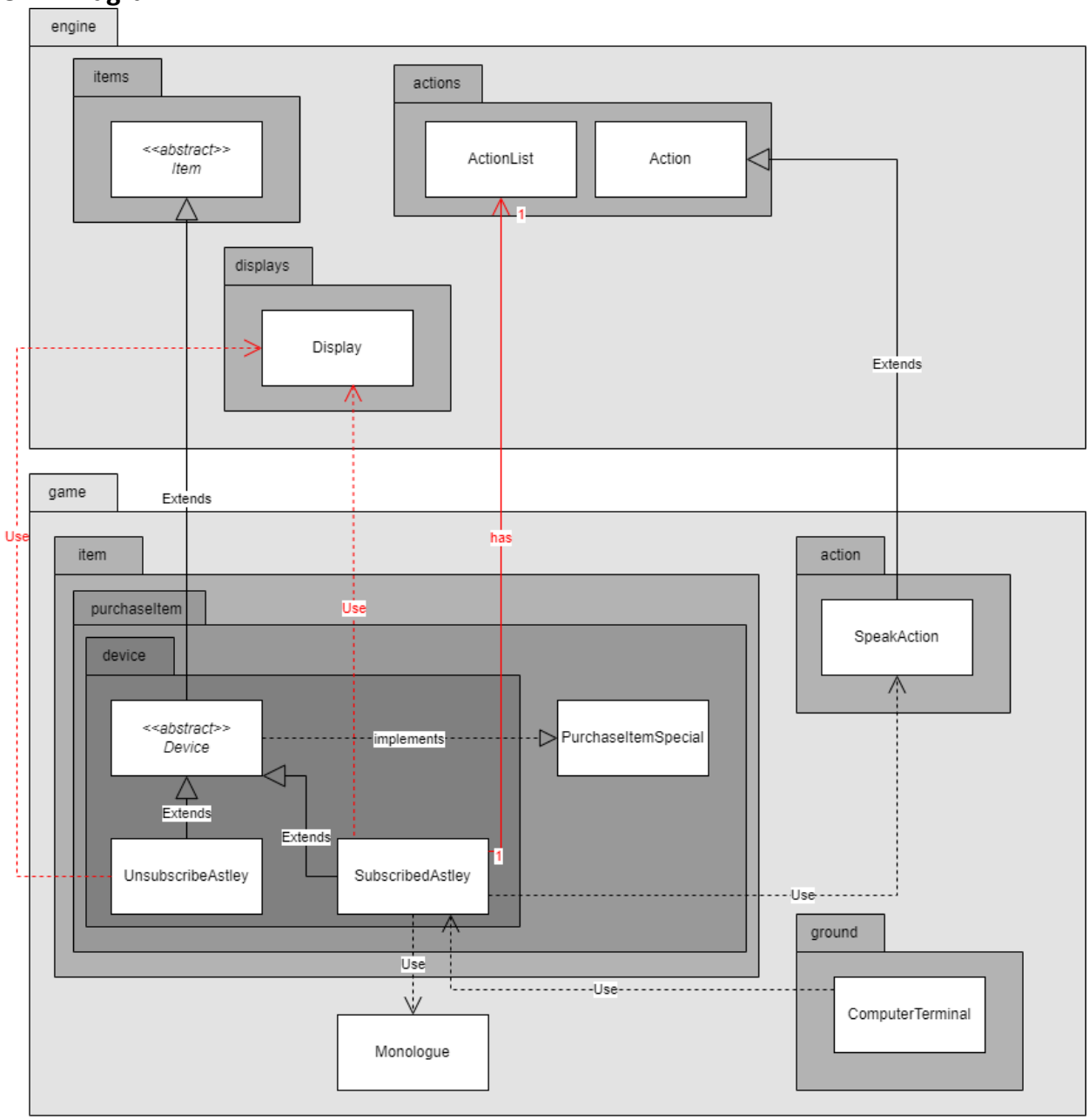
Assignment 3 Design Rationale

Requirement 3: dQw4w9WgXcQ

UML Diagram:



Design Goal:

The design goal for this assignment is to add a purchasable AI device – Astley that can be interacted with the player when bought with various monologues based on the player's condition while adhering closely to relevant design principles and best practices.

Design Decision:
In implementing the Ai Device – Astley with subscription needed, the decision was made to make a hierarchy of classes that consists of a base class called Device and two specialised classes called SubscribedAstley and UnsubscribeAstley that manage the subscription logic and other device states. This strategy makes it possible to clearly control the state and interactions of the device while isolating particular behaviours into their appropriate classes.

Alternative Design:
One alternative approach could involve integrating all features into a single class that controls the interactions and state of the device. However, the chosen design is more preferred because it is more easier to modify, extend and maintainance.

Analysis of Alternative Design:
The alternative design is not ideal because it violates various Design and SOLID principles:

1. **Single Responsibility Principle (SRP):**
    - It would be challenging to maintain and expand the single class since it would handle several tasks, including interaction handling, subscription logic, and state management.

2. **Open/Closed Principle (OCP):**
    - Modifications to the same class would be necessary for any changes to the subscription logic or interaction handling, which would raise the possibility of creating issues.

3. **Interface Segregation Principle (ISP):**
    - The class would have to implement multiple unrelated methods, leading to a bloated interface.

Final Design:
In our design, we closely adhere to SOLID principles:

1. **Single Responsibility Principle (SRP):**
    - **Application**: SubscribedAstley covers behaviours linked to subscriptions, UnsubscribeAstley represents the state when the subscription is dormant, and the Device class handles general device functionalities.
    - **Why:** This division of responsibilities improves the code's readability and makes maintenance easier.

- **Pros/Cons:** simplifies testing and code maintenance, but may necessitate handling several classes.

2. **Open/Closed Principle (OCP):**
    - **Application:** By extending the Device class, additional device kinds or subscription models can be introduced to the design without changing the existing code.
    - **Why:** This method improves the system's scalability and adaptability.
    - **Pros/Cons:** Facilitates additions in the future, but it may make the original design more complex.
3. **Interface Segregation Principle (ISP):**
    - **Application:** To maintain a clear division of responsibilities, each class defines just those methods that are pertinent to its particular function.
    - **Why:** Minimises superfluous dependencies and stops the development of bloated interfaces.
    - **Pros/Cons:** Guarantees more organised and modular code, however interface design must be done carefully.
4. **Liskov Substitution Principle (LSP):**
    - **Application:** Unsubscribe and Subscribed ImmediatelyAstley conforms to the contracts specified by the Device class and can be used interchangeably without compromising the validity of the programme.
    - **Why:** This principle guarantees that classes that are derived from a base class can replace it without changing the anticipated behaviour.
    - **Pros/Cons:** Preserves the application's integrity, although strict adherence to base class contracts is necessary.
5. **Dependency Inversion Principle (DIP):**
    - **Application:** Rather than relying on particular SubscribedAstley or UnsubscribeAstley implementations, the subscription logic is dependent on the abstraction offered by the Device class.
    - **Why:** Encourages decoupling and increases the design's adaptability.
    - **Pros/Cons:** Promotes a more modular framework, albeit more layers of abstraction might be needed.

Conclusion:

Overall, our chosen design provides a robust framework for adding the AI device – Astley to the game. By carefully SOLID principle and the requirement, we have developed a solution that is maintainable, extensible, and adheres to best practices, paving the way for future enhancements, extensions, optimizations.