Project 5 - Design for Zelda like mobile game.

CSCI-5448 Su22
Curtis Covington
Tim Coleman

Project Summary

• *What is the Project called?*

Zelda-like

• *Who is on your team (include all names)?*

Curtis Covington
Tim Coleman

• *What is the high-level overview of your semester project? What are you trying to accomplish?*

The goal is to build a simple, yet cross-platform game using Google's Flutter framework. We hope to be able to play the game we make on our phones.

What will your system do when you are done?

It will be a simple game with RPG elements. We hope to have a character be able to grow in power and traverse a map of nodes where players can make meaningful choices to help them win the game.

Project Requirements

- Our core constraint is that the game be playable on mobile devices, specifically Android and iOS devices.
- The map system should allow the user to select a node connected to their current node. They can not move to a previous node. Once a node is selected, the associated node action should be performed.
    - Trigger a random event
    - Go to the shop with random items for sale
    - Trigger a random battle
    - Trigger a boss battle]

- Player progress is tracked through nodes. When a player reaches 0 health, the game is over. When the player completes the boss node, the game is over and the player wins.

- Items should increase core stats of the players, such as strength or defense or health. Purchasing an item will increase an associated stat.

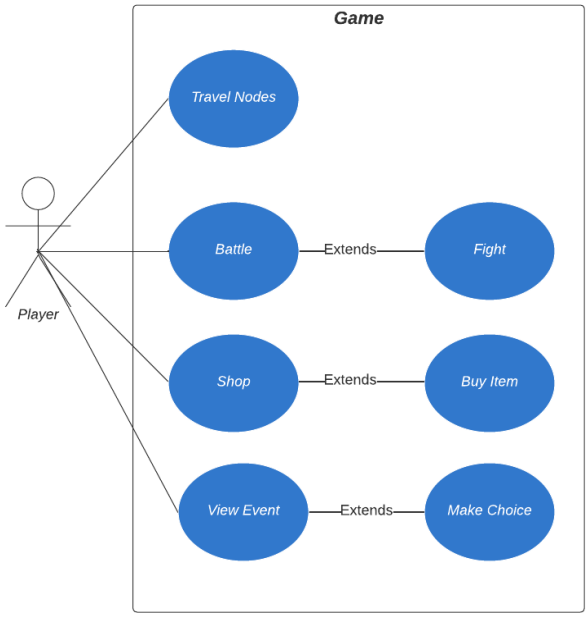
- Battles are turn based. The player will choose an action such as fight or defend. The enemy will have a random chance of choosing between 3 actions, strong attack, normal attack, or defend. The battle is over when all enemies are defeated or the player has 0 hp. The player receives a random amount of gold for winning.

- Random events can provide the player some flavor text along with a choice. Their choice should determine the outcome of the event. Be it positive or negative. A positive result could be receiving an item or gold. A negative result could be losing health.

Users and Tasks: Use Cases (Text or UML Diagrams)

Below represents a UML use case diagram showing the different ways a user would interact with the game. There is only one player allowed per game so only one user will interact with a game at a time.

Use case diagram

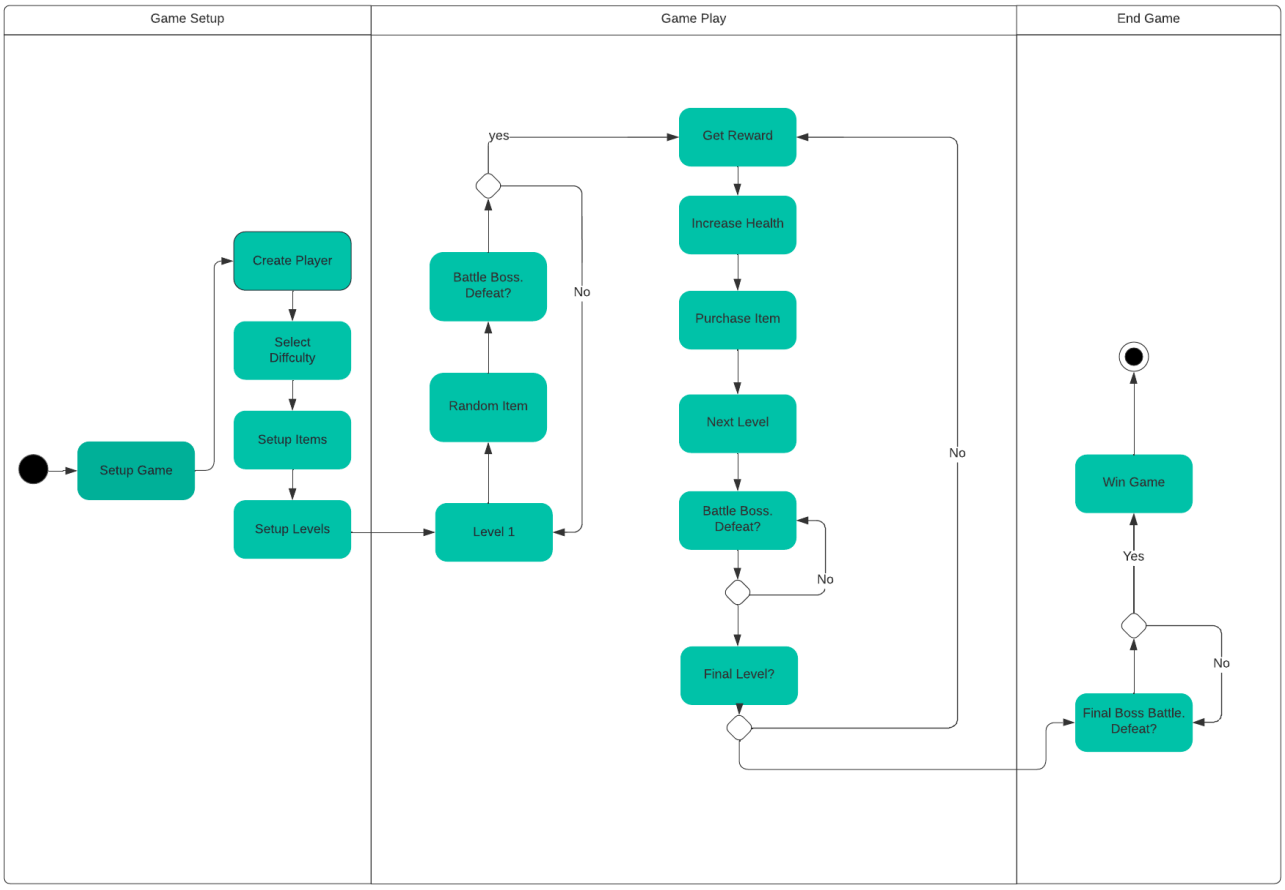


Citation: UML Reference
https://www.lucidchart.com/pages/uml-use-case-diagram

UML Activity Diagram

Below represents the Activity diagram which has 3 swimlanes to represent Game Start, Play, and End, and shows the flow of events through the system of how a user would play that game from start to completion.

Project 5 (Zelda) UML Activity Diagram

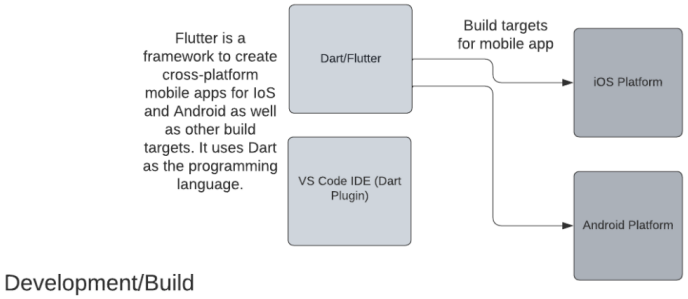


Citation: UML Activity Templates: https://www.lucidchart.com/pages/uml-activity-diagram

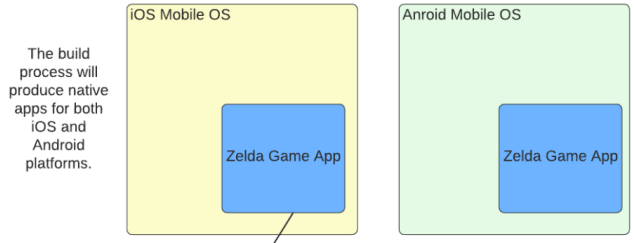
Architecture Diagram

Below represents the high level architecture from a development, runtime and component level viewpoint. The app should be fairly self-contained with minimal to no runtime external resources (e.g. rest api requests), but the plan is to use a lightweight database (Firebase) to store game contents and logging information. We may end up using a simple json file or flat file for logging though. The development environment we plan to use is either intellij or VSCode and both have support for the Flutter/Dart environment.

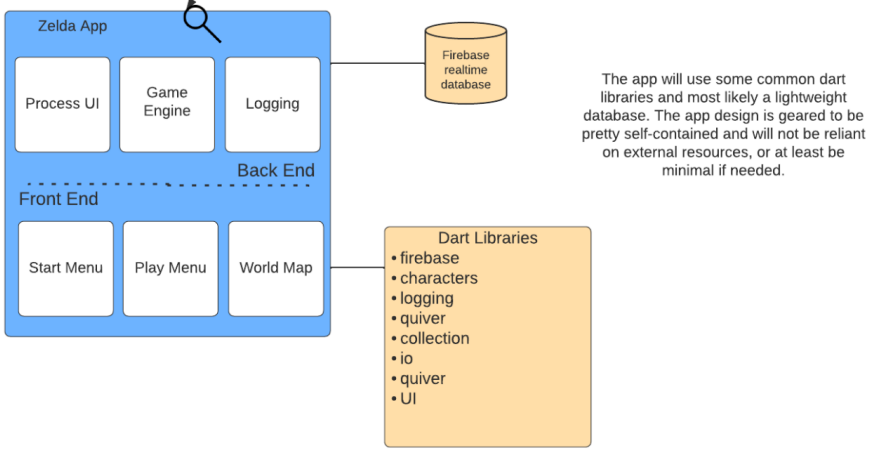
Project 5 (Zelda) UML Architecture Diagram



Development/Build



App Run Time



App Components and Libraries

Data Storage

• Discuss how you will persist data in your application. What storage technology will you use? Text files? XML? CSV? MySQL? Hibernate? Where will the data be stored? Describe the classes and libraries that you will use to access this data at run-time.

As illustrated in the architecture diagram above, the plan is to use a lightweight database that works well for mobile apps. One database that we are considering/planning is Firebase. Flutter/Dart has established libraries to connect to the database (https://docs.flutter.dev/development/data-and-backend/firebase) and there are lots of resources and documentation for the general setup to learn and build from.

The database could also be used to save the game state so a user could save the state and then resume at a later time. Firebase is a NoSQL DB so it will be a json type structure but we could store the document something like as basic as this:

```
{
  "player": "name",
  "items_aquired": [1,2,3],
  "health": 4,
  "levels_passed": [1,2]
  "health_capcity": 6
  "time": epochtime

}
```

We could also store other objects (items for instance) but it's unclear if that's really necessary as these can be stored in the objects which get instantiated at runtime. But we may think about storing them, for example:

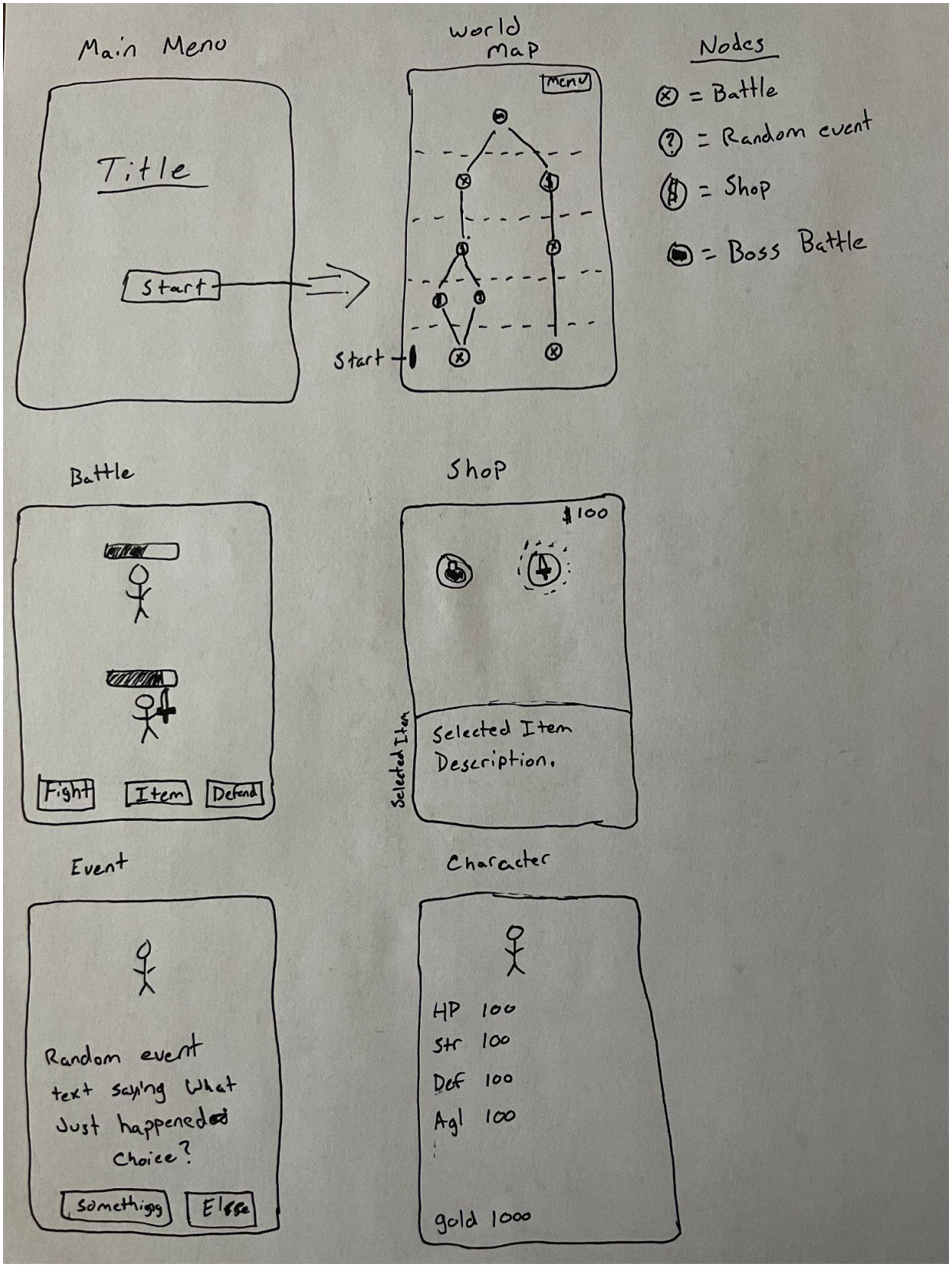
```
{
   "item": "bomb"
   "health_boost": 10
}
{
   "item": "boomerang"
   "health_boost": 20
}
```

```
{
   "item": "sword"
   "health_boost": 30
}
```

UI Mockups/Sketches

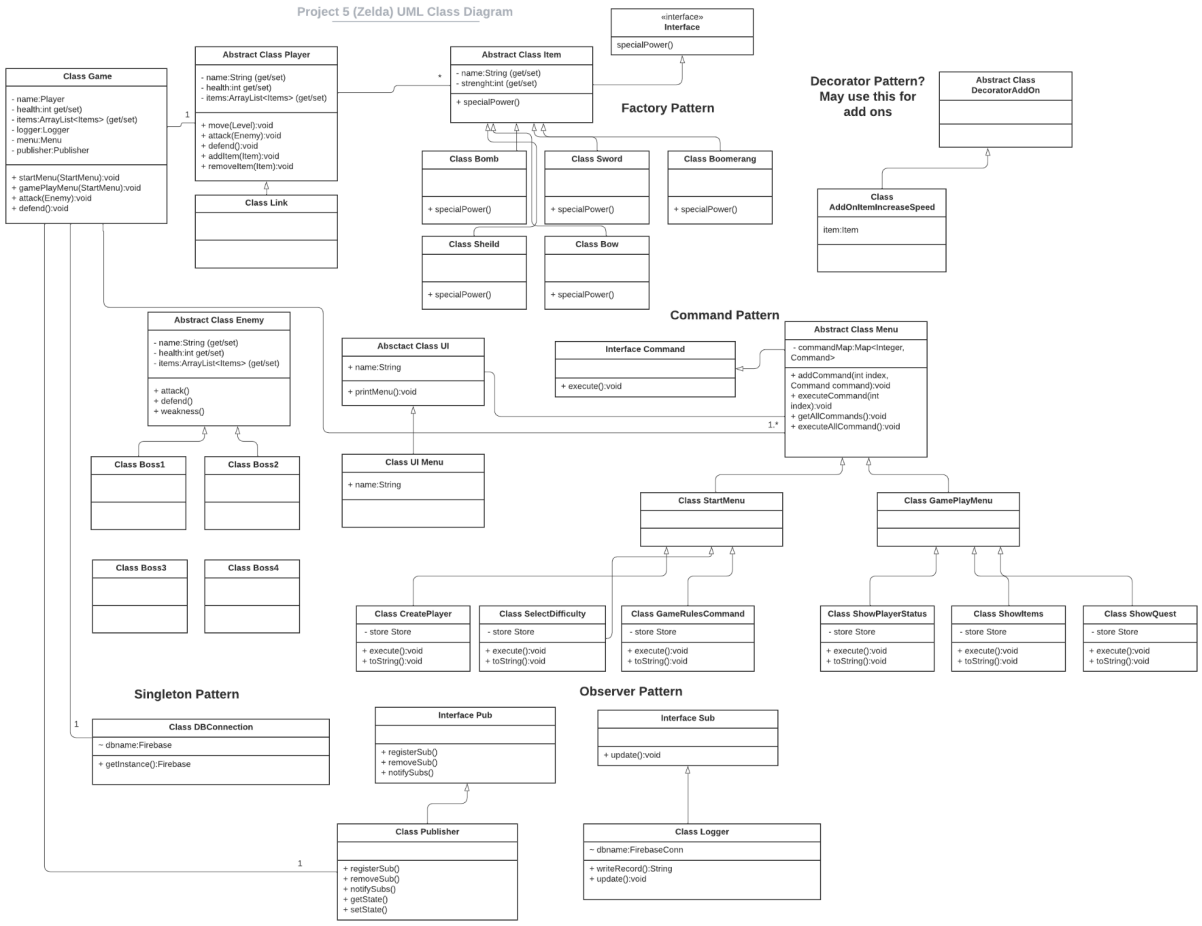
Create screen mockups for the user interface of various parts of your application. What will a user see as they work through the tasks identified in your use cases? What is the overall organization of your user interface? How will data be displayed? How will the user navigate from screen to screen?

The hand drawn sketch below shows an example of the UI screens we plan to implement and how the user will interact with the game.


Main Menu
Title
Start
world Map
Menu
Start
Nodes
= Battle
= Random event
= Shop
= Boss Battle
Battle
Fight
Item
Defend
Shop
$100
Selected Item
Selected Item Description.
Event
Random event text saying what just happened Choice?
Something
Else
Character
HP 100
Str 100
Def 100
Agl 100
gold 1000

UML Class Diagram & Pattern Use

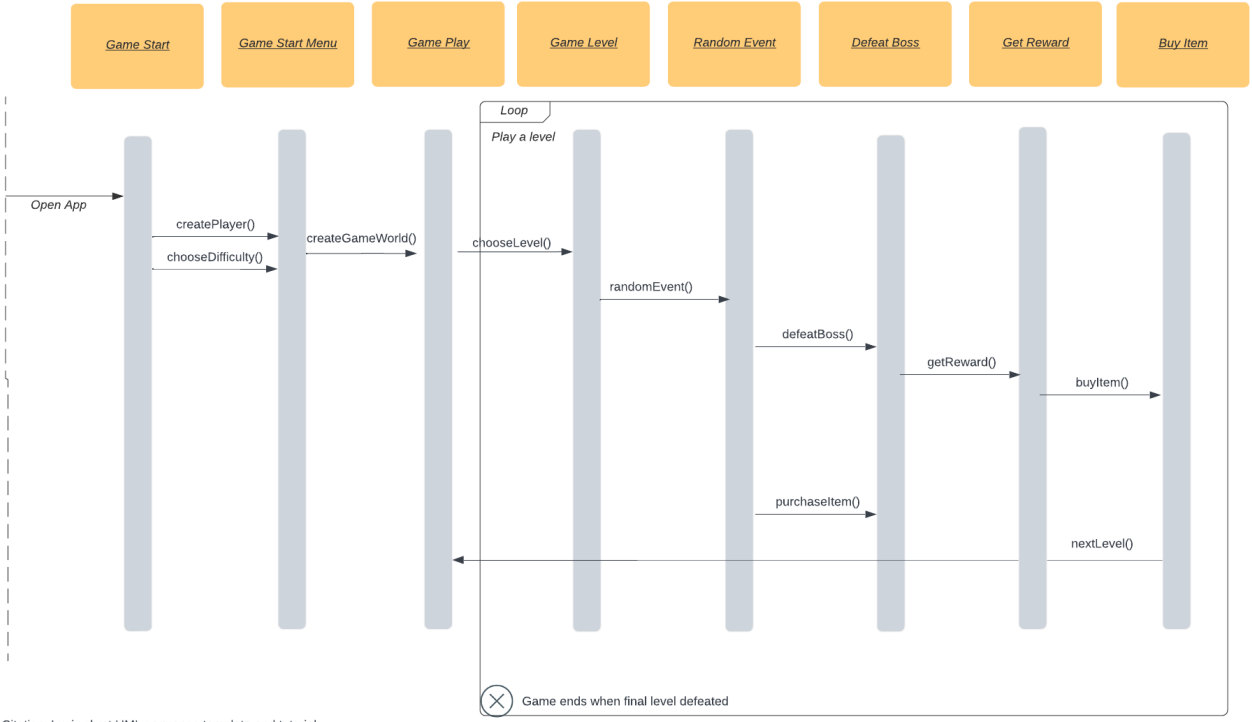
Below is the draft class diagram. As we work through the coding aspect we expect more classes, members and methods to be added, but we are fairly confident that we'd like to support 4 of the design patterns shown below (Factory, Singleton, Command and Observer), and we also may use Decorator to handle any add-ons for the game items. We noted the use of that in a question mark.



User Interactions/UML Sequence Diagrams

Below represents the sequence of the game from start to end and shows the loop when a user is actively playing the game from level-to-level until the user beats all the levels, which ends the loop and finishes the game.

Project 5 (Zelda) UML Sequence Diagram



Citation: Lucic chart UML sequence template and tutorial:
https://www.lucidchart.com/pages/uml-sequence-diagram