

IA dans les jeux

MHiCS

(Motivation and Hierarchical with Classifier System)

Architecture motivationnelle et hiérarchique à base de système de classeurs

Plan

- Contexte
- Architecture (1) : optimisation des critères
- Architecture (2) : optimisation temporelle
- Apprentissages
- Confrontation
- Conclusion
- Questions

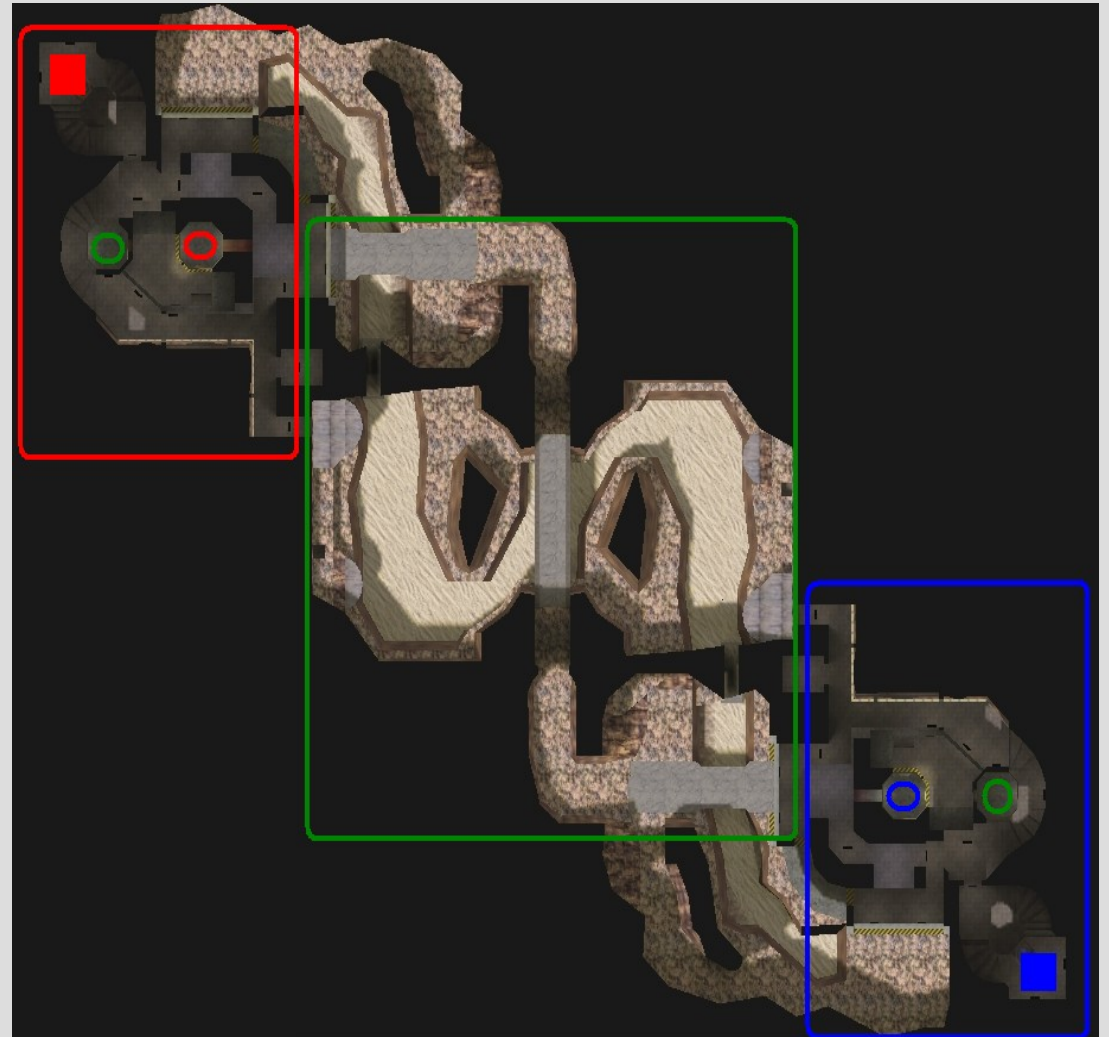
Contexte

- Type de jeu : FPS (First Person Shooter)
- Jeu : TFC (Team Fortress Classic)



Contexte

- Mode : CTF
(Capture The Flag)
- Deux équipes
 - rouge
 - bleue

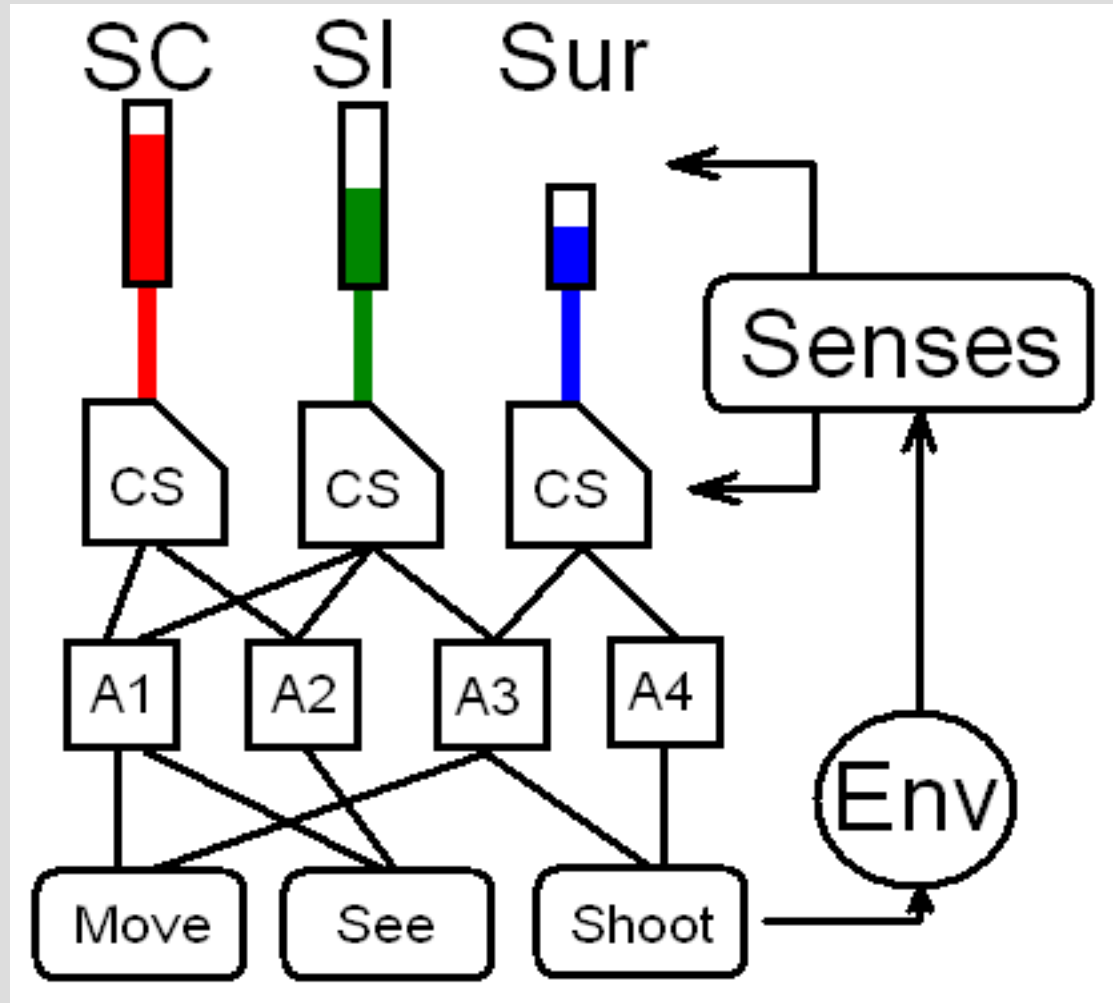


Objectifs d'un joueur

- Score collectif
(+10 pts par drapeau capturé)
- Score individuel
(+10 pts par drapeau capturé,
+1 pt par adversaire „tué“,
-1 pt par allié „tué“)
- Agressivité
(„tuer“ le plus d'adversaires possibles)
- Survie
(„mourir“ le moins souvent possible)

Architecture

- Niveau 1 : Motivations
- Niveau 2 : Système de classeurs
- Niveau 3 : Actions activables
- Niveau 4 : Ressources



Motivations

- FP : Facteur de Personnalité
 - importance donnée à une motivation
 - hauteur de la jauge
- VM : Valeur Motivationnelle
 - accomplissement d'une motivation à un instant donné
 - remplissage de la jauge
 - varie de 0 à 1 (0 = besoin satisfait)
 - fonction sigmoïde
- IM : Intensité Motivationnelle
 - $IM_{\text{motivation}} = VM_{\text{motivation}} \times FP_{\text{motivation}}$

Systeme de classeurs

- Type „Si condition alors action“
- Associés à une Force dans $[-1 ; 1]$
 - positive : aide à satisfaire motivation
 - négative : la contrecarre
- Motivation = un CS
- CS = tableau de plusieurs classeurs
- Classeur = une ligne d'un CS

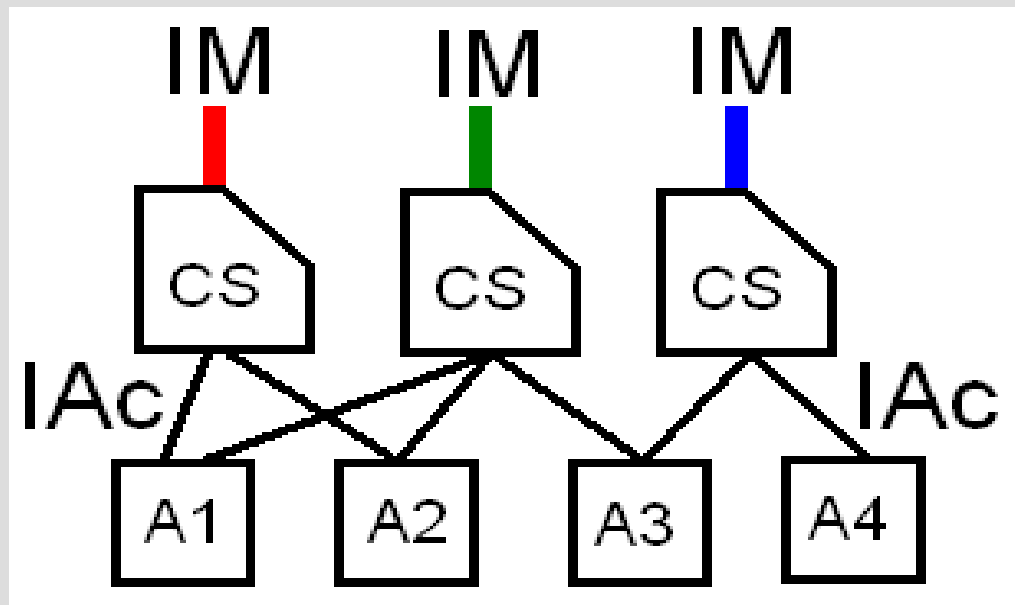
Systeme de classeurs

- Exemple CS pour „Score Individuel“

N°	Force	Conditions	Action
1	0.8	J'ai le drapeau = faux	Aller salle drapeau adverse
2	0.5	Cible dans mon équipe = faux Distance cible = proche Cible porte un drapeau = vrai	Tirer sur cible
3	0.4	Cible dans mon équipe = faux	Tirer sur cible
4	0.9	J'ai le drapeau = vrai	Aller salle drapeau allié

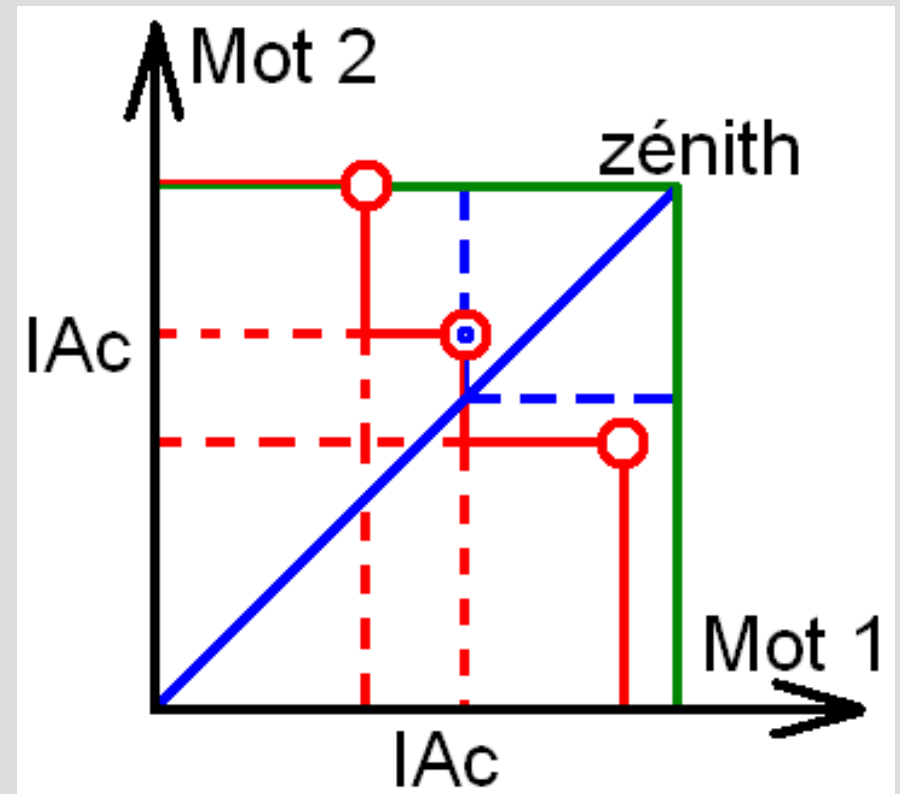
Systeme de classeurs (1)

- De chaque CS, on ne garde que les classeurs activables (condition vérifiée)
- IAc (Intensité d'Activation)
→ $IAC_{mot, act} = IM_{mot} \times \max[Force_{mot, act, CA}]$ classeurs activables



Actions activables (1)

- Ressources limitées
 - besoin de classer les actions activables
 - définir IE (Intensité d'Exécution par action)
- Problème multi-critères
 - motivation = critère
 - action = solution
- Zénith : englobe toutes les solutions
- Meilleur compromis : plus petite distance de Tchebycheff



Actions activables (1)

- Point zénith : I_{Ac}^*

$$I_{Ac}^*_{\text{motivation}} = IM_{\text{motivation}} \times \max[\text{Forces}_{M,C}]_{\text{motivations, classeurs}}$$

- Calcul IE :

→ distance point zénith

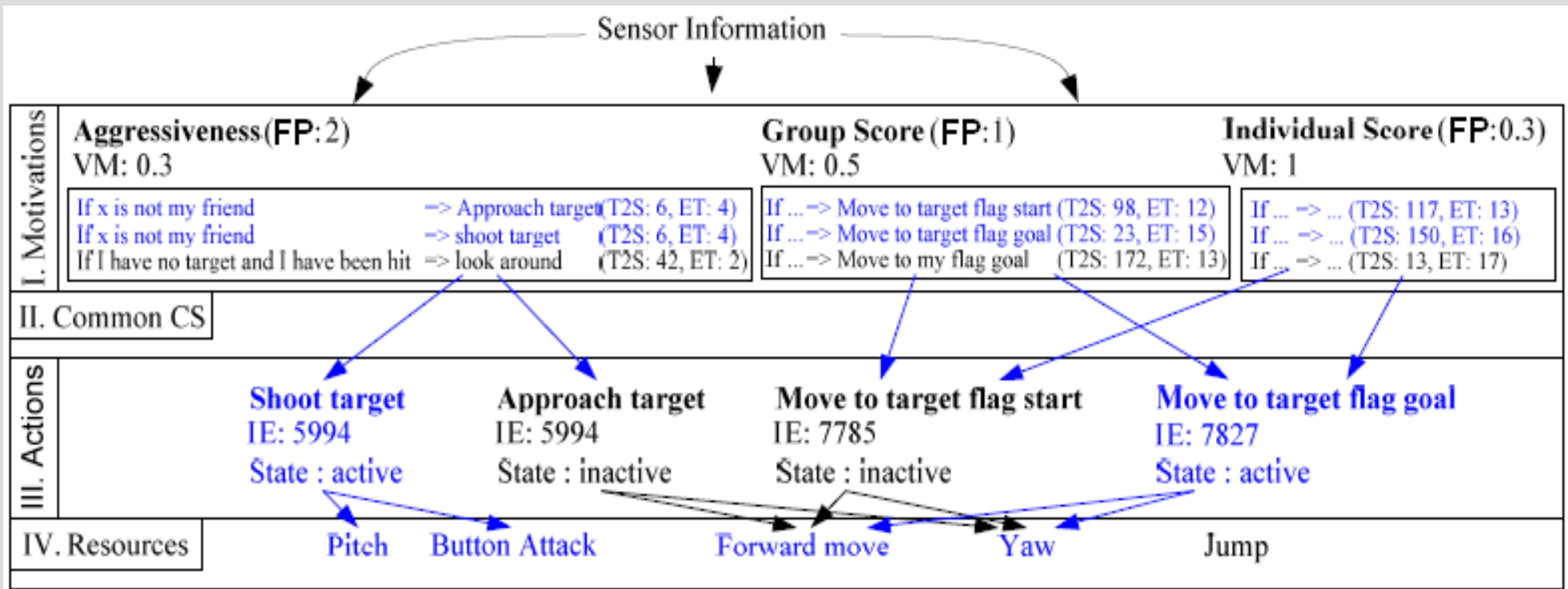
→ facteur bruit aléatoire

$$IE_{\text{action}} = \max[I_{Ac}^*_M - I_{Ac}_{M,\text{action}}]_{\text{motivations}} + \text{Bruit}_{\text{action}}$$

- Plus IE faible, plus action importante
- Tri des actions par IE croissant et allocation des ressources

Systeme de classeurs (2)

- Ajout de paramètres :
 - T2S : Time to Success, pas de temps jusqu'au calcul suivant de la VM
 - ET : Execution Time



Actions activables (2)

- Pour un classeur et une action donnée :
$$IE_{\text{classeur, action}} = (IM \times 10000) - \max[T2S ; ET]$$

T2S et ET → départager actions ayant mêmes IM
- Pour une action donnée :
$$IE_{\text{action}} = \sum [IE_{C, \text{action}}]_{\text{classeurs}}$$
- Tri des actions par ordre décroissant des IE

Ressources

- Allocation des ressources selon ordre des actions
- Une ressource ne peut être utilisée que par une action → évite tout conflit
- Si ressource déjà utilisée, action invalidée, passage à l'action suivante
- Affectation de l'environnement, nouvelle perception, on réitère le processus

Apprentissages

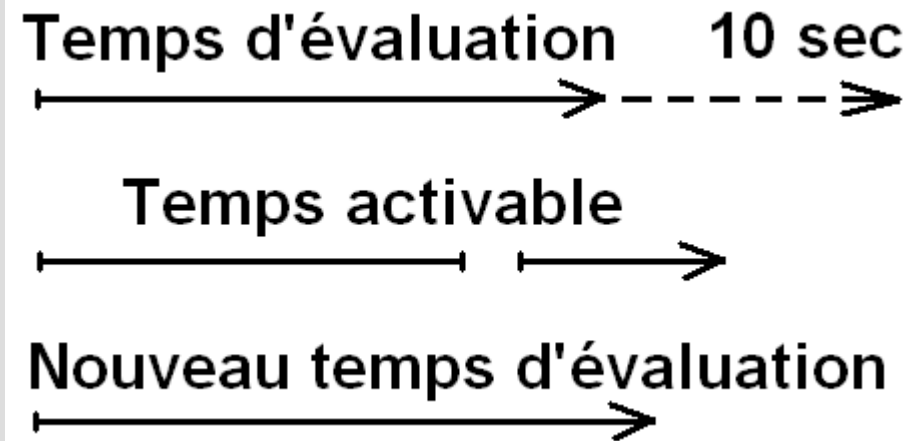
- Mise à jour des Forces
 - calcul dynamique du temps d'évaluation
 - calcul des nouvelles valeurs
- Création de classeurs
- Suppression de classeurs

Calcul dynamique du temps d'évaluation

- Durées trop hétérogènes
 - choix d'action : plusieurs fois par seconde
 - actions rapides : quelques secondes
 - actions lentes : dizaines de secondes
- Période fixe : trop peu fiable et souvent pas efficace
- Donc calcul dynamique avec borne maximale

Calcul dynamique du temps d'évaluation

- Temps de base : 20s
- Rajout max : 10s
- Mesure durée classeur activée durant ce temps
- Calcul nouveau temps :



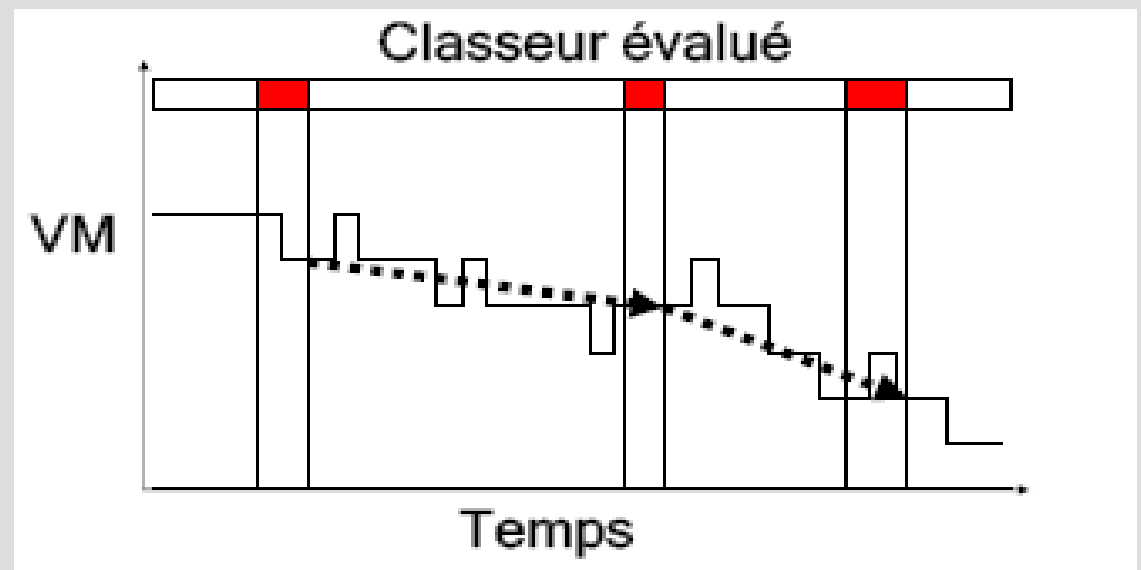
$$TE_x = TE_{x-1} + a \cdot (\text{temps activable} - TE_{x-1})$$

avec „a“ le taux d'apprentissage, fixé par exemple à 0.25

Calcul des Forces

- Mieux vaut se baser sur l'évolution de la VM plutôt que sa valeur à l'instant d'évaluation
- Calculs :
$$\text{nouvelleForce}_x = -1 \cdot (\text{VM}_x - \text{VM}_{x-1}) / (T_x - T_{x-1})$$
$$\text{Force}_x = \text{Force}_{x-1} + a \cdot (\text{nouvelleForce}_x - \text{Force}_{x-1})$$

-1 → force positive
correspond à une
diminuation de la
VM



Création de classeurs

- Un CS devrait potentiellement pouvoir affecter n'importe quelle action via une IAc
- Création de classeurs sans conditions (toujours vrais donc) pour toutes les actions n'ayant pas encore de IAc
- Affectation de leur force à -1
- Evolution de cette force selon la pertinence ou non du classeur

Création de classeurs

- Création de classeurs fictifs ajoutant des conditions supplémentaires
- Mise à jour de ces classeurs si conditions supplémentaires remplies
- Si force du classeur fictif suffisamment grande, alors création d'un nouveau classeur
- Mémorisation des créations pour éviter de créer plusieurs fois le même classeur

Création de classeurs

<i>N_o</i>	<i>TE</i>	<i>Force</i>	<i>Conditions</i>	<i>Action</i>
2	22	1,0 e-4	un membre de l'équipe du JA porte le drapeau = vrai niveau de santé du JA = !haut	Aller à la base de l'équipe
	Spécialisation possible			
	22	1,0 e-4	le JA porte le drapeau = vrai	
	Spécialisation réalisée			
	21	2,6 e-4	le JA porte le drapeau = faux	
6	21	2,6 e-4	un membre de l'équipe du JA porte le drapeau = vrai niveau de santé du JA = !haut le JA porte le drapeau = faux	Aller à la base de l'équipe

Suppression de classeurs

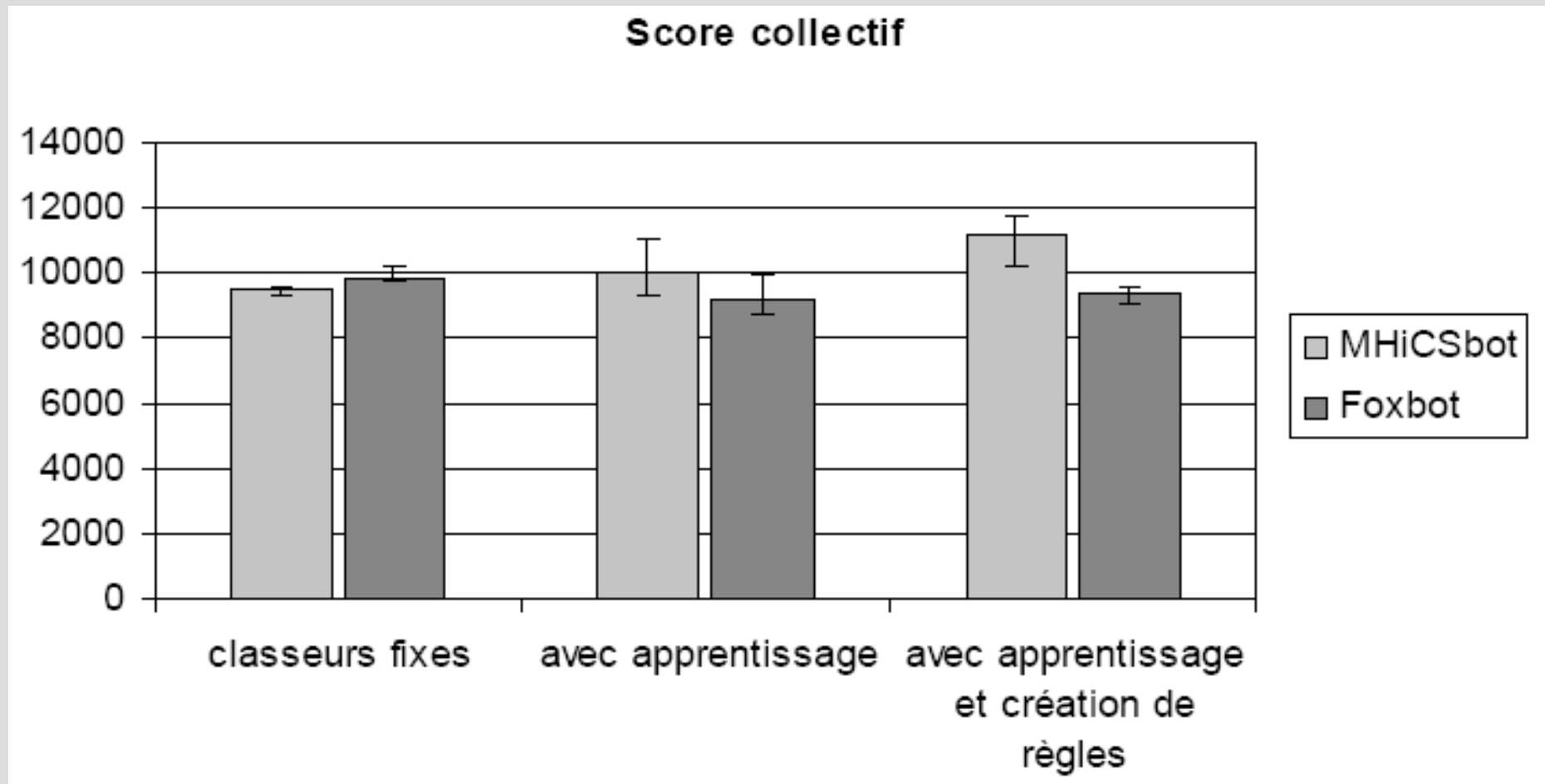
- Supprimer les classeurs avec forces faibles ?
 - sur-spécialisation et disparition des classeurs généraux
- Système de quotas
 - 25% pour forces élevées
 - 75% pour la diversité des actions
- Sur ces 75%
 - 50% pour classeurs avec 1 condition
 - 25% pour classeurs avec 2 conditions
 - 12.5% pour classeurs avec 3 conditions
 - ...

Confrontation

- MHiCS bots contre Foxbots
- Niveau joueur confirmé
- 4 vs 4
- 5 parties de 24h
- (1) Avec motivation : Score Collectif
- (2) Avec motivations : Score Collectif, Score individuel et Survie
- Sans puis avec apprentissage

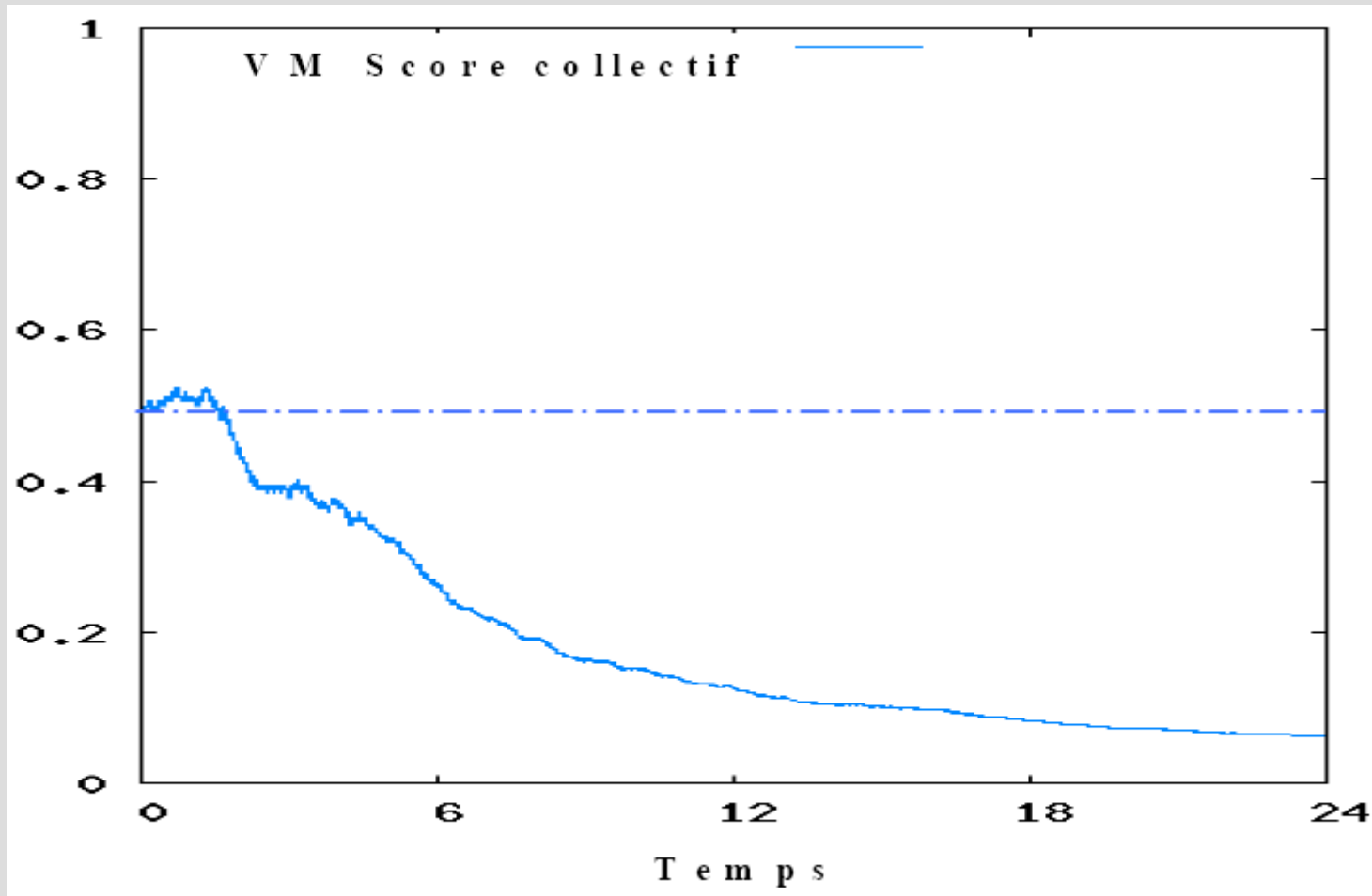
Confrontation (1)

- Résultats :

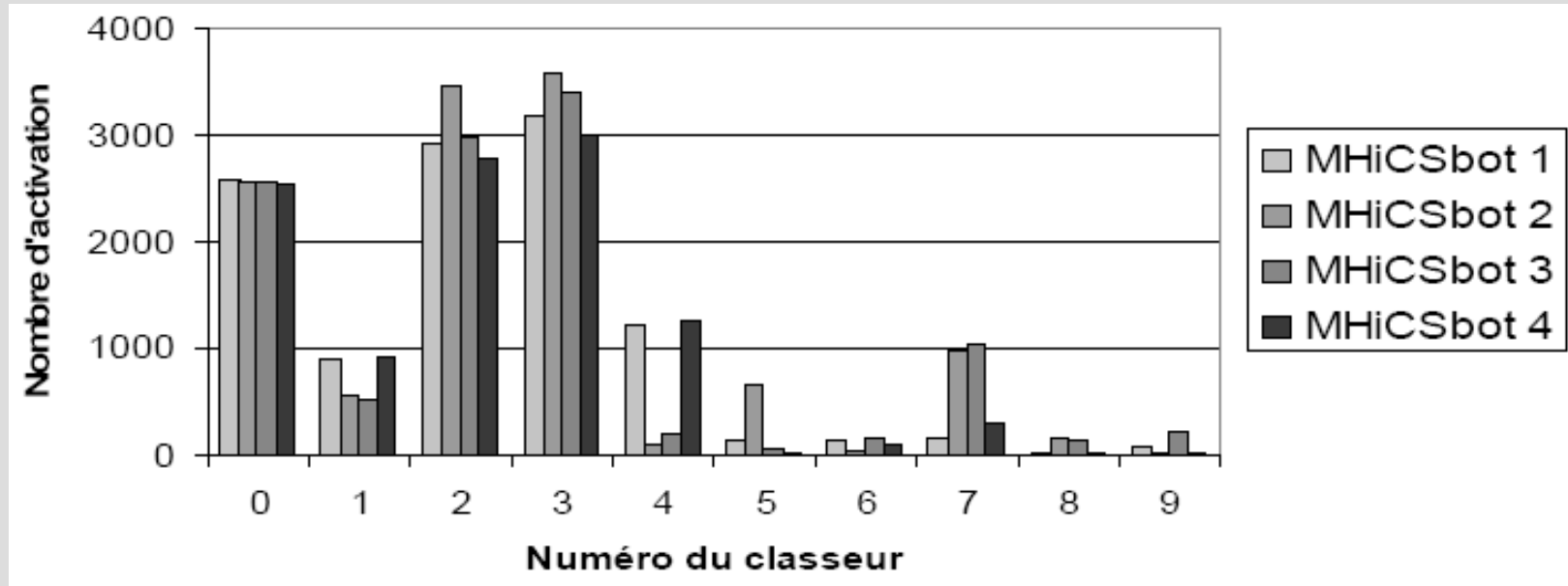


Confrontation (1)

- Evolution de la VM :



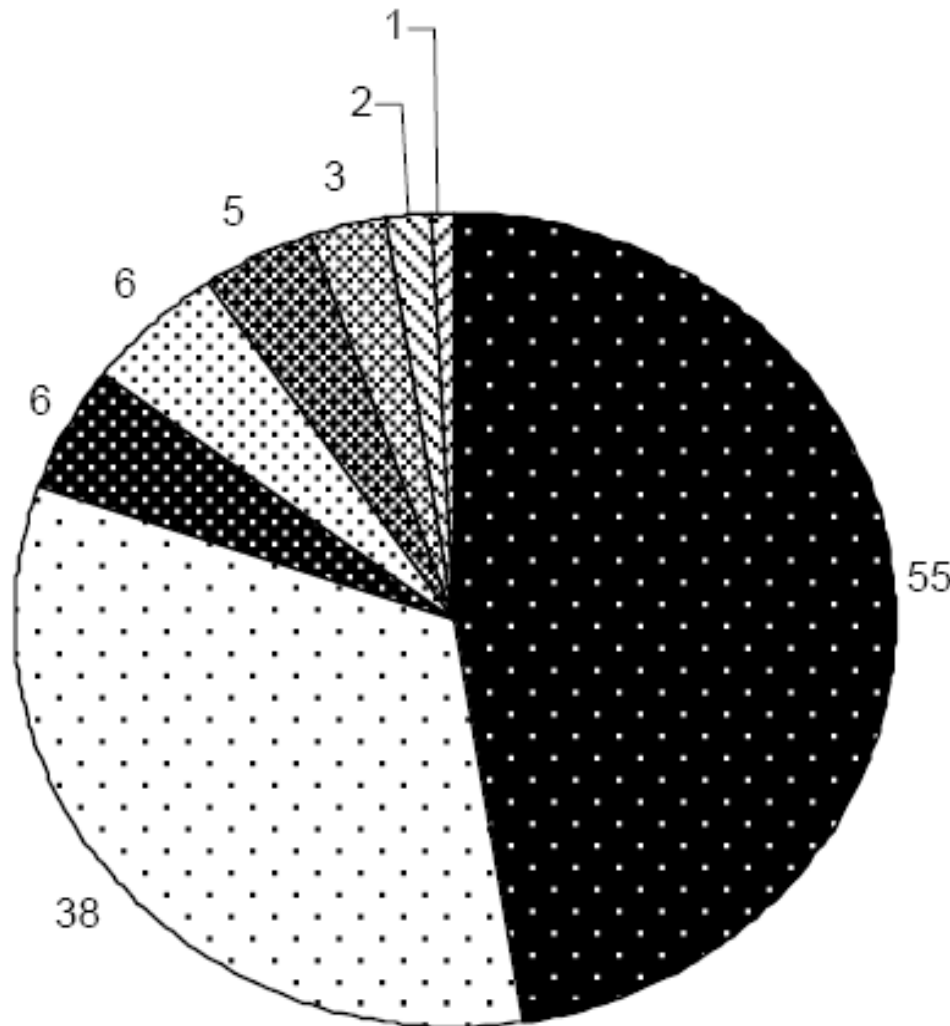
Confrontation (1)



- 1 : Rapporter le drapeau
- 2 et 3 : Regarder et Tirer sur une cible
- 4 : Aller à la salle du drapeau adverse
- 7 : Aller à la salle du drapeau de l'équipe

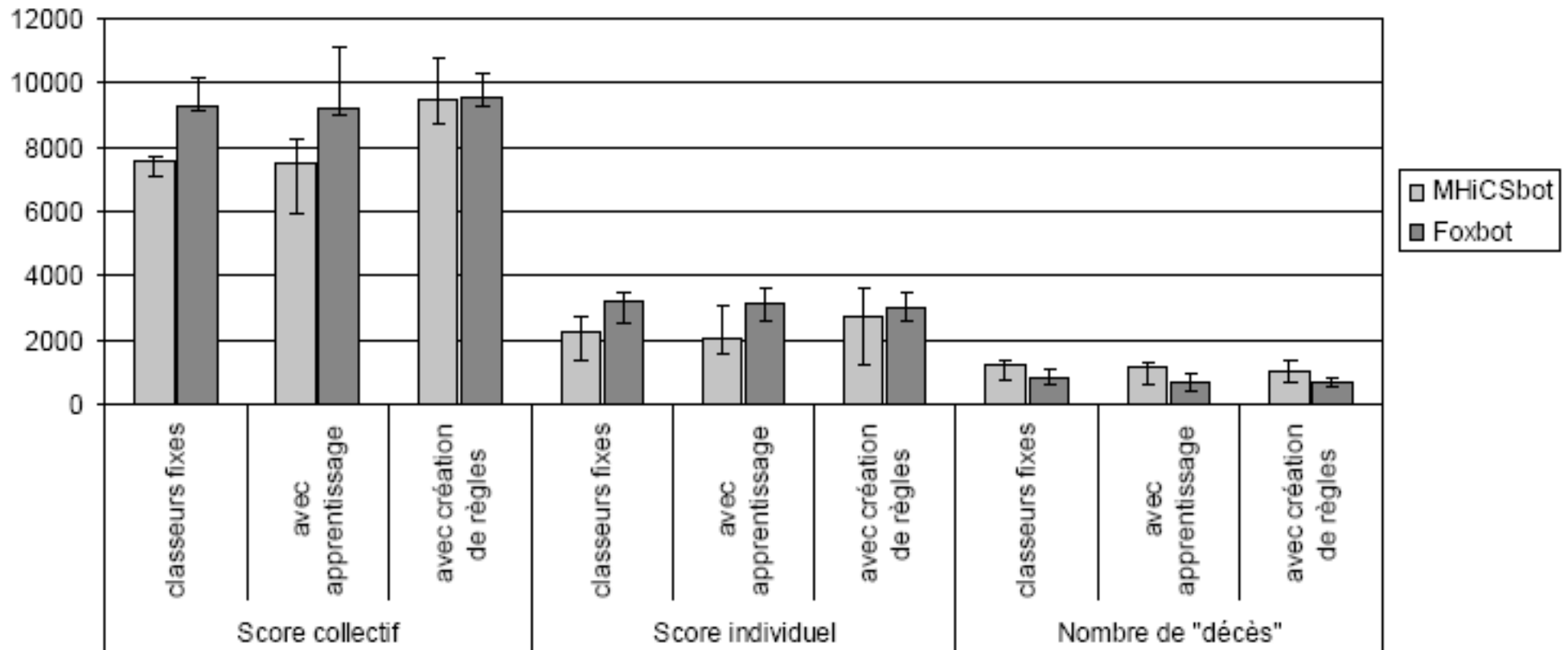
Confrontation (1)

Nombre de classeurs par action

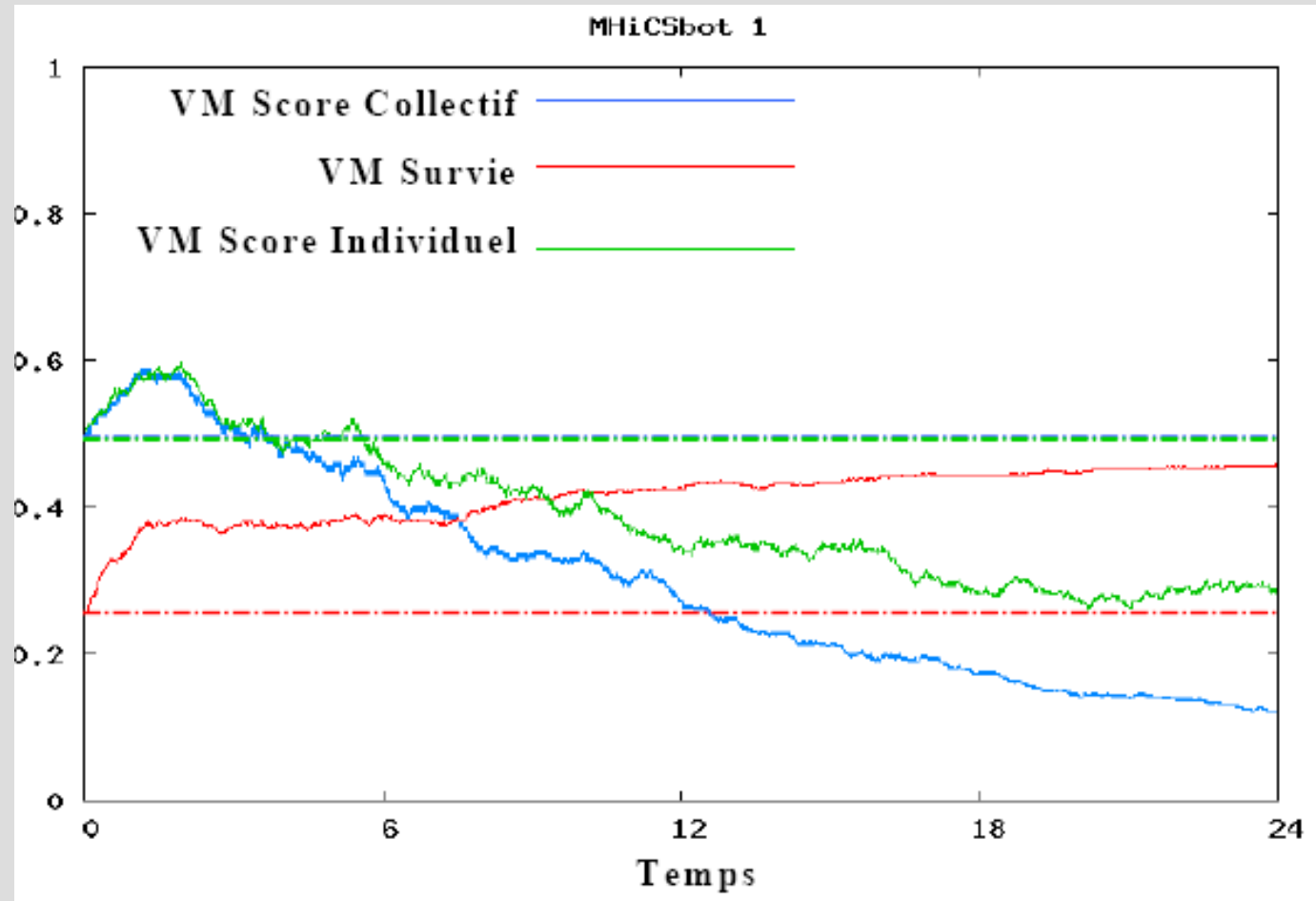


- Regarder la cible
- ▣ Tirer sur la cible
- Aller vers le drapeau ennemi
- ▣ Aller à la base de l'équipe
- ▣ Aller à la salle du drapeau adverse
- ▣ Aller à la base adverse
- ▣ Aller vers le pack de soin le plus proche
- ▣ Aller à la salle du drapeau de l'équipe

Confrontation (2)



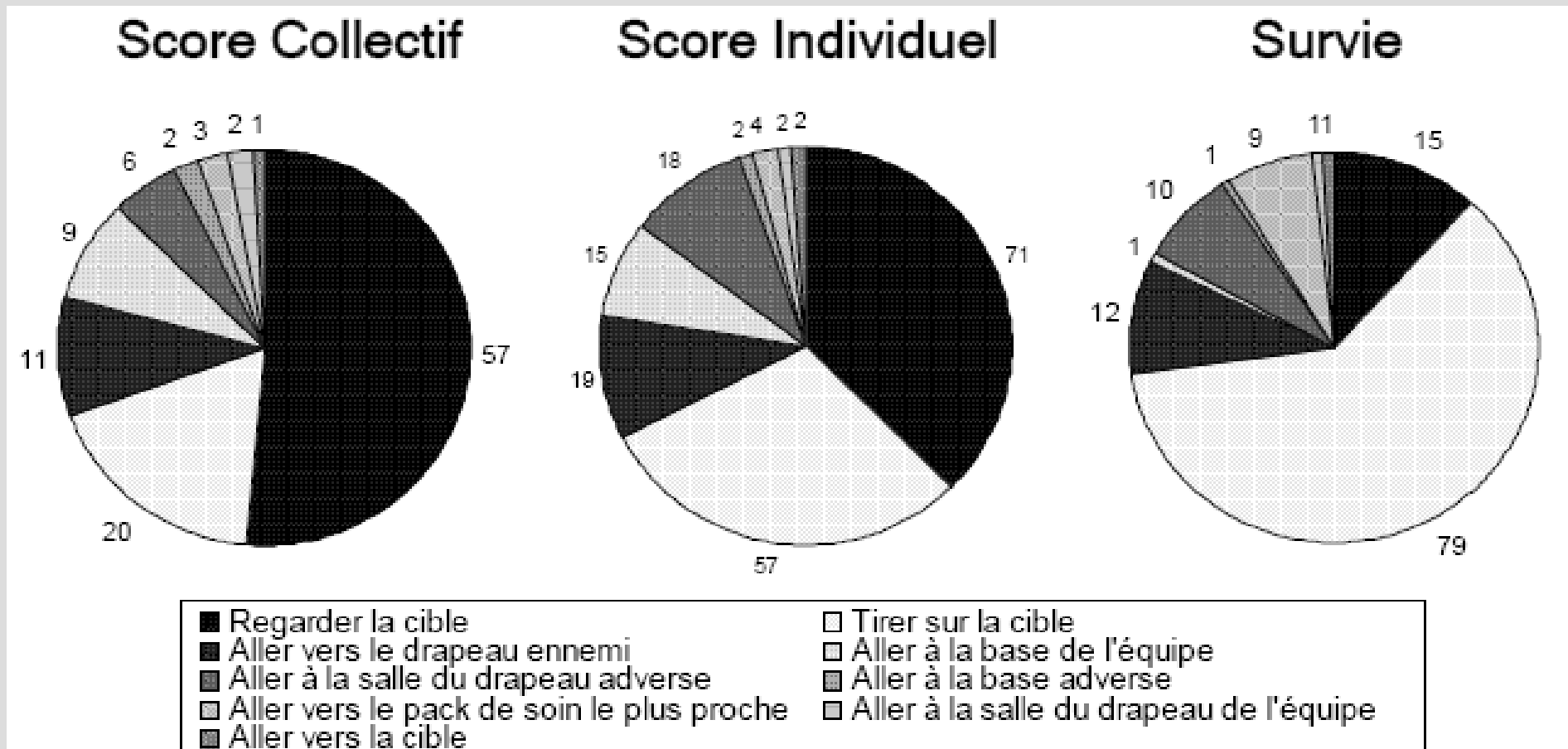
Confrontation (2)



- Meilleur bot pour la meilleure partie

Confrontation (2)

- Nombre de classeurs par motivation :



Conclusion

- Simple
 - Si condition alors action
 - Classement hiérarchique des décisions
- Efficace
 - Souvent bonnes prises de décision
- Modulaire et extensible
 - Ajout / retrait de motivations, de classeurs
 - Nouvelles conditions, perceptions
 - Nouvelles actions
- Adaptation par l'apprentissage

Conclusion

- Limitations
 - Nombre de facteurs dans les conditions
 - Explosion de mémoire si trop de classeurs fictifs pour la création
 - Temps processeur critique si trop de bots avec trop de motivations / CS / perceptions
 - Parfois, conflit des motivations

Questions

- Tout est dans le titre...