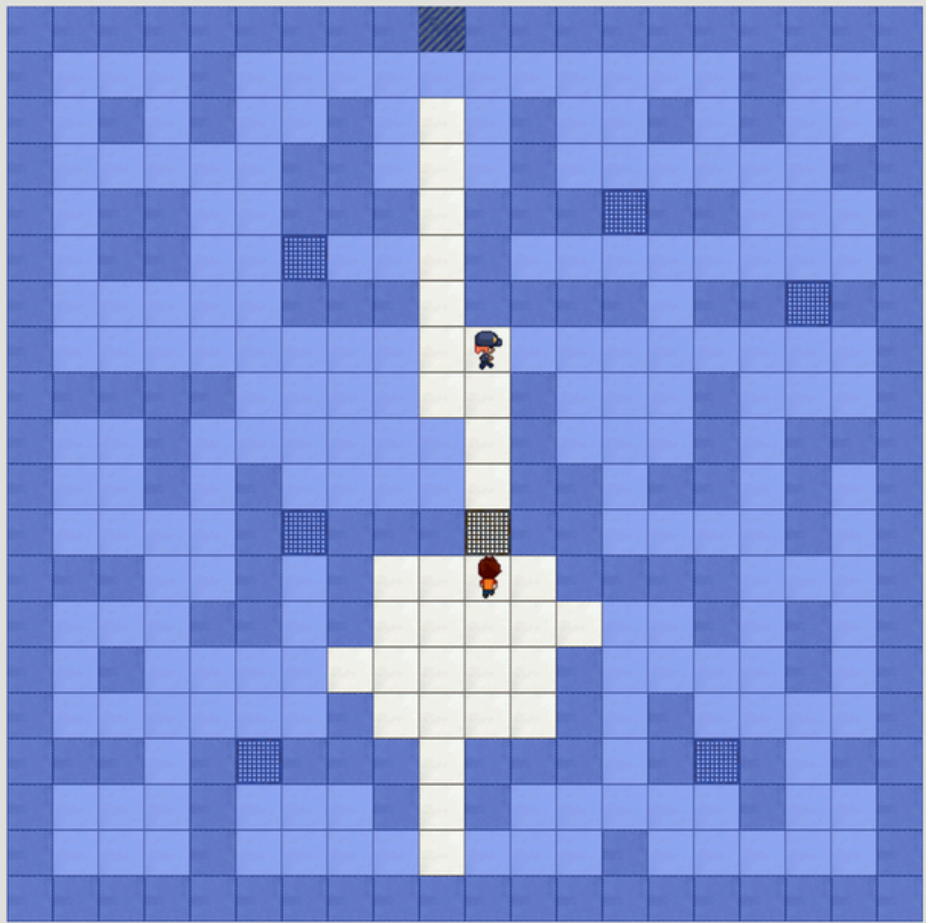


SIMULATION DE POURSUITE-ÉVASION  
COMPÉTITIVE ENTRE AGENTS INFORMATIQUES  
INTELLIGENTS

Tuteur: Guénaël Cabanes



Nombre d'itération: 4



# RAPPEL DU PROJET

---

- Simulation de poursuite-évasion
- Prisonnier **VS** Gardien
- But: S'évader par la sortie ou attraper le prisonnier



Veillez choisir un mode:

Mode interactif

Mode non interactif

Mode analyse

Parties sauvegardées

Quitter

# S O M M A I R E

---

**I. Objectifs de l'itération**

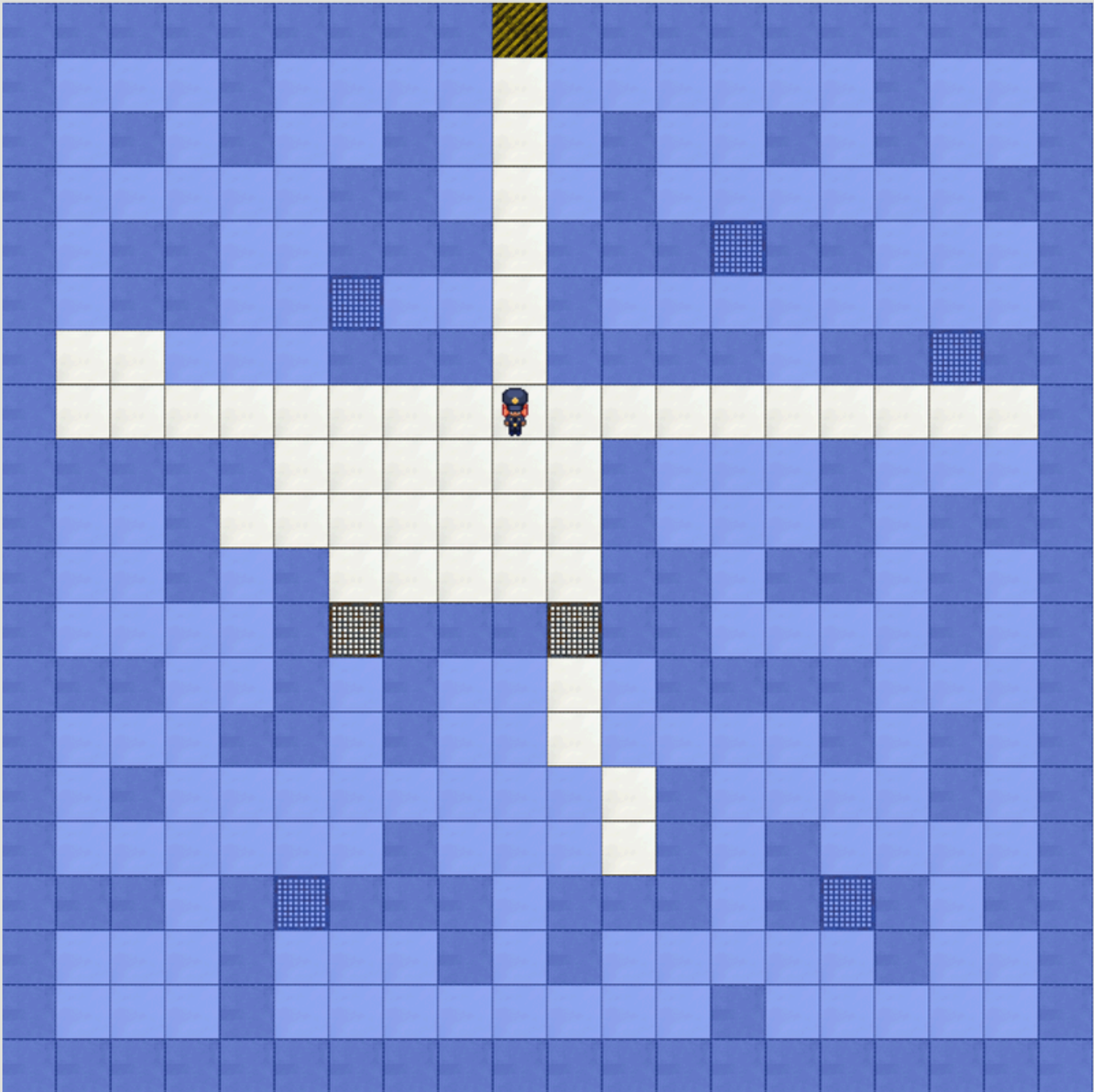
**II. Fonctionnalités développées**

**III. Évolution du diagramme de classe**

**IV. Difficultés rencontrées**

**VI. Lien entre l'étude préalable et les itérations**

**VII. Prochaine étape: itération n°5**



# I. OBJECTIFS DE L'ITÉRATION PRÉVUS

---

- Amélioration de l'affichage
- Implémentation de la librairie DJL
- Affinage de l'apprentissage du réseau de neurones
- Ajout de bonus/malus
- Optimisation du code

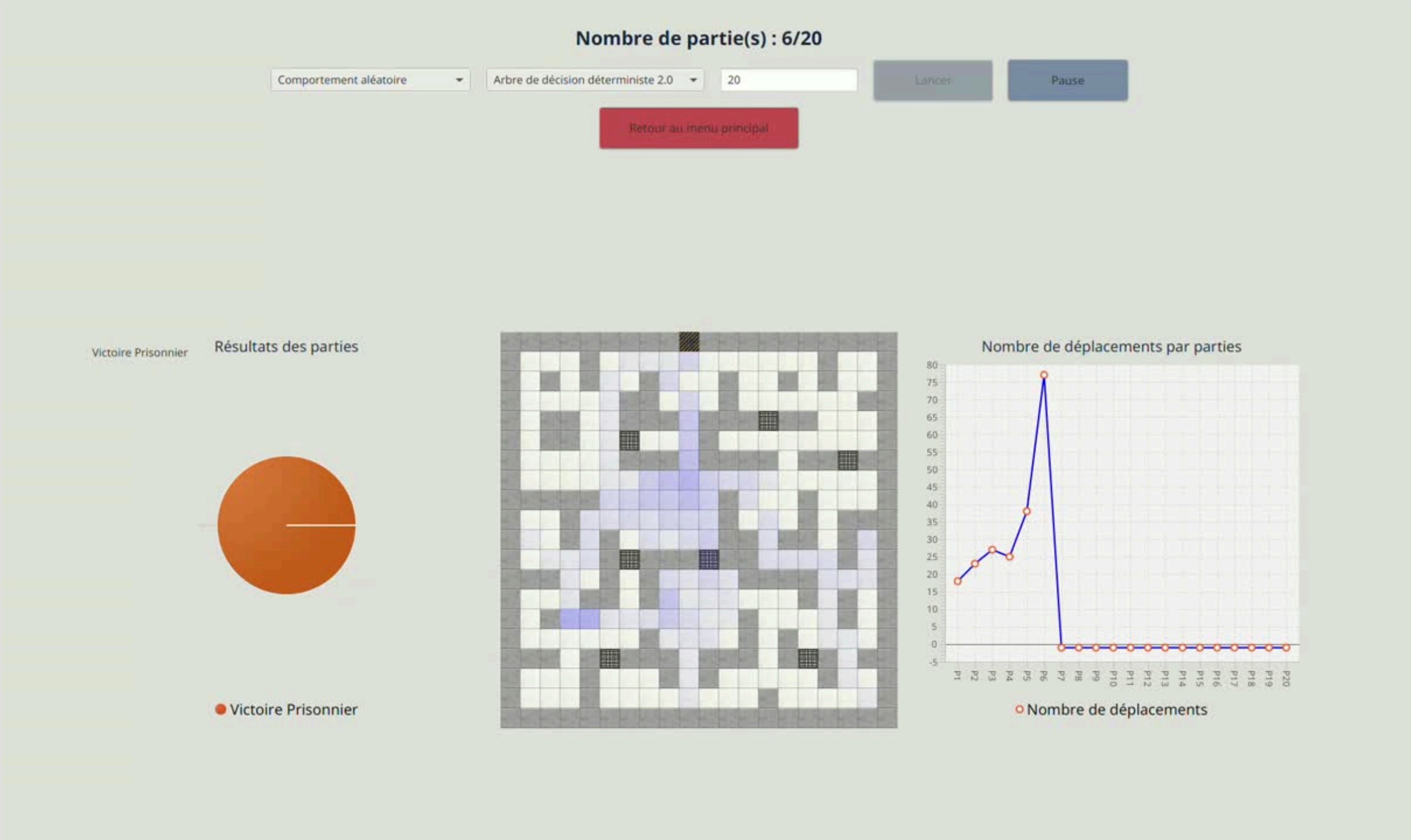
## II. FONCTIONNALITÉS DÉVELOPPÉES

---

- Implémentation de la librairie DJL (Matias & Luc)
- Ajout d'un système de sauvegarde des parties (Célie) 
- Ajout d'un système d'analyse (Maëlle) 
- Sprites directionnels (Célie)
- Amélioration de l'affichage (Célie)
- Affinage de l'apprentissage (Matias & Luc) 
- Ajout des informations sur les différents comportements (Célie) 

# II. FONCTIONNALITÉS DÉVELOPPÉES

Mode analyse:



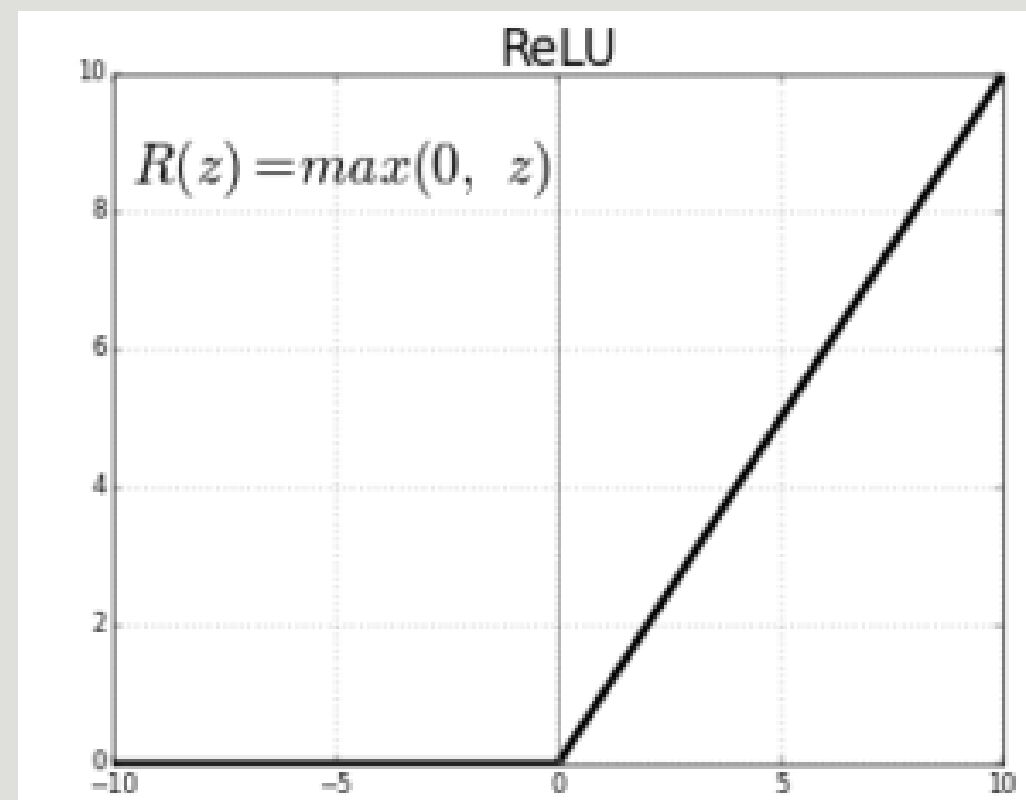


## II. FONCTIONNALITÉS DÉVELOPPÉES

---

Implémentation de la librairie DJL:

- Librairie Java pour le deep learning
- Outil complet et relativement rapide à prendre en main
- Permet d'avoir plus de contrôle sur les différents paramètres du réseau de neurones
- Problème de classification
- Fonction d'activation ReLu pour éviter l'effacement de gradient (sur les couches d'entrées)



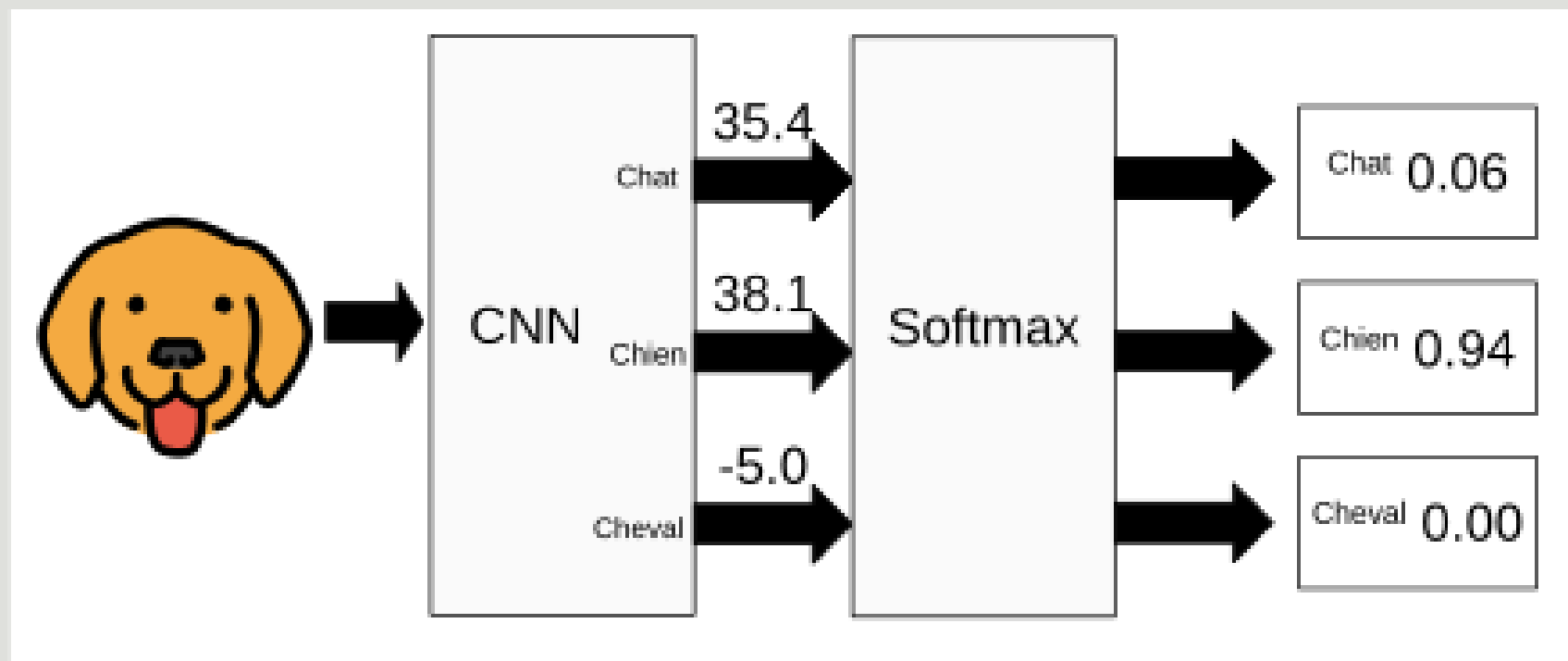
Fonction ReLu

$$\frac{d}{dx} \text{ReLU}(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases}$$

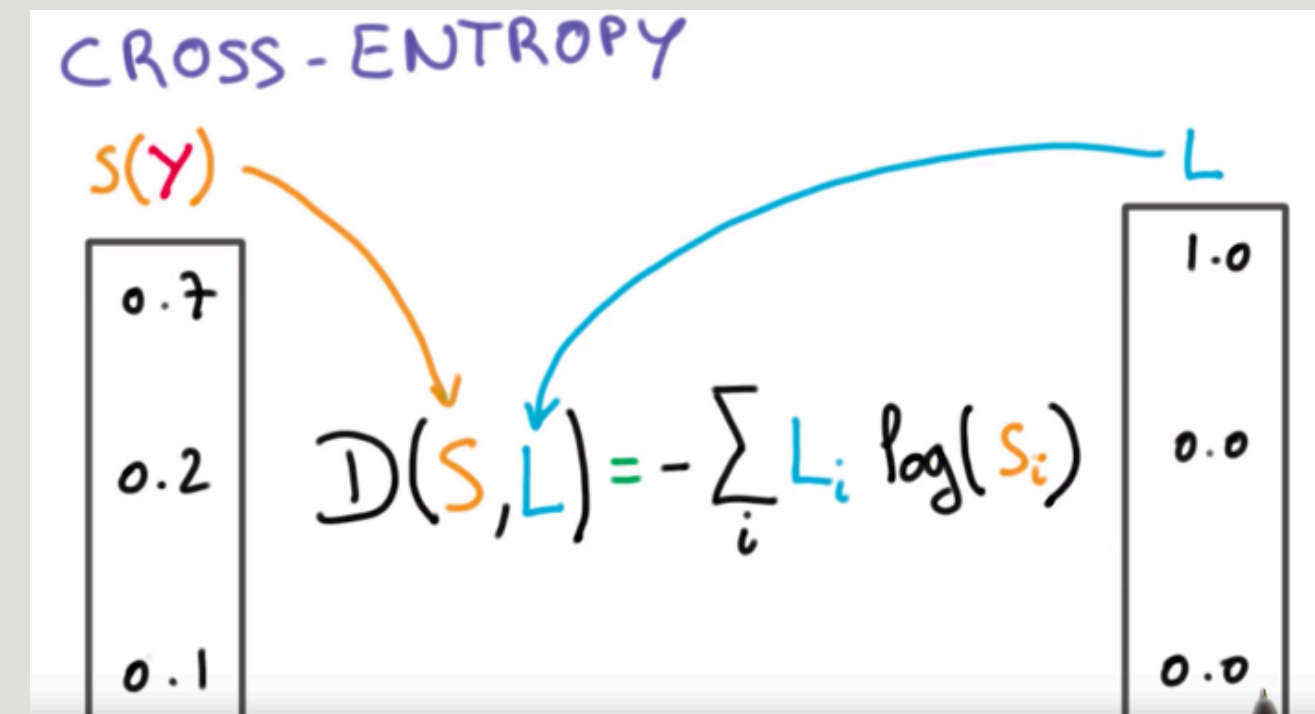
Formule de la dérivée partielle ReLu

## II. FONCTIONNALITÉS DÉVELOPPÉES

### Implémentation DJL



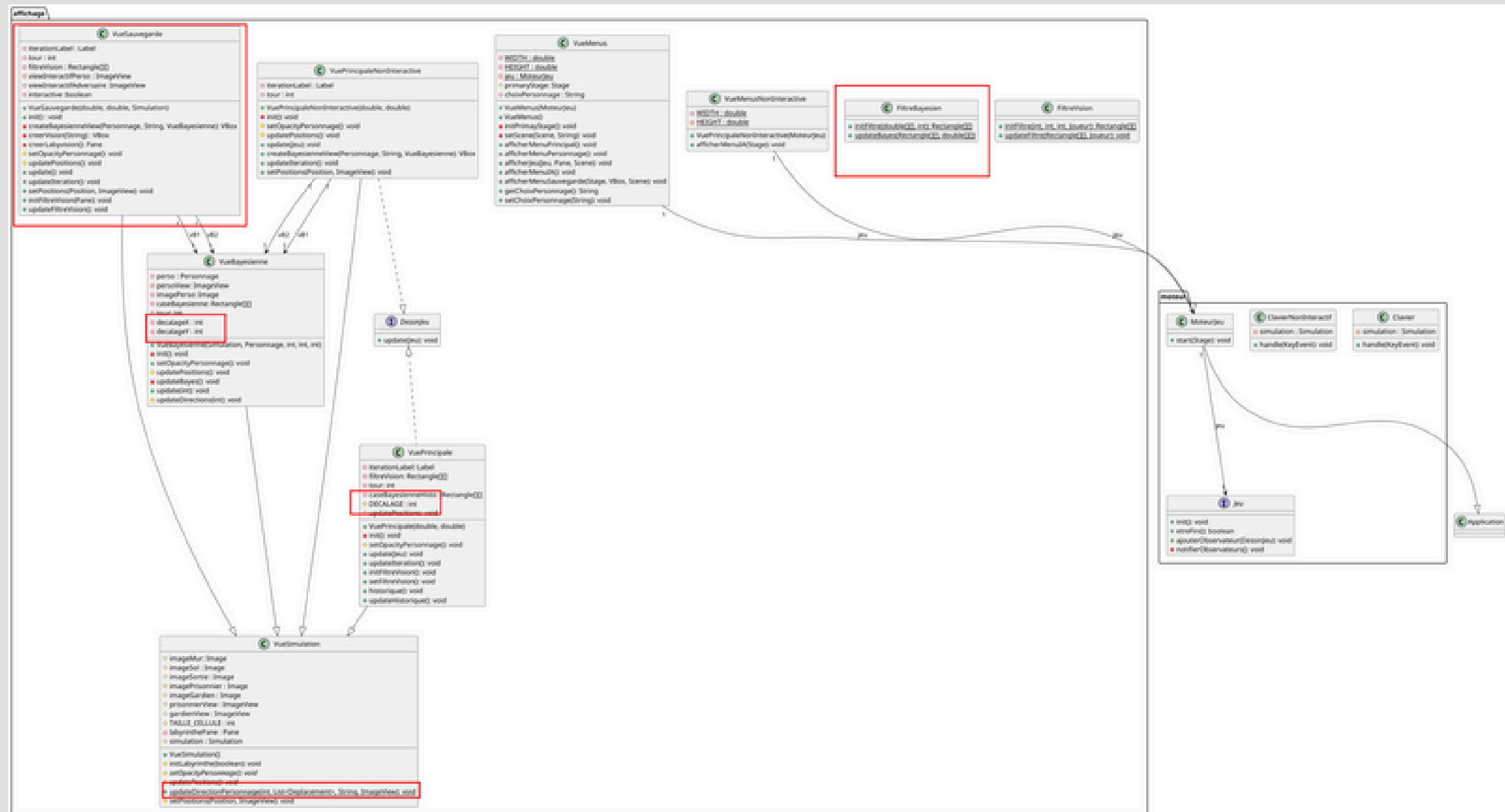
Représentation Softmax



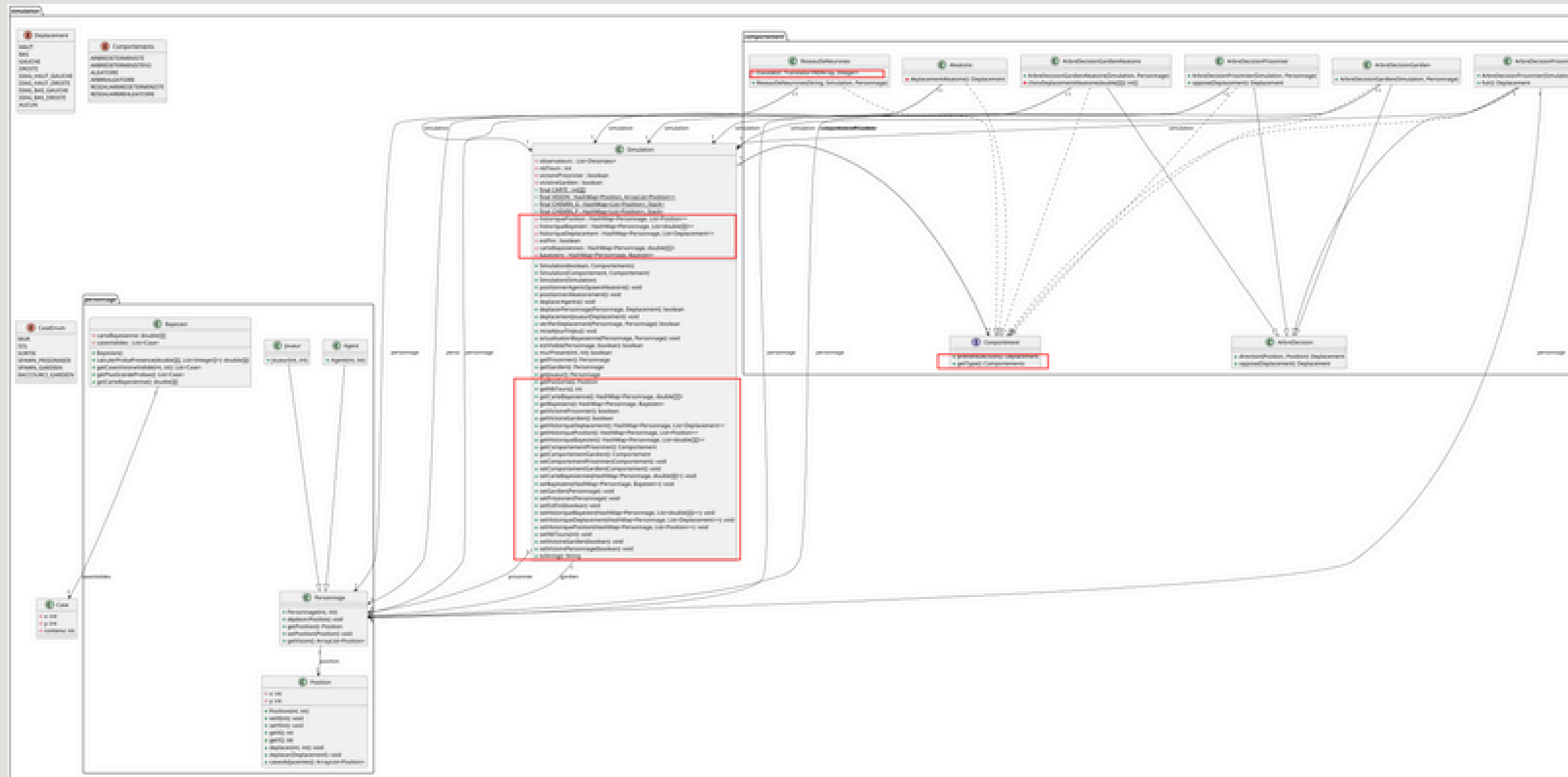
Représentation cross-entropy



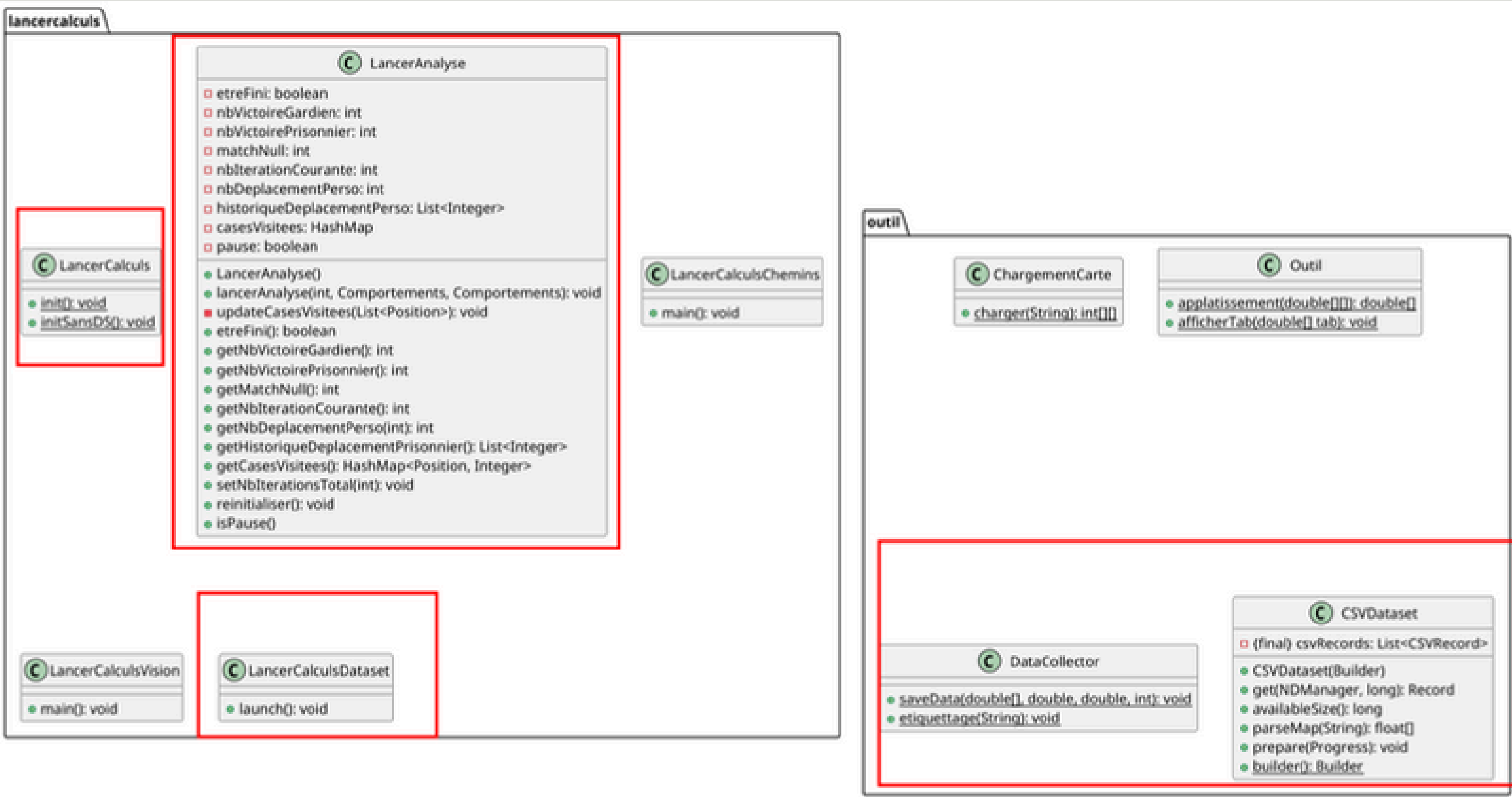
### III. ÉVOLUTION DU DIAGRAMME DE CLASSE



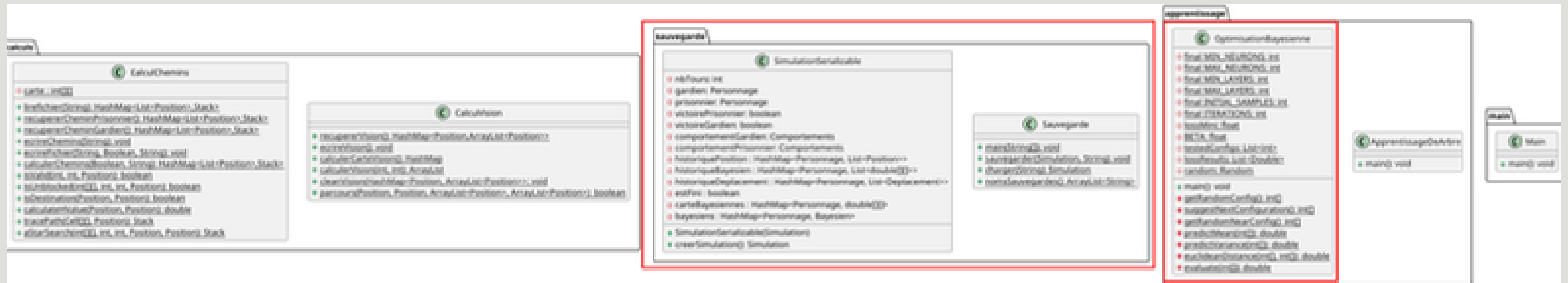
### III. ÉVOLUTION DU DIAGRAMME DE CLASSE



# III. ÉVOLUTION DU DIAGRAMME DE CLASSE



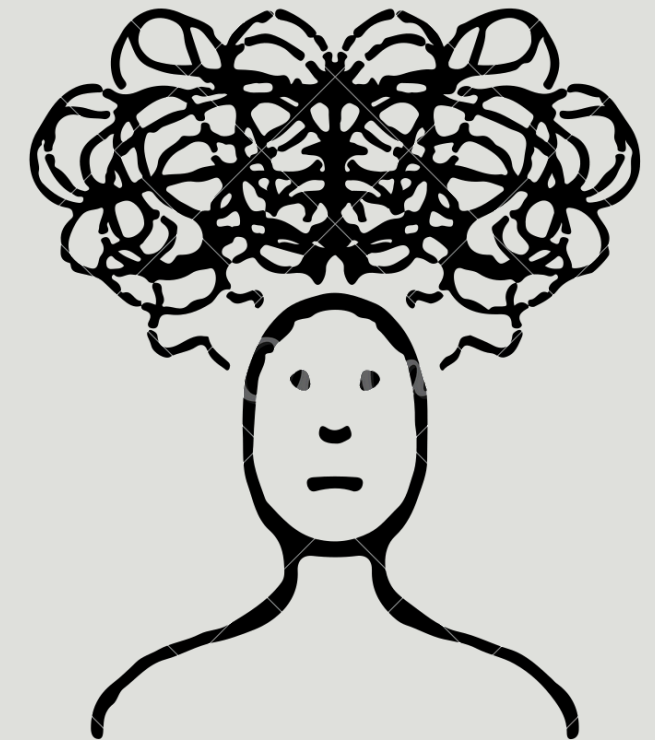
### III. ÉVOLUTION DU DIAGRAMME DE CLASSE



# V. DIFFICULTÉS RENCONTRÉES

---

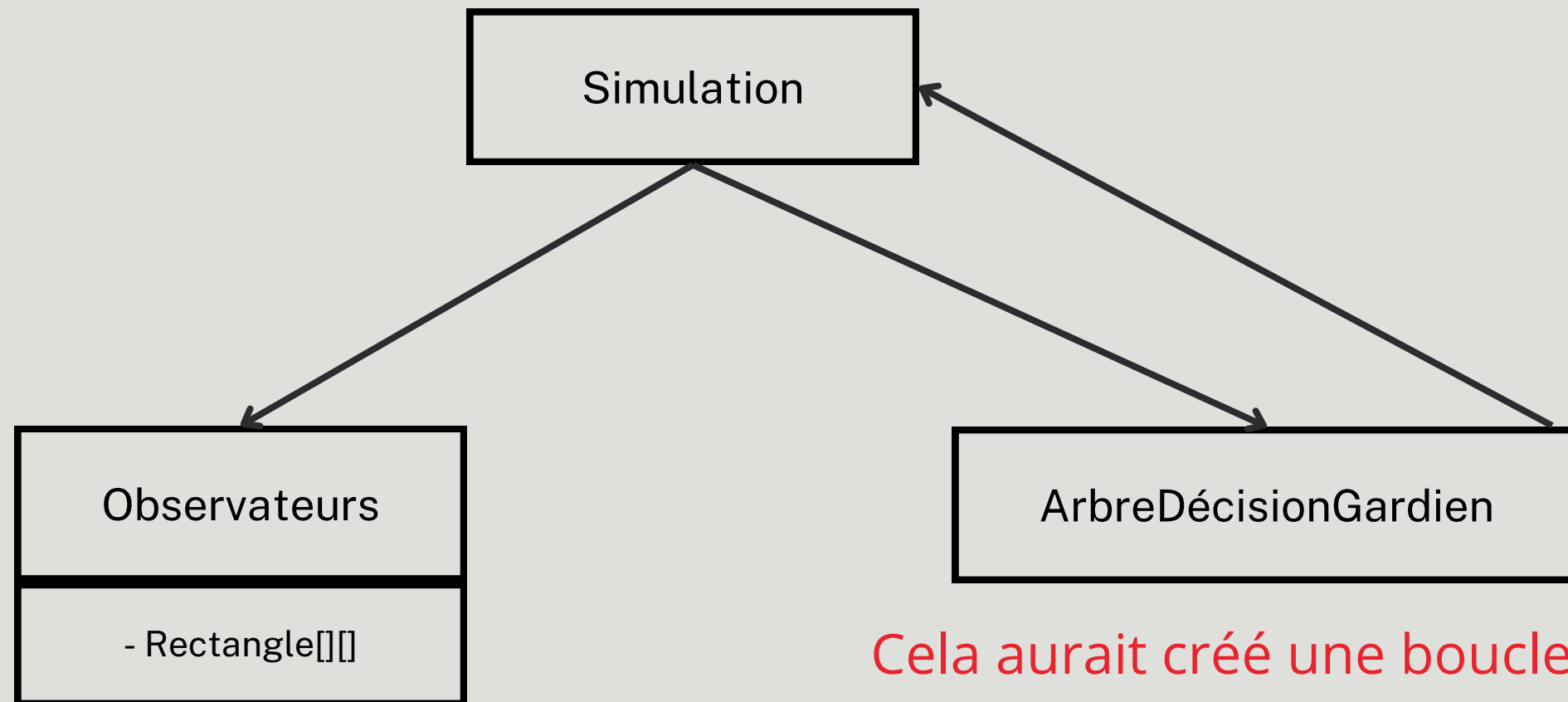
- Sérialisation de la simulation du mode de sauvegarde
- Ajout de temps réel de l’affichage du mode analyse
- Apprentissage



# V. DIFFICULTÉS RENCONTRÉES

---

Sérialisation problème :



Cela aurait créé une boucle infinie

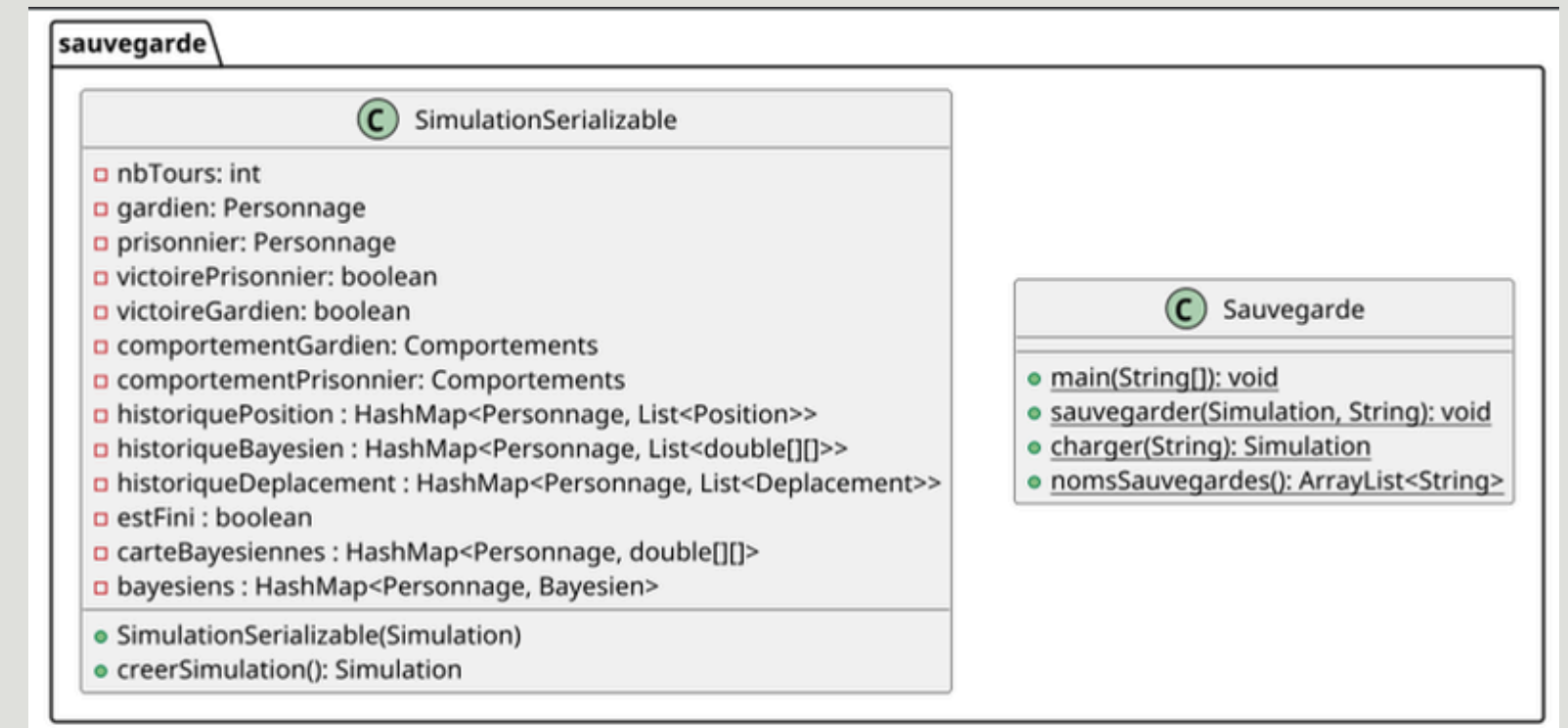
Les rectangles de JavaFx ne sont pas sérialisables.

# V. DIFFICULTÉS RENCONTRÉES

---

Sérialisation solution :

- La classe *Sauvegarde* sérialise et désérialise les parties.
- La classe *SimulationSerializable* est une copie de la simulation sans les attributs problématiques.





# V. DIFFICULTÉS RENCONTRÉES

---

## Apprentissage :

Pour 16646 apprentissage - 4160 validation

80 - 60 - 40 - 20

```
{
  "epoch": 70,
  "evaluations": {
    "validate_Accuracy": 0.26008752,
    "train_Accuracy": 0.26607993,
    "train_SoftmaxCrossEntropyLoss": 1.8765612,
    "validate_SoftmaxCrossEntropyLoss": 1.8778415,
    "train_loss": 1.8765612,
    "validate_loss": 1.8778415
  }
}
```

100 - 80 - 60 - 30

```
{
  "epoch": 100,
  "evaluations": {
    "validate_Accuracy": 0.26008752,
    "train_Accuracy": 0.2649346,
    "train_SoftmaxCrossEntropyLoss": 1.875659,
    "validate_SoftmaxCrossEntropyLoss": 1.8771691,
    "train_loss": 1.875659,
    "validate_loss": 1.8771691
  }
}
```

60 - 40 - 30

```
{
  "epoch": 70,
  "evaluations": {
    "validate_Accuracy": 0.26008752,
    "train_Accuracy": 0.26409066,
    "train_SoftmaxCrossEntropyLoss": 1.8768092,
    "validate_SoftmaxCrossEntropyLoss": 1.8770916,
    "train_loss": 1.8768092,
    "validate_loss": 1.8770916
  }
}
```

269 - 269 - 269 - 269

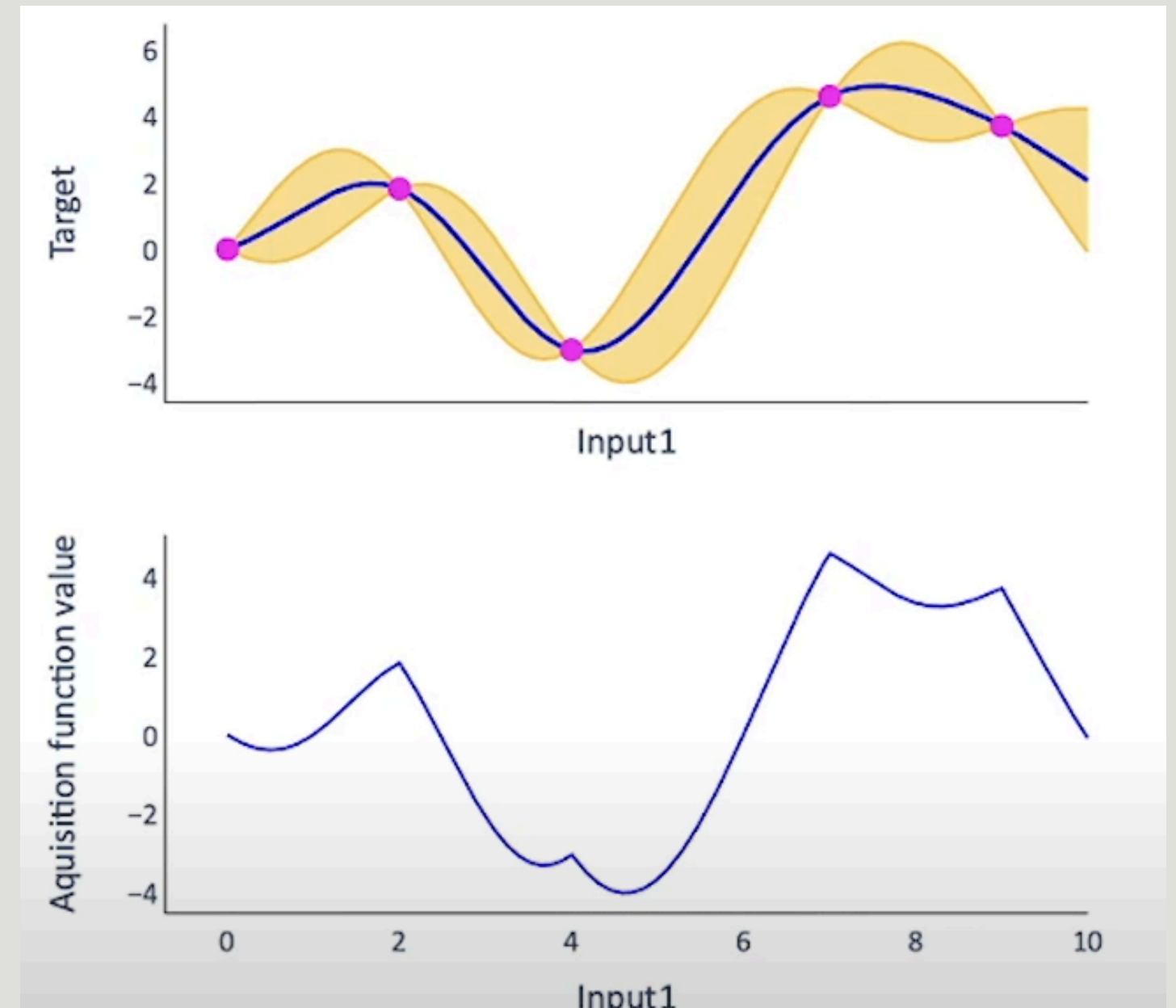
```
{
  "epoch": 150,
  "evaluations": {
    "validate_Accuracy": 0.24009721,
    "train_Accuracy": 0.258273,
    "train_SoftmaxCrossEntropyLoss": 1.8810058,
    "validate_SoftmaxCrossEntropyLoss": 1.8921794,
    "train_loss": 1.8810058,
    "validate_loss": 1.8921794
  }
}
```

# V. DIFFICULTÉS RENCONTRÉES

---

Apprentissage :

- Augmenter le nombre d'époch
- Changer les données
- Vérifier la fonction de loss cross entropie
- Vérifier la fonction d'activation pour leakyReLu
- Faire varier les hyperparamètres



*Optimisation bayésienne en LCB*

## VI. LIEN ENTRE L'ÉTUDE PRÉALABLE ET LES ITÉRATIONS

---

- Les fonctionnalités prévues de l'itération 1 à 4 dans l'étude préalable ont été **majoritairement terminées** mise à part les fonctionnalités liées au réseau de neurones.
- Nous avons pu ajouter 4 fonctionnalités supplémentaires durant ces 4 itérations:
  - Cases de raccourcis pour le gardien
  - Chargement de plusieurs cartes
  - Mode de sauvegarde des parties
  - Mode d'analyse d'un certain nombre de parties

## VII. PROCHAINE ÉTAPE: ITÉRATION N° 5

---

- Ajout d'un système de caméra de surveillance pour le gardien
- Modification de l'affichage de la vision en une version plus réaliste
- Ajout du temps réel dans le mode non interactif et dans l'historique du mode interactif
- Optimisation du code
- Affinage de l'apprentissage
- Ajout des informations sur les différents comportements

Passons à la démonstration...