Group 10 - Phase 2 - Deliverables Report

– Describe your overall approach to implementing the game.

We divided our work among all the members. We needed to create all the classes after referring to our UML diagram so each team member had been assigned to creating and implementing different classes and use cases. We decided to create and code our classes individually and we tried to stay updated with each other on the work progress. Then we discussed how the classes would merge and integrate with each other. Periodically, everyone created certain methods and classes, then we would bring everything together. This would allow us to use the different methods and fields from the entire program.

– State and justify the adjustments and modifications to the initial design of the project (shown in class diagrams and use cases from Phase 1).

There were a few changes in the fields from individual classes. New fields were added and some were removed. We implemented a few changes in how every class interacts with other classes and how the classes are connected to each other. There are new classes added now which make for a good design pattern, and help us organize our code a little better. We added a static Helper class that allowed us to easily call frequently used functions and reduce the length of function calls to singleton classes.

– Explain the management process of this phase and the division of roles and responsibilities

For the management process of this phase, we decided that each of us would try using the same IDE and UI for our project so that there is consistency throughout. And this way we would be able to help each other with any problems or doubts along the way.

The next step was to divide and distribute the work. Then we started working on coding together and made several mutual decisions on different aspects of software development for our game. We knew that the work for the project could not be divided equally among each of us, but as soon as someone would finish working on one part they would move on to working on the next part or move on to helping another one of our teammates.

Dates	Meeting notes
Feb. 28th	First meeting ● Get familiar with Maven and Javadoc ● Use assignment 2 to test and get used to design-patterns
Mar. 04th	Second meeting: Separating features and working on different branches ● Board (Karim) ● GameManager(Herbert)

	● Entities(Maggie) ● EnemiesMovements (Tony)
Mar. 12th	Third meeting: Updates on process Adding updates to the report Tony ● pom.xml ● AIPathManager class with implementation of A* pathfinding Maggie ● Implements draft classes (punishment, rewards and bonus rewards) Karim ● Implements classes for board, cells, and walls Herb ● Implements GameManager and UI **There were short meetings along the way to help each other out with clarifying different requirements and implementing the different functionality and classes.*
Mar. 20th	Writing deliverables as a group Creating merge requests
Mar. 24th	Finalizing the report Creating merge requests

– list external libraries you used, for instance for the GUI and briefly justify the reason(s) for choosing the libraries

Java Abstract Window Toolkit
Java.awt.event - Used to handle key inputs and issue suitable output
Java.awt.Image - Used to load images and animation

– describe the measures you took to enhance the quality of your code

We pushed our individual work to branches, which are centred around specific features, such as the grid/board, or movable entities. Once a feature is complete, we review it with at least one other teammate for feedback and to check compatibility with existing code. The feature branch is rebased onto the master branch and merge conflicts are resolved, tested for errors, and any bugs are fixed. Then the branch is pushed onto the master branch, ensuring the best possible code for everyone to work with.

We peer-reviewed each other's code to come to a consensus on the best and most clean implementation of the code. Since we are aware of each other's code, we were able to discuss and remove redundant classes and functions.

– and discuss the biggest challenges you faced during this phase

1. It was a little unclear to make the initial decisions on how to work on the coding part, and it was hard to see a clear picture of the implementation of the game.
2. Some of us had a good idea of how to go through the development of the game and they tried to explain to the others. It was difficult to grasp certain concepts right away, but we learned a lot of things eventually.
3. Getting everyone to work through the same GitLab repository, using branches.
4. All the code needed to be merged and integrated with each other.
5. Merge requests were a new concept for some of us and it took a while before we were able to pull, push, and merge easily. It can take some time to learn how to keep our program up to date together with everyone in the team.