Alec Bailey

Naomi Bolotin

Object Oriented Design (CSCI 3353)

17 December 2023

Labyrinth Crawl:

Final Project Write Up

Overview:

Labyrinth Crawl is my original take on the “dungeon crawl” genre of video game which typically requires players to navigate some sort of maze-like structure, fighting enemies en route to the exit. Labyrinth Crawl uses the same general idea, combining some of the features I liked best from games I have played. The map becoming progressively more illuminated as you explore is my favorite feature and what I think is the game’s defining feature. I got the idea for it from Elden Ring’s map which is completely shrouded in fog until you explore more and more of it.

Running the Game:

From the Final Project directory run the command: javac -cp . builder/*.java command/*.java decorator/*.java observer/*.java factory/*.java

Then run the command: java builder.LabyrinthGame

The game is intuitive enough to figure out from there, your character is in the chamber marked with the “o” symbol and you want to progress through different corridors to other chambers, marked “#”, looking for the exit, marked “X”.

Patterns:

Builder:

I used the builder pattern to make the Labyrinth. I think it makes sense to have done in this context, you have a large labyrinth with corridors and chambers that needs to be set up. My initial idea was to have a completely randomly generated labyrinth where all the chambers would be laid down first and based on where those chambers fell the corridors would then be added to connect it all together and then the builder pattern's purpose would have been even more clear. I chose to back away from that plan because I did not think that I could create code to generate random maps that were as tricky or as fun as just having one preset map (at least not in a couple of days).

The builder is the nervous system of this project, it has to track the player's movements from the command class and update the visual representation of the player's location. It has to inform the listener of the player's new location and update the visual representation of the labyrinth accordingly. It has to let the Player class know if a move is possible by the rules of the game. The game is even run through the LabyrinthGame class in the builder folder.

The general basic idea is I created a grid-like structure (a 2D array). Chambers could be added at even indices (0, 0 or 4, 2 for example) leaving room for corridors to be added in between. The display method allows for the labyrinth to be printed out each time the player moves and it took a ton of fine tuning throughout the project to get to where it sits now.

Command: This pattern was used to control player movements through the labyrinth. It just made sense to use since navigating the player through the labyrinth requires repeated actions of the same type, not creation of objects. Its usefulness in this game would have been more clear if I

could have figured out putting a monster in each room because then I could've had more commands than just the move command (run command or fight command come to mind immediately), but that turned out to be outside the scope of this project. Getters and setters in the Player class made for easy manipulation of the player's location through the move command. I included the Invoker class because we were shown it in class but it definitely does not serve much purpose in the project as currently constituted. I think in the future it could be used if I were to have more advanced commands like dodgeCommand or something and a player could opt to dodge and then immediately attack and have both commands queued in the invoker to mimic a real battle where perhaps you cannot simply react to whatever your opponent does.

Observer: The observer was used in the most defining and unique feature of my game and is maybe the feature I am the most proud of. The map starts off almost completely dark and becomes more illuminated as you explore. In this case the labyrinth is the subject and the visibility manager is a subscriber, tracking the player's movement through the labyrinth and updating visibility of corridors and chambers in accordance with the rules I have set up. I think it makes for a cool first play experience, the initial confusion about the map being so bare is all wiped away when you figure out you have to move downward and two additional corridors and rooms appear.

Decorator: This is where a lot of my creativity is put to good use. I implemented the two basic types of monster and three possible modifiers, ashen, earthen, and frozen (an archaic term meaning cold or frosty). I had more modifiers planned like adorned and perfected which would've been combinable with any of the basic types and any of the elemental modifiers but ran

out of time. The idea was that combat with an enhanced earthen Troll would be much different from combat with an ashen Manticore and I wouldn't have to make separate classes for each possible combination. For the scope of this project, that turned out to be too ambitious, so the modifiers or type of beast that you get as your final boss does not affect game play, they just serve to enhance the fantasy feel of the game and provide a unique death for the player to give the game some replayability.

Factory: I never intended to use factory in this project, thus why it is missing some of the features compared to what we did when we implemented it in the homework. I had thought at one point I was going to be generating a random monster for each new chamber the player entered. The factory would have been useful in this case because I could have employed the fact that it decouples creation of objects from their usage. I could have easily generated a new and somewhat uniquely randomized monster (using the decorator) for each room and then transitioned to some sort of combat mode where the player would fight that monster. Even so, the factory might have been the easiest way to create a randomized final boss so I kept it in the project.

Future Developments:

I enjoyed working on this project in the 15-20 hours I was able to put in there were a ton of things that made it into my ambitious initial proposal or conception of the game that I did not have time to implement during the busy finals week For maybe the first time ever in my computer science career I can actually see myself continuing development past the conclusion of

this course. I have no shortage of things I already want to go back and add so I will put them in list format (some with short explanations):

1. More in depth combat system (possibly involving State pattern to switch between combat and exploration mode or Strategy pattern to control monster attack patterns)
2. Monsters in every new room
3. Monster decorators that affect gameplay
4. Progression through looting vanquished monsters
5. Randomized mazes
6. Difficulty settings (probably just increasing the size of the gird from 9x9 to 25x25 if the player wants more of a challenge possibly using facade pattern to do so)
7. Singleton minotaur final boss at the end of the maze
8. Ascension (beating the game gives you better and unique starting loot but a harder challenge)
9. Visual component, ascii or otherwise (I found a website where you can plug photos in and get ascii art for them but couldn't figure out how to print them out in the terminal. The manticore is on the left and the troll is on the right,)