Homework 3 Code Review

Dylan Rowsell and Gray Keng

For this assignment, both teammates met in person and performed a manual code review over each other's code. Our group managed to find 10 issues with the code in which we needed to refactor.

#1. Finding a Center Pixel | Type: Duplicate Code | Commit: e57cdbe4
Problem:

Developers are often manually calculating the center x pixel when adding elements to the UI screen.

Solution:

A method getCentreX(int textLength) was created to return a centered X pixel for the given text. Also the getTextLength(String text, Graphics2D g2) method was created to get the text length for the method getCentreX(getTextLength(text, g2)). After this change, we can use the method we created about 10 times without duplicate code.

#2. Long Method in ItemDetection | Type: Long Method | Commit: 7ce1776a
Problem:

A large if / else statement was found inside the method onItem in the ItemDetection class which detects if the player is on an appropriate item. This made the onItem class very long.

Solution:

A new method called playerItemInteraction was created inside of the onItem method in order to reduce the size of the onItem method.

#3. Refactor Player Movement | Type: Duplicate Code | Commit: f42a2591
Problem:

The PlayerActor class, Skeleton class, and CharacterMovementThread class had long snippets of code that were identical to all when attempting to move a character.

Solution:

A new method called safeCharacterMove() was created to replace the identical code. This method allowed the player to move and collide with obstacles.

#4. Sounds Import | Type: Duplicate Code | Commit: 4fb2116d
Problem:

The getClass().getResource(“path”) is called every time we need to import a sound effect.

Solution:

The variable String soundPath[] was created to store the paths of the sound effect and the BGM instead of directly storing to URL soundURL[]. A for loop was used to store the paths to the URL instances.

#5. Zombie Rush | Type: Duplicate Code | Commit: 396064dd

Problem:

The Zombie class has the function rush(Direction direction) which is very similar to moveCharacter(Direction direction).

Solution:

Rush was adapted to use moveCharacter(Direction direction) and the method moveCharacter was modified to have to change an “isClear” boolean variable to flag in the Zombies’ rush method.

#6. Unnecessary Messages | Type: Dead Code | Commit: df2dfb30 and 3b34f74a

Problem:

There are some messages printed out while the game is running. The message was for testing use only.

Solution:

The print commands were found under different classes and deleted.

#7. Get Image | Type: Poorly Structured Code | Commit: 6e184a1f

Problem:

The PlayerActor class’s getImages() method was poorly structured and could not be scaled well.

Solution:

I created arrays to put the images into which allowed for the tile images to be loaded with a for a loop. The code now takes up much less space and it is easier to add new tiles.

#8. Merge Duplicate if Statements | Type: Duplicate Code | Commit: 7ce1776a

Problem:

There are multiple if statements that check if the game state is PLAY inside the ItemDetection class’s onItem method.

Solution:

The if statements were merged to only one statement on the top of the code. If the game state is not PLAY, it’ll stop executing the code after.

#9. Get Tile | Type: Poorly Structured Code | Commit: 8a4d5d29

Problem:

The CreateBackground class’s getImages() method was poorly structured and could not be scaled well.

Solution:

An array was created to put the images into which allowed for the sprite images to be loaded with a for loop. Future sprite image files can now be added to these arrays.

#10. Remove Item Limit from Level Class | Type: Dead Code | Commit: 92ec2688

Problem:

The method getItemLimit() inside of the Level class was not being used.

Solution:

The method getItemLimit() was removed from the code.