Describe your overall approach to implementing the game

We first divide a role based on the UML diagram. One person implements character including player and enemy, one person implements a board and other two implement a game loop. As our role is not divided by functions but classes, each member first writes a stub function so that other people can work on related areas.

We basically used discord to communicate. We first notify what change will be made, and push to make sure two people are making the same change or working on the same function.

We also decide on a concept of the game, so that makes finding or creating design and sound resources easily.

State and justify the adjustments and modifications to the initial design of the project

1. Removed the BonusCell, RewardCell, PunishmentCell class. Decided to manage the type of the cells by array rather than individual classes for efficiency.
2. Added a background class. It is a class to draw tiles. We separate it from board class, so that background class just draws a map and board class can focus on showing other elements including player, enemy, time, scores and end screen.
3. Added a sound class to add a sound.
4. Removed the left, right, up, down barrier. Instead, we made a checkWall function in character class so that both player and enemy can't pass the wall.
5. Implemented setStartscreen and setEndscreen function, which was missed in phase 1.
6. Beside that, made a minor changes such as class name Cell -> Tiles

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

Ee Tan: Created enemy movement to follow player, player class, character class, death screen, end screen.

Hammad Javed: implemented the rewards, bonus and punishment system, handled graphic/audio resources and implemented them, set animations (item pick up/setting traps/character sprites, door), starting display, level design.

Jooeun park: Creating the board class, getting points and showing scores, displaying time, writing report

Parham Kianfarian: Game(main) class, player class (player movement, player tick), collision for characters.

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

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

Used to get an action of a player.

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

Used to allow users to control character by key.

import java.awt.image.BufferedImage;

import java.awt.image.ImageObserver;

import javax.imageio.ImageIO;

Used to render a raster image.

import java.util.ArrayList;

Used to manage tiles by array.

import javax.sound.sampled.;*

Used to manage the sound of the game.

import java.io.IOException;

Used to handle an exceptions

import java.io.FileInputStream;

Used to get a file (map.txt)

Describe the measures you took to enhance the quality of your code

First and foremost, we checked the clarity of the code, variable and function name, and annotations. If the code is easily understandable to all team members then we assume that it is clear enough code.

We also paid attention to high cohesion but low coupling function design. We separate functions into different classes according to what it does, such as separating background and board classes, and making less interaction between the classes.

discuss the biggest challenges you faced during this phase.

The biggest challenge was poor imperfect game design in the first phase. Since we didn't have any experience with making games without a game engine, the plan we made in phase 1 needed to be changed a lot. There weren't any necessary functions and there were unnecessary classes, so for the first week of phase2 was chaotic. We needed a bunch of new functions to create that we didn't expect. This sometimes caused confusion about what to make, and took a little time to deal with it.

The lack of time also becomes a challenge. It was even hard to have time to gather virtually because of the different time schedules. Thus we just work individually rather than having a meeting, but the overall game quality would be better if we have more time to focus on the project.

There were minor problems such as different understanding of requirements, but those are resolved easily with enough conversation.