<u>**Assignment #3:**</u> Code Review report

CMPT276

Sameer Hossain: ([sameerh@sfu.ca](mailto:sameerh@sfu.ca)) | Dina Zeng: ([dhz1@sfu.ca](mailto:dhz1@sfu.ca))

## <u>Refactor 1:</u>

Entity classes hold many attributes which are public like screenX and screenY. They were changed to private and code was restructured to get access to the new private members to encapsulate the classes.

Deleted unused code chunks and some debug methods. Unneeded comments were deleted and added more JavaDocs to explain functions. This made the code cleaner and easier to understand. Classes involved in this refactor included Bat, Skeleton, Enemy, Player, and Slime.

**Commit SHA:** 7f257d12756c8ee1141a3683fe3f9f6090bbcf1d

## <u>Refactor 2:</u>

The *update()* method in *Player* class uses many different tasks like updating player position on the map, direction, sprites and checking collisions between the player and other objects/entities.

Hence we split the method into smaller chunks and created helper functions that would be called by *update()* which made the code cleaner and easier to read.

**Commit SHA:** 16f823f51c64b53b8c24e7b0b709ff35e6b28f4e

## <u>Refactor 3:</u>

Corrected encapsulation for the UI and game board classes. Many of the attributes were public and so we made helper functions (getters/setters) to gain access to them.

**Commit SHA:** d1313376437eaffaaf3a5cd994bb272ee1d5bbc6

**Refactor 4:**

The *CollisionChecker* class has duplicate code chunks in methods *checkItem*, *checkEntity*. These methods move the entity's hitbox along with the entity itself to check if it will collide with other entity and object hitboxes.

Therefore we made a *moveHitbox()* helper function which is called by the methods to achieve the desired results instead of copy pasting the same code several times. This allows us to make changes in one function to change multiple functions which makes our code cleaner and more organized.

**Commit SHA:** 70e2a027c605a675d337018260aa9c7f63569398

**Refactor 5:**

*BonusReward* had an unused variable, *ticksTillDeath*. This variable also had a getter and setter methods as well as a method used to decrement its count. The methods were also unused throughout the code.

To remove the bad smell, we deleted all methods associated with *ticksTillDeath* and *ticksTillDeath*. All of the deleted code is located in the BonusReward class.

**Commit SHA:** 5c76d0f4c66f31ddd01a9d787444333e19de6d02

**Refactor 6:**

In the class *GamePanel,* multiple variables such as *tileSize, screenWidth,* and *screenHeight* were highly coupled across multiple classes such as *Items* and *CollisionChecker*. They were being accessed multiple times in multiple methods and in some instances, were used in other classes entirely or for the most part.

To resolve the coupling and optimize the code, these variables were refactored into static final variables as they do not change throughout the entirety of the program. All variables are then put into a static Class *GameSettings* that keeps track of the game settings.

**Commit SHA:** f56ff3ae90d156bc48a61c8769bd0f51ee5635b8