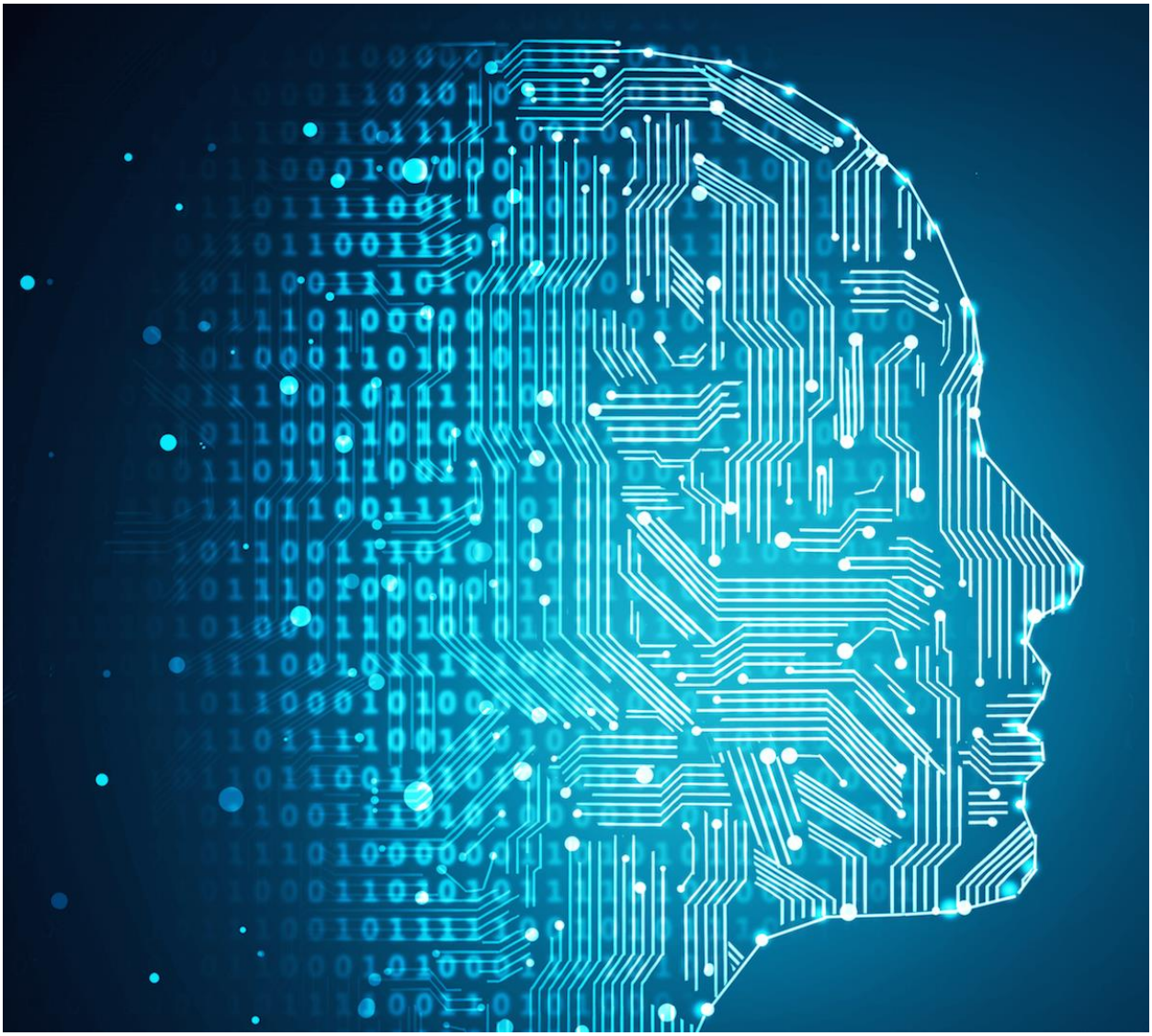


Système à base de connaissance

IFT-2003 | Intelligence artificielle | 30/11/2018

Arthur Klein | 111 244 595

Maxime Leroy | 111 244 596

Henri Longle | 111 244 597

UNIVERSITÉ LAVAL

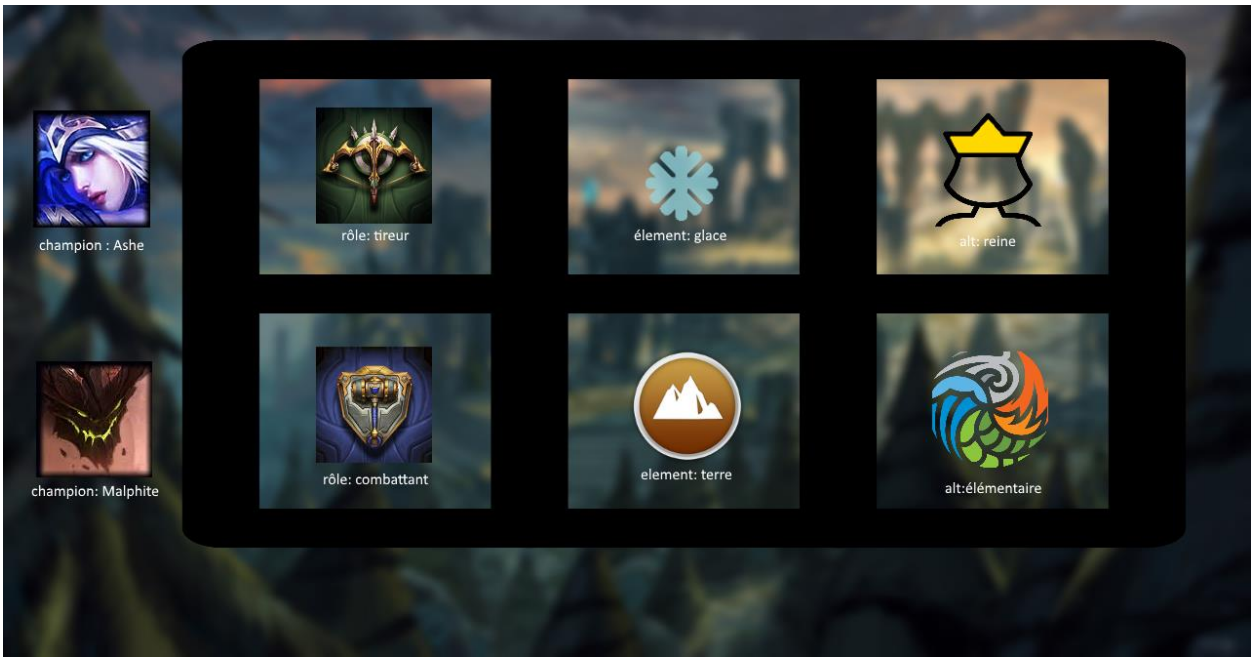
A. Description du sujet

1. Mise en contexte

Le jeu “League of Legends” est un jeu de type “MOBA” (Arène de bataille en ligne multijoueur) qui permet au joueur d’incarner un champion dans une arène dont le but est de détruire la base adverse. Chaque partie, le joueur sélectionne un nouveau champion parmi les 140 disponibles. Tous les champions disposent de leurs propres caractéristiques, par exemple le champion “Ashe” est un tireur qui utilise la magie de glace. Le joueur peut vite se retrouver perdu devant le nombre de champions.

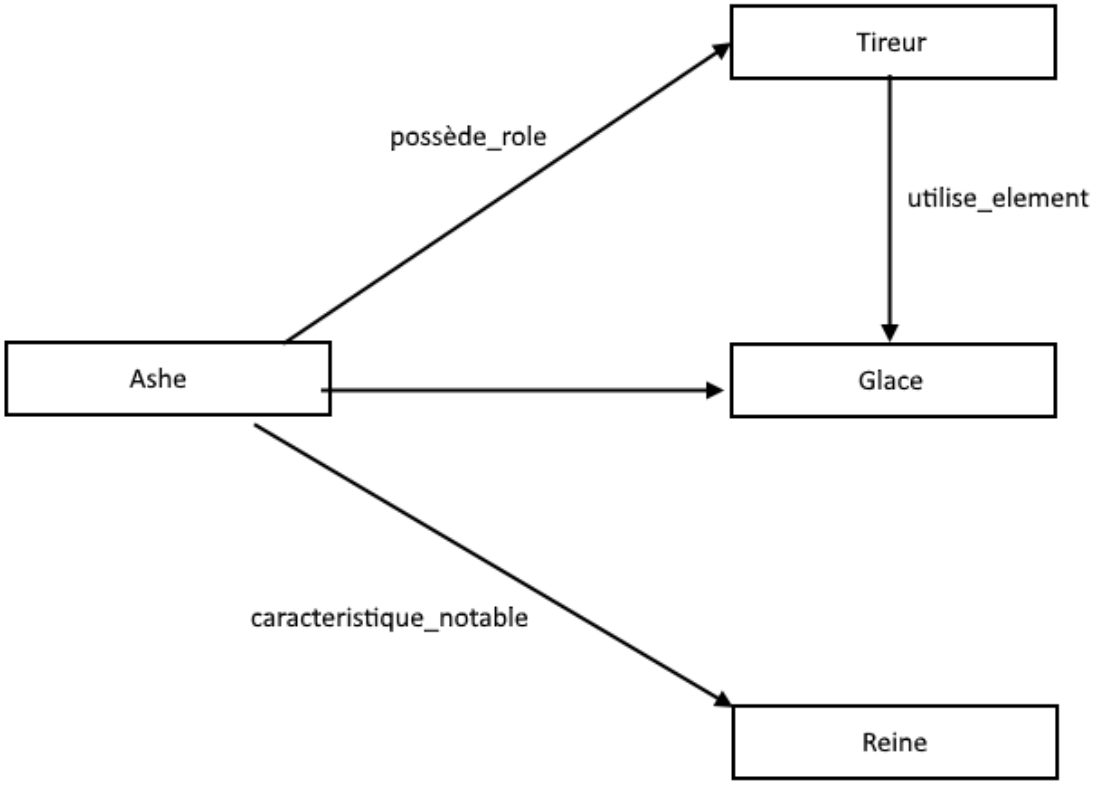
Pour aider les joueurs dans leur choix de champion, nous avons décidé de mettre en place un programme capable de déterminer un champion en fonction des préférences d’un utilisateur. L’utilisateur pourra préciser une ou plusieurs caractéristiques lui permettant au programme d’affiner les recherches et de trouver le ou les champion(s) qui lui corresponde(nt).

Voici un exemple ci-dessous :



Ashe à un role(tireur) et utilise element(glace) elle à la particularité d’être alt(reine) aussi.

2. Schéma conceptuel



3. Explication du problème

Le programme doit ressortir le(s) champion(s) ayant les mêmes caractéristiques que celles recherchées par l'utilisateur. Le programme se place donc en tant qu'expert de l'information, ayant une base de connaissances sur les différents champions existants et sur leurs caractéristiques individuelles.

L'utilisateur devra donc rentrer les différentes caractéristiques qu'il souhaite, cela aura pour effet de déterminer la conséquence, c'est-à-dire le(s) champion(s) trouvé(s). Cette conséquence est déterminée par des antécédents, ici représenté par les caractéristiques.

Ces différentes déclarations nous permettre de justifier la mise en place d'un SBC (Système à Base de Connaissances) plus précisément un SBC/règle. En effet le programme doit simuler l'avis d'un expert sur une expertise, pour cela le programme va se baser sur une base de connaissances et en tirer une conséquence en fonction d'antécédents (règles).

4. Le contenu de base

De base le SBC contient les données pour 12 champions, ainsi que 3 style(X) prédéfinis (mêlée/distance/sort). Chaque champion est décrit par un rôle, un élément, et une description alternative (alt). L'utilisateur possède deux façons différentes de trouver un champion :

Il peut d'abord définir un style qui déterminera le/les rôles associés (l'utilisateur peut préciser directement le rôle). Une fois le rôle défini, l'utilisateur choisit un élément, le programme va alors lui proposer un champion.

L'utilisateur peut aussi utiliser la recherche alternative (alt). Par exemple alt(monstre) va afficher tous les champions qui ont la spécificité d'être représenté par un monstre.

La base de connaissance se forme donc ainsi:

*"Si **rôle** et **élément** ou **alt** alors **champion**"*

B. Système développé

1. Guide d'installation

Télécharger SWI-Prolog à l'adresse suivante http://www.swi-prolog.org/download/stable.

Après son installation, compiler le fichier sbc.pro.

2. Guide d'utilisation

Afin de lancer le système à base de connaissance, écrire "play." dans le terminal.

Cette action aura pour effet de vider les données du problème enregistrées dans la mémoire du programme, pour partir sur une base saine. (Début de programme) À chaque tour, le programme va demander un nouveau prédicat.

Voici la liste des prédicats disponibles:

- champion(X) : X est un champion
- role(X) : X est un archétype de rôle, chaque champion a un archétype.
- style(X): le style(X) permet de déterminer un rôle.
- element(X): X est élément, chaque champion possède un élément principal
- alt(X): X est un élément "notable du champion", chaque champion possède un élément qui le démarque

Une fois le fait validé, le programme va conclure soit aucune, une ou plusieurs solutions possibles.

Pour quitter, entrer la commande "exit".

C. Validation

1. Présentation de cas-tests et des résultats obtenus

a. Deux faits avec des règles connues

```
?- play.
Ajoutez un fait:
|    element(feu).
Plus de nouveaux faits déduits, la BC est saturée.
Ajoutez un fait:
|    role(mage).
Nouveau fait : champion(brand)
Nouveau fait : champion(jhin)
Plus de nouveaux faits déduits, la BC est saturée.
Ajoutez un fait:
|    exit.

% Execution Aborted
?-
```

b. Deux faits avec des règles inconnues

```
?- play.
Ajoutez un fait:
|    role(toto).
Plus de nouveaux faits déduits, la BC est saturée.
Ajoutez un fait:
|    element(feu).
Plus de nouveaux faits déduits, la BC est saturée.
Ajoutez un fait:
|    exit.

% Execution Aborted
?-
```

c. Deux faits avec des règles connues

```
?- play.
Ajoutez un fait:
|    role(tireur).
Plus de nouveaux faits déduits, la BC est saturée.
Ajoutez un fait:
|    element(glace).
Nouveau fait : champion(ashe)
Plus de nouveaux faits déduits, la BC est saturée.
Ajoutez un fait:
|    exit.

% Execution Aborted
?-
```

2. Discussion sur les résultats obtenus

Après analyse des résultats, le fonctionnement du programme répond bien aux attentes quant à la résolution du problème de base.

En effet, pour les cas a) et b), dès lors que le SBC a connaissance des relations (règles), tout fonctionne parfaitement puisqu'il retourne bien les données attendues.

Afin d'améliorer les résultats obtenus, une complétion du programme avec plus de règles permettrait de gagner en précision.

D. Bilan de l'expérimentation

1. Avantages d'utilisation d'un SBC

L'application d'un système à base de connaissances peut être plus ou moins grande en fonction du nombre de règles implémentées et c'est une source hétérogène pour de nombreux types d'informations.

Aussi, ce système est beaucoup plus intelligent que des bases de données classiques, parce qu'elles traitent les données et utilisent les connaissances d'experts pour donner des solutions, des recommandations et des conseils. Ici, les solutions y sont mises en avant.

Pour finir, les connaissances peuvent être utilisées pour modifier et traiter l'information en raison du processus d'apprentissage impliqué.

2. Difficultés rencontrées pour déterminer l'expertise

Le point principal qui en ressort est la capacité et les connaissances requises à la détermination des règles, qui retranscrivent directement l'expertise dans le domaine appliqué.

Le nombre de ces règles joue également un rôle essentiel dans la précision des résultats obtenus. Aussi, établir la totalité des relations entre les champions, rôles, types, et caractéristiques demande du temps et de la concentration, car dans le cas contraire, cela influerait sur la qualité les tests.

3. Avantages et contraintes de la coquille

Les avantages sont les suivants :

- Simple dans l'utilisation
- Rapidité d'exécution
- Les fonctions ont un comportement attendu de qualité (fonctionnel et sans erreur de code)

Les contraintes sont les suivantes :

- Difficile de comprendre un code externe sans documentation précise

4. Avantages et limites du système

Les avantages sont les suivants :

- Facilité d'utilisation grâce à des phrases permettant de guider l'utilisateur
- Rapidité et capacité à trouver une ou des solutions adéquates

Les limites sont les suivantes :

- Environnement limité au terminal
- La précision dépend du nombre de données du problème

5. Améliorations à apporter au système

Afin de gagner en efficacité, et donc en précision, il serait intéressant d'implémenter une "aide au système", qui essaierait de deviner les règles expertes en utilisant un algorithme de Deep Learning.

Aussi, permettre le visionnage des faits déjà entrés via une commande, permettrait d'augmenter l'efficacité de l'utilisateur à trouver une solution.

Pour finir, le point le plus important pour améliorer l'appréhension du système, serait de développer une interface graphique accompagné d'un tutoriel.

E. Grille d'autoévaluation

Critères	De la meilleure évaluation	…		à la pire évaluation
Description du sujet **20 %**	Le sujet est clairement décrit avec un schéma conceptuel, une explication du problème à résoudre et la base de connaissances	Le sujet est clairement décrit, mais il manque un des éléments demandés.	Le sujet est décrit seulement avec un des éléments demandés.	Le sujet est ambigu ou non décrit.
Système développé **30 %**	Une implémentation a été réalisée avec au moins une 10aine de règles expertes. Tous les fichiers nécessaires à l'exécution sont fournis, dont le guide d'utilisation et/ou d'installation, et le programme s'exécute sans erreur.	Une implémentation a été réalisée avec au moins une 10aine de règles expertes. Les fichiers sont fournis et le programme s'exécute moins de 5 fois sur 10 avec erreur.	Une implémentation a été réalisée avec moins de 10 règles expertes. Les fichiers sont fournis et le programme s'exécute moins de 5 fois sur 10 avec erreur.	Aucune implémentation ou le programme s'exécute plus de 5 fois sur 10 avec erreur et/ou les fichiers sont incomplets ou non fournis.
Validation **20 %**	Un ensemble de cas-tests est présenté ainsi que les résultats obtenus et leur discussion.	Il manque soit la liste des cas-tests, soit les résultats obtenus, soit la discussion.	Il y a seulement les cas-tests, les résultats obtenus ou leur discussion.	Aucun résultat.
Bilan de l'expérimentation **20 %**	Le bilan est complet : avantages d'utiliser un SBC, difficultés rencontrées pour déterminer l'expertise, avantages/contraintes de la coquille, avantages/limites du système, améliorations à apporter au système.	Il manque un ou deux éléments dans le bilan.	Il manque plus de deux éléments dans le bilan.	Peu ou pas de bilan.
Expression écrite **5 %**	Le rapport ne contient aucune faute (vocabulaire, grammaire, syntaxe, etc.).	Le rapport ne contient pas plus d'une dizaine de fautes (vocabulaire, grammaire, syntaxe, etc.).	Le rapport ne contient pas plus de 5 fautes par page (vocabulaire, grammaire, syntaxe, etc.).	Le rapport contient plus de 5 fautes par page (vocabulaire, grammaire, syntaxe, etc.).
Présentation du rapport **5%**	Le format demandé est respecté. Le rapport est paginé.	Le rapport n'est pas paginé ou il manque un élément du format.	Le rapport n'est pas paginé. Il manque 2 éléments du format.	Il y a plus de 2 éléments du format qui ont été oubliés.