



Manuel Utilisateur

Master 1 ANDROIDE : Arthur Esquerre-Pourtère, Jerome Arjonilla

2020

Robotique en essaim et apprentissage de comportements coopératifs

Préparation à l'utilisation

Une grande partie du projet n'a pas été codée par nos soins. En effet, nous utilisons le simulateur robotique "Roborobo3" créé par Nicolas Bredeche.

Afin de comprendre au mieux notre manuel utilisateur, il est primordial d'avoir lu et utilisé le simulateur robotique "Roborobo3". En effet, durant ce projet, nous avons implémenté de nouvelles fonctionnalités, mais une grande partie des fonctionnalités existent déjà.

Afin d'utiliser nos améliorations, il faut installer notre version de "Roborobo3" pour cela, il suffit de le télécharger en allant sur ce lien [Git](#).

Pour l'installation, il est recommandé d'utiliser Linux. En effet, le simulateur "Roborobo3" est mis à jour pour les utilisateurs de Linux et MacOS. Cependant, n'ayant pas accès à un appareil possédant MacOS, nous n'avons pas pu tester si notre version fonctionne pour celui-ci. Ainsi, il est conseillé d'utiliser Linux.

Par la suite plusieurs choix s'offrent à vous pour utiliser les fonctionnalités qui nous avons implémentées.

- Soit, en utilisant et modifiant le fichier "p_androide.properties" situé dans le document "config". En faisant cela, vous pourrez directement utiliser nos améliorations, et ajuster les paramètres en fonction de vos besoin.
- Soit, en rajoutant vos propres améliorations à la suite de nos améliorations. Et pour cela, il faut
 - Soit coder à la suite de notre code dans les fichiers
 - * "PandroideAgentObserver"
 - * "PandroideController"
 - * "PandroideSharedData"
 - * "PandroideWorldObserver"
 - Soit, en clonant notre projet, et pour cela, il suffit de suivre les instructions disponibles dans la version de base du "_README.txt". Ainsi vous hériterez de nos fonctionnalités.

Si vous choisissez de rajouter vos propres améliorations en rajoutant du code, il est important de savoir maîtriser le C++. En effet "Roborobo3" est codé dans ce langage.

Amélioration

En plus des fonctionnalités de base disponibles permettant de changer les paramètres en fonction de nos besoins (le nombre d'agents présents sur la carte, le temps de réapparition de la nourriture, ...), nous avons rajouté les fonctionnalités suivantes :

- Recréation de l'expérience de base
- Ajout de l'expérience contrôle 1
- Ajout de l'expérience contrôle 2
- Augmentation de la difficulté de l'environnement
- Variable permettant de fixer la quantité de nourriture prise par les agents.

Toutes les nouvelles fonctionnalités sont facilement accessibles et réglables. Pour cela, il suffit d'aller dans le fichier "p_androide.properties" situé dans le document "config", ou en allant dans votre fichier ".properties" héritant du projet "p_androide".

```
gReservoirType = 0          # 0: one per agent (default); 1: global reservoir, only if gSelectionMethod=0
gSelectionMethod = 0        # optional: (default is 0) ; 0: random ; 1: first; 2: best; 3: fitprop, 4: population of clones
gNbOfArena = 20            # Number of arena for the population of clones
gEnergyRequestFixedValue = 0.7 # value must be between 0 and 1, ignored if gEnergyRequestOutput is true,
gRegrowTimeIncreasingStep = 10 # Value of the increase at each generation for the variable RegrowTime
gFinalRegrowTime = 400     # Final value for the variable RegrowTime, ignored if lower than gPhysicalObjectDefaultRegrowTimeMax
```

Figure 1: Variable

Recréation de l'expérience de base

Pour reproduire l'expérience de base, il faut fixer la valeur de la variable "gReservoirType" à 0.

Ainsi les agents stockeront les génomes reçus dans un réservoir individuel. Et à la prochaine génération, chaque agent qui doit revivre sélectionnera au hasard un génome dans son propre réservoir.

En revanche, cette option est disponible seulement si "gSelectionMethod" égale à 0, car la sélection du génome est aléatoire.

Plus de précisions sur l'expérience de base sont disponibles sur notre rapport.

Ajout de l'expérience contrôle 1

Pour reproduire l'expérience contrôle 1, il faut fixer la valeur de la variable "gReservoirType" à 1 (au lieu de 0).

En faisant cela, nous allons remplacer le réservoir individuel de nos agents par un réservoir global. Et la sélection du génome à la prochaine génération se fera dans ce réservoir.

En revanche, cette option est disponible seulement si "gSelectionMethod" égale à 0, c'est à dire, si la sélection du génome est aléatoire.

Plus de précisions sur l'expérience contrôle 1 sont disponibles sur notre rapport.

Ajout de l'expérience contrôle 2

Pour utiliser l'expérience contrôle 2, il faut modifier la valeur de "gSelectionMethod" et fixer la valeur à 4. En faisant cela, on indique au simulateur qu'on souhaite créer des différentes arènes contenant des populations de clones, et que la sélection s'effectuera sur la meilleure arène (i.e. celle qui a le plus d'agents encore en vie à la fin de la génération).

De plus, l'expérience contrôle 2, il faut indiquer le nombre d'arènes souhaité, et pour cela, il faut modifier la valeur de la variable "gNbOfArena".

Plus de précision sur l'expérience contrôle 2 sont disponibles sur notre rapport

Augmentation de la difficulté de l'environnement

Il est possible d'augmenter la difficulté de l'environnement (le temps de réapparition maximum de la nourriture) au fur et à mesure des générations. Et pour cela, il faut indiquer la valeur souhaitée finale avec la variable "gFinalRegrowTime".

Ainsi, le niveau de difficulté de l'environnement commencera par la valeur de la variable "gPhysicalObjectDefaultRegrowTimeMax" (variable de base de "Roboro3") à la valeur de la variable "gFinalRegrowTime". L'augmentation se fera progressivement à chaque génération et ceci à partir de la 20 000 itération.

La variable "gRegrowTimeIncreasingStep" permet de gérer le pas d'augmentation de la difficulté.

Grâce à cela, les agents auront un environnement changeant, et ils devront apprendre à s'y adapter.

Variable permettant de fixer la quantité de nourriture prise par les agents

Dans la version de base de "Roboro3", il n'est pas possible de choisir facilement la quantité de nourriture prise par chaque agent. Nous avons donc implémenté une fonctionnalité permettant de choisir cette quantité. Et pour cela, il suffit de modifier la valeur "gEnergyRequestFixedValue".

La valeur de cette variable correspond au ratio entre la quantité de nourriture que l'on souhaite prendre et la quantité maximale de nourriture pouvant être prise et doit donc être comprise en 0 et 1, avec 0 qui signifie qu'on ne prend aucune nourriture, et 1 qui signifie qu'on prend toute la nourriture disponible. Cette variable fixera la quantité prise pour tous les agents.

De plus, pour activer cette option, il faut que la variable "gEnergyRequestOutput" soit désactivée (c'est à dire égale à false). En effet, cette variable permet aux agents de choisir de façon dynamique leur quantité.