Project 7

Project Name
2D Dungeon Game - Warped Souls

Project Repo
Github Repo Link

Team Members
Scott McCall, Sitong Lu, Adithiyya Kishoore

Final State of System Summary

We implemented six design patterns:

- Singleton Pattern: Creating the Board
- Strategy Pattern: To change attack and movement behaviors of enemies
- Command Pattern: For Button Functionality
- Observer Pattern: To document the changes in stats
- Factory Pattern: To create enemies.
- Iterator Pattern: To scroll through the list of enemies.

The player spawns on the board and can move around with the cost of action points. It is a turn based game and the player and the enemies get a set number of action points each turn which get refreshed at each turn.

Each grid has a different altitude and a hidden altitude. The player can take damage by falling from a higher altitude to lower altitude.

We currently have enemies that exhibit homing movement and random movement when blocked from getting to the player. Enemies also change from weak attacks to strong attacks when their HP is low.

The player has an inventory for storing items. The player can buy items using gold from the shop which spawns randomly on the grid. He is also able to dig for potions by spending action points.

The player can attack the enemies by clicking on the map. The player wins by moving to the treasure at the far end of the board.

SCRAPPED IDEAS

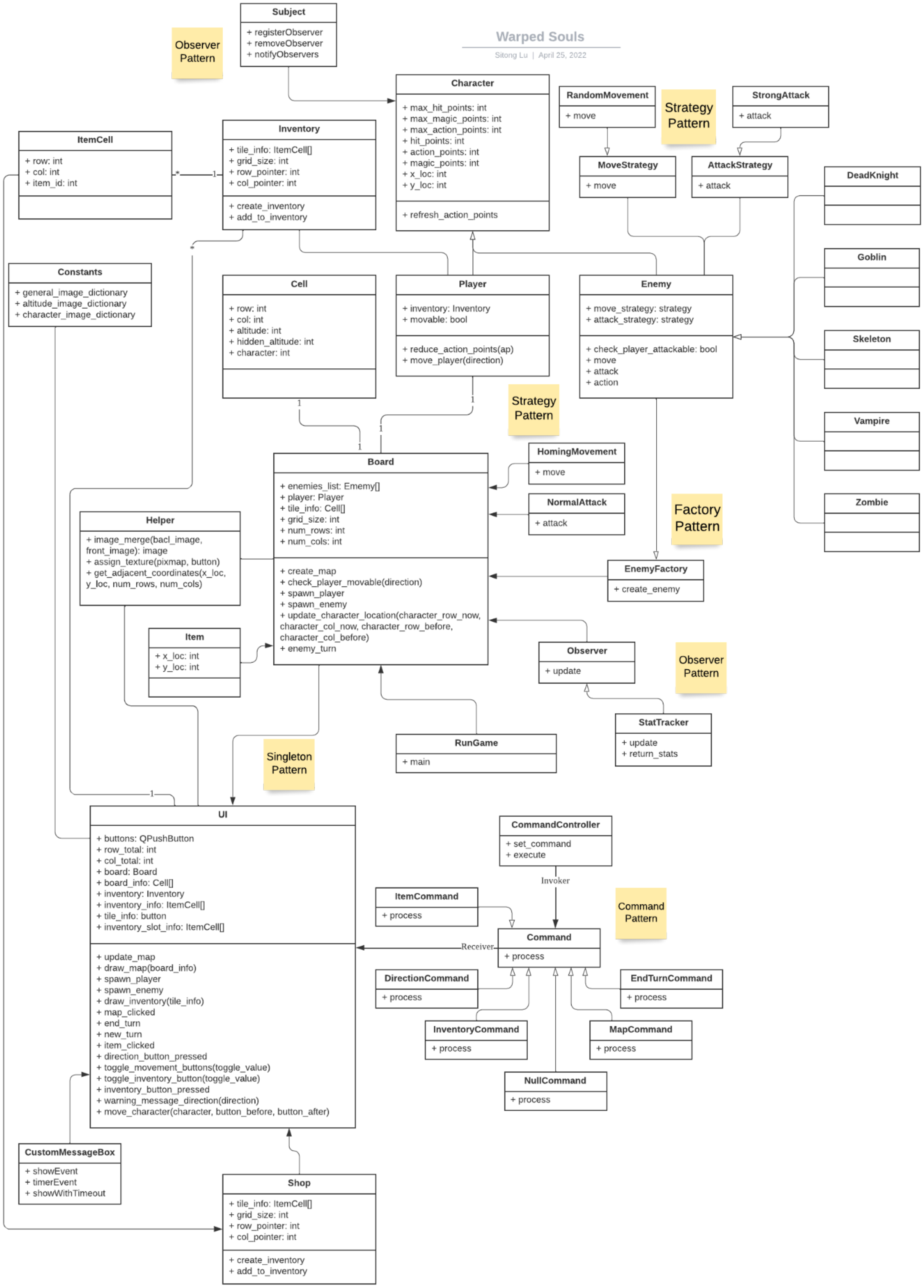
Certain enemies would have a random chance to drop items when killed. However, due to time constraints the enemy dropping items idea was scrapped.

Upon collecting the treasure, the Dead Knight (Boss Enemy) would spawn and the player would go back to the start to win the game.

There was to be a save and load functionality. But the game was quite quick, so this didn't make much sense.

There was also supposed to be ranged attacks, but unfortunately that was also scrapped due to the timelines.

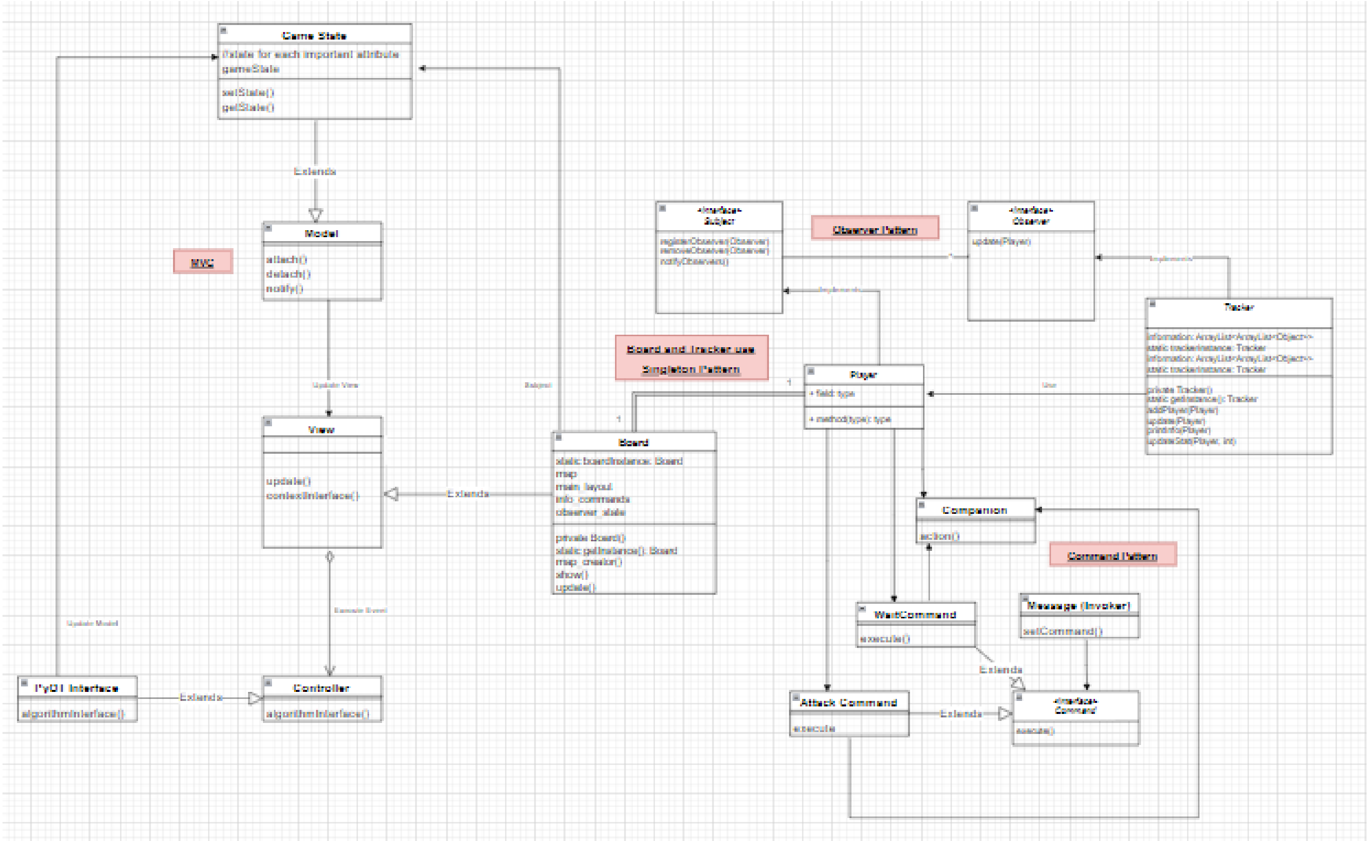
Class Diagram
For Project 7:
Link to the hi-res version:
https://drive.google.com/file/d/1fm-q9nZ_S-EHMoL9idthpnFooZxDfxCP/view?usp=sharing



Warped Souls

Sitong Lu | April 25, 2022

For Project 5:
Link to the hi-res version:
https://drive.google.com/file/d/1JB8cCX5ORB-8ZbhoARe4PQv4MGMlw1rI/view?usp=sharing



Comparison Statement

- Support the diagrams with a written paragraph identifying key changes in your system since your design/work was submitted in Projects 5 and 6

The key change in our system since the work we submitted in Project 5 and 6 was that we moved all the methods that were set up for handling button-pressed events from the UI class to a new Command class, mainly for decoupling the usability of the original UI class to make sure that it only handles UI and initialization related actions, including setting up all the buttons and the board to play with in the window, all the message boxes that the program is going to use during gameplay, and the method for spawning all the different characters and objects at the start of a game. The rest of the changes were primarily adding more design patterns into the system, like the Observer pattern and the Factory pattern.

Third Party Code vs. Original Code Statement

The third party code we used was to make the backend sleep while the front end still displayed.

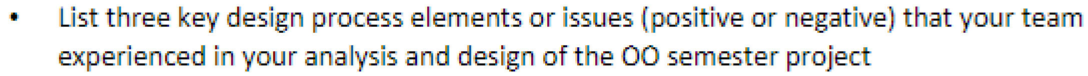
Reference:
https://stackoverflow.com/questions/17960159/qwait-analogue-in-pyside

Plus the one used in CustomizedMessageBox.py to make a customized message box that can automatically close after a set amount of time.

Reference:
https://gis.stackexchange.com/questions/137593/how-to-programmatically-close-a-qmessagebox-without-clicking-ok-or-x

Everything else is original code.

OOAD Process Statement



- List three key design process elements or issues (positive or negative) that your team experienced in your analysis and design of the OO semester project

- One of the process issues we encountered during our development of the semester project was one of Steve McConnell's Classic Software Mistakes: an overly aggressive schedule. During our analysis and design phase in Project 5, we initially had ambitions for our project that ended up not coming to fruition. This included enemies dropping items as well as the ability to specifically target enemies from far away. Unfortunately, we were not able to accomplish everything we intended within the short 4 week timeframe.
- We also struggled to find homes for responsibilities during our design phase. Specifically, we originally had a lot of logic in our button_clicked() method in our UI file before realizing that it would be better to implement a Command pattern that takes care of this logic.
- One of the successful techniques we used was commonality analysis to determine supertypes which would make our code structure neater. For example, we decided that both the player and enemies should have a common superclass since they both have action points, health points,

etc. so this lead to us determining the need for a character class during the design phase.