CMPT 276: Introduction to Software Engineering

REPORT PHASE 4 - GROUP 09

Minh Phat Tran	301297286
Chun Hei Yiu	301329448
Steven Xia	301444281
Miles Burkholder	301405683

I. Introduction

In this phase, our group will finalize the process of developing the game project. Also, the overall description of planning stages for the project, solving problems as well as the most important lessons we have learned will be summarized in this report.

II. Tutorial

We chose to do a video tutorial for this phase, the link to the video is here: https://www.youtube.com/watch?v=3WhS6OGyz-0

III. Game Description

Our game “Easter Bunny Hunt” is a classic 2D arcade-style dungeon crawl game where the user plays as the Easter Bunny. The user navigates through a maze hoping to collect all the lost Easter eggs while dodging the hunter's traps as well as thorn bushes. The maze contains various moving enemies which the Easter Bunny must avoid such as Hunters, Bats and Wolves. Look out for wolves because they can smell where the Easter Bunny is and track them down. Bonus egg will randomly pop up and provide extra help for the Easter Bunny, like freezing all enemies for 5 seconds. The Easter Bunny needs to collect all their eggs to open a portal so they can escape safely back to Easter Island.

IV. How Our Final Product Differ From Our Original Design

Overall we have been very faithful to our original game design and our class hierarchy and their methods are very similar to those presented in our UML diagram. A notable change from our UML was the addition of a map class to handle reading and initializing the game map of objects(translating the text file short map into objects required many final attributes and this functionality was not needed in the maze class, only the array of objects was). Another notable change was the addition of the GameTimer class used to handle all timers for the game (since adding timer functionality in the maze made it messy and created a large data/attribute clump). We extracted these classes from the maze class in order to make it more cohesive, smaller, and less coupled. We also did this to allow for much easier testing since the individual classes would be more cohesive and testable.

As far as the game requirements we very minorly changed some features such as making the wolf track the player instead of the hunter (we thought it would make more sense since the wolf can smell) and having the traps be scattered throughout the map instead of the hunter dropping them randomly (this allowed us to pre-design the maps strategically placing the traps in locations that would make the game more challenging to the user as opposed to random). Other than these changes we included all other functionality promised in the requirements such as our bonus features.

V. **Important Lessons Learned**

For a few of our team members this was their first time using git and working on a software game with a team, our biggest takeaway while using git is that we needed to make more frequent commits. We would often work on larger functionality so we would not commit until we were satisfied however because of this we encountered annoying merge conflicts due to multiple people working on the same class. We also learned the importance of spending more time designing smaller cohesive components, while we worked very hard on our design we still were unable to foresee some of the problems that would arise in the future, mainly testability. Our design was focused around a large maze class that used all our other objects to put the game together however because it was handling too many things it became very difficult to test individual functionality as it wasn’t small and cohesive.

Finally, we learned the importance of communication. While working on a group project with many moving parts it’s crucial that members are all constantly kept up to speed. We accomplished this through frequent meetings and constant online presents through Discord. Our communication allowed us to work productively as a unit and make abrupt changes whenever necessary.

VI. Modifications since phase 3

In Phase 3, we completed the refactoring assignment and updated our code, some notable updates include creating a map class to handle generating the map of objects for our game, and we also added a game timer class to handle all our games timer functionality including bonus durations/respawn/deletion and in-game playing time. Both the game timer and map classes were extracted from the maze class to make it more cohesive and less large. By creating these two classes, we improve the encapsulation of information and decrease the high coupling in the original Maze Class.

VII. Conclusion

We are all very proud of our game “Easter Bunny Hunt” and are pleased that we were able to stick to our design, fulfill our initial requirements, and make our vision come to life. Creating this game allowed us to go through the whole software development cycle and learn from each step. The project equipped us with new skills for working as a group and as individuals.