SimBin Programmer Test

Introduction

Welcome to the SimBin programmer test. The purpose of the test is to provide you with an opportunity to show us how you solve programming problems. It is therefore important that you try to solve the problems presented in a way that communicates to us what you consider important when programming. The test is not a speed-programming test, note that this means that the primary focus is not to provide a complete implementation of the SimbInvaders game, but for you to highlight and develop the parts you consider important.

Make sure you comment and explain design decisions and tradeoffs performed, either with the code or separately. You are encouraged to do an UML diagram to illustrate the design of your solution. A free tool for this is Dia (http://www.gnome.org/projects/dia) or StarUML (http://staruml.sourceforge.net/en/). Do not hand in diagrams in some obscure format, use either pdf or jpg (or dia, uml).

Problem

The task is to implement a version of the venerable Space Invaders, the gameplay must have some parts in common with the original Space Invaders game, but you are very welcome to extend or restrict the gameplay in any way you see fit. As we are testing programming skills, it is not necessary that it is fun, but it should have some gameplay element. Art and sound resources are provided in the Data directory, everything is in 2D.

Requirements

Besides being similar to the original Space Invaders your game must meet the following requirements:

- The game must have a main menu from which the player can start playing or quit the game. When the player is done playing, the game should return to the main menu and provide the player with the same options it did in the beginning.

- The main game objects are the players ship, enemies and shots but keep in mind that more game objects might be added in the future.

- The player must have three lives. When all lives are lost, the game is over.

- Lives left and score must be rendered in the players HUD.

- All resources, art and sound, must be used by the game. You're welcome to replace or extend existing resources if you want to.

Implementation

For the implementation (which may be partial) the below defines the development environment:

- You must use the API classes under the "3rdParty/" directory. The classes should be viewed as being part of an anonymous third party library which you're not allowed to change except if you need to fix some bugs.

- Do not introduce any 3rd party API's platform specific calls.

- Only use C++ and standard libraries.

- An implementation must compile, link and run in Visual Studio 2003 and/or Visual Studio 2005. Visual Studio 2005 Express is available for free (http://msdn.microsoft.com/vstudio/express/visualc/download/). All necessary files must be included in your mail to us.

- Use Build/SimBinVaders.sln for Visual Studio 2005.

- Put your code in Application.cpp and of course new files in the same directory.

- Working directory must be set to the Data/ directory.

Try to find the right balance between flexibility and simplicity when designing and implementing your solution.

Time Constraints

There is no time constraint as such, but you are as stated not expected to carry things too far. Usually it is intended that the result is
returned within a couple of days or so, but it depends on the circumstances under which you operate.

Good Luck!

/SimBin