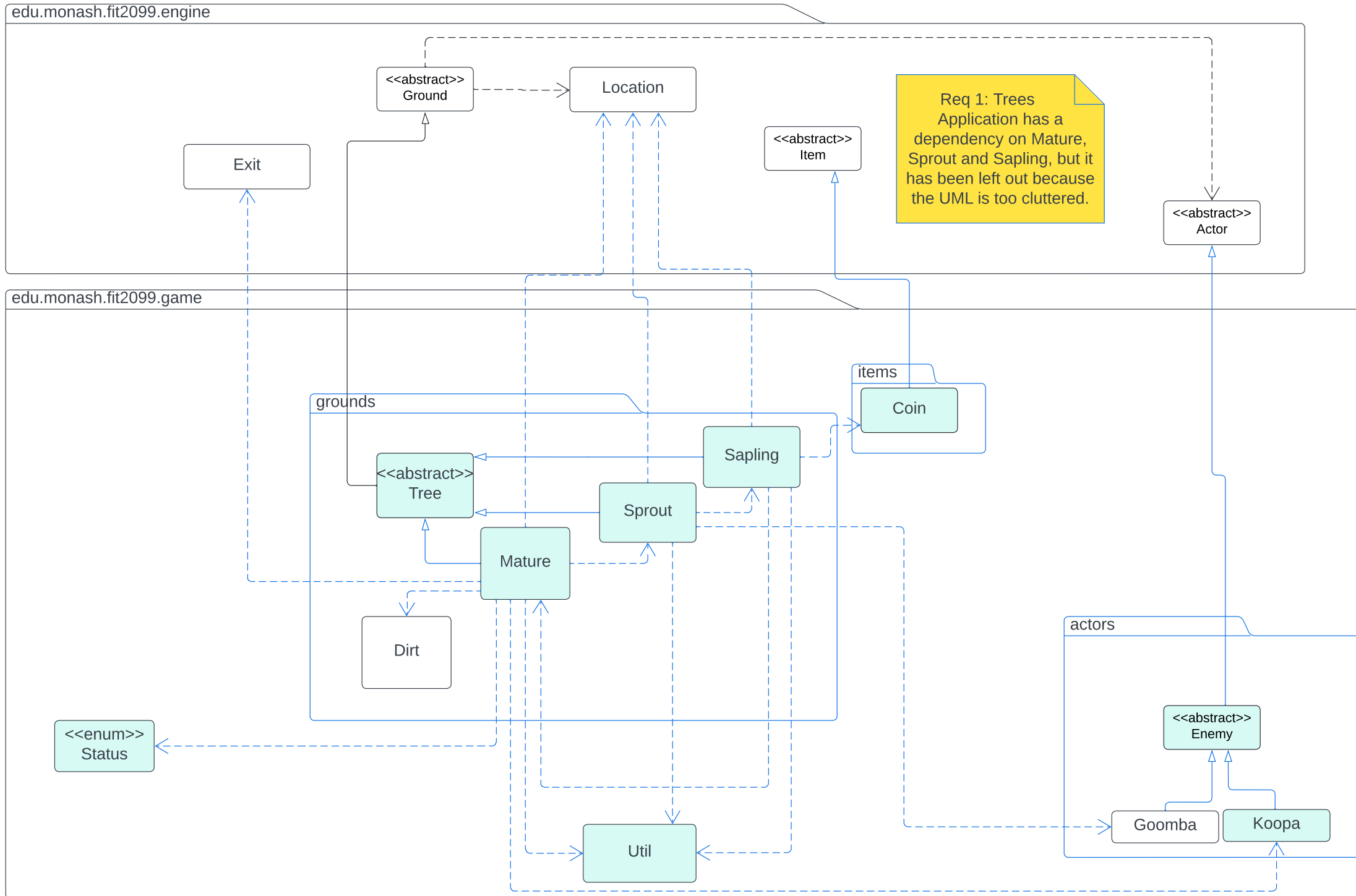
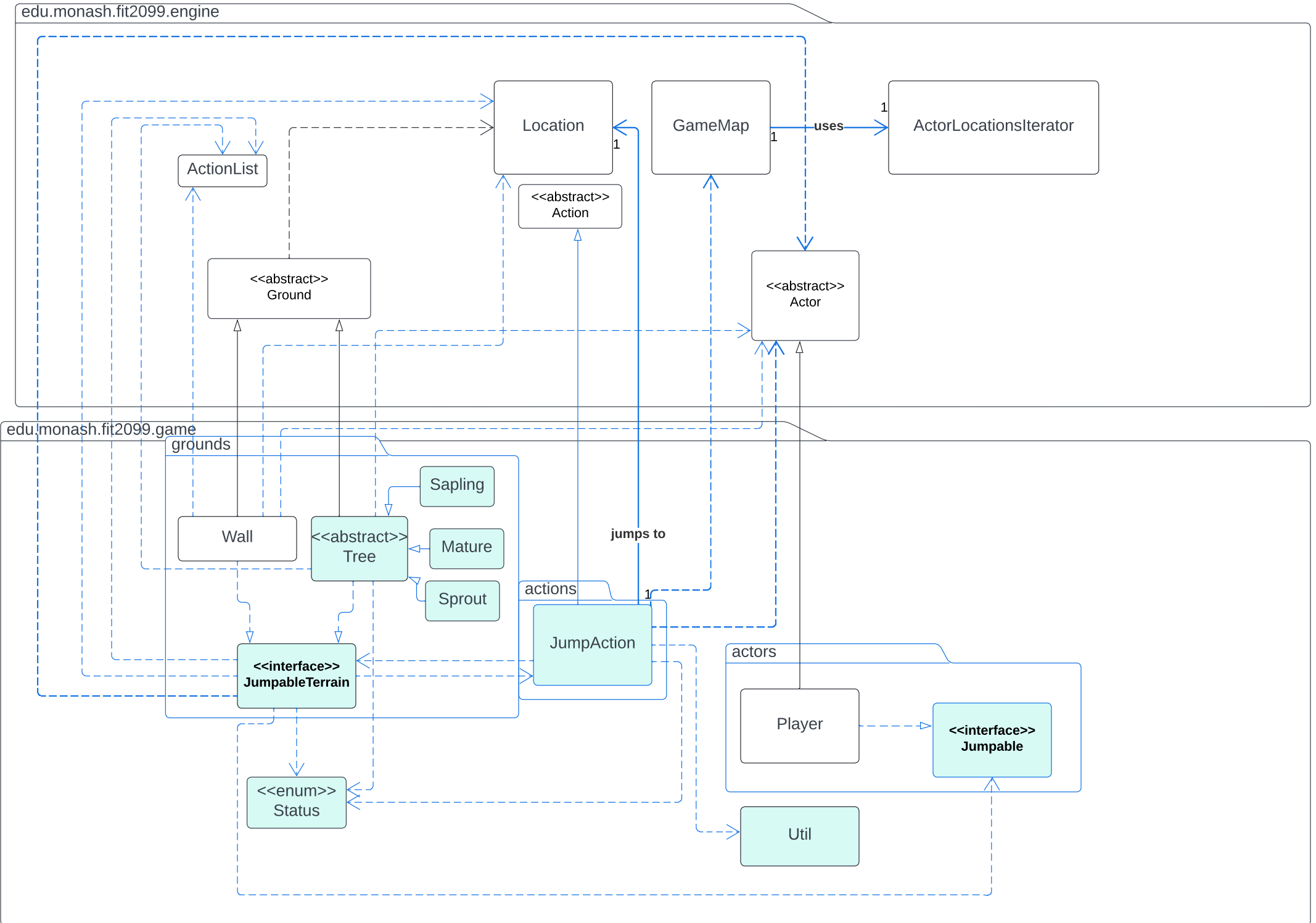


edu.monash.fit2099.engine



Req 2: Jump

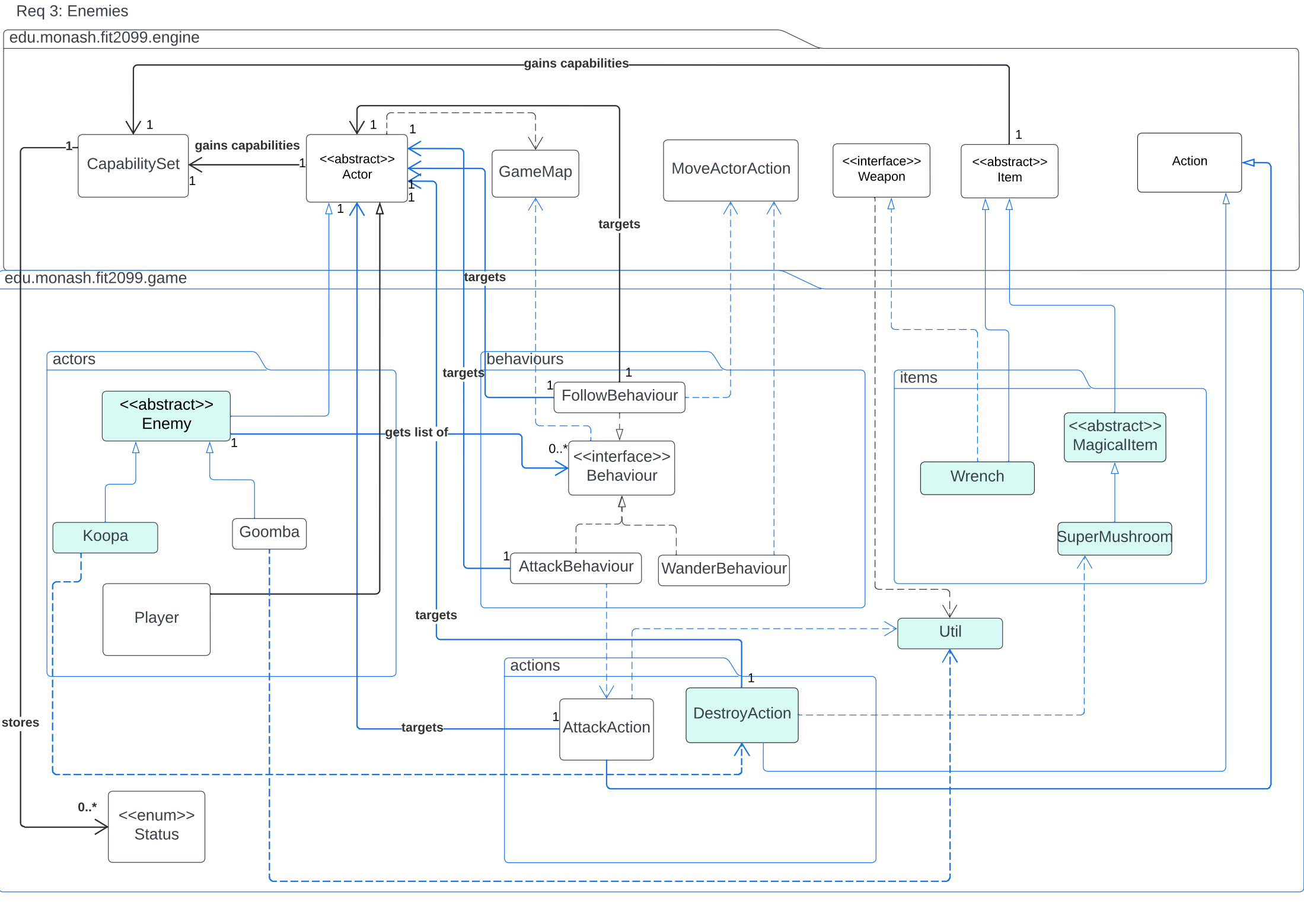


```

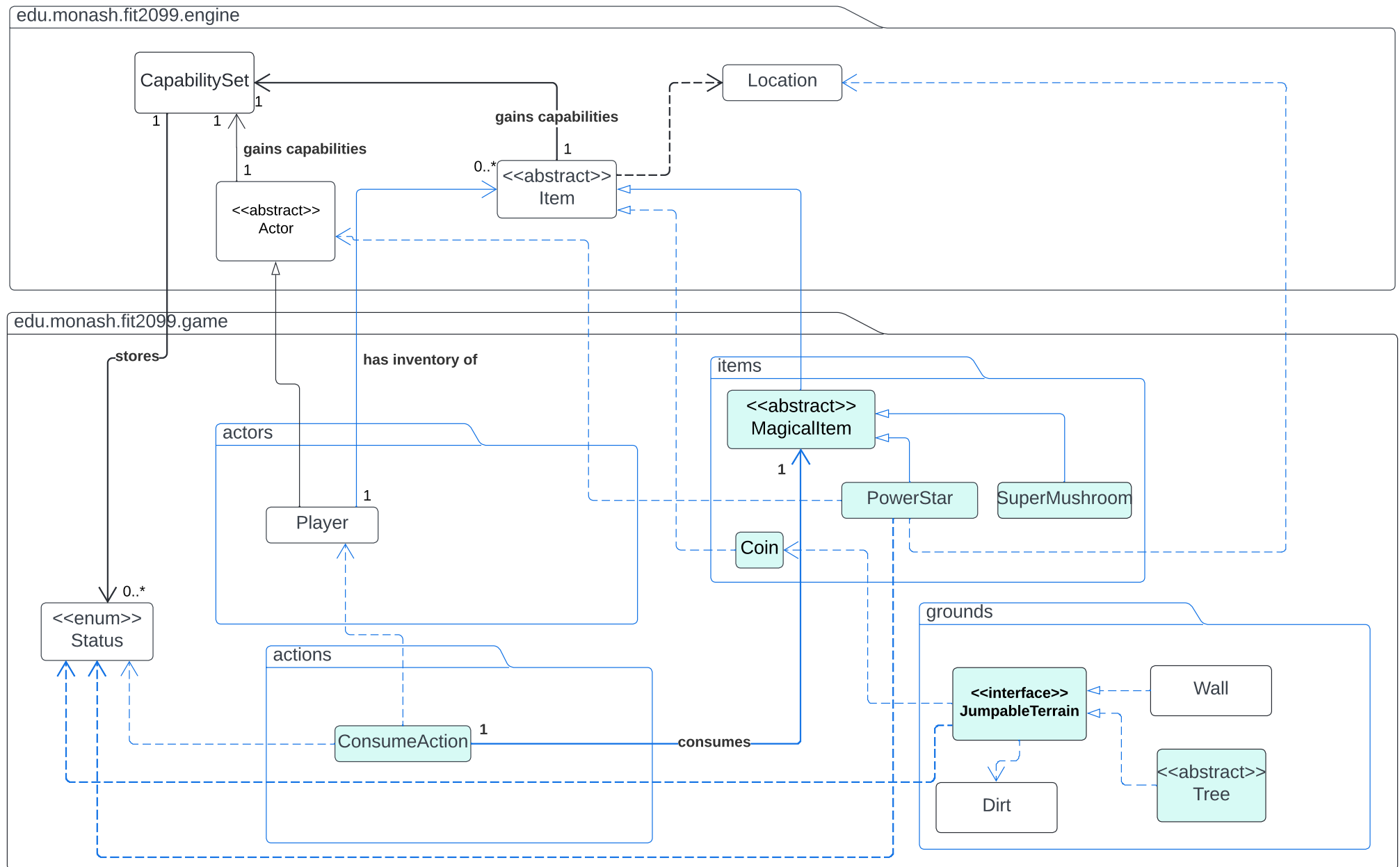
classDiagram
    class CapabilitySet
    class Actor["<<abstract>> Actor"]
    class GameMap
    class MoveActorAction
    class Weapon["<<interface>> Weapon"]
    class Item["<<abstract>> Item"]
    class Action
    class Enemy["<<abstract>> Enemy"]
    class Koopa
    class Goomba
    class Player
    class FollowBehaviour
    class Behaviour["<<interface>> Behaviour"]
    class AttackBehaviour
    class WanderBehaviour
    class Wrench
    class MagicalItem["<<abstract>> MagicalItem"]
    class SuperMushroom
    class Util
    class AttackAction
    class DestroyAction
    class Status["<<enum>> Status"]

    CapabilitySet --> Actor : gains capabilities
    Actor --> GameMap : targets
    Actor --> MoveActorAction : targets
    Actor --> Weapon : targets
    Actor --> Item : targets
    Actor --> Action : targets
    Actor --> Enemy : gets list of
    Actor --> FollowBehaviour : targets
    Actor --> AttackBehaviour : targets
    Actor --> AttackAction : targets
    Actor --> DestroyAction : targets
    Actor --> Status : stores
    GameMap --> Actor : targets
    MoveActorAction --|> Action
    Weapon --|> Item
    Item --|> Action
    Enemy --|> Actor
    Koopa --|> Enemy
    Goomba --|> Enemy
    Player --|> Actor
    FollowBehaviour --|> Behaviour
    AttackBehaviour --|> Behaviour
    WanderBehaviour --|> Behaviour
    Wrench --|> MagicalItem
    SuperMushroom --|> MagicalItem
    Util --|> Action
    AttackAction --|> Action
    DestroyAction --|> Action
    Status --> Actor : stores

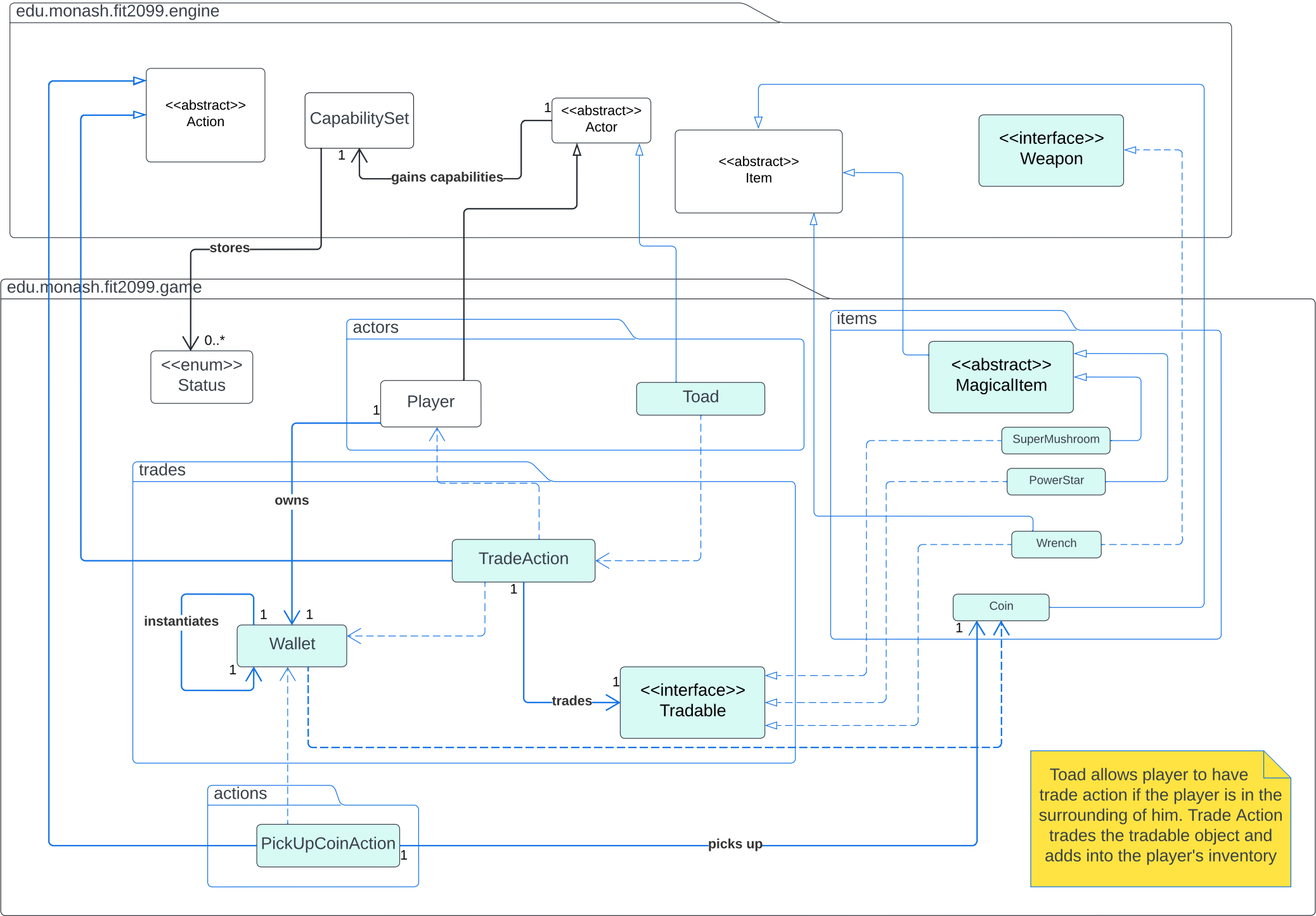
```

[illegible]

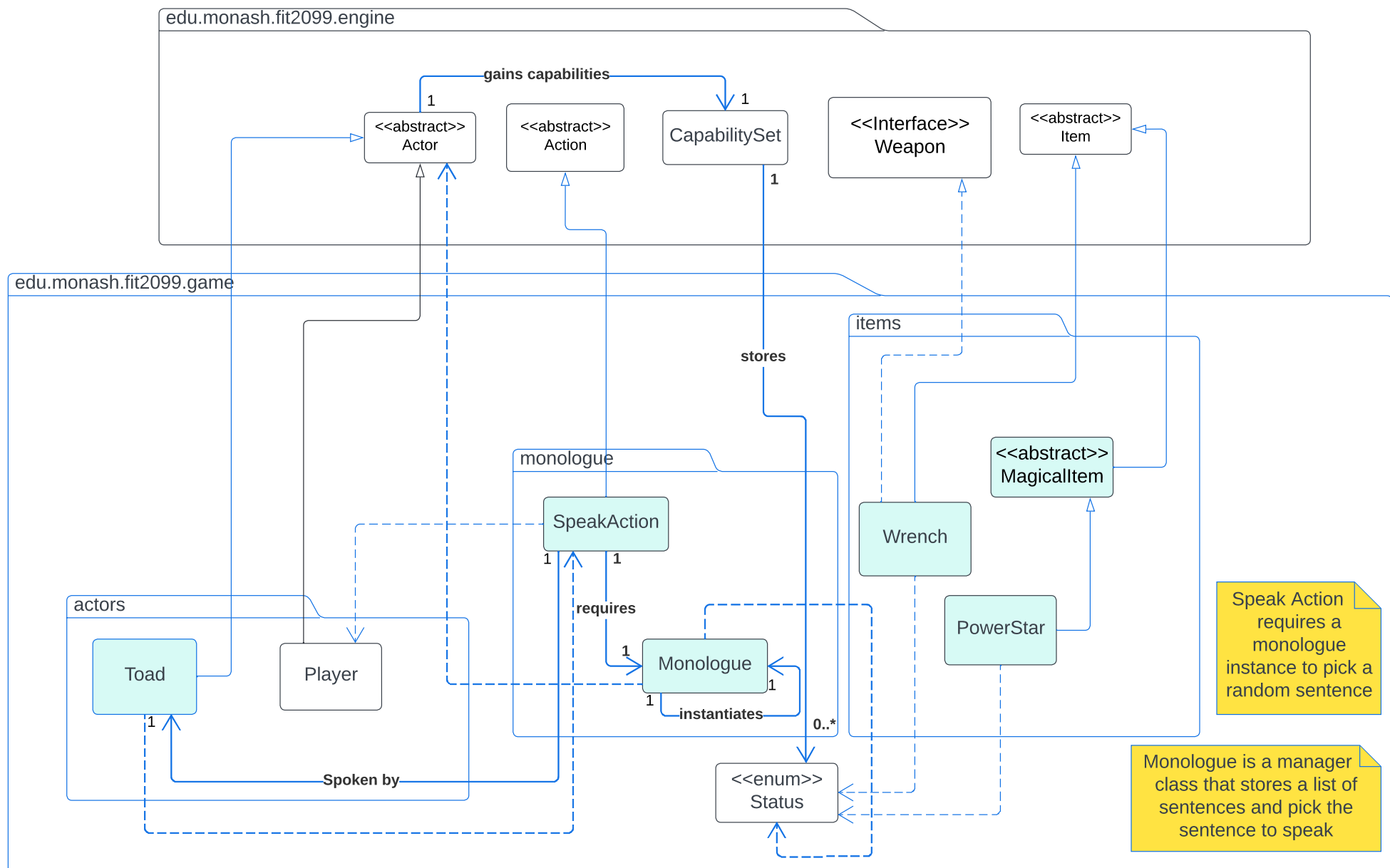
Req 4: Magical Items



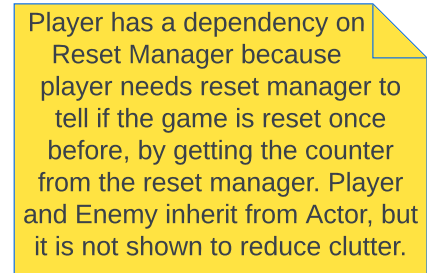
Req 5: Trading



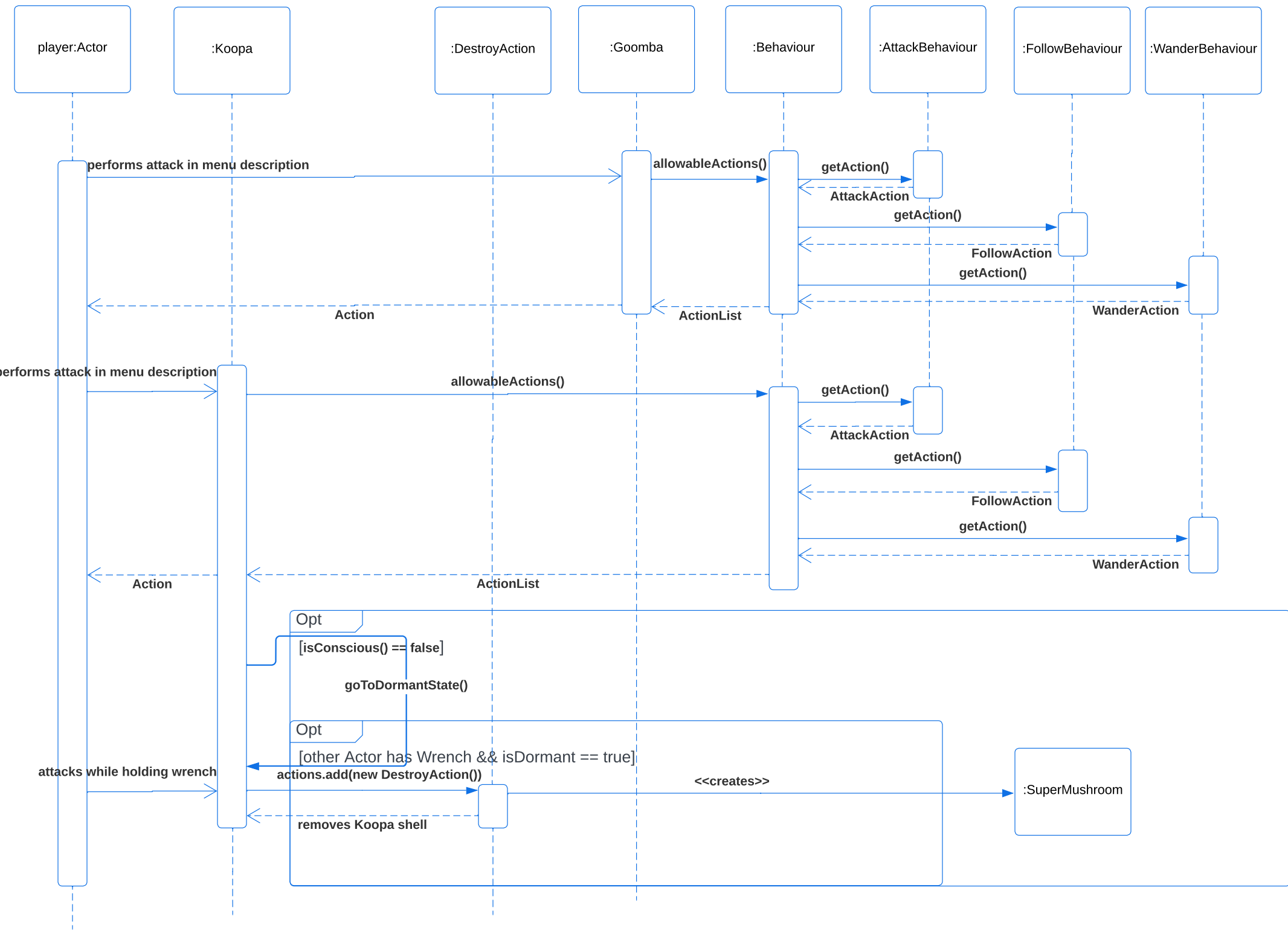
Req 6: Monologue



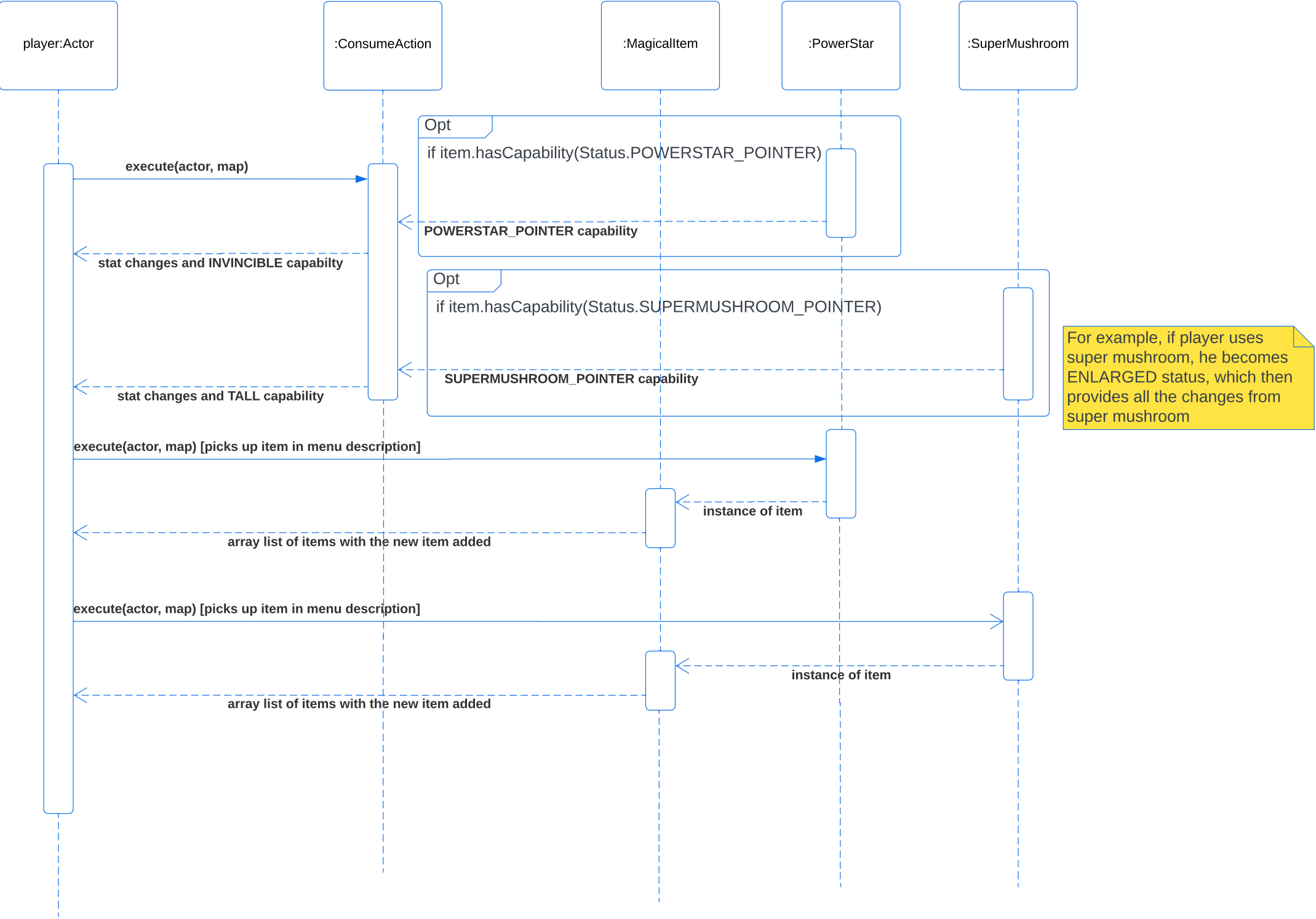
edu.monash.fit2099.engine



Sequence Diagram for Req 3



Sequence Diagram for Req 4



Sequence Diagram for Req 5

Tradable is an interface for any object that can be traded

TradeAction class gets the price from the tradable object and gets the balance from player's wallet to perform the transaction. If a transaction is successful, the wallet's balance is updated, and the tradable object is added to the player's inventory. A message is then printed out.

