) for development of games
- Five revisions have been released so far

- The foundation of XNA is DirectX (See last week)
- XNA was built ground up and is not built on managed DirectX
- XNA utilizes DirectX 9 in the background
- XNA is, essentially, managed DirectX

XNA, DirectX and OpenGL

- There are two main industry wide used 3D API's
 - DirectX and OpenGL
- While earlier versions of DirectX often left a lot to be desired, DirectX 9.0 is (or perhaps was) a serious product worth a longer look, especially if you want to target multiple Microsoft platforms
- With DirectX 9.0, you have a choice of two 3D APIs on the Microsoft Windows platform.
 - OpenGL is generally seen as the industry standard

- Even if you stay with the OpenGL for 3D API you can still learn general API techniques using DirectX
- Just like OpenGL, DirectX too used to be aimed at developers coding in C/C++. That's natural, because compiled C/C++ code is faster than interpreted Visual Basic
- To make .NET a success, they need to attract as many developers as they can.
- All of their crucial APIs are available in a managed form, so they can be easily incorporated into applications written in C#, Visual Basic .NET, and other .NET languages.

- Despite C# being slower than C++
- Despite XNA being slower than DX
- They will still be used... and perhaps become standard
- XNA, although a form of managed DX is still a graphics API

- The windows Graphics Device Interface (GDI) is much too slow for creating real-time action games.
- Due to the limited use of the GDI DirectX was developed to give programmers the necessary tools to go through those extra stages and write professional games
- DX is designed as a low-level API that integrates smoothly with Windows and the Win32 API
- XNA/DX can be used to access video, audio, input devices and networking capabilities without writing one line of GDI or using the standard Win32 libraries
- Using XNA to work with these systems should not cause conflict with Windows, Win32 or the GDI

A Windows Primer

- Windows is a shared, cooperative, multitasking operating system.
- Video games, by their nature break the rule. Games usually take over everything; they also need high performance so generally only game is run at once.
- XNA, DirectX and OpenGL enable you to access the hardware without going through the normal windows channels

Assessments

- Assessment 1: 11th April
- Assessment 2: 9th May

The End