**Use Cases - Phase 2**

Group: Group 9

### **Use Case Name:** Choose Wizard Monster’s Behavior

**Use Case ID**: UC-11  
**Scope**: ROKUE LIKE - Phase 2  
**Level**: User-Goal  
**Primary Actor**: Wizard Monster

#### **Stakeholders and Interests:**

* **Player**: Expects the wizard monster to add unpredictability and challenge based on the game’s current state.
* **Developer**: Wants to implement time-based, dynamic behavior using the Strategy pattern for seamless transitions.

#### **Preconditions:**

* The game must be in **running/play mode**.
* The wizard monster must have spawned (triggered by game logic).
* The total game time or remaining time is tracked and updated regularly.

#### **Postconditions:**

* The wizard monster performs actions dynamically based on the percentage of time remaining and disappears after completing its action.

#### **Main Success Scenario:**

1. The wizard monster spawns in the hall.
2. The system calculates the percentage of time remaining in the game.
3. Based on the remaining time, the wizard monster chooses one of the following behaviors:
   * **<30% time remaining**:
     + Teleports the hero to a random empty location once.
     + Disappears immediately afterward.
   * **>70% time remaining**:
     + Changes the rune’s location every 3 seconds until the time threshold drops below 70%.
   * **30%-70% time remaining**:
     + Remains idle in its current location.
     + Disappears after 2 seconds without taking any action.
4. If the time threshold changes while the wizard monster is present, it dynamically switches to the appropriate behavior.
5. The wizard monster disappears once its action is completed or its behavior duration ends.

#### **Extensions:**

* **a. System fails at any time**:
  + The player restarts the game; wizard monster’s state is reset.
* **2a. Behavior conflict at threshold boundaries**:
  + The system resolves the conflict by applying the most appropriate behavior and logs the transition.

#### **Frequency of Occurrence:**

* The wizard monster appears randomly or at preset intervals, triggering the described behavior.

**Use Case Name:** Save/Load Game

**Use Case ID**: UC-12  
**Scope**: ROKUE LIKE - Phase 2  
**Level**: User-Goal  
**Primary Actor**: Player

#### **Stakeholders and Interests:**

* **Player**: Wants to save the game state and load it later to continue from the exact point of progress.
* **Developer**: Aims to implement the feature using file-based storage (e.g., serialization) for saving and loading game states.

#### **Preconditions:**

* The game is in **running/play mode** (for saving).
* The main menu has a **Load** button for accessing previously saved files.

#### **Postconditions:**

* The game state is saved in a file or loaded from a selected file without losing progress.

#### **Main Success Scenario (Saving):**

1. The player clicks the **Save** button during gameplay.
2. The system prompts for a file name or generates one automatically (e.g., timestamp-based).
3. The system collects all relevant game data, including:
   * Hero’s position, time remaining, lives remaining.
   * Inventory contents (enchantment types and quantities).
   * Monster states (type, position, status).
   * Locations and statuses of objects (e.g., chests, runes).
4. The game state is serialized and stored in the chosen file format (e.g., JSON, XML, or Java serialization).
5. A confirmation message is displayed (e.g., “Game saved successfully”).

#### **Main Success Scenario (Loading):**

1. The player clicks the **Load** button on the main menu.
2. A list of saved files is displayed.
3. The player selects a file.
4. The system deserializes the selected file and restores the game state, including all hero, monster, and object positions.
5. The game resumes from the saved state.

#### **Extensions:**

* **a. System fails at any time**:
  + The player restarts the game.
* **2a. Save file is missing or corrupted**:
  + The system displays an error message (e.g., “Save file not found or corrupted”).
* **2b. Save permission denied**:
  + The system alerts the player to retry saving with appropriate permissions.

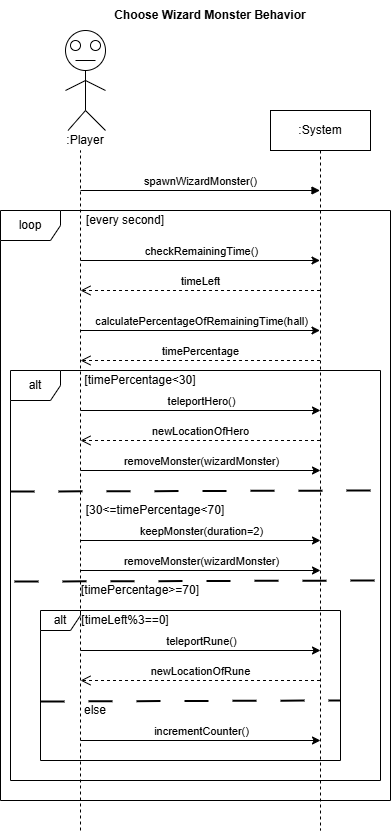
#### **Technology and Data Variations List:**

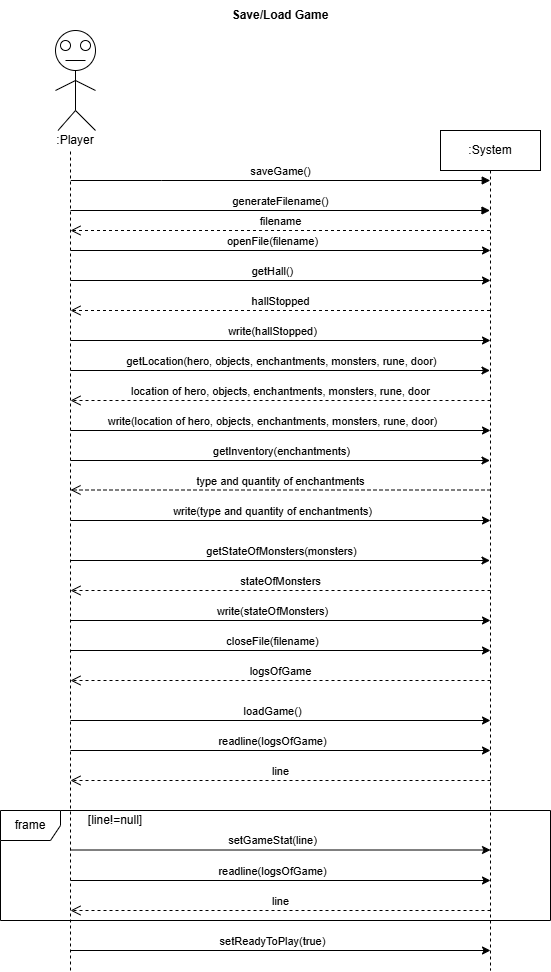
* Alternative file formats (e.g., JSON via Jackson) may be used for better readability.
* Multiple save files can be stored in a directory accessible by the game.

#### **Frequency of Occurrence:**

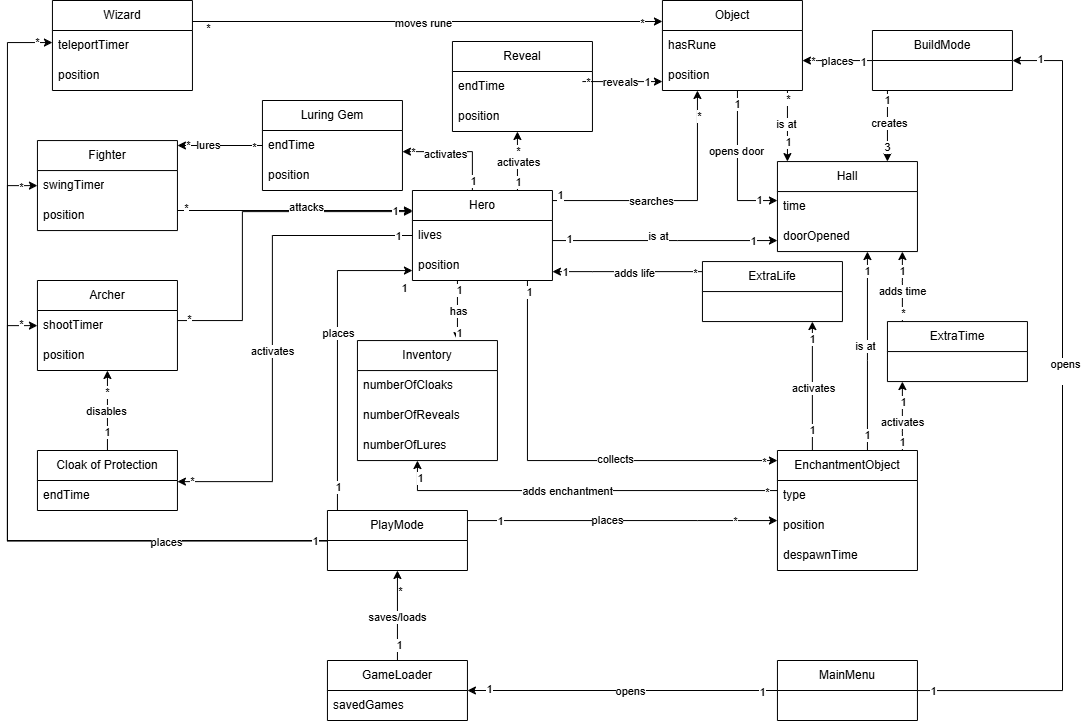
* The player can save the game at any time during gameplay.
* The player can load a saved game from the main menu before starting a new session.

**System Sequence Diagram - Phase 2**





**Domain Model- Phase 2**



**Operation Contracts - Phase 2**

**Operation:** SpawnWizard()

**Preferences: Use Cases:** Choose Wizard Monster’s Behavior

**Pre-conditions: -**The game must be in running mode

-The wizard monster must have spawned

-Total time and remaining time is tracked and updated regularly

**Post-conditions:** 1a: If less than 30% of the total time remains, the monster concludes the hero is close to losing, moves the player to a random empty location once, and disappears.

1b: If more than 70% of the total time remains, the monster concludes the hero is doing well, making the game challenging by changing the location of the rune every 3 seconds.

1c: If the remaining time is between 30%-70%, the monster is indecisive, stays in place for 2 seconds, then disappears without doing anything.

**Operation:** SaveGame()

**Preferences: Use Cases:**  Save Game

**Pre-conditions:**  -The game is already in playing mode

-Playing mode has save button

**Post-conditions:** The game state is saved in a file or loaded from a selected file without losing progress.

**Operation:** LoadGame()

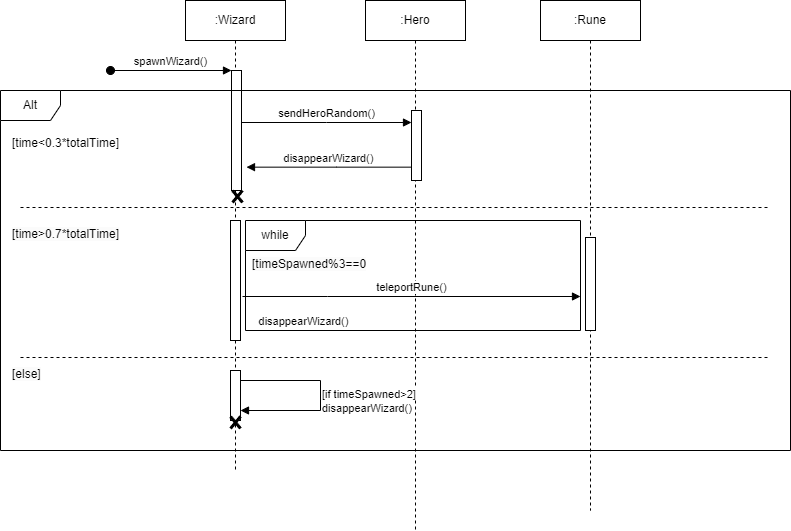
**Preferences: Use Case**s: Load Game

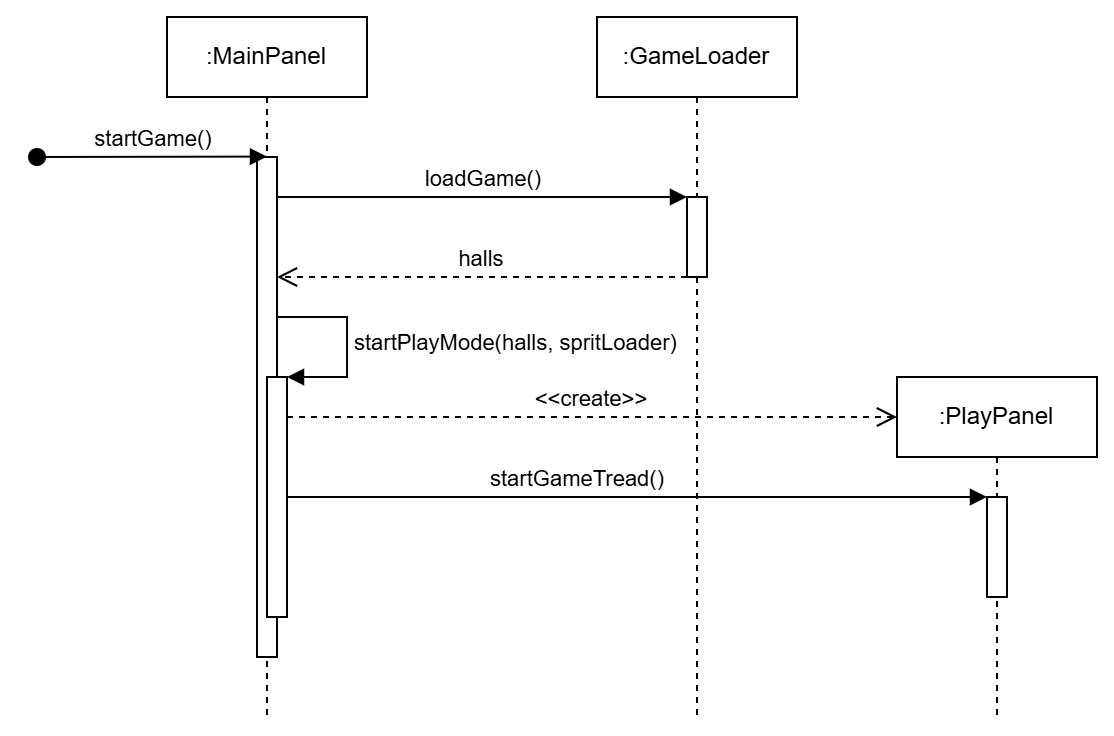
**Pre-conditions:**  -The game is already in playing mode

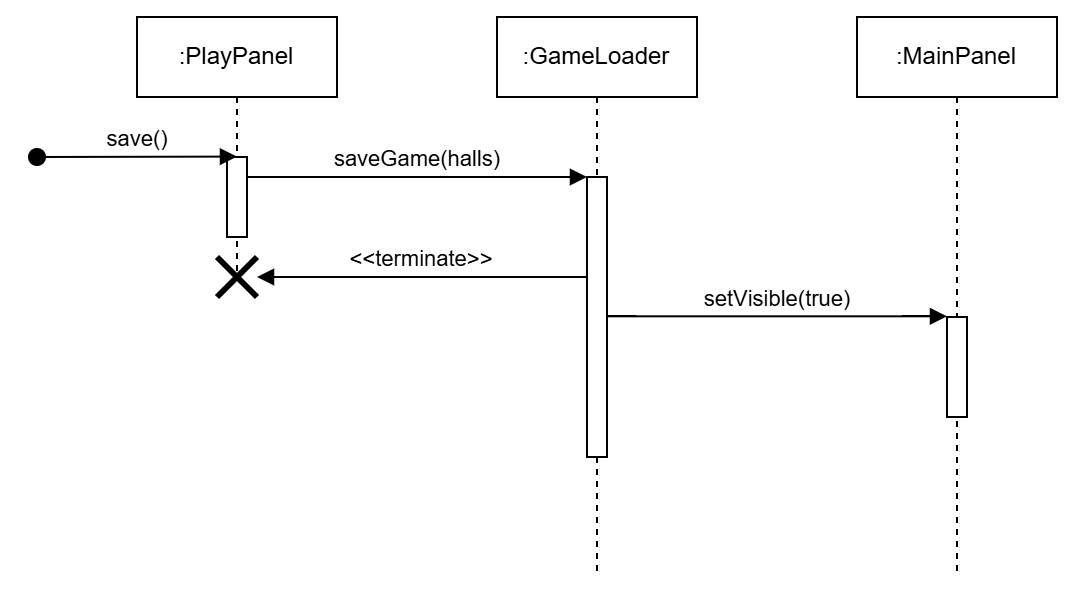
-The main menu has load button

**Post-conditions:** Previously saved game will load correctly

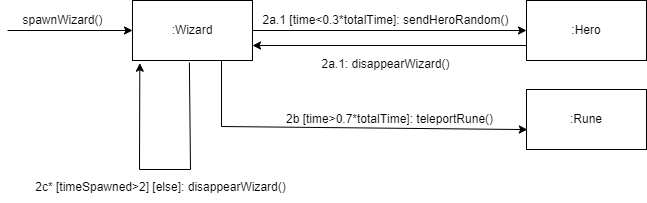
**Sequence Diagram - Phase 2**

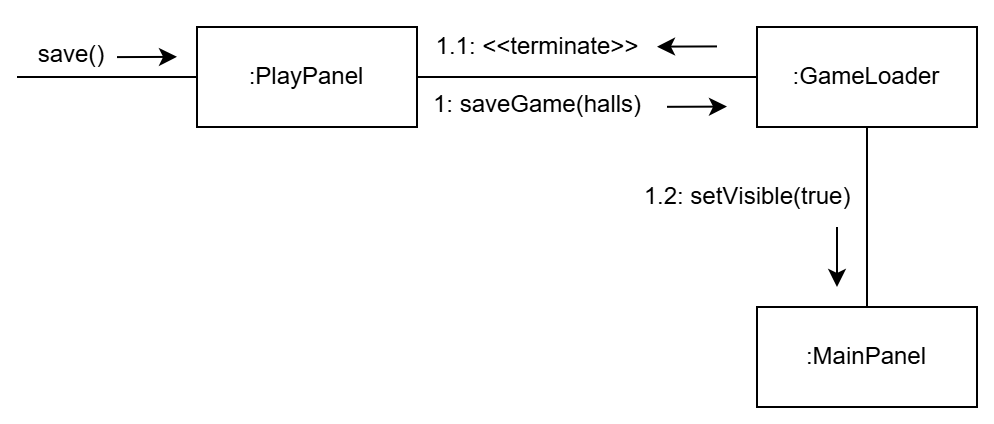
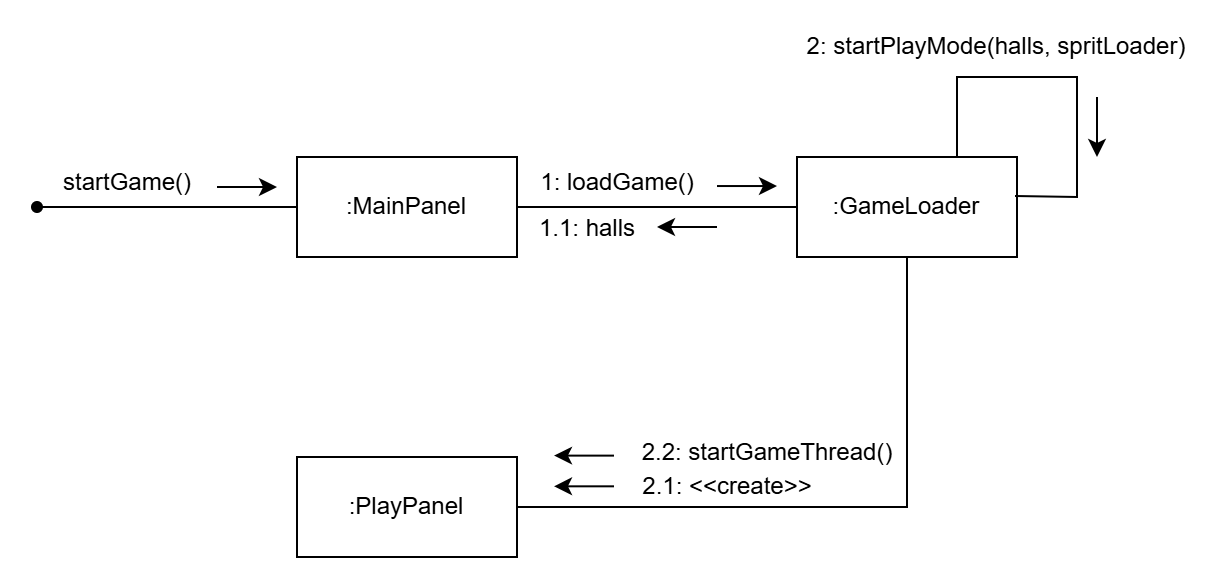
****

****

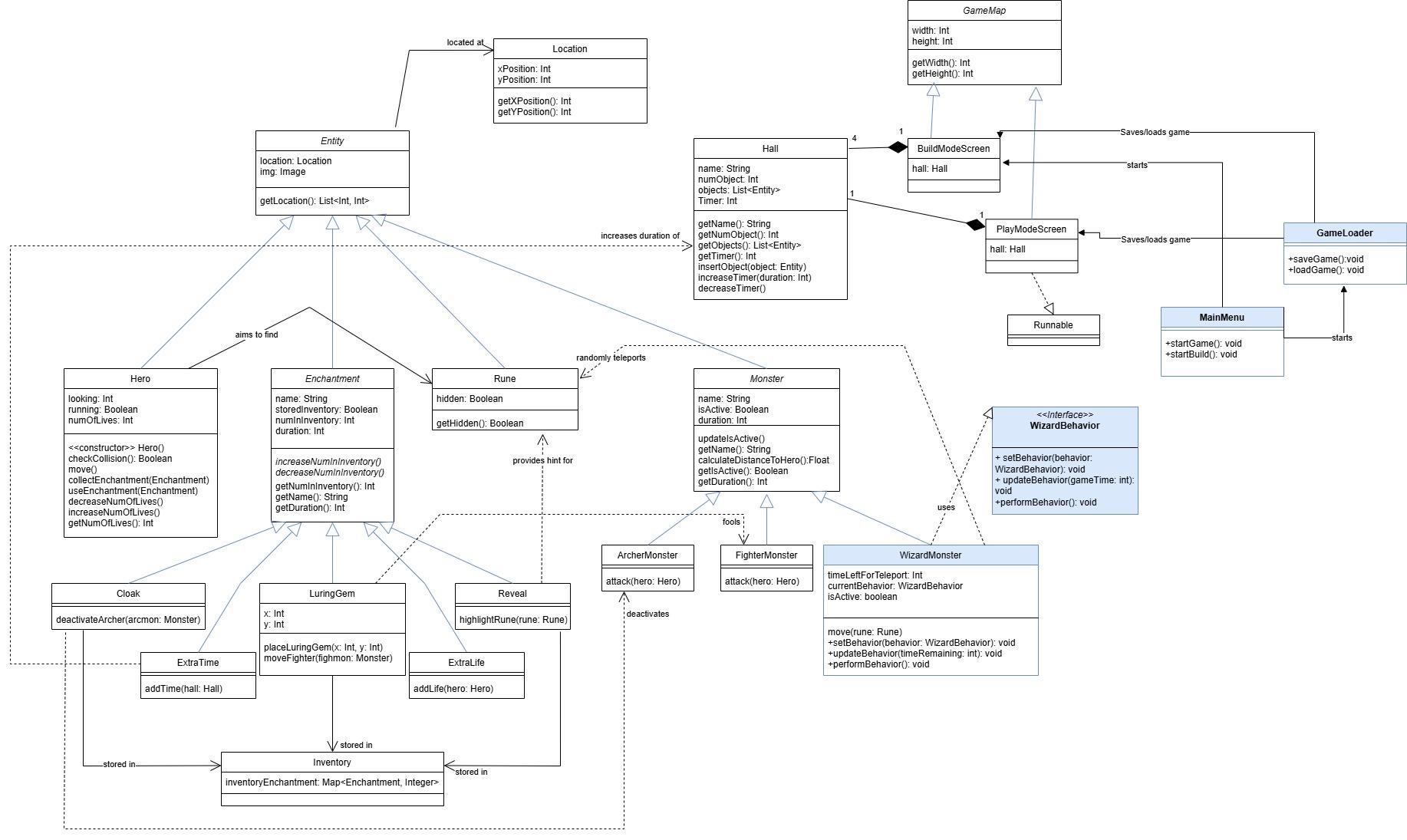
****

**Communication Diagram - Phase 2**

****

****

**Class Design Diagrams - Phase 2**

****