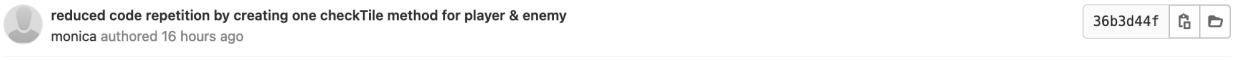
Assignment 3: Code Review - Kirby & Monica

CollisionHandler.java

Code duplication:

- Our CollisionHandler class had two separate checkTile() methods:
    - checkTile(Player entity), for checking if the player collided with a tile
    - checkTile(Enemy entity), for checking if an enemy collided with a tile
- These methods were identical except for the parameter type
- Since Player and Enemy are both subclasses of Character, we reduced code duplication by combining these methods into one checkTile method under the Character superclass

Reference the commit below to see the changes:



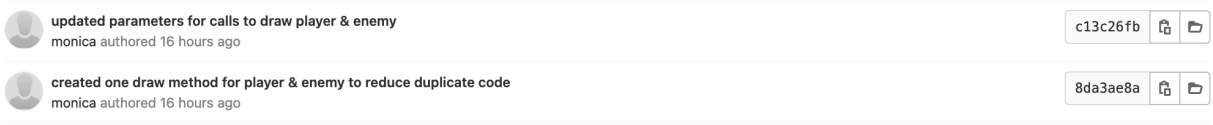
Player.java and Enemy.java

Code duplication:

Both classes had a draw() method that was nearly identical, except for one statement in the Player draw() where there is an animation when it collides with a punishment.

We replaced the draw methods in player & enemy with one draw method for both in Character, reducing the duplicated code. We added an if statement in the new method for the Player animation.

Reference the commit below to see the changes:

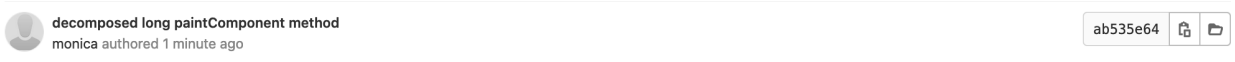


Gamescreen.java

- ***Long method***
- ***Complicated Switch statement:***

This class has a method called paintComponent which contains a switch statement with 7 cases (different game states). Because of the number of cases the method was long and difficult to understand. We extracted methods to execute the commands under each case, making the paintComponent method shorter and easier to understand.

Reference the commit below to see the changes:



GameScreen.java, MouseInput.java, GameOverMenu.java, GamePauseMenu.java, TitleScreenPanel.java, and GameWinMenu.

Duplication: Restart & player states have a lot of repeated code

- The for loops within the switch statement “RESTART” have been deleted. While it was repeated code, it also served no purpose since the loops would run on the “PLAYING” state regardless.

High coupling - needs a lot of classes to work (hurts testability)
Reduce coupling: grouped some of the dependencies together and/or created bigger abstractions. Our overall refactoring of code is what helped reduce the coupling, which can be found in all of our commits past Dec.8.

Unused imports and variables(check warnings in intellij)
Removed excess imports that were never used (eg. javax.imageio.ImageIO, game.objects.Exit, java.awt.image.BufferedImage, and java.io.FileInputStream)

- These imports were covered by simply importing the entire package in one line via the “*” character (eg. import java.awt.*)
- Removed unused variables in game state screens (eg String title, String developers, int buttonWidth, and int spacing never used in all states).

Refactoring code and ***Poor documentation***(via function creation to control override functions and fill unnecessary auto blocks)

Instead of having the main function operate under implemented mousePressed(), I decided to create a function in line with the keyHandler() that takes in the truth values and mouseInputs. This way I can utilize the other implemented mouse functions rather than having them stay empty as required overrides. Also added comments to specify the if statements' purpose.