



Project : a text-based adventure game

- MAITREHENRY NATHANAEL
- VESSELLA EMILIE
- DARTIGUES MATHIEU

08/12/2023

Organisation

In group :

- Search our game's story
- Think about it in Java : identify the different classes, methods...
- Create the UML on paper then on computer (Visual Paradigm)
- Write the report

Individually :

- Create project's structure, write classes, methods, tests...
-

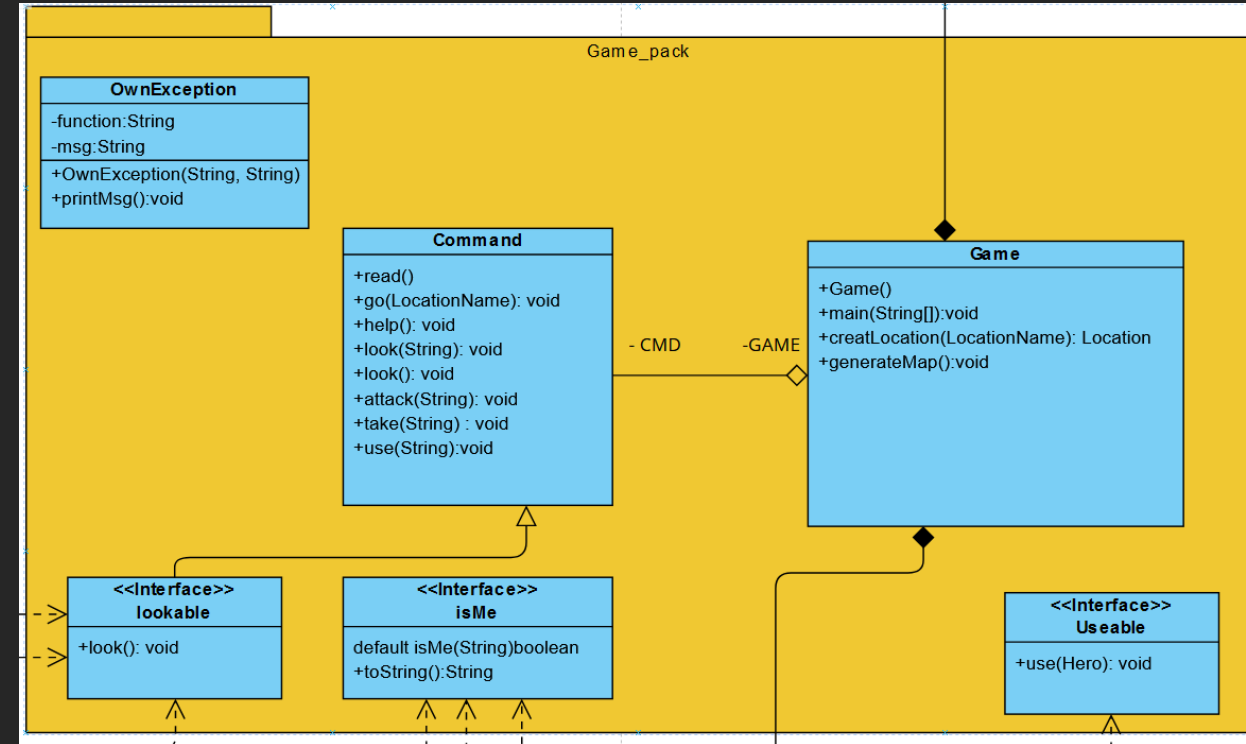
Game pack :

Game :

- Core of the game
- Generate map, initialize characters, handle commands

Command :

- Perform actions : aggregation to access Game functionality



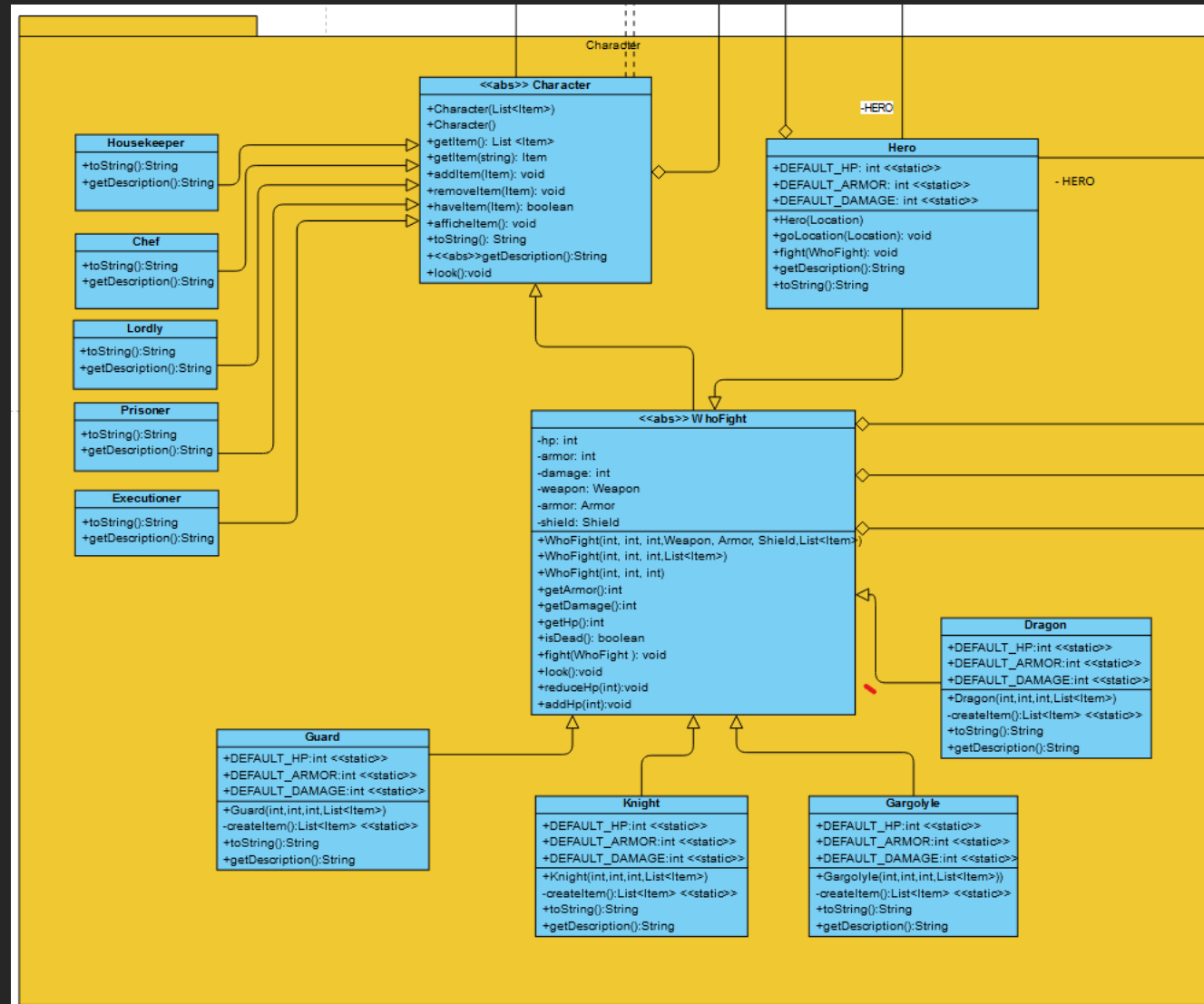
Characters

Character :

- Describe a character : add/remove/list items
- Abstract class : don't instance a Character, must override getDescription() method

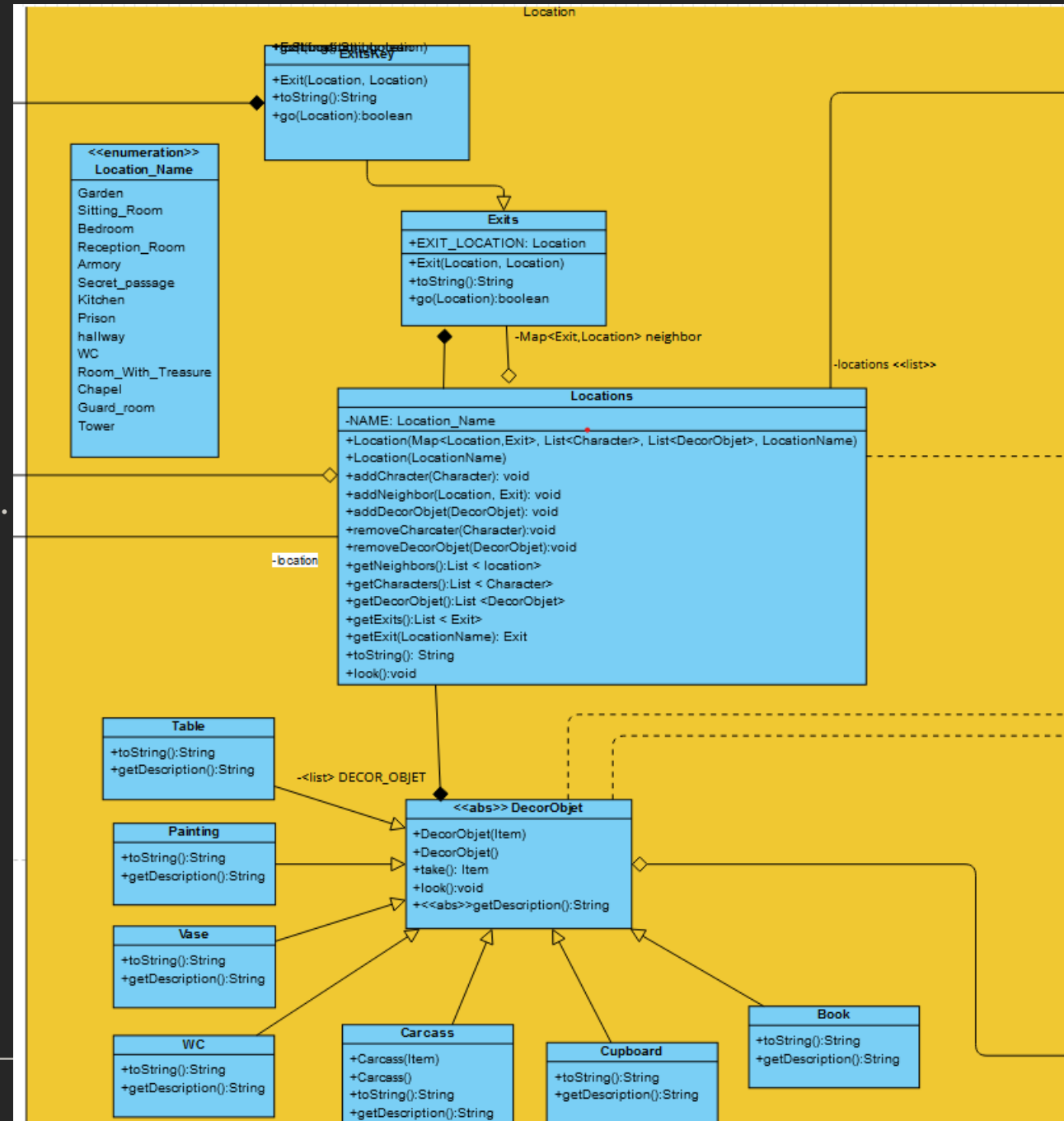
WhoFight :

- A character that can fight characters properties (heritage) + attacks methods
- Abstract class : same reasons



Locations

- Enumeration to save all possible locations.
- A DecorObject exist only in a location : composition
- Exits are locations

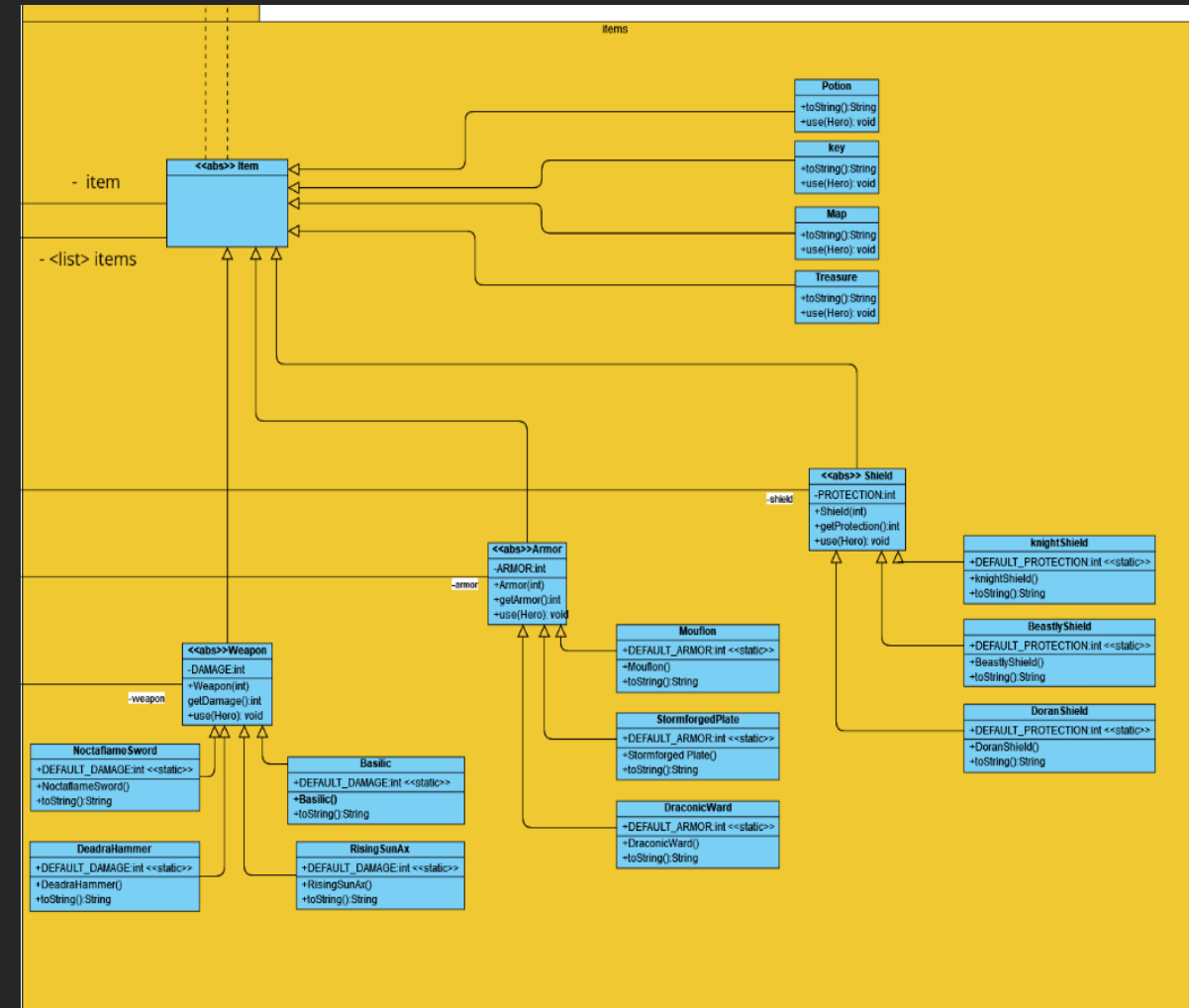


Items

- Implements useable's interface : each item has different functionality when used

Example : a potion add heal

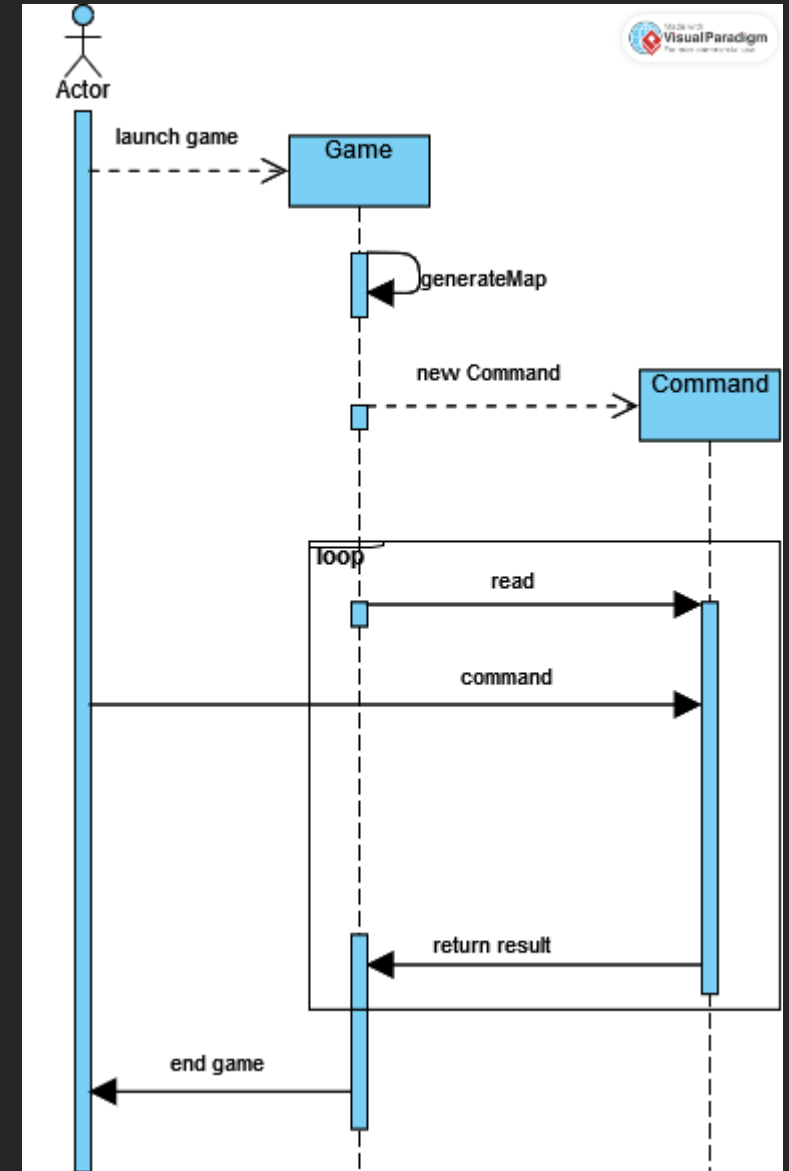
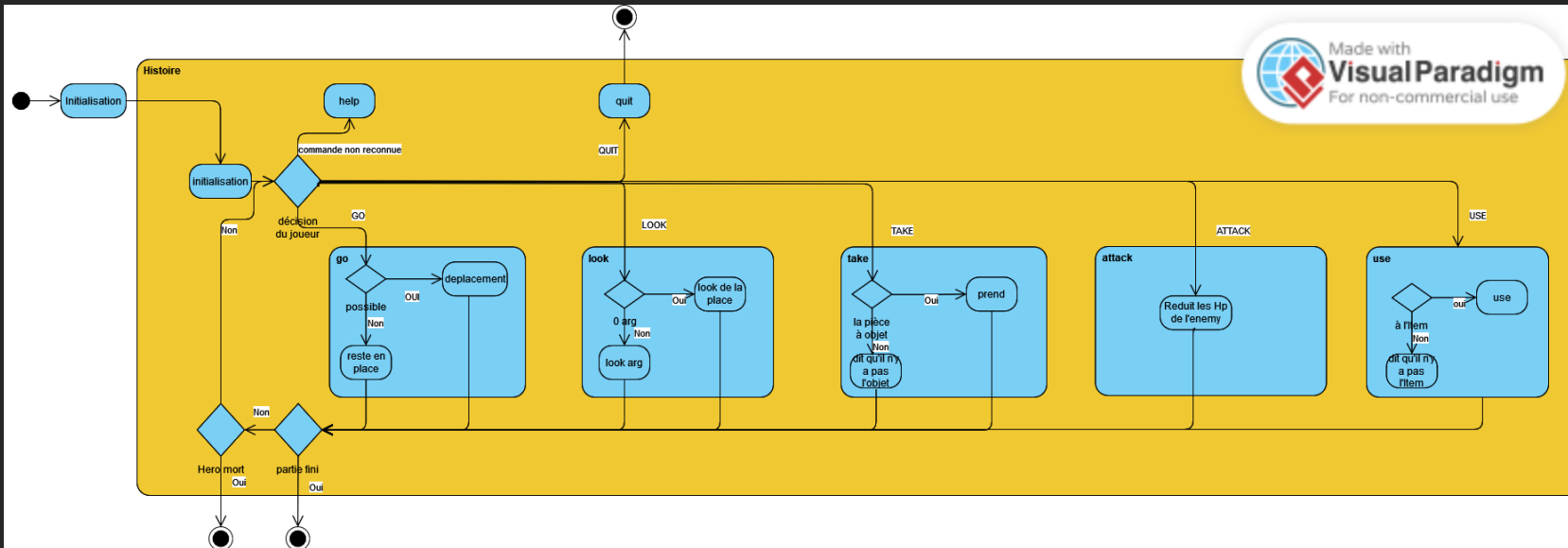
- Abstract class : describe property of a large type of items (Shield -> DoranShield, KnightShield...)



Other diagrams

Sequence diagram

State diagram



Conclusion

