

CMPT 276 Project - Group 17

Phase 4: Report and Presentation

Created By:
Jonas Lam
Yanjun Qian
Brendan Shen
Sheila Nicholson

The Game:

The premise of the game is that the main character is a student attending SFU. In the midst of finals season, they unintentionally slip into a deep slumber. Utterly exhausted from constant studying, assignment deadlines, and cramming of missed lectures, they slowly blink their eyes open and are shocked to find themselves lost in the deep, dark maze known as Robert C. Brown Hall. The player must evade a rabid bear, a zombie professor, or a failed exam to make their way out. With the help of copious amounts of caffeine, ever-moving A+ papers, and a cozy bed or two along the way, they just might be able to escape from the labyrinth that is Robert C. Brown Hall. That is, if they are not defeated by the many piles of books, vortexes that respawn the student's location, and the decelerating hidden smoke.

Game Objective: Collect all reward items (A+ papers, coffees, beds) and exit the level. The game ends if the player is caught by the enemy or if their score becomes negative.

For the most part, the final game remained true to the overarching theme of our original design. All characters, items, maps, and tiles outlined in Phase 1 were implemented as specified. However, there were several specifications from our original design that were ultimately not implemented. They are as follows:

- No options menu was implemented in the main menu, allowing the player to configure additional settings or find information about how to play the game.
- No final screen for losing the game was implemented. When the player wins, the final score and time elapsed are displayed using text over the game screen. When the player loses, the program abruptly ends, and the screen closes. Originally, we had planned for the player to be able to replay or return to the main menu once the game had ended.
- No sound effects were added to the game. Initially, we had planned to include background music, as well as sound effects for when the player encountered an item.

The UML diagram was also updated during the implementation of the game. Certain classes, such as HUD, PunishmentItem, and RewardItem, were removed, while others, such as endTile, NoteSteppableTile, TileManager, CollisionChecker, AssetSetter, GamePanel, Key package, GameTerminator, and UI, were added. For the most part, our original UML diagram was quite useful for the initial implementation but ultimately had some oversights due to our lack of knowledge about game development. Accordingly, irrelevant parts were omitted during development and refactoring stages.

The most important lessons learned during the planning, implementation, testing, and refactoring of this game were:

- Start early and maintain open communication. Especially during Phase 2 we should have tried to get the bulk of the implementation started earlier. It was a bit of a rush to finish Phase 2 but ultimately we were able to complete everything on time. Holding more

frequent meetings could have helped us avoid this situation. The halfway deadline was a good addition to pace ourselves but we think we could have done better in terms of scheduling - but we did alright!

- Recognizing the importance of researching online information and resources. As it was everyone's first time delving into game development, there were numerous knowledge gaps, likely contributing to our slow start in Phase 2. Ultimately we discovered and shared some very useful resources, but accessing them earlier could have resulted in a smoother progression.
- More knowledge of the project as a whole (all stages) would have helped in visualizing how we should code our game. We understood that the requirements were deliberately kept vague to allow for more creativity and flexibility, but having to find 8+ things to refactor for assignment 4 when we basically did our refactoring prior to learning what refactoring was did not help in terms of time management of the project in conjunction with other classes.

Tutorial:

- In addition to creating a JAR executable, one can boot up the game by pulling from github and running Main.java with maven installed appropriately. You will then be greeted with the menu screen, where you can choose from three difficulties, or an option to terminate the game by clicking the 'exit' button down below. Alternatively, simply closing the window in the top right with the X icon will suffice, as this game does not save data between runs. Use WASD to control your character (W = up, A = left, S = down, D = right), and collect all the beds, coffee (speed boost) and A+ papers (will randomly teleport around), while avoiding the pile of books, vortexes (will teleport you to a random location), mysterious smoke tiles (you have no way of knowing where it is until you hit it, when you will be significantly hindered - just remember!), and most importantly, the enemy/main boss of each level. Find your way to the exit through systematic navigation and tactical jukes of the enemy to escape, upon which you will be greeted by a congratulatory exit screen displaying your score and the total time elapsed. Try to get a record!
- The above is shown and described in full detail via subtitles (no audio) in the youtube video linked below.

****please mention that you can find the JAR file in the Game directory and the Javadoc stuff in Game/JavaDoc directory as per this line from phase 4 document:**

“You can also push these artifacts to your repository and specify their location in your report.”

How to Play - see video (thus, no screenshots):

<https://youtu.be/WK0A-dm7zJ8>