

Sequence Diagram 1 (Trees) Documentation :

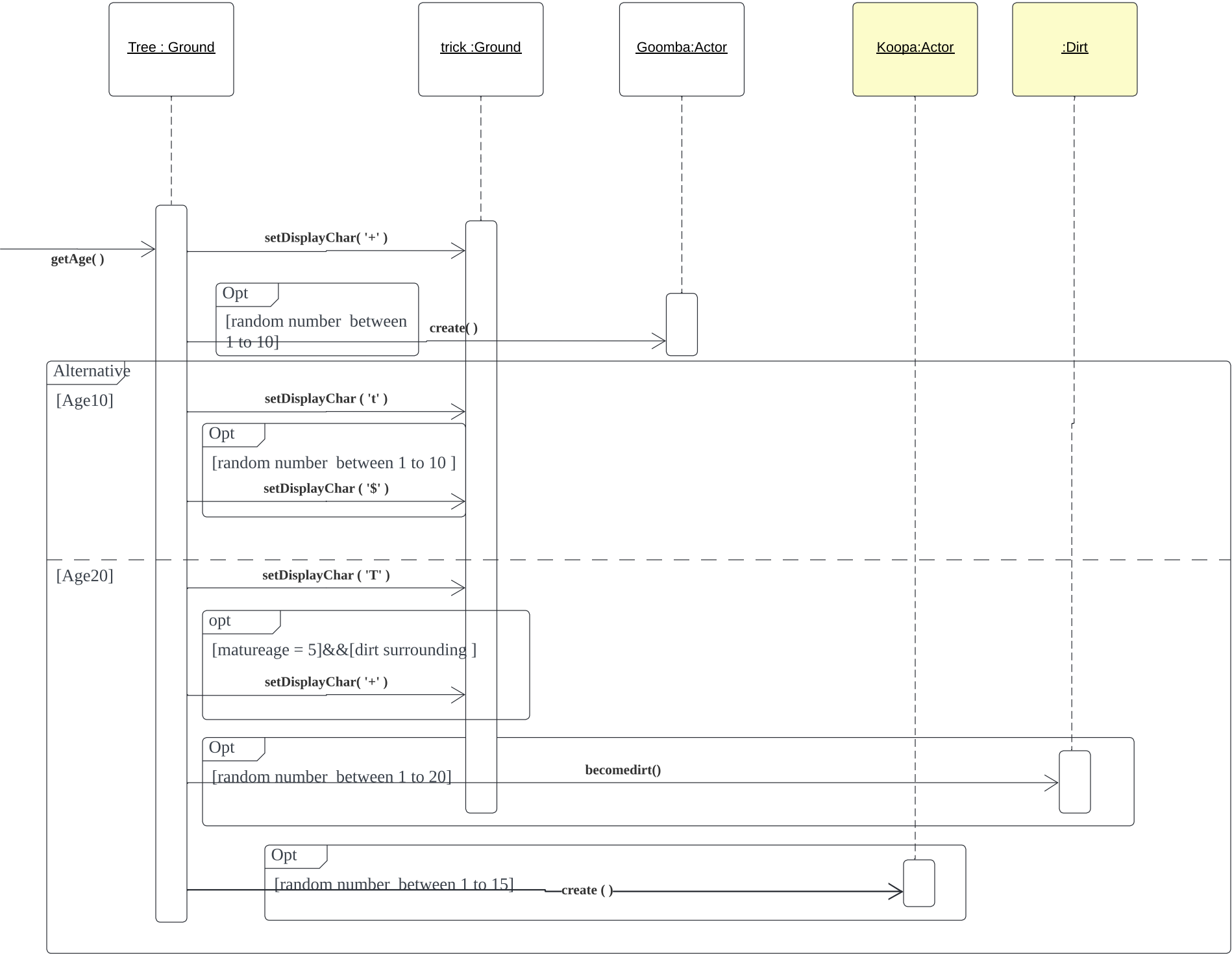
getAge(): The tree calss will have a age attrubute and this attribute will store the number of the turns.

Tree will use its parent method (super) to setDiaplayChar()
3 types of tree .

For the sprout it will generate random number to create the goomba in 10%.

The second type of the tree (Sapling), this type of tree will display itself as well using method form parent. And a if function will drops coins in 10% rate.

Matrue trees will change display to T, and it will have a mature age to set sprout around . And it will change dirt in 20% rate each turn and create koopa in 15%rate each turn.



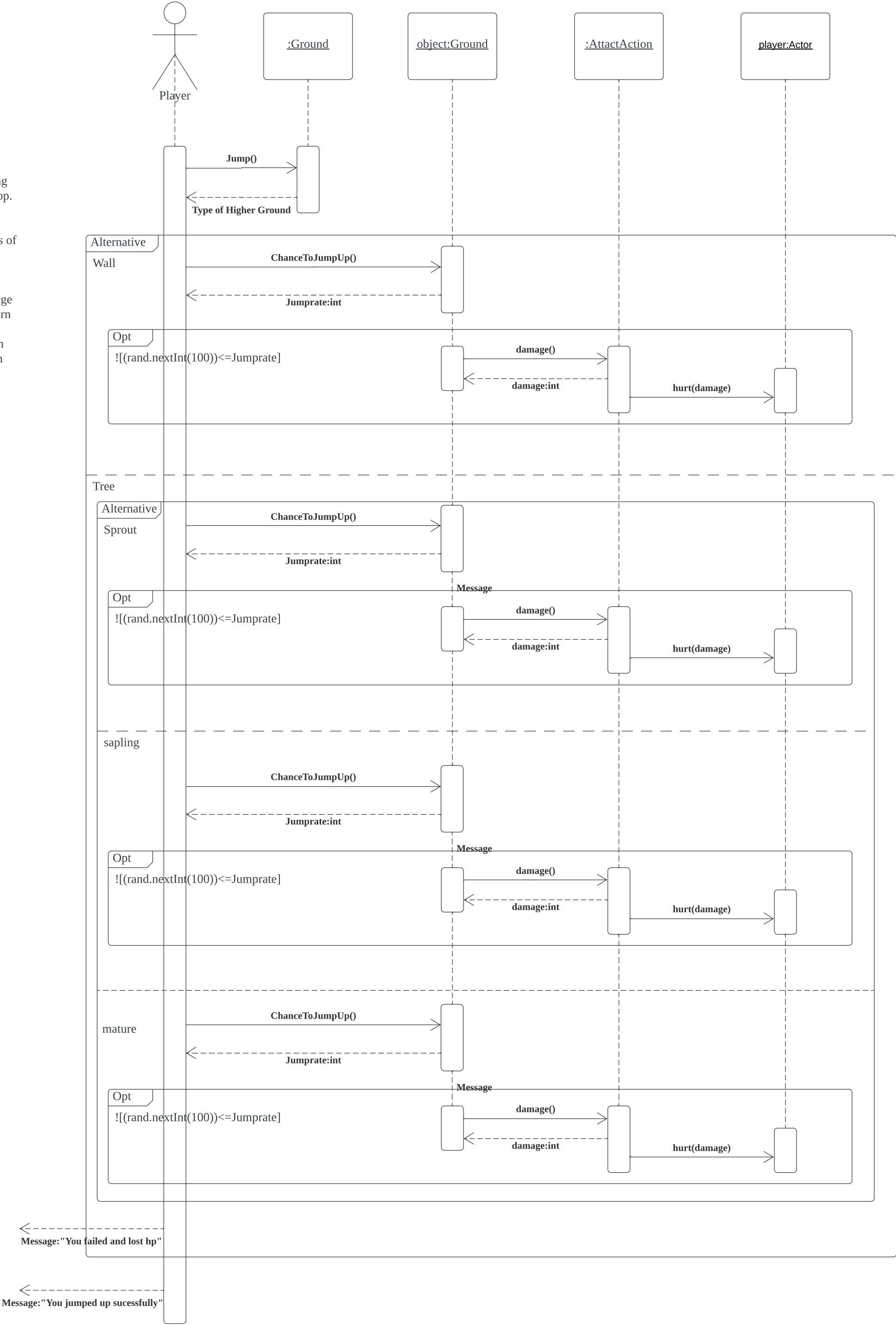
Sequence Diagram 2 (Jump) Documentation:

When the program runs into the alternative statement, it need a input to compare with the statement, Jump() is a new method I created, it will return the type of the ground that player's moving will meet. The return value will be caught by the alternative loop. We have 2 different situations: Wall and Tree.

And inside the Tree statement, I divided it into 3 different kinds of Tree: Sprout, sapling and Mature tree.

For the rand.nextInt(100) part, it should indicates the random number that computer assigned to the player each turn. The range is from 1 to 100. And the ChanceToJumpUp() method will return each Ground's type's rate to jump. All the opt loops will compare the random number created each turn with the jump rate, and it will return a damage which is an attack action to player.

Jump() -> return Type
alternative (Type == type) -> get Jump rate: int
random number > jump rate -> damage() -> return damage: int
damage:int -> hurt(damage)



Sequence Diagram 3 (Enemies) Documentation:

When the enemy is ready to attack the player, it will call the allowableActions() method, it will return the ActionList which only has attackAction for enemies. So it will attack player at that turn.

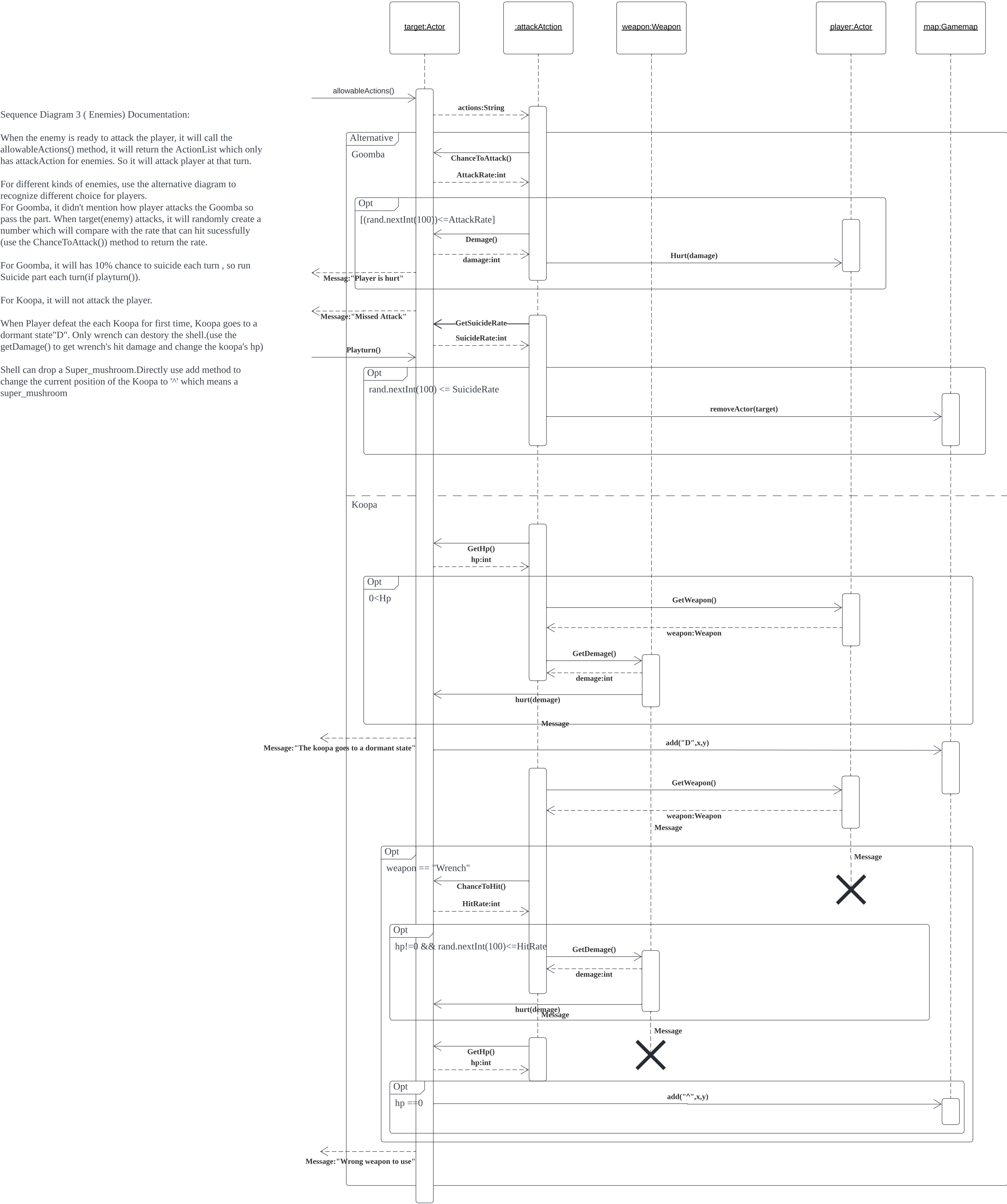
For different kinds of enemies, use the alternative diagram to recognize different choice for players.
For Goomba, it didn't mention how player attacks the Goomba so pass the part. When target(enemy) attacks, it will randomly create a number which will compare with the rate that can hit sucessfully (use the ChanceToAttack() method to return the rate.

For Goomba, it will has 10% chance to suicide each turn , so run Suicide part each turn(if playturn()).

For Koopa, it will not attack the player.

When Player defeat the each Koopa for first time, Koopa goes to a dormant state"D". Only wrench can destory the shell.(use the getDamage() to get wrench's hit damage and change the koopa's hp)

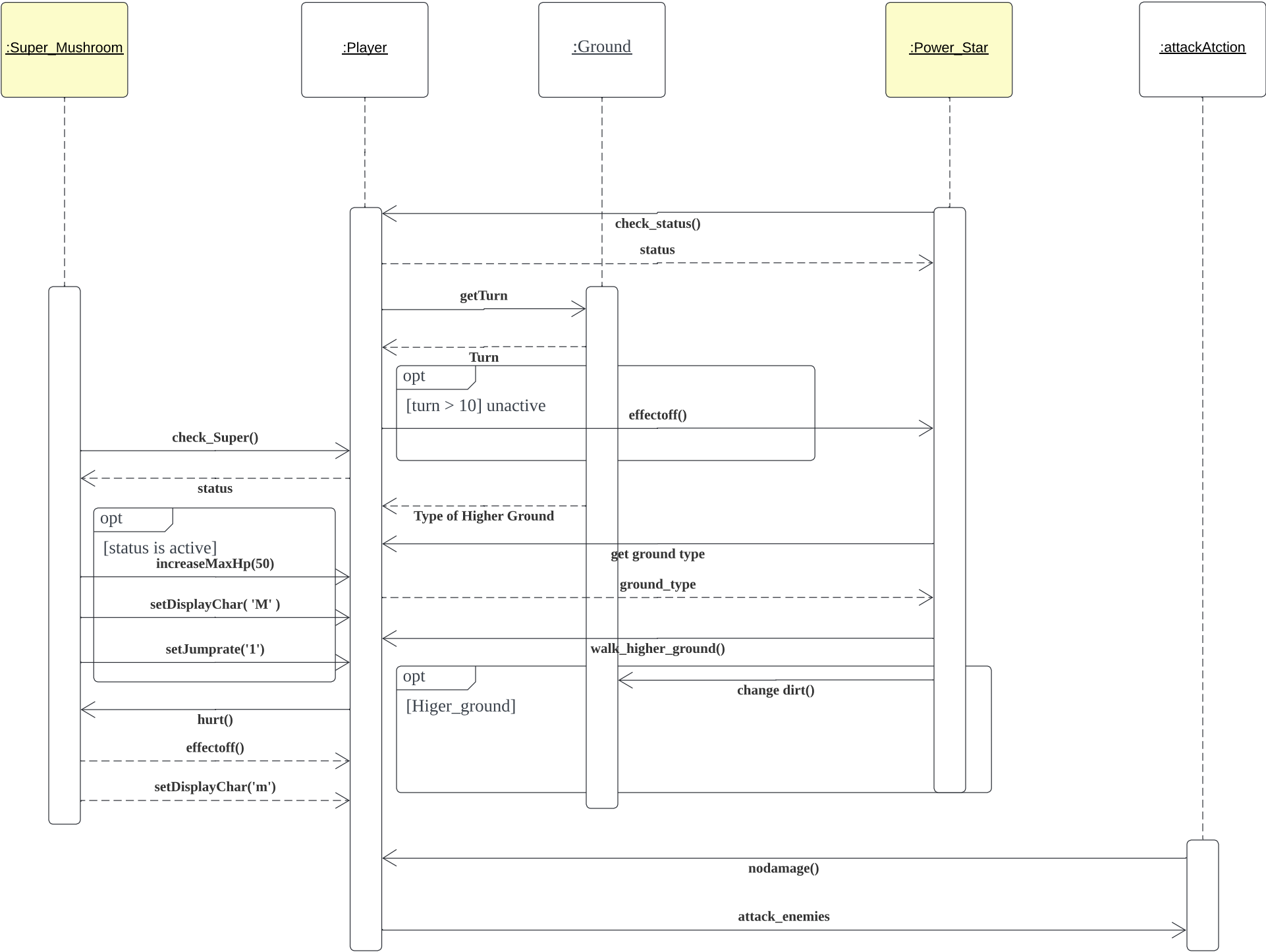
Shell can drop a Super_mushroom.Directly use add method to change the current position of the Koopa to '^' which means a super_mushroom



Sequence Diagram 4 (magical items) Documentation:

Two new classes have been created and extened from items for magical items (super mushroom & power star). super mushrron class is interacte with player class directly. The first thing is checking the status of super_mushrrom is active or not. If it is active, the super mushrron class will increase the MaxHP of player of 50 and set m to M, the jump rate will be 100%. Once player hurts it will return the effectoff from super mushrron and the higher letter M will be set back to 'm'.

The new class power star will be interact with 3 exiting class (player ground and the attack action). If the status is on to TALL, the player will check the turn with ground to make sure it is less than 10 if more then the effect will off. Otherwise, the power star will walk the high ground and change it to dirt. And there will no damage to the player when it meet emnemies and it will attck enemies.



Sequence Diagram 5 (Trading) documentation:

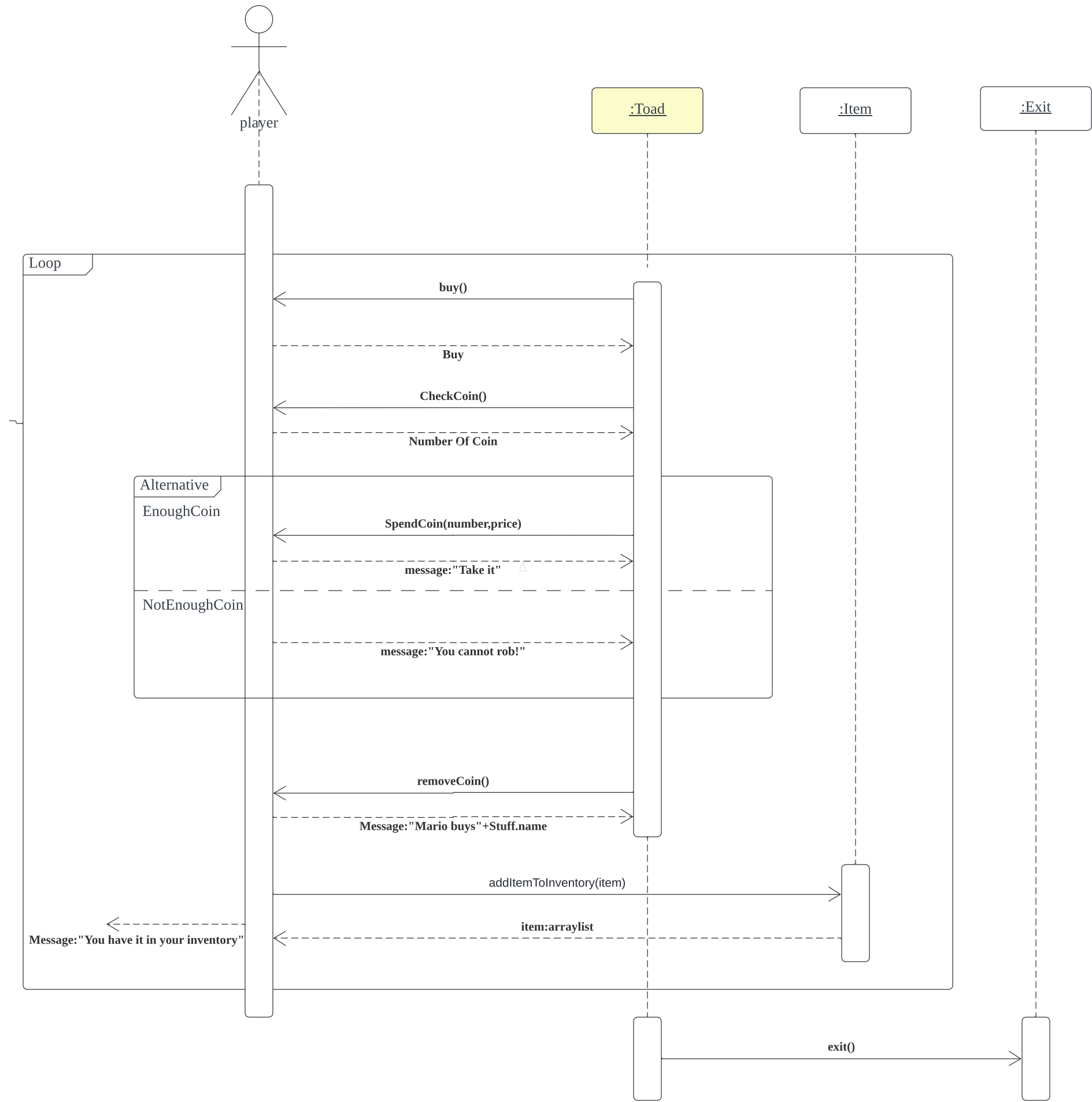
Toad is a new class which create a new object called toad. This object has no attributes but several methods: display the store menu, ask for the player's choice, change the player's coin and check if the player has enough money to buy.

The new class also will exit when player choose to not buy anything.

After checking the money, player will add the item which is bought at toad from item class and return it into the arraylist as an element.

add item into arrayList:
use ArrayList.put("Super_mushroom")

Change the player's coin:
Number of the coin - the price of the item



Sequence Diagram 7 (restgame) documentation:

The relationship between ResetManager and enemies classes (Goomba and Koopa) is clean then up , which means all enemies are killed .
And to the ground class , trees have 50% to change to dirt and all coins will be removed .
To do with player , the player will be heal up to the max. And all status of play will be clear to normal mode.

