

CS 2630: Data Structures

Program 4 (30 points)

Objectives

After this program assignment, students should be able to:

- Expertly create a C++ Windows Desktop Application using the wxWidgets Graphical Framework
- Familiarize themselves with new programming concepts (GUI elements, etc.) through independent study
- Incorporate concepts of Polymorphism into their programs

Description

For this programming assignment, you will create a GUI-based game. In this game, a balloon can fly vertically upward in a game window shown to the player. There will be a canon placed at a specific point on the window. The canon should fire a bomb with every click of a “fire” button. You may decide the direction of the bomb (horizontal, vertical, etc.) If the bomb hits a balloon, then the balloon and the bomb both should disappear. Additionally, hitting a balloon with the bomb should score the gamer some points! The points should depend on the type of the balloon. There should be at least two different types of balloons: you have been provided with images for red and yellow balloons. If the bomb misses the balloon, the gamer will lose points.

Introduction

You have been provided a complete set of files for a wxWidgets project that provides a starting point for you to program a complete game. This starting point is very minimal, the program will allow you to create a Bomb and a Yellow Balloon. I will demonstrate the starting code during lecture. Do NOT change any of the header files that you are given, except for extra credit purposes. I have provided MANY comments throughout the starter code, read carefully and use Google liberally to complete the implementation files for each class. Note, any extra credit must be thoroughly documented. You have also been provided an executable file for Program 4 (Program4.exe) as an example of meeting minimum requirements. Note that the minimum requirements demonstrated in the sample program would earn you a C- (approximately a 68%). It is a very simple version developed with two different types of Balloons and one Bomb. In this executable, balloons are all moving in a vertical direction and bombs are fired in a horizontal direction. You may (and should) improve any aspects of the game in any way you want to (e.g., graphics, options, gameplay, etc.).

As part of the assignment, you have been provided with the images (two different balloons and one bomb) used for the sample executable program, as well as a few extra images that you may choose to use (or not) - you may use any pictures of your choice for your balloon and bomb design (or any extra features you design).

Minimum Program Requirements (≈ 68%)

Here is a summary of minimum requirements:

1. 2 different types of balloons with different speeds;
2. 1 type of bomb moving in different direction from the balloons;
3. A score board which will change whenever a balloon and a bomb (or 2 bombs) collide;
4. The game should go on until all the balloons are popped.

For your program, you will be making at least three derived classes of `MyObject`: two types of balloons and one type of bomb. Furthermore, these objects will be stored in a `MyObjectList` class to handle them. In the starter

code provided, the entire framework for this game has been provided to you. You must read carefully through my comments and figure out how to complete the minimum requirements of the game.

For example, you need to complete the code that determines what happens when two objects collide. Here are the collision rules:

1. When two balloons collide, nothing will happen.
2. When a bomb collides with a balloon, both the colliding objects should disappear, and the user's score should increase.
3. When two bombs collide, both bombs will disappear, and the user's score should decrease.
4. If the bomb misses all balloons and disappears off the right border of the gameplay area, then the bomb should be deleted and the player should lose points.

Hints and Suggestions

Here are some hints and suggestions to help you along:

1. The balloons should start from a random position at the bottom of the screen. In the simplest case, the balloons should move upwards. But you can be creative and make the balloons fly in any way you wish (the game should still be playable though). This makes the game a little more interesting 😊
2. It would be nice if the gamer could control what types of objects and how many are added to the gameplay area via the control panel (extras controls and code). Maybe the application could give the player a way to randomize what types of objects enter the gameplay areas. These are all extra credit suggestions.
3. You can assume that a bomb is fired from a particular place in the window. The gamer needs to click a button (fire) to fire a bomb to pop the balloon. In the simplest case, the bomb should go in a horizontal direction. But, again, you can be creative to make it realistic (maybe add ballistic trajectories?). You can also make use of mouse control to change the direction (considered as extra credit).
4. If a balloon is missed (not popped by the bomb) in the round of its flight, the balloon should loop back to the bottom of the screen. Make sure the flight path makes sense, or not, you can be creative.
5. You need to move all the balloons and bombs in the list and check for collisions.
6. Use a timer. When the timer ticks, move the balloons and bombs. This functionality is essentially already implemented for you in the starter code. But, as mentioned in comments in the starter code, there is so much room for extra credit here – for example, it would be nice if the user could control the speed of objects somehow.
7. To make the bitmap files for the Balloons, you can download bitmap images from internet. Make sure a copy of the image is in the project directory with your code, otherwise the program will crash. Likewise, if you want to copy your executable program to a separate folder to package up the final product, make sure a copy of each image used in your game is in the same folder as the executable program. You will probably need to edit images you download from the internet. I suggest using GIMP (<https://www.gimp.org/>) to edit any images you find. These instructions worked for me when exporting .BMP files to import into Visual Studio:

<https://www.curlybrace.com/words/2013/08/30/exporting-visual-studio-compatible-bitmaps-from-gimp/>

8. At the minimum, you must override the `move` and the `popScore` function to change the speed, direction and score associated with different balloons and bombs. You can also add extra members to your derived balloons and bombs (you can change the header files for classes derived from `MyObject`).
9. Creativity is strongly encouraged. The remaining 32% of your Program 4 grade is based on extras you add to the project. You may also earn additional extra credit above 100% for interesting project additions – these points will be assessed by me on an "as it strikes me" basis. If you plan to get extra credit, you must thoroughly document what the extra credit is and how you have implemented it.
10. You need to work hard to get this extra credit. For example, only changing the game background or including an extra class of `Balloon/Bomb` with the 'SAME' functionality will not earn you too much credit. Your extras should demonstrate good logic and/or design principles.
11. For those of you interested in more complex layouts, I watched the following tutorial video on YouTube to help with the initial layout I gave you in the starter code:

<https://www.youtube.com/watch?v=kPB5Y6ef9dw>

Submission

You will NOT be submitting this program to Gradescope. Instead, you will submit a .ZIP file of your Visual Studio Solution – make sure to exclude unnecessary files like the hidden “.vs” folder. Remember, the hidden folder “.vs” is **HUGE**. **DO NOT add this to your .ZIP file!! DO NOT submit a 200MB project folder**, it will take me forever to download student submissions for grading and **you will lose points for doing so**.

I would also like a copy of your executable program. You may include this in another folder added to your .ZIP file. Remember, the folder with your executable program **MUST** have a copy of all images loaded in your program.

I will run your executable program for testing and grading. I will also inspect various aspects of your code, including comments and polymorphism-related implementation. I will also evaluate your extra credit and determine the remaining 32% of your project score (and potentially give you additional credit beyond 100%).

When you are sure you have a working version of your program, submit the Final version to Canvas. Any submissions after the due date will be assessed a penalty per day late (I will determine late penalties based on class assessment).

Rubric

Criteria	Possible Points
Appropriately derived balloon and bomb classes based on <code>MyObject</code>	6
Appropriately completion of <code>MyObjectList</code> implementation	4
Player score appropriately updated according to the given rules	4
Correct game play – player can fire bombs and pop all balloons	6
Documented and working extra credit functionality	10 + ?
Total	30 + ?