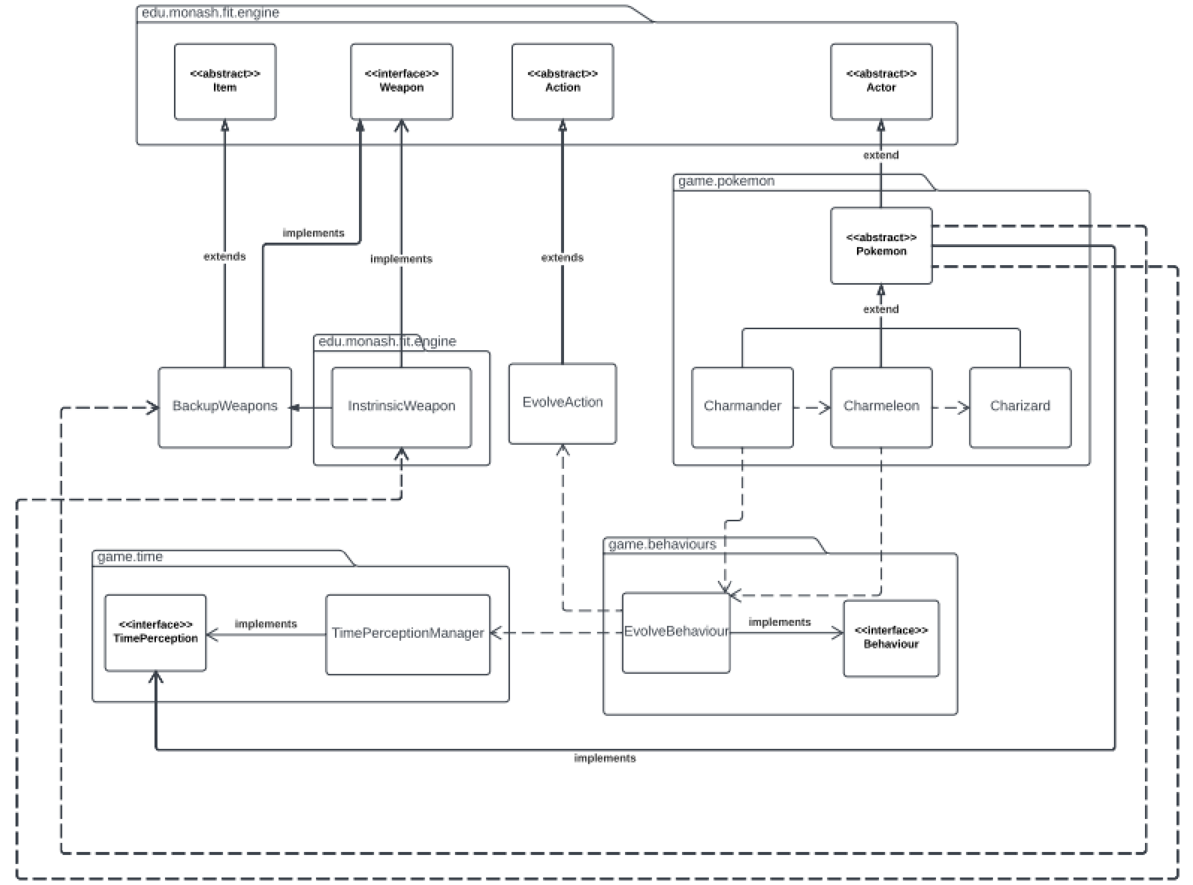
FIT2099 Assignment 3: New and Updated UMLS

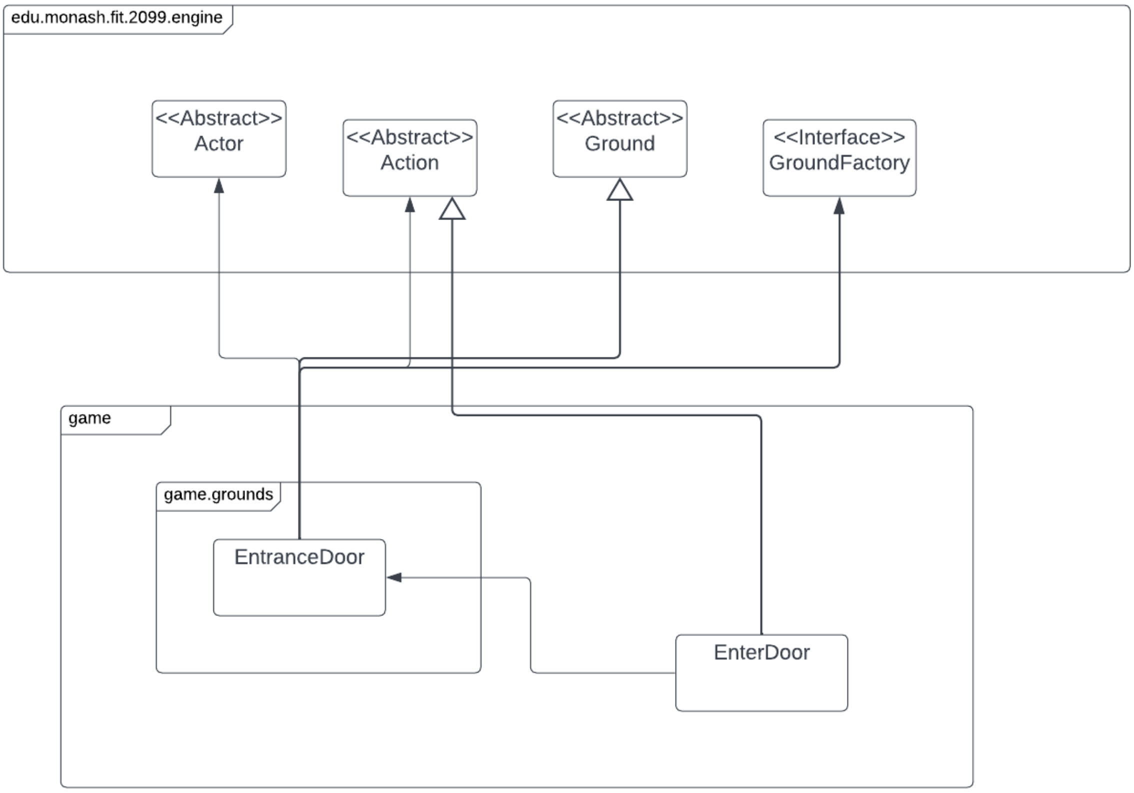
Requirement 1:



Rationale:

Evolution into our game has been added as a new behaviour which gets registered to the Pokémon which capable of evolution. These include Charmander and Charmeleon, with Charmeleon being a new class extending the abstract class Pokemon. Charizard (also a new class) does not register our new behaviour called “EvolveBehaviour” as it does not evolve. The EvolveAction class which extends Action has also been created which Charmander and Charmeleon call in order to evolve. This EvolveAction class helps obey SRP, as the individual Pokemon classes use this class rather than having separate evolution actions in their own class. EvolveAction also extends the abstract Action class, so we can make sure that our new evolution system does not affect the existing application, adhering to OCP. EvolveBehaviour also depends on TimePerceptionManager and more specifically the turn attribute which allows the Pokémon to evolve every 20 turns. All three Pokemon classes can also use their attacks by calling InstrinsicWeapon and BackupWeapon classes.

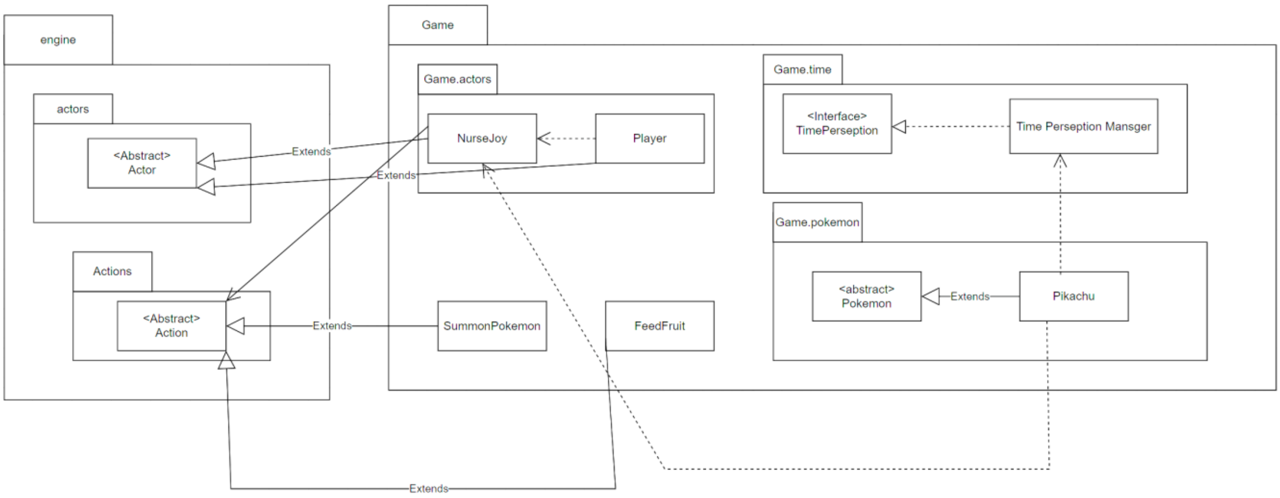
Requirement 2:



Rationale:

We add a new ground class ‘EntranceDoor’ which extends from the engine Ground abstract class. Within this class we implement the allowable action of ‘EnterDoor’ which is a new action that is implemented. The classes also utilise the GameMap and GroundFactory interface classes in order to transport the player and to get the position of the player when nearby the EntranceDoor. The EnterDoor Action is responsible for removing the player from the current map and adding them to the newly added map. We inherit the Ground class from the engine and use EntranceDoor as an extension of it, similarly we also inherit from the Action class for EnterDoor, thus ensuring the that the class is responsible for less functionality leading to fewer dependencies to other classes. This meets the Single Responsibility principle and leads to better code organization. We can always overwrite and extend the behaviours within these classe while keeping it closed for modification adhering to the Open Closed Principle. As it inherits from Ground and Action it is completely substitutable for the base class and avoids misusing inheritance, thus meeting the Liskov Substitution principle. We don't implement unnecessary methods or functions and ensure that higher level modules don't rely on lower-level ones thus satisfying the Interface Segregation Principle and the Dependency Inversion Principle, respectfully.

Requirement 3 (Creative Mode)::

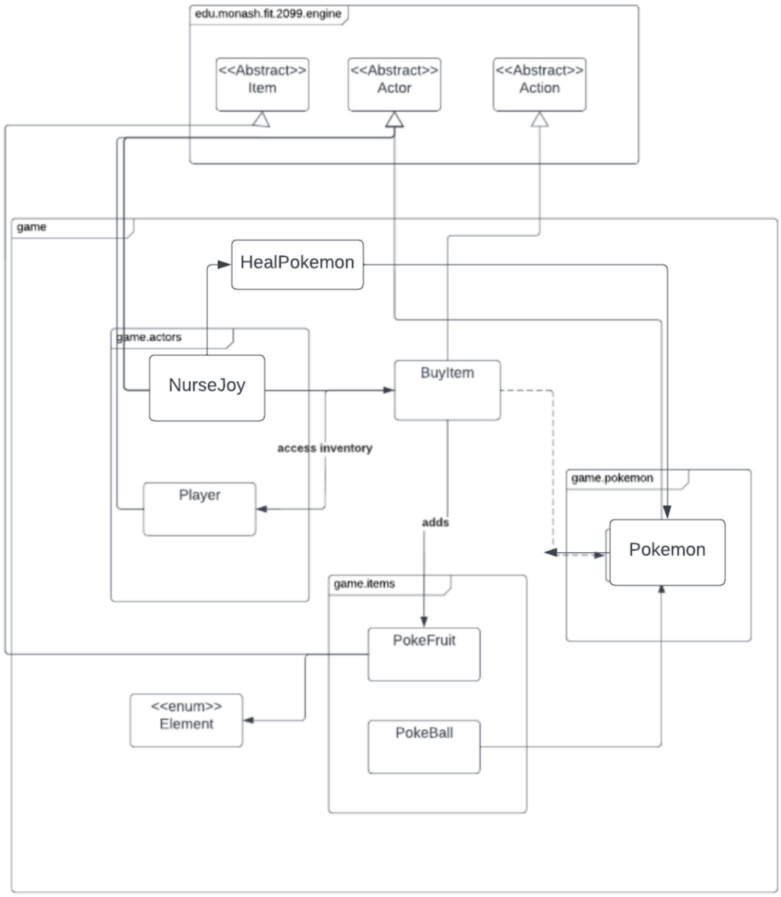


Rationale:

we added a “Pikachu” another pokemon that extends the abstract class pokemon in the game. Pokémon package. Pikachu depends on nurse joy as it can only be bought and can not be caught. Pikachu can be fed and summoned by the player as well. The pokemon isn't affected by the sunlight and yet heals during the night. The Pikachu also has a backup weapon that uses the special attack lightning similar to the other pokemon.

SOLID Principles and details are further explained in the creative mode table.

Requirement 4 (Creative Mode):

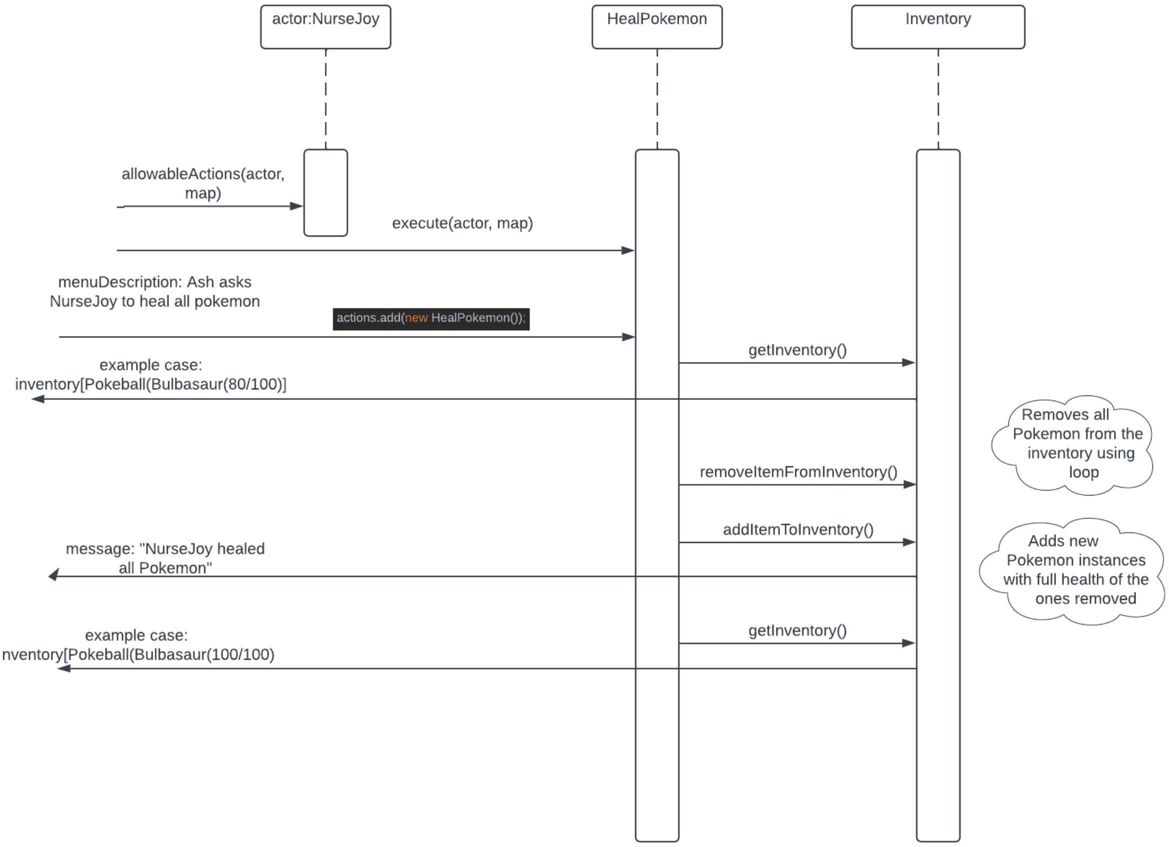


Rationale:

The NurseJoy has been updated with its new features as mentioned in our creative mode table. We added a HealPokemon action and extended it from action and implemented it as an action for NurseJoy. We further added new PokeBall items and new PokeFruit items.

SOLID Principles and details are further explained in the creative mode table.

Interaction Diagram for the above feature:



```
actions.add(new HealPokemon());
```

actor:NurseJoy
HealPokemon
Inventory
allowableActions(actor, map)
execute(actor, map)
menuDescription: Ash asks NurseJoy to heal all pokemon
getInventory()
example case: inventory[Pokeball(Bulbasaur(80/100)]
Removes all Pokemon from the inventory using loop
removeItemFromInventory()
addItemToInventory()
Adds new Pokemon instances with full health of the ones removed
message: "NurseJoy healed all Pokemon"
getInventory()
example case: nventory[Pokeball(Bulbasaur(100/100)