Peer Review for group-5

Reviewed by Group 19

Positives

The project is very large with several packages, it seems they use several beneficial design patterns such as factory and singleton and follow design principles like composition over inheritance. The code is well documented in several places and is not overly commented in places where it would not be necessary. Proper names are used according to the design patterns that are used, making the code easy to comprehend. Furthermore the coding style is consistent throughout the project, thus also increasing the comprehensibility of the code.

The project is split into many packages with no unnecessary dependencies between Model, View and controller as we assume the project uses the MVC design pattern. The apparent modular design seems to aid in the ease to add and remove functionality. As there are no complicated algorithms and it is a 2D program it is not expected to have performance issues and we found none, even with many entities on screen at once.

There are a number of interfaces which provide good abstractions for the different objects in the game. For example, IBlock and IEnemyType are examples where the underlying objects are not shown and instead the code is written in terms of the interfaces.

Points to work further on

At the current state of the project as we are reviewing it, the model has some external dependencies on several libraries. For instance, there are dependencies on javafx on most of the classes in the enemyTypes package . Furthermore there are dependencies on fxgl in the classes included in the package blocks. This means that the model is extremely reliant on these external libraries which is not very good, this is by far the most important problem we discovered with the project. This also leads to the code not being as reusable as it could be. Because these libraries are utilised in so many parts of the model, some changes or updates such as deprecated methods on these libraries would lead to change being necessary in many classes.

As of now the testing of the program is fairly rudimentary so it would also be great to see that the model has full test coverage, or atleast testing for the more complicated logic.

The RAD and SDD are lacking in some aspects such as the RAD missing class responsibilities and SDD only contains an UML diagram of the model.

The extensive use of the external library fxgl some methods can be slightly harder to understand than others but the library contains good naming and therefore it is not a major problem.

The abstraction MenuController is not used as an abstraction but instead only for code reuse but the abstract class does not contain much code as well as an abstract method which is odd.