Phase two Report

Overall approach to implementing the game:

Our goal for this section of the assignment was to follow our previously created UML diagram and implement each class and object according to the said diagram. We did anticipate that there might be places where we need to add or modify certain things but for the most part the plan was to follow the UML diagram. After a couple of online meetings to coordinate everything we divided the UML diagram among ourselves and started to work on our own respective sections. We also had online meetings on a need basis to resolve any issues during the coding of our respective parts. Furthermore, we had a minimum of one weekly meeting to make sure we stayed coordinated and on track. Other than the weekly meeting we frequently communicated virtually when working on the code to resolve various coding and other issue. As a last measure to make sure we didn't fall behind schedule we created a rough checklist of all the things that we needed to do with regards to the project, we maintained this list mainly in our discord group and added to this list as we went along, we then removed any of the tasks that we completed.

Adjustments and modifications to the initial design of the project:

As we got deeper and deeper in to the making of each of the classes we had to modify parts of our UML diagram, most of the classes remained the same and their relations to one another were also mostly maintained. We did however rename some of the classes so that they better represent what they do. Some examples of the changes that we made include but are not limited to: changing the name of classes like renaming our main board to game panel, breaking down certain classes into other sub-classes so that we could more easily implement them, an example of this would be breaking down the character class to include: enemy,enemy generator and a player class. Other examples of classes we added that were not originally on our UML diagram include the key handler class for better dealing with keyboard key presses, and the level generator class. All of the above modifications were made to make the implementation of the project easier specifically for working in a group environment. Most of the main ideas however remained the same and were implemented very closely to the first UML diagram.

Phase two Report

Management process of this phase and the division of roles and responsibilities:

Like mentioned above we hand a few initial meetings to coordinate ourselves and talk about how we plan to go forward with the making of the project. After our initial meetings we decided to have a minimum of one meeting per week for scheduling and updating each other about the progress we each had made. We also had meetings on a need basis when ever there was an issue with any of the coding. As for the division of labour we initially started out by breaking down our UML diagram classes in to four roughly equal sections with respect to the amount of coding needed, we also tried to break up the diagram in to sections with related classes. After doing so we each started working on our respective parts and pushed completed sections to git lab. As we went further though it became apparent that there was more overlap among our parts then originally anticipated, so near the second half of our development we all worked on any uncompleted sections/classes that we could find. Moreover a lot of communication was done over discord and in our online meeting so whenever any of our group members needed help with there coding section we all tried our best to help out with that section.

External libraries you used, for instance for the GUI :

we only used a few external libraries other than the ones offered in java JDK 15 examples of a couple of them include Apache commons and the the google-json library which was used for parsing each of the levels when creating them at the start of each round. The external libraries used are not necessary but they helped us simplify the implementation of some of the code.

Phase two Report

Measures we took to enhance the quality of our code:

During the coding process we tries our best to write elegant and clean code to start with, however, sometimes this was not easily possible at first so all members of the group took part in correcting/ optimizing any section of code that they thought they could improve up on. Furthermore, we made temporary comments along the way to indicate what was changed and what the implications of this change might be. All this is separate from the permanent JAVADOCS we made for each method to make sure that our group members as well and other viewers could understand what the code does.

The biggest challenges we faced during this phase:

The biggest challenge we faced during this phase was the coordination among ourselves, we were all used to working on code on our own however most of us did not have much experience with working on something this big in a group, so initially it was somewhat difficult to get the hang of dividing the tasks up among ourselves and working together in a productive fashion. However, after the first few days of working on the project we got used to the workflow and it was easier to divide and conquer the remaining sections of the code. Near the end of the project we were quite comfortable with working with one-another and dividing up the work to be as efficient as possible.