Assignment 1

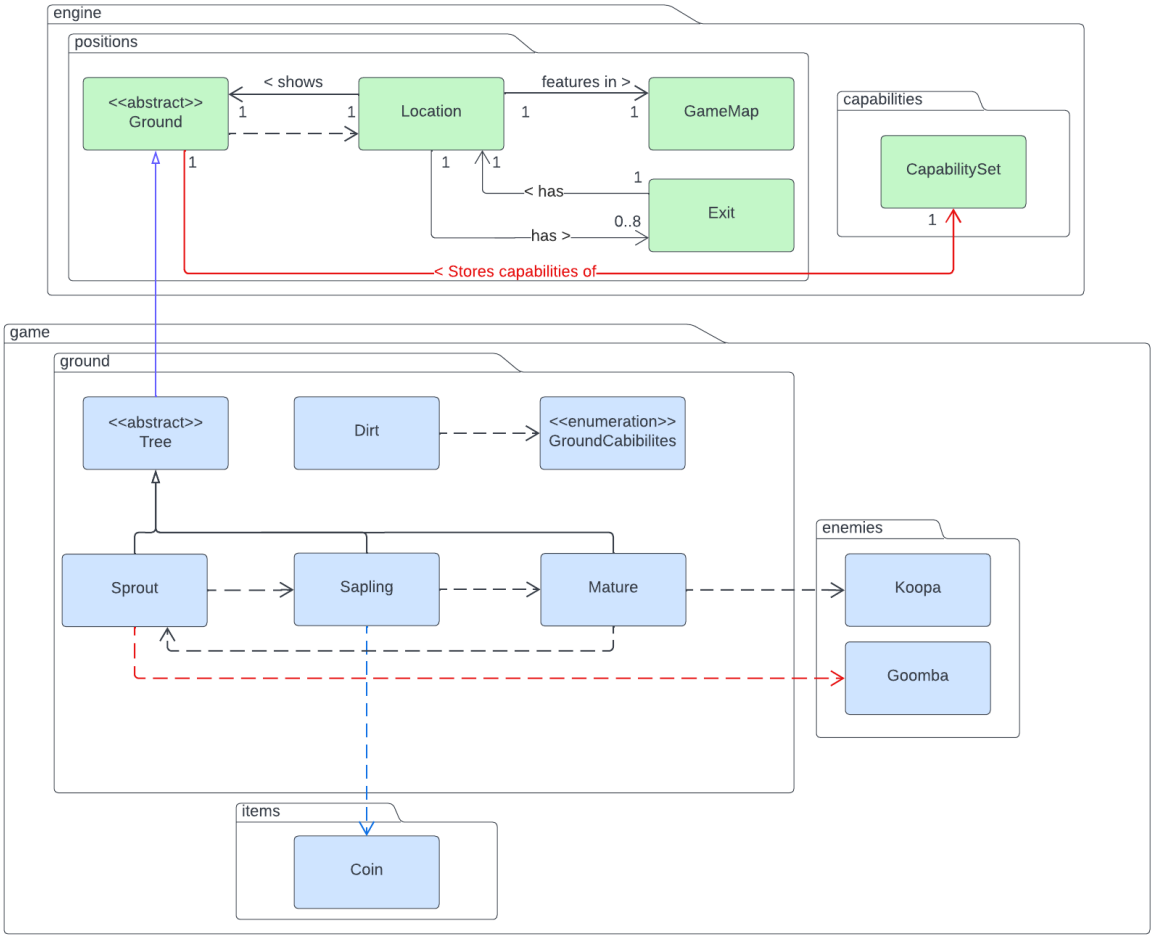
Matthew Siegenthaler, James Huynh, Chun Kau Mok

Contents

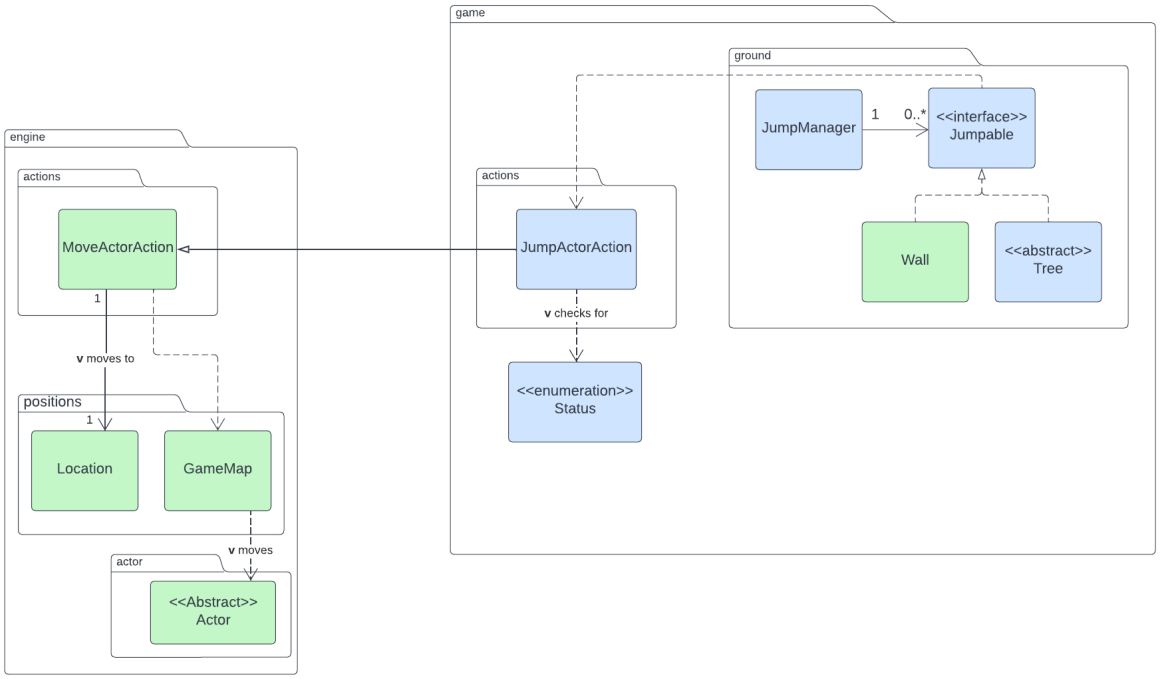
Class diagrams 2
REQ1 2
REQ2 3
REQ3 4
REQ4 5
REQ5 5
REQ6 6
REQ7 6
Sequence diagrams 7
REQ1 Mature.tick() 7
REQ2 JumpActorAction.execute() 8
REQ5 trading 8

Class diagrams

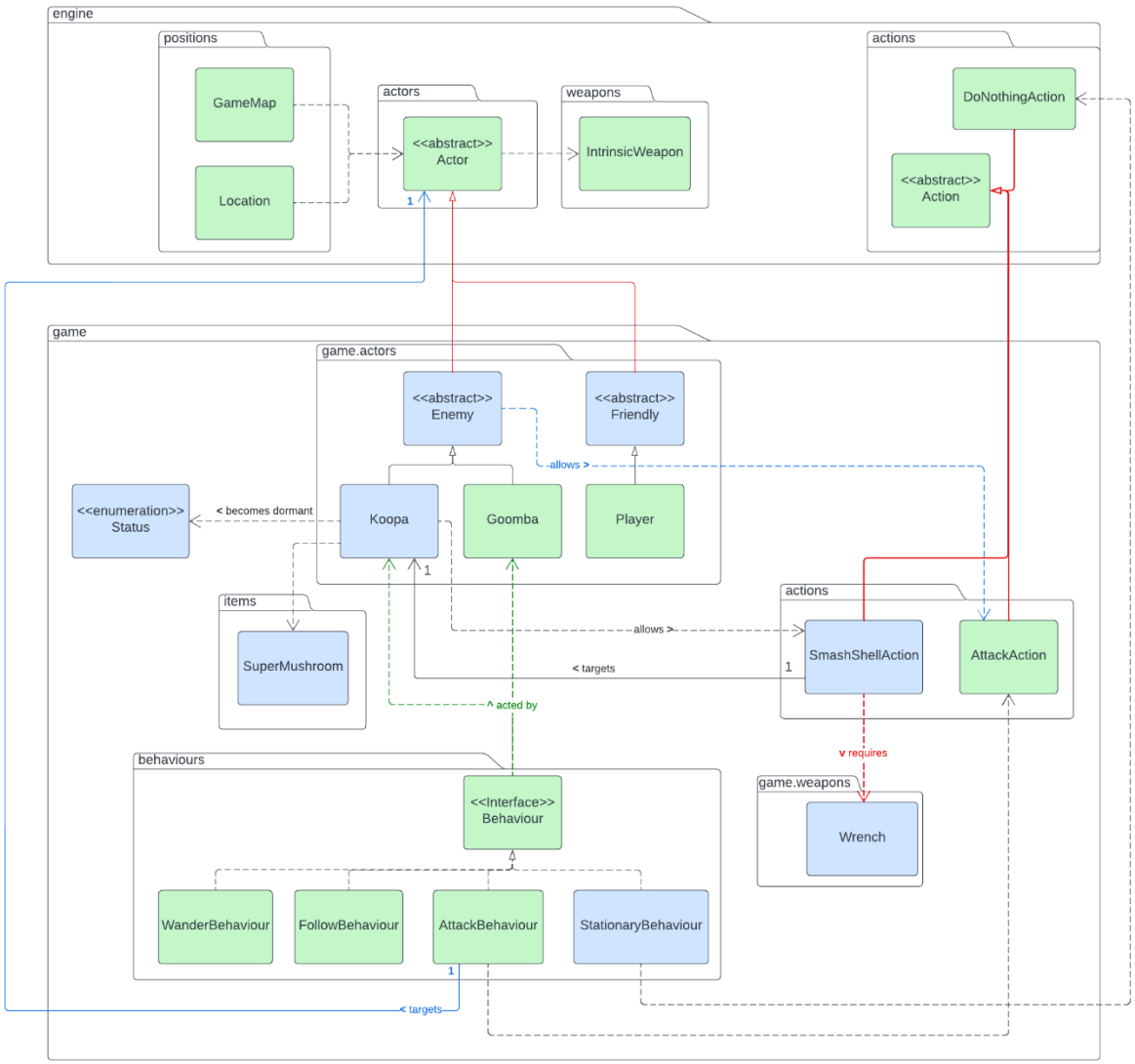
REQ1



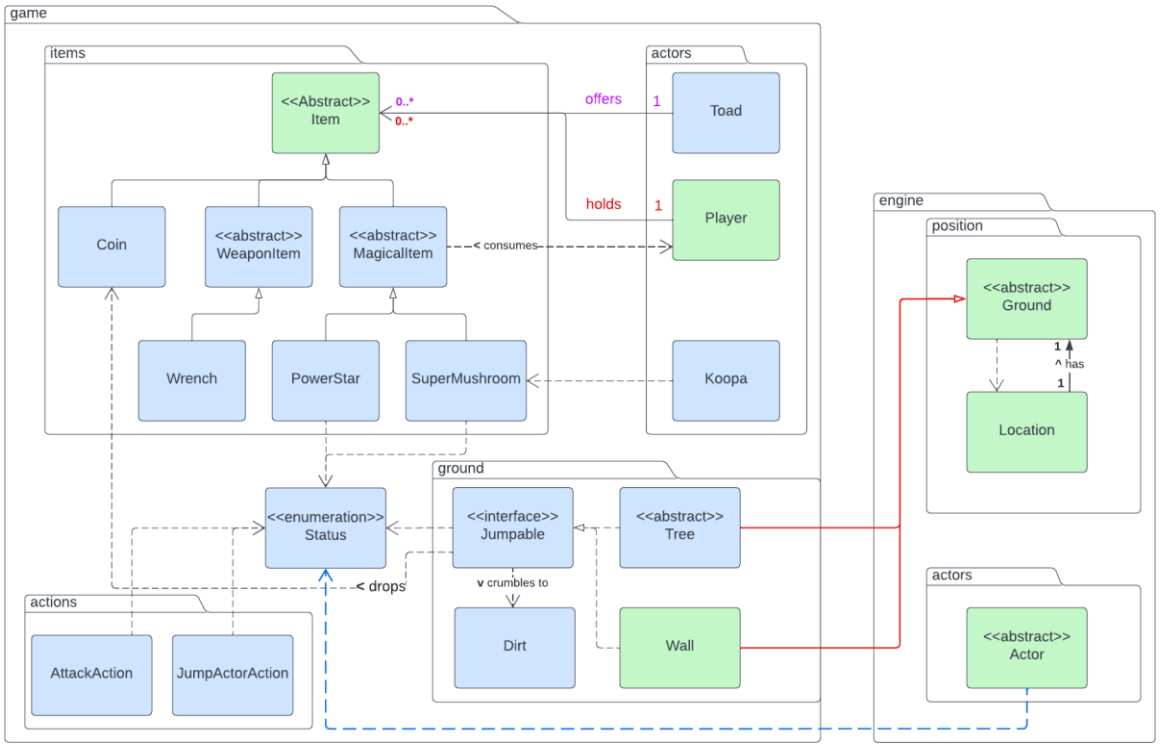
REQ2


game
ground
JumpManager
1
0..*
<<interface>>
Jumpable
Wall
<<abstract>>
Tree
actions
JumpActorAction
v checks for
<<enumeration>>
Status
engine
actions
MoveActorAction
1
v moves to
positions
1
Location
GameMap
v moves
actor
<<Abstract>>
Actor

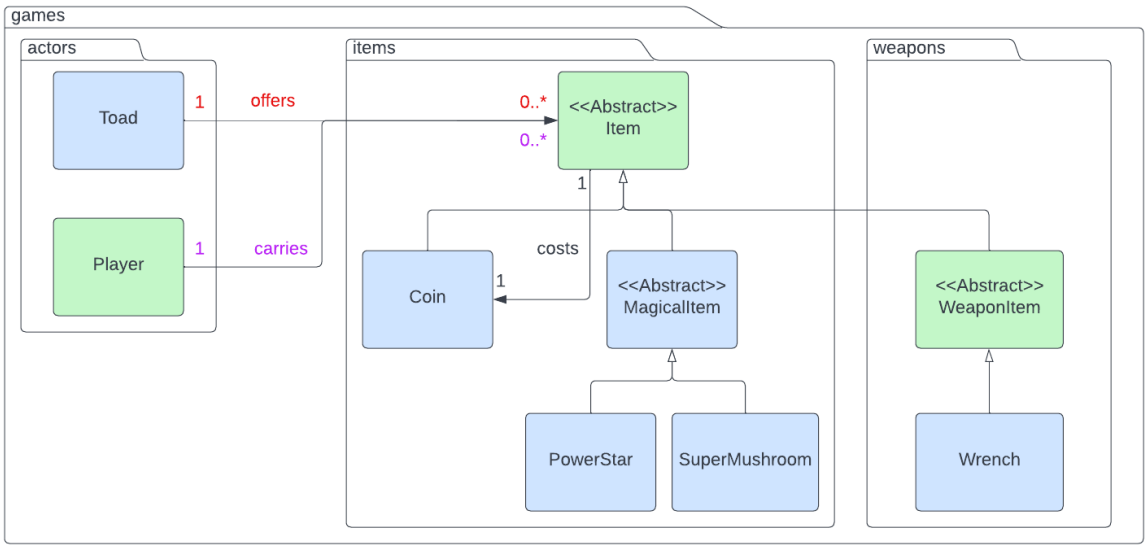
REQ3


engine
positions
GameMap
Location
actors
<<abstract>>
Actor
1
weapons
IntrinsicWeapon
actions
DoNothingAction
<<abstract>>
Action
game
game.actors
<<abstract>>
Enemy
<<abstract>>
Friendly
allows >
<<enumeration>>
Status
< becomes dormant
Koopa
Goomba
Player
1
items
SuperMushroom
actions
allows >
SmashShellAction
AttackAction
1
< targets
^ acted by
v requires
behaviours
<<Interface>>
Behaviour
game.weapons
Wrench
WanderBehaviour
FollowBehaviour
AttackBehaviour
StationaryBehaviour
1
< targets

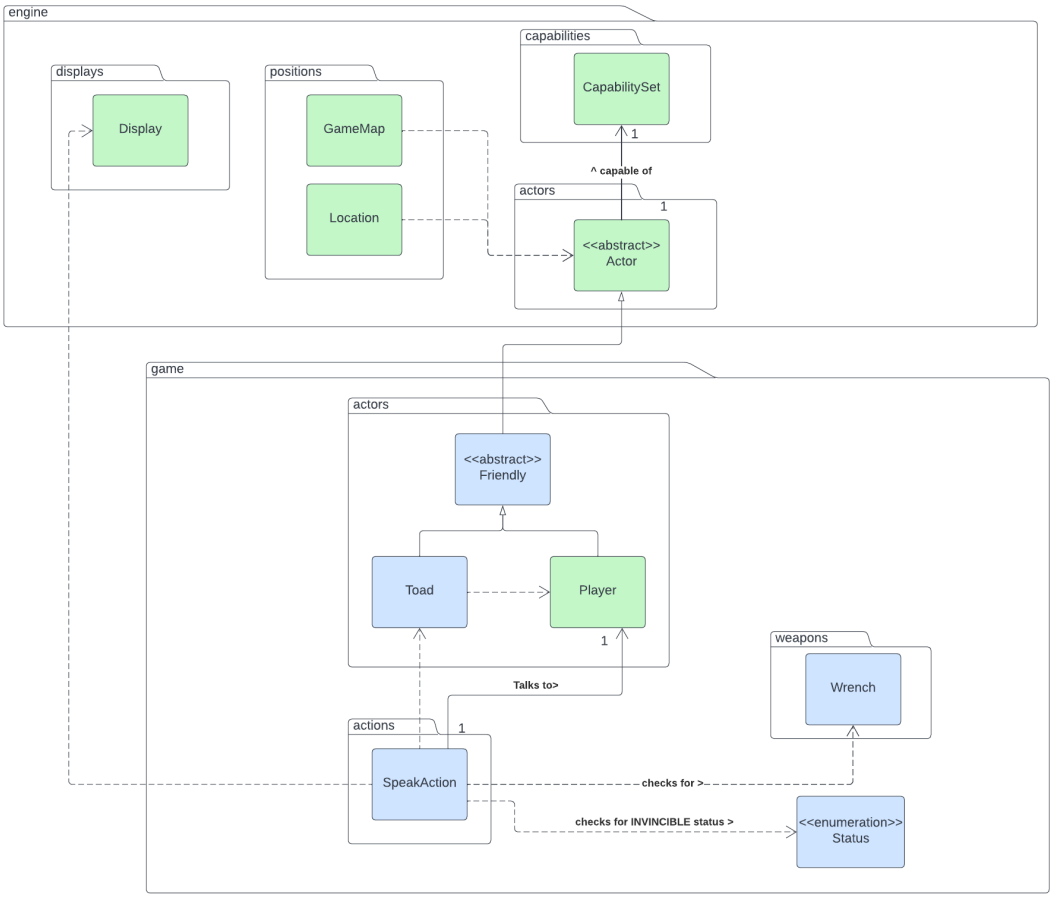
REQ4


game
items
actors
<<Abstract>>
Item
0..*
0..*
offers
1
Toad
holds
1
Player
Coin
<<abstract>>
WeaponItem
<<abstract>>
MagicalItem
< consumes
Wrench
PowerStar
SuperMushroom
Koopa
engine
position
<<abstract>>
Ground
1
^ has
1
Location
ground
<<enumeration>>
Status
<<interface>>
Jumpable
<<abstract>>
Tree
v crumbles to
< drops
Dirt
Wall
actions
AttackAction
JumpActorAction
actors
<<abstract>>
Actor

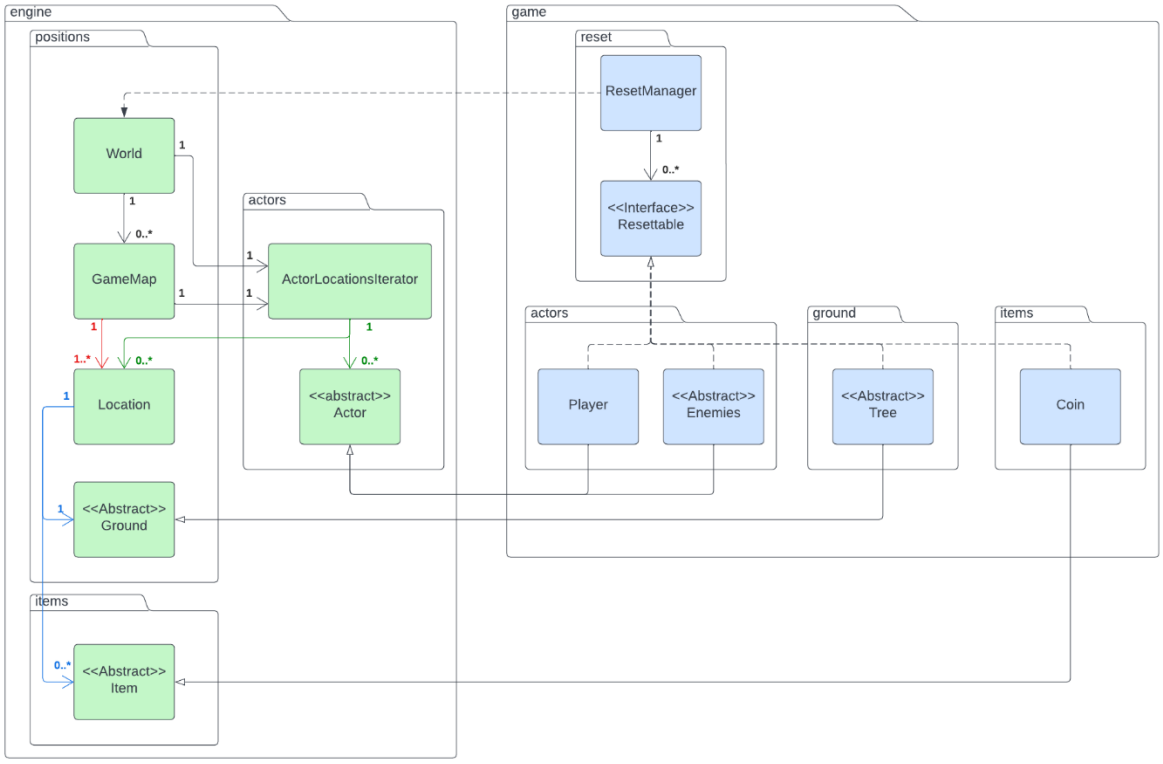
REQ5


games
actors
Toad
1
offers
Player
1
carries
items
0..*
0..*
<<Abstract>>
Item
1
costs
Coin
1
<<Abstract>>
MagicalItem
PowerStar
SuperMushroom
weapons
<<Abstract>>
WeaponItem
Wrench

REQ6

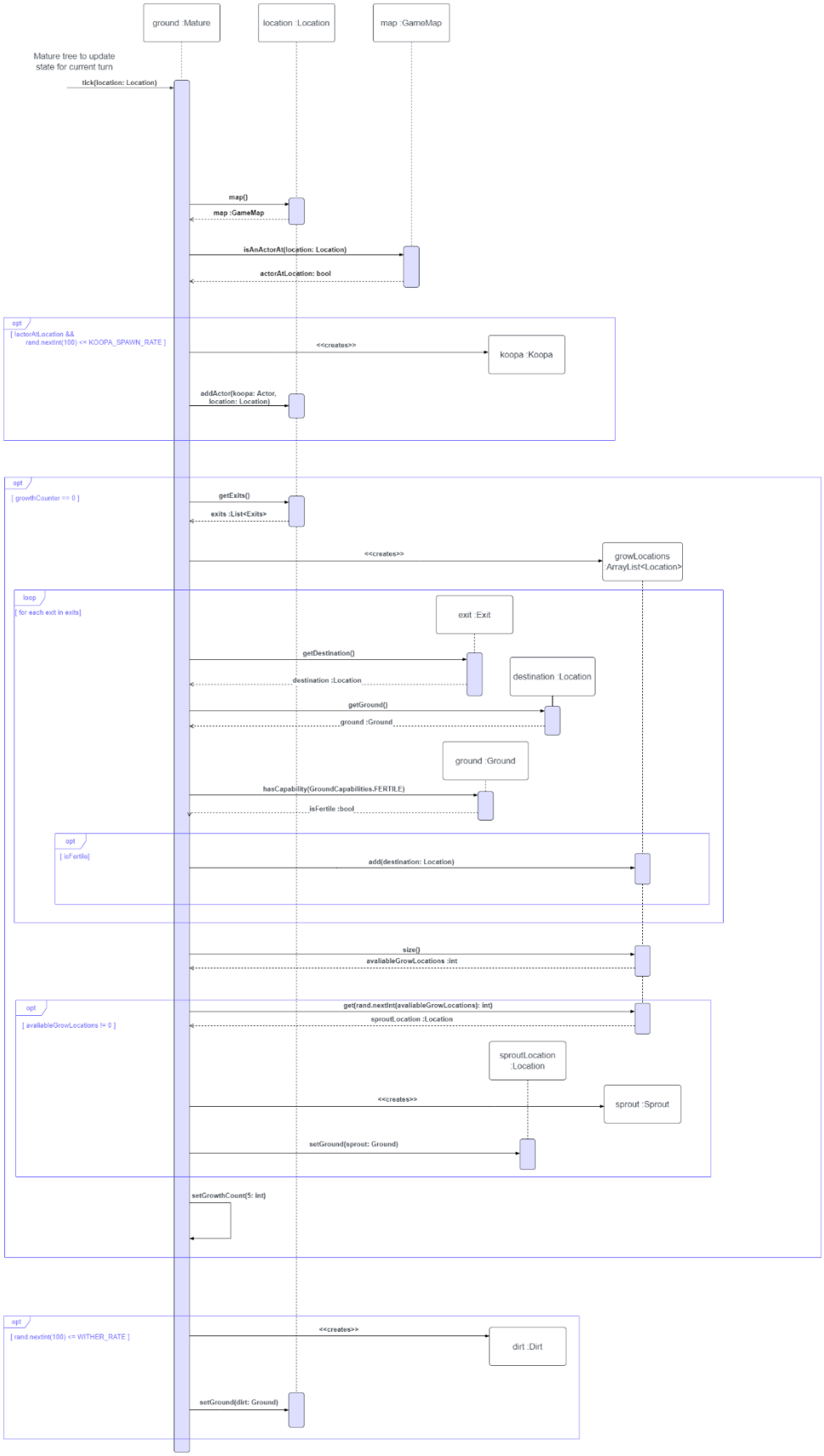

engine
displays
Display
positions
GameMap
Location
capabilities
CapabilitySet
1
^ capable of
actors
1
<<abstract>>
Actor
game
actors
<<abstract>>
Friendly
Toad
Player
1
weapons
Wrench
Talks to>
actions
1
SpeakAction
checks for >
checks for INVINCIBLE status >
<<enumeration>>
Status

REQ7

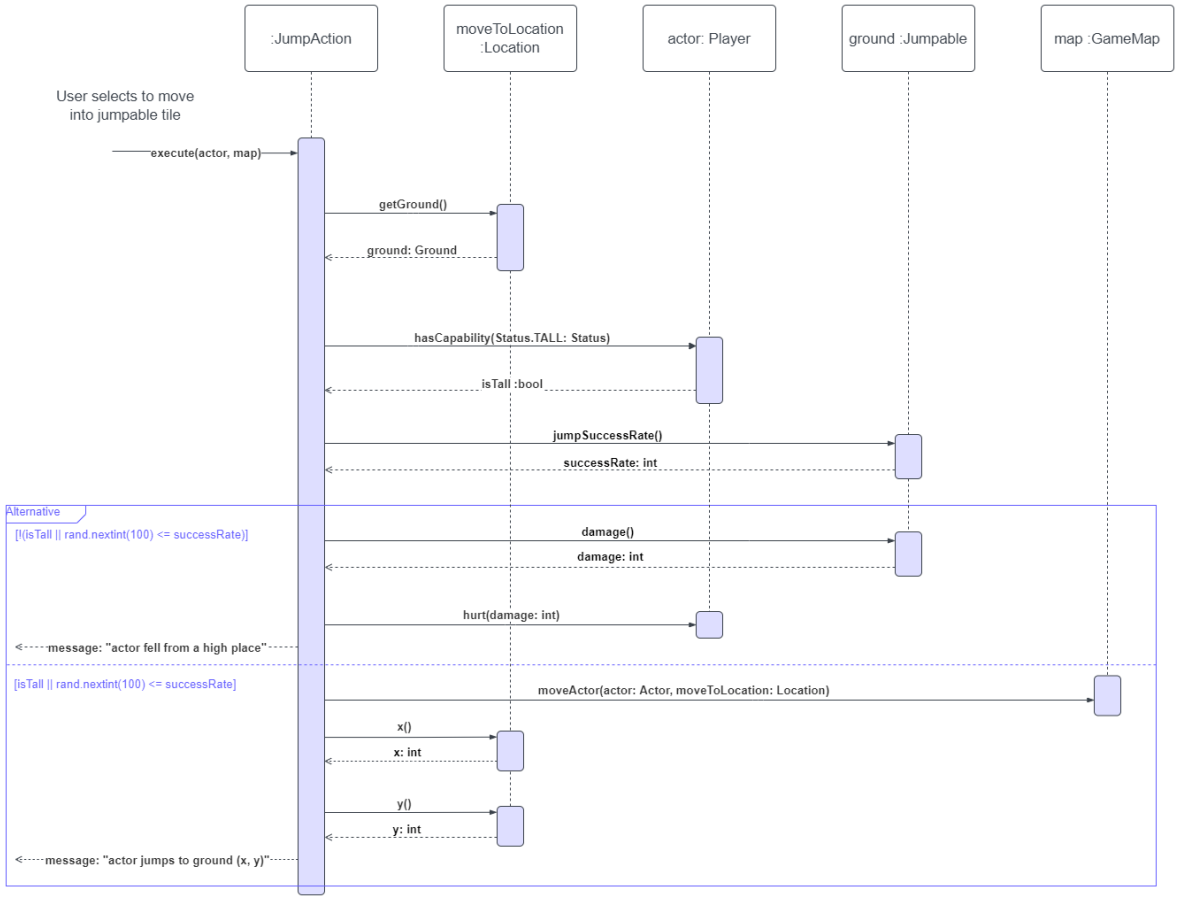
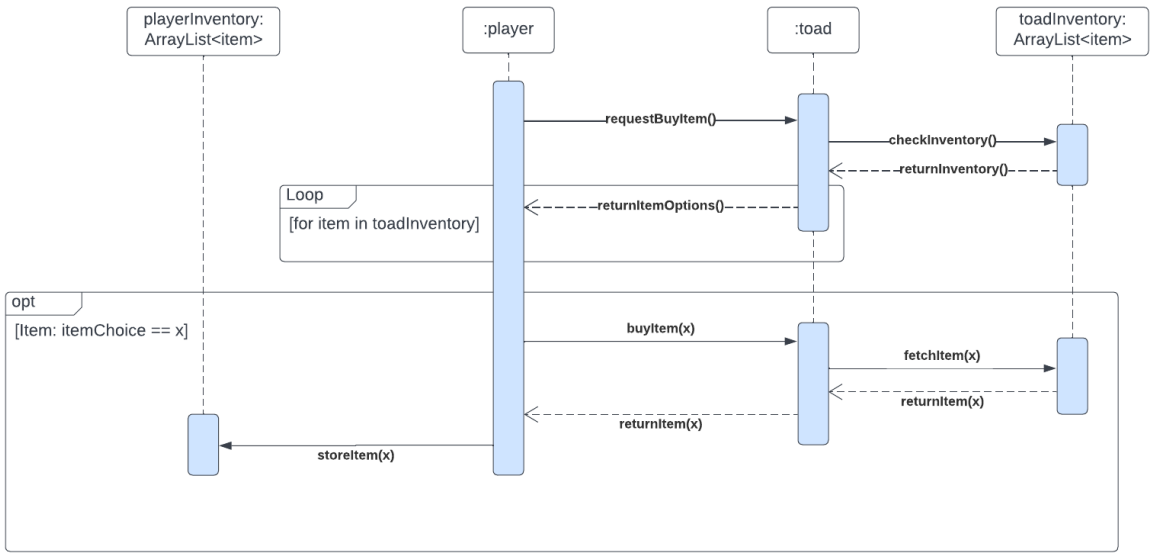

engine
positions
World
1
1
0..*
GameMap
1
1
1
1..*
0..*
Location
1
<<Abstract>>
Ground
1
actors
1
1
ActorLocationsIterator
1
0..*
<<abstract>>
Actor
items
0..*
<<Abstract>>
Item
game
reset
ResetManager
1
0..*
<<Interface>>
Resettable
actors
Player
<<Abstract>>
Enemies
ground
<<Abstract>>
Tree
items
Coin

Sequence diagrams

REQ1 Mature.tick()



REQ2 JumpActorAction.execute()



REQ5 trading