Deadline: 14-Dec, Total mark: 50% of the module's mark

Arkanoid videogame:

This assignment is inspired by an old famous game knowns as "Arkanoid". This game was popular in the 80s and early 90s. In this game there are multiple lines of bricks at the upper third of the screen, and the aim of the game is to break these bricks using a ball that bounces back when it hits a brick or hits the left/right end of the window. The player moves a paddle at the bottom of the screen left and right to prevent the ball from touching the bottom end of the screen. As soon as the ball hits the bottom end of the screen, the game is over. A screen view of the classic game is shown in figure 1.

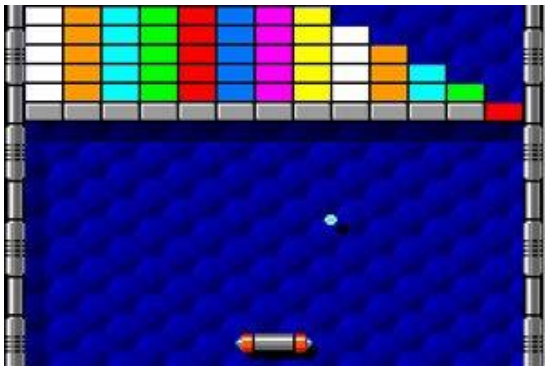

Figure 1: A screen view of Arkanoid.

In this assignment you are required to create a version of "Arknoid" game using MFC. The game should have the following specifications:

1. It should have multiple bricks types, each type has a colour and score assigned to it.
2. The game should have at least 3 rows of bricks, the type of these bricks should be assigned randomly.
3. It should have a score counter.
4. When the ball is lost, a message stating "Game Over: your score is "then the score should be displayed.
5. The control of the paddle should be either by the mouse or by the keyboard (both functionalities must be available).
6. The game should have the ability to save and load a game.
7. The game should have menus and dialogs allowing the user to set the speed of the ball, the grid size, and the colour of the paddle.
8. The game should have a "reset" button allowing the game to start from beginning and resting the score to 0.
9. The "about" dialog should have your full name and student ID number.

Deliverables of the project: All Deliverables should be uploaded to vital.

- Report.
- Executable program (must run in L3 Computer labs!).
- Source code (must compile on Visual Studio in the L3 Computer labs – please 'zip' the solution into one file).

Report specifications:

Section 1: Introduction

An introduction to the objectives of this assignment and an analysis of the project specification with a focus on the functionality of the program. The original problem specification should be included in the report as an appendix.

Section 2: Methodology and Implementation

Detailed description of any mathematics or physical mechanisms that are implemented in the event handlers and how they are implemented. Your explanation should be clear and concise. All variables or symbols must be explained when they appear in the report for the first time.

Section 3: Objects and classes

Describe what object and classes you are using and how they relate to each other.

Section 4: GUI design and role of Event Handlers

Detailed explanation of your design of the GUI and how the GUI can be used to meet the functionality required by the specification. For example, if you click a button on the GUI, what is going to happen?

Also give necessary screenshot captured when your program is running to demonstrate that output from your work. Describe the properties that you need to set or change for each of the components in your application and clearly show in which event handlers the properties are changed or updated. You can use a table to present the information in an organised manner.

Section 5: Source code

A copy of programme code that contains definition of data members, function members of the classes and all event handlers. Give sufficient comments so a reader can easily understand what you intend to do for each of the variables or event handlers.

Also provide a description at the beginning of each event handler to describe its overall functionality. You can copy the code into a Word document, format the code in an easily read style, and then add text comments to explain the code.

Summary

Does your program work? What test have you done to prove that it works? Does it meet all the requirements? What difficulties you have had in the design and coding stage? If you did not finish the project, what is the reason?