

## Group 4

Regan Li

Zhaoyue Yuan

Code Smell 1 – detectEnemyCatchPlayer method should be moved to CollisionDetector.java

Commit ID: 3c2eca90

**Problem:** detectEnemyCatchPlayer method is used to check whether enemies have capture player, it is in Screen class. According to the single responsibility principle in the design principles We should move this method to CollisionDetector class, because CollisionDetector class is responsible for detecting collision conflicts in game.

**Fix:**

Move detectEnemyCatchPlayer method from Screen class to CollisionDetector class, and update related method call code.

Code Smell 2 – remove directionEntered variable from AStarFindPath.java

Commit ID: 3c2eca90

**Problem:**

In the original design, the directionEntered variable was used to record the direction of the next node to be reached, but later, the coordinates of the next node were directly used instead of the direction, so the directionEntered variable is actually useless and needs to be deleted.

**Fix:**

Remove the directionEntered variable from the ExpansionList constructor in the AStarFindPath.java file, and update the ExpansionList constructor call in AStarFindPath.solve method.

Code Smell 3 – Change the name of the AStarFindPath class to Pathfinder

Commit ID: 3c2eca90

**Problem:**

AStarFindPath is a sentence that is not suitable as a class name, while Pathfinder is a noun that is more suitable as a class name.

**Fix:**

Rename AStarFindPath to Pathfinder

Code Smell 4 – Adjust the construct of Pathfinder

Commit ID:057d8c7c

**Problem:**

In the original design, a Pathfinder object was created every time when the shortest path need to be searched, resulting in a large amount of memory overhead. However, in fact, only need to update start and goal to find shortest path.

**FIX:**

PathFinder retains only one construct parameter maze.

Code Smell 5 – Change the void solve() method in Pathfinder to LinkedList<int[]> shortestPath (int [] start, int [] goal)

Commit ID: 7bf364dc

**Problem:**

The original solve() method did not have parameters, and each path finding requires recreating Pathfinder object. Now, it is passed in start and goal instead. In addition, the original solve method did not have a return value. To obtain the path, you should also call GetPath method. Now just need call shortestPath method to get path.

**FIX:**

Change method name, parameters, return values, and related call. After finish compute optimal path, backward find all nodes visited from source to target.

Code Smell 6 – change public LinkedList<int[]> getPath() in Pathfinder to private LinkedList<int[]> solution2Path(ExpansionList solution) {

Commit ID:7bf364dc

**Problem:**

The original getPath method has no parameters and uses the pathfinding results of the entire class for path calculation. After changing the pathfinding method to a single call, it is necessary to change the method to an algorithm that backtracks the path based on the target value

**FIX:**

Cooperate with the shortestPath method to receive the target node as a parameter and trace back from the target node to the source node