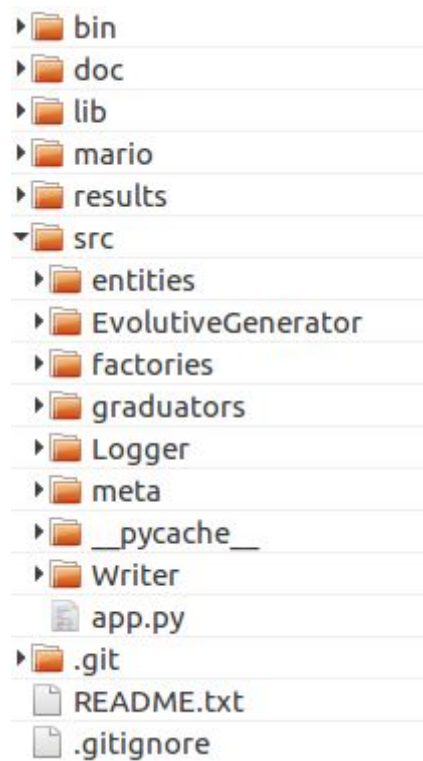


Annexe

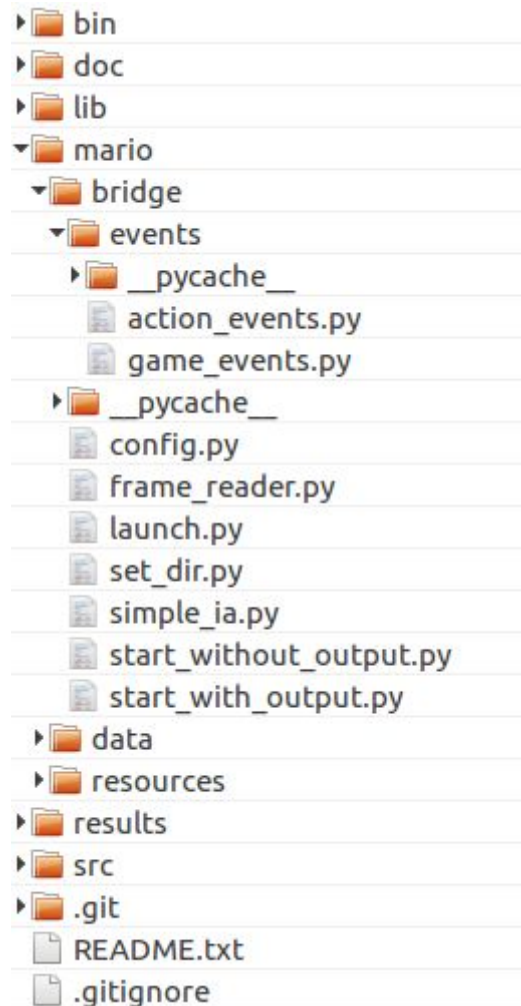
Arborescence globale du projet



Les codes sources présents dans cette annexe ont été rassemblés par thèmes.

Étape 1 : Adapter le jeu

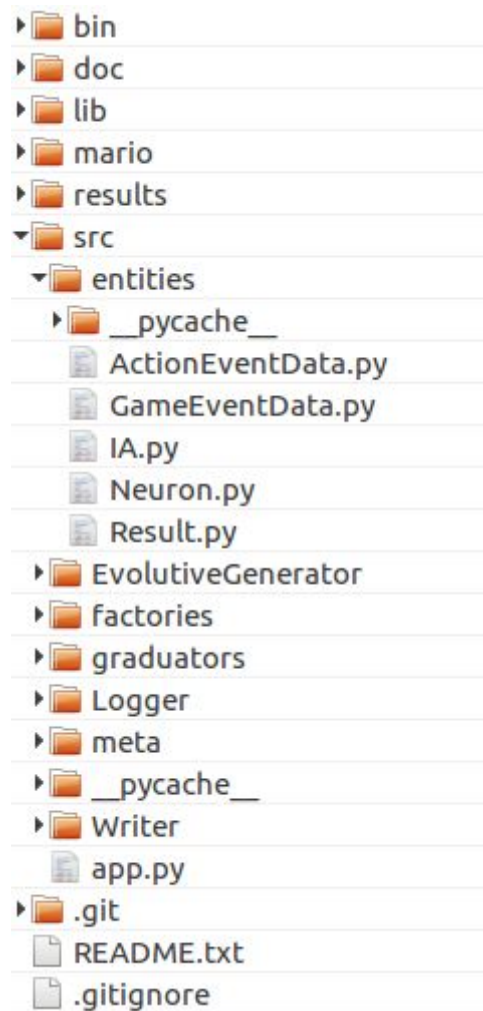
Dossier /mario/bridge



- Modifications du jeu non incluses ici (5000 lignes de codes dans 29 fichiers).
- *FrameReader* : composant chargé de lire ce qui arrive en jeu et de le traduire en évènements compréhensibles par les intelligences artificielles.

Étape 2 : Modéliser les intelligences avec des *GeneticElement*

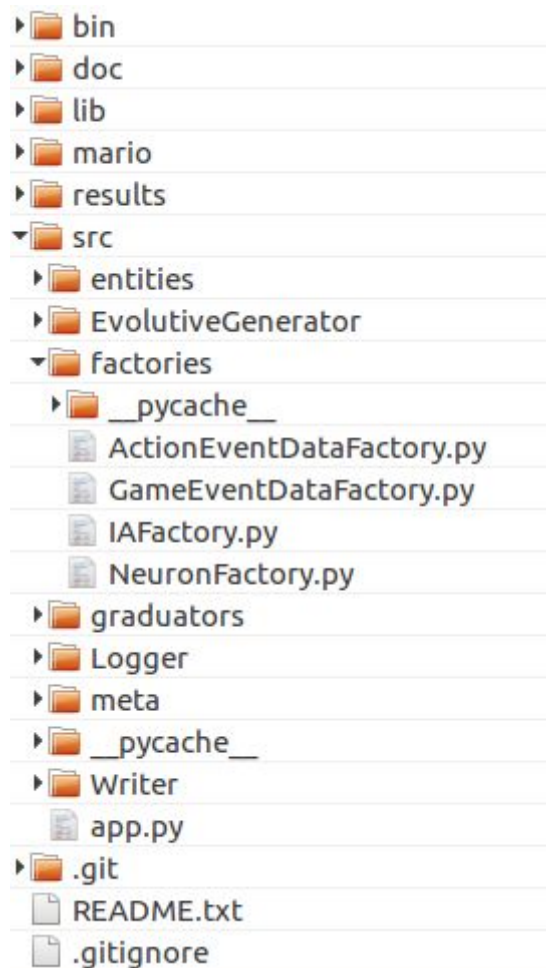
Dossier /src/entities



- *GeneticElement*
- *IA*
- *Neuron*
- *GameEvent*
- *ActionEvent*

Étape 3 : Manipuler les *GeneticElement* avec les *GeneticElementFactory*

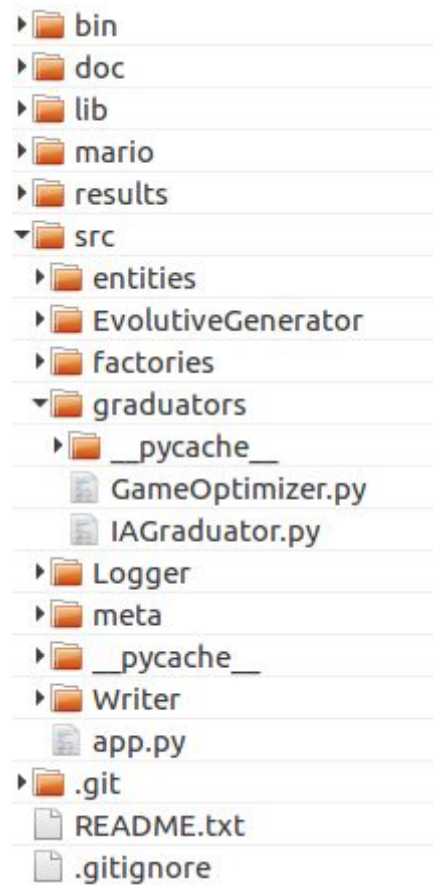
Dossier /src/factories



- *GeneticElementFactory*
- *IAFactory*
- *NeuronFactory*
- *GameEventFactory*
- *ActionEventFactory*

Étape 4 : Évaluer les intelligences avec l'*IAGraduator*

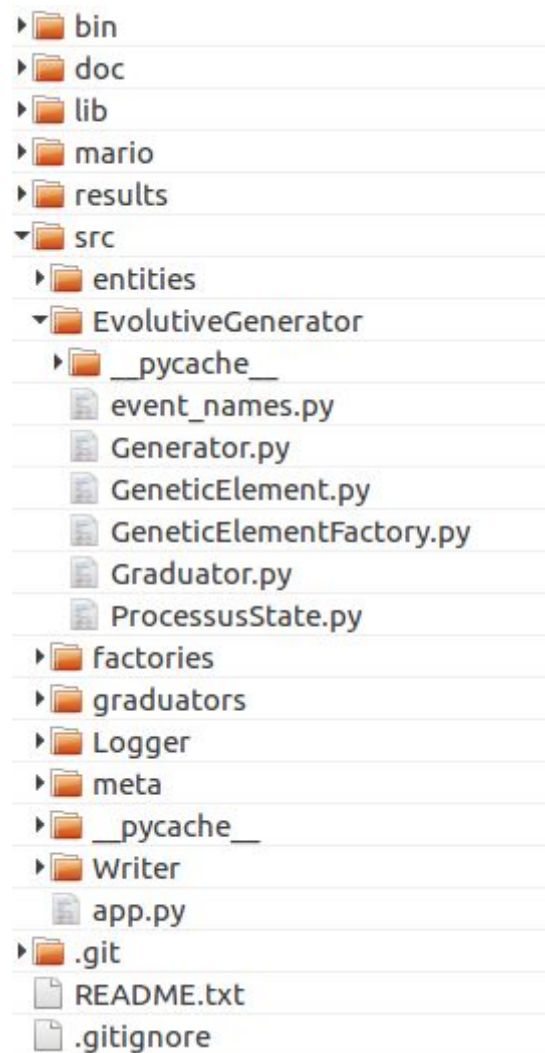
Dossier /src/graduator



- *Graduator*
- *IAGraduator*
- *GameOptimizer* : Optimise le temps d'évaluation, notamment grâce à la détection des boucles.

Étape 5 : Créer l'algorithme génétique avec le *Generator*

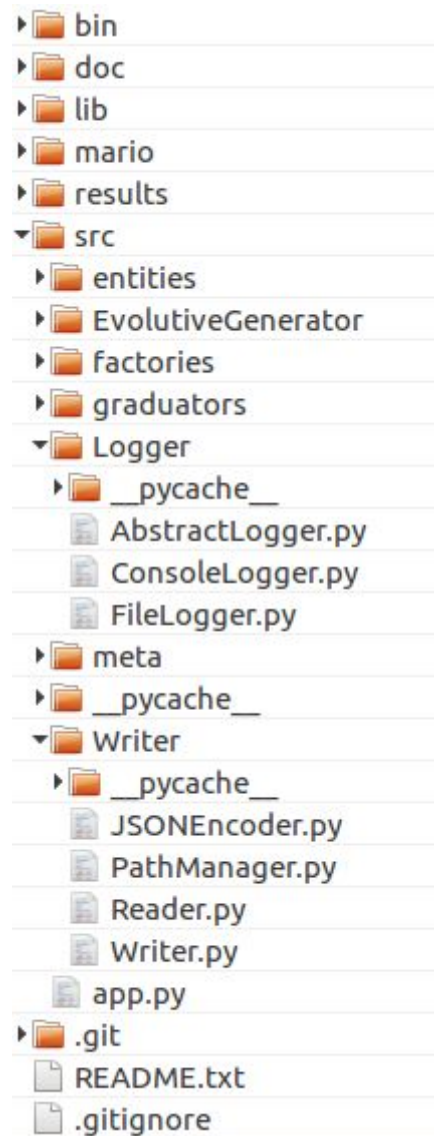
Dossier /src/EvolutiveGenerator



- *Generator*

Étape 6 : Enregistrer les données dans des fichiers

Dossiers /src/Writer et /src/Logger



- *JSONEncoder*
- *PathManager*
- *Reader*
- *Writer*

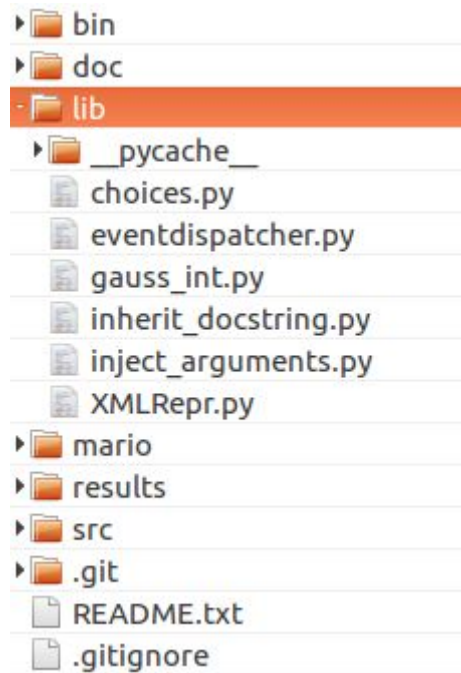
- *AbstractLogger*
- *ConsoleLogger*
- *FileLogger*

Étape 7 : L'application utilisable en ligne de commande

- */src/app.py*

Bibliothèques utilisées

Dossier /lib



- *EventDispatcher* (créé par moi sur d'autres projets)
- *XMLRepr* (créé par moi pour l'occasion)
- *inject_arguments* (créé par moi pour l'occasion)
- *inherit_docstring* (pris sur Internet)
- *gauss_int et choices* (créé par moi pour l'occasion)