

Amandine Fouillet
Laura Guillemot

Intelligence artificielle

*Aligner n pions sur une grille $x*y$*



27/01/2016

Sommaire

1 Présentation des jeux

2 Intelligence artificielle

3 Bilan et suggestions

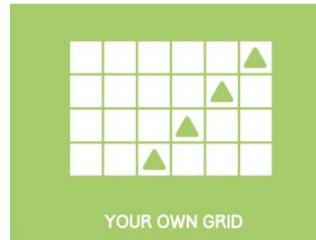
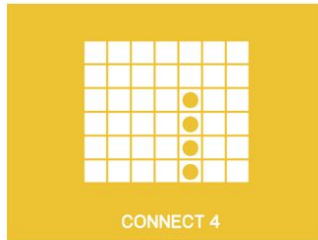
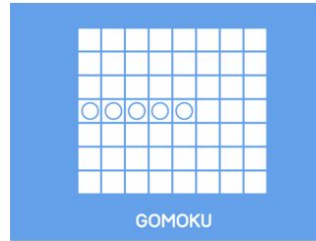
4 Démonstration



Présentation des jeux

Page d'accueil

LAG



Copyright. © 2016 Amandine Fouillet & Laura Guillemot

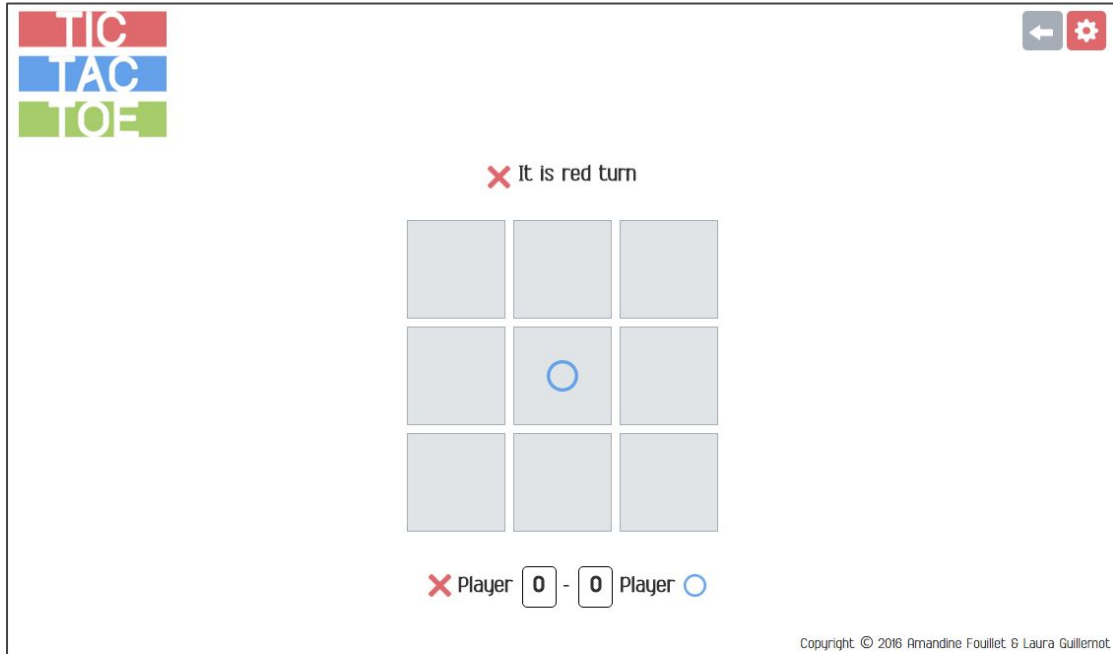
4 jeux

- Morpion
- Gomoku
- Puissance 4
- Grille libre



Présentation des jeux

Morpion

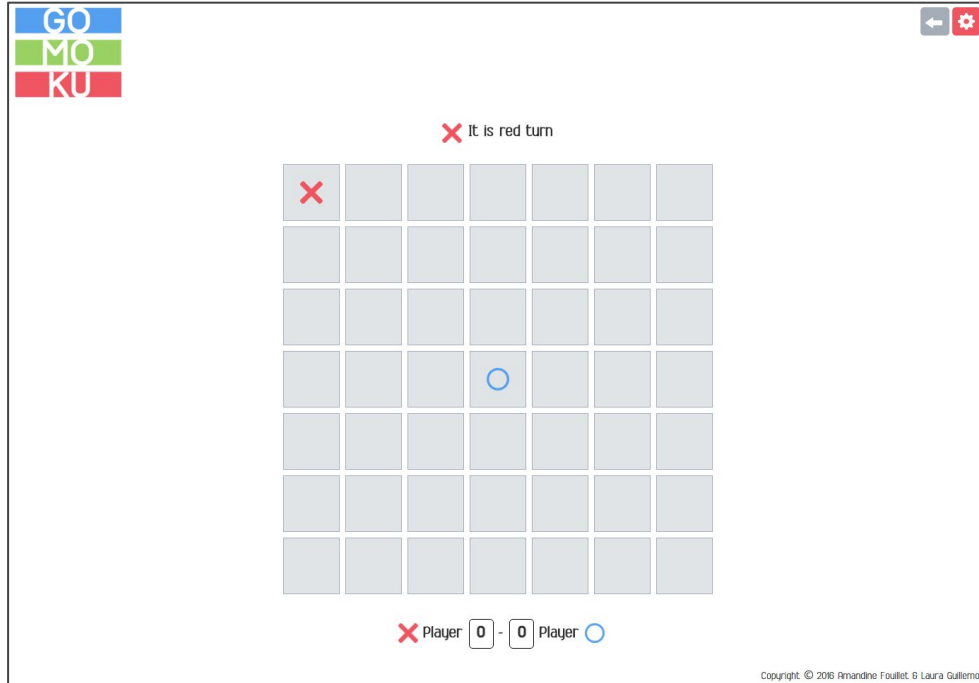


- Grille de 3*3
- Aligner 3 jetons
- Jouer
 - Contre un humain
 - Contre l'ordinateur
- Trois niveaux
 - Random
 - Medium
 - Hard



Présentation des jeux

Gomoku

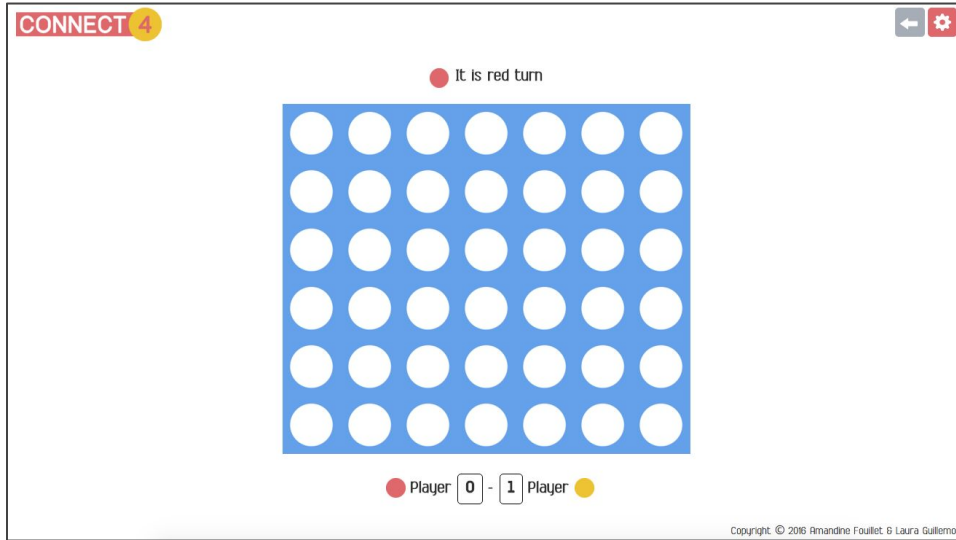


- Grille de 15*15
→ 7*7 ici
- Aligner 5 jetons
- Jouer
 - Contre un humain
 - Contre l'ordinateur
- Trois niveaux
 - Random
 - Medium
 - Hard



Présentation des jeux

Puissance 4

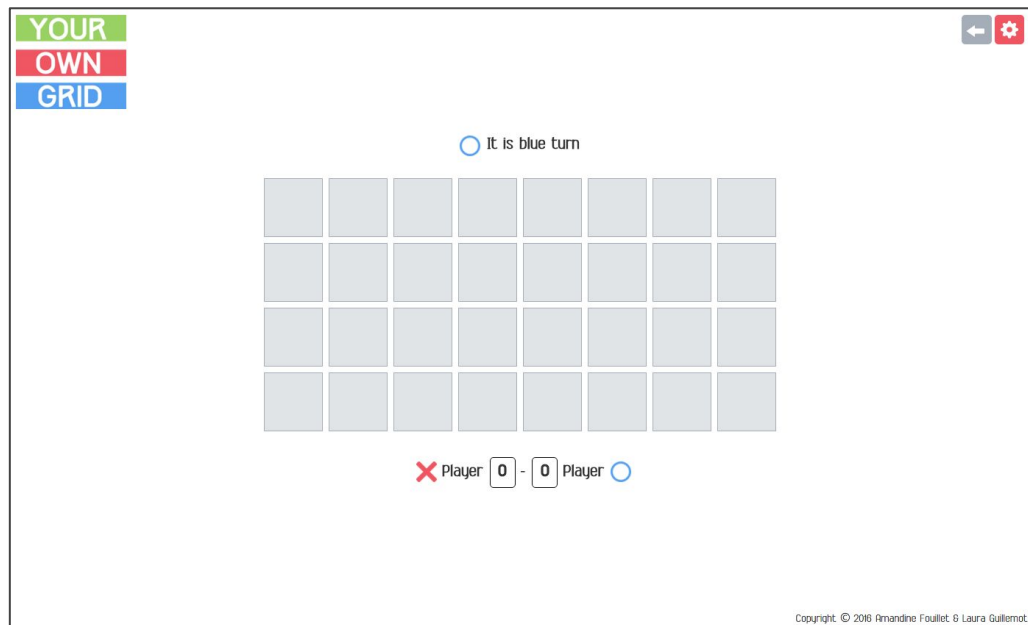


- Grille de 6*7
- Aligner 4 jetons
- Jouer
 - Contre un humain
 - Contre l'ordinateur
- Trois niveaux
 - Random
 - Medium
 - Hard



Présentation des jeux

Grille libre



- Grille de $x*y$
- Aligner n jetons
- Jouer
 - Contre un humain
 - Contre l'ordinateur
- Trois niveaux
 - Random
 - Medium
 - Hard
- Génération d'un jeu aléatoire à l'ouverture



Intelligence artificielle

Langages & code



- 2 fichiers html
 - index.html
 - game.html
- 3 fichiers js
 - graphique.js
 - game.js
 - ia.js
- 1 fichier css
 - game.css
 - (+ bootstrap)



Intelligence artificielle

Modélisation du jeu

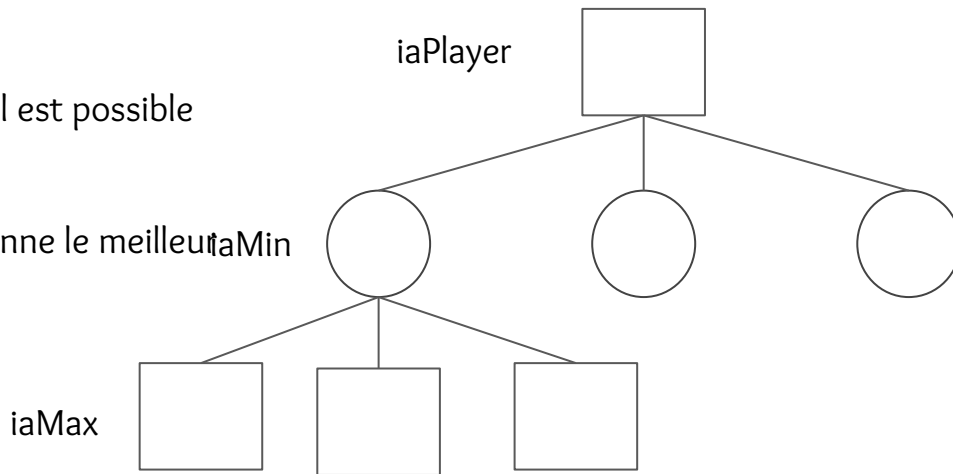
- La grille de jeu est un tableau $x * y$
- Chaque case du tableau peut prendre trois valeurs
 - grille[x][y] = 0 si la case est libre
 - grille[x][y] = 1 si la case appartient au joueur rouge
 - grille[x][y] = 2 si la case appartient au joueur bleu



Intelligence artificielle

Joueur artificiel

- Utilisation de l'algorithme min-max avec un élagage alpha/beta
- L'IA est un joueur **max**
- On simule un coup dans chaque case ou il est possible de jouer
- On garde les coordonnées du pion qui donne le meilleur résultat pour l'IA



Intelligence artificielle

Élagage alpha/beta

- beta - Meilleur coup possible pour le joueur min
- alpha - Meilleur coup possible pour le joueur max

```
function iaPlayer(grid, depth)
function iaMax(grid, depth, alpha, beta)
function iaMin(grid, depth, alpha, beta)
```

iaMax	//Termine la recherche si tmp >= beta if(tmp >= beta) return tmp //Mise à jour de alpha avec tmp alpha = Max(alpha,tmp)
iaMin	//Termine la recherche si tmp <= alpha if(tmp <= alpha) return tmp //Mise à jour de beta avec tmp beta = Min(beta, tmp)

tmp valeur de retour de la fonction iaMin/iaMax



Intelligence artificielle

Gestion des niveaux

3 niveaux

- **Easy** - On appelle la fonction randomPlayer à la place de la fonction iaPlayer
Joue un pion aléatoirement sur une des places disponible
- **Medium** - On appelle la fonction iaPlayer avec une profondeur de recherche de 4
→ Plus ou moins rapide en fonction de l'état du jeu
- **Hard** - On appelle la fonction iaPlayer avec une profondeur de recherche de 6
→ Très très lent



Intelligence artificielle

Heuristique

function iaRanting(grid)

- Si le jeu est terminé (s'il y a un gagnant)
 - Si l'IA est gagnante on retourne $1000 - \text{nombre de coups joués}$
 - Si l'IA est perdante on retourne $-1000 + \text{nombre de coups joués}$(Le nombre de coups joués = nombre de cases qui ne sont pas à 0)
- Si le jeu n'est pas terminé
 - On estime le coup de la grille de jeu pour l'IA et pour le joueur adverse
 - On retourne le coup de la grille pour l'IA - le coup pour le joueur adverse
→ On cherche à maximiser le coup de la grille pour l'IA

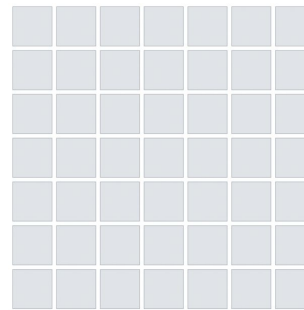


Intelligence artificielle

Estimation du coup de la grille

function alignToken(grid, nb_align, player)

- Parcours de la grille
 - En largeur (toutes les lignes)
 - En hauteur (toutes les colonnes)
 - En diagonale du Nord Ouest vers le Sud Est
 - En diagonale du Nord Est vers le Sud Ouest
- Pour chaque ligne on compte
 - Le nombre de cases libres successives ($grille[x][y] = 0$)
 - Le nombre de cases du joueur ($grille[x][y] = 1$ ou $grille[x][y] = 2$)
 - Si on rencontre une case qui n'appartient pas au joueur on remet les compteurs à 0
- À la fin de chaque ligne on ajoute à l'estimation totale le nombre de cases libres et le nombre de cases du joueur sur cette ligne
- Une fois toutes les lignes parcourues, on retourne l'estimation totale



Bilan et suggestions

Problèmes rencontrés

- Lenteur du javascript
→ Mise en place d'un worker
- Règles du gomoku
→ Mise en place de plusieurs jeux de plateau
- L'IA joue plus la défense que l'attaque, elle joue toujours à l'endroit qui minimise les possibilités de gagner de l'adversaire
→ Pas de solution pour l'instant
- Réduire intelligemment le champ de recherche de la grille du gomoku (15*15)

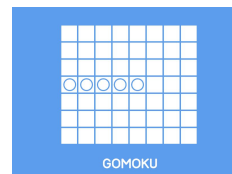


Bilan et suggestions

État des jeux

Morpion

- IA imbattable (toujours à égalité contre un humain jouant parfaitement bien)
- Réponse instantanée

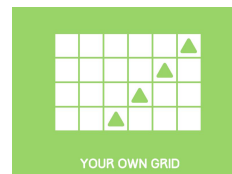
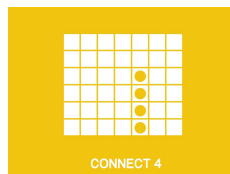


Gomoku

- IA plutôt bonne
- Mauvaise grille (7*7)
- Réponse lente

Puissance 4

- Un humain jouant correctement peut battre l'IA
- Réponse instantanée



Grille libre

- Plus x et y sont grands, plus l'humain a de chance de gagner contre l'IA
- Plus x et y sont grands, plus l'ordinateur est lent
- Avec un nombre de pions à aligner trop faible, l'IA perd



Bilan et suggestions

Ce qu'il reste à faire

- Intelligence artificielle
 - Améliorer les heuristiques
 - Augmenter la performance
- Graphisme
 - Responsive
 - Animation css
- Bonus
 - Implémenter d'autres jeux de plateaux



Démonstration

