CVG Programming 2: Assessment 2

This is the second of two assessments that contribute to your mark for the Programming 2 module. Each assessment counts for 50% of the total marks for the module.

You are reminded that the work that you submit for this assessment must be entirely your own. If you copy some or all of another student's work, or write any, or all, of your program with the assistance of anyone else, you will be subject to disciplinary procedures. These may result in you being awarded zero marks for the whole assessment, or other, more severe penalties. If you allow another student to copy some or all of your work, you will be subject to the same disciplinary procedures and penalties as if you had copied the other student's work.

You must hand in your program documentation, program directory, including the code files, resources and corresponding vs.net project file (preferably on a CD, a floppy disc will not do) to the Music, Media and Performance School office, by 12.00 p.m. on Friday, 9th May 2008. The program code must contain a comment giving your name and a statement that the program is all your own work. See below for expanded submission details. You may be required to explain what your program code does, in an oral interview with me, or with another member of staff.

Assessment Introduction

The aims are to demonstrate that you can address the real-time constraints of programming an interactive graphical system.
This assessment involves using object oriented programming, facilitated through the C# language, to produce a two-dimensional (2D) game using the XNA API.

The Assessment

You are required to produce a basic 2D game using C# and XNA. The game should included relevant documentation, use sprite images and sounds.
This part of the assessment counts for half the total marks.

Example games would be things like Space Invaders or Asteroids. Whatever you choose it must have some element of originality to it. It should not be a direct copy of something you have already seen. If you are attempting to put together a portfolio it looks more impressive if your game is relatively original.

I can only give little credit for artistic and modelling ability so don't waste too much time on this. This is a relatively open ended specification so make sure that if you have any doubts about your ideas you discuss them with me before implementing them, especially if you have any doubt as to how relevant they are.

Although the game is of your choosing you should remember scope. Do not attempt to make a full 2D game. The game at minimum should be one level, more than this are unnecessary.

Resources

To aid you in the game production there are resources available on activehelix that contain basic images and sounds. Additionally you are permitted to use images and sounds that are not your own as long as you credit the person or website from where they were obtained. You can use your own images/sounds if you prefer.

Examples of features to include are

- Sprite Effects - movement, scaling, rotation, colour variation, brightness variation, alpha blending. The following effects will gain extra marks.
- Moving sprites along straight lines at an angle to the X and Y axes.
- Moving sprites along circular or other non-linear paths.
- Synchronising the rotation of a sprite with its direction of movement.
- Simple animation, using sequences of sprites.
- Collision detection.
- Visual explosions with synchronised sound.
- Other sound effects and music.
- A scrolling background. Depending on the type of game, this could be, for example:
  - a horizontal tiled background scrolling in both X and Y directions, or
  - a multi-layer vertical background with different scrolling rates for the layers, to give an illusion of depth.
- A splash screen, possibly with visual effects and music, while the game is loading.
- A menu screen, which might allow the player to set options, levels of difficulty of play, starting and quitting the game, etc.
- Text messages. For example, to indicate the state of the game, the player's score, the frame rate and other information about the internal working of the program, etc.
- Other visual devices, for example energy bars, life icons, etc., to indicate the game state.
- Player control of one or more objects in the game, using the keyboard and/or the mouse.
- A variety of program-controlled sprite behaviours.

- Programmed behaviours, that take account of the actions of other player-controlled or program-controlled sprites, will gain extra marks.

Examples of document features are

- Clarity of structure
- Quality of language
- Completeness and clarity of the description of the object and features of the game
- Cross referencing your description to the program code
- Completeness and clarity of the instructions for playing the game
- A realistic assessment of what is achievable in relation to the capabilities of the hardware platform as you understand it

Submitting your work

You must submit:

- The complete project directory, all its sub-directories, the debug and release sub-directories and an EXECUTABLE. Obviously this must therefore contain sound and image files.
- A document (in a common format e.g. pdf, txt, doc) explaining:
  - What you wanted to do
  - What problems you had
  - What special features/functions have you used
  - Techniques, effects and features of the game
  - Game object and inputs the player uses to play the game
- A statement saying the work is your own, or if you have used the work of others or had particular help from others this must be clearly acknowledged here.

On the following media:

The submission should include a CDROM, memory stick or any other up-to-date storage, along with a hard (printed) copy of your source code. Please remember floppy is not a reliable media.

To:

The MMP School office

By:

12.00 p.m. on Friday, 23rd May 2008.

With the following taken into account:
All your source files should start with a comment giving your name, the project name, and the module title (CVG Programming 2, Assessment 2, 2007/2008).

Tackling the Problem

The program should be tackled in stages, using the notes and example code to complete one stage before moving on to complete a further stage. For this reason it before writing your program it is good to stop and think about the design. Think how you are going to accomplish the different game tasks; are there any fundamentals that could be shared? Think about the elements of the game, such as the bitmap image you have to hit and how you are going to represent them. Now is the time to think about program structure and to ask questions.
If you need or want help during this stage ask I can give you advice.

After coming up with a design, the design doesn't have to be flawless, then its time to get started on the program. Write your code in sections, such as code one class or aspect of the game, make sure that section works and is robust before moving on to another. In this way it becomes easier to problem solve.

Commenting and Coding Style

You are marked on your comments; you are therefore reminded that your code should be more than adequately commented. See the resource file online for an example of the commenting you are expected to give.

Although I don't expect your coding style to match mine I do expect it to be consistent. I also expect 1 Class to be declared in 1 header and its member functions to be defined in 1 source file. For very small classes this does not apply. Techniques such as using a bitmap class will earn you more marks.

You are expressly permitted to be able to use resources, including code, developed in the tutorials for the assessment. It is expected that such code will help you greatly in developing the framework for this assessment.

Assessment Criteria

Use of 2D Features

Marks will be awarded for the number and variety of sprites, the range of visual techniques and the range of programmed and user-controlled behaviours of sprites.

Implementation of Additional XNA Components

This includes sound techniques, the range of programmed and user-controlled features that are used in the game. The marks will also depend on the effectiveness and quality of the features.

Design, Implementation and Originality

This includes the technical document, completeness of the game, originality of the game and playability.