Estimations de la position d'un agent mobile dans un environnement bornée grâce aux modèles de Markov cachés

Cédric Beaulac

Université du Québec à Montréal

2 Juillet 2015

Introduction

- Estimation de la position inconnue d'un agent en déplacement dans un environnement bornée.
- Utilisation des modèles de Markov cachés et des modèls de semi-Markov cachés pour modéliser et résoudre cette problématique
- Mise au point d'une intelligence artificielle capable d'effectuer cette tâche de façon autonome.

Plan de le présentation

Intelligence artificielle et jeu vidéo

Intelligence artificielle *IA*

- Construction de programmes informatiques qui peuvent reproduire un comportement humain
- Interdisciplinaire : Psychologie, Mathématique, Informatique et autres
- Dans plusieurs environnement, incluant les jeux vidéo

Jeu vidéo

- Terminologie : Avatar, *bot*, Jeux compétitifs (*PvP*), Carte de jeu.
- Actuellement, l'intelligences artificielle n'est tout simplement pas intelligente ou bien elle triche.
- Créer une *IA* qui peut performer dans un contexte de jeu vidéo compétitif. Immiter le processus d'estimation d'un être humain. Performer sans tricher.

Jeu vidéo : Carte de jeu

Figure: Carte de jeu de *League of Legends*

Modèle de Markov caché (HMM)

- Une variable markovienne standard sous-jacente
- Les observations ne sont pas les réalisation de la variable de Markov
- Les observations sont une variable aléatoire dépendante de la variable markovienne
- Les observations peuvent être de n'importe quelle distribution
- L'inférence se fait avec avec les observations

Modèle de Markov caché : Formalisme

- Variable de Markov : q_t, variable aléatoire q et indice t
- Espace de états : $S = \{s_1, s_2, ..., s_n\}$
- Probabilité de transition $a_{i,j} = P[q_t = s_j | q_{t-1} = s_i]$
- Matrice de transition $\mathbf{A}$
- Distribution initiale μ où $\mu_i = P[q_1 = s_i]$

Modèle de Markov caché : Formalisme

- Variable observée : X_t
- Loi d'observation : $b_i(x_t) = P[X_t = x_t | q_t = s_i]$
- Si les observations sont de poisson : $b_i(x_t) = e^{-\lambda_i} \lambda_i^{x_t} / x_t!$
- Séquence d'états visités : $q_1, q_2, ..., q_T$
- Séquence d'observations : $x_1, x_2, ..., x_T$

Modèle de Markov caché

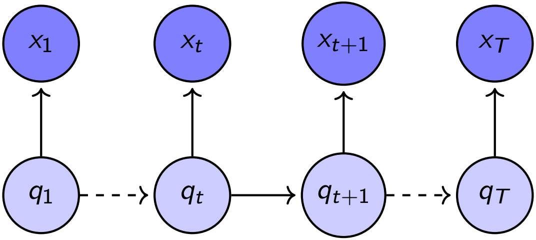
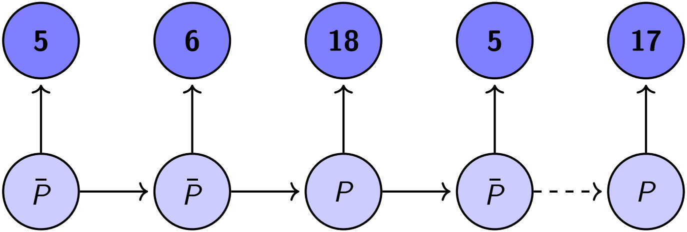


Figure: Illustration de T réalisations d'une chaîne de Markov caché. Notons que seuls les x_t sont observés. Les éléments pâles ne sont pas observés.

Modèle de Markov caché : Exemple

- Variable markovienne q_t : température
- $S = \{P, \bar{P}\}$
- Observations x_t : Nombre de parapluies vendus par un kiosque.
- Les observations sont de poissons $\lambda_P = 20$ et $\lambda_{\bar{P}} = 5$
- On veut faire de l'inférence sur la température en utilisant le nombre de parapluies vendus.

Modèle de Markov caché : Exemple



Modèle de Markov caché : Inférence

- Algorithme *forward* permet d'évaluer la valeur $P[x_1, x_2, x_3, ..., x_t \text{ et } q_t = s_i] \ \forall t$ et $\forall s_i$
- Coefficients $\alpha_t(i) = P[x_1, x_2, x_3, ..., x_t \text{ et } q_t = s_i]$
- $P[x_1, x_2, ..., x_t] = \sum_i P[x_1, x_2, ..., x_t \text{ et } q_t = s_i] = \sum_i \alpha_t(i)$
- $P[q_t = s_i | x_1, x_2, ...x_t] = \frac{P[q_t = s_i \text{ et } x_1, x_2, ...x_t]}{P[x_1, x_2, ..., x_t]} = \alpha_t(i) / \sum_i \alpha_t(i)$

Modèle de Markov caché : Inférence

- Algorithme *forward* :

$$\begin{aligned}\alpha_{t+1}(j) &= P(x_1, x_2, ...x_t, x_{t+1} \text{ et } q_{t+1} = s_j) \\ &= \sum_{i}^{n} P(x_1, x_2, ...x_t, x_{t+1} \text{ et } q_t = s_i \text{ et } q_{t+1} = s_j) \\ &= \sum_{i}^{n} \alpha_t(i) a_{ij} b_j(x_{t+1}).\end{aligned}$$

- Valeurs initiales : $\alpha_1(i) = \mu_i b_i(x_1)$

Modèle de Markov caché : Inférence

- Algorithme *backward* :

$$\begin{aligned}
\beta_t(i) &= P(x_{t+1}, x_{t+2}, ..., x_T | q_t = s_i) \\
&= \sum_{j=1}^{N} P(x_{t+1}, x_{t+2}, ..., x_T \text{ et } q_{t+1} = s_j | q_t = s_i) \\
&= \sum_{j=1}^{N} a_{ij} b_j(x_{t+1}) \beta_{t+1}(j).
\end{aligned}$$

- Valeurs initiales : $\beta_T(i) = 1$

Modèle de Markov caché : Inférence

- Algorithme de *Baum-Welch*
- Algorithme *EM* pour HMM
- Permet l'estimation de tous les paramètres des HMM par maximisation de la vraisemblance

Modèle de Markov caché : Inférence

- Calcul des valeurs attendues (*Expected values*) :
- $\gamma_t(i) = P[q_t = S_i | x_1, x_2, ..., x_T] = \alpha_t(i)\beta_t(i)/V_T$.
- $\xi_t(i,j) = P[q_{t-1} = s_i, q_t = s_j | x_1, x_2, ..., x_T] = \alpha_{t-1}(i)a_{i,j}b_j(x_t)\beta_t(j)/V_T$.

- En maximisant la vraisemblance ($V(x, q)$), on obtient :

$$
\begin{aligned}
\hat{a}_{i,j} &= \frac{\sum_{t=2}^{T} \xi_t(i,j)}{\sum_j^n \sum_{t=2}^{T} \xi_t(i,j)} \\
&= \frac{\sum_{t=2}^{T} \xi_t(i,j)}{\sum_{t=2}^{T} \gamma_t(i)} \\
&= \frac{\sum_{t=2}^{T} P[q_{t-1} = s_i, q_t = s_j | x_1, x_2, ..., x_T]}{\sum_{t=2}^{T} P[q_t = s_i | x_1, x_2, ..., x_T]}.
\end{aligned}
$$

- On peut aussi obtenir des estimateurs pour μ et b.

Modèle de Markov caché : Inférence

- En maximisant la vraisemblance ($V(x,q)$), on obtient :

$$
\begin{aligned}
\hat{a}_{i,j} &= \frac{\sum_{t=2}^{T} \xi_t(i,j)}{\sum_{j}^{n} \sum_{t=2}^{T} \xi_t(i,j)} \\
&= \frac{\sum_{t=2}^{T} \xi_t(i,j)}{\sum_{t=2}^{T} \gamma_t(i)} \\
&= \frac{\sum_{t=2}^{T} P[q_{t-1} = s_i, q_t = s_j | x_1, x_2, ..., x_T]}{\sum_{t=2}^{T} P[q_t = s_i | x_1, x_2, ..., x_T]}.
\end{aligned}
$$

- On peut aussi obtenir des estimateurs pour μ et b.

Modèle de semi-Markov caché (HSMM)

Modèle de semi-Markov : Introduction

- Généralisation du temps demeuré à chaque états.
- Habituellement géométrique de paramètre : $1 - a_{i,i}$.
- Maintenant deux variables : Variable markovienne q_t et la variable de durée l_t.
- Différent d'une chaîne de Markov à temps continue.

Modèle de semi-Markov caché : Introduction

- Demeure l_t unité de temps à l'état sous-jacente s_i.
- l_t observations provenant de s_i.
- Plus d'informations mais plus de paramètre à estimer.
- Notation très complexe.

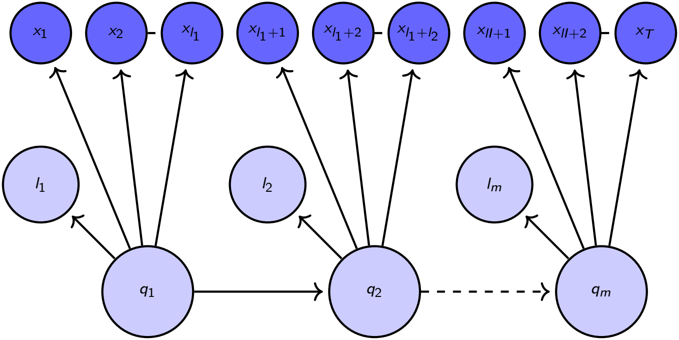


Figure: Illustration d'une séquence d'observations d'une chaîne de semi-Markov cachée. Ici, m variables markoviennes furent réalisées et T observations furent obtenues. De plus, $ll = \sum_1^{m-1} li$. Les éléments pâles ne sont pas observés.

Modèle de semi-Markov caché : Inférence

- Nous travaillerons avec un cas particulier, un modèle de Markov caché à durée explicite : $P(s_i, l|s_j, l_{t_p}) = P(s_i|s_j)P(l|s_i)$
- Inférence avec *forward*, *backward* et *Baum-Welch*.
- Modifications importantes de ces algorithmes

Modèle de semi-Markov caché : Inférence

- $\alpha_t(s_i) = P(q_t = s_i, F_t = 1, x_{1:t}) = \sum_l P(x_{t-l_t+1:t}|s_i, l_t)P(l|s_i)\sum_{j\neq i} P(s_i|s_j)\alpha_{t_p}(s_j)$.
- $\beta^*_t(s_j) = P(x_{t+1:T}|q_{t+1} = s_j, F_t = 1) = \sum_{l_{t_n}} P(x_{t+1:t+l_{t_n}}|q_{t_n}, L_{t_n})\beta_{t_n}(s_j)P(l_{t_n}|s_j)$..
- *Baum-Welch* : $\hat{a}_{i,j} = \frac{\sum_{t=2}^{T} \alpha_t(i)a_{i,j}\beta^*_t(j)}{\sum_j^n \sum_{t=1}^{T} \alpha_t(i)a_{i,j}\beta^*_t(j)}$

Modélisation de la problématique

- La position de l'agent sera l'état
- Elle est inconnu, donc caché.
- Problème d'estimation de l'état.
- Nous aurons des observations dépendantes de l'état.
- Contexte : Intelligence artificielle dans un jeu vidéo

Modélisation : Formalisme

- Quadriller la carte de jeu. L'ensemble de ces n cases formera l'espace des états S.
- Vecteur de probabilité initiale connue de tous. Implémenter directement dans le jeu.
- Probabilité de transition $a_{i,j}$: Probabilité de se déplacer d'une case à l'autre

Modélisation : Probabilités de transition

- Représente en quelque sorte les connaissances sur l'agent mobile.
- Inconnu, peut être estimé de plusieurs manières.
- Estimé empiriquement par Hladky (Hladky, 2009).
- Partit très importante de l'estimation.

Modélisation : Fonction d'observation

- L'avatar reçoit de l'information de son environnement.
- Si nous le voyons directement, pas d'estimation.
- Si nous le voyons pas, nous gagnons de l'information tout de même.
- Défi de modélisation de ces observations

Modélisation : Champs de vision

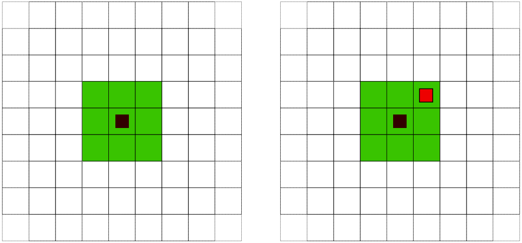


Figure: Exemple d'un champ de vision. Classique dans plusieurs jeux vidéo.

Modélisation : Fonction d'observation

- Notons W, l'ensemble des cases observées (les cases vertes).
- $b_i(x_t) = P[X_t = x_t | q_t = s_i] = 1 - \mathbf{1}[s_i \in W]$.
- Probabilité d'observer qu'il n'est pas là sachant qu'il est la $=$ 0.
- Dans la situation d'intérêt, nous n'observons par l'agent. Dans ce cas, $s_i \in W$ alors $b_i(x_t) = 0$.

Modélisation : Objectif

- Estimé en temps réelle la position de l'agent mobile à l'intérieur de l'espace bornée.
- Apprendre au fil du temps le comportement de l'agent pour mieux estimer sa position.

- Utilisation de l'algorithme *forward* :

$$
\begin{aligned}
P[q_t = s_i | x_1, x_2, ...x_t] &= \frac{\alpha_t(i)}{\sum_i \alpha_t(i)} \\
&= \frac{b_i(x_t) \sum_j^N \alpha_{t-1}(j) a_{ji}}{\sum_i b_i(x_t) \sum_j^N \alpha_{t-1}(j) a_{ji}}.
\end{aligned} \tag{1}
$$

- Souvenons-nous : $s_i \in W$ alors $b_i(x_t) = 0$

Modélisation : Estimation en temps réelle

- L'important, c'est l'élimination des chemins.
- Rappellons : $\alpha_{t+1}(j) = b_j(x_{t+1}) \sum_i^N \alpha_t(i) a_{i,j}$
- Souvenons-nous : $s_i \in W$ alors $\alpha_t(i) = 0$. Nous simplifions donc le calcul des sommations.
- Nous éliminons tous les chemins qui passent par s_i au temps t.
- Grande force de cette méthodologie.

Modélisation : Apprentissage automatique

- Comme les observations ne font qu'éliminer progressivement certaines cases.
- La matrice de transition représente une importante partie de l'estimation en temps réelle
- Nous voulons que l'intelligence artificielle s'adapte à son adversaire comme le ferais un joueur humain.
- Une *IA* sans apprentissage est facile à battre.

Modélisation : Apprentissage automatique

- Nous allons utiliser *Baum-Welch* pour estimer la matrice de transition en fonction de nos observations.
- Nous aurons de multiples séquences d'observations.
- Supposons que les m parties que nous jouer contre notre adversaires sont indépendantes.

$$\hat{a}_{i,j} = \frac{\sum^{m} \sum_{t=2}^{T} \xi_t(i,j)}{\sum^{m} \sum_{j}^{n} \sum_{t=2}^{T} \xi_t(i,j)}.$$

Modélisation : Apprentissage automatique

- Afin de garder une certaine flexibilité, nous utilisons une matrice hybride, ne considérant que quelques des dernières parties.
- Une partie de la matrice sont des connaissances générale (20%).
- L'autre partie est la matrice estimé à partir des 10 dernières parties. (80%).
- Nous avons ainsi développer un *bot* qui ne perderas pas de la même manière plusieurs parties de suite.

Algorithmie et programmation

Algorithmie : Choix des algorithmes

- Algorithmes de Markov caché
- Aucune raison de ne pas croire en l'hypothèse de la géométrique. Aucune case particulière, pas de stratégie déjà établis.
- Algorithmes plus rapides d'exécution.
- Nettement plus simple à programmer.

Algorithmie : Défi de programmation

- α et β sont des produits de probabilités. Converge rappidement vers 0.
- Série de tranformation à effectuer pour empêcher la convergence et pour récupérer les valeurs originales.
- Calcul de la vraisemblance complexe.

Programmation : Objectif

- Programmer une intelligence artificielle qui estime la position d'un agent mobile et qui apprend de ces expériences passés.
- Développer un jeu vidéo nous permettant de mettre à l'épreuve notre intelligence artificielle.

Programmation : base de l'environnement de test

- Jeu où l'estimation de la position de son adversaire était d'une grande importance.
- Carte quadrillé avec positions discrètes.
- Jeu tour par tour.
- Un déplacement par tour sur les cases adjacentes.
- Deux joueurs en opposition.
- Le joueur humains veut atteindre un objectif, le *bot* veut l'en empêcher en le touchant.

- Ne triche pas.
- Débute avec un ensemble d'information que tout joueur possèdent : position initiale et règlements du jeu.
- Performe bien et apprends.

- Après le tour du joueur, elle obtient les observations de sont environnement et estime les probabilités de chaque position selon l'algorithme *forward*.
- La position la plus probable est l'estimation de la position de son adversaire.
- Détermine la distance entre cette position et les case qui l'entour grâce à l'algorithme de Dijkstra.
- En cas d'égalité, elle choisi la case la plus révélatrice.

Intelligence artificielle : Apprentissage

- À la fin de la partie, elle prend compte de l'état finale du match
- Ajoute cette partie à ses connaissances.
- Estime la matrice de transition à l'aide de *Baum-Welch* à partir de quelques-unes des dernières parties.

Expérimentations et résultats

Expérimentations : environnement de test

- Jeu équitable.
- Pas de victoire trivial.
- Motive la création de stratégie

Expérimentations : environnement de test

Figure: Exemple d'un champ de vision. Classique dans plusieurs jeux vidéo.

Expérimentations : statistique de test

1. Évaluer la précision de l'estimation.
2. Distance entre l'estimation de la position de l'agent et sa véritable position.
3. Distance moyenne pour une partie complète.

Expérimentations : Paramètres

Comparer notre statistique de test pour plusieurs intelligences.

1. Intelligence sans connaissance et sans apprentissage.
2. Intelligence avec connaissance et sans apprentissage (Hladky).
3. Intelligence avec connaissance et avec apprentissage (notre intelligence).

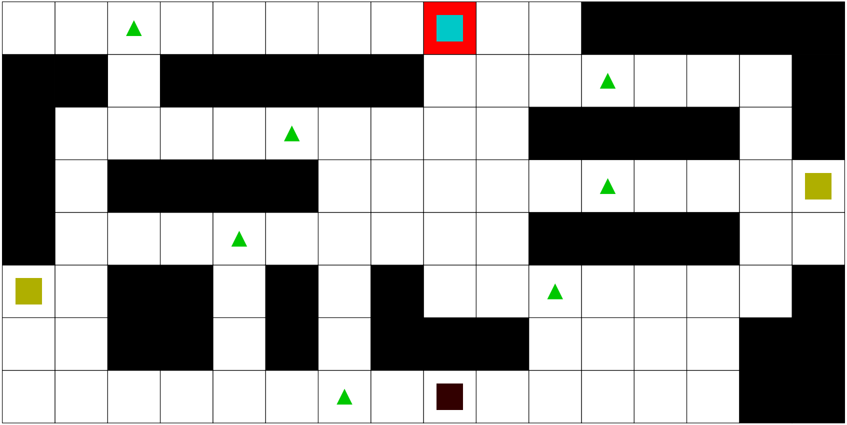
Expérimentations : Paramètres

1. Dix parties avec exactement la même stratégie.
2. Huit parties avec une stratégie. Puis changement de stratégie pour huit autres parties.
3. Alternance entre deux stratégie pendant douze parties.

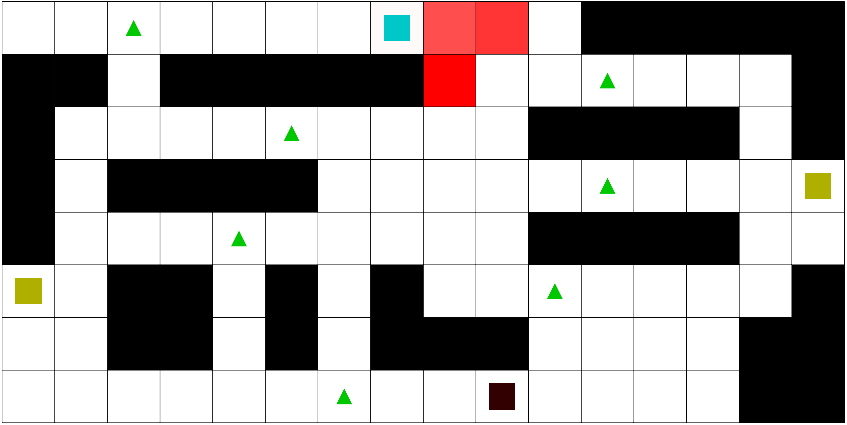
Résultats : Carte de chaleur

1. Représentation de la carte de jeu.
2. Affiche les probabilités estimés par un dégradé de rouge.
3. Nous aide a visualiser l'estimation en temps réelle et l'apprentissage.

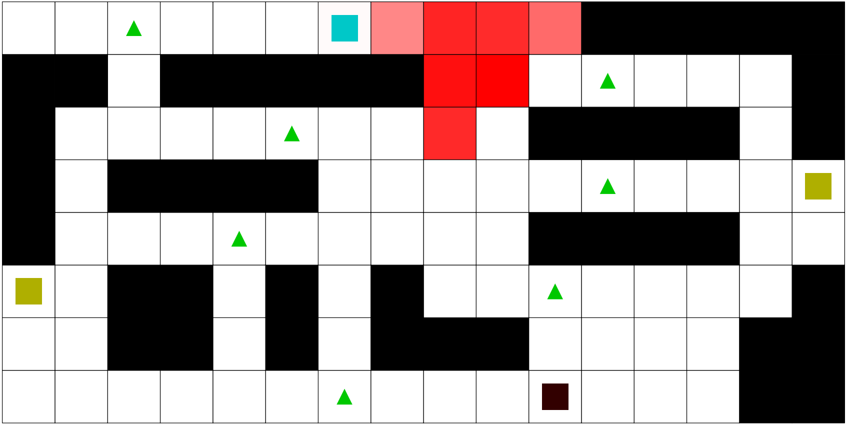
Carte de chaleur : Stratégie A, Partie # 1, Tour # 1

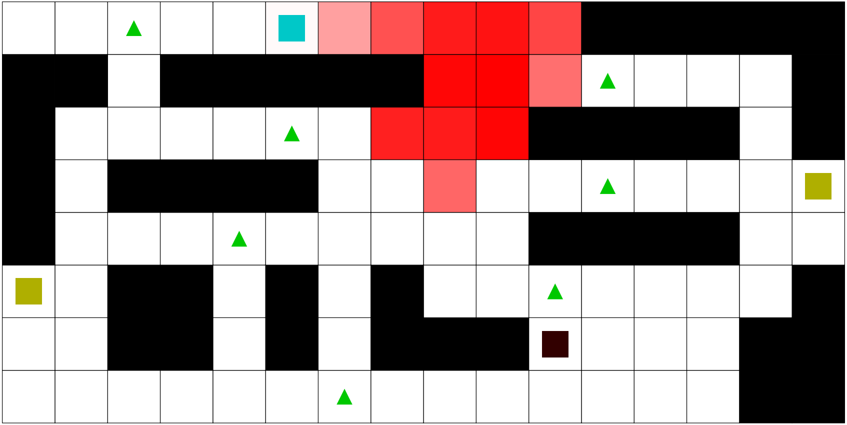


Carte de chaleur : Stratégie A, Partie # 1, Tour # 2

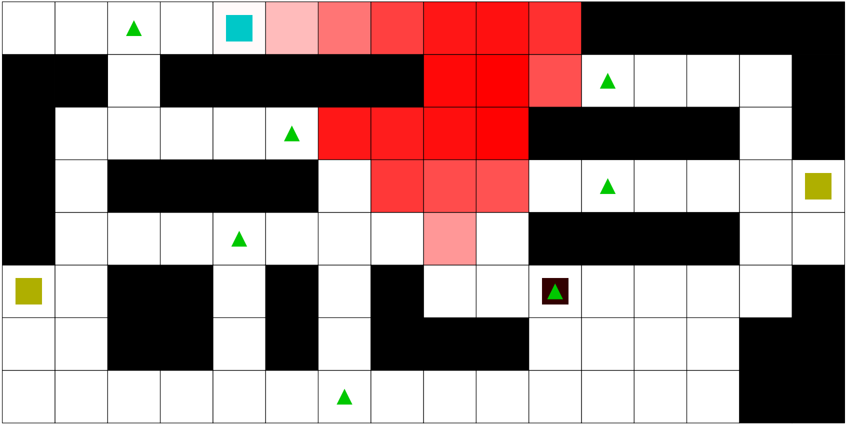


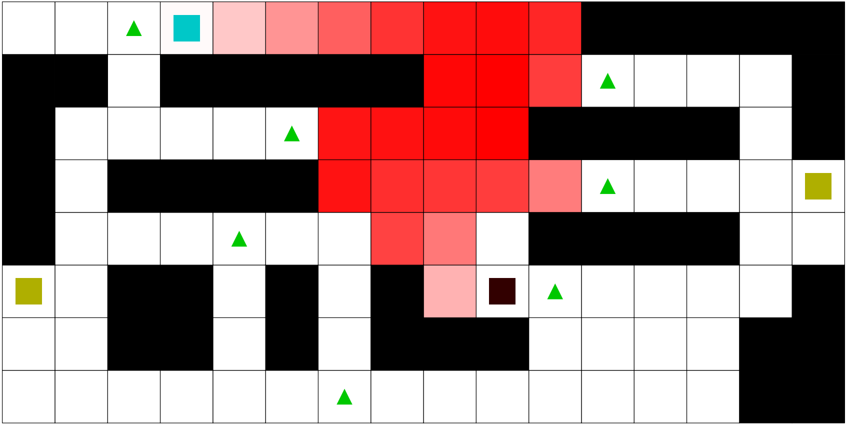
Carte de chaleur : Stratégie A, Partie # 1, Tour # 3



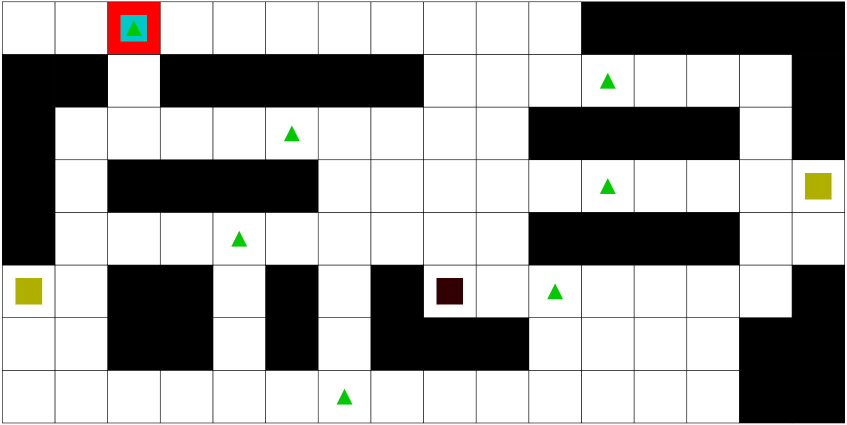


Carte de chaleur : Stratégie A, Partie # 1, Tour # 5

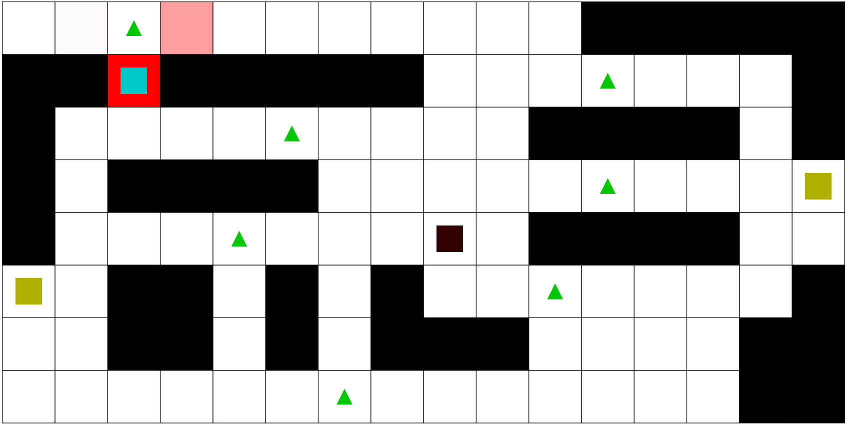




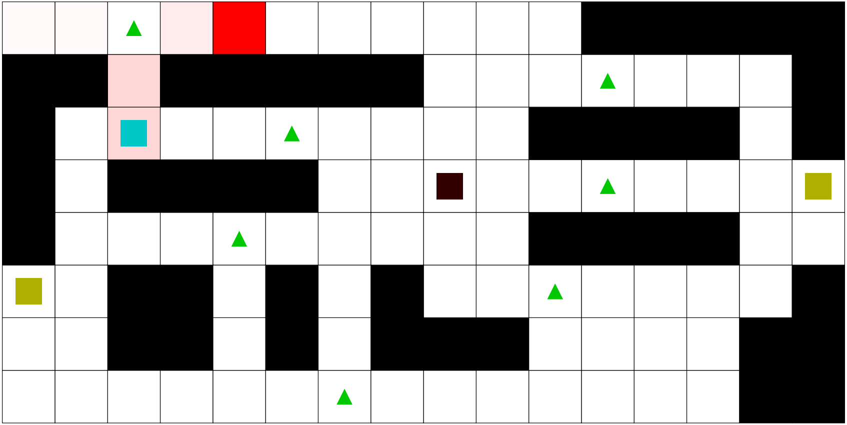
Carte de chaleur : Stratégie A, Partie # 1, Tour # 7



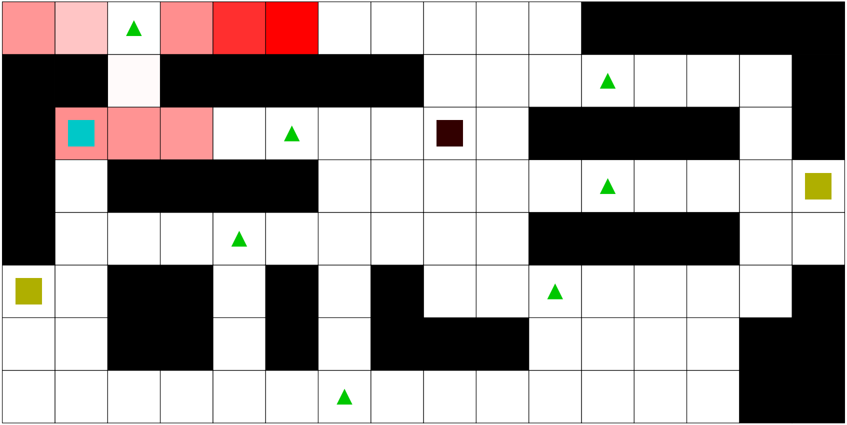
Carte de chaleur : Stratégie A, Partie # 1, Tour # 8



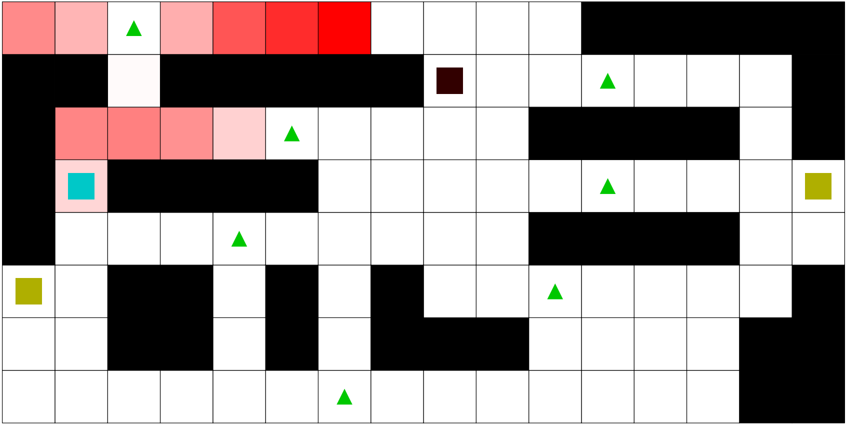
Carte de chaleur : Stratégie A, Partie # 1, Tour # 9



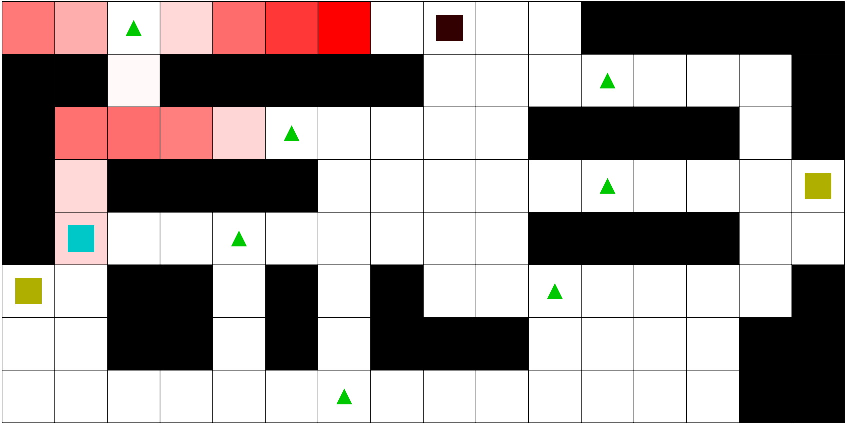
Carte de chaleur : Stratégie A, Partie # 1, Tour # 10



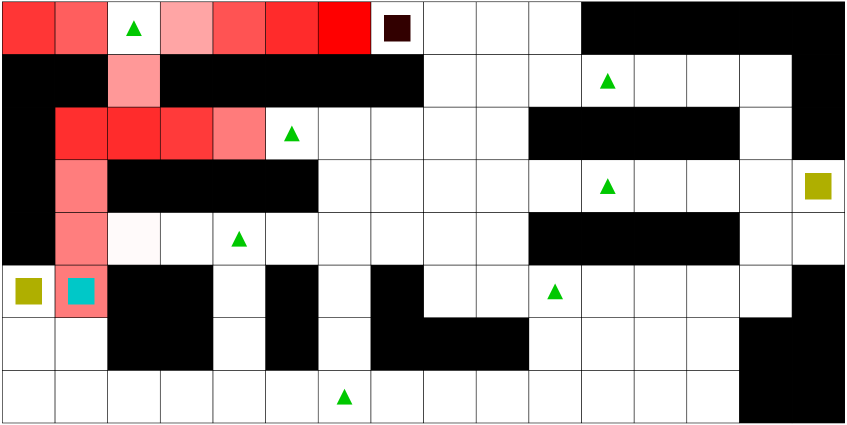
Carte de chaleur : Stratégie A, Partie # 1, Tour # 11



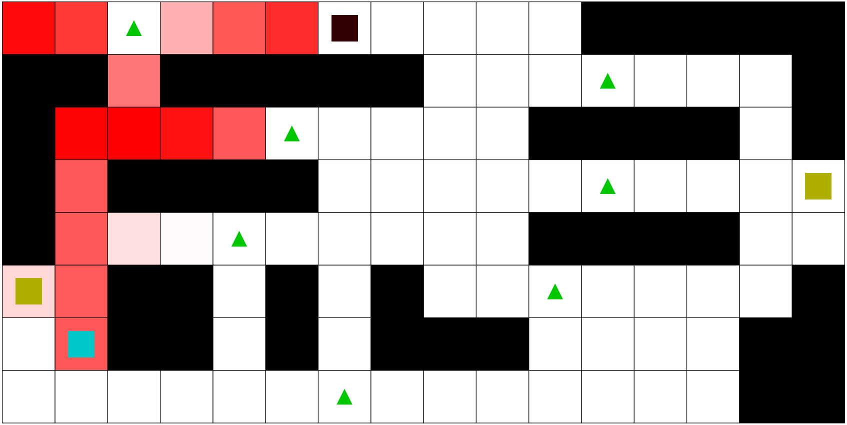
Carte de chaleur : Stratégie A, Partie # 1, Tour # 12



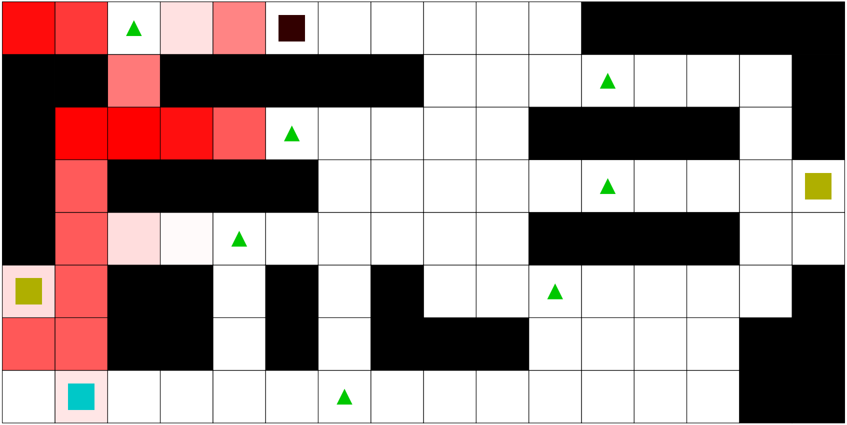
Carte de chaleur : Stratégie A, Partie # 1, Tour # 13



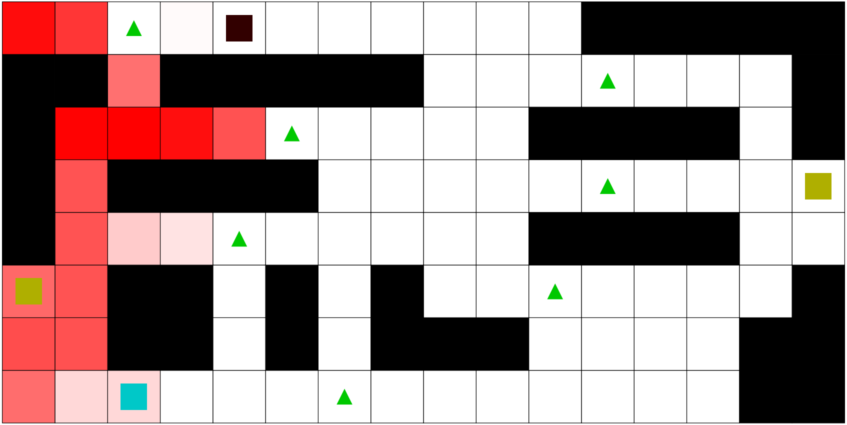
Carte de chaleur : Stratégie A, Partie # 1, Tour # 14



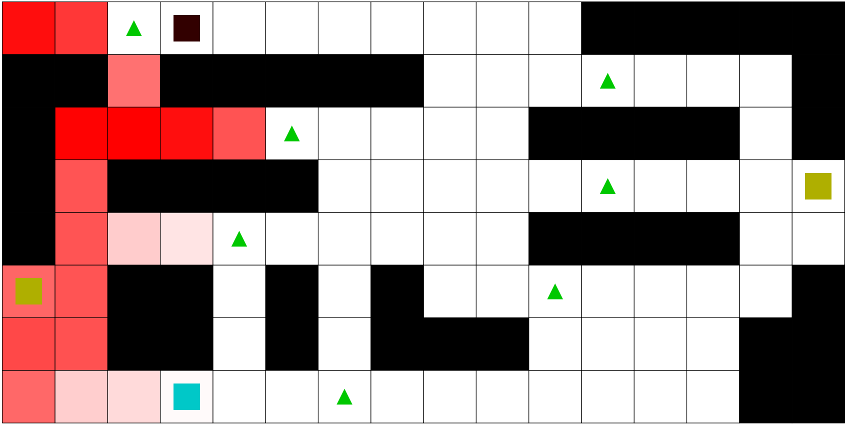
Carte de chaleur : Stratégie A, Partie # 1, Tour # 15



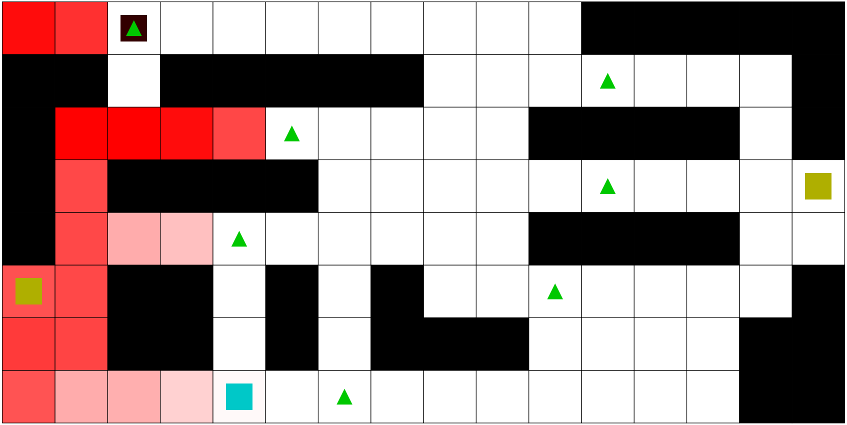
Carte de chaleur : Stratégie A, Partie # 1, Tour # 16



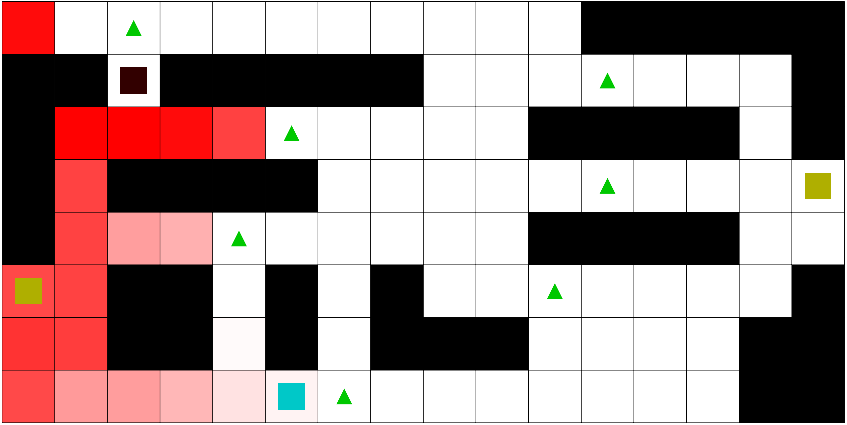
Carte de chaleur : Stratégie A, Partie # 1, Tour # 17

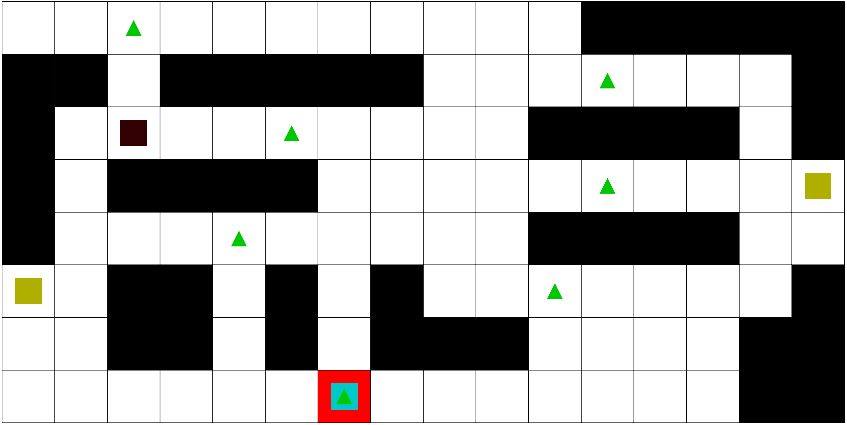


Carte de chaleur : Stratégie A, Partie # 1, Tour # 18

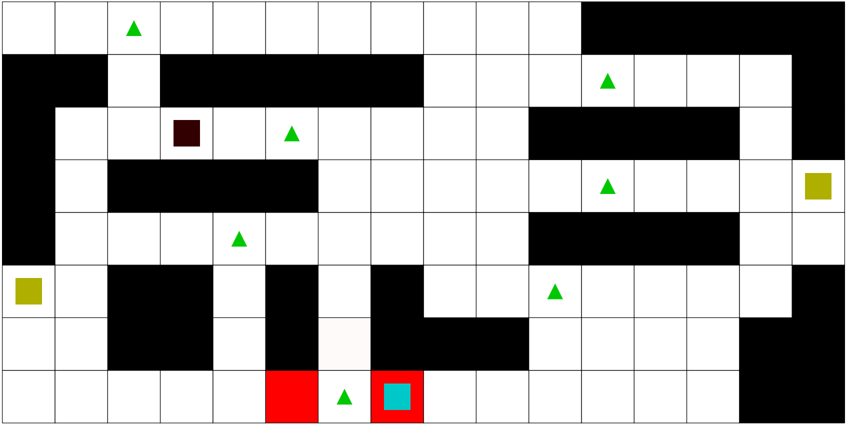


Carte de chaleur : Stratégie A, Partie # 1, Tour # 19

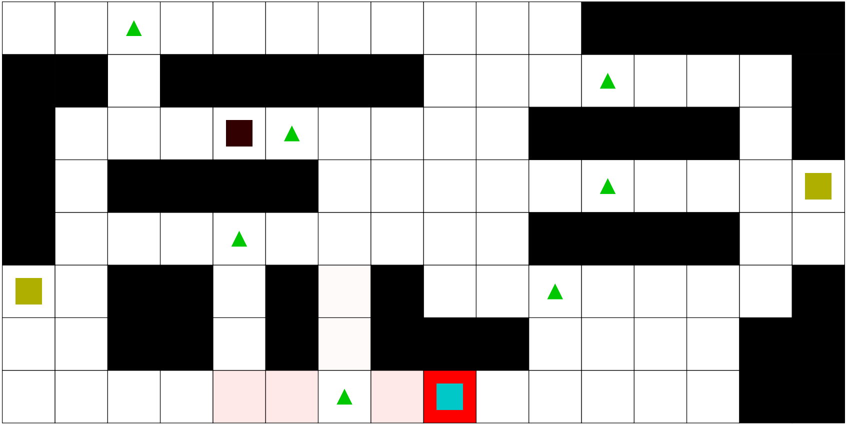




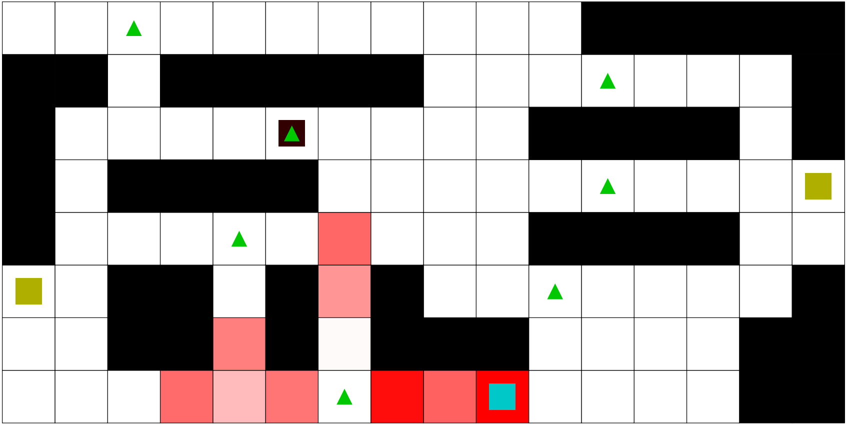
Carte de chaleur : Stratégie A, Partie # 1, Tour # 21



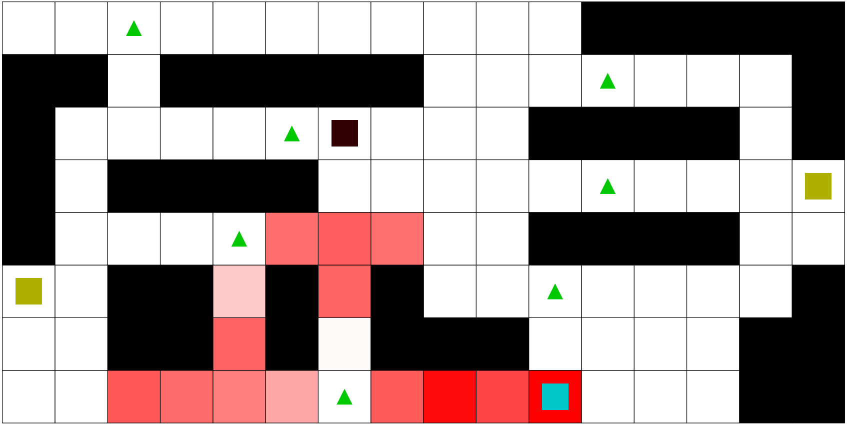
Carte de chaleur : Stratégie A, Partie # 1, Tour # 22



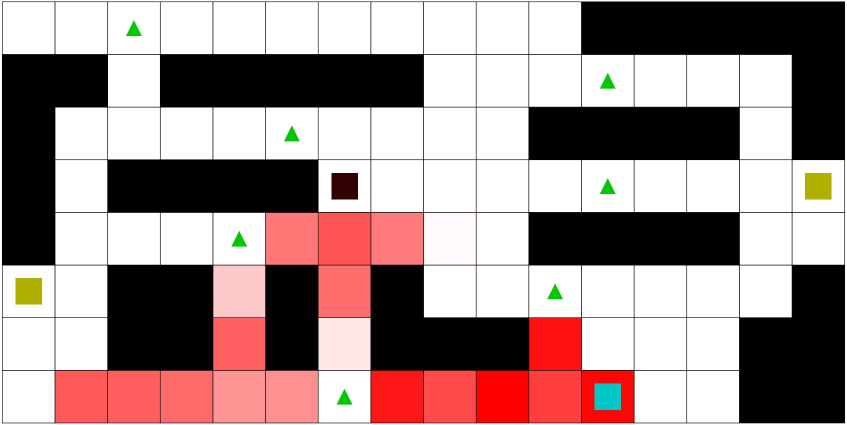
Carte de chaleur : Stratégie A, Partie # 1, Tour # 23



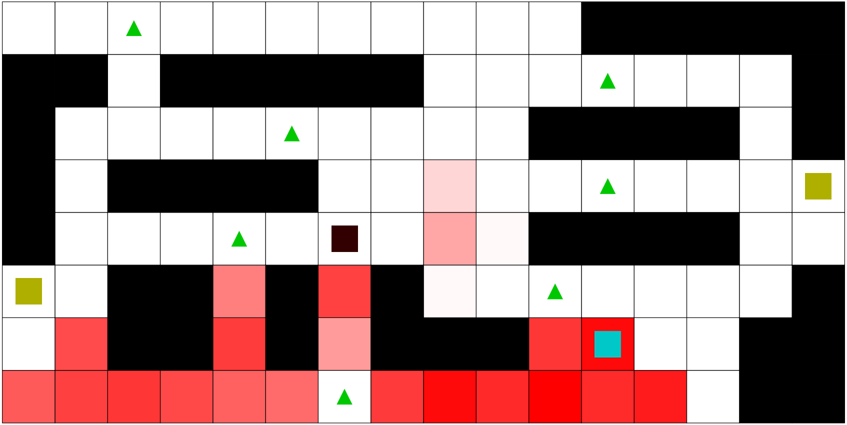
Carte de chaleur : Stratégie A, Partie # 1, Tour # 24



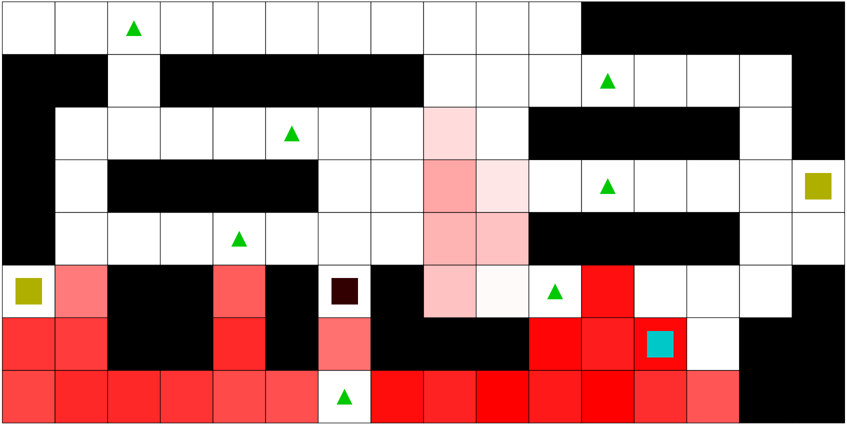
Carte de chaleur : Stratégie A, Partie # 1, Tour # 25



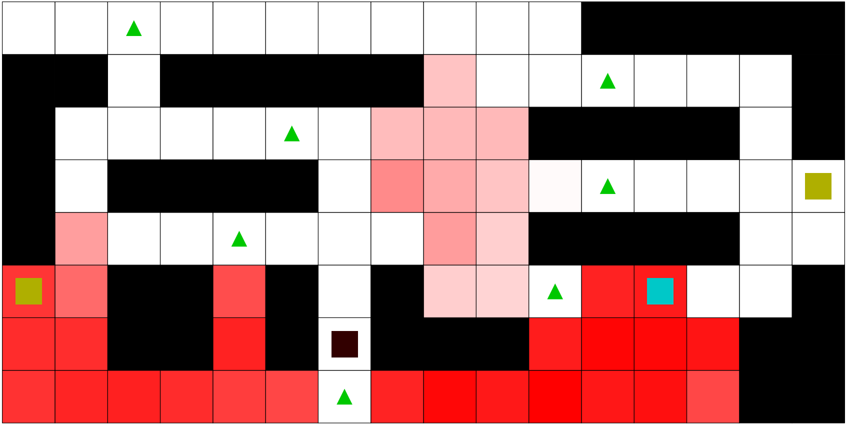
Carte de chaleur : Stratégie A, Partie # 1, Tour # 26



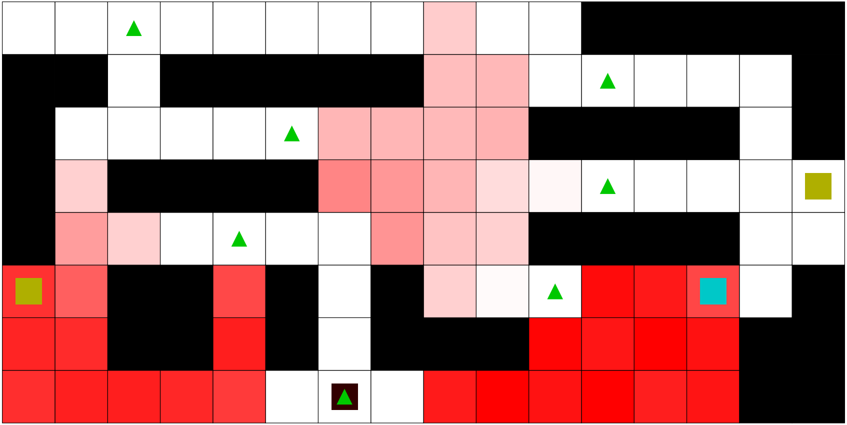
Carte de chaleur : Stratégie A, Partie # 1, Tour # 27



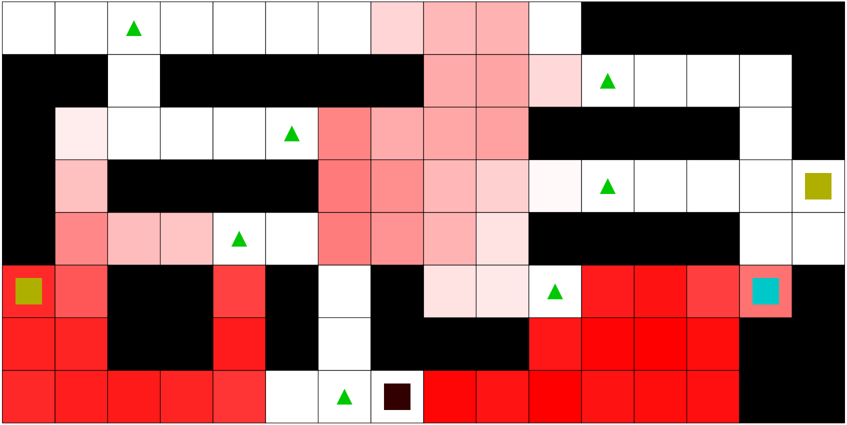
Carte de chaleur : Stratégie A, Partie # 1, Tour # 28



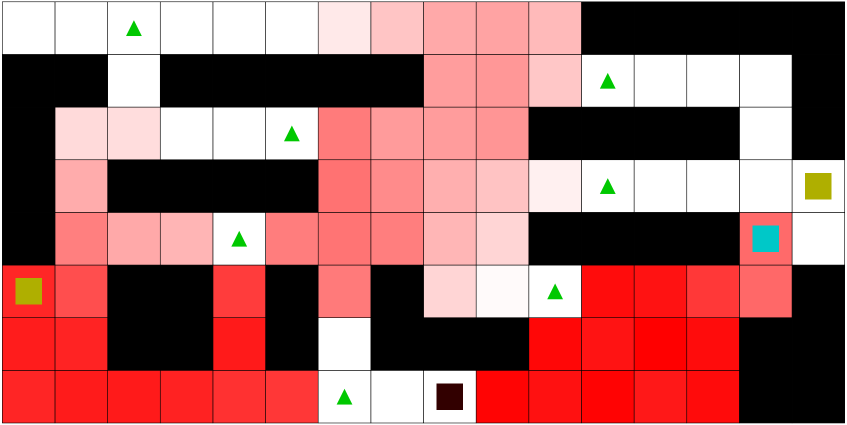
Carte de chaleur : Stratégie A, Partie # 1, Tour # 29



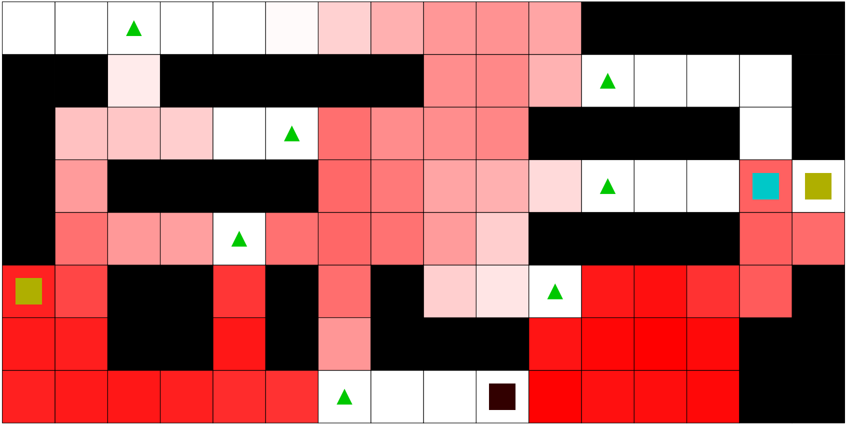
Carte de chaleur : Stratégie A, Partie # 1, Tour # 30



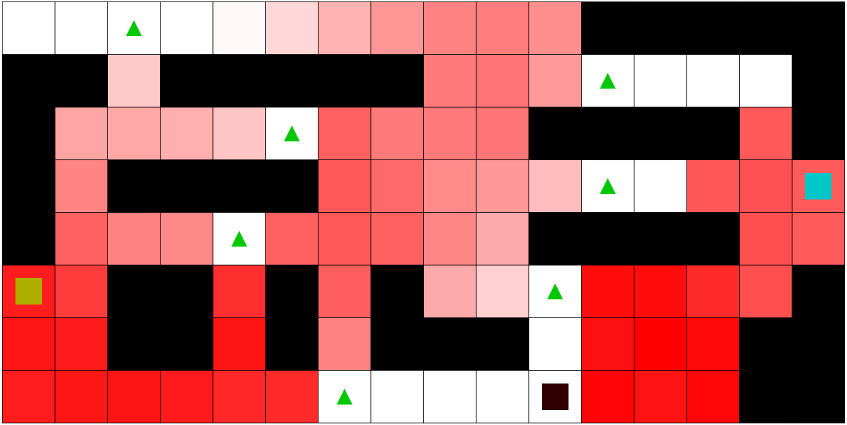
Carte de chaleur : Stratégie A, Partie # 1, Tour # 31



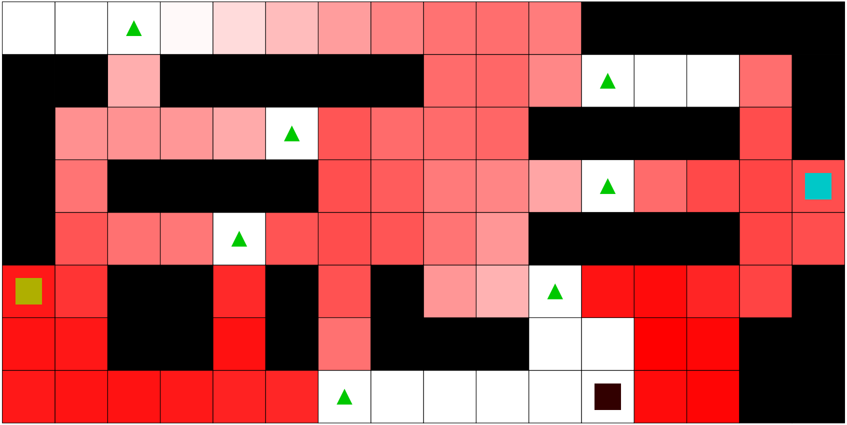
Carte de chaleur : Stratégie A, Partie # 1, Tour # 32

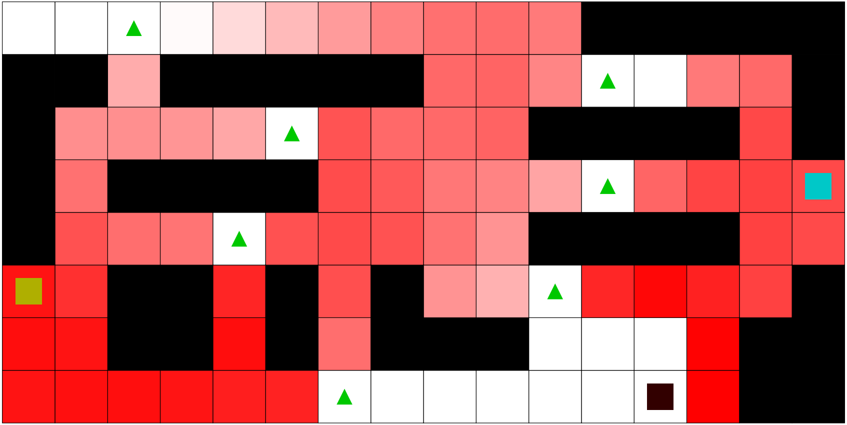


Carte de chaleur : Stratégie A, Partie # 1, Tour # 33

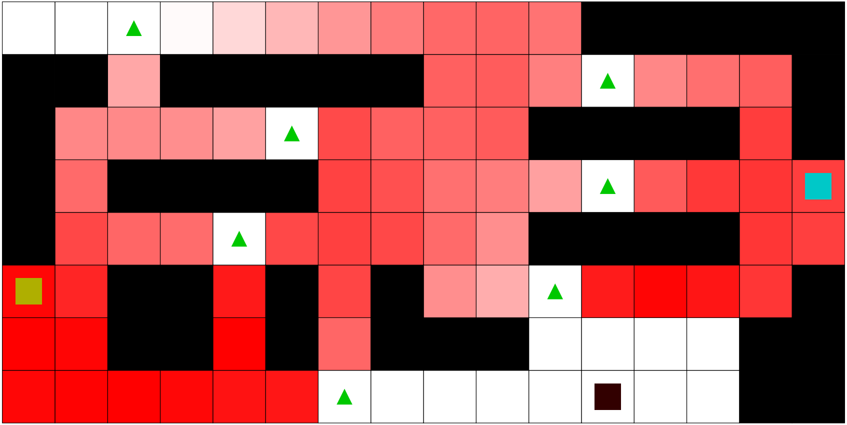


Carte de chaleur : Stratégie A, Partie # 1, Tour # 34

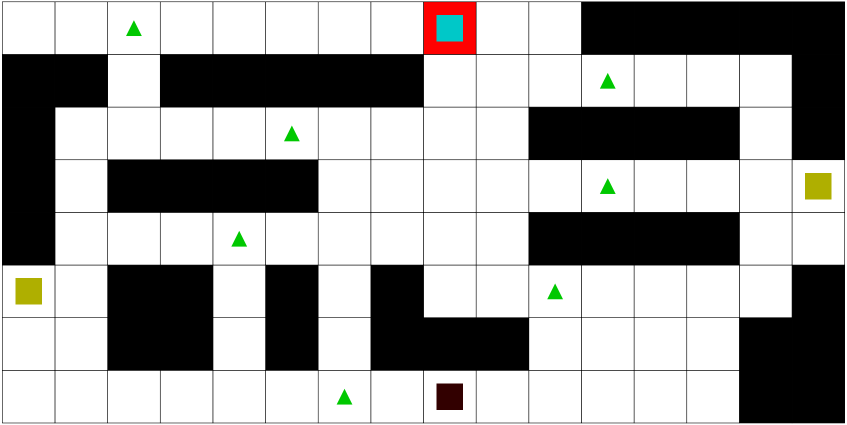




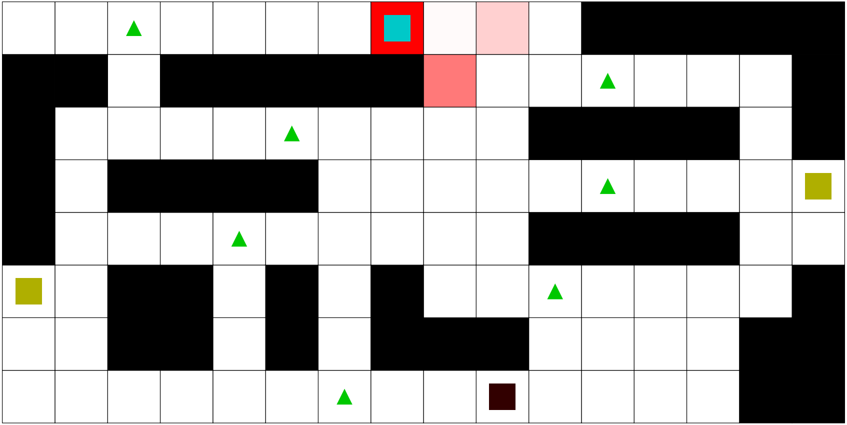
Carte de chaleur : Stratégie A, Partie # 1, Tour # 36



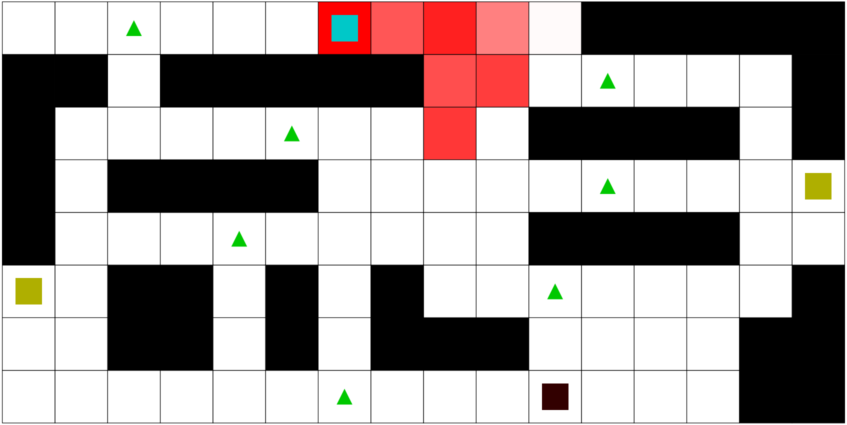
Carte de chaleur : Stratégie A, Partie # 10, Tour # 1



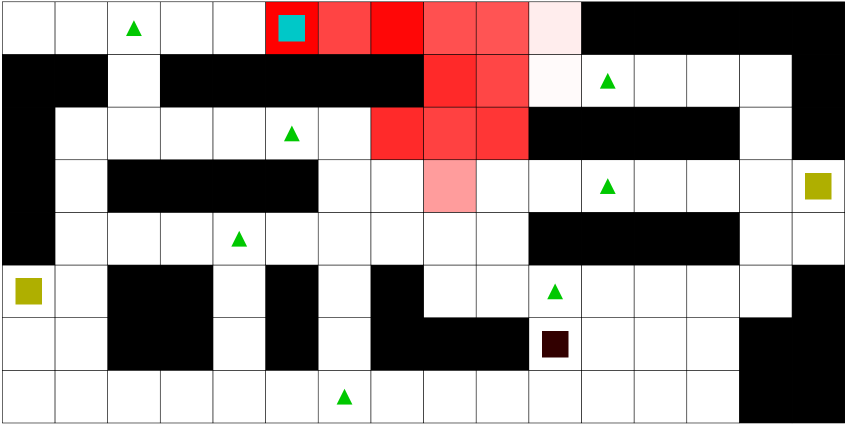
Carte de chaleur : Stratégie A, Partie # 10, Tour # 2



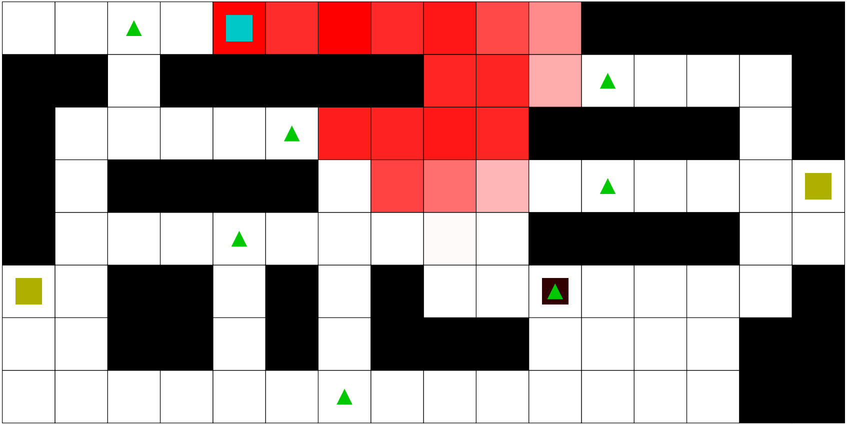
Carte de chaleur : Stratégie A, Partie # 10, Tour # 3



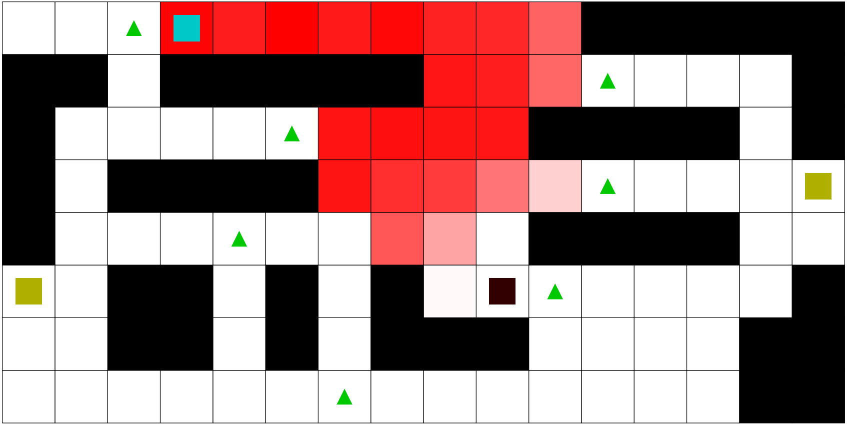
Carte de chaleur : Stratégie A, Partie # 10, Tour # 4



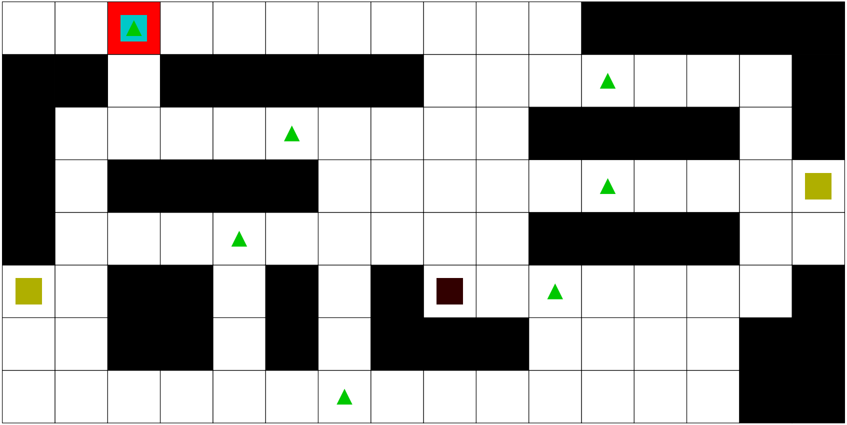
Carte de chaleur : Stratégie A, Partie # 10, Tour # 5



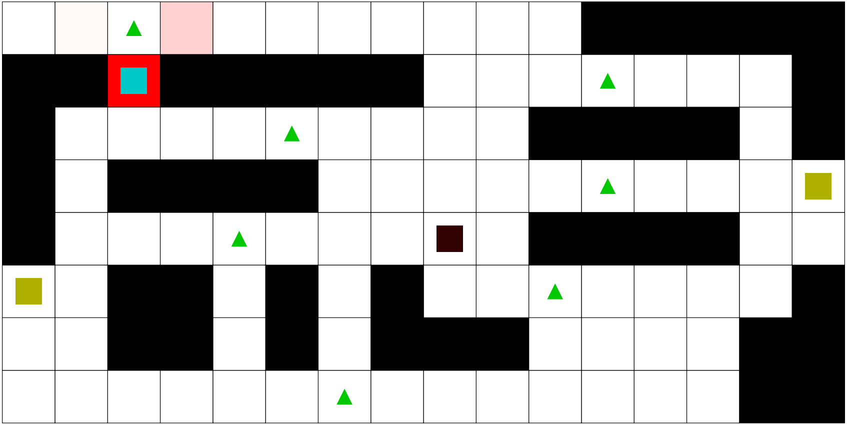
Carte de chaleur : Stratégie A, Partie # 10, Tour # 6



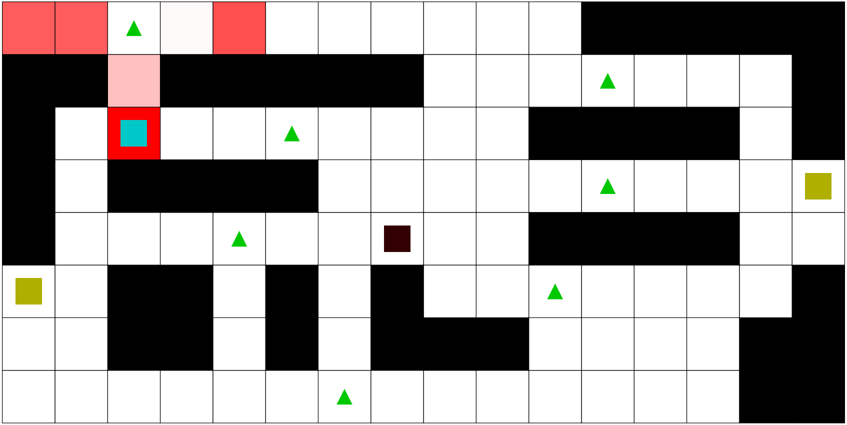
Carte de chaleur : Stratégie A, Partie # 10, Tour # 7



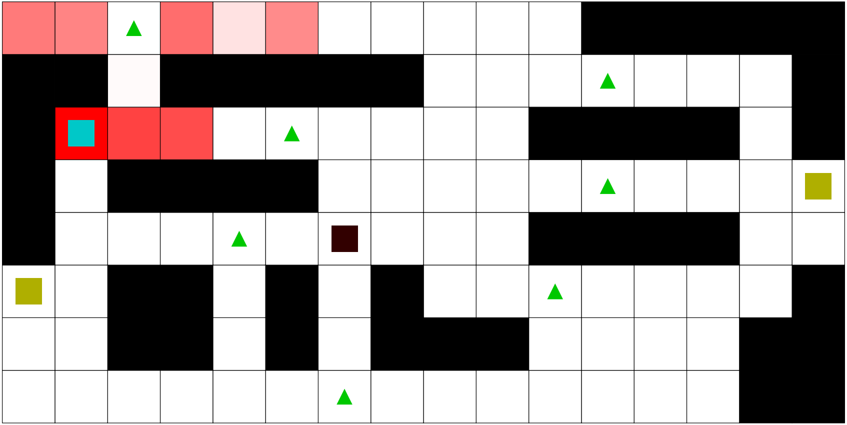
Carte de chaleur : Stratégie A, Partie # 10, Tour # 8



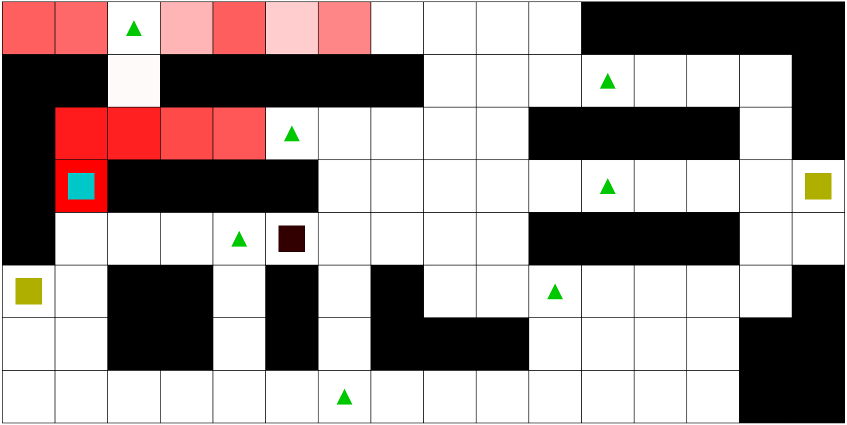
Carte de chaleur : Stratégie A, Partie # 10, Tour # 9



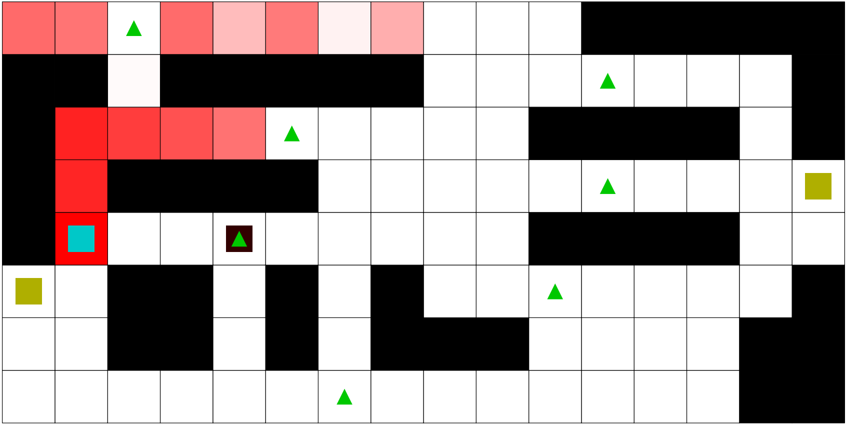
Carte de chaleur : Stratégie A, Partie # 10, Tour # 10



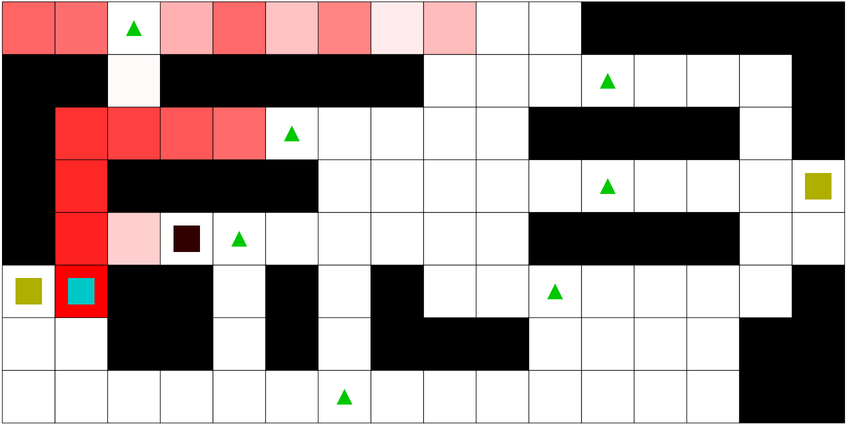
Carte de chaleur : Stratégie A, Partie # 10, Tour # 11



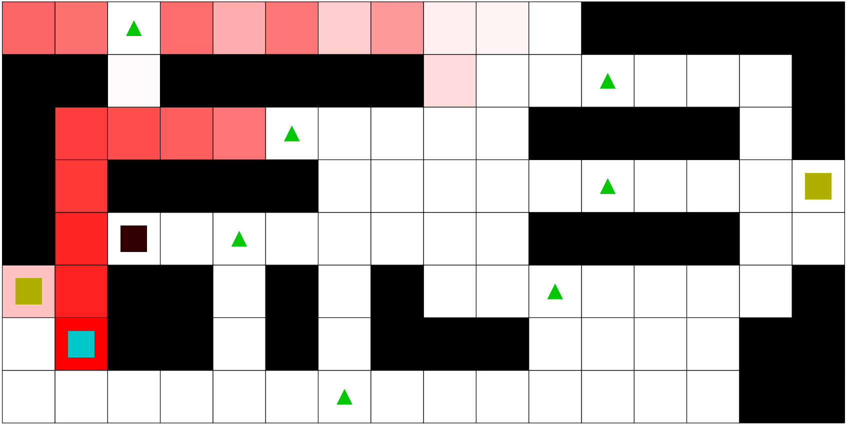
Carte de chaleur : Stratégie A, Partie # 10, Tour # 12



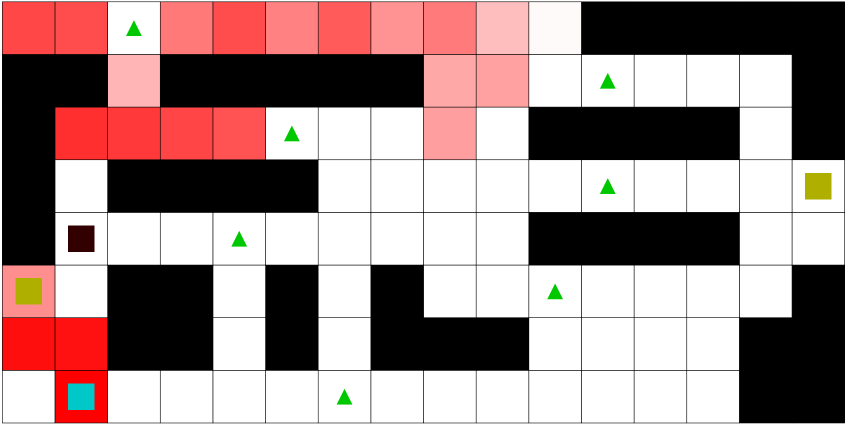
Carte de chaleur : Stratégie A, Partie # 10, Tour # 13



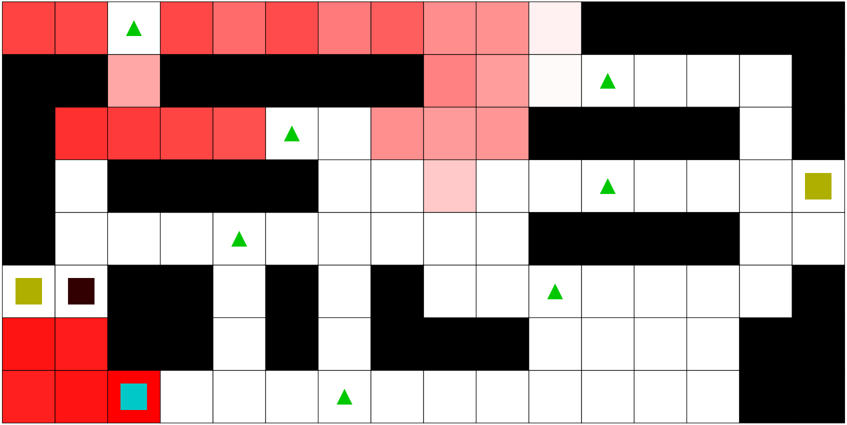
Carte de chaleur : Stratégie A, Partie # 10, Tour # 14



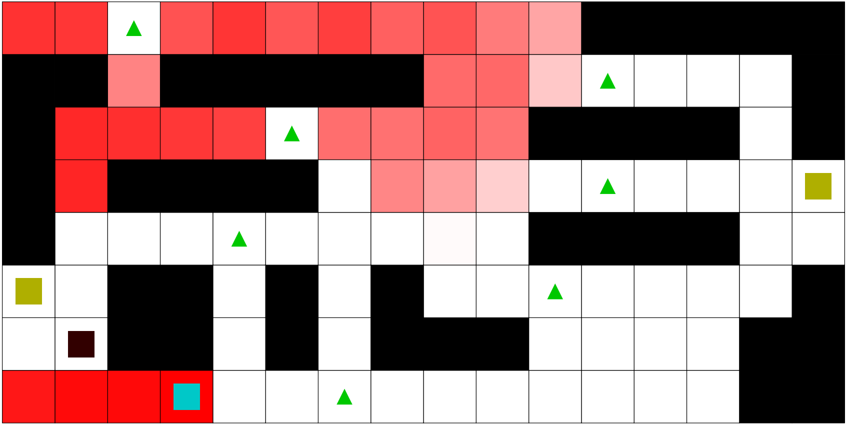
Carte de chaleur : Stratégie A, Partie # 10, Tour # 15



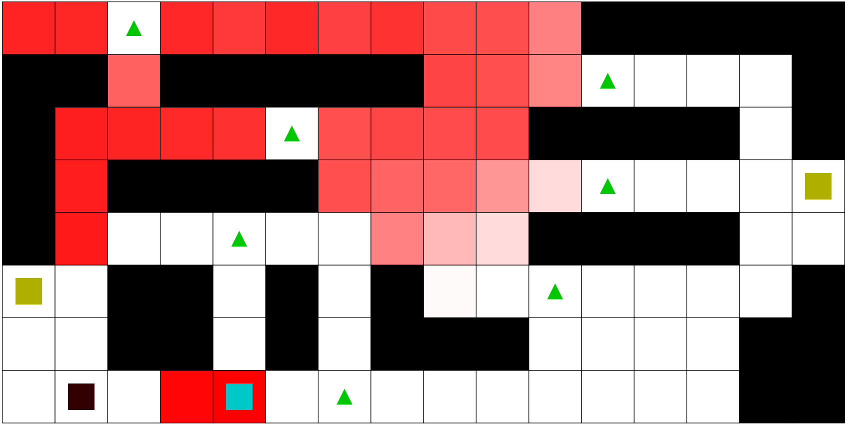
Carte de chaleur : Stratégie A, Partie # 10, Tour # 16



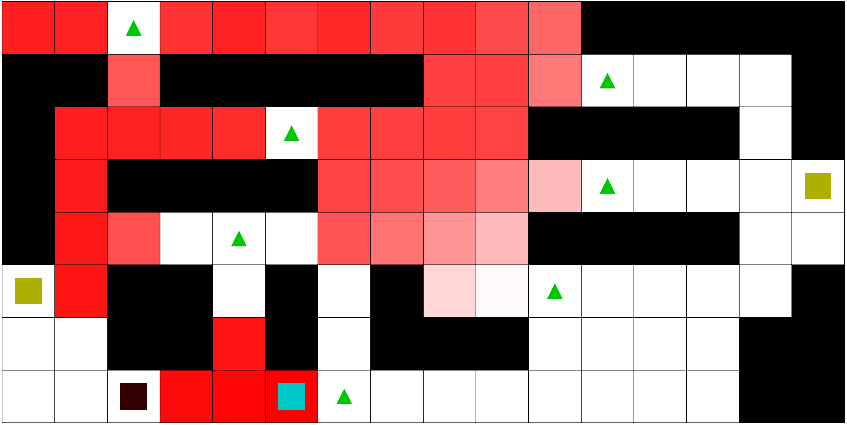
Carte de chaleur : Stratégie A, Partie # 10, Tour # 17



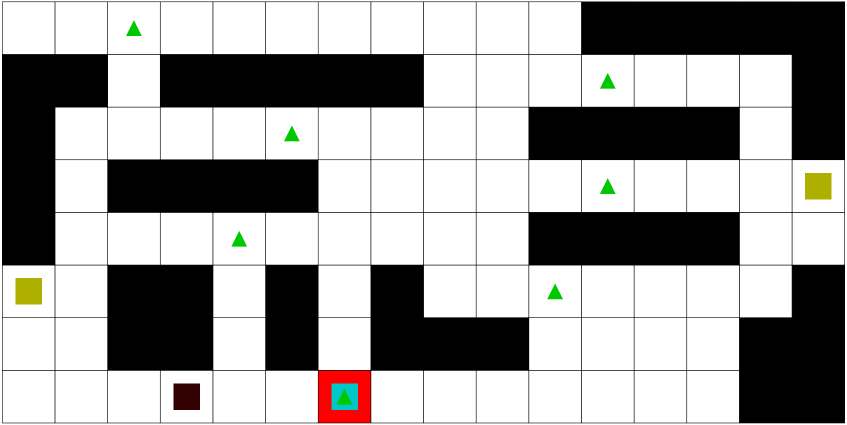
Carte de chaleur : Stratégie A, Partie # 10, Tour # 18



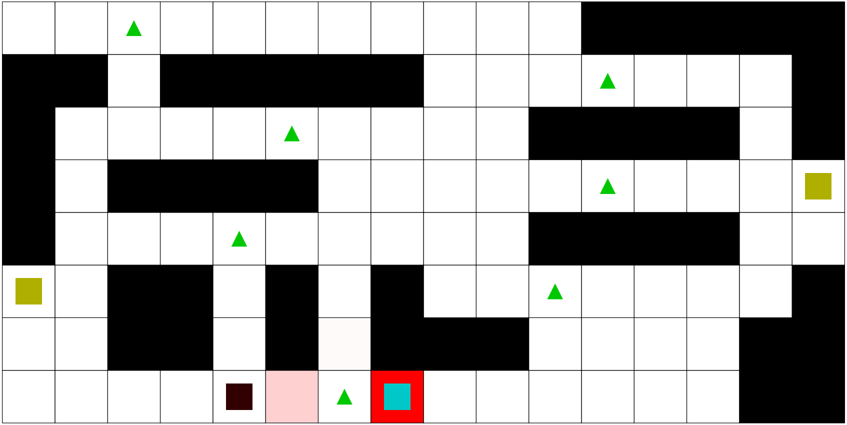
Carte de chaleur : Stratégie A, Partie # 10, Tour # 19



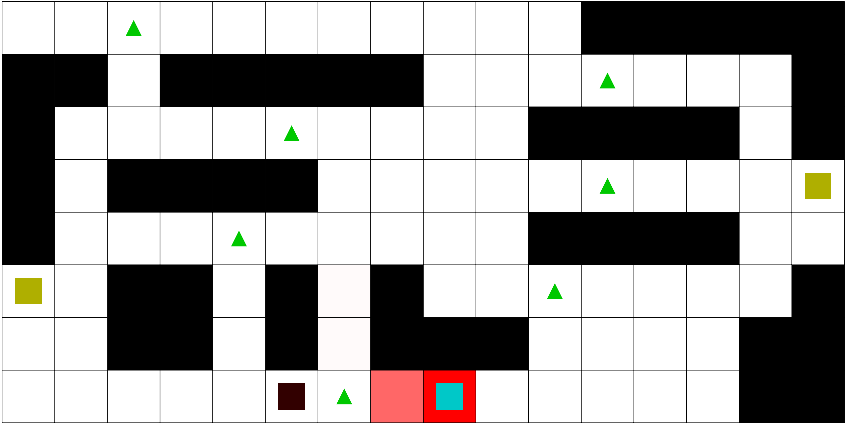
Carte de chaleur : Stratégie A, Partie # 10, Tour # 20



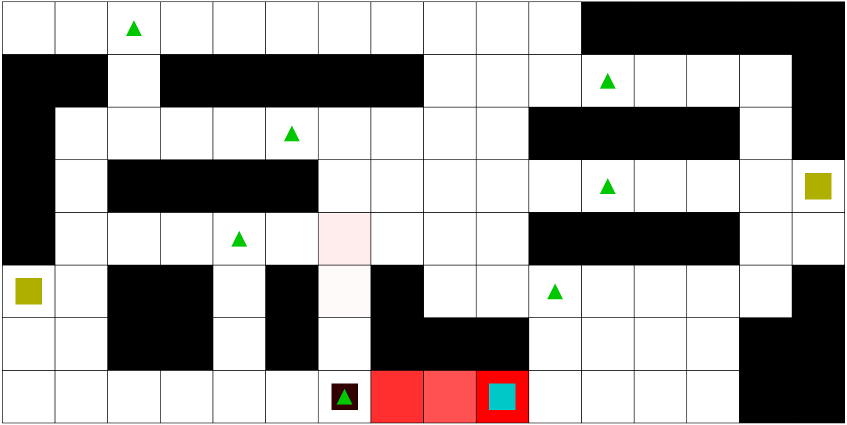
Carte de chaleur : Stratégie A, Partie # 10, Tour # 21



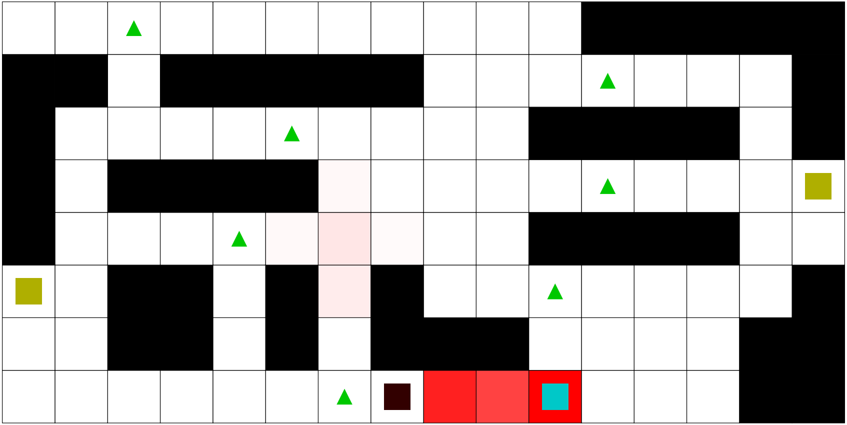
Carte de chaleur : Stratégie A, Partie # 10, Tour # 22



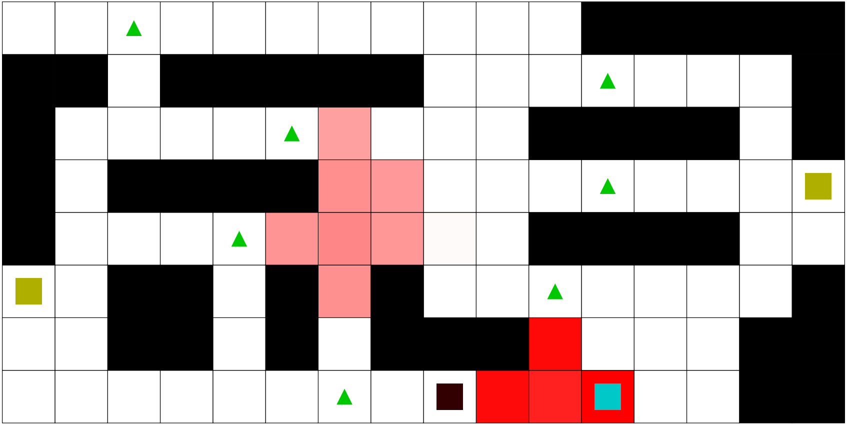
Carte de chaleur : Stratégie A, Partie # 10, Tour # 23



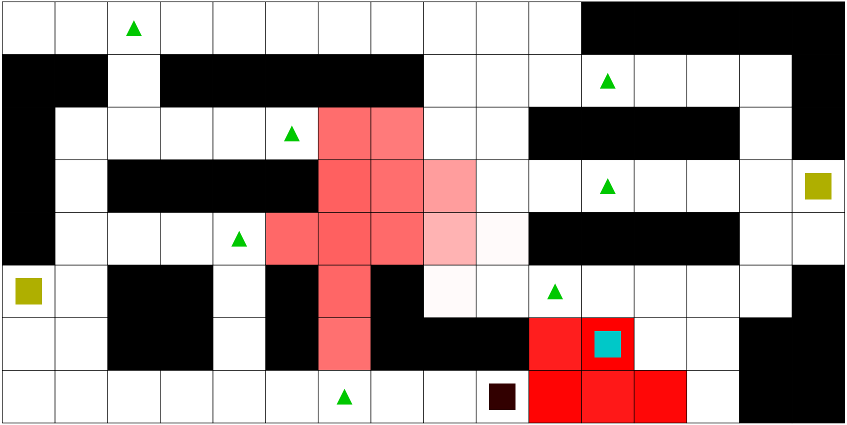
Carte de chaleur : Stratégie A, Partie # 10, Tour # 24



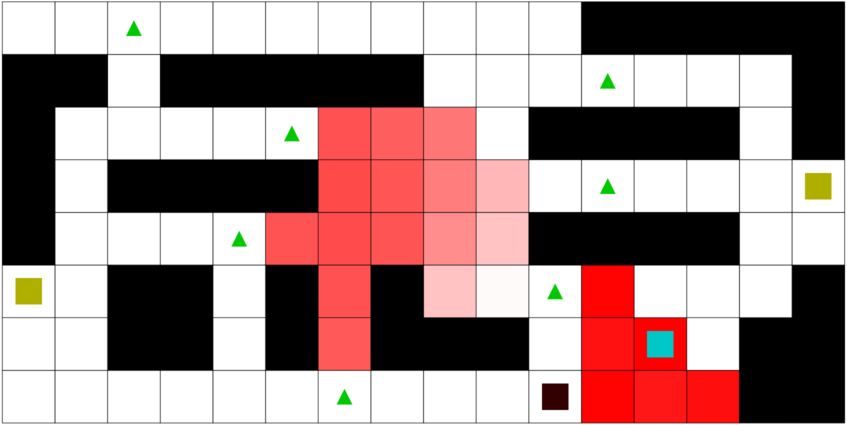
Carte de chaleur : Stratégie A, Partie # 10, Tour # 25



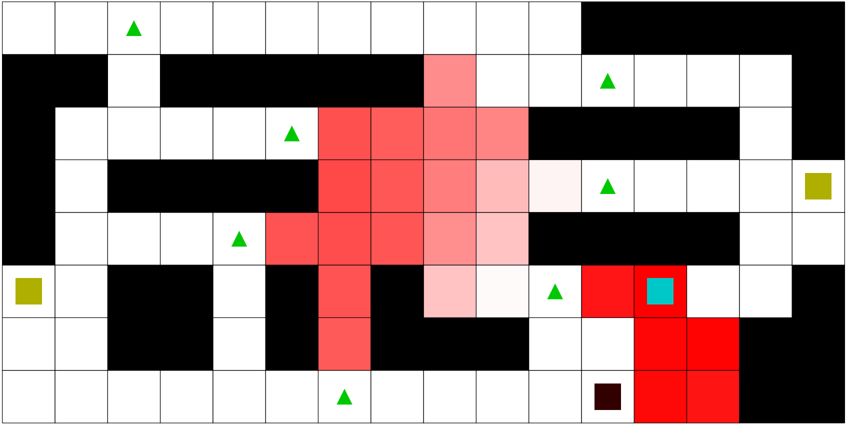
Carte de chaleur : Stratégie A, Partie # 10, Tour # 26



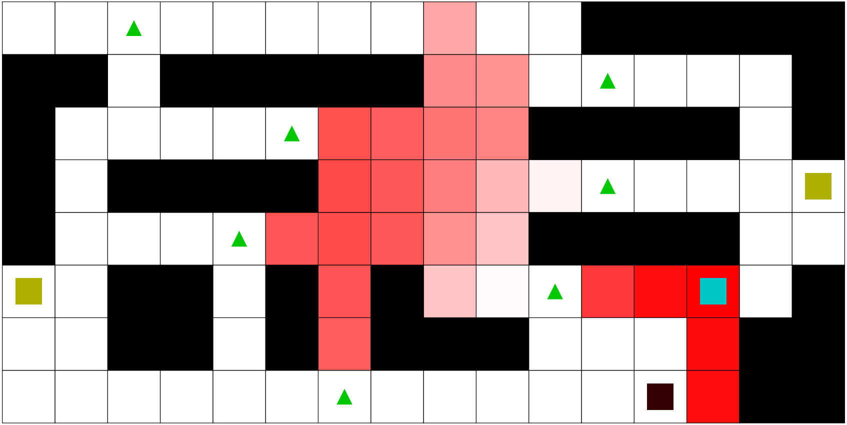
Carte de chaleur : Stratégie A, Partie # 10, Tour # 27



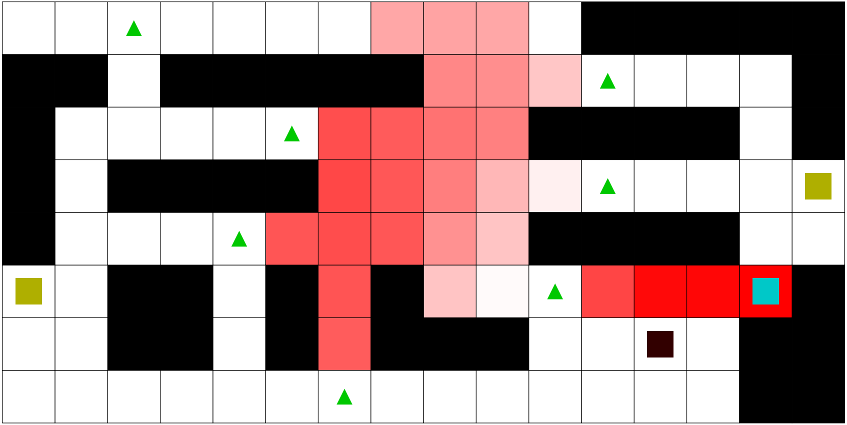
Carte de chaleur : Stratégie A, Partie # 10, Tour # 28



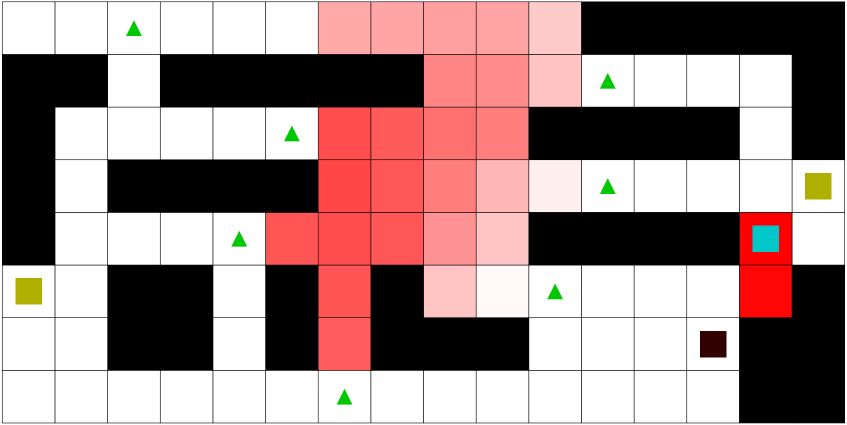
Carte de chaleur : Stratégie A, Partie # 10, Tour # 29



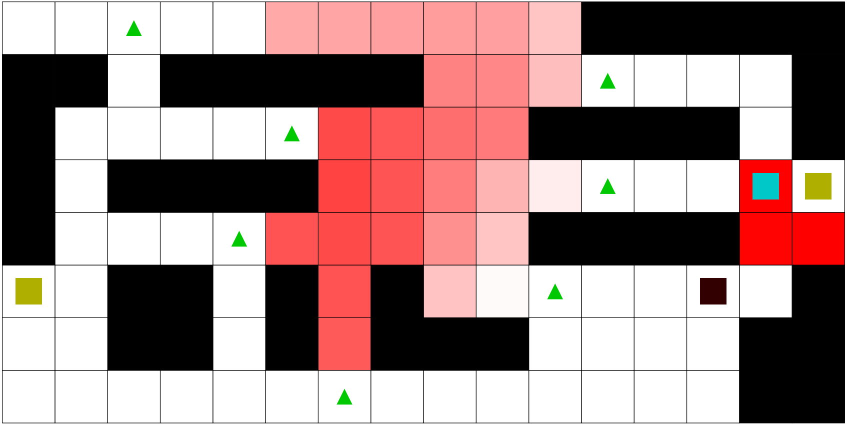
Carte de chaleur : Stratégie A, Partie # 10, Tour # 30



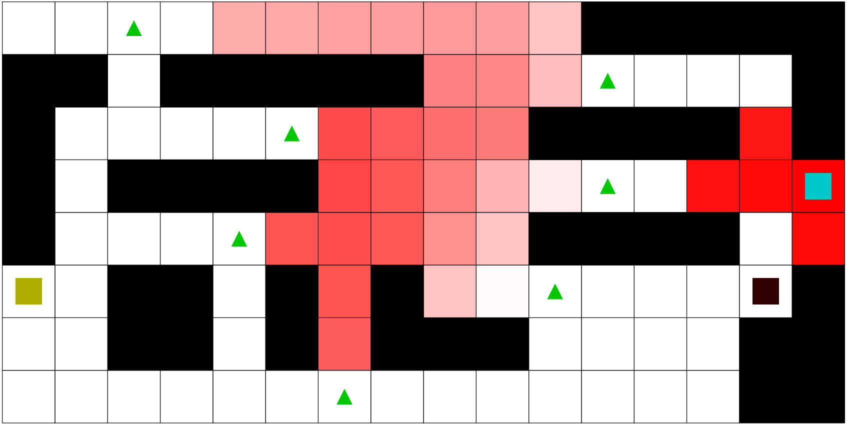
Carte de chaleur : Stratégie A, Partie # 10, Tour # 31



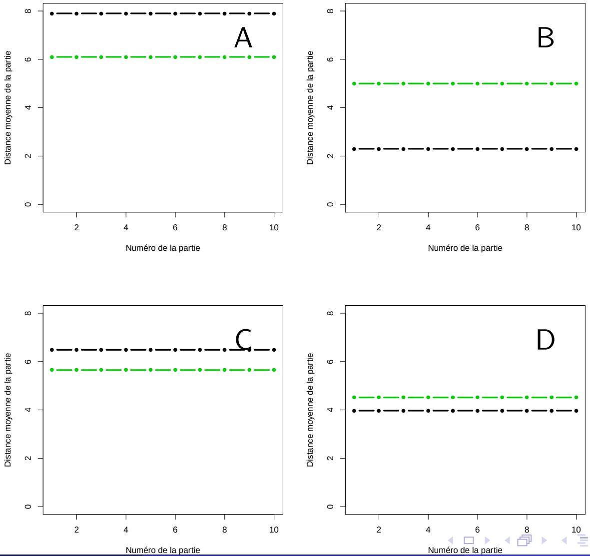
Carte de chaleur : Stratégie A, Partie # 10, Tour # 32



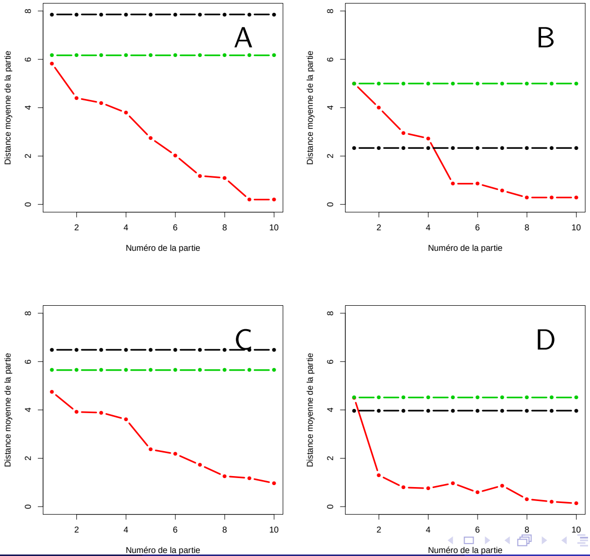
Carte de chaleur : Stratégie A, Partie # 10, Tour # 33



Premier scénario : Évolution de la statistique de test.



Premier scénario: Évolution de la statistique de test.

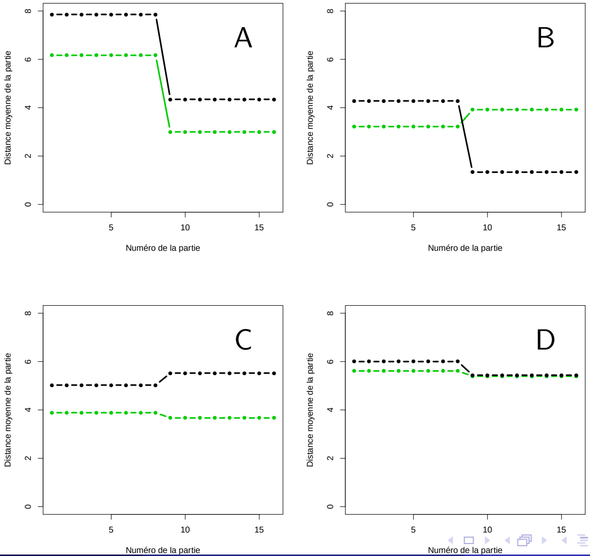


Premier scénario: Comparaison de la statistique de test.

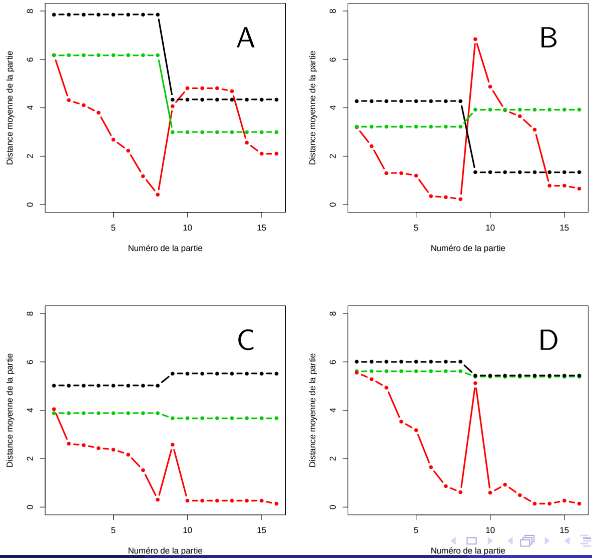
Stratégie	Partie #10	
	Valeurs t	Valeurs p
A	-7.3049	6.225e-09
B	-3.2673	0.00619
C	-5.7857	4.376e-07
D	-9.7531	4.356e-12

Table: Résultats des tests pour les quatre stratégies du premier scénario.

Deuxième scénario : Évolution de la statistique de test.



Deuxième scénario : Évolution de la statistique de test.

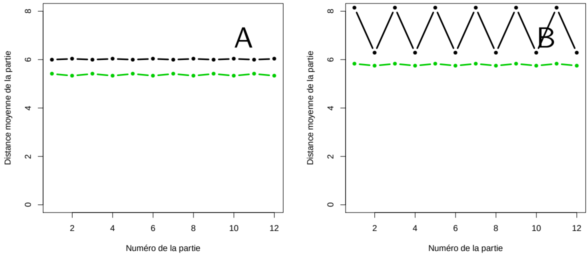


Deuxième scénario : Comparaison de la statistique de test.

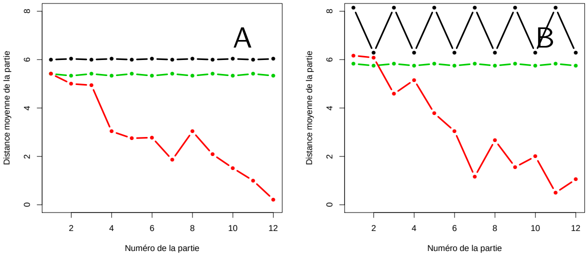
Changement	Partie #8		Partie #9		Partie #16	
	Valeurs t	Valeurs p	Valeurs t	Valeurs p	Valeurs t	Valeurs p
A	-6.9473	1.42e-08	0.7686	0.4492	-0.7168	0.2419
B	-6.4082	9.993e-08	2.7693	0.007986	-4.5768	4.114e-05
C	-6.3822	9.97e-08	-1.1967	0.2386	-5.0932	2.241e-05
D	-7.2901	4.362e-08	-0.2683	0.7894	-8.0063	1.387e-09

Table: Résultats des tests pour les quatre changements de stratégies du deuxième scénario.

Troisième scénario : Évolution de la statistique de test.



Troisième scénario : Évolution de la statistique de test.



Troisième scénario : Comparaison de la statistique de test.

Alternance	Partie #11		Partie #12	
	Valeurs t	Valeurs p	Valeurs t	Valeurs p
A	-5.3526	6.451e-06	-7.8525	1.783e-09
B	-6.8703	1.983e-08	-6.2715	2.553e-07

Table: Résultats des tests pour les deux alternances de stratégies du troisième scénario.

Conclusion

Conclusion

- Estimation en temps réelle précise et efficace.
- Peu d'effort pour un ordinateur, peut s'appliquer à plusieurs domaines.
- L'apprentissage nécéssite un plus d'ajustements.
- Outils très puissants.
- Quelques améliorations : son, HSMM, aglomération d'états. . .

Fin