***Bomberman***

John Mapelli

Alex Yurkowski

Catherine Kromkowski

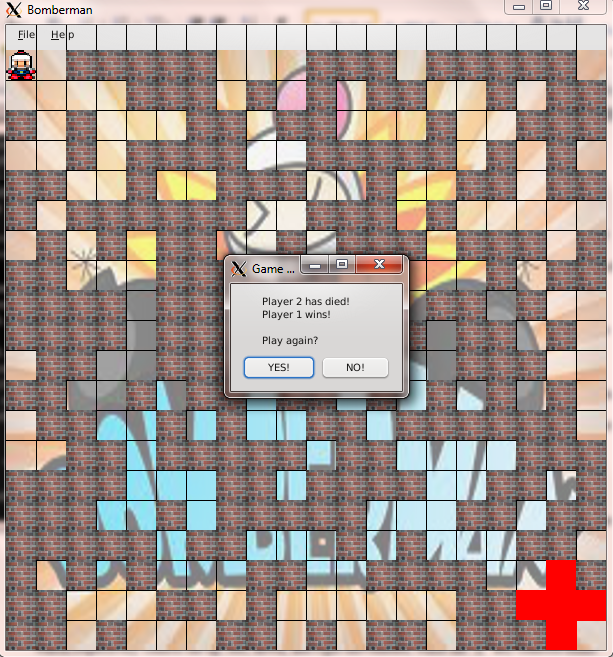
Joe Fetsch

Professor Scott Emrich

Fundamentals of Computing, Spring 2012 (CSE 20212)

May 2, 2012

For our final project, we decided to develop the popular computer game, *Bomberman*. *Bomberman* is a simple video game in which users compete by strategically placing bombs on a course in order to kill their opponent, while avoiding danger to themselves, or to destroy obstacles in the path of the user. Exploding bombs are also capable of detonating other bombs or destroying the first block encountered in a given direction. If a user is in the path of a bomb, i.e. the same row or column, and no obstacles separate the user from the explosion, the user’s avatar is killed, and the user consequently loses. After encountering and destroying a block, explosions will no longer propagate in that direction. Explosions only propagate in four directions, up, down, left, and right. That is to say, explosions do not explode circularly but instead form a plus sign.



A screen shot of *Bomberman* after a bomb has exploded and a winner has been declared.

The version simulated in this *Bomberman*interpretation is a two player competitive game in which player movements are controlled by keyboard inputs. While extending the game to support more than two players would not have been difficult, we chose to limit the number of players to two as practically speaking, fitting more than two individuals on one keyboard would be difficult and uncomfortable for the users. Player One controls his character using the “W”, “A”, “S”, and “D” keys, which respectively control the up, left, down, and right movements, while the “E” key controls bomb placement. Similarly, Player Two controls his character’s movement using the arrow keys, controlling their respective directions, and places bombs using the “/” key. Both users can have up to three bombs on the board at any given time, and each bomb detonates after three seconds, barring an early detonation caused by another bomb’s explosion.

The mechanics of our *Bomberman* game were implemented using multiple C++ classes. Perhaps the most important class for the implementation of the game is the *board* class. The board class provides the basis for storing information about the map, more specifically, the objects placed on it. The board class is composed of *cell* classes, which in turn have two subclasses that inherit from cell. These two classes are *bomb* and *block*. These two subclasses represent locations at which bombs and blocks are placed on the board. There is also a *man* class which represents each player’s avatar and contains information about their current life status. The full relationship between our classes can be seen most clearly in our UML diagram appended to this report.

These classes all interact and interface with the user using the QT graphics library. This interaction primarily takes place in the “mainwindow” file. We used QT to provide output to the user in terms of creating and updating a display. QT allowed us to import images for the background, blocks, and players to give the game a more user-friendly and exciting experience. In addition, the QT library assisted in the reception of input through key presses which drive the main movement of each player and placement of bombs throughout the game as well as supporting in the manipulation of menu items and message boxes.

Our program has shown no issues or bugs, and no problems have been encountered since finalizing the *Bomberman* simulation. We have tested and proven our code to work on both the Linux lab machines and on a PC remotely accessing the Notre Dame student machines.

In conclusion, we used QT to create and implement a *Bomberman* game in C++ utilizing the major tenants of object-oriented programming. Our final product is fully functional and operates as desired.

**UML Diagram:**

