Special ball design - group 66

This document describes the design of a special ball using responsibility driven design and UML. Starting with the drafted requirements.

Functional Requirements

Must haves

- The Cannon must be able to shoot a special ball
- The special ball must attach to the grid of balls when shot
- When the special ball hits another special ball, all the balls surrounding the special balls must pop according to the normal rules
    - Normal rules mean balls pop if there are three or more adjacent balls of the same color
    - The special ball acts as a same colored ball for all adjacent balls

Should haves

- The special ball should look like a bomb
- When the special ball is hit by another special ball, the overall score should be doubled

Could haves

- The special ball could have an animation when it pops
- The special ball could have different variants

Would/Won’t haves

- The special ball won’t have another size than the normal balls

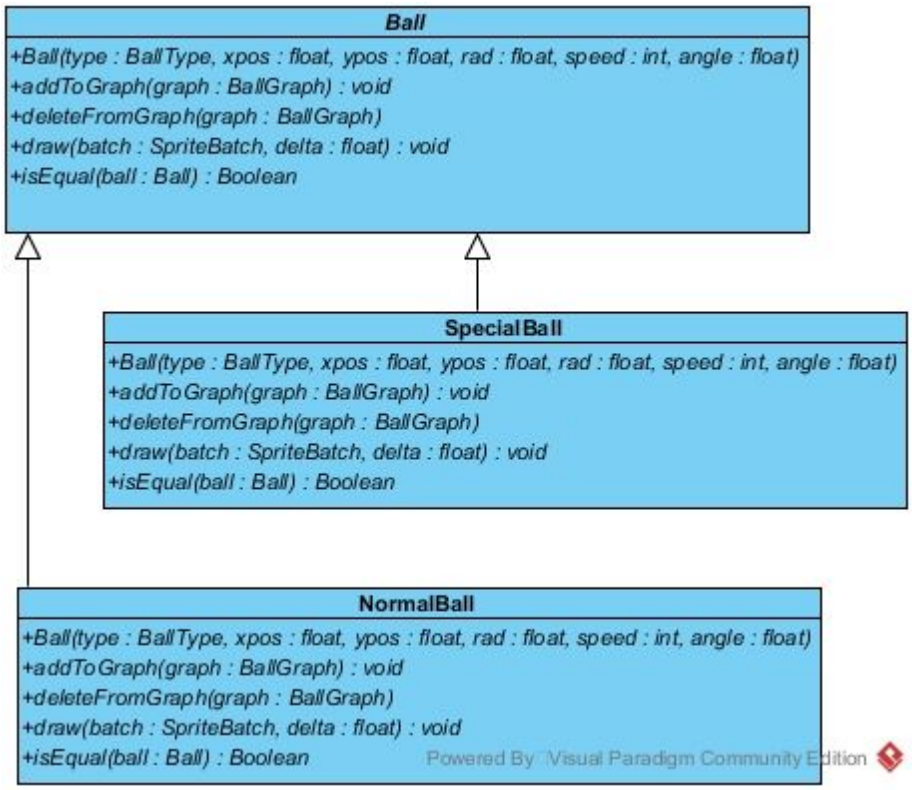
Non-functional Requirements

The Special balls will be implemented according to the Factory Method design pattern which has an abstract class “Ball” and can be implemented as a normal ball and as a special ball.

Class Responsibility Collaborator (CRC)

Class	**Responsibility**	**Collaborations**
Ball	Defines an abstract ball, which can be extended in various ways.	ColoredBall, SpecialBall, TopBall
ColoredBall	Draws the colored ball in the game. And defines to which other balls its equal.	BallGraph, BallManager, Cannon
SpecialBall	Draws the special ball in the game. And defines to which other balls it is equal.	BallGraph, BallManager, Cannon
TopBall	Makes sure other balls can connect to it, but is not equal to any other ball	BallGraph, BallManager, Cannon

UML Diagram



Technical Design and Implementation

To implement this feature the design of the balls in the game had to be changed. Because of future features it is good to make the design reusable. Therefore the ball is now implemented according to the Factory design method. It is now possible to make different kind of balls that will all work with the classes of BallManger and BallGraph.

The method isEqual is added to the balls to define to which type of balls a ball is equal. In this way special functions can be implemented without changing the BallGraph class.