

Bérénice Lemoine

Laboratoire d'Informatique de l'Université du Mans (LIUM), France
berenice.lemoine@univ-lemans.fr

Encadrant : Pierre Laforcade
Directeur : Sébastien George

Contexte de recherche

- ❖ Recherche en ingénierie des EIAH
- ❖ Génération d'activités de jeu de type Roguelite
- ❖ Entraînement aux connaissances déclaratives

Entraînement

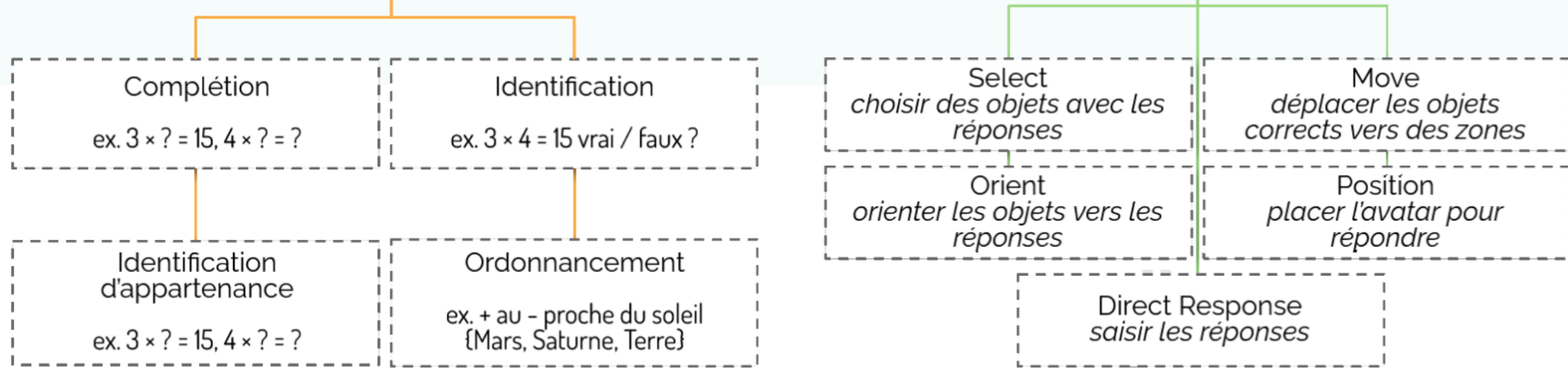
Besoin de répétition

Besoin d'activités variées

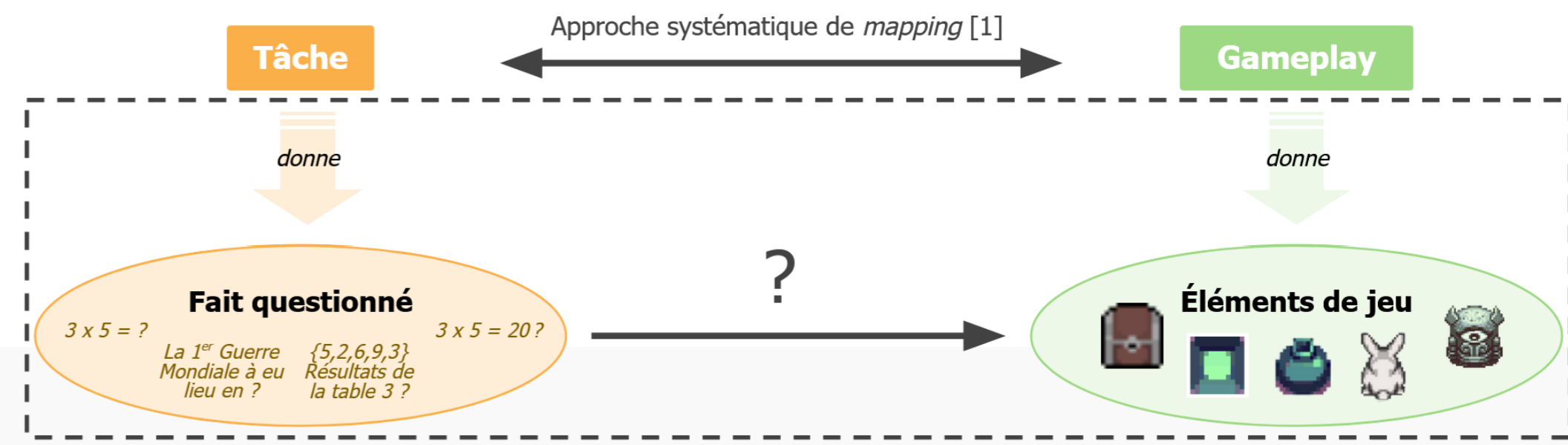
Approche systématique de mapping [1]

Type de Tâches

Catégories de Gameplays



[1] Lemoine et al., "An approach for mapping declarative knowledge training task types to gameplay categories", Computer supported education, 2024



[1] Lemoine et al., "An approach for mapping declarative knowledge training task types to gameplay categories", Computer supported education, 2024

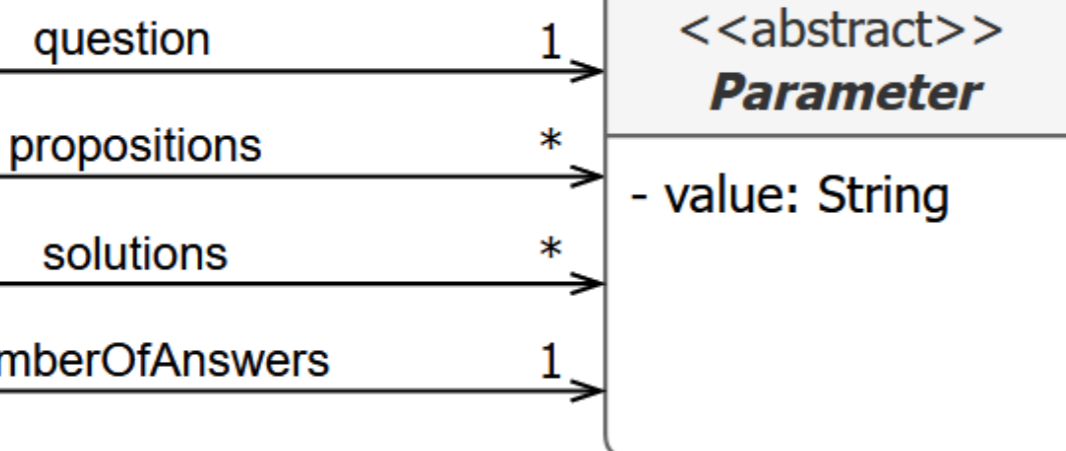
Objectif : Adresser le problème de l'alignement pédagogique-ludique, au niveau algorithmique, pour la génération d'activités de jeu d'entraînement aux connaissances déclaratives

Comment traduire les questions sur les faits en éléments de jeu variés ?

Objectif & Problème de recherche

QuestionedFact

- originalFact: Fact

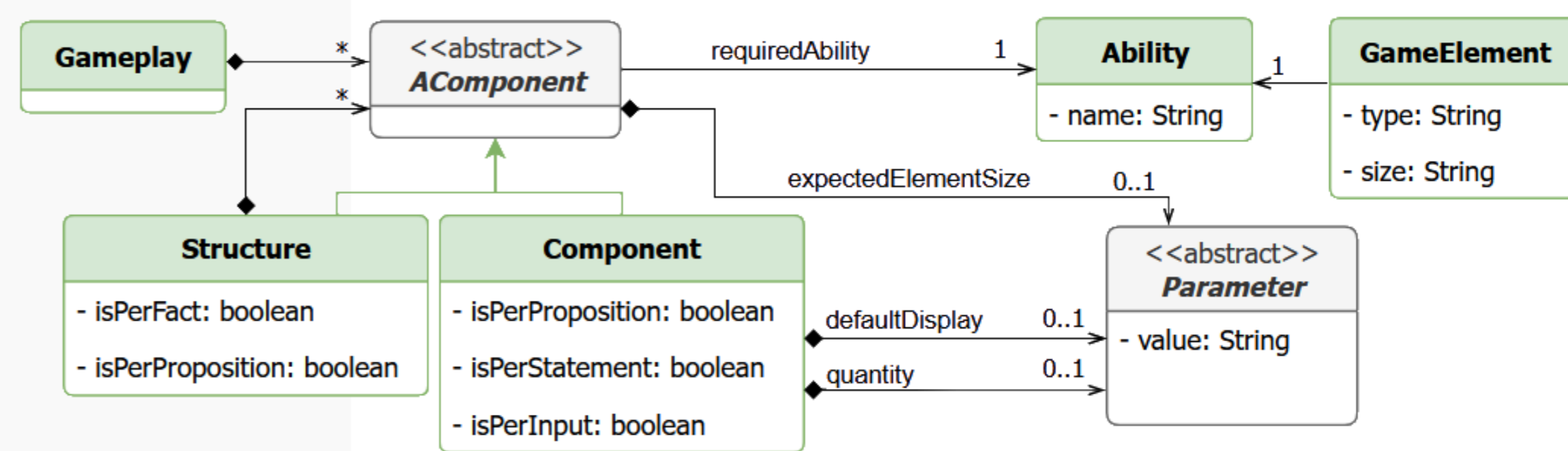


- ❖ Modélisation permettant une représentation sous forme **générique** des questions sur les faits = indépendant d'un domaine didactique
- ❖ Approche par paramétrisation = instantiation des paramètres utiles uniquement

Modélisation de questions sur les faits

Modélisation de gameplays

- ❖ Modélisation à partir d'abilities et d'intentions
- ❖ Ability = comportement des éléments de jeu (ex. un bloc peut être poussé ⇒ pushable)
- ❖ Intention = objectif d'entraînement (ex. affichage de propositions, de question, etc.)

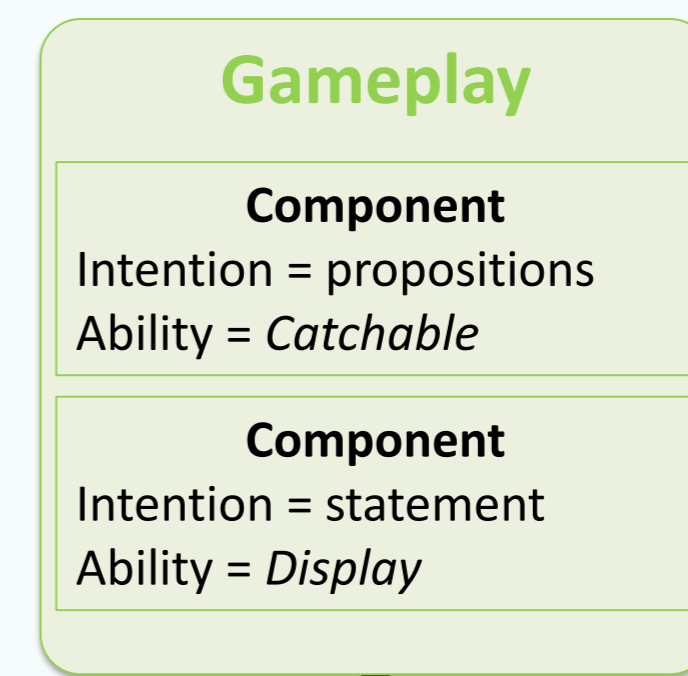
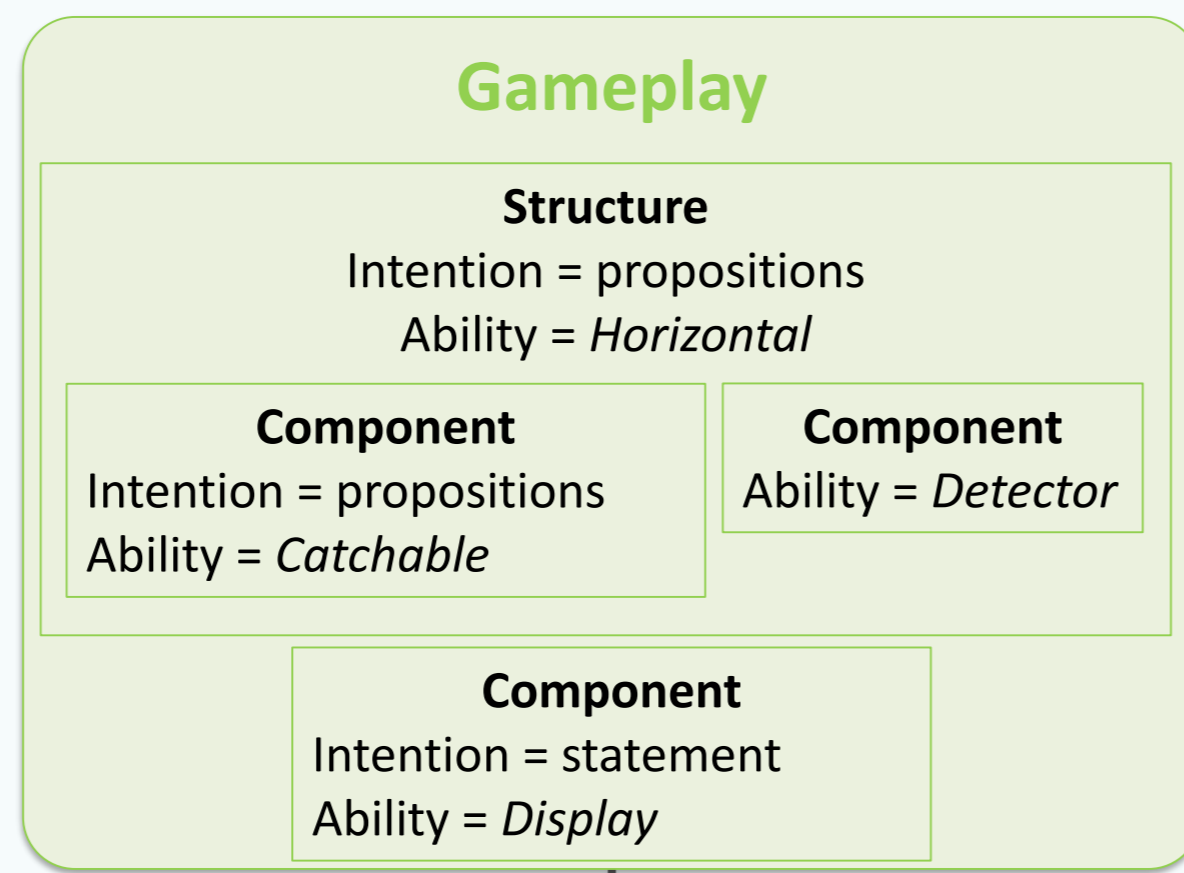
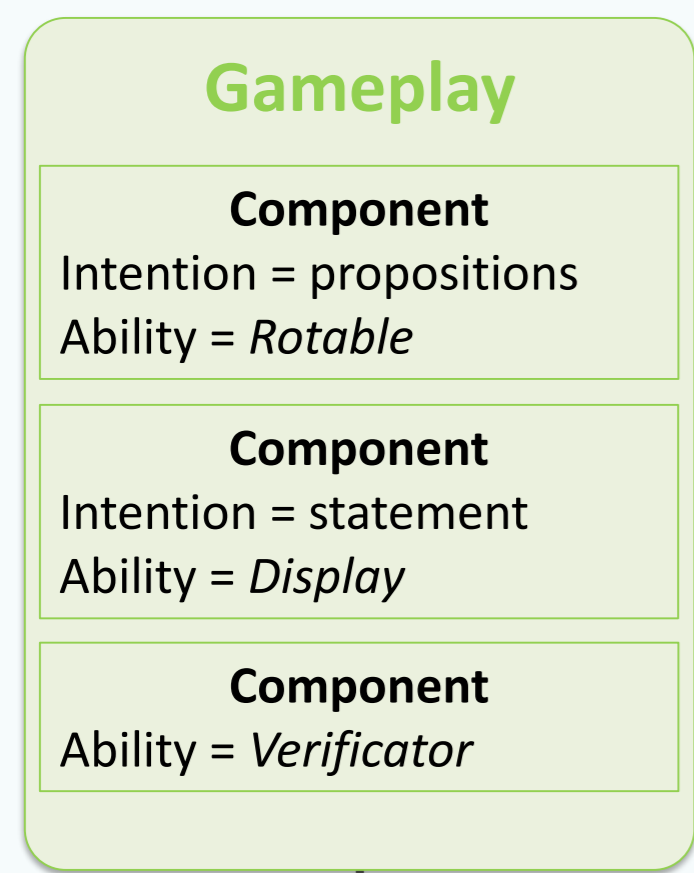
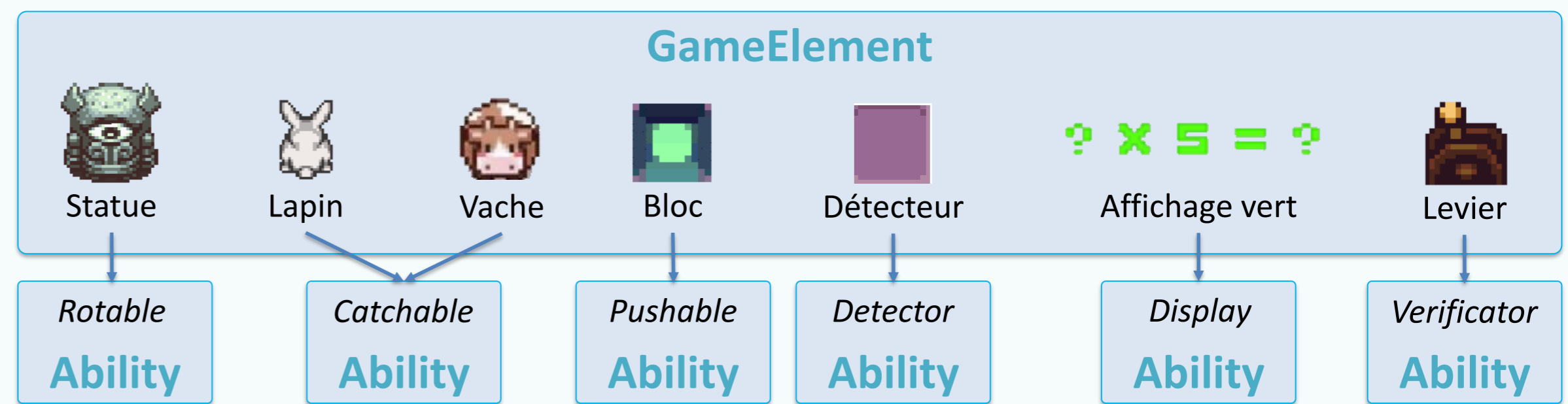


- ✓ Augmentation de la variété
- ✓ Indépendance des éléments de jeu (ajout d'éléments ≠ ajout de gameplays)
- ✓ Réduction du temps de conception (ajout de gameplays ≠ ajout d'éléments de jeu)

Algorithme illustré

QuestionedFact

Original fact : 2 x 10 = 20
Question = 2 x 10 = ?
Propositions = {2, 10, 20}
Solutions = {20}
Number of answers = 1



Génération

Génération

OU

