

Examen final IA01 A09
Université de Technologie de Compiègne

Durée : 2H

Les documents ne sont pas autorisés

Tous ordinateurs, toutes communications sont interdits

Utilisez quatre copies séparées :

- *une copie pour la partie I*
- *une autre copie pour la partie II*
- *une autre copie pour la partie III*
- *une autre copie pour la partie IV*

1. Qu'avez-vous retenu du cours (5 points) ?

- 1 - Que signifie l'acronyme RNA ? Donnez trois propriétés remarquables d'un RNA.
- 2 - Dans quels buts utilise-t-on une ontologie en ingénierie des connaissances ?
- 3 - Que veut dire l'acronyme OWL ? A quoi sert-il ?
- 4 - Quel est l'intérêt des méthodes de programmation des algorithmes génétiques par rapport aux méthodes traditionnelles de résolution de problème ?
- 5 - Que signifie l'acronyme SMA ? Qu'est-ce qu'un agent ?

2. Frames (5 points)

On rappelle qu'un frame peut être représenté par une structure de données à trois niveaux dont le premier sert à repérer le *frame*, le suivant (*slot*) à décrire une propriété, et le troisième (*facet*) à donner des informations plus précises sur la propriété en question. On peut spécifier la structure de données correspondante comme suit :

```
<frame> ::= (<frame-name> {<slot> }+)  
<slot>  ::= (<slot-name> {<facet>}+)  
<facet> ::= (<facet-name> <value>) | ( $IF-ADDED <demon-name> )  
                                     | ( $IF-REMOVED <demon-name> )  
                                     | ( $IF-NEEDED <demon-name> )
```

où + indique une occurrence ou plus.

Dans un système de frames, les facettes sont définies une fois pour toutes, alors que les slots et les frames eux-mêmes sont définis par l'utilisateur en fonction des besoins de la représentation.

Dans ce qui suit, on considère seulement les facettes : \$value, \$default, \$if-needed, \$if-added et \$if-removed.

Par ailleurs, on distingue deux types de frames, le premier représentant des *concepts*, le second représentant des *individus* (manifestations de ces concepts).

On souhaite représenter à l'aide de ce système des figures géométriques dans un plan muni d'un repère orthonormé.

Exemples :

```
(setq Figure '(Figure (type ($value concept))
                      (couleur ($default "noir"))
                      (nom ($if-needed generer-nom))
                      )
)
```

```
(setq Rectangle '(Rectangle (type ($value concept))
                             (is-a ($value Figure))
                             (nom ($default "R"))
                             (longueur ($default 0))
                             (largeur ($default 0))
                             (perimetre ($if-needed calculer-perimetre))
                             (affichage ($if-added dessiner-rect)($if-removed effacer-rect))
                             )
)
```

1. Création d'un individu (2 points)

On suppose que l'on dispose d'une fonction *make-individual* permettant de créer un individu à partir d'un concept.

Exemple :

```
> (make-individual 'Rectangle 'longueur 5 'largeur 2)
R0
```

1.1. Ecrire le frame associé au rectangle R0. Quelle est la couleur de ce rectangle ?

1.2. Peut-on créer un rectangle de couleur rouge ? Si oui, écrire la commande correspondante.

1.3. Ecrire une fonction lisp *allowed-slots* retournant la liste des slots autorisés pour un concept donné.

2. Accès à l'information (3 points)

2.1. On souhaite disposer d'une fonction *getv* permettant d'obtenir la valeur d'une propriété pour un individu donné. Exemples:

```
> (getv 'R0 'nom)
"R"
> (getv 'R0 'perimetre)
14
> (getv 'R0 'largeur)
2
>(getv 'R0 'couleur)
"noir"
```

Décrire le fonctionnement de *getv* pour chacun de ces exemples.

2.3. Ecrire l'algorithme puis le code Lisp de la fonction *getv* permettant d'obtenir la valeur d'une propriété pour un individu donné.

3. Représentation objet et CLOS (5 points)

On se propose de concevoir un jeu vidéo, inspiré d'un jeu japonais basé sur une interaction multimodale. Le joueur est immergé dans un univers 3D interactif peuplé de créatures fantastiques qu'il doit vaincre. Pour ce faire, il dispose d'une sarbacane dans laquelle il peut souffler. La sarbacane est dotée d'un capteur d'haleine et d'un capteur de force du souffle. L'utilisateur a la possibilité de manger ou non certains aliments et de souffler ensuite sur les créatures pour tenter de les détruire.

En tant que concepteur de ce jeu, vous devrez commencer par décrire la hiérarchie de ces créatures, puis vous programmerez les attaques possibles.

3.1 Représentation objet (2 points)

On suppose des créatures fantastiques, divisées en plusieurs groupes : les morts-vivants, les fantômes et apparitions, les créatures mythologiques. Les morts-vivants se décomposent en vampires, zombies, momies. Les fantômes et apparitions se décomposent en djinns, démons, poltergeist. Une créature fantastique est caractérisée par un nom, une date de naissance, une force, une position 3D. Un mort-vivant est caractérisé par une date de mort et une date de résurrection. Dracula est un vampire qui démarre la partie avec une force de 10 et Michael Jackson est un zombie avec une force de 50.

1. Donner une représentation graphique de la modélisation objet de ces créatures fantastiques. Dans cette représentation, vous exprimerez les liens d'héritage, d'instances ainsi que les attributs.
2. Ecrire en CLOS une représentation des objets : créature fantastique, mort-vivant, vampire ainsi que de l'instance Dracula.

3.2 Fonction d'attaque (3 points)

Vous devez programmer la fonction d'attaque des vampires. Cette fonction prendra en entrée une liste des aliments détectés par le détecteur d'haleine. Cette liste peut être vide ou composée de 1 ou 2 aliments. Les aliments détectés sont représentés par des a-lists composées du nom de l'aliment et d'un coefficient de saveur : valeur négative pour le salé et valeur positive pour le sucré.

Exemple :

```
(ail . -10), (fromage . -5), (sel . -1), (chocolat . 10),  
(fraiseTagada . 10)
```

La saveur permet de déterminer l'impact de l'aliment sur la force du vampire. Les fraises Tagada et le chocolat augmentent les forces de +10, l'ail les diminue de -10, le fromage de -5 et le sel de -1. Une composition de deux aliments de mêmes saveurs (salées ou sucrées) additionne les coefficients associés. Une composition de deux aliments de saveurs différentes (salée et sucrée) multiplie les coefficients associés et diminue la force du vampire (e.g. : une composition chocolat/sel diminue les forces de -10, une composition chocolat/ail diminue les forces de -100).

Un vampire meurt si ces forces sont inférieures à 50.

On imagine que pour un zombie, seul le sel permet de diminuer les forces, les autres aliments n'ayant aucun impact. Pour d'autres créatures fantastiques, on imagine des flèches empoisonnées.

Ecrire en CLOS la fonction d'attaque d'un vampire qui prend en paramètre la liste des aliments détectés et opère la modification des forces du vampire attaqué.

3.3 Réflexion (1 point)

Imaginons à présent des êtres humains dans la scène 3D. Ces humains peuvent être mordus par des vampires. Ils se transforment alors en vampire. Comment modéliseriez-vous ces êtres humains et changeriez-vous alors la représentation précédemment proposée ?

4. Raisonnement à partir de cas (RàPC) (5 points)

Problème :

Nous traitons le problème de l'estimation du prix d'un jeu vidéo d'occasion. Nous ne disposons pas de règles ou de méthodes permettant d'estimer le prix d'un jeu donné. Par contre, nous disposons des annonces suivantes :

- La première annonce concerne le jeu vidéo : Mario party 8 sur Wii. C'est un jeu de type « Action/Aventure », pour 1, 2, 3 ou 4 joueurs et jouable à partir de 3 ans. Le jeu est en très bon état. Son prix : 35€
- La seconde annonce concerne le jeu vidéo : Virtual Tennis 2009 sur PS3. C'est un jeu de type « plate forme », pour 1, 2, 3 ou 4 joueurs et jouable à partir de 3 ans. Le jeu est en bon état. Son prix : 30€
- La troisième annonce concerne le jeu vidéo : Two World sur X-box 360. C'est un jeu de type « jeu de rôle », pour 1 à 8 joueurs et jouable à partir de 12 ans. Le jeu est en on état. Son prix : 15€

En discutant avec un vendeur de jeux vidéo, nous avons appris que :

- Suivant la console, les types de jeux recherchés ne sont pas les mêmes (les plus recherchés étant les plus chers):
 - La Wii est une console familiale, elle est faite pour les sportifs, les joueurs qui aiment jouer en communauté, les enfants, les femmes.
 - La X-box360 est destiné aux fans de doom-like et autres jeux FPS (first-person shooter).
 - La PS3 concerne les fans de RPG (jeux de rôle massivement multi-joueurs) mais aussi les fans de jeux de vitesse (courses).
- Suivant l'état du jeu (très bon, bon, mauvais, etc.) les prix peuvent variés (exemple un Mario party 8 sur Wii en très bon état coute 35€, le même jeu mais en bon état seulement ne coute plus que 25€).

Question :

1. Rappelez le principe du raisonnement à partir de cas, nommez puis explicitez les différentes étapes le composant. (1 point)
2. Notre objectif est de déterminer le prix de vente d'un jeu vidéo. Déterminez les descripteurs pertinents pour le cas cible. Donnez une représentation (sous forme de liste) d'un cas et de la base de cas. (1 point)
3. Donnez l'algorithme de la fonction *CasLePlusProche*(*MonCas*, *MaBaseCas*) permettant de retrouver dans la base de cas : *MaBaseCas* le cas le plus proche du cas : *MonCas*. Vous supposerez qu'une fonction *Similarité*(*Cas1*, *Cas2*) existe et renvoie la mesure de similarité entre deux cas.(1.5 points)
4. Définissez des règles d'adaptation afin d'adapter le cas trouvé puis donnez l'algorithme de la fonction *AdapterCas*(*MonCas*, *CasTrouvé*) permettant d'adapter la solution du cas trouvé le plus proche : *CasTrouvé* en fonction du cas : *MonCas*.(1.5 points)