Cake Blob - Assignment 3: Code Review

Bad Code Smell #1: Code Duplication

A major smell we found in our GameBoard class was a duplication of a method for checking around an object for walls. To fix this redundancy, combined the checkWalls() and enemyCheckWalls() into one general checkWalls(), and moved it into their parent class Entities. This allows both the player and enemy class to inherit the checkWalls() method. Next, since the gameWalls arraylist is globalized in the GameBoard class, to access this arrayList, we added the ArrayList<Wall> gameWalls as a parameter to the checkWalls(ArrayList<Wall> gameWall) method. Now we are able to simply call the method for the player, and all the indexed enemies.

Bad Code Smell #2: Feature Envy

After having combined the enemyCheckWalls() and checkWalls(), this method makes more sense if it is placed under the Entities class rather than the GameBoard class. The reason for this is it allows both subclasses to inherit the method rather than having to create two independent methods or one giant method that handles both of their checkWalls mechanics. As a result, the code duplication aided in solving a feature envy issue as well.

Furthermore, from GameBoard class, we have decided to move the enemyKillPlayer() and enemyCheckEnemies() methods to Enemy Class. These methods seem more innate to the moving enemy only, and reduces the amount of features our GameBoard class is attempting to handle. Furthermore, this aids in reusability of the methods if we were to add more moving enemies or even different types of moving enemies in the future.

In addition, from GameBoard class, we also moved enemyDirection() to Enemy class. The method now takes a parameter of type Player as it is dependent on the players' position, and establishes reusability for the method when testing.

Bad Code Smell #3: Classes that are too large and/or try to do too much

When we were refactoring our code as a result of feature envy, we moved both enemyCheckWalls() and checkWalls() methods out of our GameBoard class. Additionally, we moved enemyKillPlayer(), enemyCheckEnemies(), and enemyDirection() methods to the Enemy class. In doing so, we were also able to clean up our GameBoard.java class, which has many of our functions in it, and is quite large. However, because this class acts as the “brain” of our program, we haven’t tried to make it too small either as we want it to include methods that are necessary for the program to run, and not have too high coupling between our various classes.

Bad Code Smell #4: Lack of Documentation

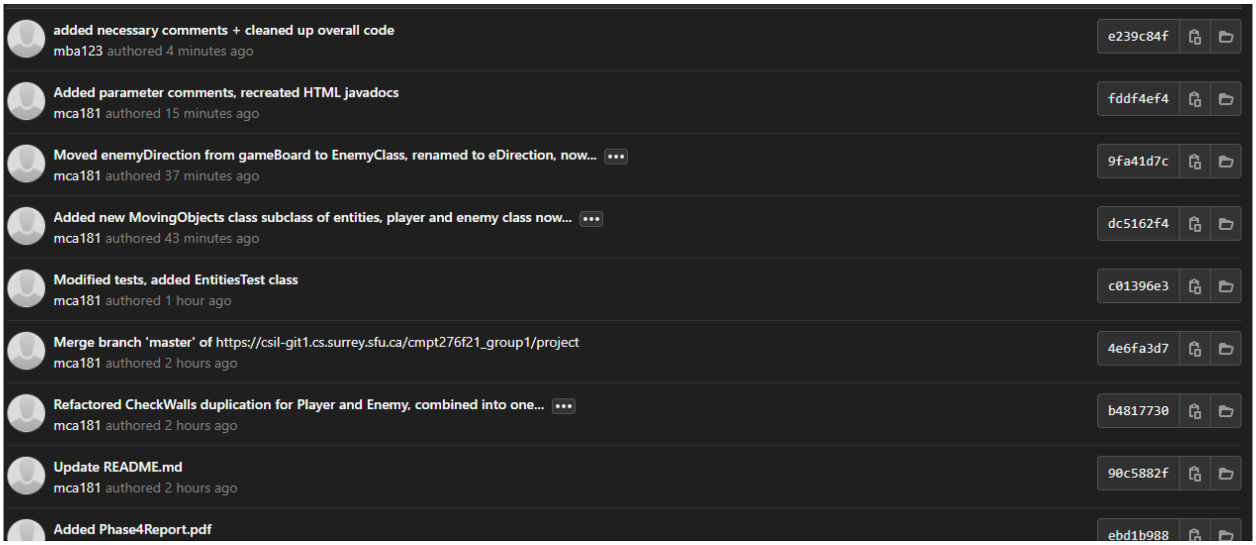
When creating and submitting our game for phase 2, we did create Javadocs comments for most of our classes and methods, however, a couple were ignored. We have added additional comments for these. Additionally, in this refactoring process we have created some new functions and classes. As a result, we have provided documentation for our eDirection() method and the MovingObjects class (including the new checkWalls() Method inside).

Also, in our refactoring process we added parameters to functions that we didn't have previously (in order to be able to move them to a better class, etc.) - and so we added documentation for these additional parameters.

Bad Code Smell #5: Refused Bequest

Some of the data values and methods in the Entities class were not being used by the classes inheriting the Entities class. The boolean values used for wall detection and the checkWall() method was only needed for moving objects - the Player and the Enemy class. So, to fix this bad code smell, we created a subclass movingObjects() and moved those data values and the method to that class. Furthermore, we made the moving objects - the player and the enemies - to extend the new inherit instead of the main Entities class.

Git history for refactoring (all included in this list):



added necessary comments + cleaned up overall code
mba123 authored 4 minutes ago — e239c84f

Added parameter comments, recreated HTML javadocs
mca181 authored 15 minutes ago — fddf4ef4

Moved enemyDirection from gameBoard to EnemyClass, renamed to eDirection, now... •••
mca181 authored 37 minutes ago — 9fa41d7c

Added new MovingObjects class subclass of entities, player and enemy class now... •••
mca181 authored 43 minutes ago — dc5162f4

Modified tests, added EntitiesTest class
mca181 authored 1 hour ago — c01396e3

Merge branch 'master' of https://csil-git1.cs.surrey.sfu.ca/cmpt276f21_group1/project
mca181 authored 2 hours ago — 4e6fa3d7

Refactored CheckWalls duplication for Player and Enemy, combined into one... •••
mca181 authored 2 hours ago — b4817730

Update README.md
mca181 authored 2 hours ago — 90c5882f

Added Phase4Report.pdf — ebd1b988