

PROJET HEX 205-206

Table des matières

I – Introduction.....	2
1 - Présentation du sujet	2
2 – Ce qui marche.....	2
3 – Ce qui ne marche pas	2
II – Diagramme d’architecture.....	3
III – Test.....	4
1 – Pion.....	4
2 – Plateau.....	4
3 – Ordinateur aléatoire	4
4 – Ordinateur intelligent	4
IV – Fonctionnement jeu	5
1 – Plateau.....	5
2 – Joueur.....	5
3 – IHM.....	5
V– Bilan	6
1 – Difficulté rencontré	6
2 – Ce qui est réussi.....	6
3 – Amélioration possible	6

I – Introduction

1 - Présentation du sujet

Voici notre projet Hex entre les groupes 205-206.

Lien de notre GitHub : <https://github.com/SM-Factory/Hex-205-206>

Lien vers la Release : <https://github.com/SM-Factory/Hex-205-206/releases/tag/Release>

2 – Ce qui marche

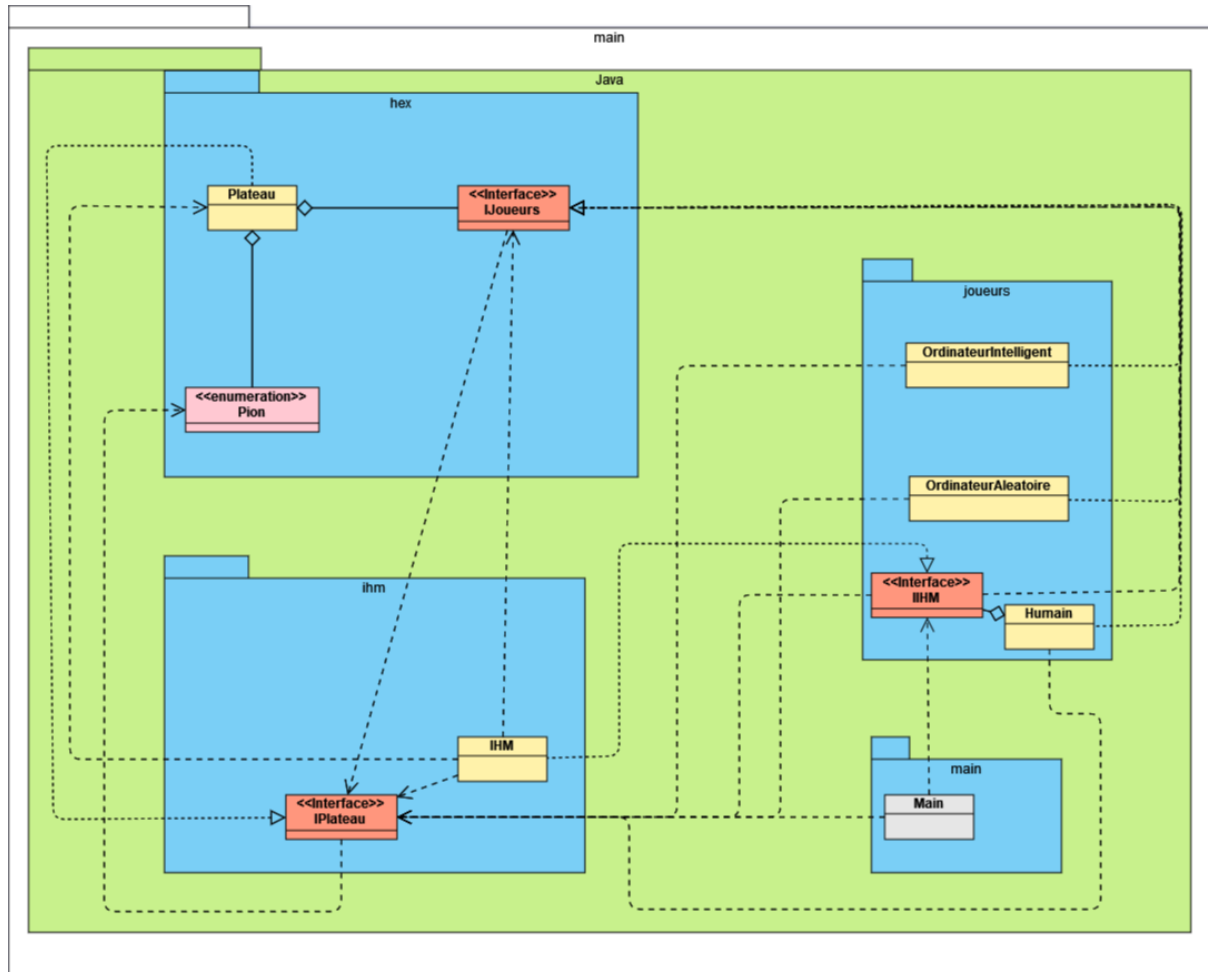
Le jeu se lance et on peut jouer une partie classique en joueur contre joueur, joueur contre ordinateur ou ordinateur contre ordinateur. Chaque ordinateur peut soit jouer aléatoirement ou de manière intelligente. Cela est prédéfini en début de partie.

Interface sous forme de texte dans la console.

3 – Ce qui ne marche pas

Nous n'avons pas pu réaliser la règle optionnelle du gâteau permettant d'échanger les pions lors du premier tour. Également, nous n'avons pas intégré d'interface graphique mais seulement textuelle. Enfin notre intelligence artificielle manque un peu d'optimisation.

II – Diagramme d'architecture



III – Test

1 – Pion

Nous avons testé les caractères de chaque pion ('X','O','').

2 – Plateau

Nous avons testé la création d'un plateau, son affichage et la jouabilité d'un tour.

Nous avons également testé si une partie est finie et son gagnant.

3 – Ordinateur aléatoire

Nous avons testé si notre ordinateur jouait sur une case du plateau qui existait qui était libre.

4 – Ordinateur intelligent

Nous avons testé si notre ordinateur jouait sur une case du plateau qui existait qui était libre.

IV – Fonctionnement jeu

1 – Plateau

Pour modifier les règles du jeu, il suffit de créer une classe qui implémente IPlateau et donc vous pouvez modifier les règles tant que toutes les méthodes de l'interface sont codées.

2 – Joueur

Pour ajouter un type de joueur, il suffit de créer une classe implémentant l'interface IJoueur. Dans cette dernière, il n'y a que la méthode jouer, qui retourne une chaîne de caractère correspondant aux coordonnées du coup jouer.

3 – IHM

Pour ajouter une interface, il suffit de créer une classe implémentant l'interface IIHM. Cette nouvelle classe doit coder toutes les méthodes de l'interface. Pour vous aider, lisez la documentation de l'interface.

V– Bilan

1 – Difficulté rencontré

Lors de ce projet, l'une des principales difficultés à été de se mettre d'accord sur l'architecture de notre application. Nous avons pris du temps pour nous décider. Puis cela a été difficile de concevoir un ordinateur intelligent de rien, donc nous nous sommes inspirés des stratégies avec des arbres de possibilités.

2 – Ce qui est réussi

Nous avons réussi à programmer le jeu avec toutes ses règles de bases de manière jouable.

3 – Amélioration possible

Nous n'avons pas pu réaliser l'interface graphique ni la règle du gâteau donc cela pourrait améliorer notre programme. De plus notre ordinateur intelligent n'est pas très optimisé est met beaucoup de temps à calculer son premier coup dans le cas où le plateau est grand.