

Assignment 3: Code Review

Hanjie Liu and Harmeet Rai

Change in Class Hierarchy

Smell: Enemy behavior and Collision class hierarchy is confusing

The Enemy class has its own collision class, but it was the same function as player collision. We combine all collision functions into the Player collision class, so every collision that happened on board will behold by the Player class rather than the Enemy or Lava pit itself. This change was made in PlayerCollision.java, EnemyCollision.java, and LavaPitCollision.java. Files in commit ID 15a7bcd6.

Unused Class

Smell: Redundant class that has been neutralized

The LavapitCollision.java and EnemyCollision.java should be removed because it has been neutralized as we change the Hierarchy on the whole collision implementation. It will be never used and redundant in our program, and PlayerCollision.java has been doing their work.

MovableEntity.java, which is an interface, is never used. There is only an Enemy entity that can move(except the player) so Movable is the unique attribute within Enemy, no need to use an interface. There are unused classes in treasure.java, Medkit.java, levell.java, lavapit.java, ground.java, gamecontroller.java, enemy.java, DemonTower.java, Demon.java

File commit ID: dfb91900

Unused Variables

Smell: Remove unused variable

- Square.java has unused variable tempEntities
- GameController.java has unused variable emptyHeart

File commit ID: dfb91900

Unused Method

Smell: Method never called

There is a class "EnemyAnimation()" in Enemy class is never called. And also getSpeed(), setSpeed() in Demon.java are never called as Speed Attribute is never used. The changes are made in Enemy.java and Demon.java, commit ID: 15a7bcd6.

Lack of documentation

Smell: lack of documentation on Demon class.

We found we lack a lot of documentation on Demon class, especially Movement implementation and Image loading. More useful comments has been added above the function in order to help comprehensive the code.

Commit ID: 15a7bcd6

Class refectory

Demon.class getPositionClosestToPlayer() method is used to get the next position of the demon, including up, down, left, right and motionless 5 kinds of movement position.

For the 4 directions of up, down, left and right, the demon position (enemyX, enemyY) and direction need to be modified frequently, which leads to redundant code of getPositionClosestToPlayer(), and it is easy to write the parameters wrongly which leads to wrong calculation. To avoid the above two problems, we can extract the method for the next operation of demon, and the result is as follows:

```
``java
// current square with smaller dist
private Square closestSquare;
private Square closestSquare; /**
 * update closestSquare and smallDist if square with the direction has smaller dist
 * @param squares game map
 * @param smallDist current small dist
 * @param direction current direction
 * @return
 */
double next(Square[][] squares,double smallDist,Direction direction){
    // next position of enemy
    int enemyY = getYIndex()+direction.getY();
    int enemyX = getXIndex()+direction.getX();
    // next position of enemy is valid?
    if(!valid(squares,enemyY,enemyX)){
        return smallDist;
    }
    // position of player
    Player player = Player.objectify();
    int playerY = player.getXIndex();
    int playerX = player.getYIndex();
    // calculate distance
    double nextDist = calcDist( playerX,playerY, enemyX, enemyY);
    // update if current distance is smaller
    if (nextDist < smallDist) {
        // update smalldist
        smallDist=nextDist;
        // update closest square
        closestSquare = squares[enemyY][enemyX];
        // update direction
        setDirection(direction);
    }
    return smallDist;
}
```