Final Project Report

Paul and Ken’s Bomber Man

**Summary:**

Bomberman is a simple 2D Game where the object is to destroy destroyable blocks to find the entrance to the next level. Players are vulnerable to any explosions, including those caused by their own bombs, and they are vulnerable to monsters. Coming into contact with these vulnerabilities will cause the player to lose a life. When both players have lost their lives the game ends.

**Classes:**

Managers

**ContentManager** – The content manager is a static class that stores the game content that is loaded during the LoadContent method of Game1. This class is used to easily access Textures and Sounds throughout the solution.

**CollisionManager** – The collision manager holds most of the logic regarding collision. It prevents players from moving inside walls and detects collisions between players and monsters and explosions.

**KeyBindings** – the keyBindings class binds keys to actions that can be performed, mainly movement and bomb dropping. The KeyBindings object is then passed to a player object when a player is constructed so that the player may use the bindings. This simplifies multiplayer functionality by keeping the controls for different players separate from the player class itself.

Game Objects

**GridCell**: An array of grid cells make up the backbone of the game. Grid cells are used to keep track of where walls are, whether or not the wall is destructible etc. An array is used to easily check for objects neighboring each other. Each GridCell holds a Vector2 representing the coordinates of the grid in the array, a Rectangle representing the space on the screen occupied by the cell, and awall (possibly null of the grid space is empty)

**Bomb:** When a bomb is dropped, it is placed at the center of the GridCell that the player is standing on. The array is then checked for the adjacent GridCells in four directions around the bomb. If there is a destructible wall, or a null value in the GridCells Wall property, then an explosion is added to the corresponding grid space. The bomb also calls the collision manager to check if there is a player standing where the explosion is to happen.

**Explosion:** Animates a portion of an explosion. Several explosion objects are created when a bomb explodes, so that the shape of the explosion can be more controlled. For example, if a bomb is surrounded by non-destructible walls, the explosion will not

**Player:** Represents a player character. The score and lives for all players are shared as static properties, the game is cooperative. The player class handles the movement and animation of a player character, and drops bombs.