CMPT 276 - Phase 4 Report

Group 22:

Michael Plunkett (mplunket@sfu.ca)
Dina Zeng (dhz1@sfu.ca)
Sameer Hossain (sameerh@sfu.ca)
Salman Rafiei (salman_rafiei@sfu.ca)

Overview of our game

Our game has a main menu, settings, pause, and game over screen, as well as the in-game screen. There are three levels which must be completed in sequence to win.

Story:

Dungeon Manager Escapes is a 2D maze game in a fantasy dungeon setting. You play as a distinguished priest who has been fired from the church due to downsizing. Struggling to pay his inn fees, he answered a job on CraigsRecord for the position of "building" manager at a sketchy company called “LIVE LLC.”. However, on the first day of work, he regrets his decision as the building in question is a dungeon. After going over the terms & conditions he found a loophole which says if he can escape the dungeon the contract is terminated.

However, the boss has yet to give him a skeleton key. He now finds himself having to collect keys spread across the dungeon while avoiding his coworkers(the mobs). The dungeon itself is filled with a thick miasma that constantly depletes the health of those not native to the dungeon and if that wasn’t enough, holy powers don’t work either. Thankfully there's a constant supply of potions, dropped by unprepared adventurers, to recover health.

Difficulties:

- Peaceful: no health depletion, reduced potion spawn rate
- Easy: some health depletion
- Medium: moderate health depletion
- Hard: High health depletion, increased potion spawn rate, decreased potion lifetime

Items:

Key

- +25 points
- Need to collect all keys in order to use door

Potion

- +50 points, +50 health
- Spawn randomly and despawn after fixed time

Door

- +50 points
- Proceed to next level

Spikes:

- -50 points, -50 health

Enemies:

- Bats - Move randomly
- Skeletons - Follow a preset path
- Slimes - Move towards the player

How our plan changed over time

Original plan:

- 3 levels & 1 difficulty
- Health depletes over time, you lose if player runs out of health or score is negative
- 1 enemy that chases the player & 1 punishment(trap) that injures the player
- 1 reward(key) & 1 bonus(health potion)
- Basic tiles(wall, floor, door)

Changes we made:

- Our final project stayed more or less true to our original concept. We structured our code similar to our original UML diagram with a couple of changes to make the development process easier such as adding more classes and renaming
- The UI is different from our original mockup as it was very primitive and we did not have a concrete vision of how our game was going to look.
- Originally there were supposed to be 2 enemy types, 1 to chase the player and the other patrols. We started off with a basic enemy type that just wanders aimlessly as we didn't know how to implement pathfinding hence we ended up with 3 enemy types.
- We implemented 2 entity classes, 1 for moving characters (player & enemies) and 1 for inanimate objects (keys, spikes, doors). It made more sense to have the punishment(spikes) be an item that the player interacts with than being part of the enemy class.
- The door also extends the item class rather than being part of the tiles/map components as it is an interactable object rather than a map tile.

What we learned

Git:

- Managing a git repository, creating README, using .gitignore
- Basic skills & becoming proficient with committing, pushing, pulling, and using branches
- Resolving merge conflicts
- Creating meaningful branches for specific tasks
- Understanding git's history graph was incredibly useful

Java

- Design patterns and high level object oriented design
- Importance of encapsulation and singleton classes which made code cleaner and easier to read, had we done so in the beginning our classes would be less coupled

- Maven (build automation) made it easier to build the project in different environments without worrying about the dependencies, processes, etc. since we were using 2 different IDEs(IntelliJ & VS Code)
- Javadocs (documentation) which makes it easier to understand the implementation of functions
- JUnit (Unit and Integration tests)

Teamwork/communication: these things are important…

- Managing communications via discord with different channels for different things
- Holding bi-weekly online SCRUM meetings
- Holding in-person sprint sessions
- Sharing documents and staying up to date with each others work
- Delegating workload

Include link to our video:

https://www.youtube.com/watch?v=FEo5Omps2wY

References:

Game structure:

1. Ryisnow. (n.d.). How to Make a 2D Game in Java. YouTube. Retrieved April 9, 2023, from https://www.youtube.com/playlist?list=PL_QPQmz5C6WUF-pOQDsbsKbaBZqXj4qSq

Game assets:

2. Pixel_Poem. (n.d.). *PIXEL_POEM DUNGEON TILESET*. Retrieved April 9, 2023, from https://pixel-poem.itch.io/dungeon-assetpuck
3. 患者長ひっく(patient head scratch). (n.d.). マルモニカ【無料・商用利用可】- *x0y0pxfreefont - booth*. x0y0pxFreeFont. Retrieved April 9, 2023, from https://00ff.booth.pm/items/2958237