Cassidy Carpenter
John Stibbards
Thomas Young

Project 5 Status Summary

- **Work Done:** Written description of the work done in the first two weeks of your project and (in the case of multi-person teams) the breakdown of work across team members.
    - Cassidy
        - I have been working on creating the class files for the 'Rooms'. This includes the Enemy, Trap/Puzzle, Item, and Default rooms. Each of the rooms have their own idiosyncrasies and the puzzle room has an additional puzzle question class that deals with creating the different types of questions/puzzles to ask the player. The puzzle questions will be another thing within itself in complexity, but as of now the general layout of how we want the questions to be asked is set up. As for the other rooms, the general functionality of each room type is also completed. That being said, the rooms need to be created and set up into a map as we continue development.
    - John
        - I have begun structuring and laying out the code for the factories we will be using in our project. There are a total of 5 factories, floor, room, item, enemy, and puzzle factories that will manage the creation of their respective objects in the game. A problem that has required some attention is what complexity of factory these should be, but for now I have built the factories on the understanding that they can perform their job in the game properly as only simple factories, which has streamlined implementation.
    - Thomas
        - I have been working on creating the entity, item, and skill classes. I had to restructure how they worked because our original plan to make items decorators for entities made the program unnecessarily complicated. I changed it so entities now have a list of items equipped to them. Basic battle mechanics work between entities, however, enemies do not intelligently select moves, and we need the graphics side of the game to give input.
- **Work Estimate:** Provide an estimate of how much more work needs to be done for you to have implemented the design that you presented in Homework 4 (or describe how your design is changing as a result of making progress in implementing your system).
    - We have done a good amount of work, however we still have a handful of things to implement. Our current plan still seems feasible by the deadline. The things we need to do are:

        - Create graphics side of the game
            - One group member has a large asset library which means we don't have to make things like sprites
            - I believe this will be the largest hurdle we have to face since this will be handling both display and input for the game.
        - Create the Game object to setup and start the game
        - Connect the parts each of us has created so that they are able to communicate. An example of this is starting a battle when the player goes into an enemy room.
- **Screenshots:** This is an example of a text-based visualization of battles

```
Thomas || Health: 100 || Mana: 100 || Attack: 20 || Defense: 40 ||  Equipped items: BiggerHat, Sword
Player used: absorb head
Big Bad Boy || Health: 97 || Mana: 90 || Attack: 30 || Defense: 60 ||  Equipped items: Sword, HAT!!!, ring
Enemy used: regain mana
Thomas || Health: 100 || Mana: 100 || Attack: 20 || Defense: 40 ||  Equipped items: BiggerHat, Sword
Player used: absorb head
Big Bad Boy || Health: 94 || Mana: 90 || Attack: 30 || Defense: 60 ||  Equipped items: Sword, HAT!!!, ring
Enemy used: Sit on head
Thomas || Health: 100 || Mana: 100 || Attack: 20 || Defense: 40 ||  Equipped items: BiggerHat, Sword
Player used: absorb head
Big Bad Boy || Health: 91 || Mana: 70 || Attack: 30 || Defense: 60 ||  Equipped items: Sword, HAT!!!, ring
Enemy used: Sit on head
Thomas || Health: 100 || Mana: 100 || Attack: 20 || Defense: 40 ||  Equipped items: BiggerHat, Sword
Player used: absorb head
Big Bad Boy || Health: 88 || Mana: 50 || Attack: 30 || Defense: 60 ||  Equipped items: Sword, HAT!!!, ring
Enemy used: Sit on head
Thomas || Health: 100 || Mana: 100 || Attack: 20 || Defense: 40 ||  Equipped items: BiggerHat, Sword
Player used: absorb head
Big Bad Boy || Health: 85 || Mana: 30 || Attack: 30 || Defense: 60 ||  Equipped items: Sword, HAT!!!, ring
Enemy used: regain mana
```

- **Test Cases:** Our test cases for this project are easy as there is only one real test case, that being a player who wants to play the game. While each individual aspect of the project will still need to be tested, the program as a whole only has the test case of playing the game.
- **Patterns:** In addition, now that you have more of your system implemented please describe if you have been able to incorporate the use of design patterns into your prototype. If so, describe the patterns you have incorporated into your design and how they are helping you or your design.
    - Since our last report we have kept and changed a few of the patterns we are using.
        - **Factories:** We are continuing to use simple factories to create the main pieces of the game: The main floor, rooms, items, enemies, and puzzles. This has been useful because it simplifies creating each floor of the game into a clean flow of FloorFactory creating the floor, which calls the RoomFactory to create each room, and final the RoomFactory calling the enemy, item, and puzzle factories to make the resources each room needs. All of this is done just by asking the FloorFactory to produce a floor, abstracting the entire process.
        - **Decorator:** In our original report we said that we were going to use the decorator pattern to equip items to entities. This has come out to be over

complicating the feature and so we have simplified this to entities being an aggregate of items. This way we are able to equip/handle items easily while also allowing for the expansion of item types in the future.

- **MVC:** We are still planning on using the mvc pattern to handle the graphics, input, and state of the game. The skeleton code for this has been created, but we have yet to do the full implementation which means this could change.
- **Observer:** This will also be used to update the visuals of the game whenever there is a change in the game state. Again, we have not implemented this but this should work well so we don't plan to change it.
- **Strategy:** We are continuing to use the strategy pattern to implement skills and this has worked well so far. This will let us implement more skills in the future with as little modification as possible.

Plan For Next Iteration

We still need to finish a few steps of developing the game before the next deadline. We have the skeleton of the program created, we just have to add the full functionality of things not yet implemented, such as the graphics. One of the major factors is the creation and running of the game instance so that it can actually be played where all the parts are properly connected. This will include implementing the GUI (with mvp pattern) and getting player interaction. Once these steps are completed the functionality of the project is largely in place, and the main hurdle at this point is to populate our game with some objects that vary (items, enemies and puzzles) to make the gameplay interesting.