**Projet IA-Quarto - Compte rendu n°01**

|  |  |
| --- | --- |
| **Motif / type de réunion:** réunion de lancement | **Lieu:** bibliothèque |
| **Présent(s) (retard/excusés/non excusés):**   * Toute l’équipe-projet : groupe 31 (Adrien CAILLET, Raphaël DELAIR, Allan DENIS, Elie GRENIER) | **Date / heure de début / durée:**  22/11/2017 de 8h30 à 10h |

**Liste de diffusion:** [elie.grenier@laposte.net](mailto:elie.grenier@laposte.net) ; [delair.raphael@gmail.com](mailto:delair.raphael@gmail.com) ; [adrien.caillet2@gmail.com](mailto:adrien.caillet2@gmail.com) ; [allandenis974@gmail.com](mailto:allandenis974@gmail.com) ; [nicolas.durand@enac.fr](mailto:nicolas.durand@enac.fr)

# Ordre du jour

1. S’assurer de la compréhension du sujet par tous les membres de l’équipe
2. Organisation du projet sur GitHub
3. Découpage du projet en module
4. Définir précisément les objets python utilisés

# Information échangées

* Qu’est-ce qu’un coup/un tour au jeu du Quarto ?

Première possibilité : un tour est soit poser une pièce, soit donner une pièce. Donc chaque branche de l’arbre correspond ou bien à poser une pièce, ou bien à donner une pièce.

Deuxième possibilité : un tour (ou un coup) est un couple d’un placement de pièce par le joueur A et d’une pièce donner ((i,j), P) au joueur B par A. Chaque branche de l’arbre correspond donc à un de ces couples.

* Utiliser tableau numpy ou liste de listes. Pas d’intérêt a priori pour numpy (sauf pour Allan) ; on ne peut pas faire de tableau np d’objet Piece, contrairement aux listes de listes.
* Représentation d’une pièce par un n-uplet ou par n attributs.
* Tour de table sur la représentation d’un jeu dans la classe Game : les attributs et les méthodes utilisés.
* Implémentation dans un premier temps d’un algorithme min-max, puis d’un alpha-beta.

# Remarques / Questions

* Remarque1 : quelques problèmes avec Git
* Remarque2 : contacter le référent pour s’assurer de ne pas faire fausse route
* Comment implémenter l’algorithme min-max ?
* Penser à jouer au Quarto pour appréhender une stratégie pour la fonction d’évaluation de l’algorithme

# Décisions

# 1. Les pièces seront représentées par des classes Piece dont l’attribut est un n-uplet de 0 et 1 pour décrire les caractéristiques.

# 2. Création d’un dossier « Projet » et « Prise en main » (pour les tests GitHub) ; dans « Projet » : « Code », « Indication », « Réflexion »,  « Ebauches »

3. On définit un **tour** d’un joueur A par le couple ((i,j), P) où (i,j) est la position de la pièce donnée par B au joueur A et placée par A et P la pièce issue du bag donnée par A à B. On définit un **coup** comme étant une subdivision d’un tour (ou bien (i,j), ou bien P). Ainsi chaque branche de l’arbre comportera un **tour** i.e. ((i,j), P). On implémentera l’algorithme minmax avec ce formalisme.

4. Création d’une classe Game qui représente un jeu et/ou un tour. Définition des attributs. Ecriture de certaines méthodes

# Actions à suivre / Todo list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Responsable** | **Délai** | **Livrable** | **Validé par** |
| Apporter un jeu de Quarto | Raph | 🡪27/11 | Le jeu | Tout le monde |
| Fonctions qui vérifient les alignements | Adrien | 🡪27/11 | Modification dans le module | Tout le monde |

Alternativement, insérer un lien vers la [todo list du projet](https://docs.google.com/previewtemplate?id=0AiIQsNq53QmIdFh5ZmNzdG8zanpOWnRhN1N3SHMxT3c&mode=public)

## *Date de la prochaine réunion*: 29/11/2017

Pas de document particulier à apporter