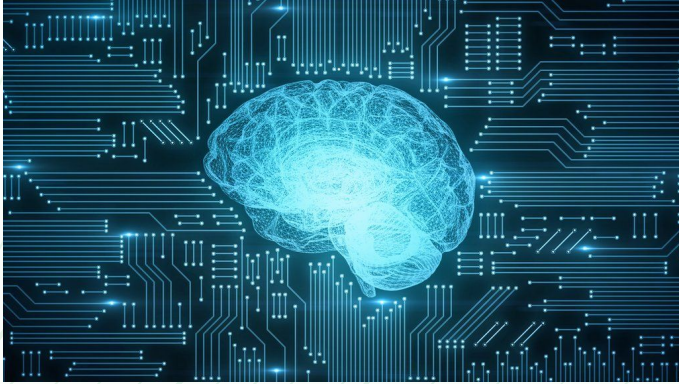


Projet de Programmation
Université de Bordeaux - Master Informatique

Jeux d'enchères

Nahaki Barry Mathias Crema Maxime Dumonteil
Nicolas Hubner Maxime Meyrat Alex Pepi

Introduction au projet



Programme / IA



Jeux d'enchères (Poker / Blackjack / Baccarat / Farot)



Jeux à information partielle

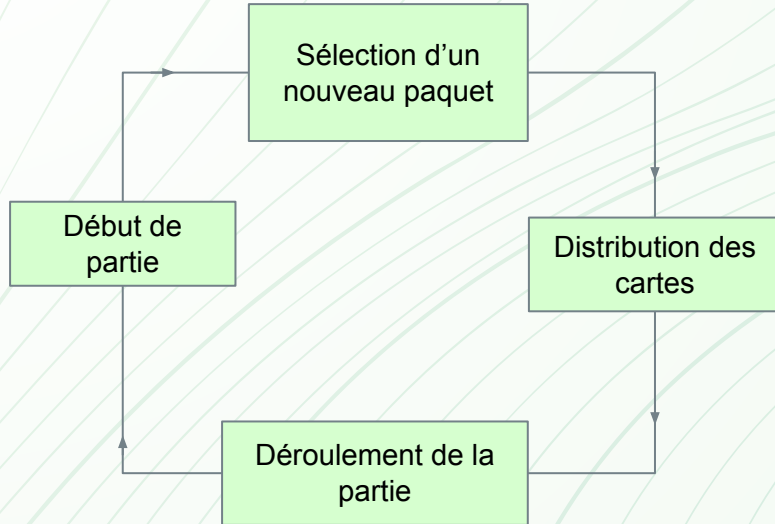
- Gains incertains
- Calcul des probabilités

Blackjack

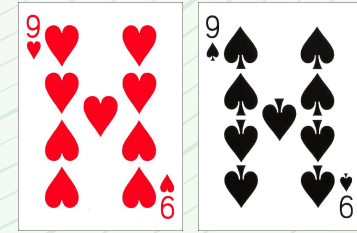


Personnalisation

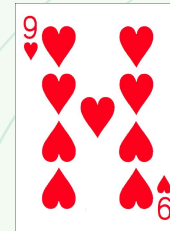
Renouvellement de paquet



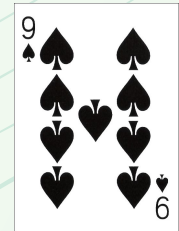
Split



Main de départ



Main 1°



Main 2°

Analyse de l'existant



Algorithmes existants



Développement

**MOTEUR
DE JEU**



IA

- Personnel
- Adapté au besoins de l'IA

- Inspiré de stratégies gagnantes
- Prise en compte des stratégies improductives/néfastes

Les besoins inhérent au fonctionnement du logiciel

❖ Moteur de jeu

- Respecter les règles
- Afficher l'état du jeu
- Afficher les résultats d'une partie
- Exécuter une ou plusieurs parties de Blackjack avec 1 à 7 joueurs.
 - Demander la mise d'un joueur
 - Demander les actions aux joueurs

❖ Un joueur du jeu

- Fournir ses actions et sa mise de départ au moteur de jeu
- IA ou humain : pas de différence pour le moteur de jeu

❖ Le lanceur de moteur

- Créer le moteur de jeu
 - Initialiser le moteur de jeu avec les paramètres renseignés par l'utilisateur.

Présentation générale du fonctionnement de l'application

Démonstration du lanceur de moteur

```
→ src git:(master) X ./run-launcher.sh --help
```

Un programme pour lancer des parties de blackjack.

Usage:

- h, --help
Affiche cette aide.
- j, --joueurs {humain|ia-basique|ia-hilo} [<nombre>] [{humain|ia-basique|ia-hilo} [<nombre>] ...]
Les types de joueurs à utiliser ainsi que le quantité.
- a, --argent <nombre> [<nombre> ...]
L'argent de départ de chaque joueurs. L'argent est donné aux joueurs dans l'ordre de --joueurs.
- p, --parties <nombre>
La quantité de parties à exécuter. Optionnel (défaut=1).
- mn, --mise-min <nombre>
La mise minimum durant les parties. Optionnel (défaut=5).
- mx, --mise-max <nombre>
La mise maximum durant les parties. Optionnel (défaut=100000).
- d, --paquets <nombre>
La quantité de paquets de cartes à utiliser. Optionnel (défaut=1).
- v, --verbose
Tout les textes sont affichés. Optionnel (défaut=seulement les résultats des parties sont affichés).
- f, --fichier <chemin/vers/fichier>
Le fichier à utiliser pour remplir les paramètres à la places des arguments de la ligne de commande. (Optionnel).

Notice d'aide du lanceur de moteur

Présentation générale du fonctionnement de l'application

Démonstration du lanceur de moteur

```
→ src git:(master) X ./run-launcher.sh --joueurs humain --argent 200 --parties 2 --mise-min 10 --mise-max 1000 --paquets 2 --verbose
```

Launch parameters:

Players : ['Human Player "humain 1":200']

Deck used : 2

Bets : [10, 1000]

Game : 2

Verbose : Yes

Exemple d'exécution du lanceur de moteur



**1 joueur
humain**



**argent de
départ :
200 \$**



2 parties



**2 paquets
de cartes**



**min : 10 \$
max : 1000 \$**

Présentation générale du fonctionnement de l'application

Déroulement de la première partie

```
humain 1
La mise minimale est : 10
La mise maximale est : 1000
Vous avez : 200
Veuillez entrer la mise que vous souhaitez mettre :
100
```

Partie 1 : Choix de la mise de départ

----- Table de jeu -----

Joueur "humain 1" :

9♠ 8♠ (17)

Mise: 100\$

Croupier :

Q♦ ? (10)

humain 1

Veuillez entrer le code de l'action que vous souhaitez faire :

c - Tirer une carte

r - S'arrêter (rester)

d - Doubler sa mise (et recevoir une dernière carte)

a - Abandonner

d

Partie 1 : Tour de jeu du joueur

Résultat pour "humain 1" :

9♠ 8♠ 3♦ (20)

humain 1 a gagné 200\$

humain 1 a 400\$

Croupier :

Q♦ 3♠ J♠ (23)

Partie 1 : Résultat de la partie

Présentation générale du fonctionnement de l'application

Déroulement de la seconde partie

```
humain 1
La mise minimale est : 10
La mise maximale est : 1000
Vous avez : 400
Veuillez entrer la mise que vous souhaitez mettre :
150
```

Partie 2 : Choix de la mise de départ

```
----- Table de jeu -----
Joueur "humain 1" :
  J♠ 10♠ (20)
  Mise: 150$
Croupier :
  10♥ ? (10)

humain 1
Veuillez entrer le code de l'action que vous souhaitez faire :
c - Tirer une carte
r - S'arrêter (rester)
d - Doubler sa mise (et recevoir une dernière carte)
a - Abandonner
r
```

Partie 2 : Tour de jeu du joueur

```
Résultat pour "humain 1" :
  J♠ 10♠ (20)
humain 1 a gagné 150$
humain 1 a 550$

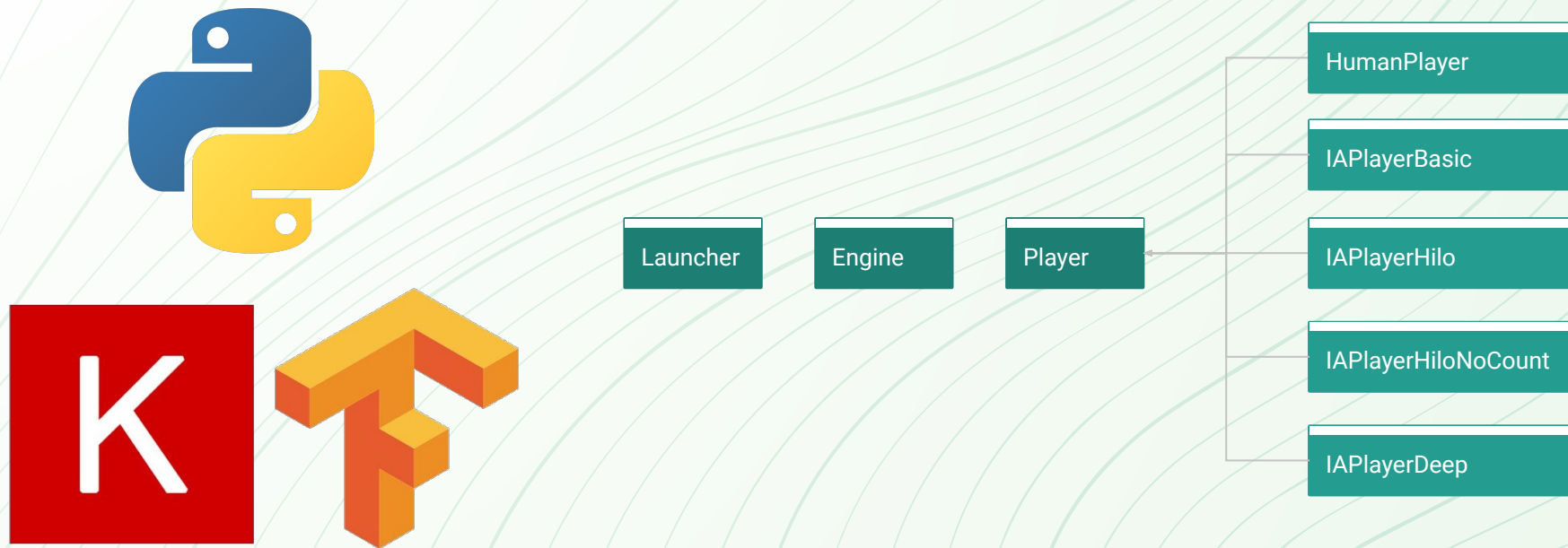
Croupier :
  10♥ 4♥ 4♠ (18)
```

Partie 2 : Résultat de la partie

```
Résultat des parties :
humain 1 a 550$
```

Résumé des parties

Choix architecturaux & conception



Choix architecturaux & conception

c Player

- m get_bet(self)
- m next_action(self)
- m notify_new_card(self, card)
- m notify_game_start(self, min_bet, max_bet, deck_id, money)
- m notify_new_bud_card(self, card)
- m notify_new_dealer_card(self, card, is_before_refill)
- m notify_deck_refill(self)
- m notify_game_end(self, win_status, money_earned)

HumanPlayer

IAPlayerBasic

IAPlayerHilo

IAPlayerHiloNoCount

IAPlayerDeep

IA : la stratégie Basique

Exemple :

La somme de la main de l'IA vaut 18



Elle calcule la probabilité de piocher une carte de valeur inférieure ou égale à 3



La probabilité est de 68%



L'IA pioche une carte

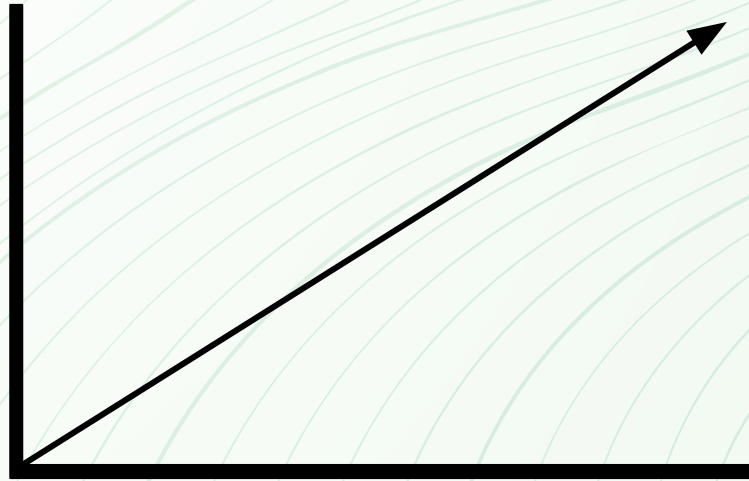
IA : la stratégie Hi-Lo (High-Low)

Carte	Valeur
2, 3, 4, 5, 6	+1
7, 8, 9	0
10, Valet, Dame, Roi, As	-1

4 de Carreau + **7** de Pique + **Roi** de trèfle → **1 + 0 - 1 = 0**

L'algorithme de mise des IAs

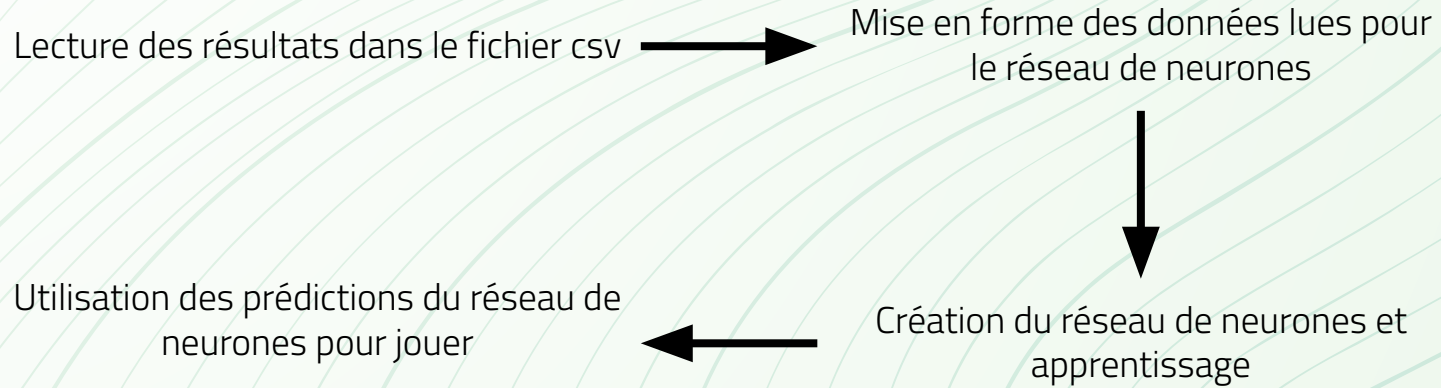
Nombre de cartes comptées



Mise faite par l'IA

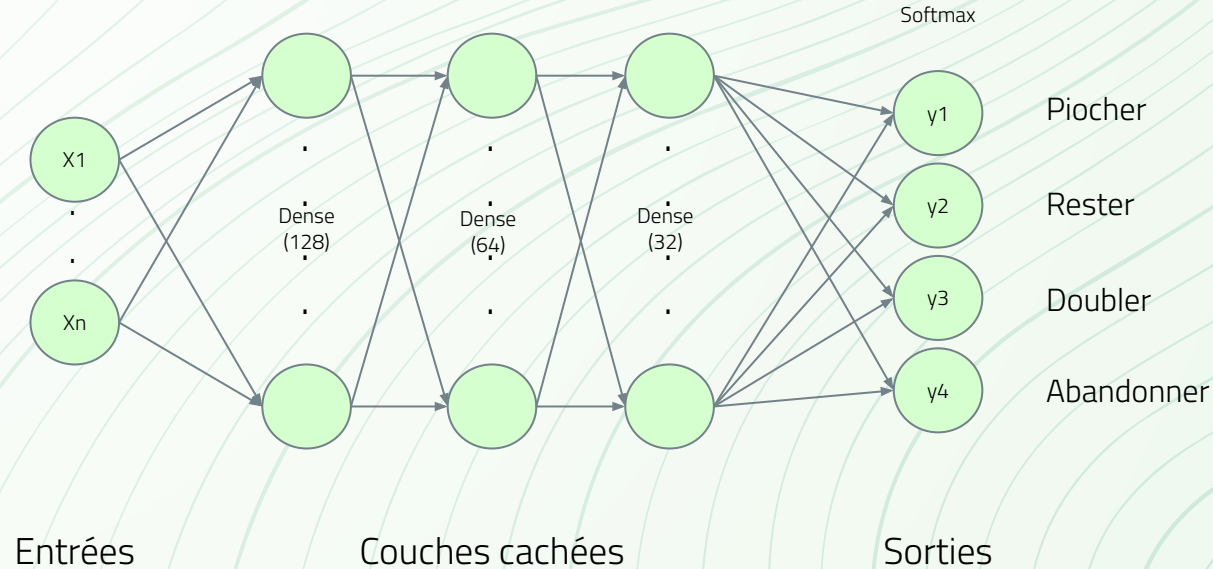
IA : stratégie Deep Learning

Description du principe :



IA : stratégie Deep Learning

Schéma du réseau :



Conclusion

- Résultats obtenus

Stratégie	Victoire
Basique	34 %
Hilo	37 %
Hilo (avec comptage des cartes)	38 %
Deep learning	~ 70 %

Conclusion

Éléments à finaliser :

- Simplification du lancement de l'application
- Affichage des statistiques sur un ensemble de parties
- Amélioration des intelligences artificielles

21

Merci

Avez-vous des questions ?