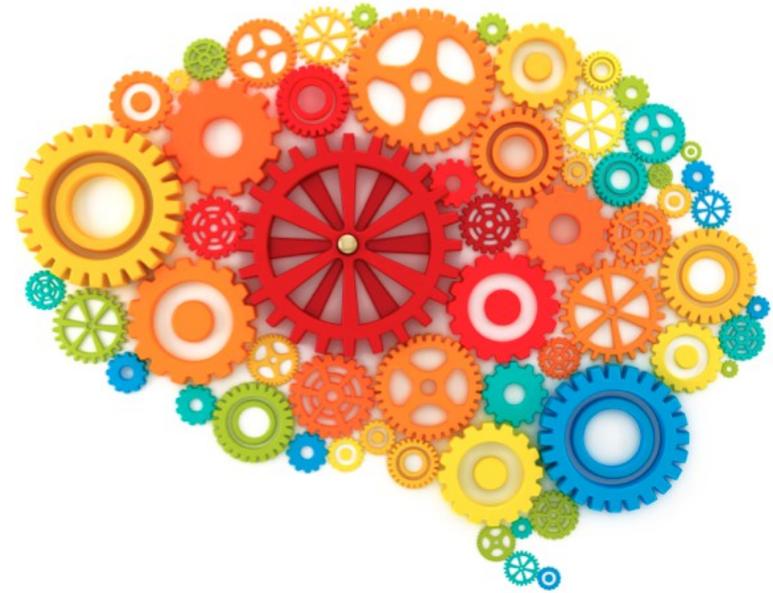




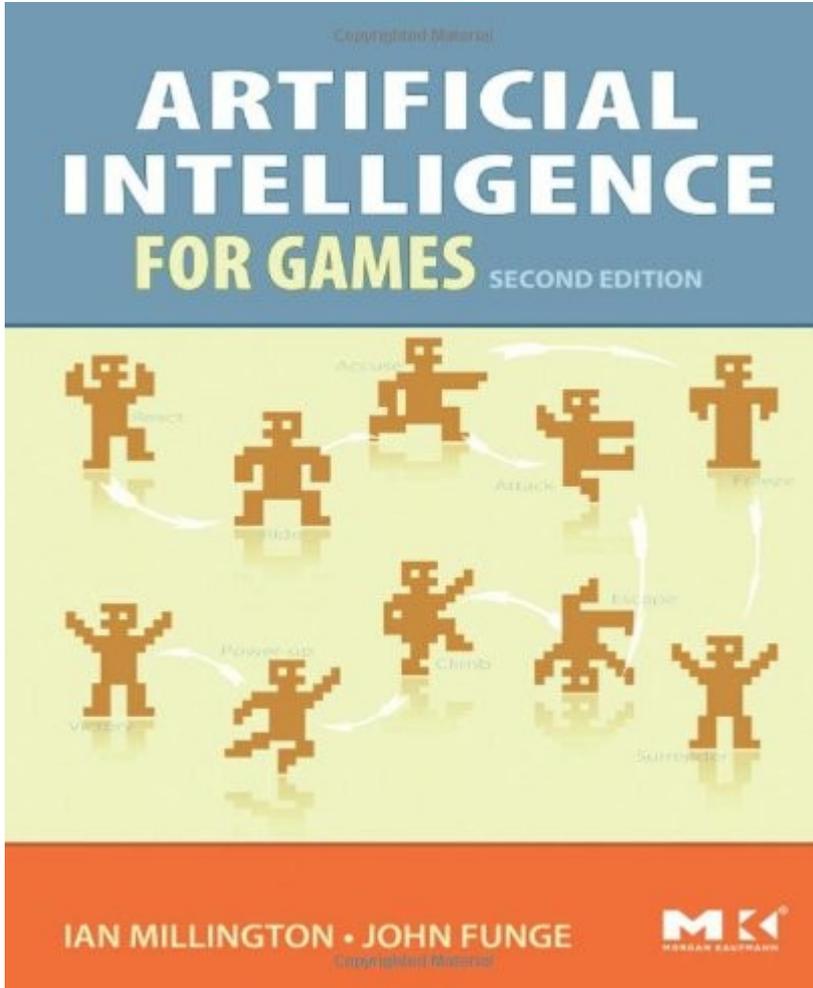
Algorithmes pour le jeu

Intelligence Artificielle



Artificial intelligence for games

Ian Millington, John Funge



- Qu'est ce que l'IA ?
- Objectif de l'IA dans un jeu video
- Exemple IA et techniques
 - Comportement réflexe
 - Planification
 - Décision haut niveau
 - Apprentissage

- **Intelligence Artificielle ?**
 - Émuler/simuler des comportements
 - Résoudre des problèmes qui nécessitent de l'I
 - Résoudre des problèmes encore non résolus
- **Ces transparents**
 - Pas un cours d'IA
 - A quoi peut servir IA en jeu

- A quoi sert une IA ?

- A quoi sert une IA ?
 - Plaisir du joueur
- Qu'est ce qu'une bonne IA ?
 - En Recherche/science :
 - Reproduire humain
 - Résoudre des problèmes complexes
 - Pour un jeu, servir le gameplay !!!
 - Sensation de jeu (ex : half life)
 - Rapide et simple à mettre en oeuvre

Question à se poser
Une IA oui, mais pour quoi faire ?

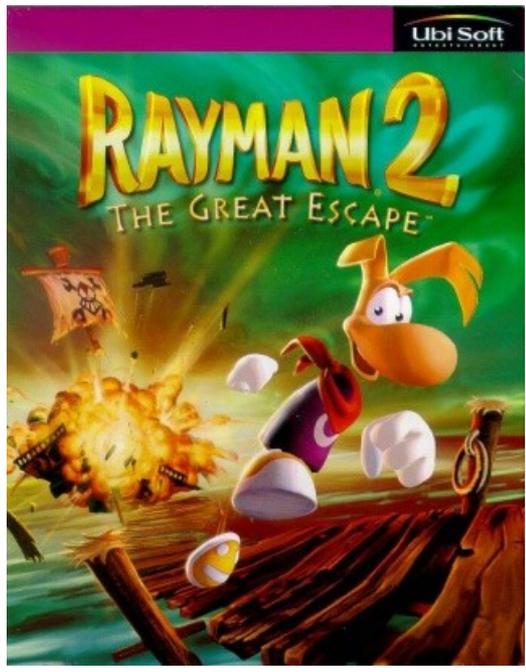
- Qu'est ce que l'IA ?
- Objectif de l'IA dans un jeu video
- Exemple IA et techniques
 - Comportement réflexe
 - Planification
 - Décision haut niveau
 - Apprentissage

- Ne pas faire des choses stupides !!
- Objectifs
 - Proposer un adversaire au joueur
 - Renouveler le jeu
 - Proposer des défis au joueur
 - Assurer une cohérence
 - Construire un système complexe





Obj 1 : Paraître intelligent



(2000)





Obj 1 : Paraître intelligent



(2012)





Obj 1 : Paraître intelligent



(2012)





Obj 1 : Paraître intelligent

(2011)

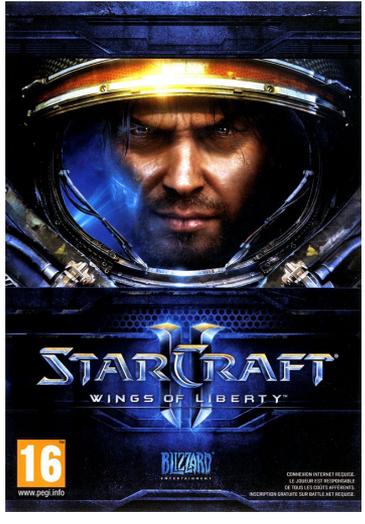


- **Objectif**
 - Avoir un comportement normal
 - Semble simple de l'extérieur
- **Essentiel**
 - Une mauvaise IA détruit la crédibilité
 - Le système doit **paraître** intelligent
- **Interaction entre éléments**

- Différent de faire une IA
 - Erreurs réalistes
 - Gamasutra Intelligent Mistakes: How to Incorporate Stupidity Into Your AI Code (Mick West)
 - Anthropomorphisme
 - Ordinateur laisse gagner
 - Précision
 - Stupide mais pas trop



Obj 2 : Proposer un adversaire au joueur ¹⁵/₂₄₄



(2010)

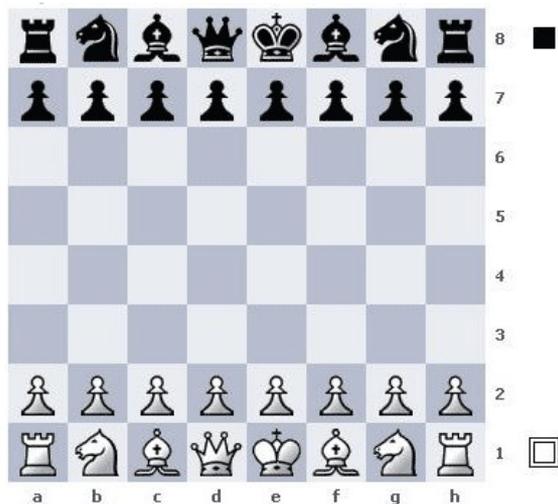


- **Objectif**
 - Ne pas jouer au hasard
 - Proposer des adversaires à la hauteur du joueur
 - la a les mêmes contrôles que le joueur

- **Objectif**

- Ne pas jouer au hasard

- Proposer des adversaires à la hauteur du joueur
- la a les mêmes contrôles que le joueur



Information visible



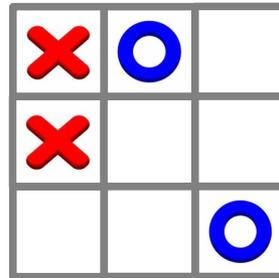
Information cachée

- TicTacToe

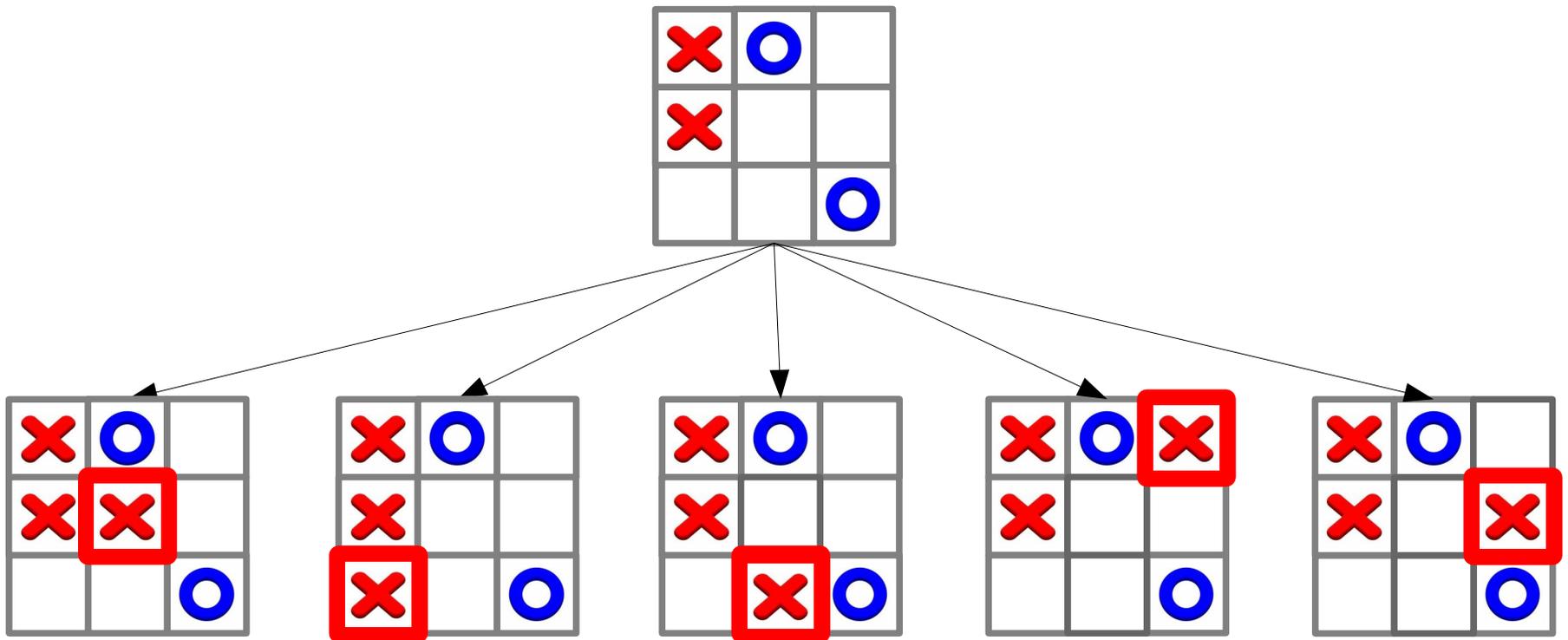


(wikipedia)

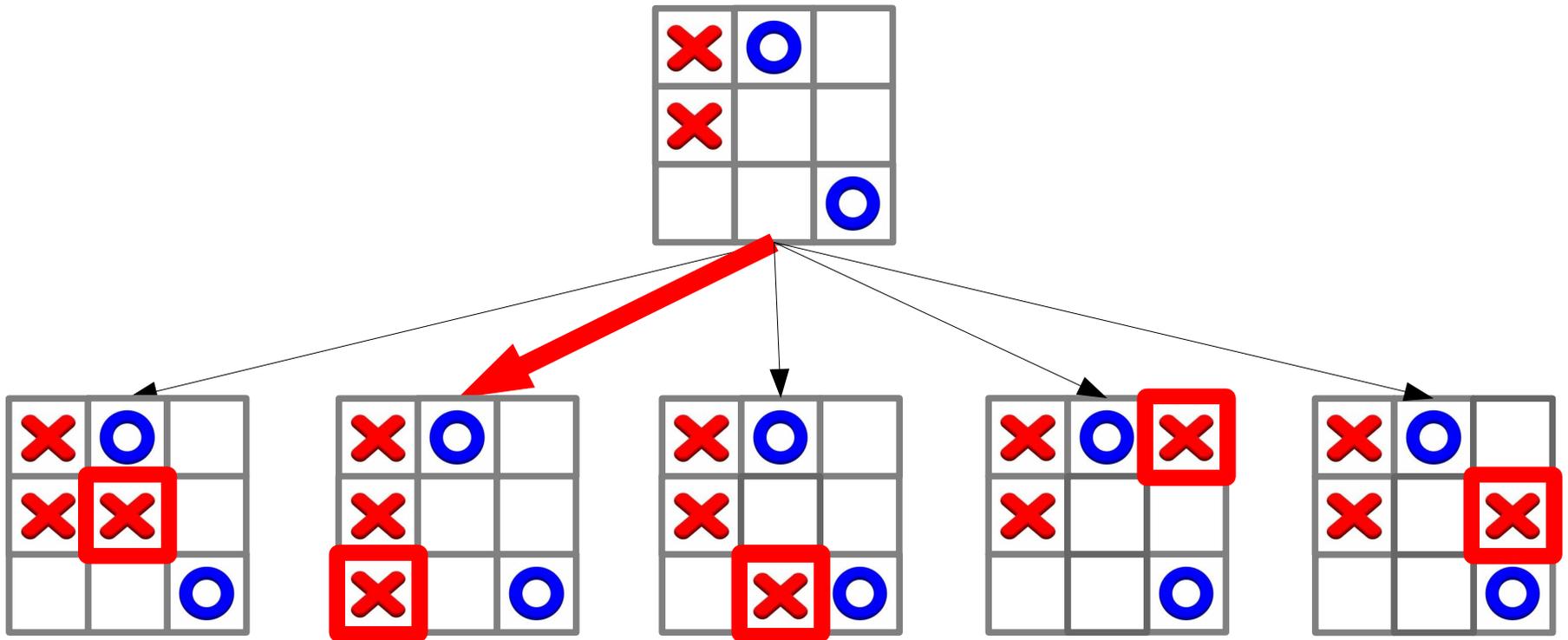
- TicTacToe



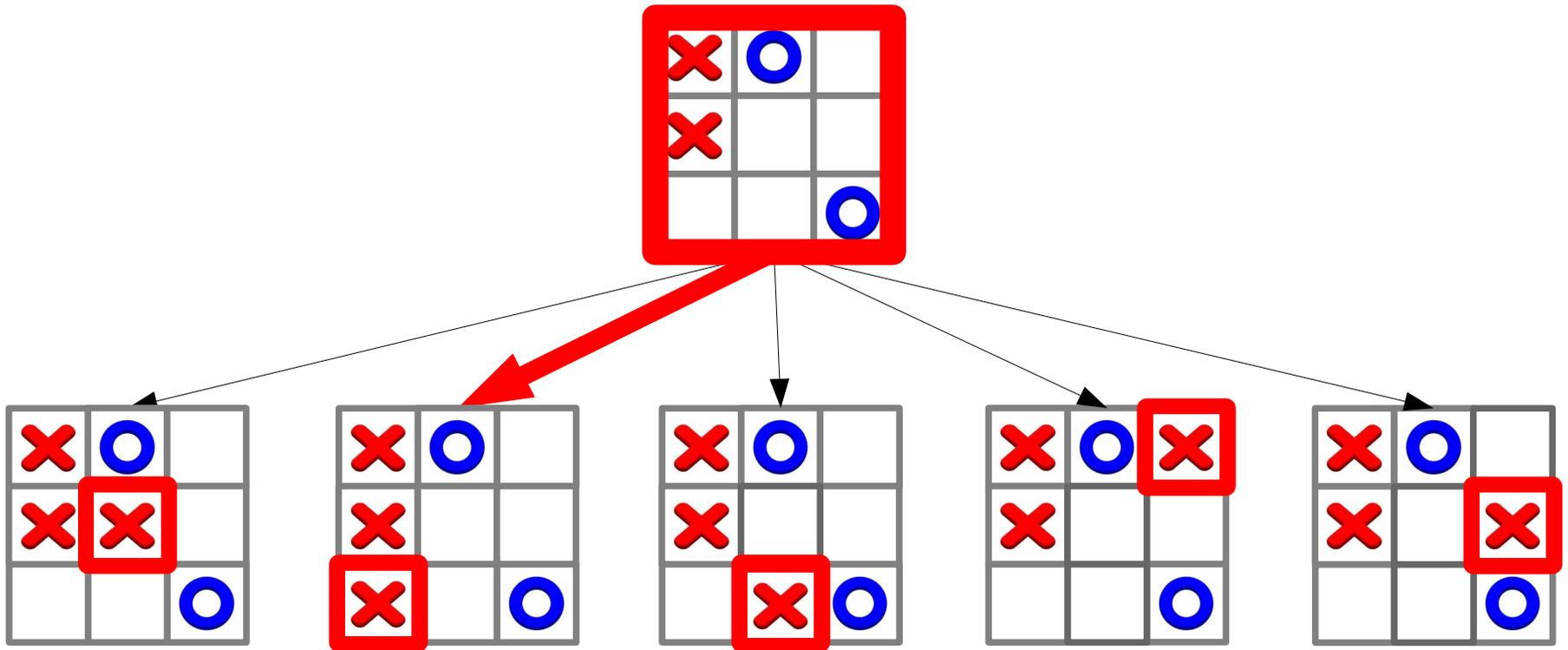
- TicTacToe



- TicTacToe

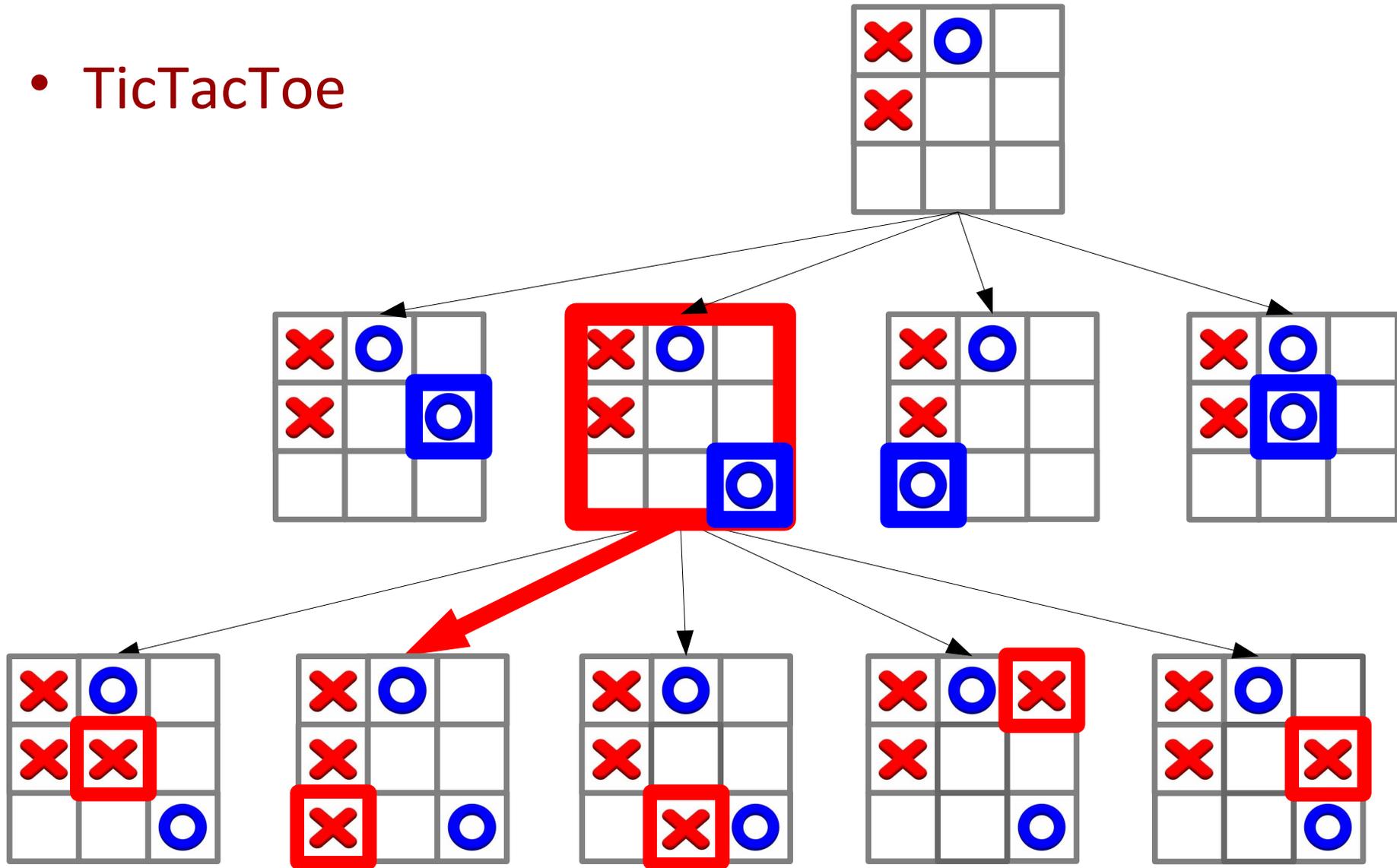


- TicTacToe



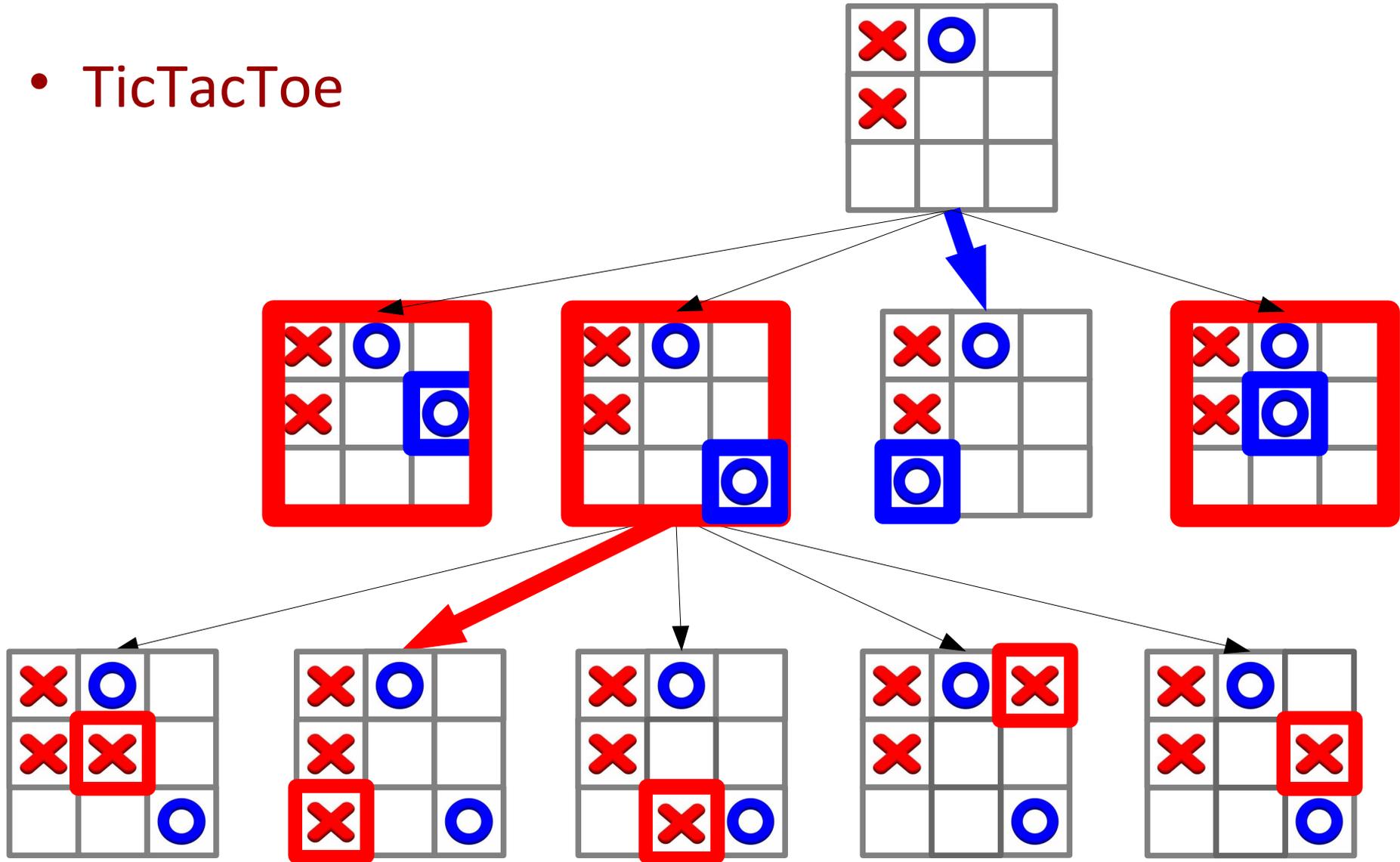
Algorithm Min-Max

- TicTacToe

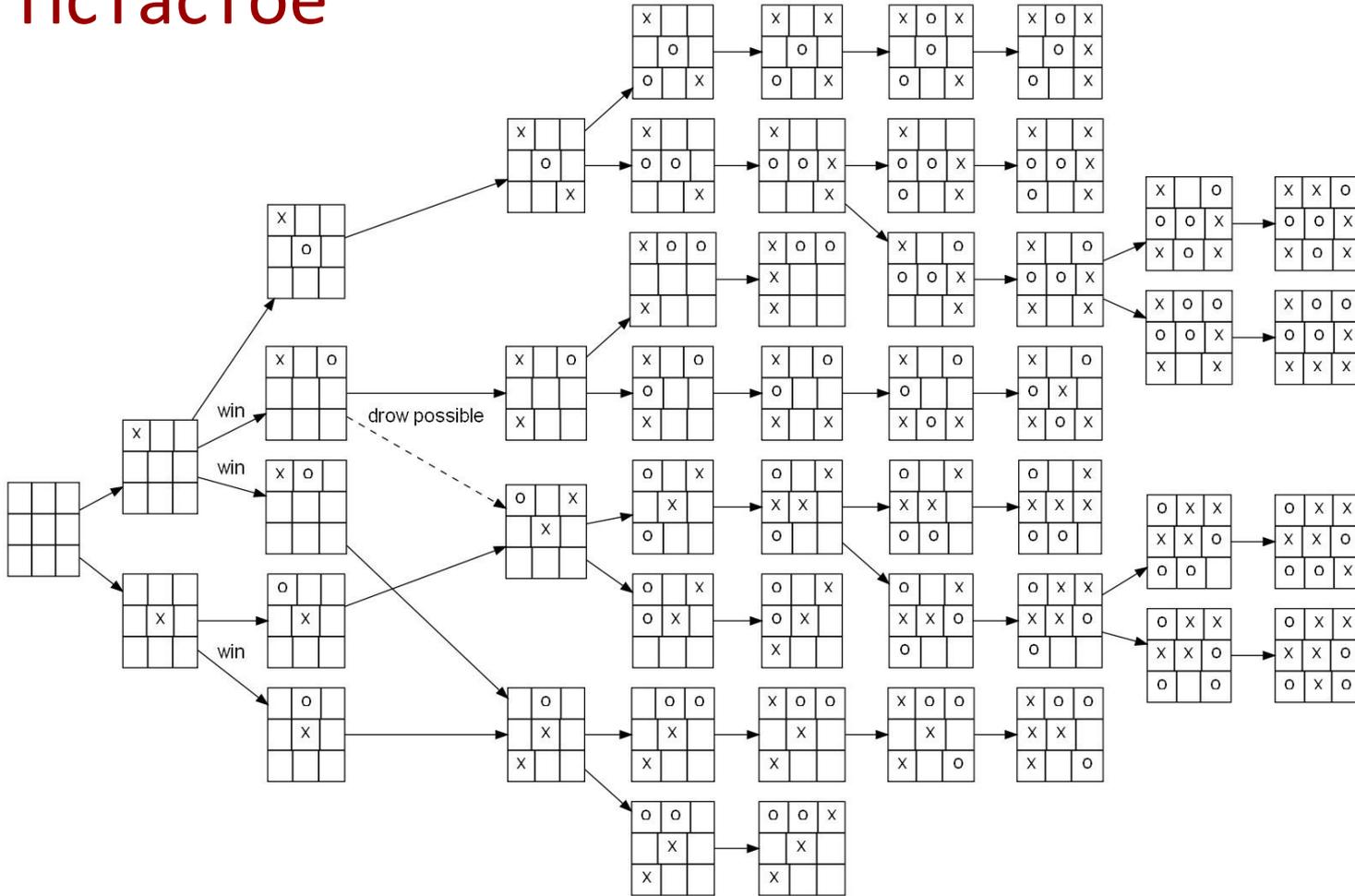


Algorithme Min-Max

- TicTacToe

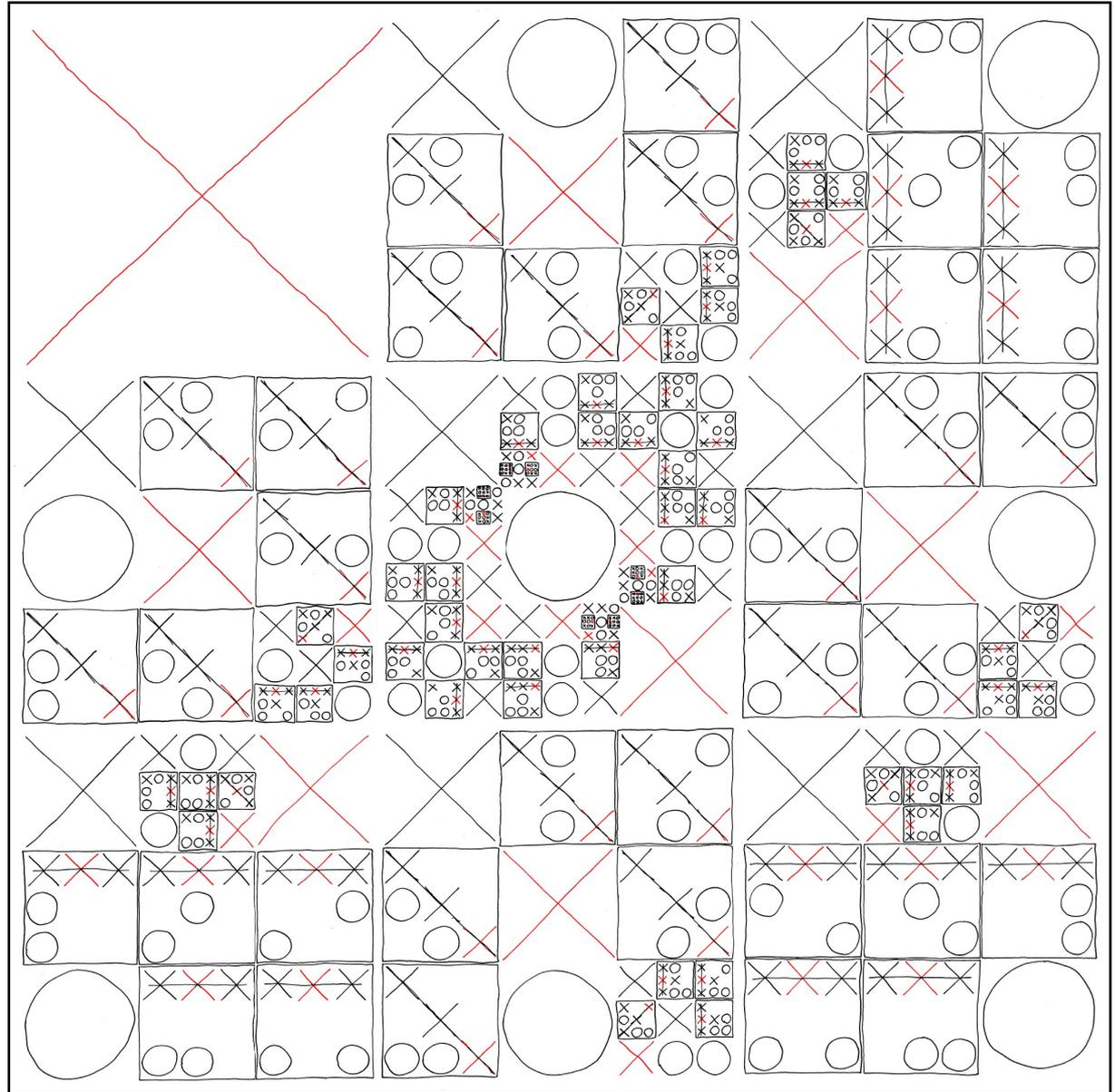


- TicTacToe

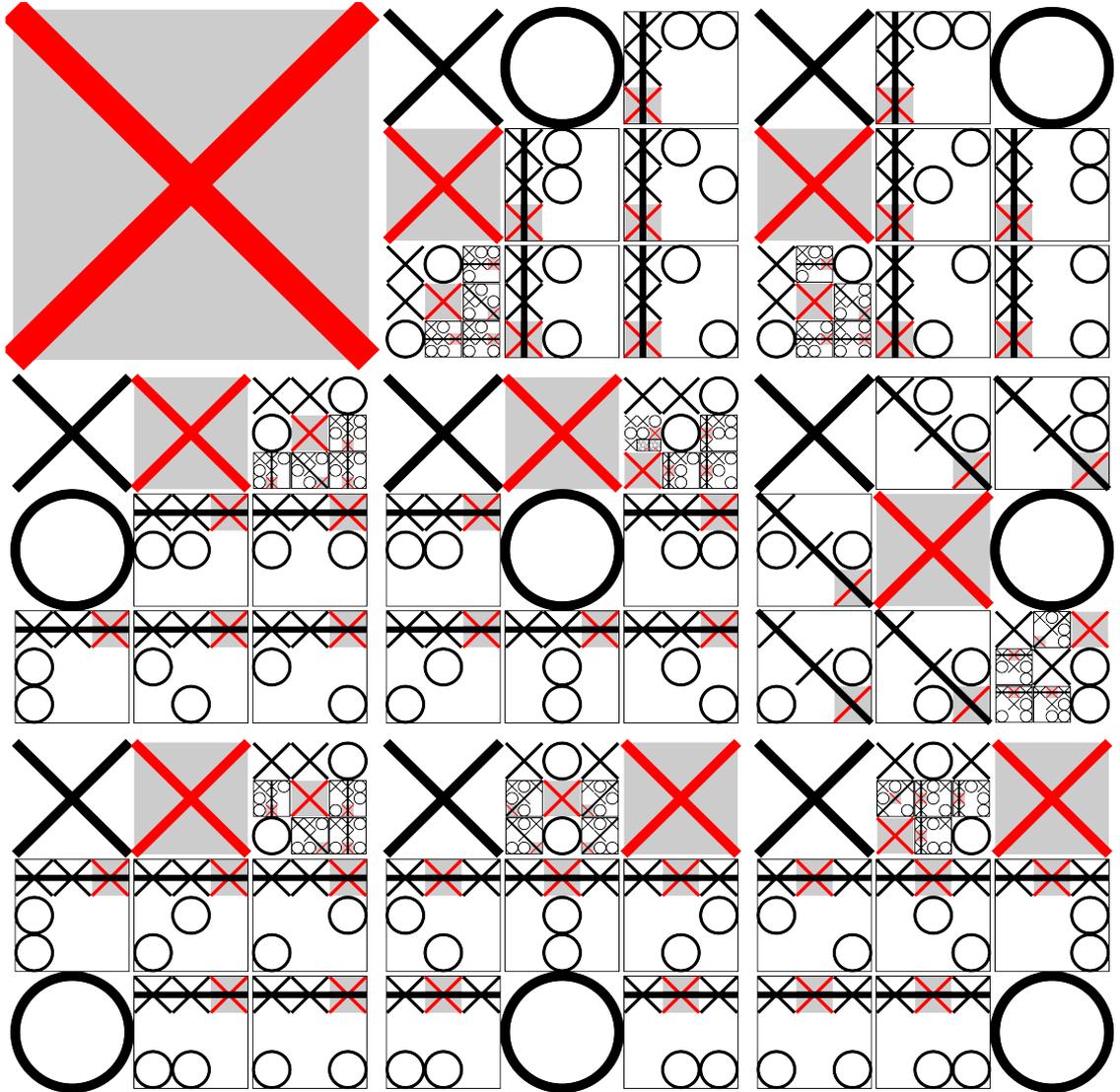


(wikipedia)

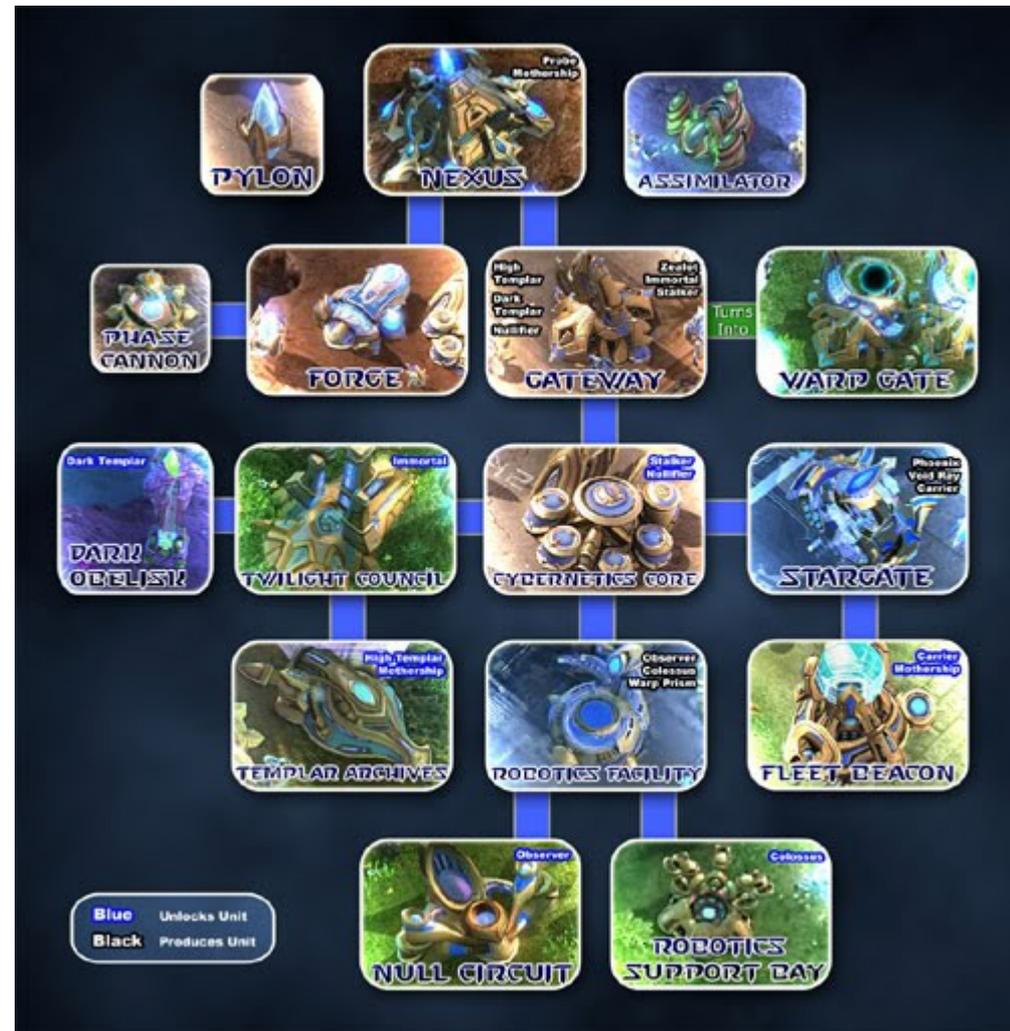
- TicTacToe
 - xkcd



- TicTacToe
 - Xkcd
 - wikipedia



- Evolution chamber
 - Building orders



- Spawn identiques
- Objectif :
 - Proposer des jeux renouvelables
 - En terme d'événements (de tetris à d'autres ..)
 - En terme de scénarisation (en fonction de carac)
 - En terme de comportement
- S'adapter aux envies du joueur ?

Obj 3 : Ne pas être prédictible

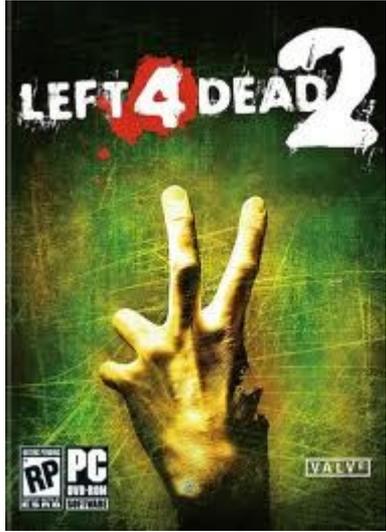
30
244



(2005)



Obj 3 : Ne pas être prédictible



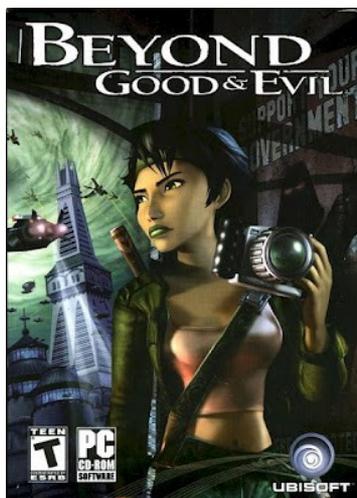
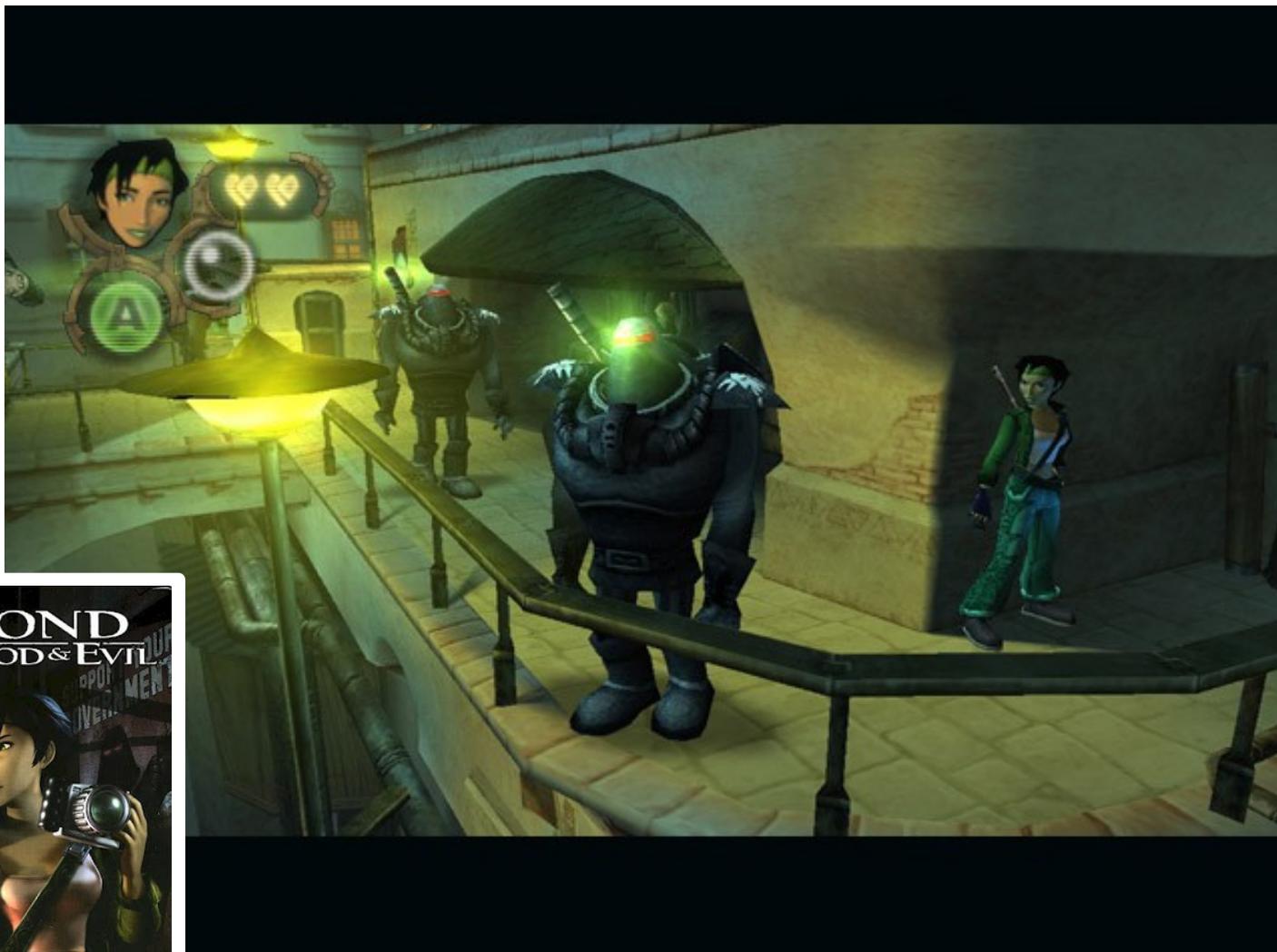
(2009)



- **Objectif**
 - Représenter le comportement des unités
 - De manière réaliste
 - Comme élément de gameplay (diversion, ...)
- **Condition : Le joueur doit le comprendre**



Obj 4 : Modéliser l'individu



(2000)

Obj 4 : Modéliser l'individu

34
244

(2012)



Obj 4 : Modéliser l'individu



(2005)



Obj 4 : Modéliser l'individu

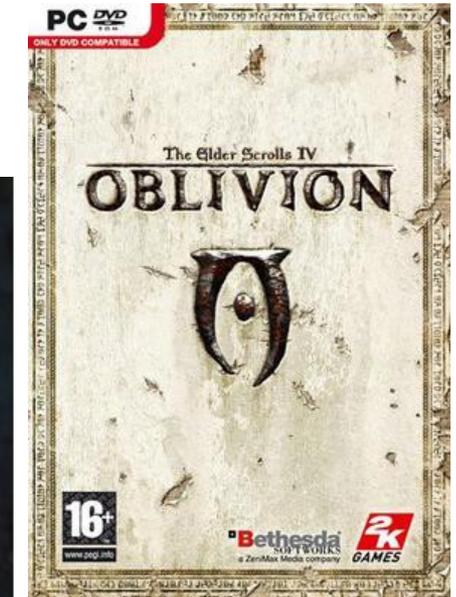
(2011)



- **Objectif**
 - Proposer un monde entier
 - Dans lequel le joueur interagit
 - Avec éventuellement des conséquences (sys complexe)
- **Le joueur fait partie du monde**
 - Comprend ses influences
 - Empoisonner eau
 - Déplacer des moutons
 - « Rhétorique procédurale »

Obj 5 : Produire des mondes cohérents 38 244

(2006)



- Vol du seau Skyrim

(2011)



Obj 5 : Produire des mondes cohérents 40 244



(2004)

- **Systeme complexe (automate cellulaire)**



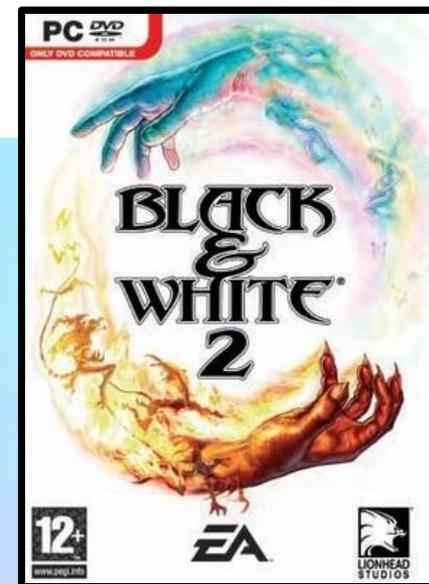
- **Objectif**
 - Faire de l'IA un élément que le joueur utilise
 - En manipulant des choses (système complexe)
 - En utilisant de l'apprentissage
- **Le joueur interagit avec des créatures**
 - Sensation de vie



Obj 6 : IA comme coeur de jeu

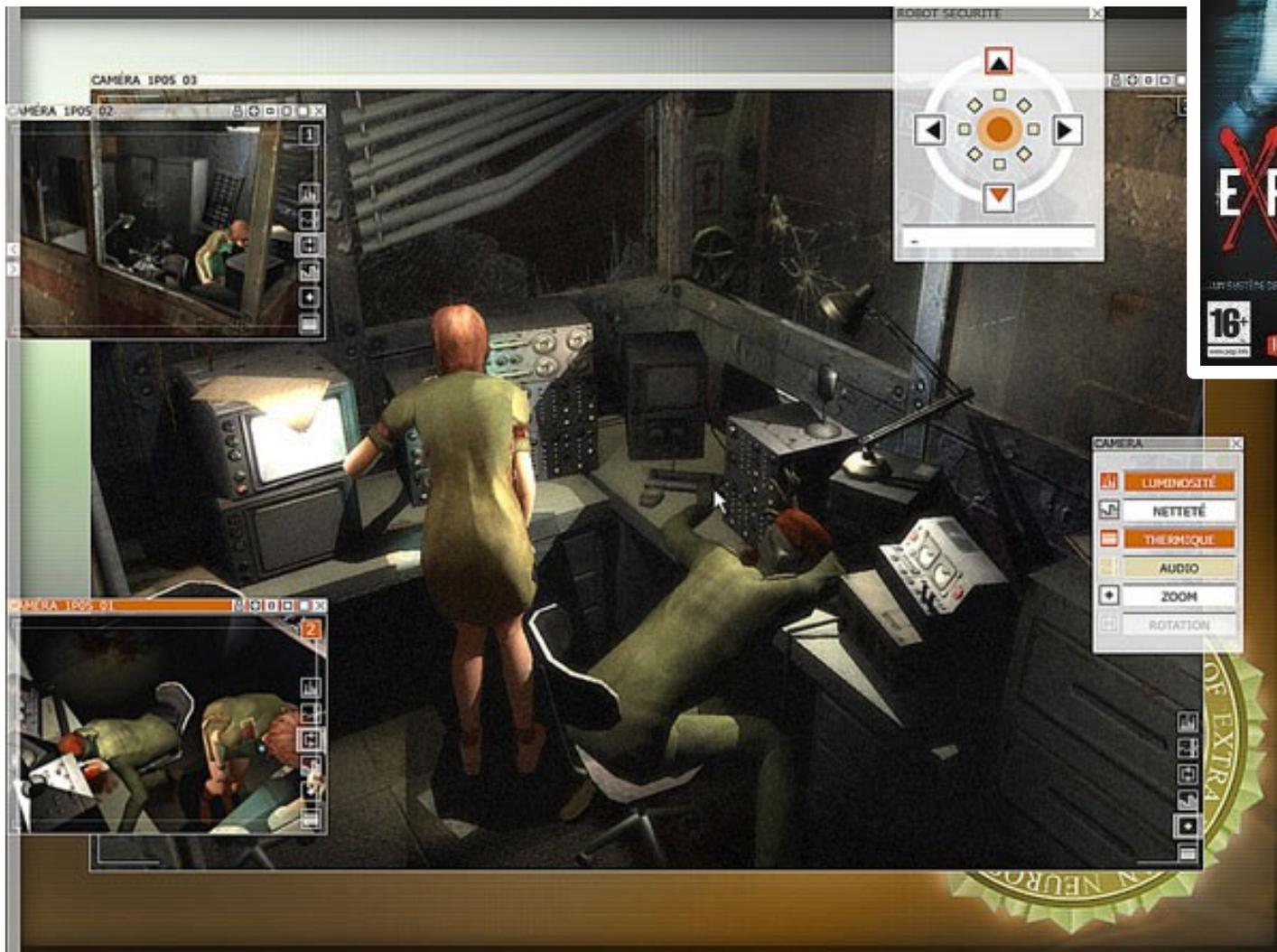
43
244

(2005)





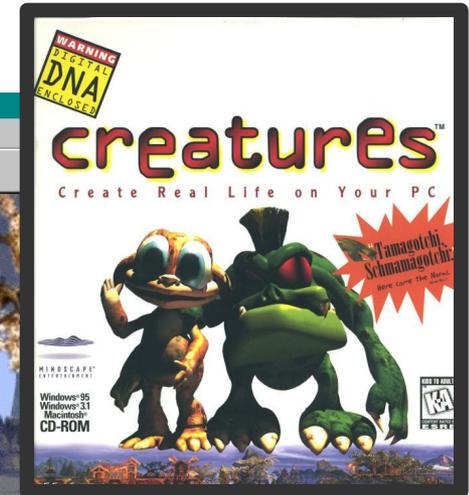
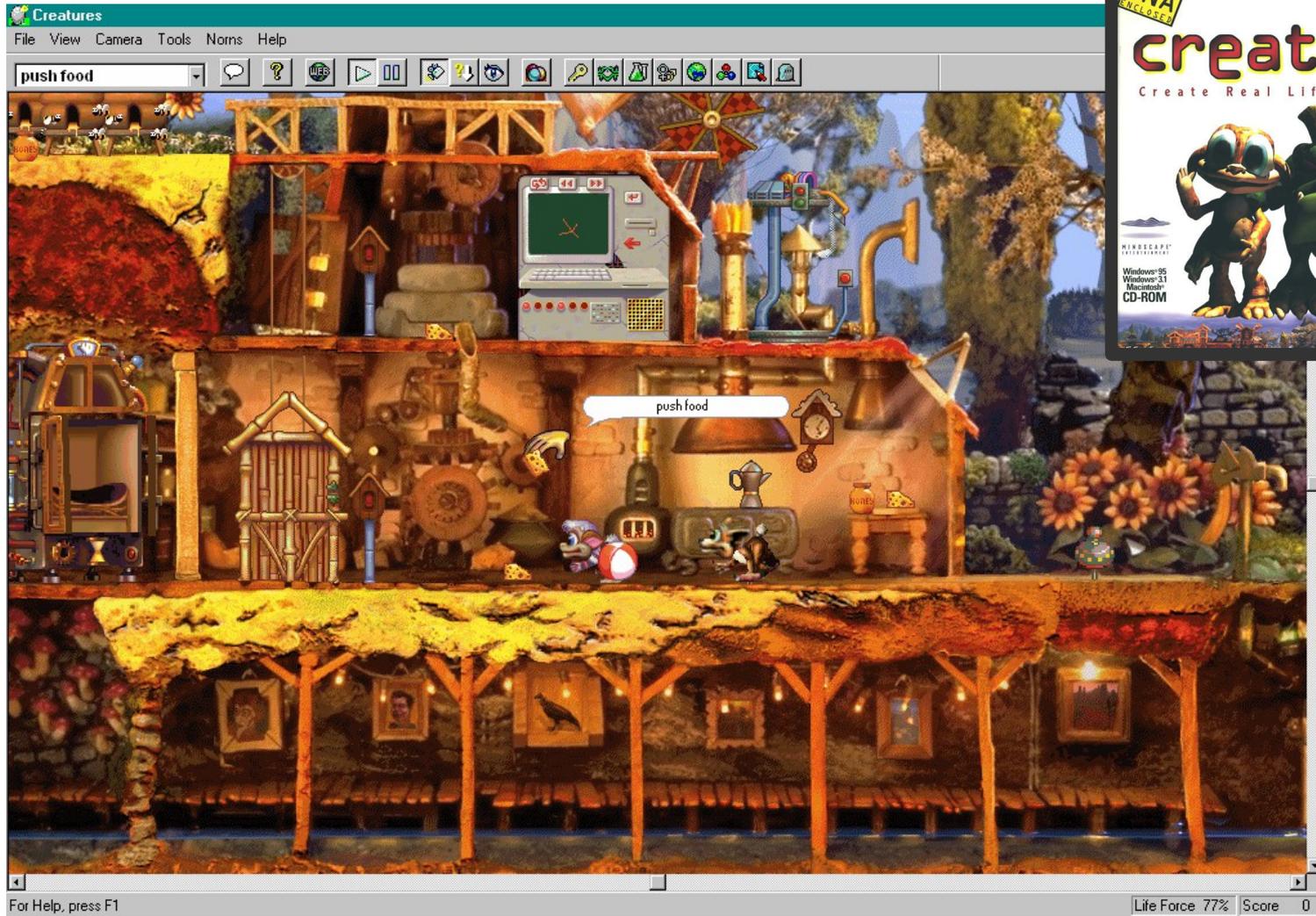
Obj 6 : IA comme coeur de jeu



(2007)

Obj 6 : IA comme coeur de jeu

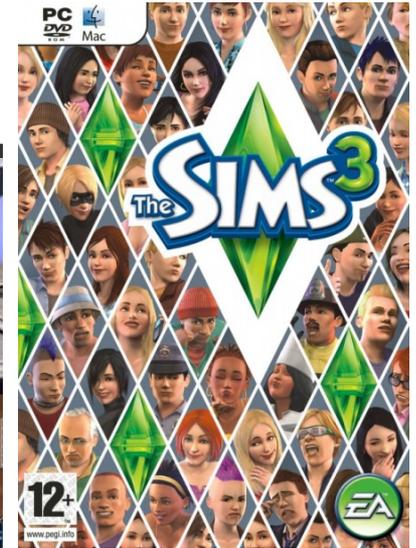
45
244



(1996)

Obj 6 : IA comme coeur de jeu

46
244



(2009)

- Qu'est ce que l'IA ?
- Objectif de l'IA dans un jeu video
- Exemple IA et techniques
 - Comportement réflexe
 - Planification
 - Décision haut niveau
 - Apprentissage

Comment faire une IA ?

- Cela dépend du but
- Quel est l'objectif de l'IA
 - Proposer un monde complexe et cohérent (JDR)
 - Proposer des ennemis intelligents ? (FPS..)
 - Le but du joueur ? (créature, ..)
 - Un adversaire conséquent ?
- Conséquence
 - Choisir le type d'IA adapté : pas trop complexe

Du tower defense ...

50
244



Revenge of the titans

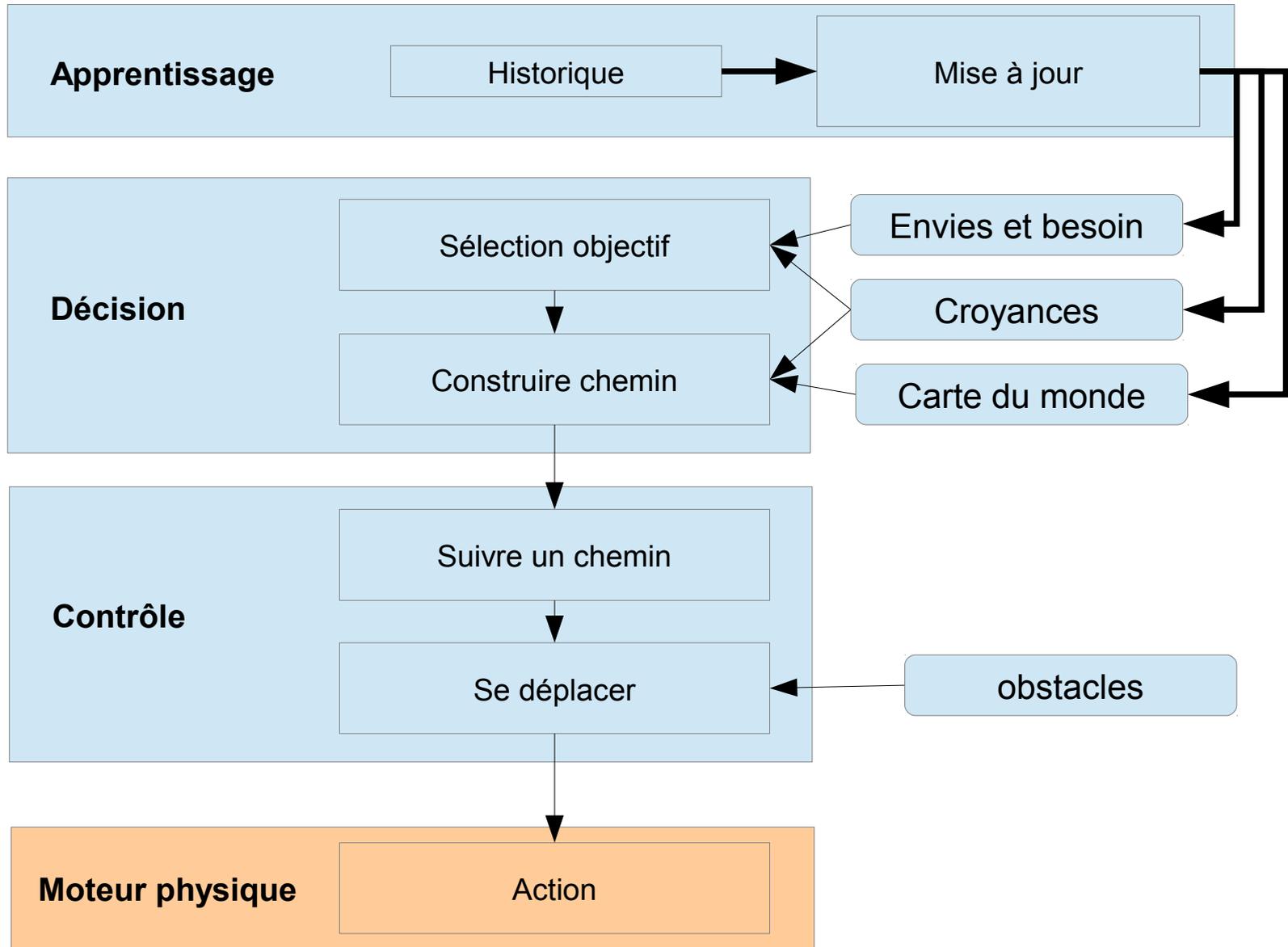
A league of legends

51
244



- **Tower defense**
 - Savoir se déplacer
 - Savoir suivre un chemin
 - Évitement d'obstacles
 - Planifier une trajectoire
- **League of legends**
 - Savoir s'adapter aux circonstances
 - Comportement de groupe
 - Émuler un comportement (tufluc)
 - Apprentissage (GACS)

Organisation d'une IA complexe





1 – couche action



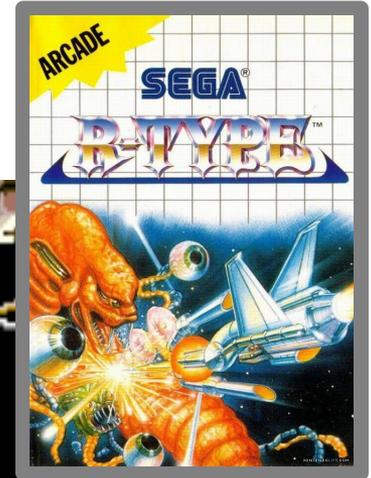
Moteur physique

Action

- **Objectif**
 - IA simple pour des entités simples
 - Pas de surprise
- **Exemple**
 - Se diriger vers le personnage le plus proche
 - Suivre un trajet
- **Défis pour le joueur**
 - Intuiter les déplacements futurs
 - Complexité vient du nombre

Architecture purement réactive

57
244



(1987)

- **Déplacement direct**
 - Rapprocher de la cible

```
If (cible.x < x)      x = x - 1;  
                    else x = x + 1;  
If (cible.y < y)     y = y - 1;  
                    else y = y + 1;
```



Architecture purement réactive

(1999)



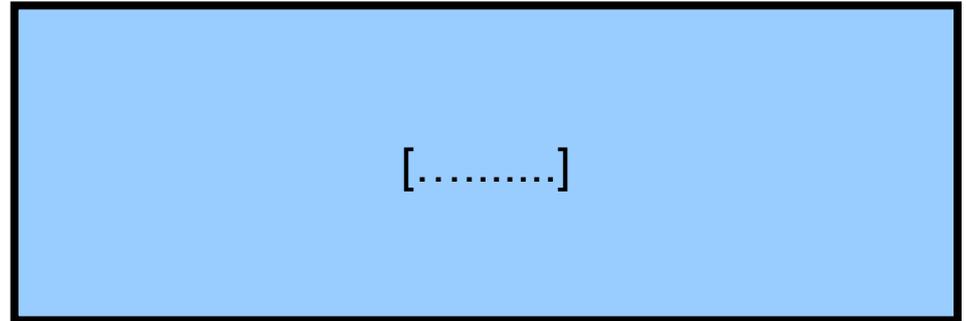
- **Déplacement direct**
 - Algorithme de base

If (devant.vide) alors avance
sinon si (gauche.vide) tourneGauche
sinon si (droite.vide) tourneDroite
sinon demi-tour.

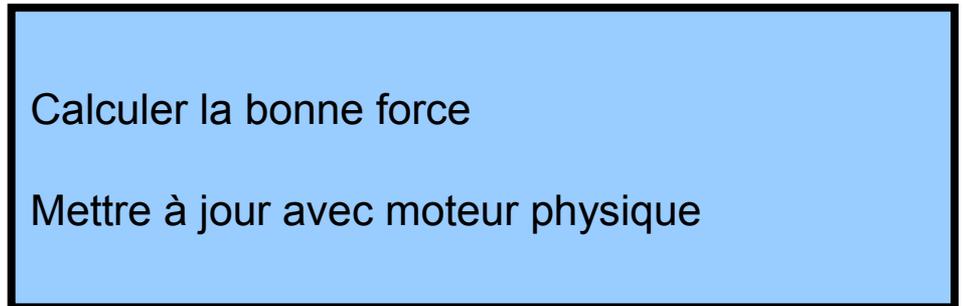


2.1 - Savoir se déplacer

- **Déplacement direct**
 - Rapprocher de la cible
 - Algo chuchurocket



- **Boids**
 - Utilisant des forces
 - Utilisant des potentiels



- Plusieurs possibilités
 - Action exécutée directement



```
If (droite)      x=x-1;  
                 else x=x+1;  
If (gauche)     y=y--1;  
                 else y=y++1;
```

- Plusieurs possibilités

- Action exécutée directement



```
If (droite)      x=x-1;
                  else x=x+1;
If (gauche)     y=y--1;
                  else y=y++1;
```

- Moteur avec dynamique



```
x=x+vx
y=y+vy

vx=vx+ax
vy=vy+ay
```

- Moteur avec une dynamique



- **Avantage**
 - Réutiliser précédemment
 - Réalisme
- **Inconvénient**
 - Contrôle accélération
 - Se déplacer plus si simple

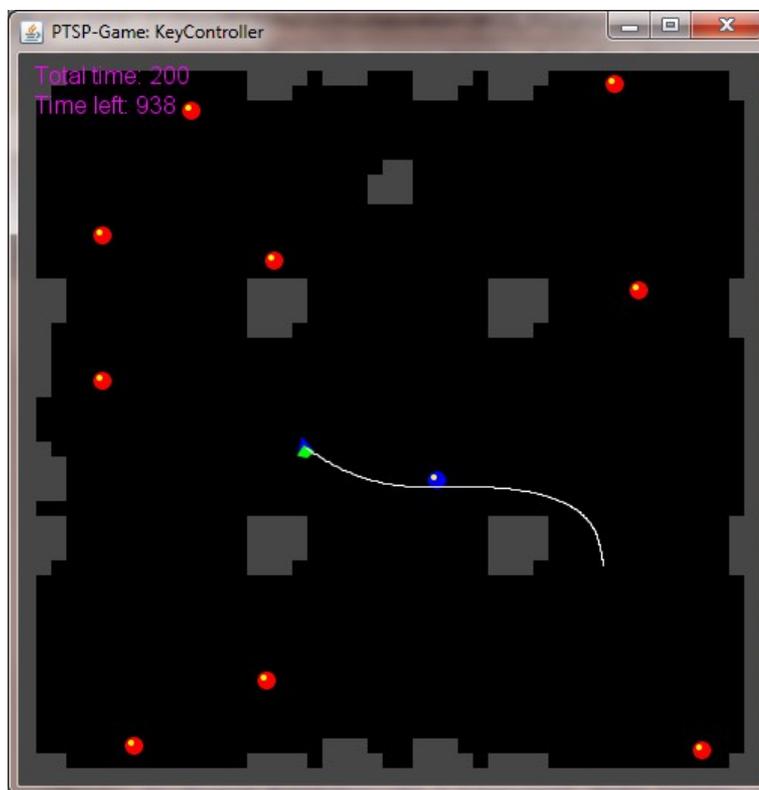
1 – couche action

- Compétition IA : <http://www.marioai.org/>





- Différences
 - Voyageur de commerce
 - Physical TSP (<http://www.ptsp-game.net/>)



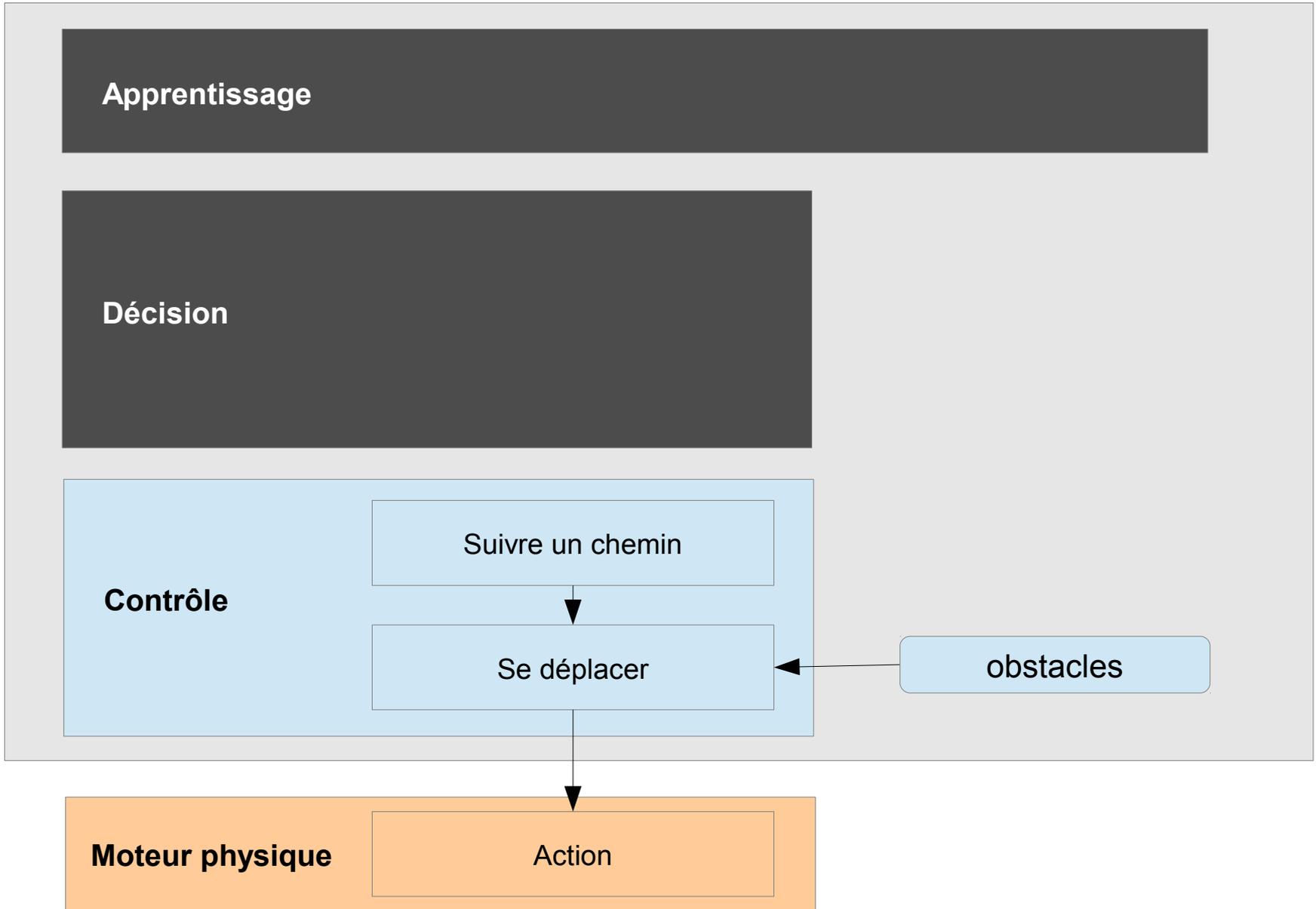
1 – couche action



Moteur physique

Action

2 – couche contrôle



- **Exemple**
 - Se diriger vers le personnage le plus proche
 - Éviter des petits obstacles

2.1 - Savoir se déplacer

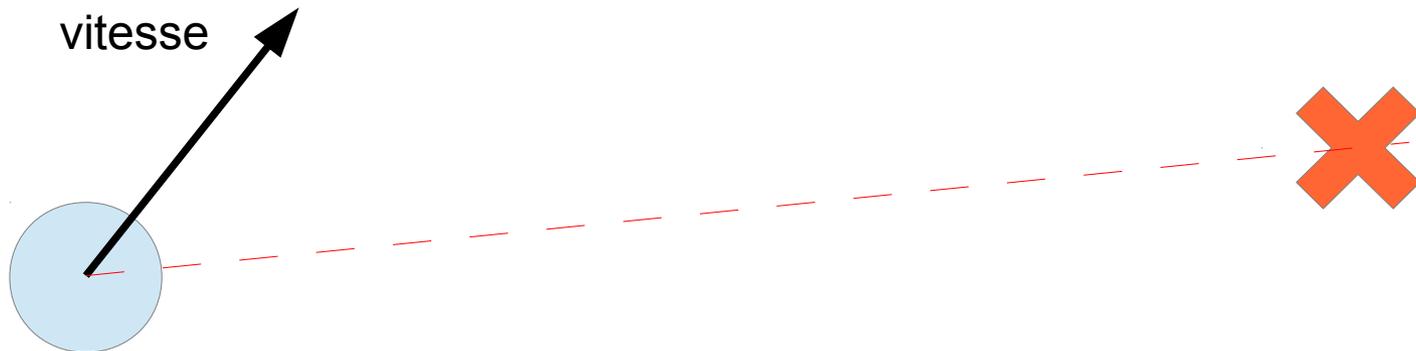
- **Exemple**
 - Se diriger vers le personnage le plus proche
 - Éviter des petits obstacles

- **Jeux**
 - FPS
 - Shoot-em-up
 - ...

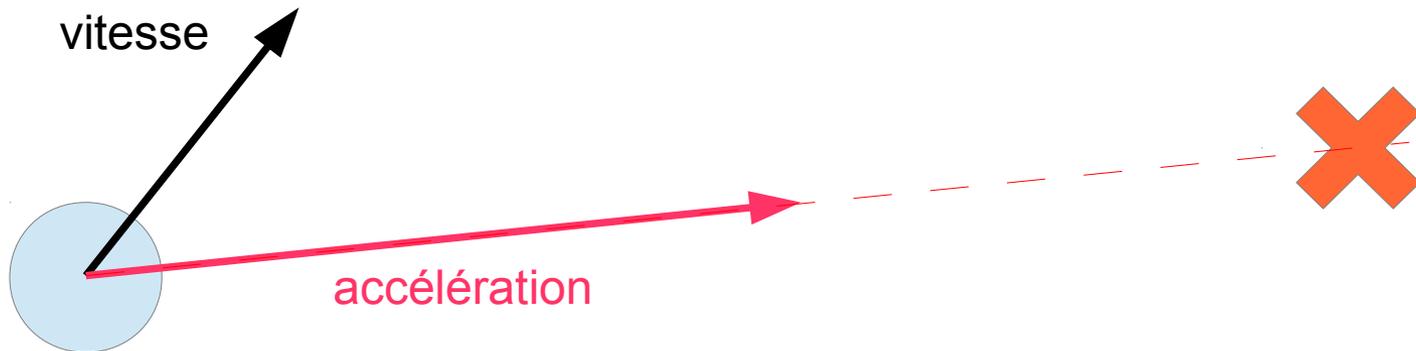


Rayman Origins (2012)

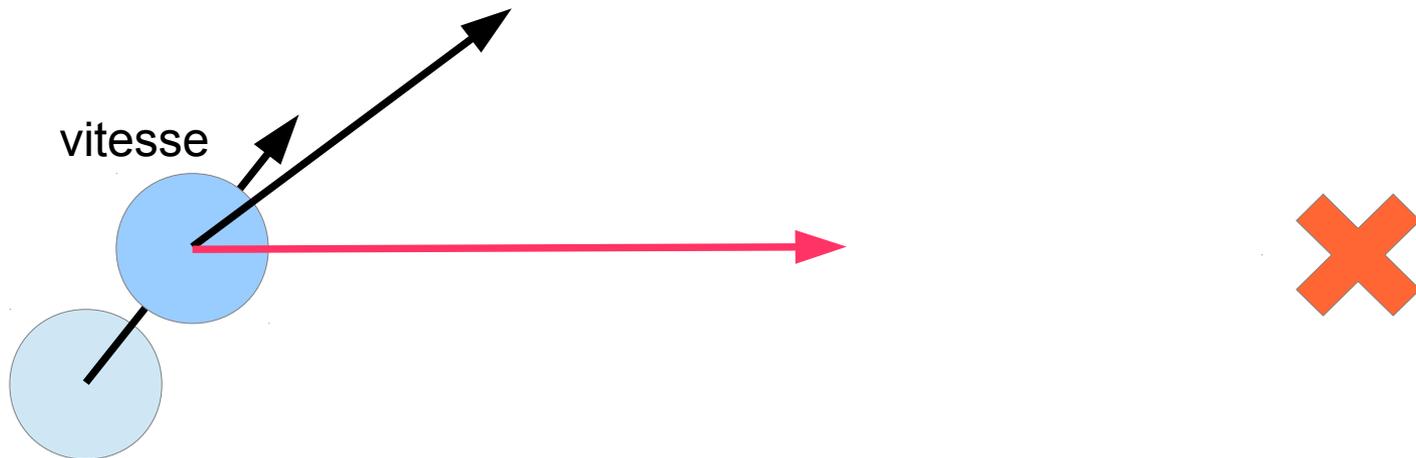
- **Boids**
 - Craig Reynolds



- **Boids**
 - Solution 1: accélération directe
 - $A_x = (b_x - p_x)$
 - $A_y = (b_y - p_y)$

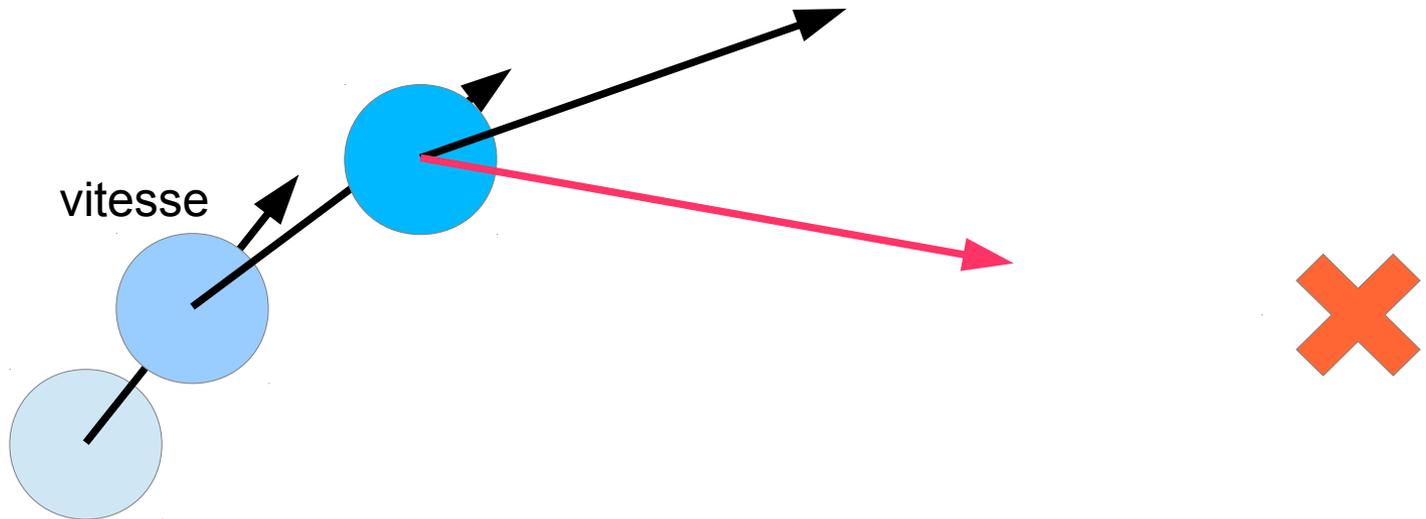


- **Boids**
 - Solution 1: accélération directe



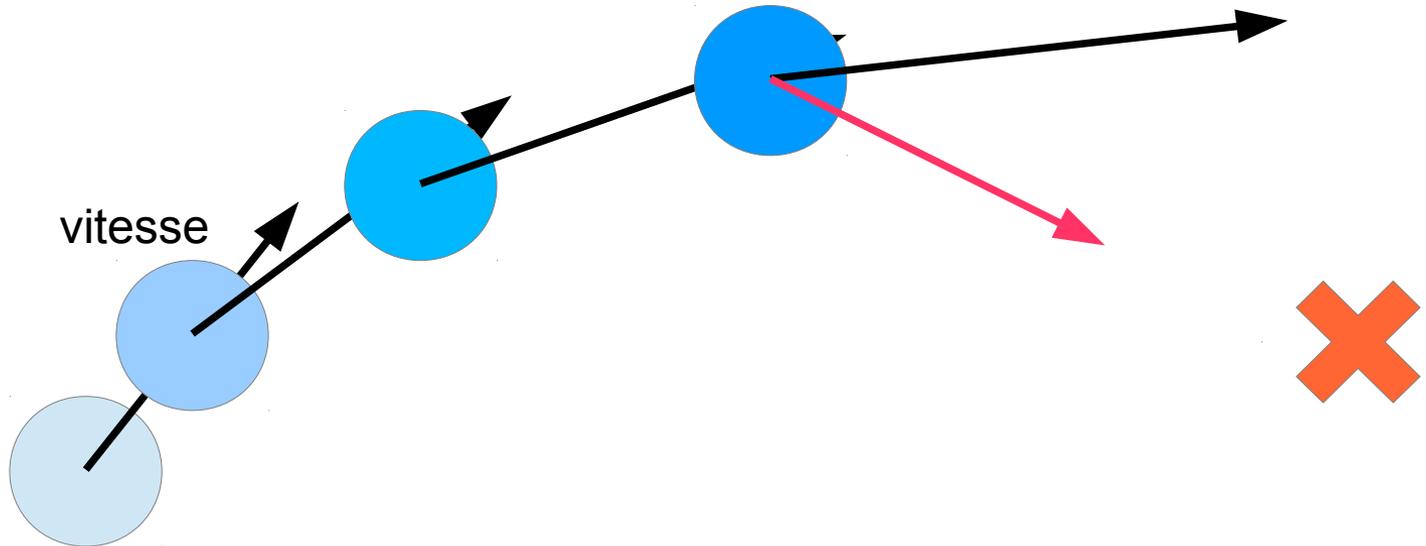
2.1 - Savoir se déplacer

- **Boids**
 - Solution 1: accélération directe



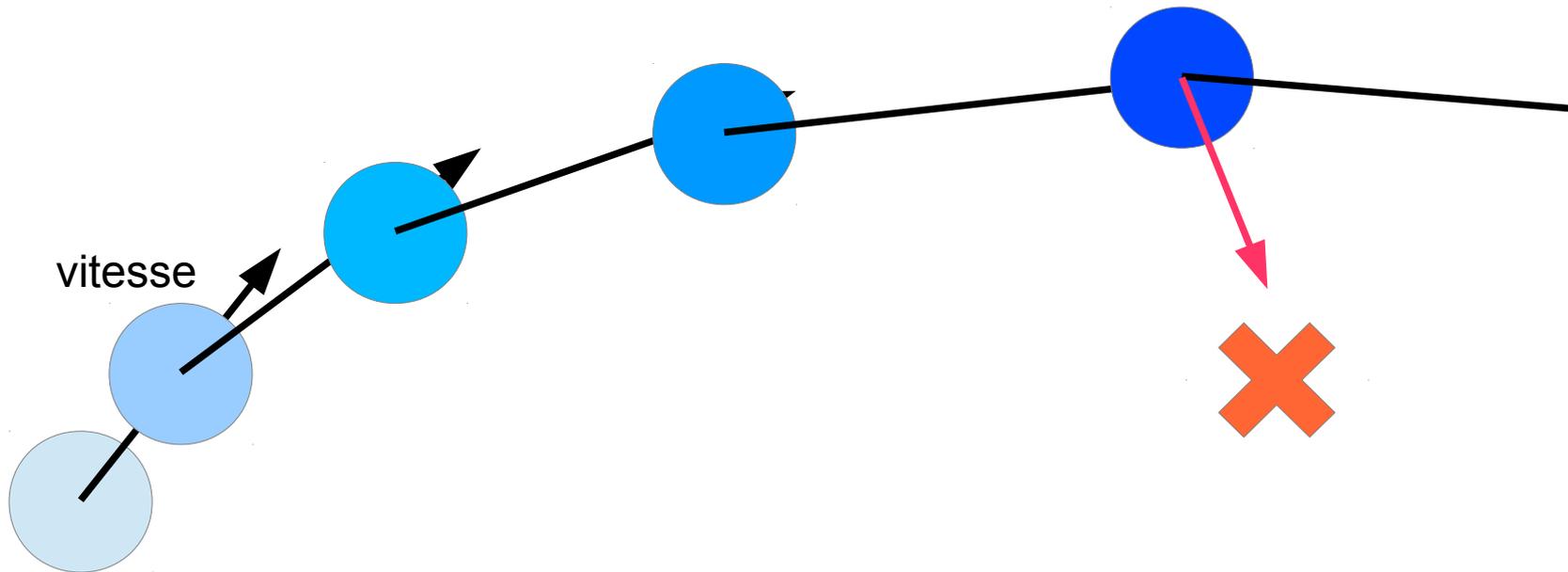
2.1 - Savoir se déplacer

- **Boids**
 - Solution 1: accélération directe



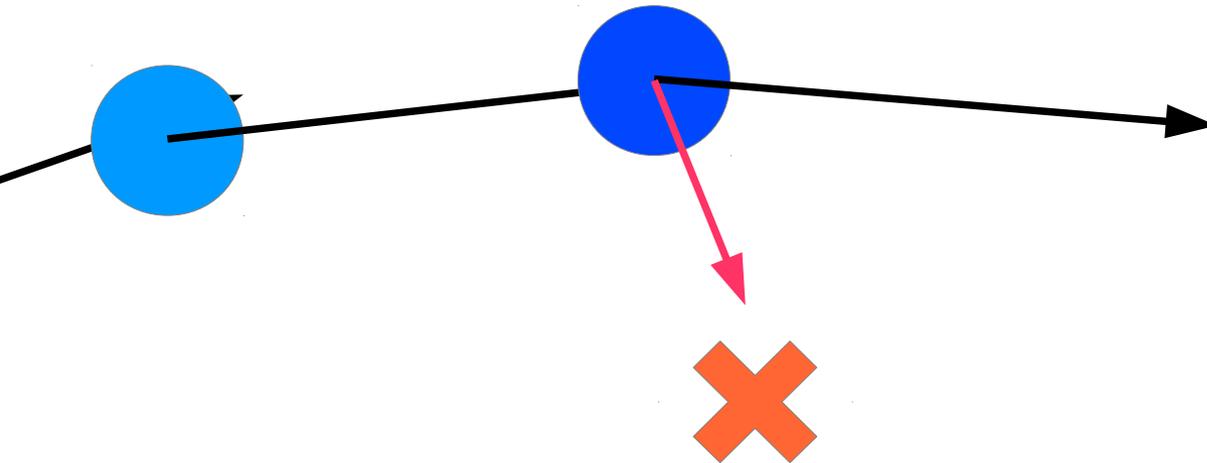
2.1 - Savoir se déplacer

- **Boids**
 - Solution 1: accélération directe



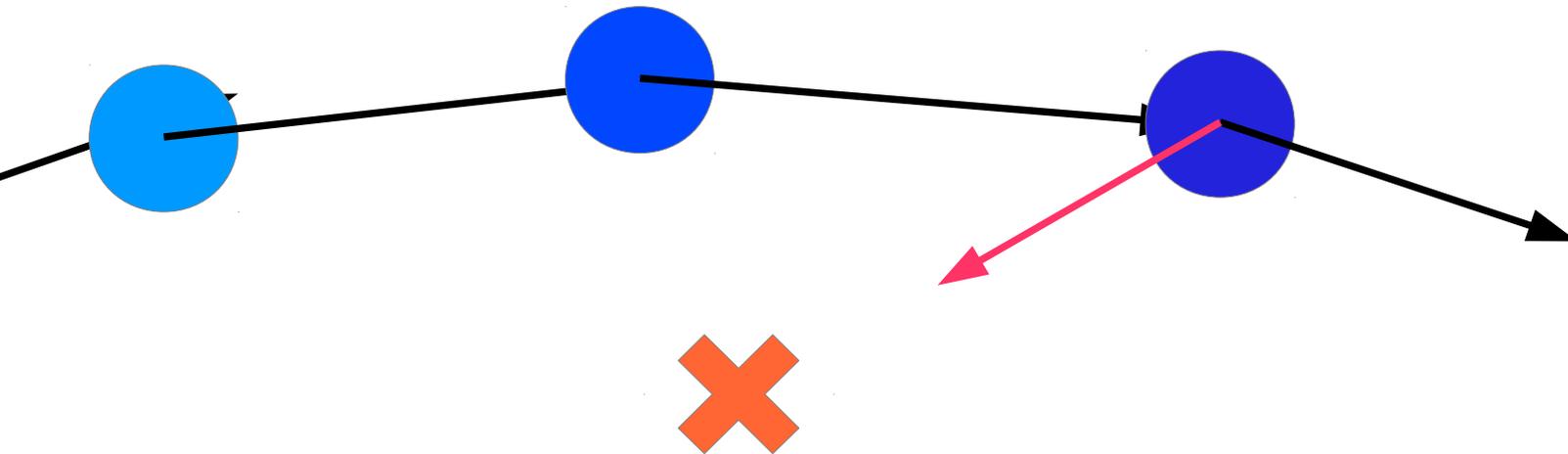
2.1 - Savoir se déplacer

- **Boids**
 - Solution 1: accélération directe

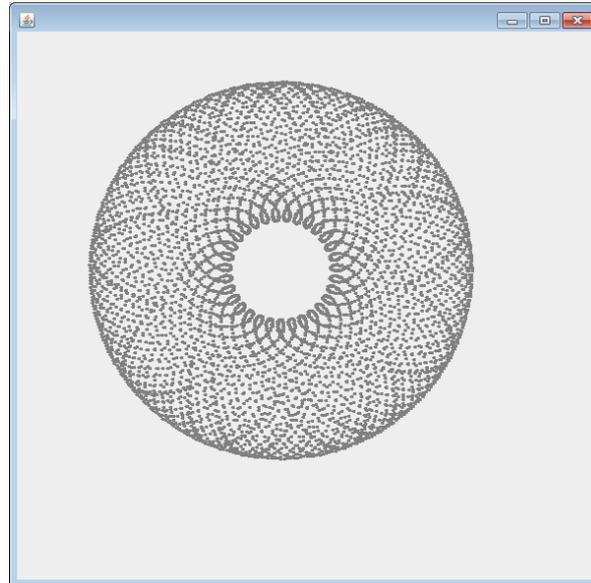
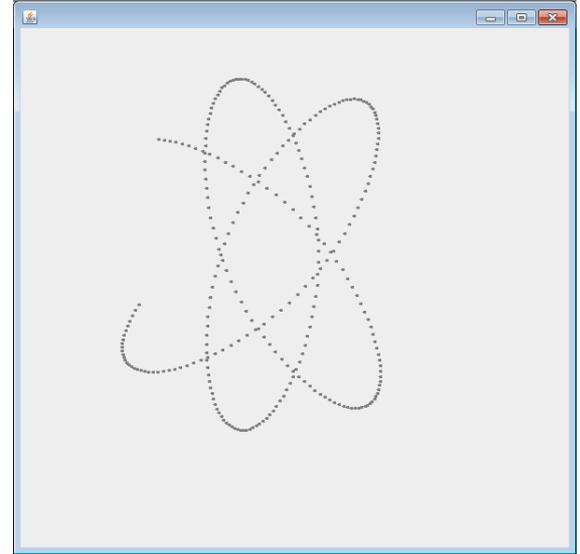
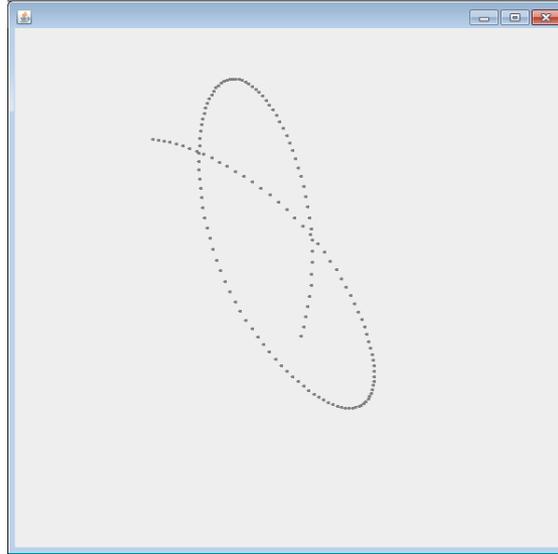
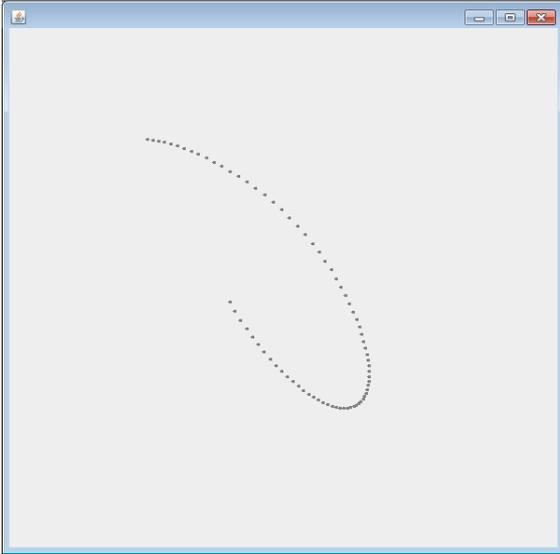


2.1 - Savoir se déplacer

- **Boids**
 - Solution 1: accélération directe

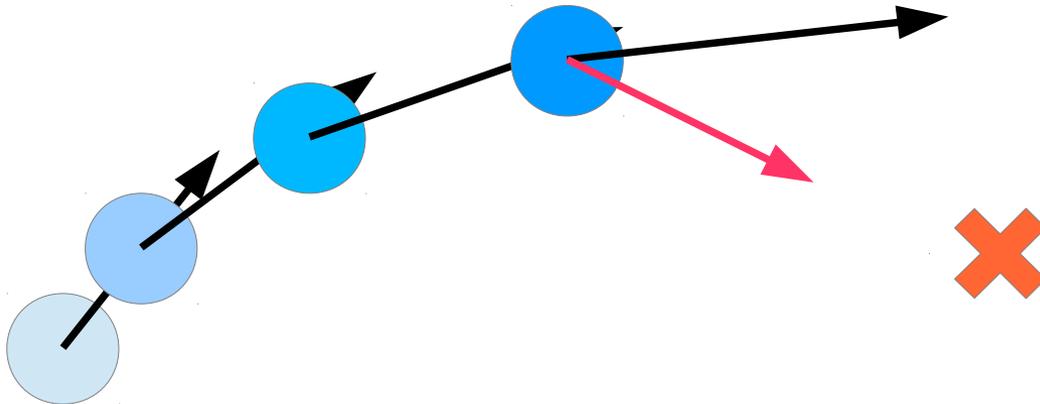


2.1 - Savoir se déplacer

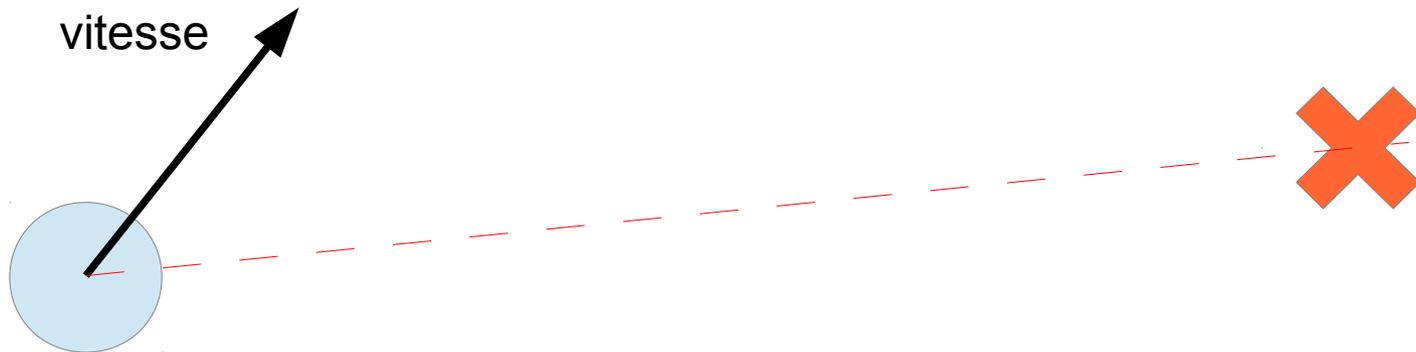


2.1 - Savoir se déplacer

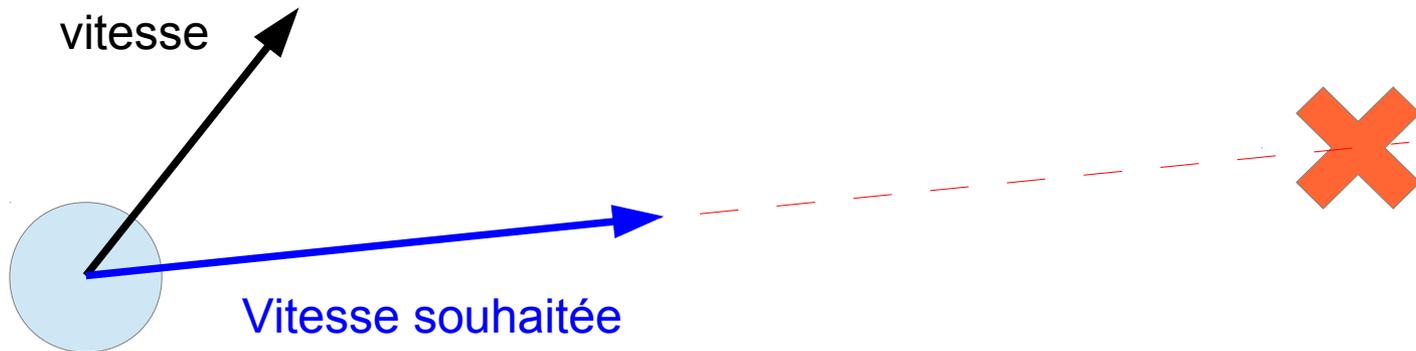
- **Boids**
 - Solution 1: accélération directe
- **Démo1**



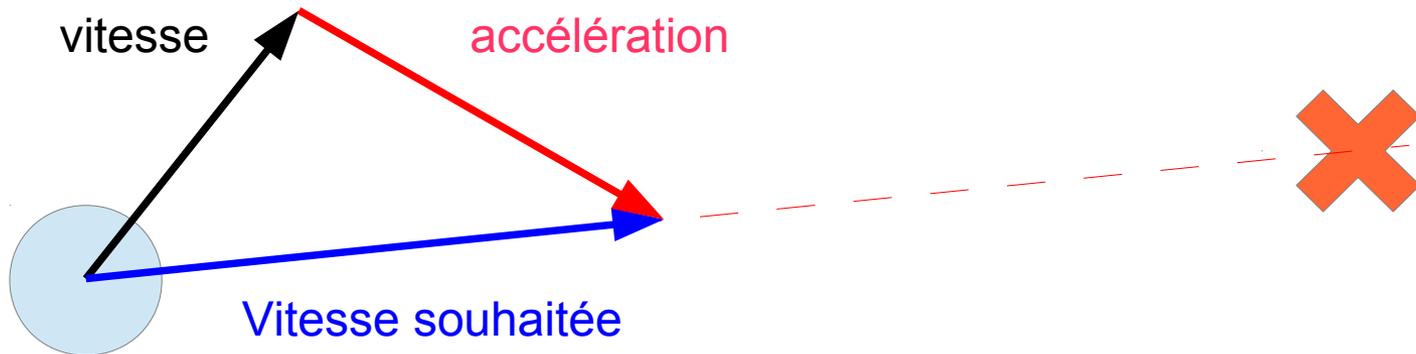
- **Boids**
 - Craig Reynolds



- **Boids**
 - Craig Reynolds
 - Sur la vitesse

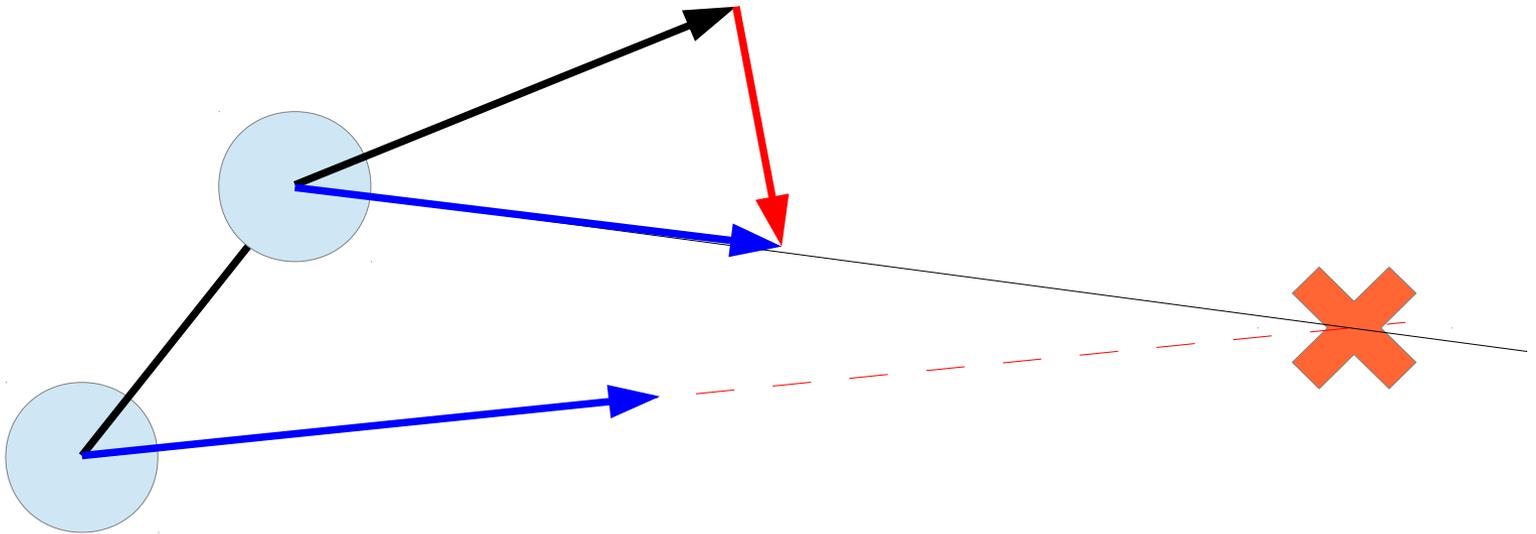


- **Boids**
 - Craig Reynolds
 - Sur la vitesse



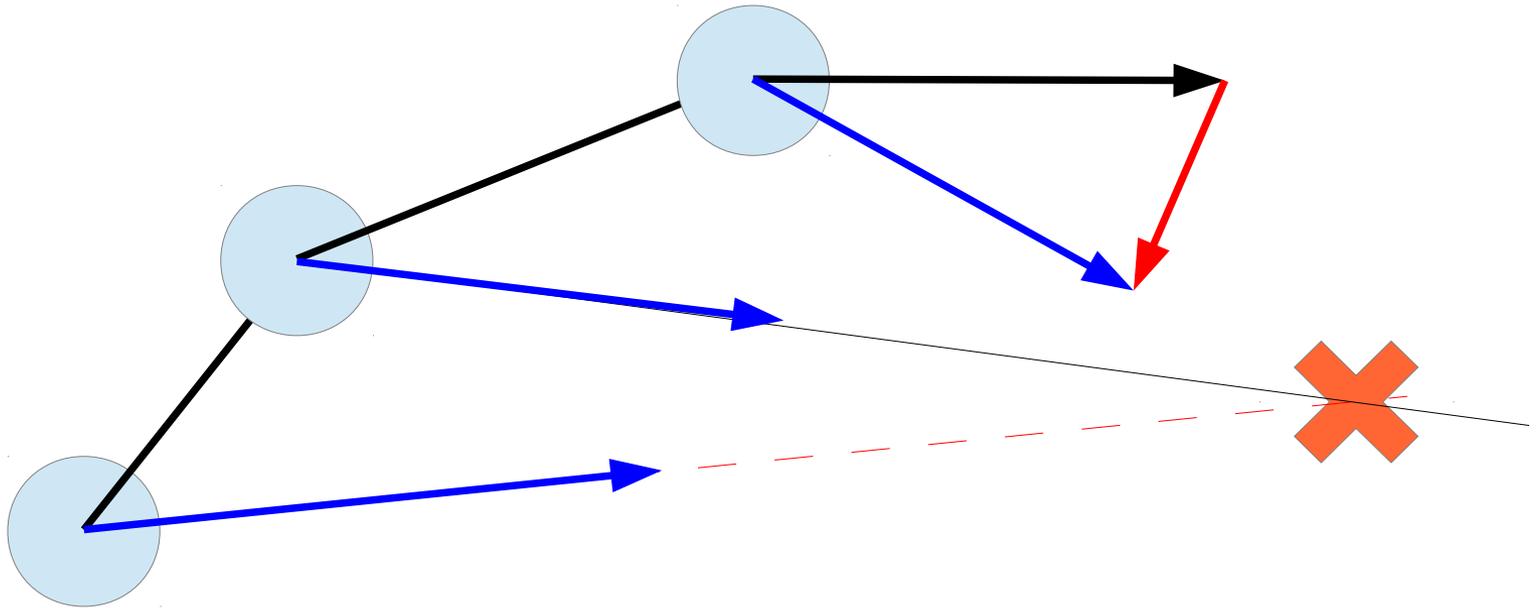
2.1 - Savoir se déplacer

- **Boids**
 - Craig Reynolds
 - Sur la vitesse

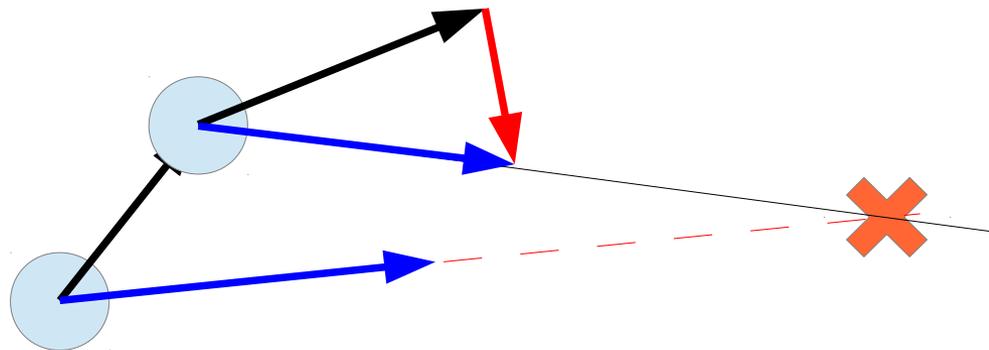


2.1 - Savoir se déplacer

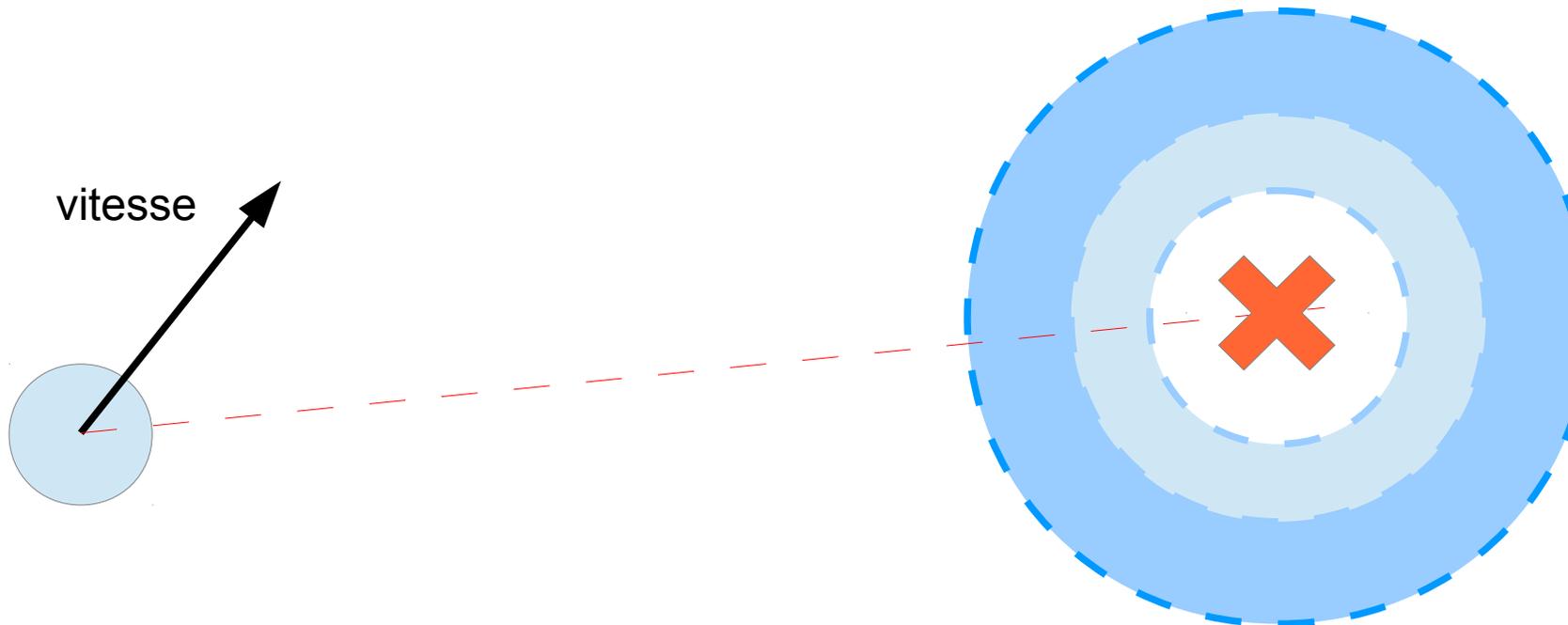
- **Boids**
 - Craig Reynolds
 - Sur la vitesse



- **Boids**
 - Craig Reynolds
 - Sur la vitesse
- **Démo2 - AgentSeek**

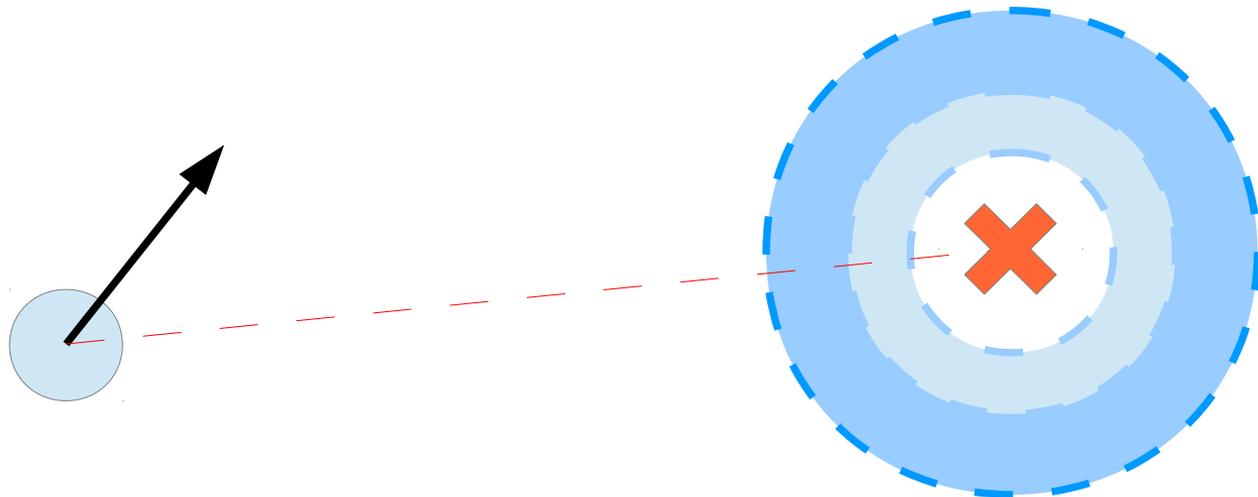


- **Boids**
 - Craig Reynolds
 - Ralentissement à l'arrivée

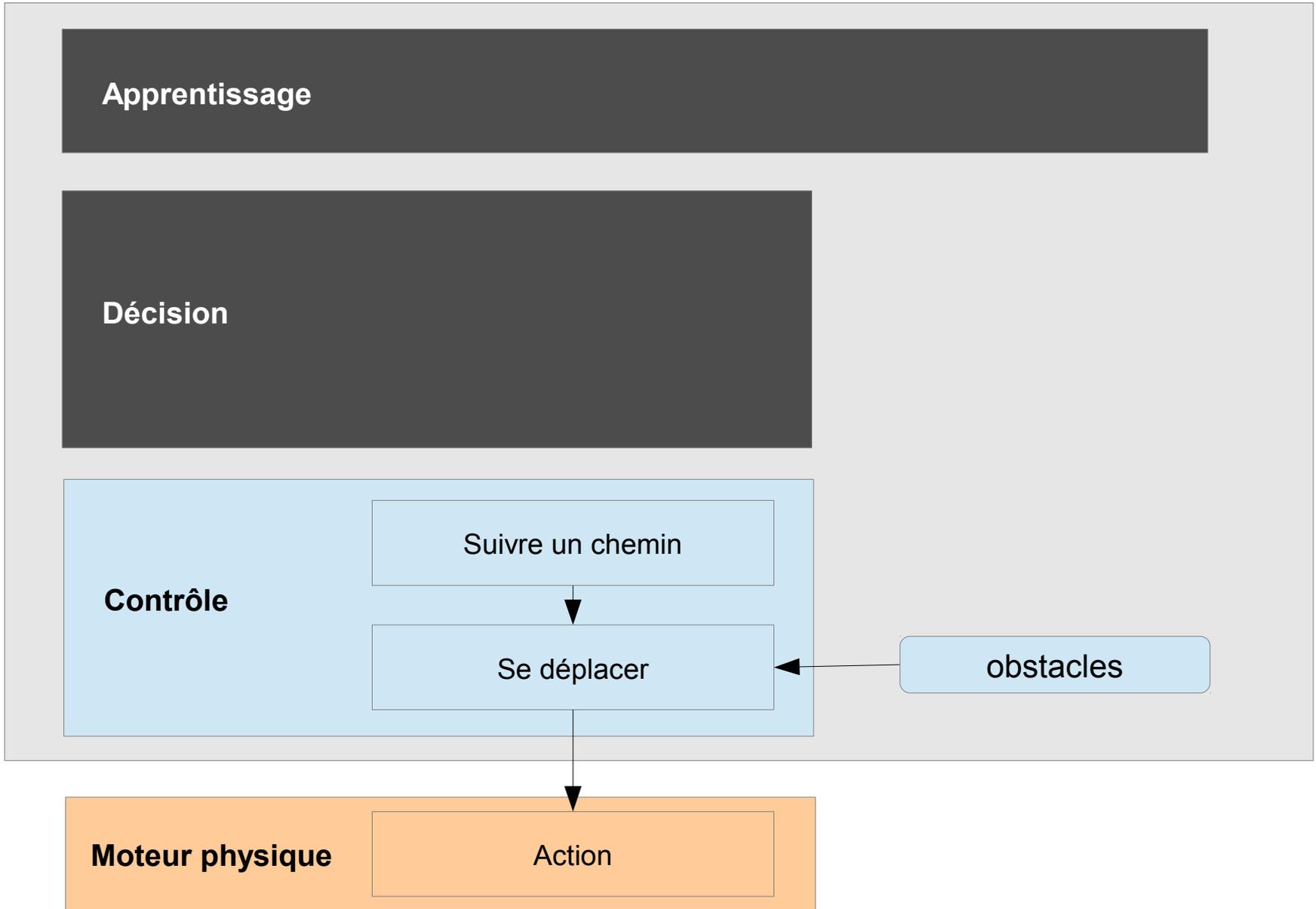


2.1 - Savoir se déplacer

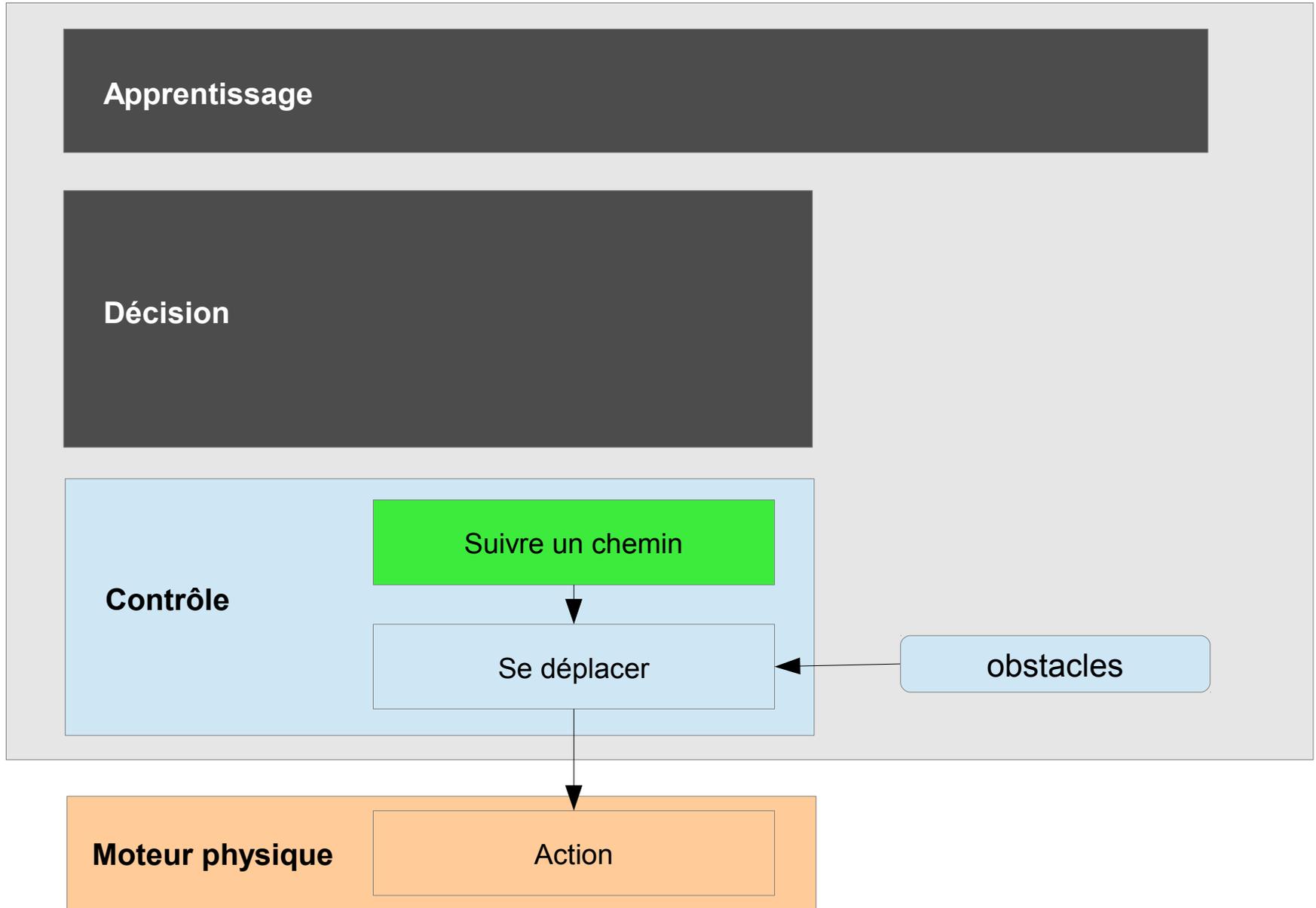
- **Boids**
 - Craig Reynolds
 - Ralentissement à l'arrivée
- **Démo3 - AgentArrival**



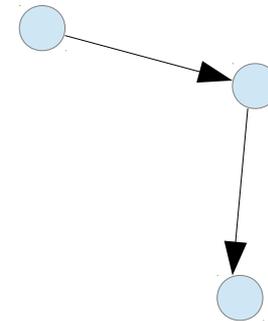
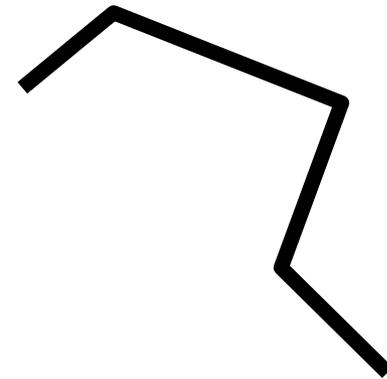
2 – couche contrôle



2 – couche contrôle

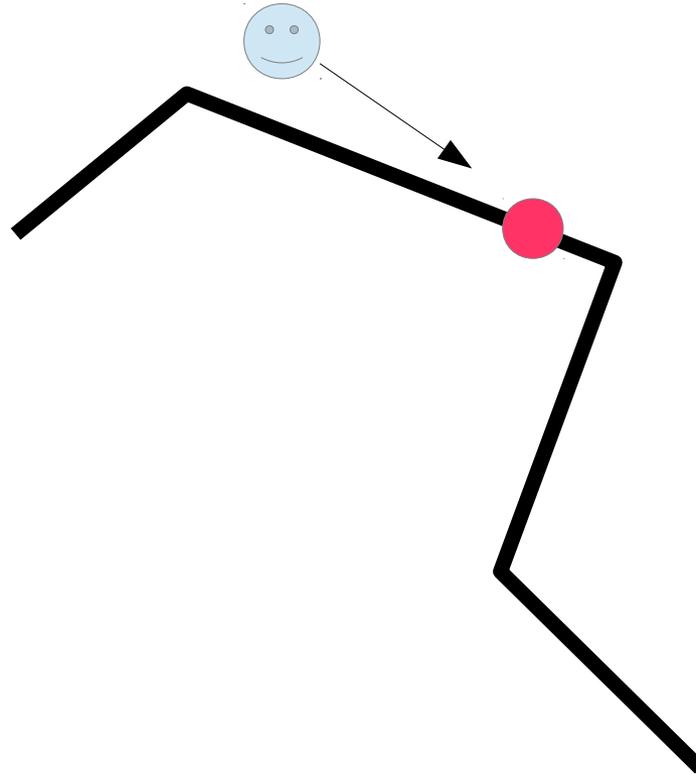


- **Plusieurs possibilités**
 - Suivre un chemin continu
 - Suivre un chemin discret
- **Pas tout à fait la même chose**
 - Cf angles



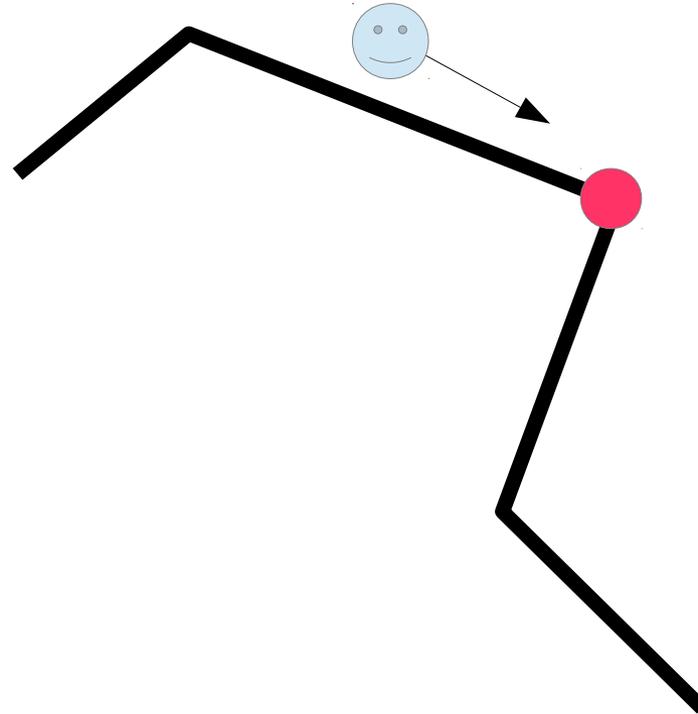
2.2 - Suivre un chemin

- Suivre un chemin continu



2.2 - Suivre un chemin

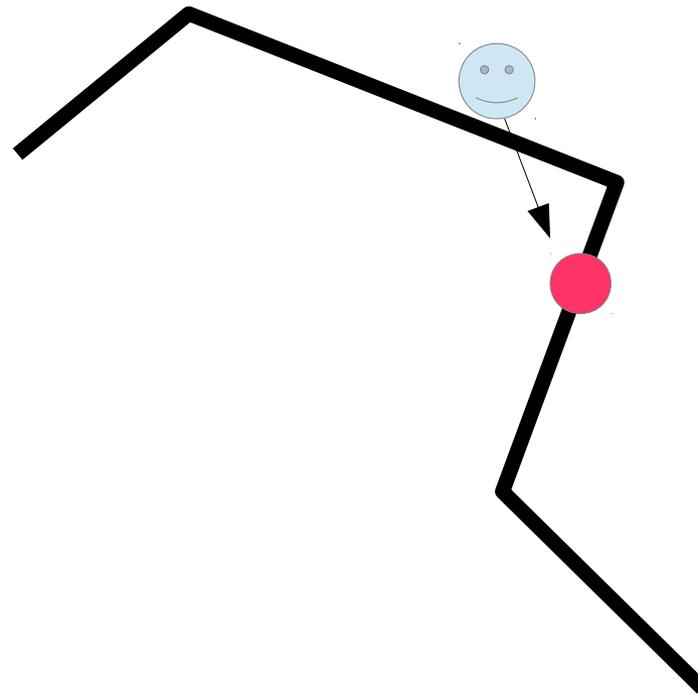
- Suivre un chemin continu



2.2 - Suivre un chemin

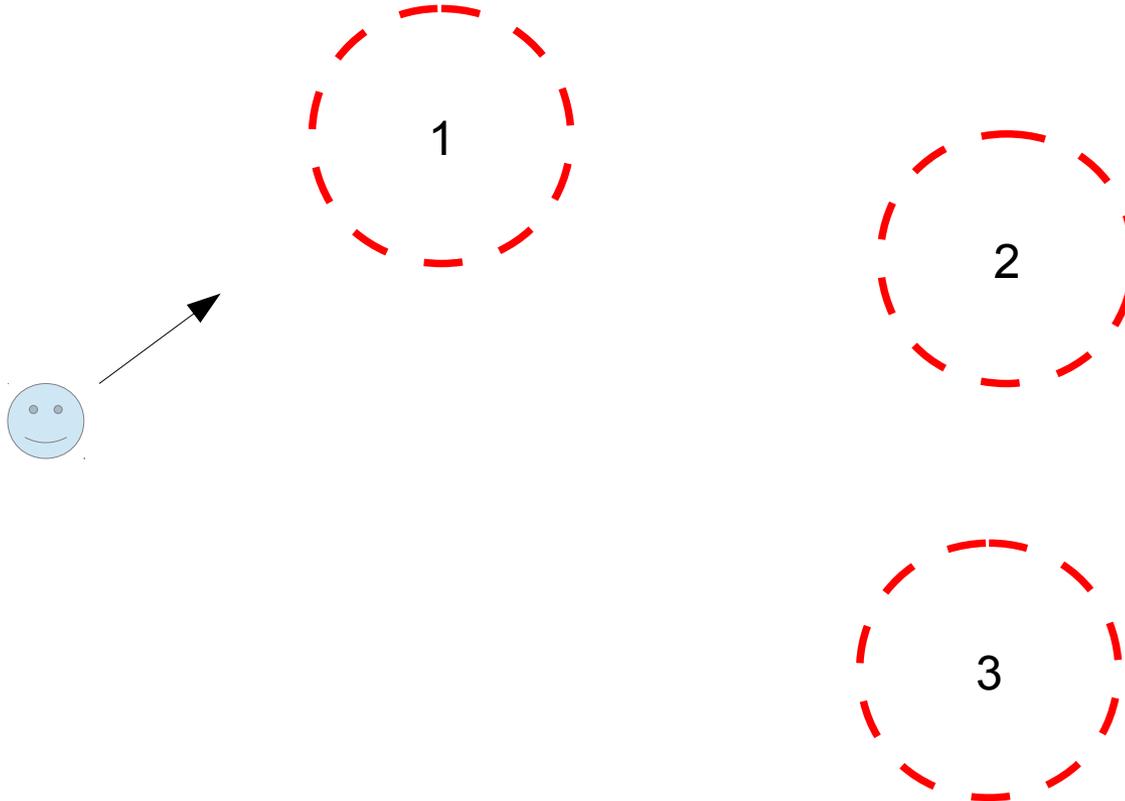
94
244

- Suivre un chemin continu
 - Point de contrôle (coordonnées linéaires)



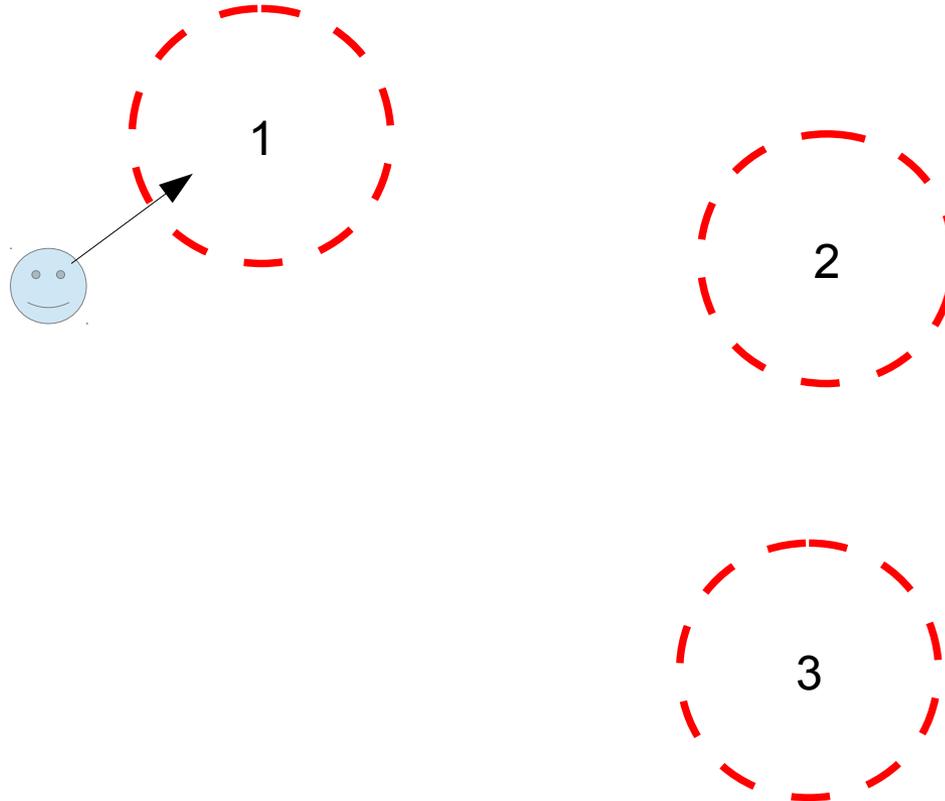
2.2 - Suivre un chemin

- Suivre un chemin discret
 - Liste de Points de contrôle



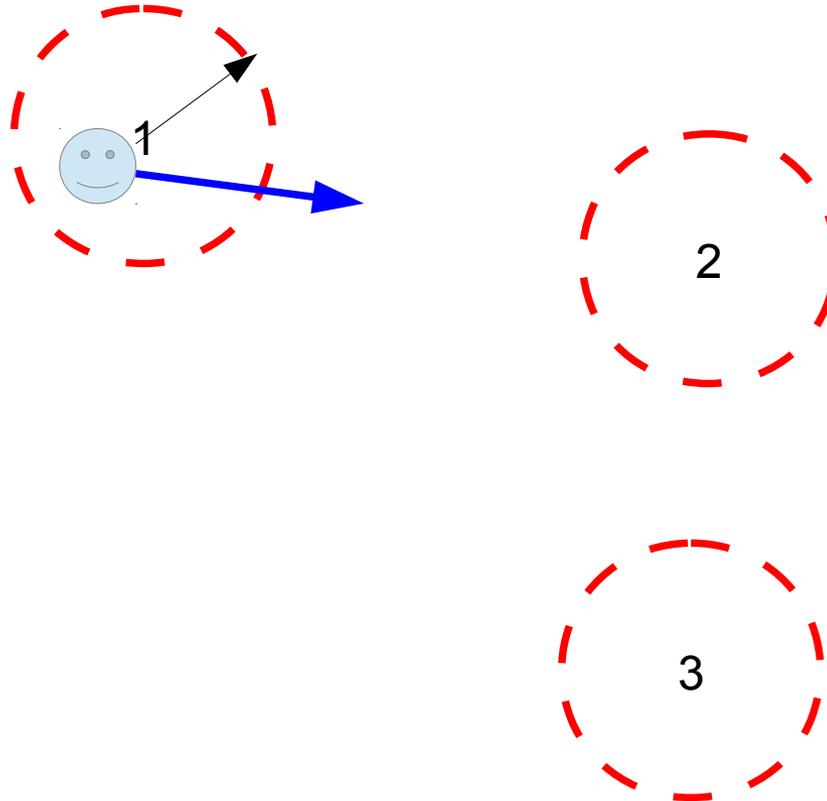
2.2 - Suivre un chemin

- Suivre un chemin discret
 - Liste de Points de contrôle



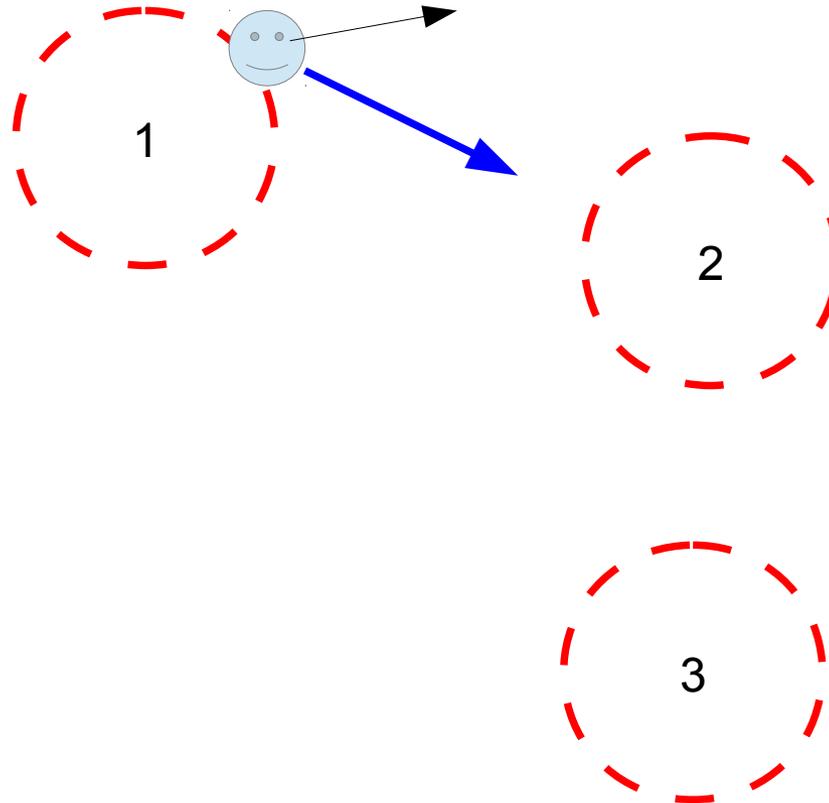
2.2 - Suivre un chemin

- Suivre un chemin discret
 - Liste de Points de contrôle



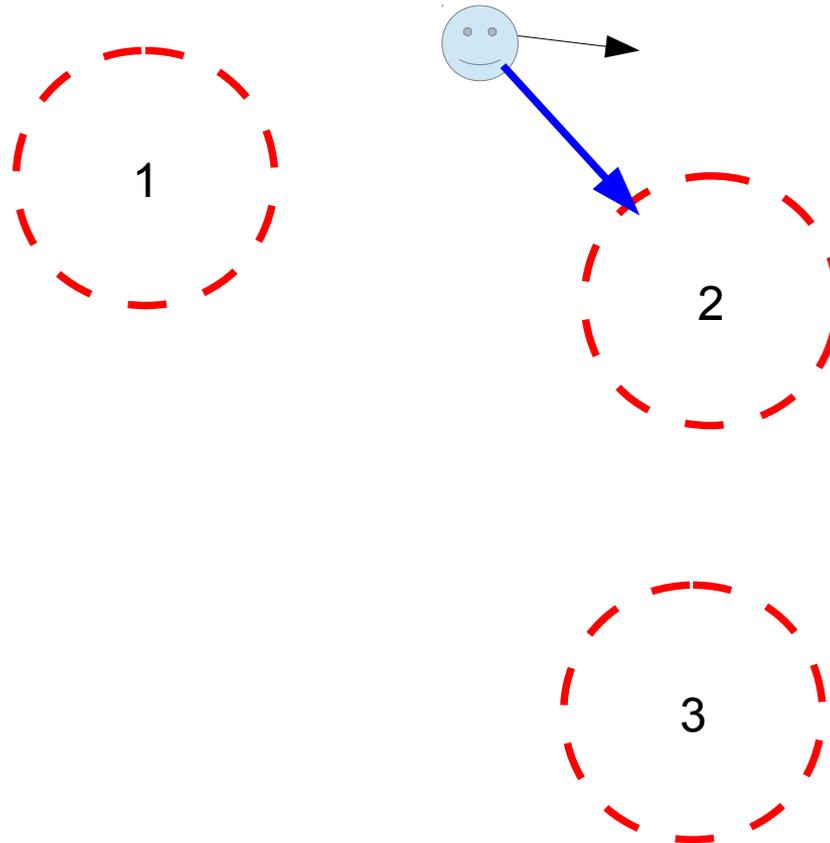
2.2 - Suivre un chemin

- Suivre un chemin discret
 - Liste de Points de contrôle



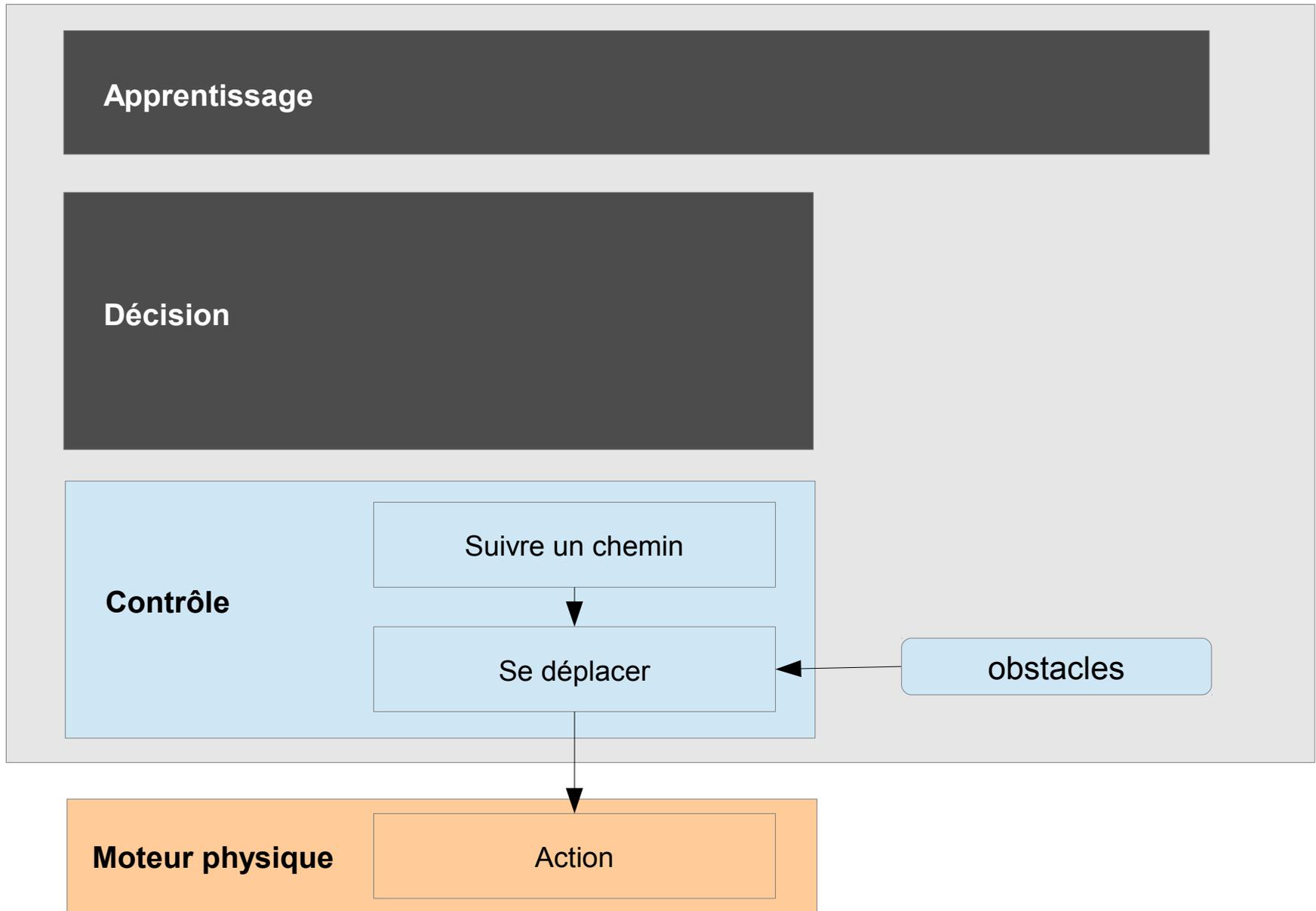
2.2 - Suivre un chemin

- Suivre un chemin discret
 - Liste de Points de contrôle

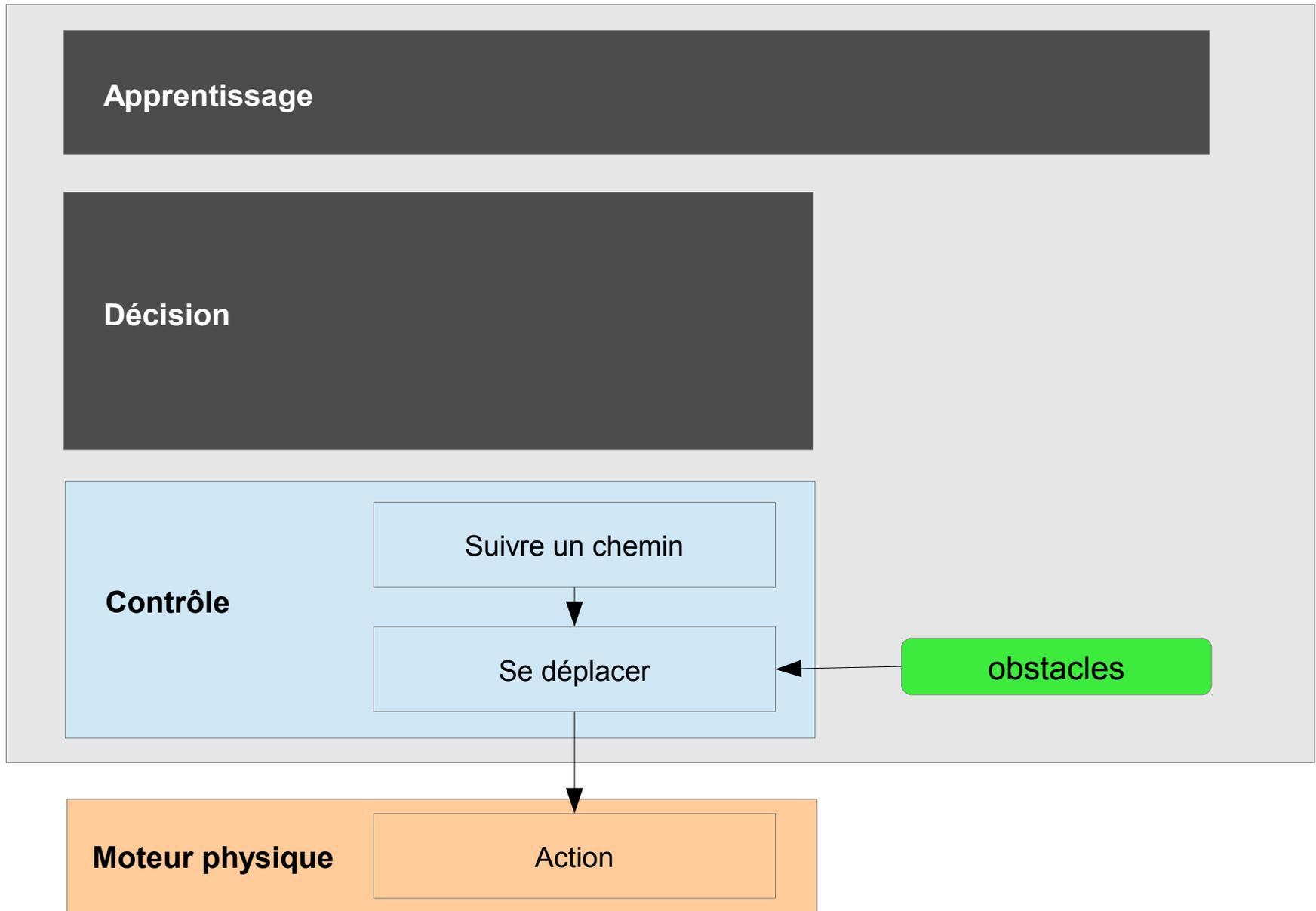


- Suivre un chemin discret
 - Liste de Points de contrôle
- Démo4 – suivre chemin

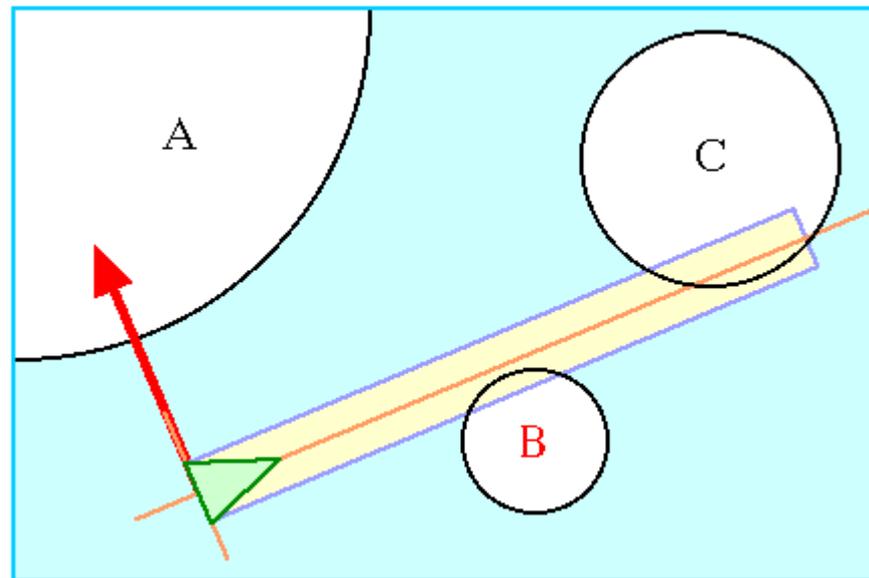
2 – couche contrôle



2 – couche contrôle



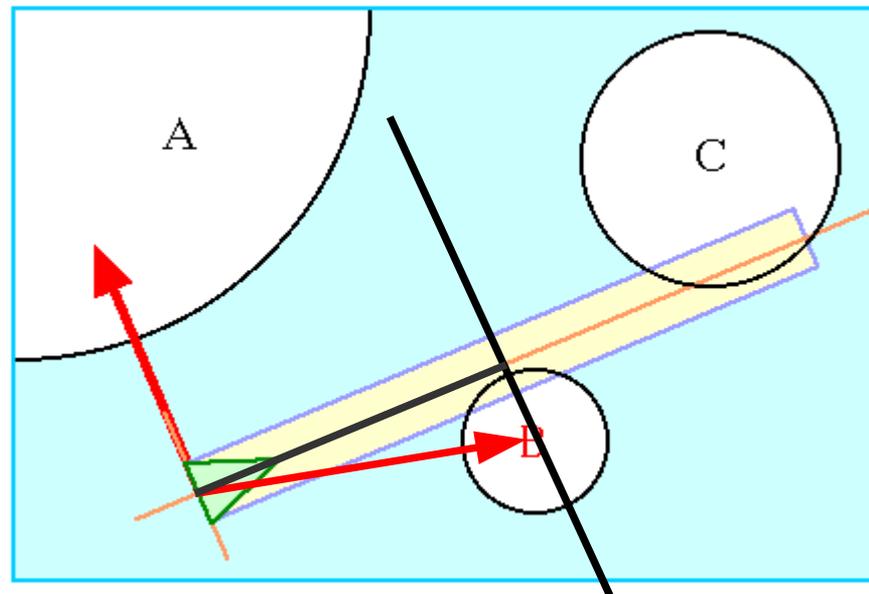
- Deux éléments
 - Détecter un obstacle
 - Éviter un obstacle



2.3 - Éviter des obstacles

104
244

- Deux éléments
 - Détecter un obstacle (vitesse et vecteur obstacle)

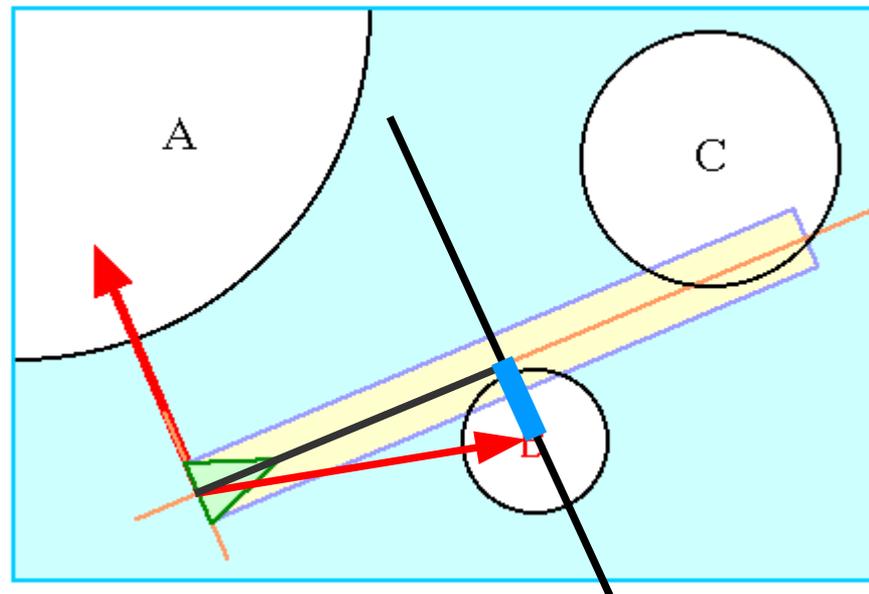


Craig Reynolds – Steering behaviours

2.3 - Éviter des obstacles

105
244

- Deux éléments
 - Détecter un obstacle (vitesse et vecteur obstacle)
 - $|\text{produit vectoriel}| > \text{taille obstacle} \rightarrow \text{Collision}$

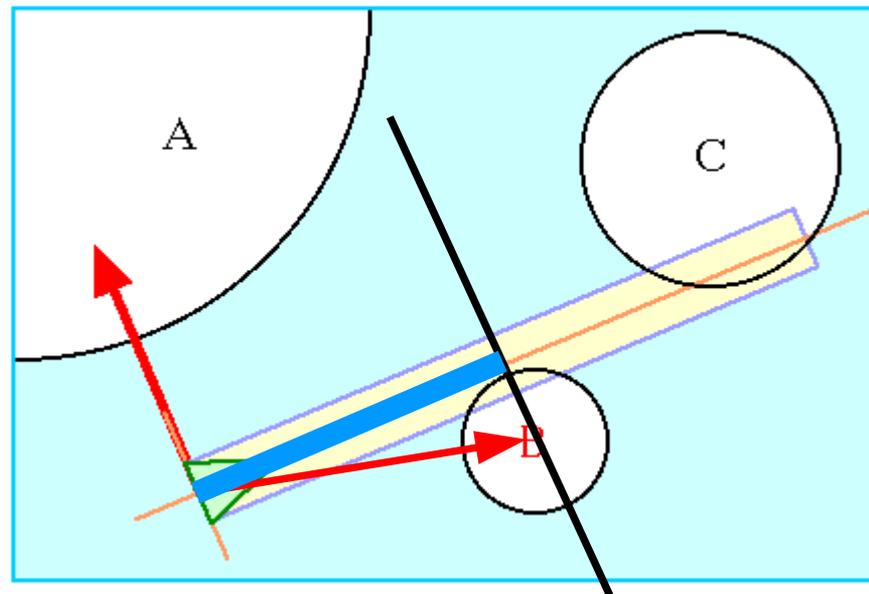


Craig Reynolds – Steering behaviours

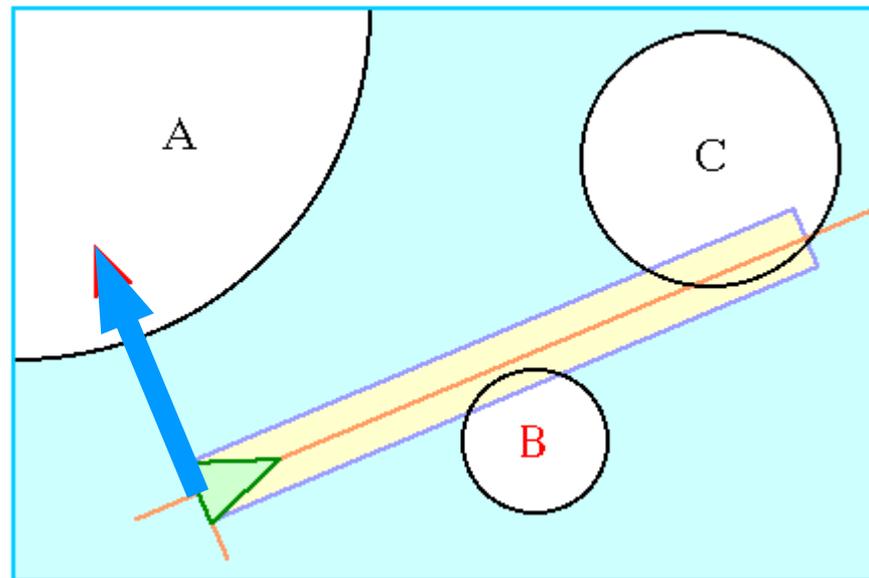
- Deux éléments

- Détecter un obstacle (vitesse et vecteur obstacle)

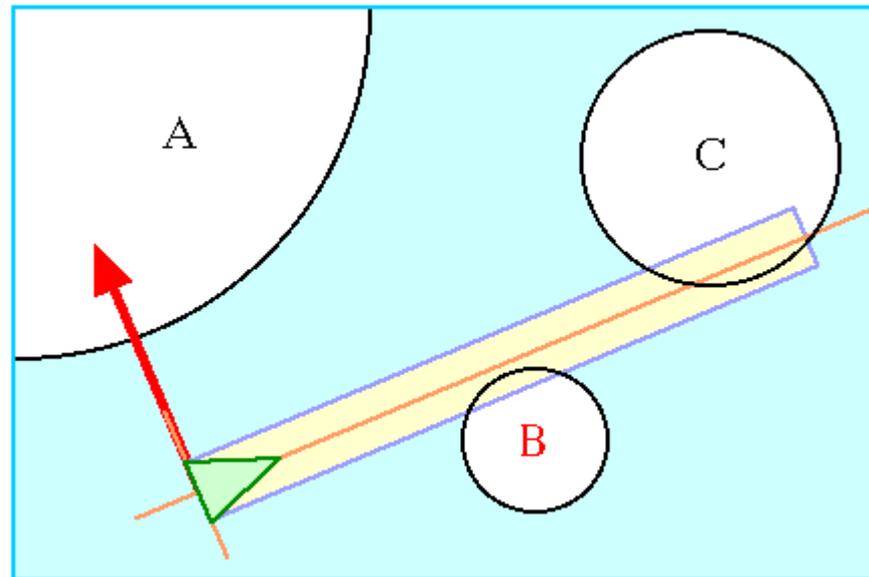
- $|\text{produit vectoriel}| > \text{taille obstacle}$ → Collision
 - $|\text{produit scalaire}| < \text{distance de vision}$ → Proche



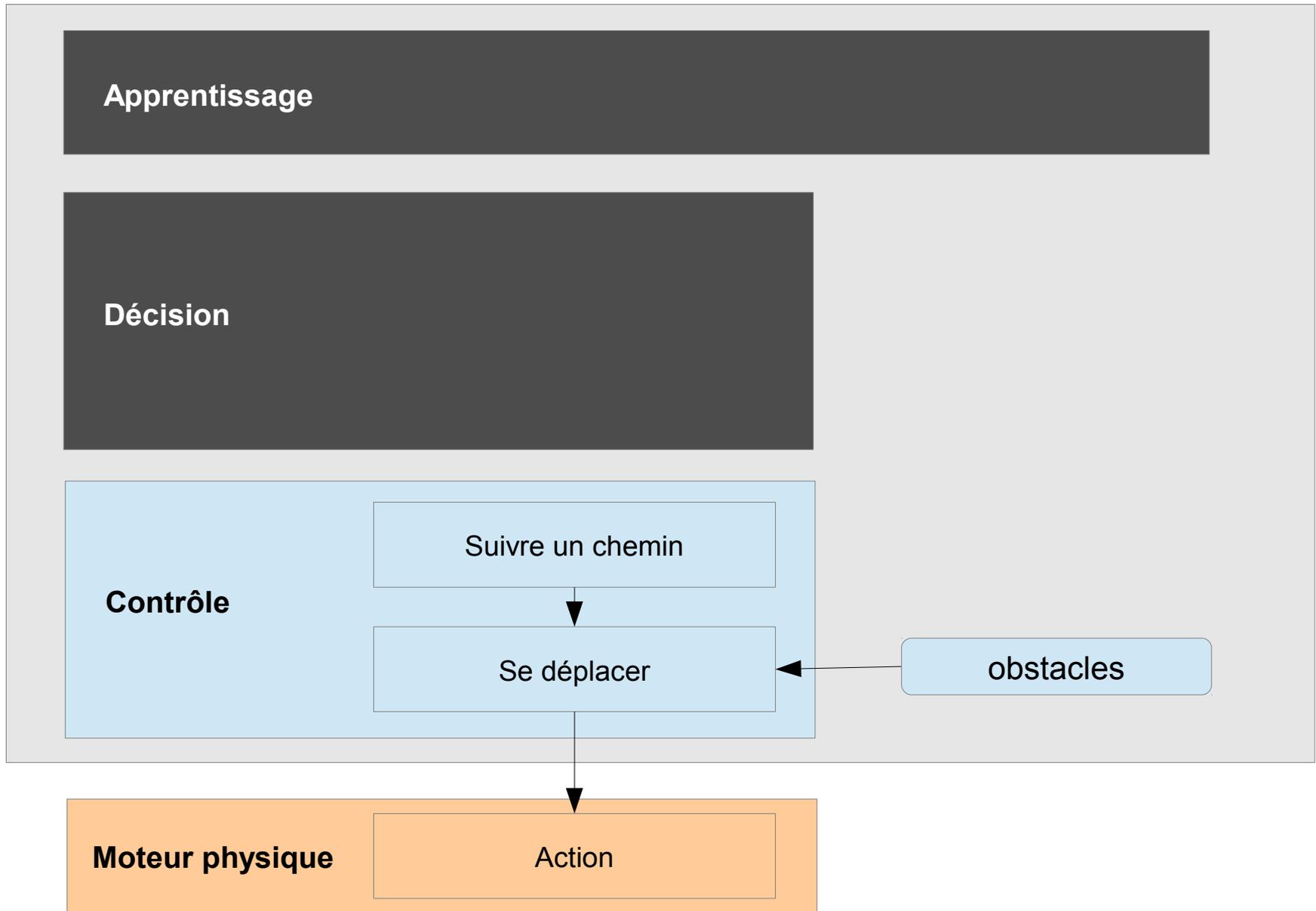
- Deux éléments
 - Détecter un obstacle
 - Éviter un obstacle



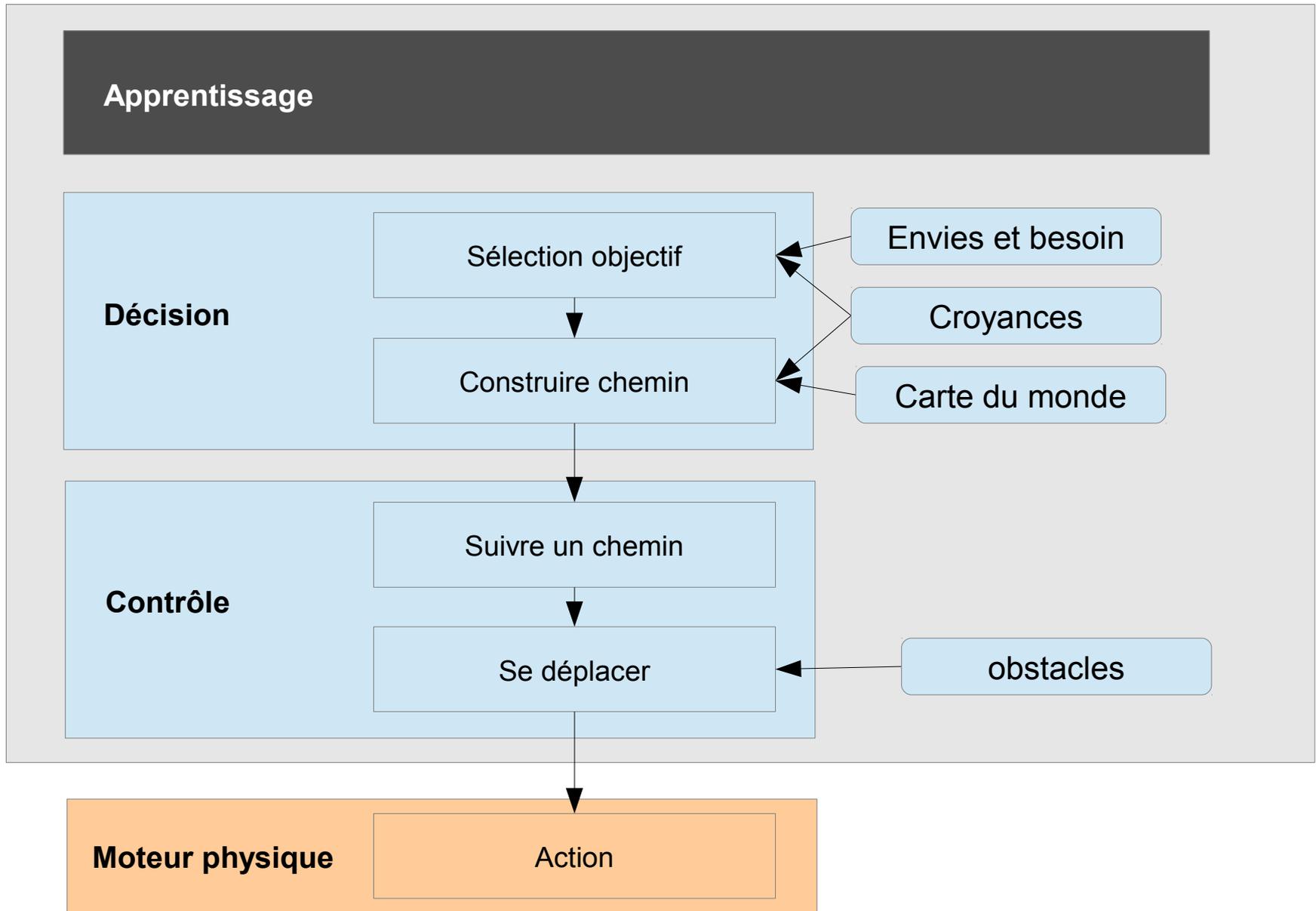
- Démo5 – évitement obstacle



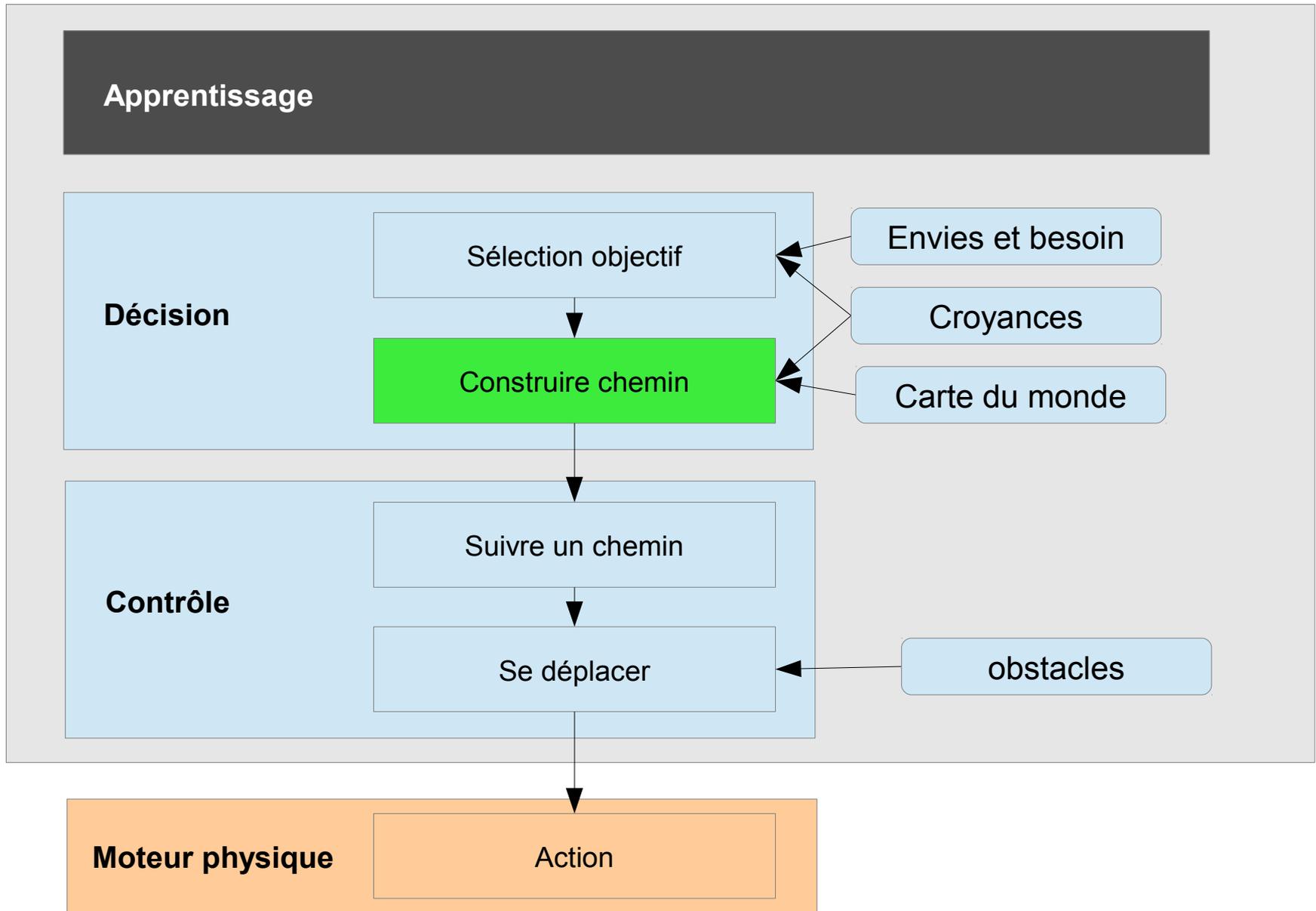
3 – couche Décision



3 – couche Décision



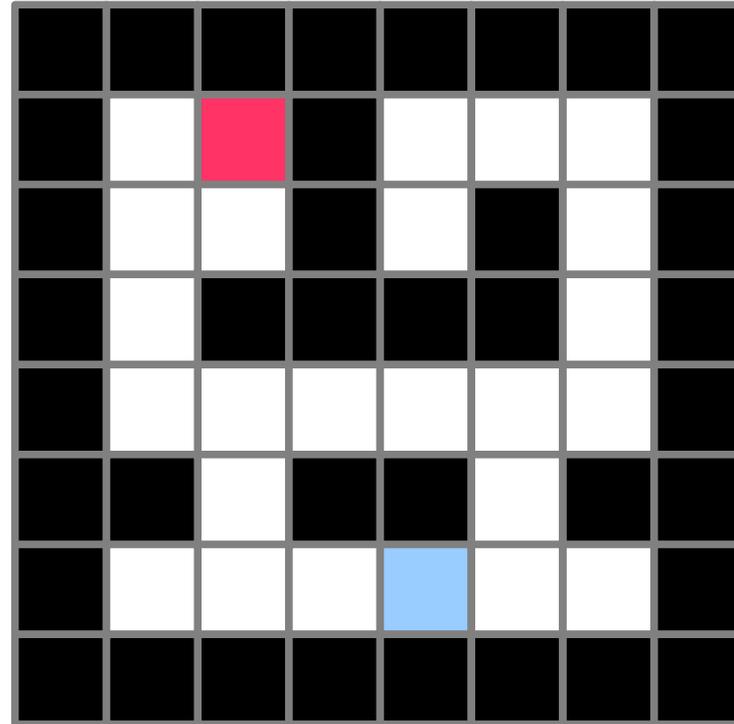
3 – couche Décision



- **Planification**
 - Trouver une suite de lieu
- **Problème classique en IA**
 - Wave algorithm
 - Djikstra
 - A*

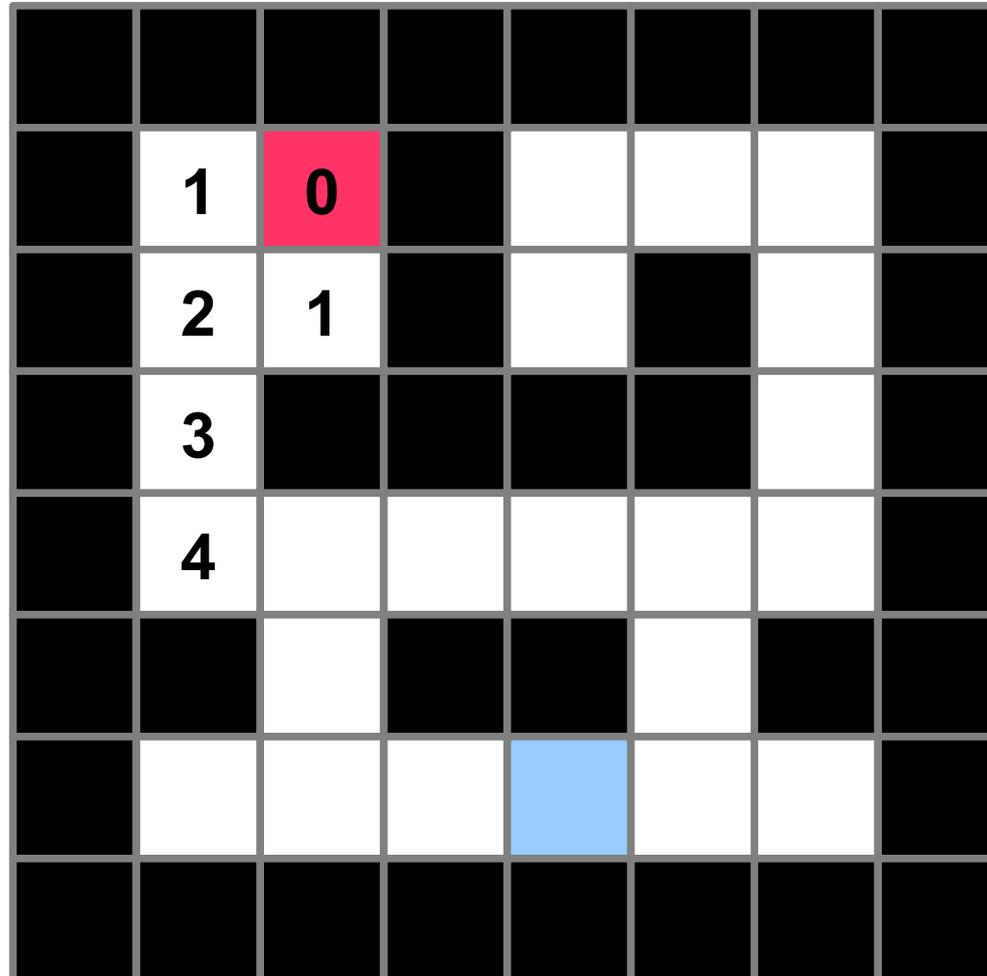
3.1 – construire un chemin

- **Algorithme de la vague**
 - Simple propagation



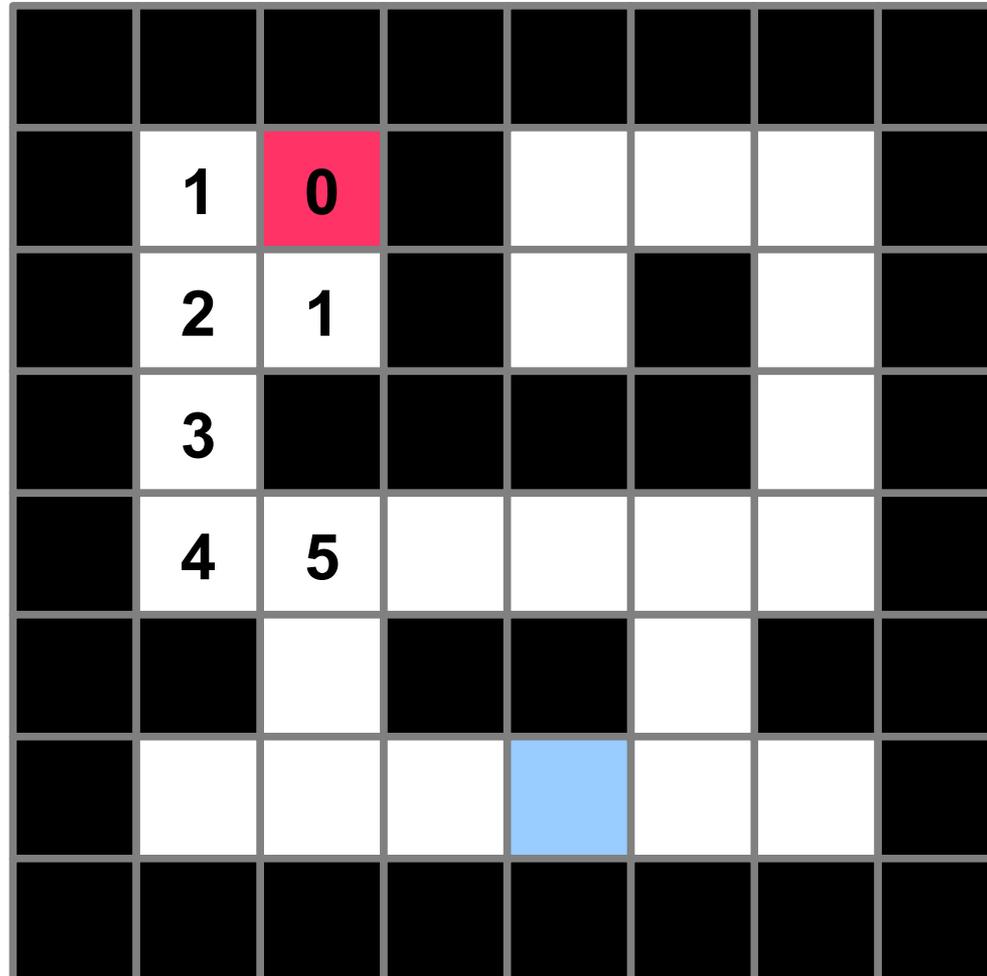
3.1 – construire un chemin

- Algorithme de la vague



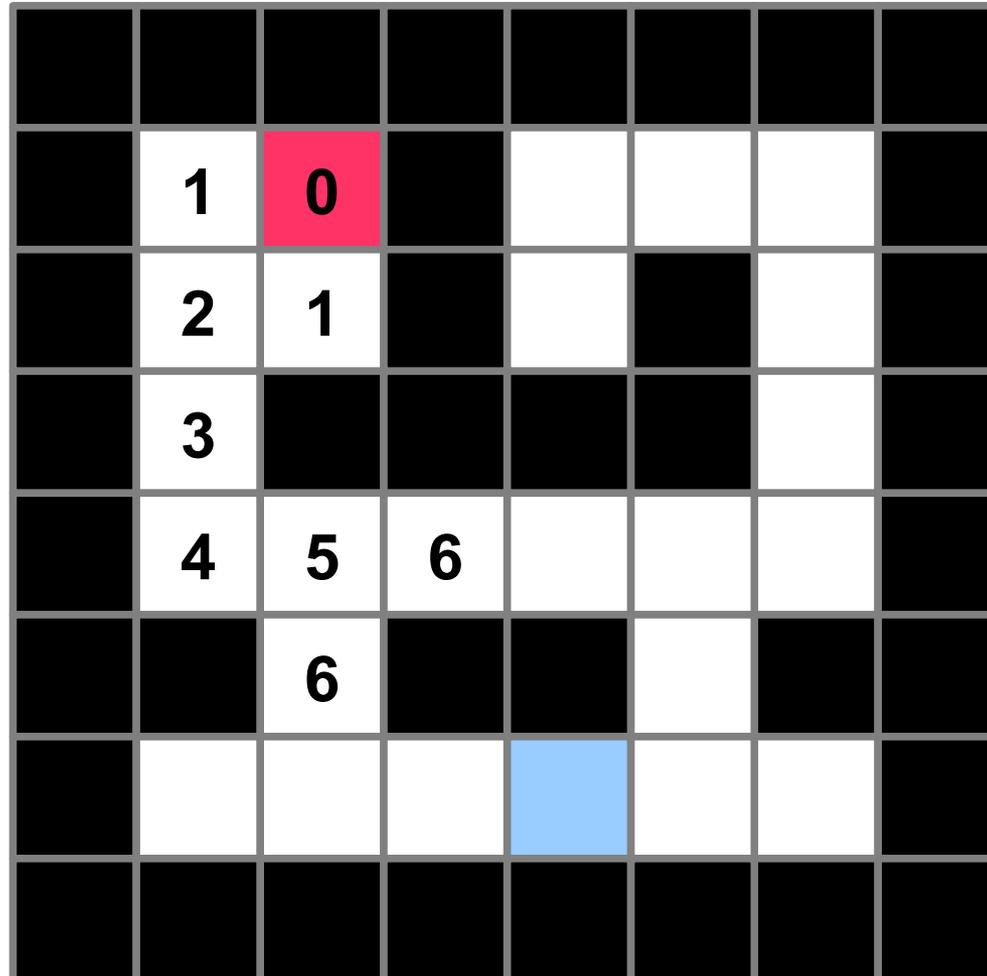
3.1 – construire un chemin

- Algorithme de la vague



3.1 – construire un chemin

- Algorithme de la vague



3.1 – construire un chemin

- Algorithme de la vague

	1	0					
	2	1				11	
	3					10	
	4	5	6	7	8	9	
		6			9		
	8	7	8	9	10	11	

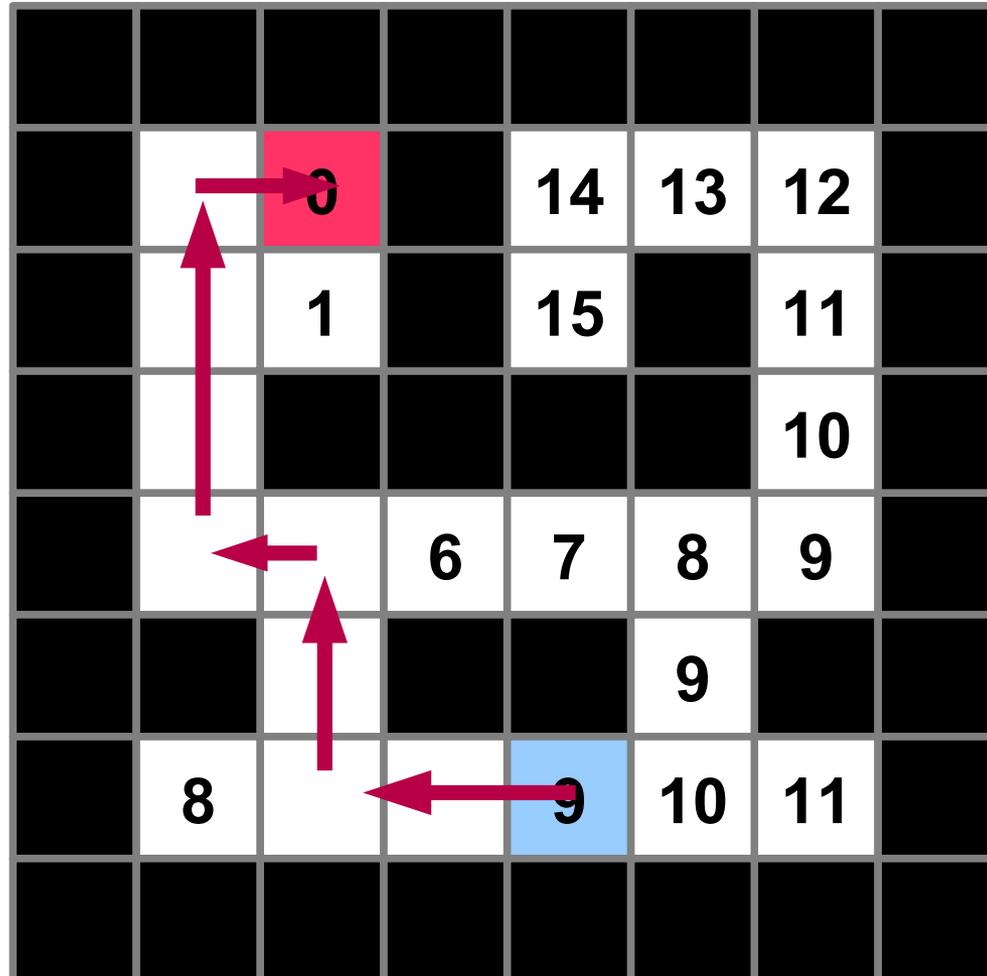
3.1 – construire un chemin

- Algorithme de la vague

	1	0		14	13	12	
	2	1		15		11	
	3					10	
	4	5	6	7	8	9	
		6			9		
	8	7	8	9	10	11	

3.1 – construire un chemin

- Algorithme de la vague



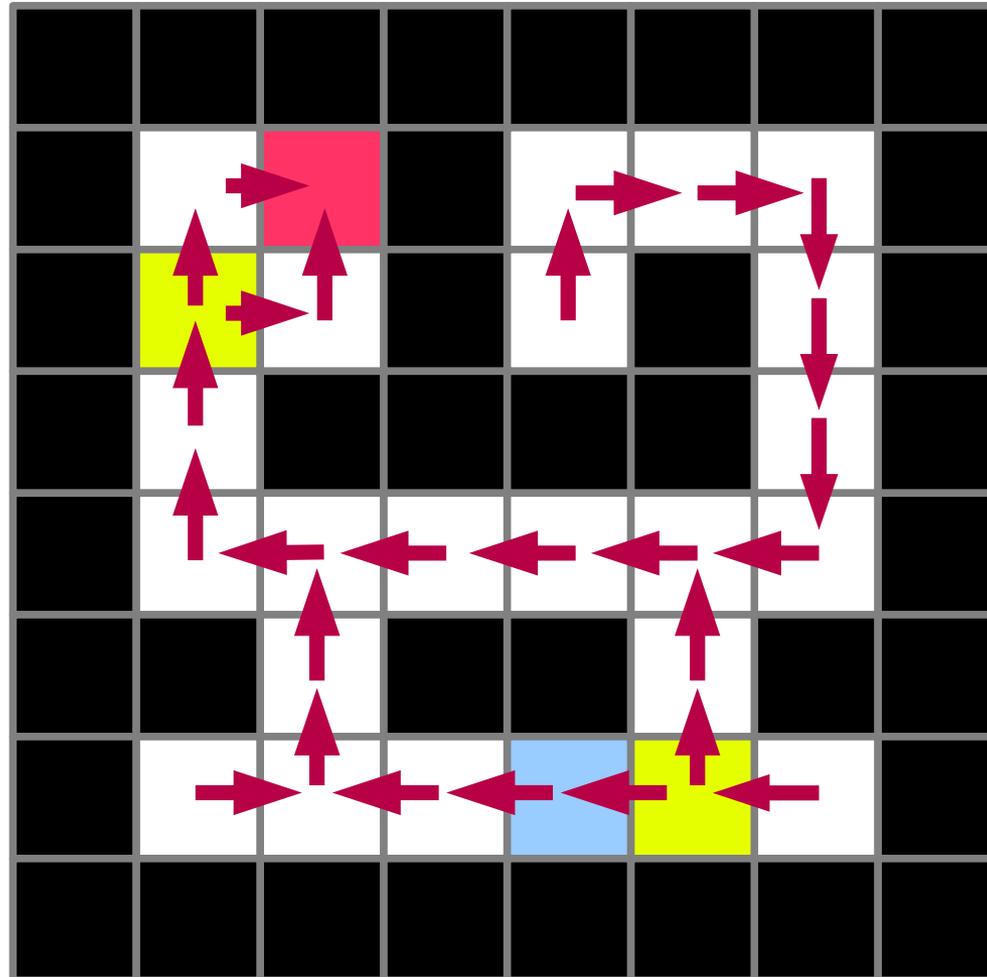
3.1 – construire un chemin

- Algorithme de la vague

	1	0		14	13	12	
	2	1		15		11	
	3					10	
	4	5	6	7	8	9	
		6			9		
	8	7	8	9	10	11	

3.1 – construire un chemin

- Algorithme de la vague



3.1 – construire un chemin

- **Algorithme de la vague**
 - Ajout difficulté ?
 - Danger

	1	0		14	13	12	
	2	1		15		11	
	3					10	
	4	5	6	7	8	9	
		8			9		
	12	11	12	11	10	11	

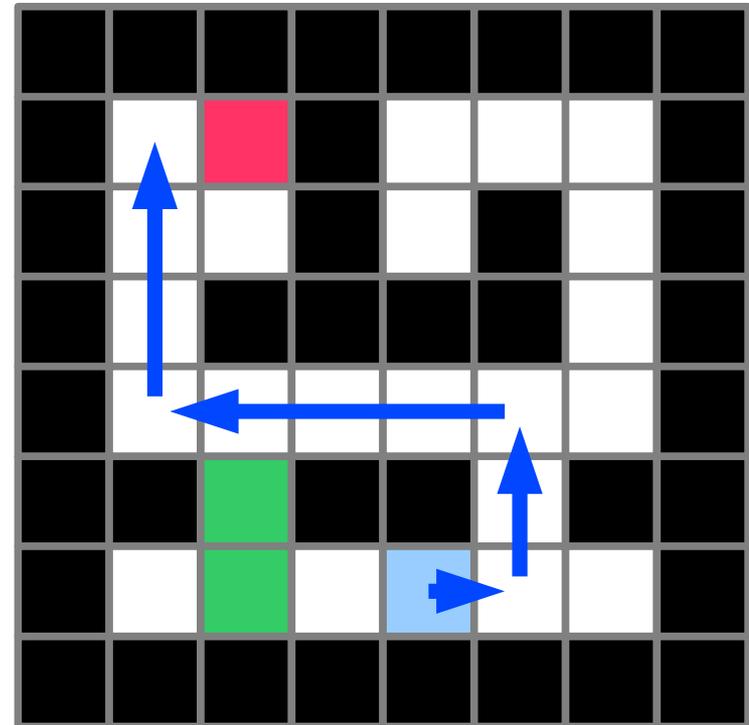
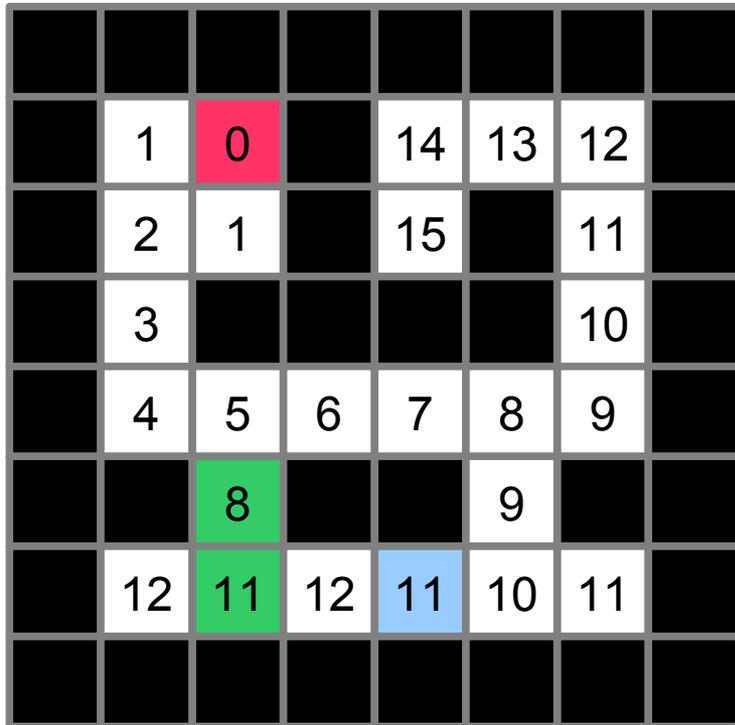
3.1 – construire un chemin

- **Algorithme de la vague**
 - Ajout difficulté ?
 - Danger

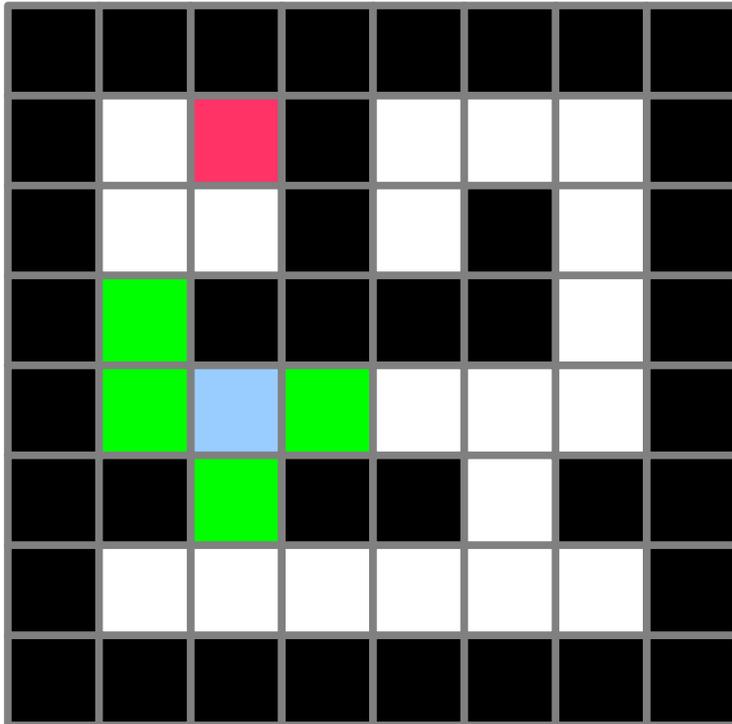
	1	0		14	13	12	
	2	1		15		11	
	3					10	
	4	5	6	7	8	9	
		8			9		
	12	11	12	11	10	11	

3.1 – construire un chemin

- **Algorithme de la vague**
 - Ajout difficulté ?
 - Danger

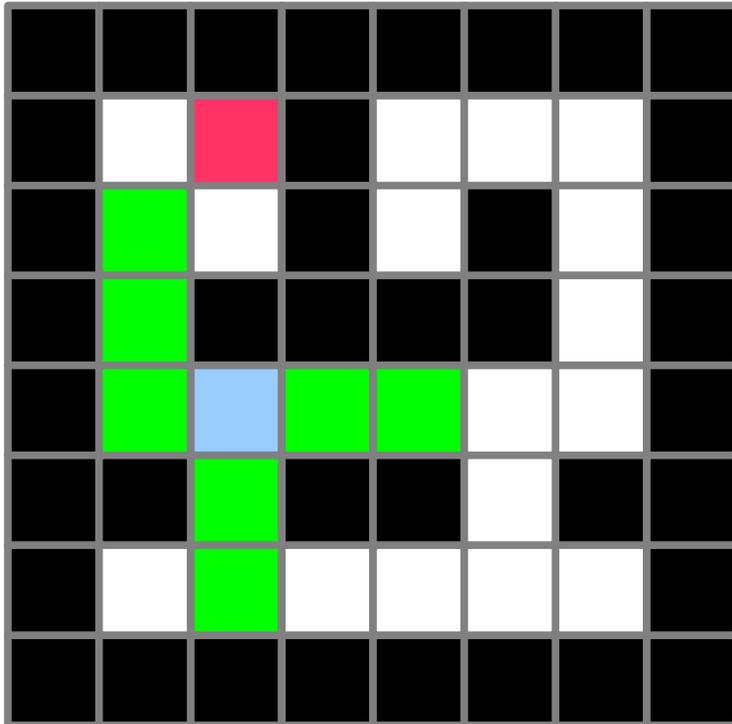


- **Autres algorithmes**
 - Dijkstra (Développer par rapport au départ)



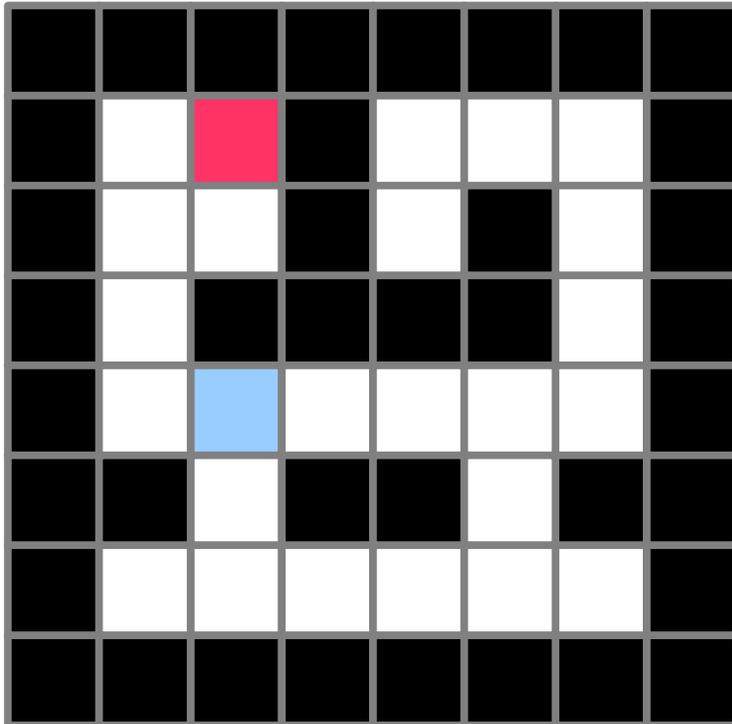
Dijkstra

- **Autres algorithmes**
 - Dijkstra (Développer par rapport au départ)



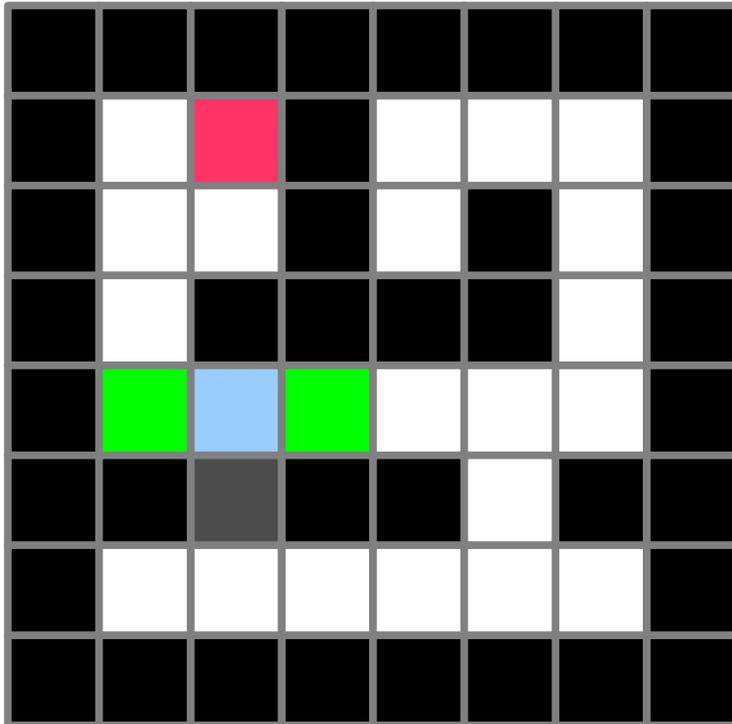
Dijkstra

- Autres algorithmes
 - Dijkstra (Développer par rapport au départ)
 - A^* (Développer le plus prometteur)



A^*

- Autres algorithmes
 - Dijkstra (Développer par rapport au départ)
 - **A*** (Développer le plus prometteur)

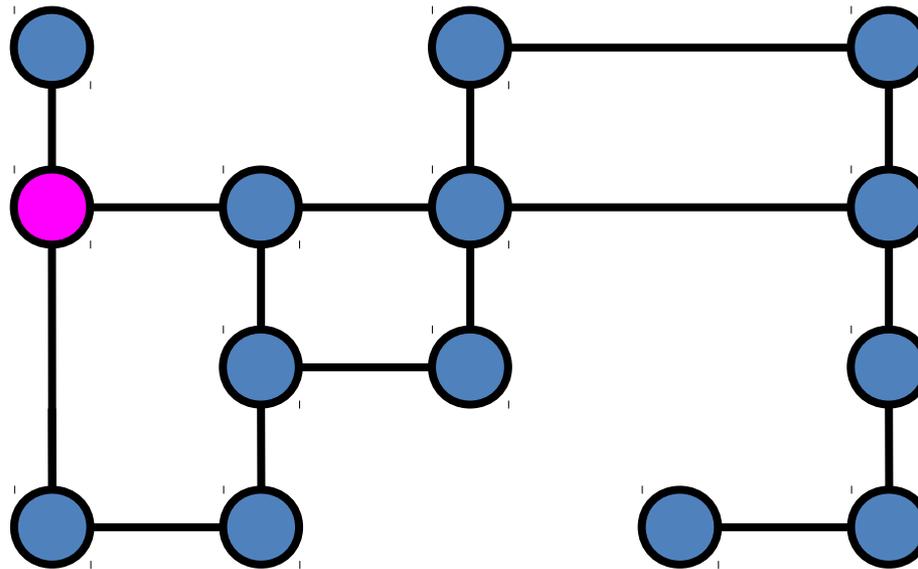


A*

- **Démo6**
 - Replanification en cas de modification

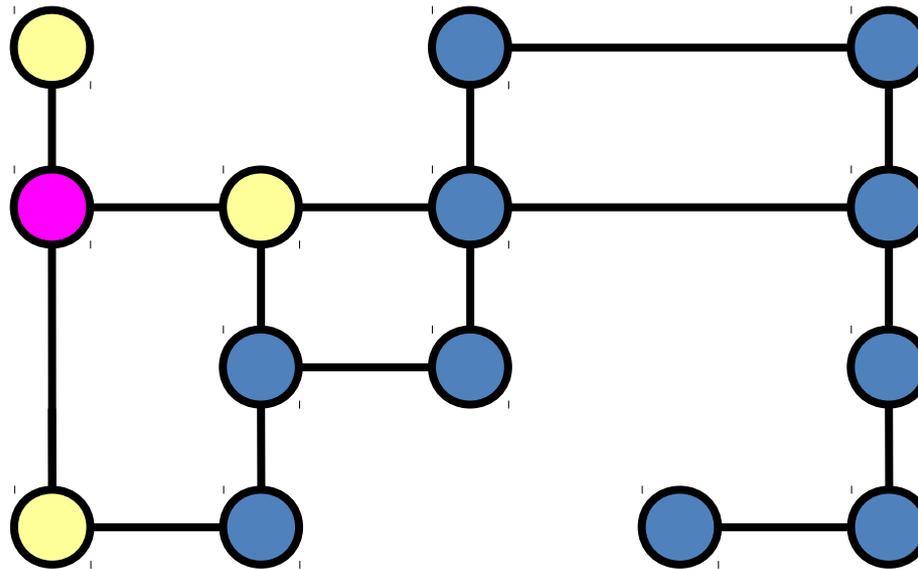
3.1 – construire un chemin

- **Algorithme de recherche de plus court chemin**
 - départ



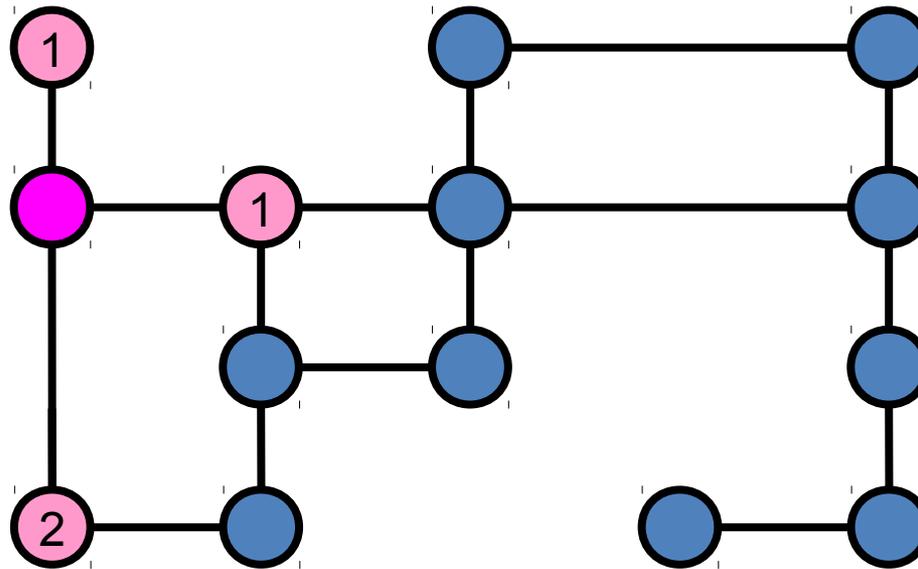
3.1 – construire un chemin

- **Algorithme de recherche de plus court chemin**
– développement



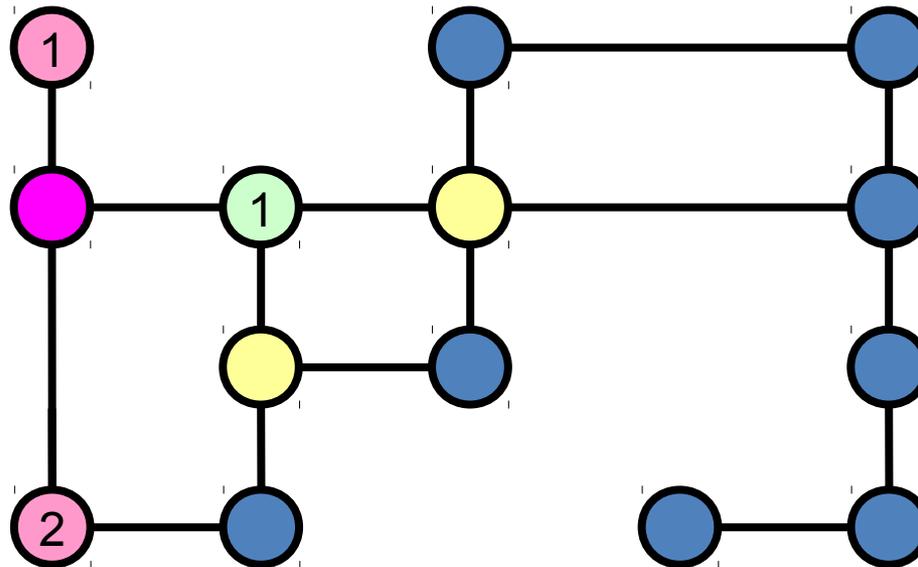
3.1 – construire un chemin

- **Algorithme de recherche de plus court chemin**
 - Mise à jour des coûts



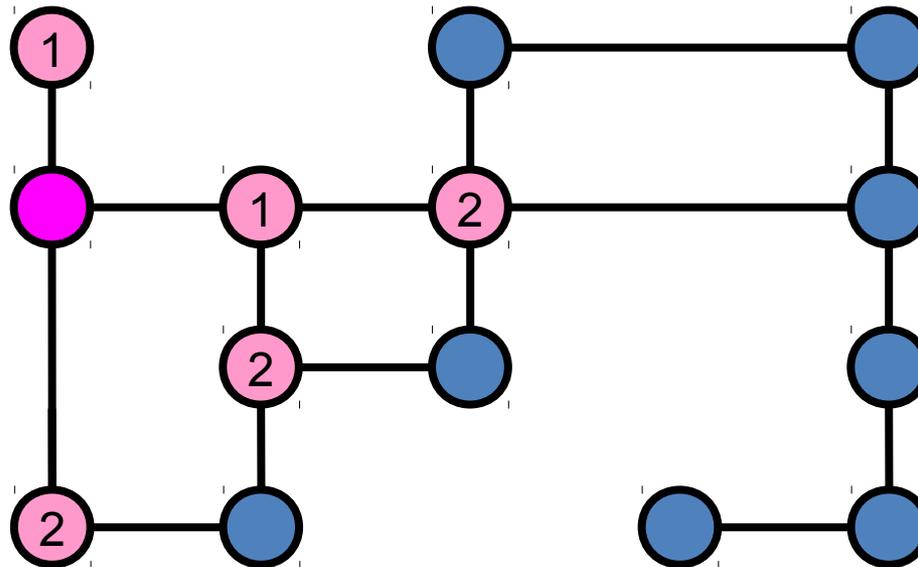
3.1 – construire un chemin

- **Algorithme de recherche de plus court chemin**
 - Développement suivant



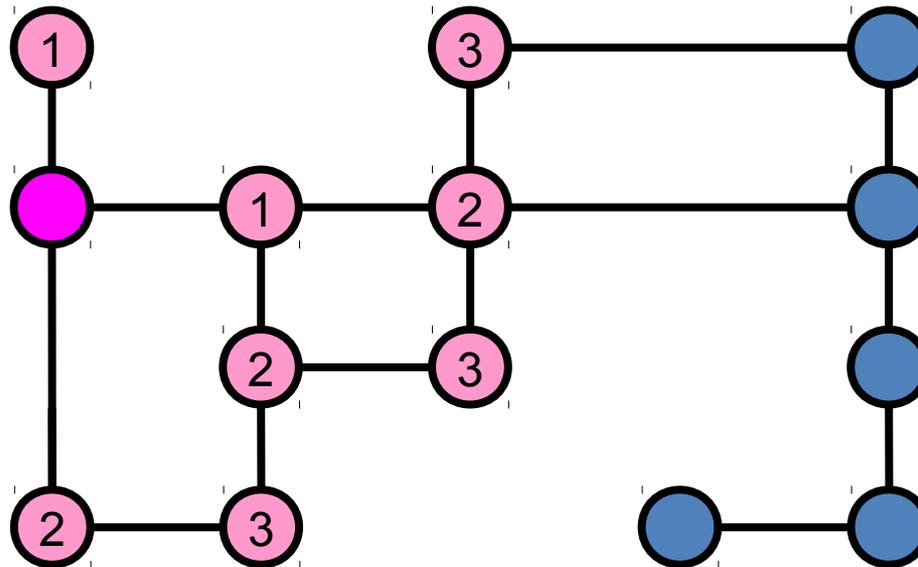
3.1 – construire un chemin

- **Algorithme de recherche de plus court chemin**
 - Mise à jour des couts



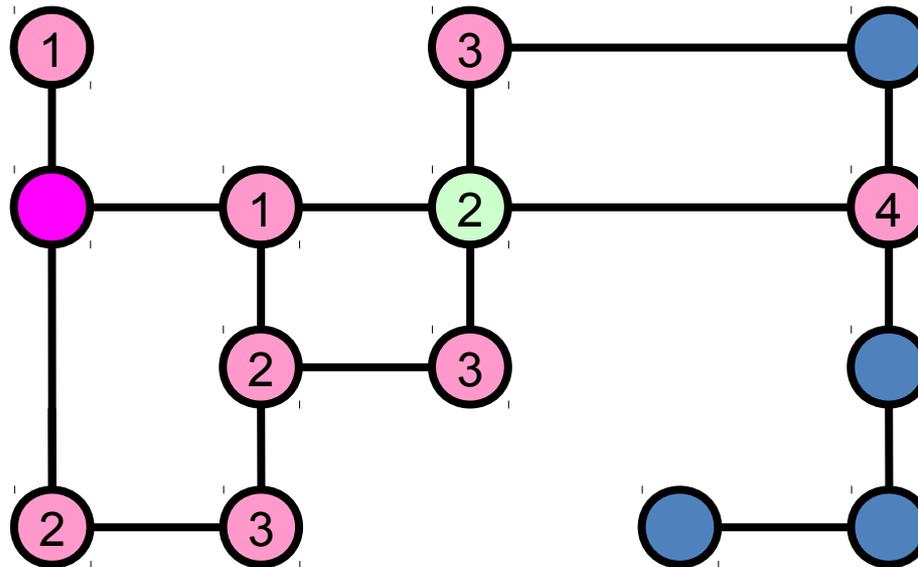
3.1 – construire un chemin

- **Algorithme de recherche de plus court chemin**
 - Mise à jour des couts



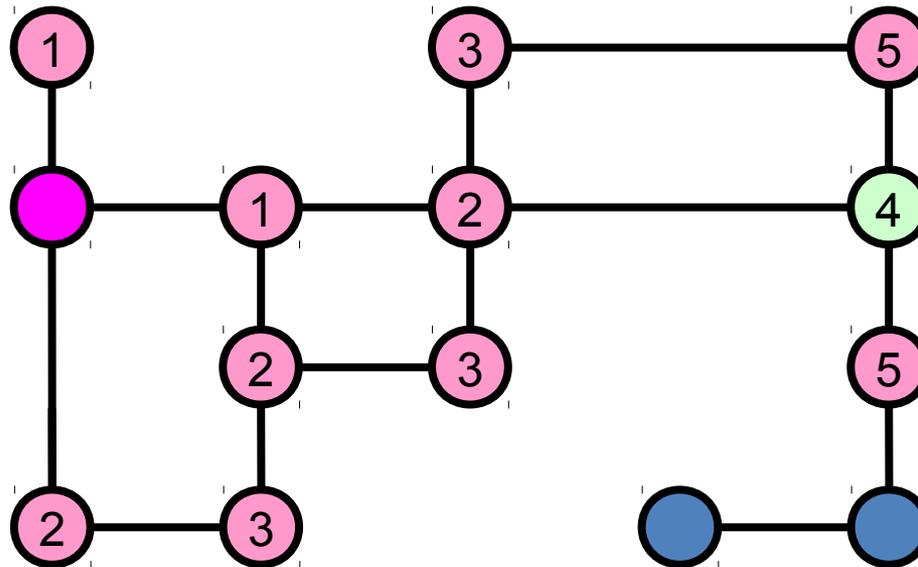
3.1 – construire un chemin

- **Algorithme de recherche de plus court chemin**
 - Mise à jour des couts



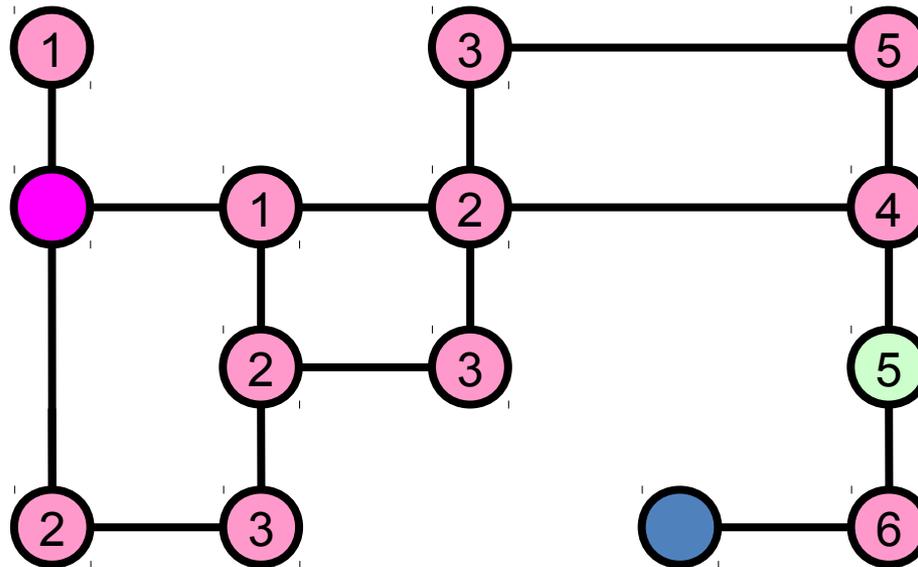
3.1 – construire un chemin

- **Algorithme de recherche de plus court chemin**
 - Mise à jour des couts



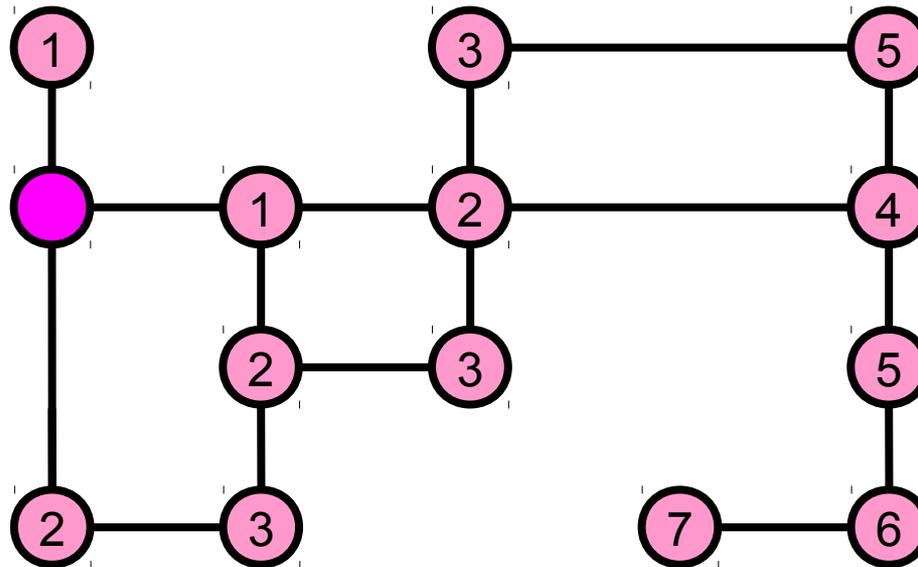
3.1 – construire un chemin

- **Algorithme de recherche de plus court chemin**
 - Mise à jour des couts



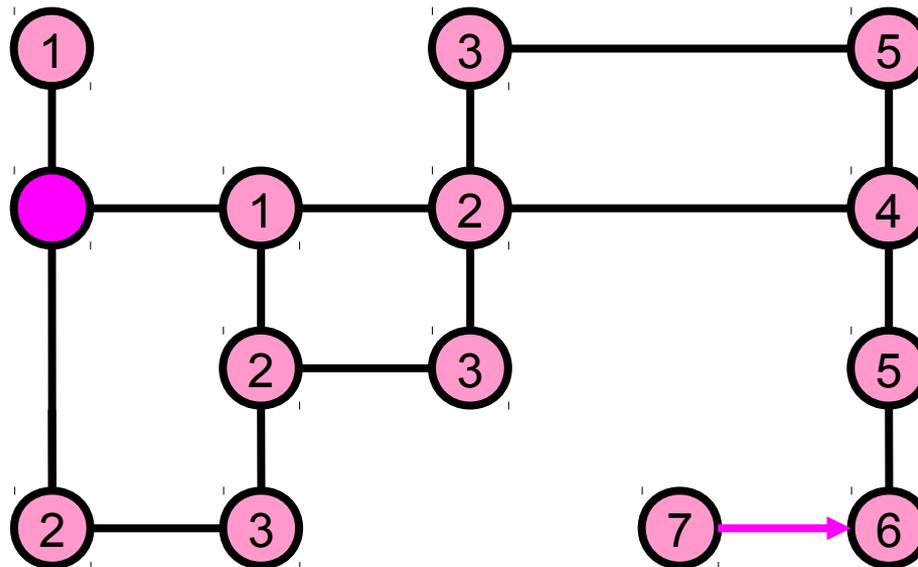
3.1 – construire un chemin

- **Algorithme de recherche de plus court chemin**
 - Mise à jour des couts



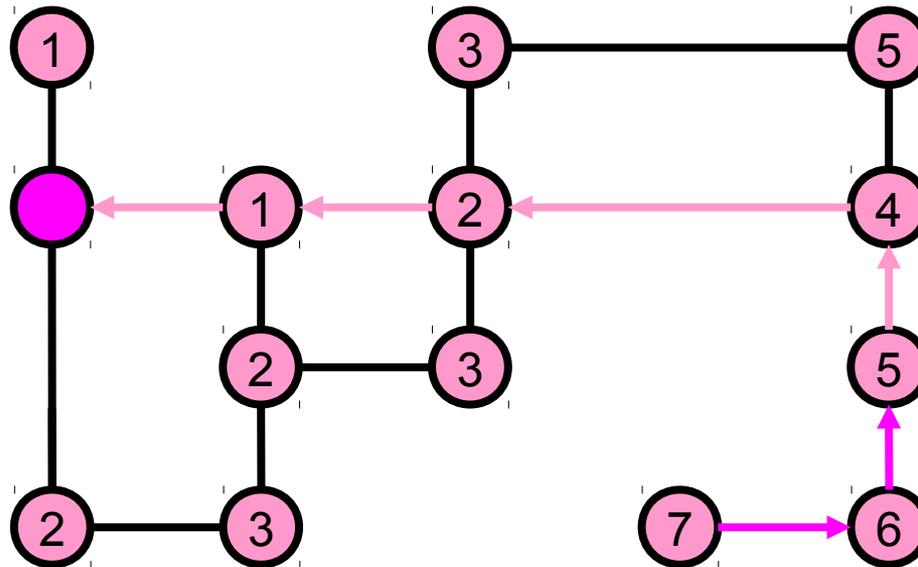
3.1 – construire un chemin

- **Algorithme de recherche de plus court chemin**
 - Recherche du chemin

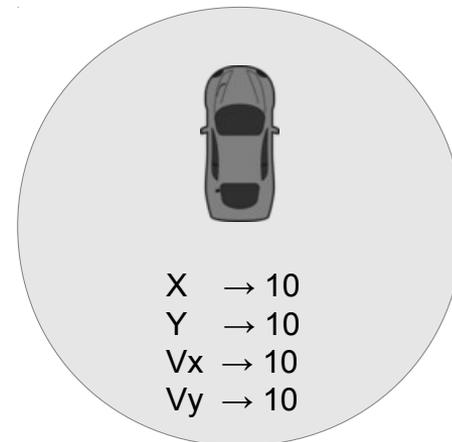


3.1 – construire un chemin

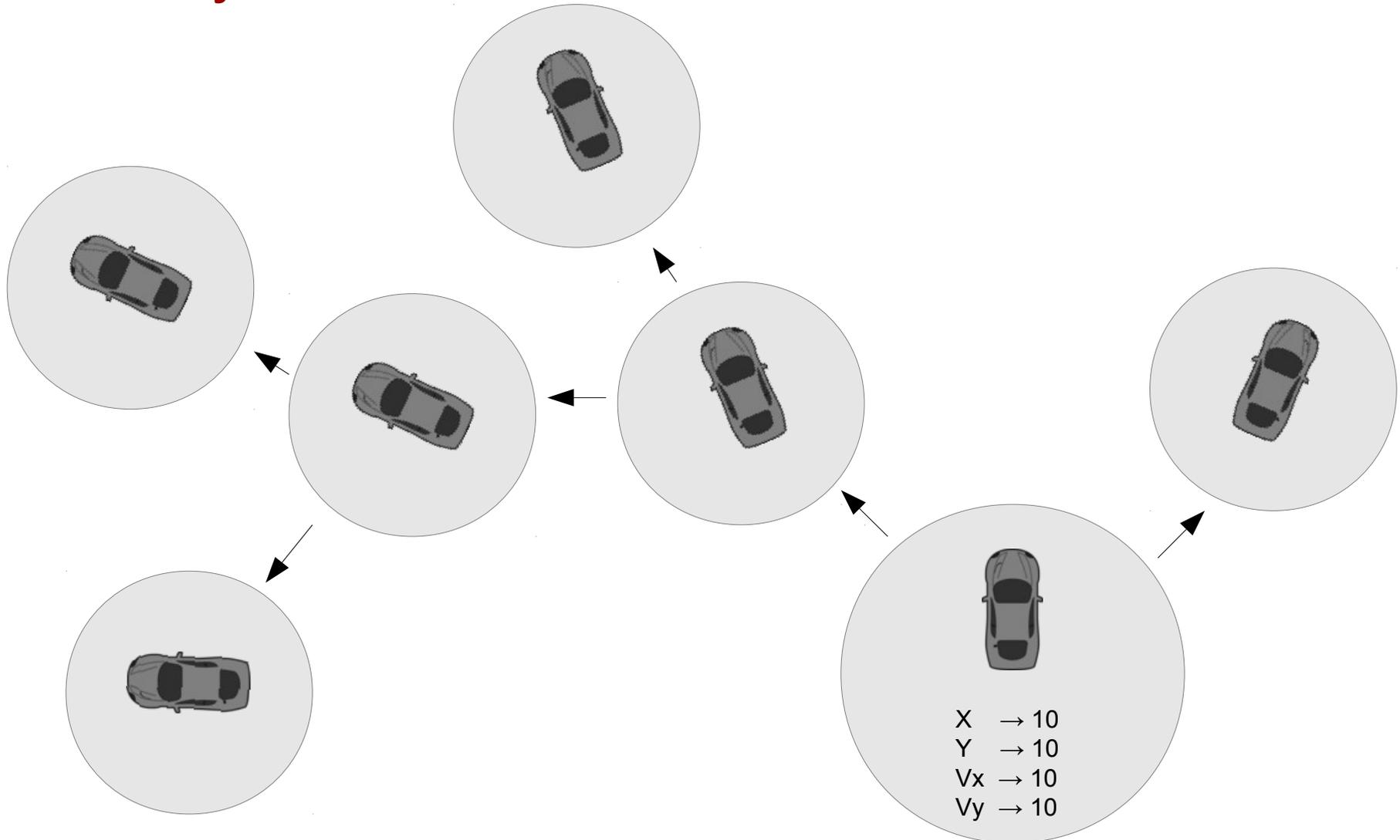
- **Algorithme de recherche de plus court chemin**
 - Calcul du chemin en retour arriere



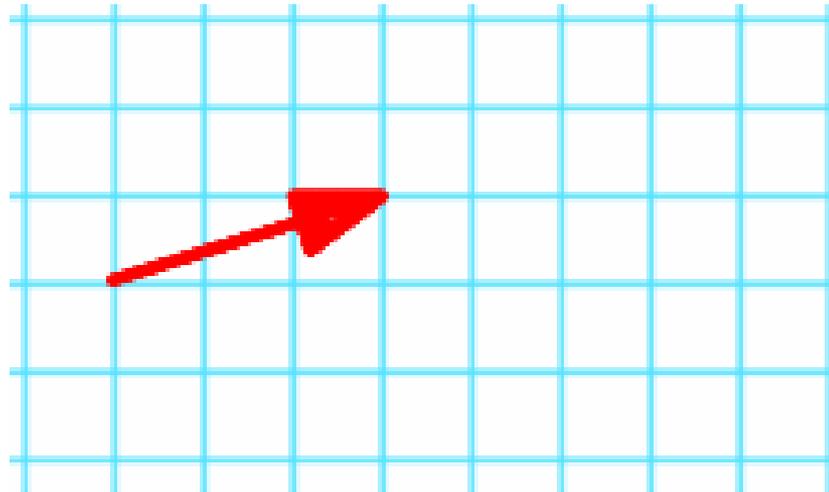
- **Trajectoire non holonome**
 - Contrôle de voiture (x, y, v_x, v_y)



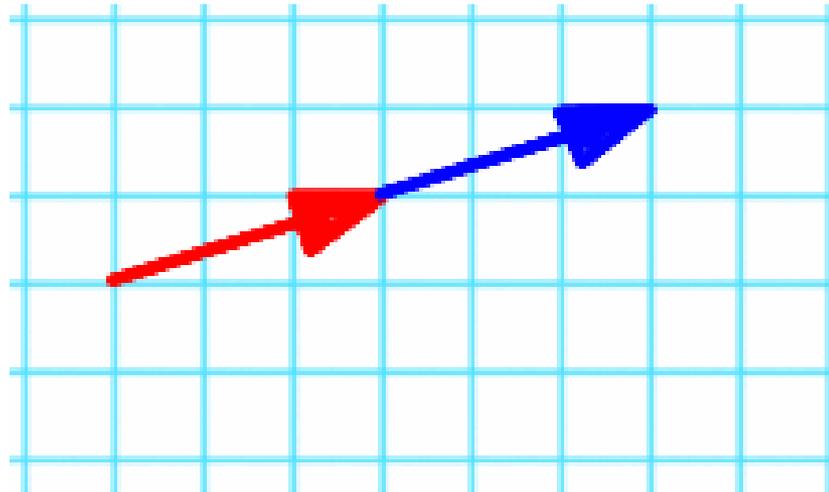
- Trajectoire non holonome



- **Trajectoire non-holonome**
 - Jeu sur papier RaceTrack

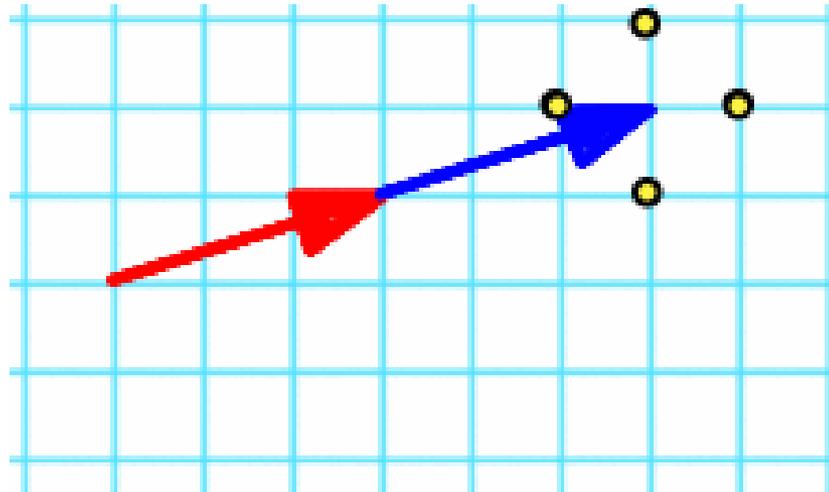


- **Trajectoire non-holonome**
 - Jeu sur papier RaceTrack



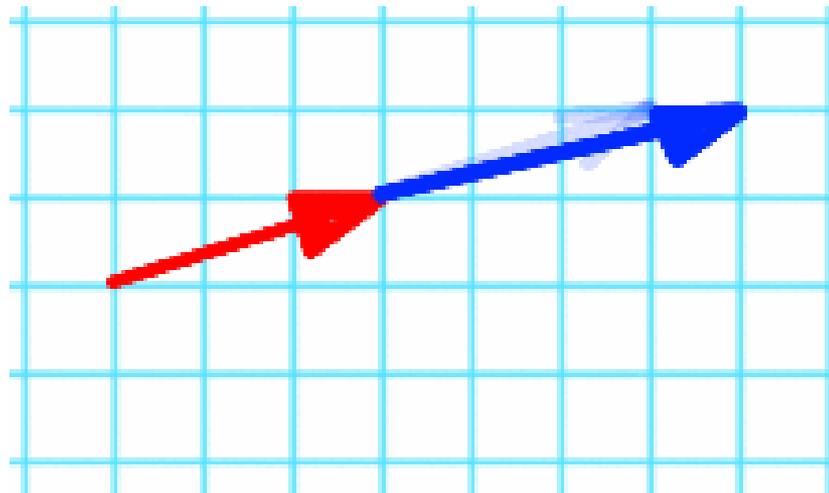
http://www.sjbaker.org/paper_and_pencil_games/graph_racers/

- **Trajectoire non-holonome**
 - Jeu sur papier RaceTrack

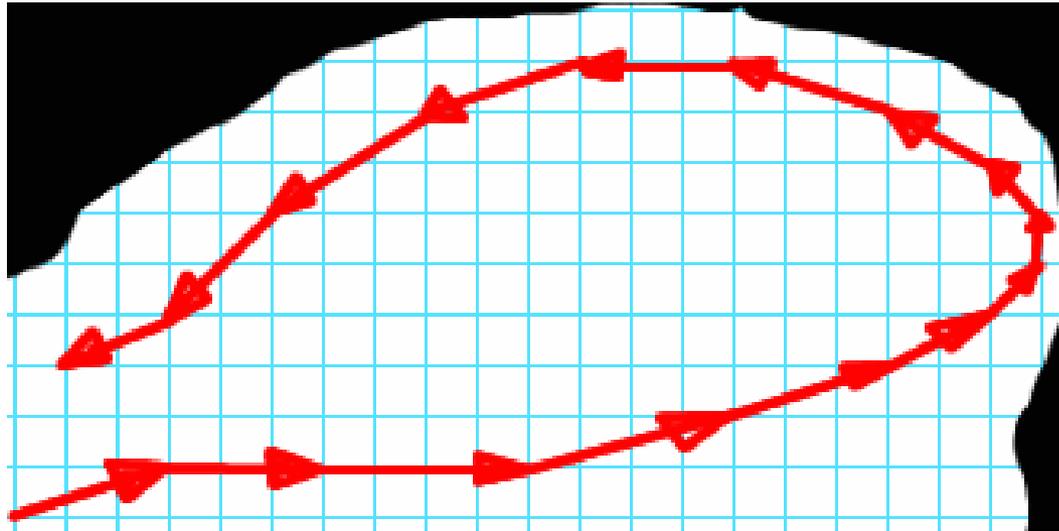


http://www.sjbaker.org/paper_and_pencil_games/graph_racers/

- **Trajectoire non-holonome**
 - Jeu sur papier RaceTrack



- Trajectoire non-holonome
 - Jeu sur papier RaceTrack

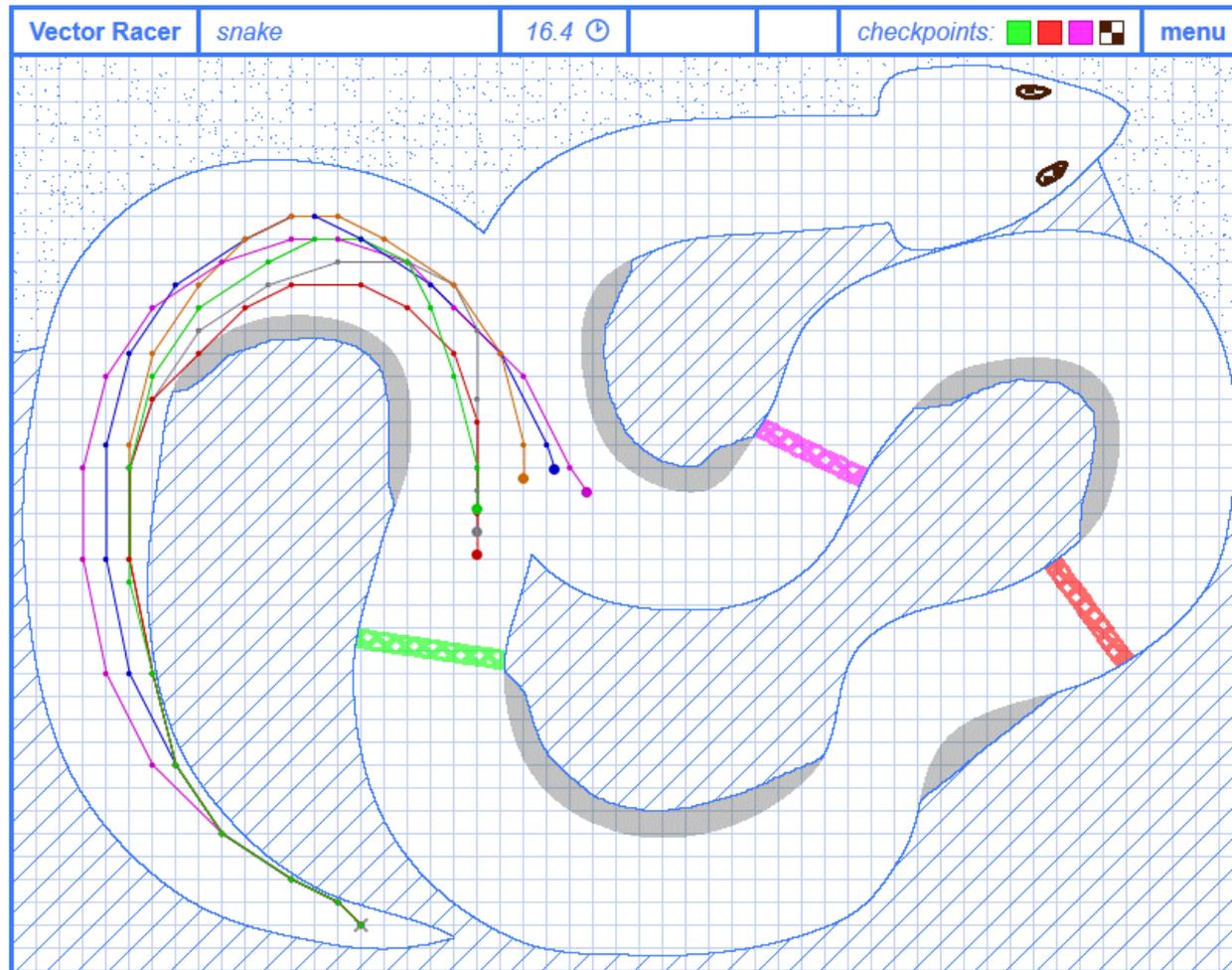


3.1 – Espace d'état complexe

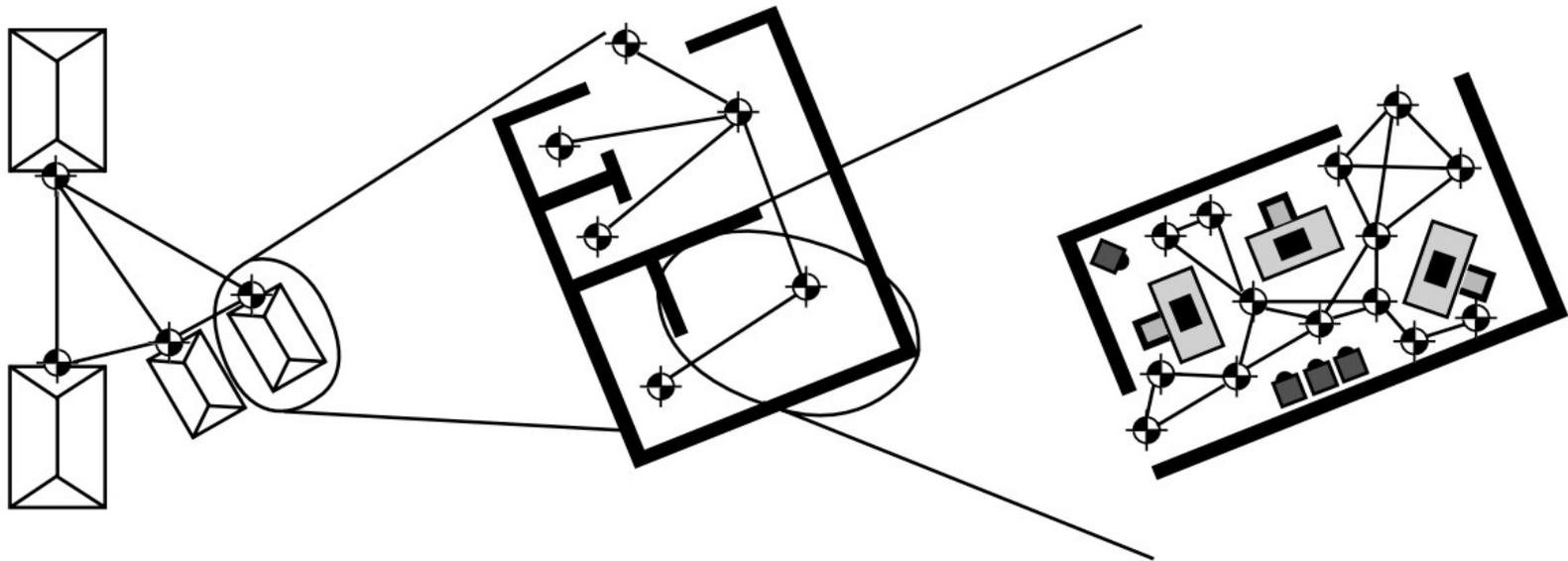
161
244

- **Trajectoire non-holonome**

<http://www.harrrmade.com/vectorracer/>



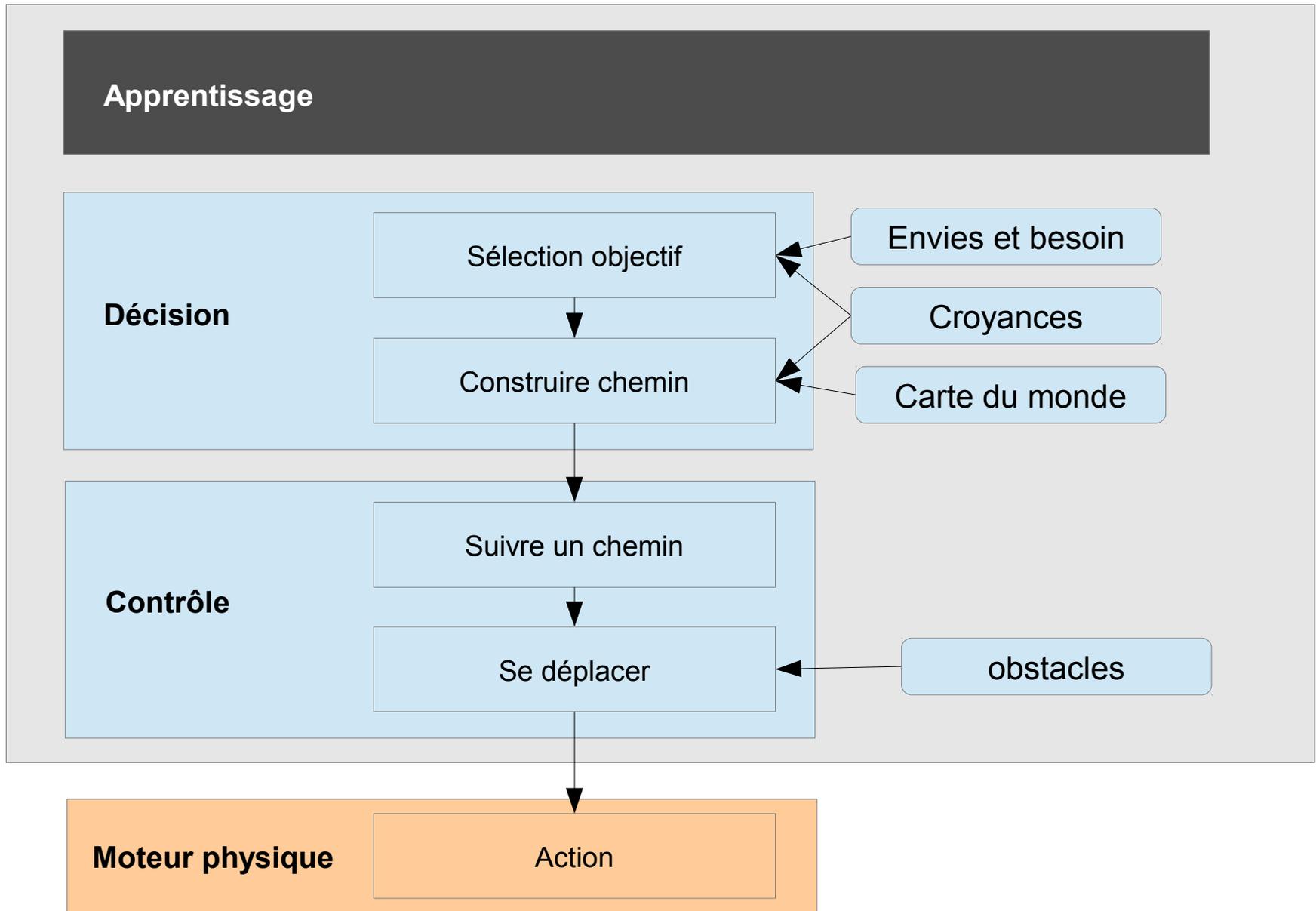
- **Planification hiérarchique**
 - Raisonner à différents niveaux
 - En fonction du lieu



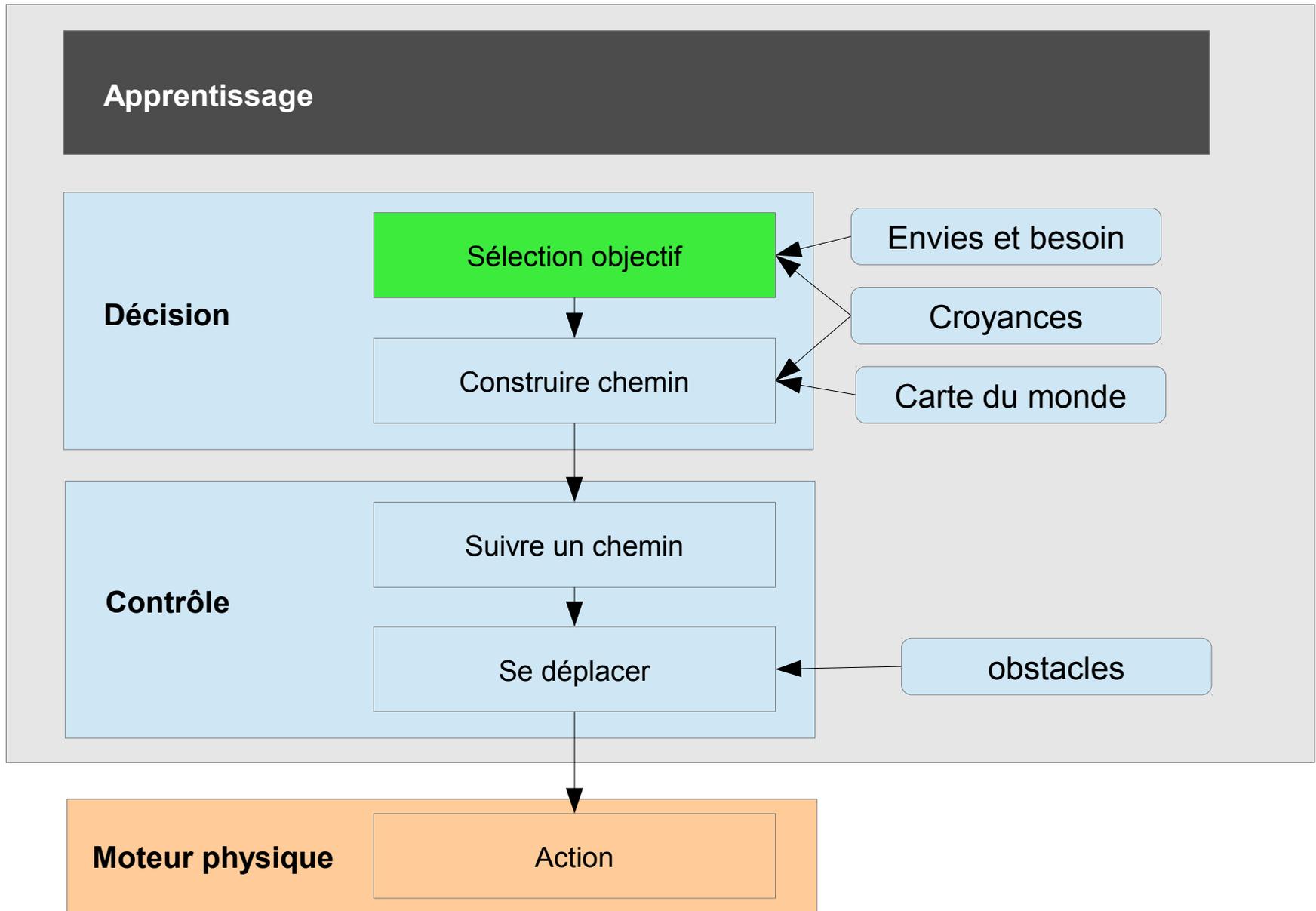
« *Artificial intelligence for games* » (Millington / Funge)

- **Démonstration7**
 - Coupler
 - Planification
 - Navigation

3 – couche Décision



3 – couche Décision



- **Personnage sait**
 - Construire un chemin vers but
 - Naviguer, éviter obstacles
- **Comment choisir le but ?**
- **Plusieurs approches**
 - "Behavior Tree"
 - Planification haut niveau
 - "Practical reasoning"
 - D'autres ...

- Scripts
- Automates
- Behavior Tree
- Practical reasoning – BDI
- Planification haut niveau

- **Scripts**

- Entité dans un Jdr
 - Fixée
 - Modifiable (langage)

Si (demande renseignement)
alors devient hostile
sinon reste amical

Si (hostile)
alors attaque_unité

- **Scripts**

- Entité dans un Jdr
 - Fixée
 - Modifiable (langage)

Si (demande renseignement)
alors devient hostile
sinon reste amical

Si (hostile)
alors attaque_unité

- **Never Winter Nights**

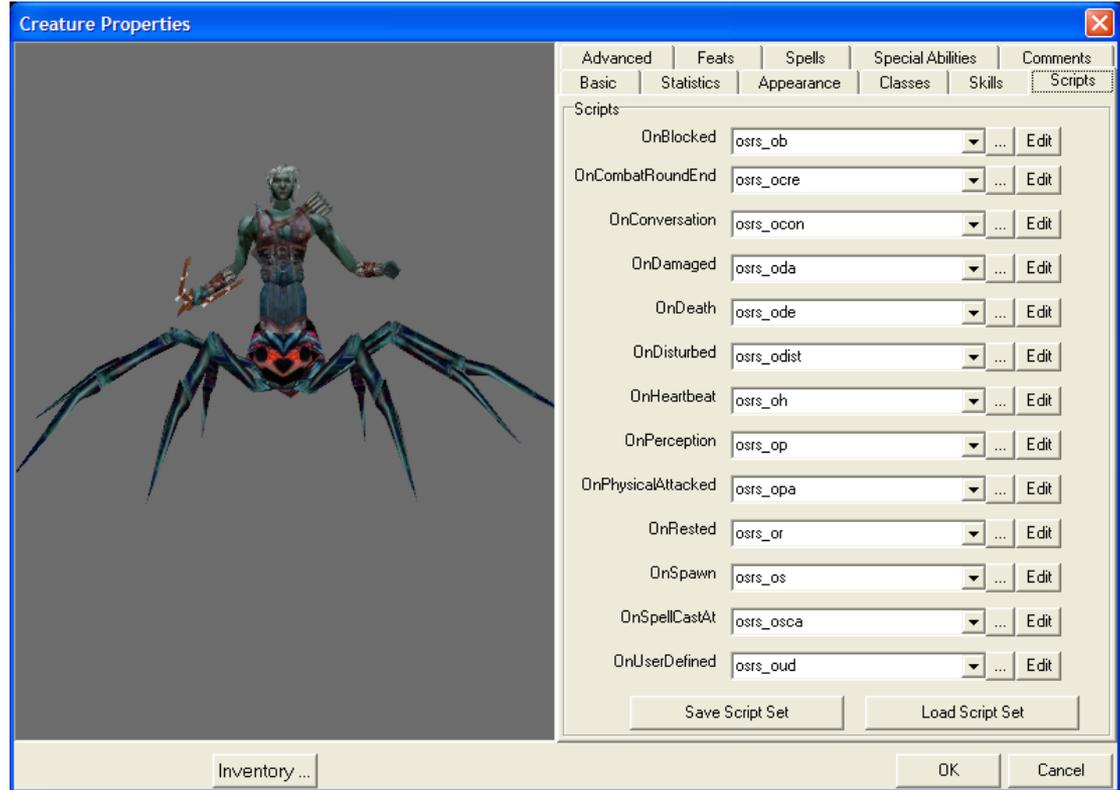
- Permettre d'être MJ
- Aura Toolbox
 - NWScript (Objets, NPCs)
 - Dialogues

(2002)



3.2 - Sélection d'objectifs

Gestion des Scripts



Gestionnaire dialogues



- Scripts
- Automates
- Behavior Tree
- Practical reasoning – BDI
- Planification haut niveau

- **Machine à état fini**
 - Troupe et état
 - Fixée
 - Évoluable

Switch(état)

- **patrouille :**

- Déplacer

- Si ennemi état ← attaque

- **Attaque**

- Diriger vers unité

- Attaquer unité

- Si mort état ← défense

- **Defense**

- Si attaquer état ← attaque

- Automates

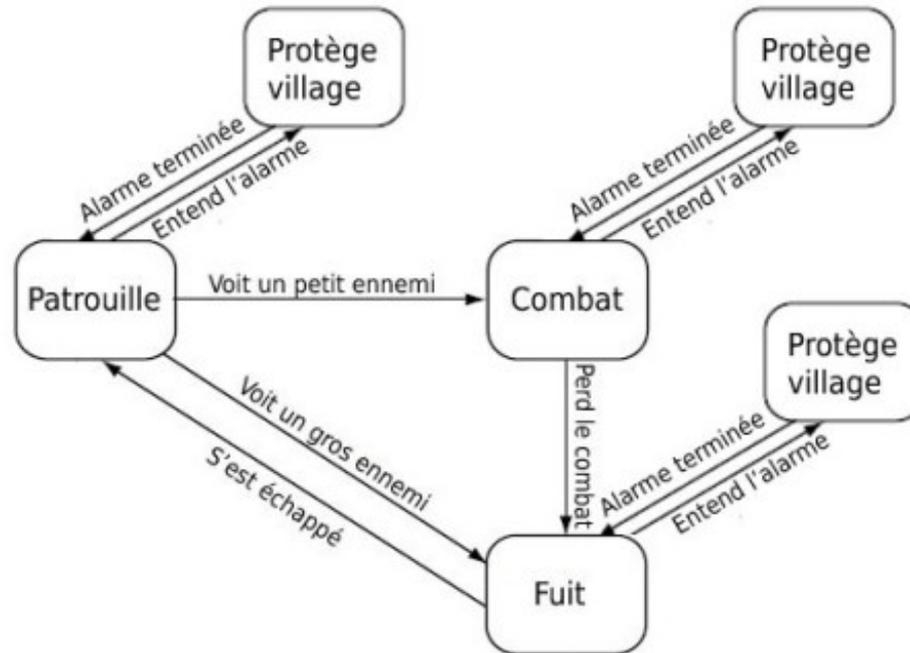


FIG. 3 – Un évènement d'alerte est ajouté au cycle d'activité du soldat. (Millington, 2006)

- Scripts
- Automates
- Behavior Tree
- Practical reasoning – BDI
- Planification haut niveau

- Coder une suite d'actions

- Behavior Trees

- <http://bjoernknafla.com/introduction-to-behavior-trees>

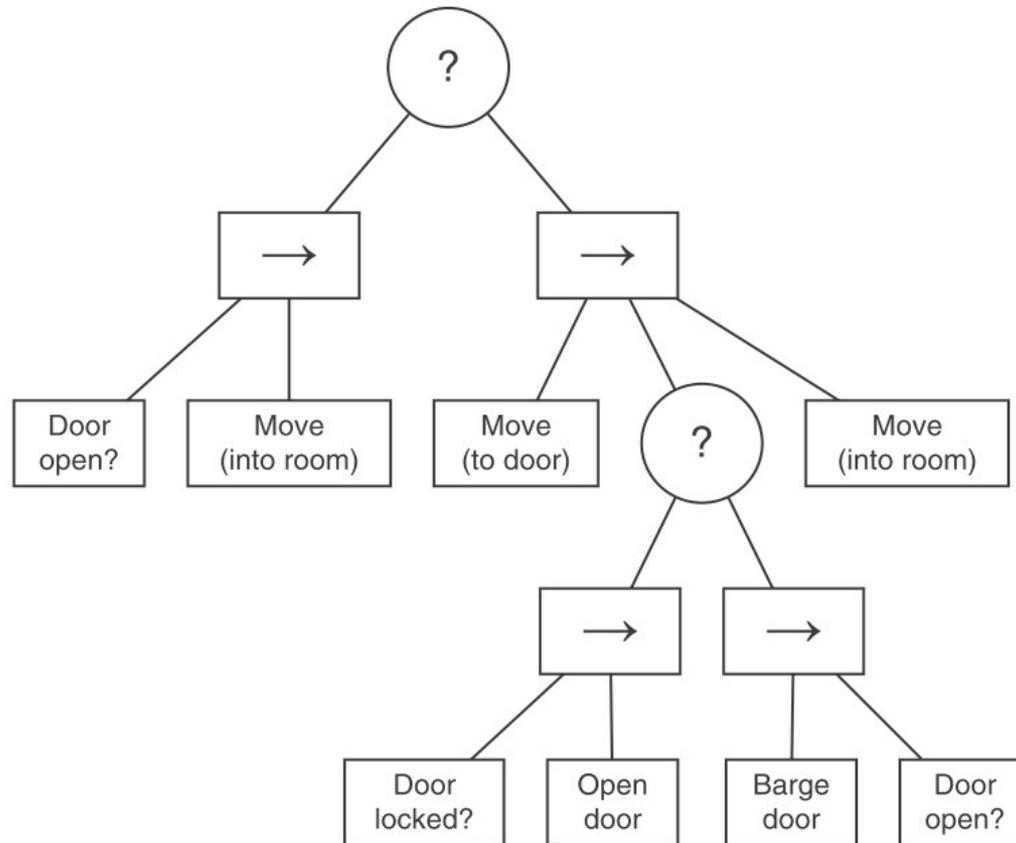
- Arbre de comportement

- Conditionnelles

- Séquences d'actions

3.2 - Sélection d'objectifs

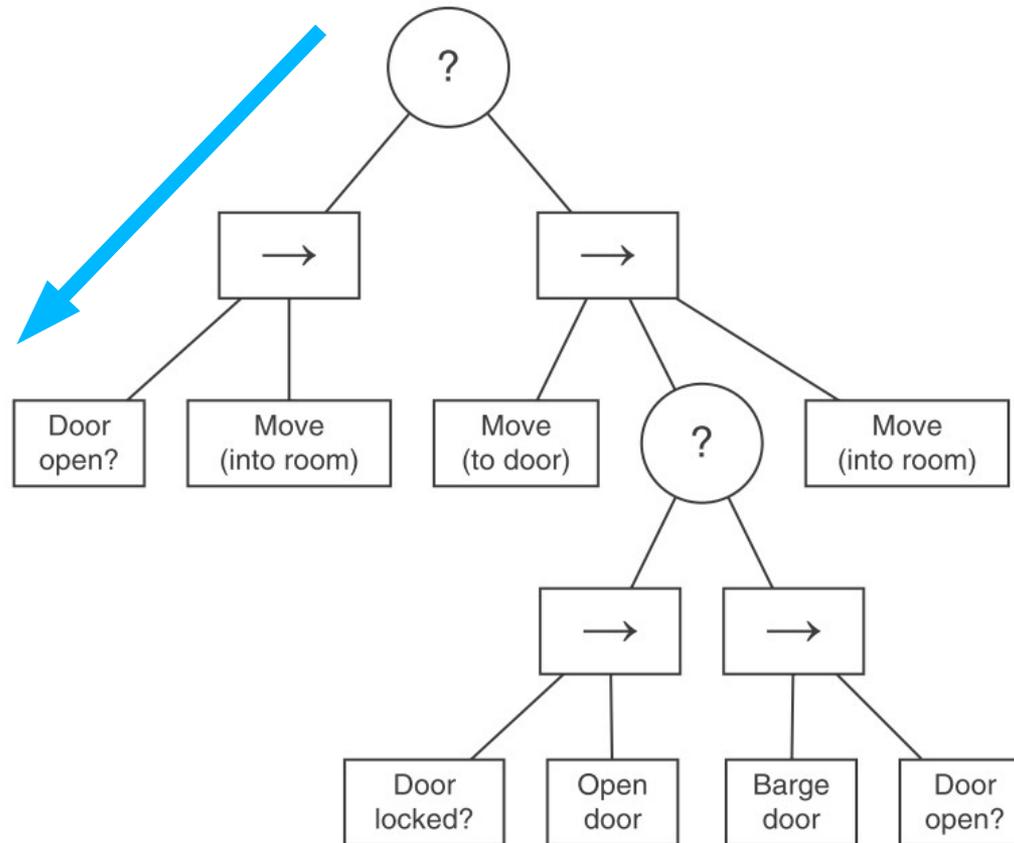
- Coder une suite d'actions
 - Behavior Trees



« Artificial intelligence for games » (Millington / Funge)

3.2 - Sélection d'objectifs

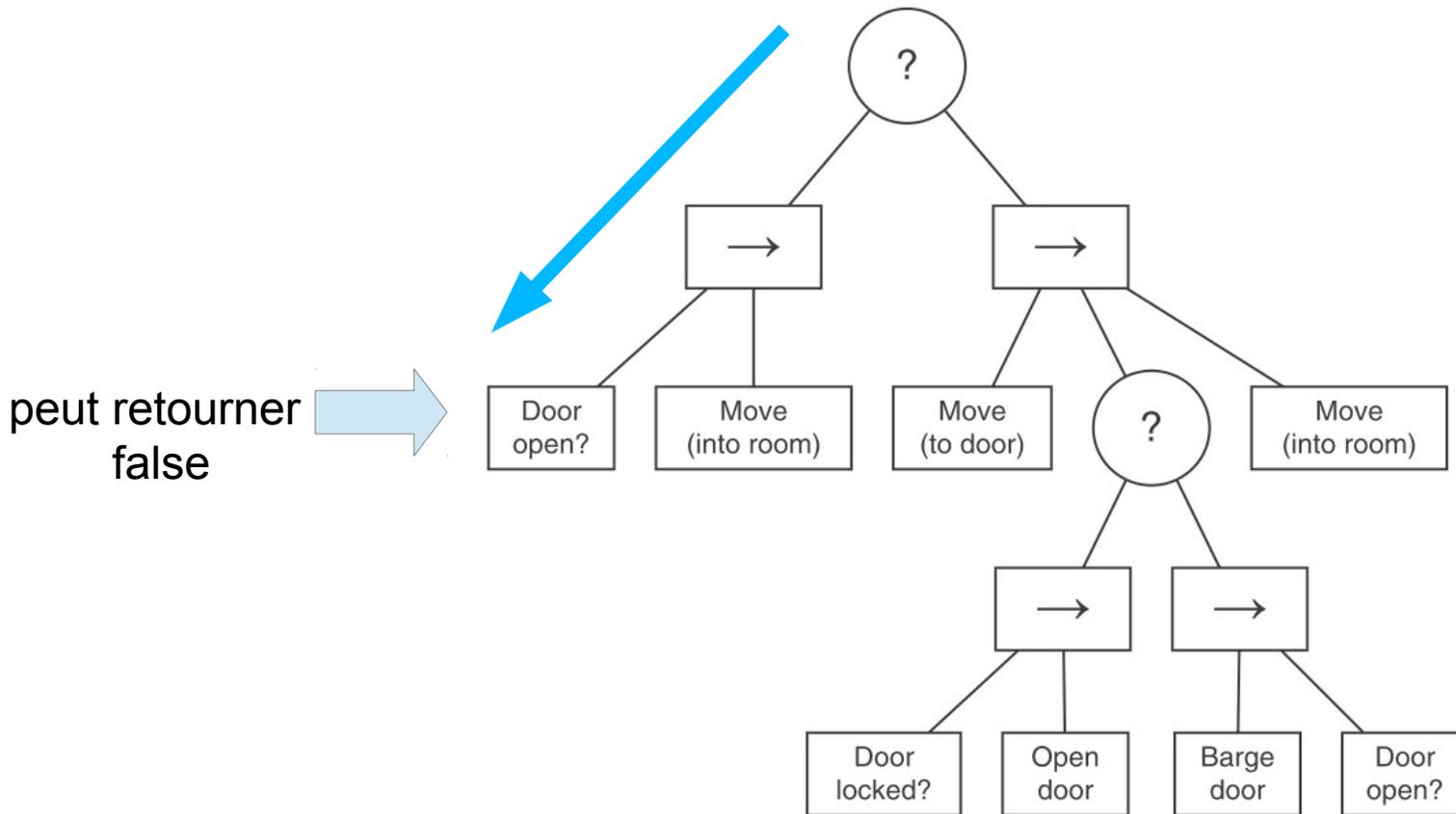
- Coder une suite d'actions
 - Behavior Trees



« Artificial intelligence for games » (Millington / Funge)

3.2 - Sélection d'objectifs

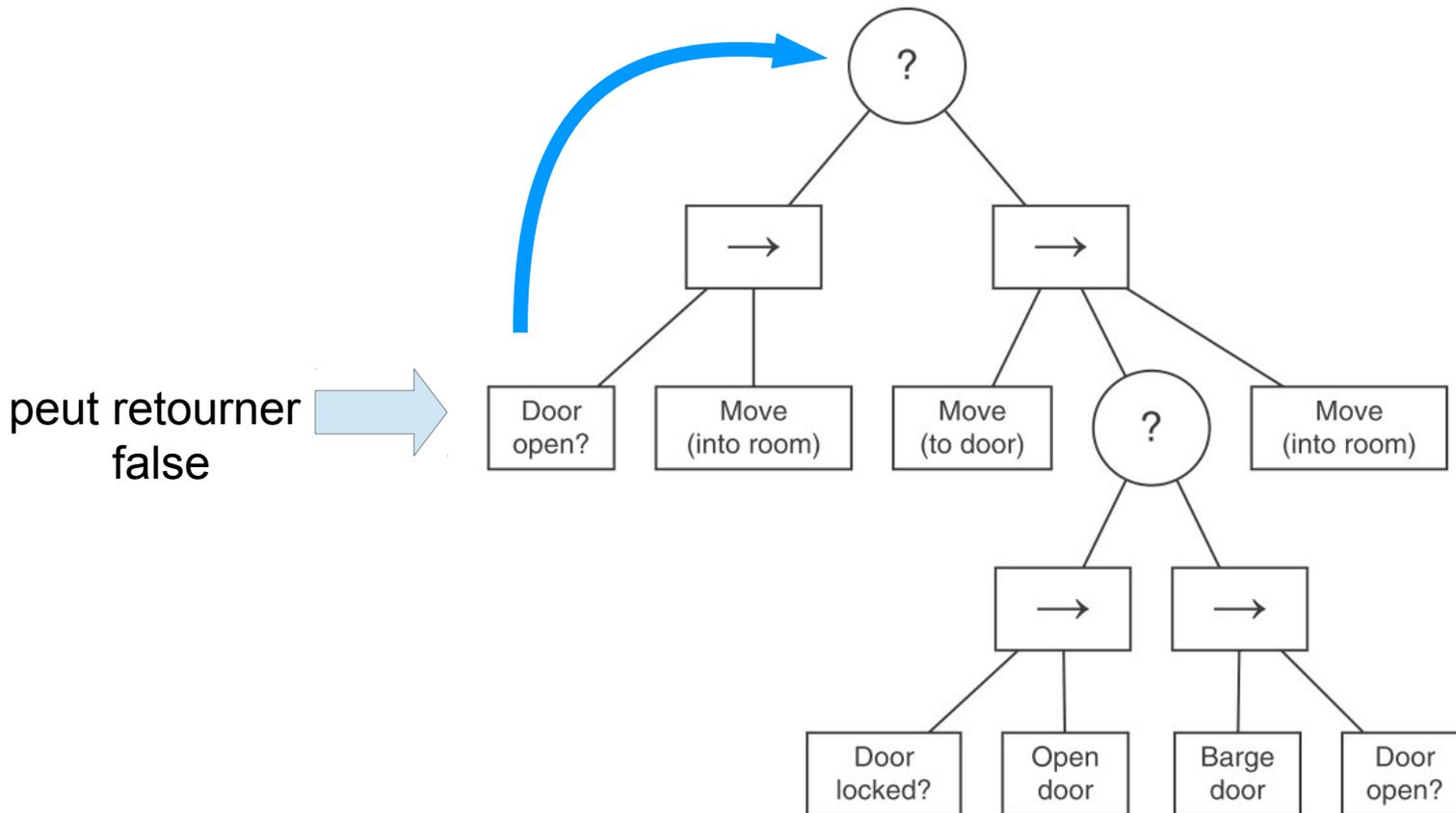
- Coder une suite d'actions
 - Behavior Trees



« Artificial intelligence for games » (Millington / Funge)

3.2 - Sélection d'objectifs

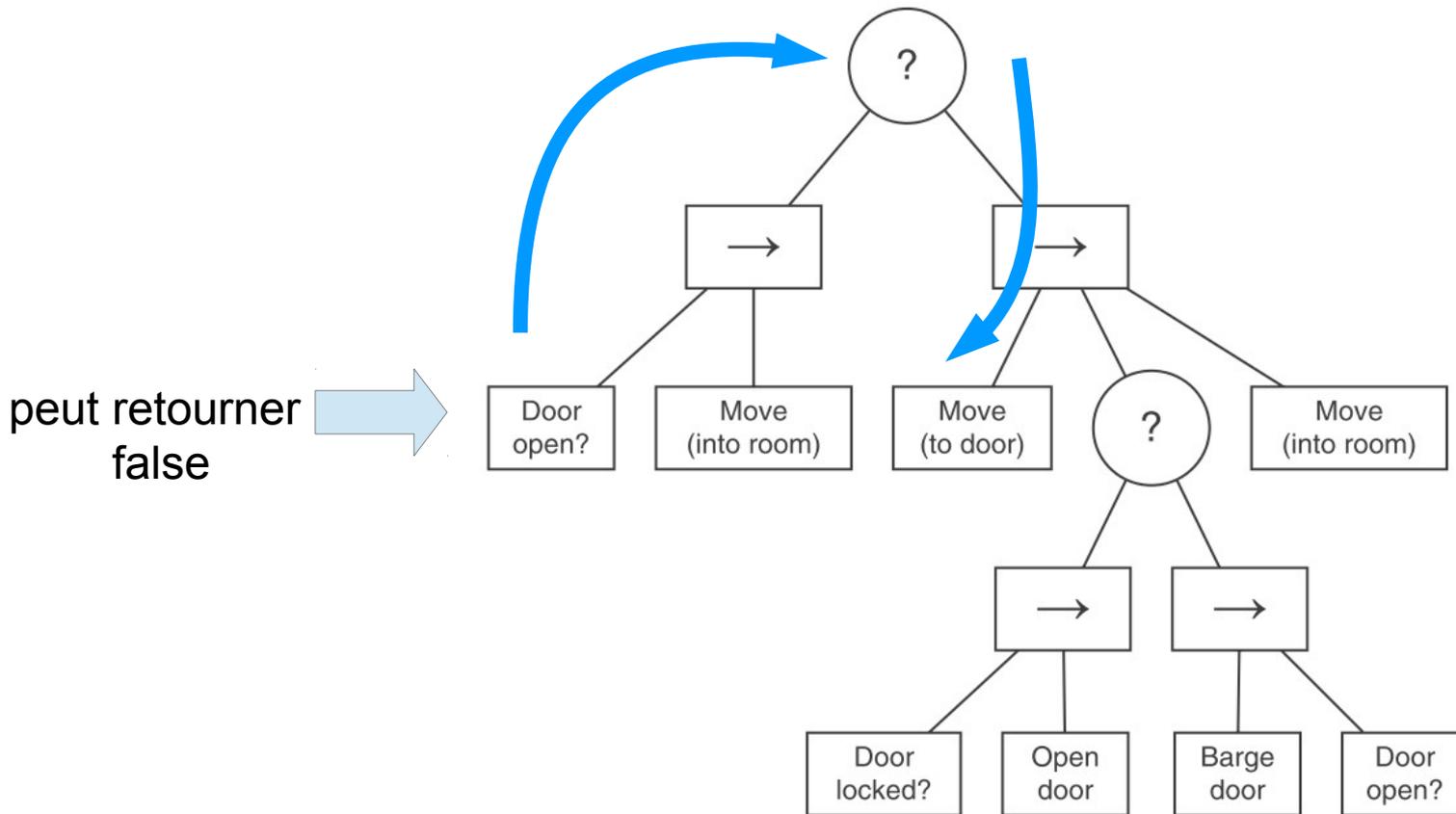
- Coder une suite d'actions
 - Behavior Trees



« Artificial intelligence for games » (Millington / Funge)

3.2 - Sélection d'objectifs

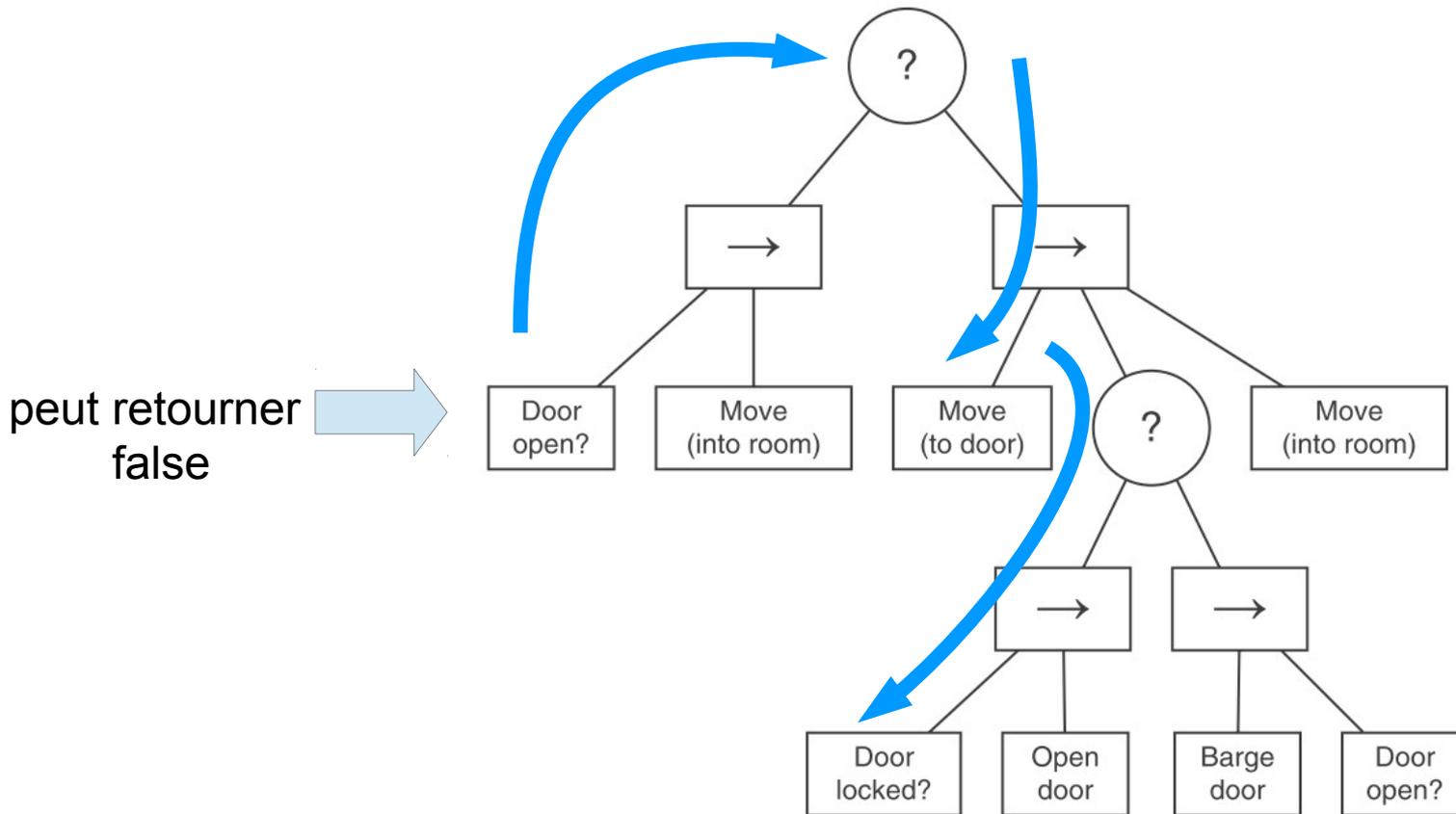
- Coder une suite d'actions
 - Behavior Trees



« Artificial intelligence for games » (Millington / Funge)

3.2 - Sélection d'objectifs

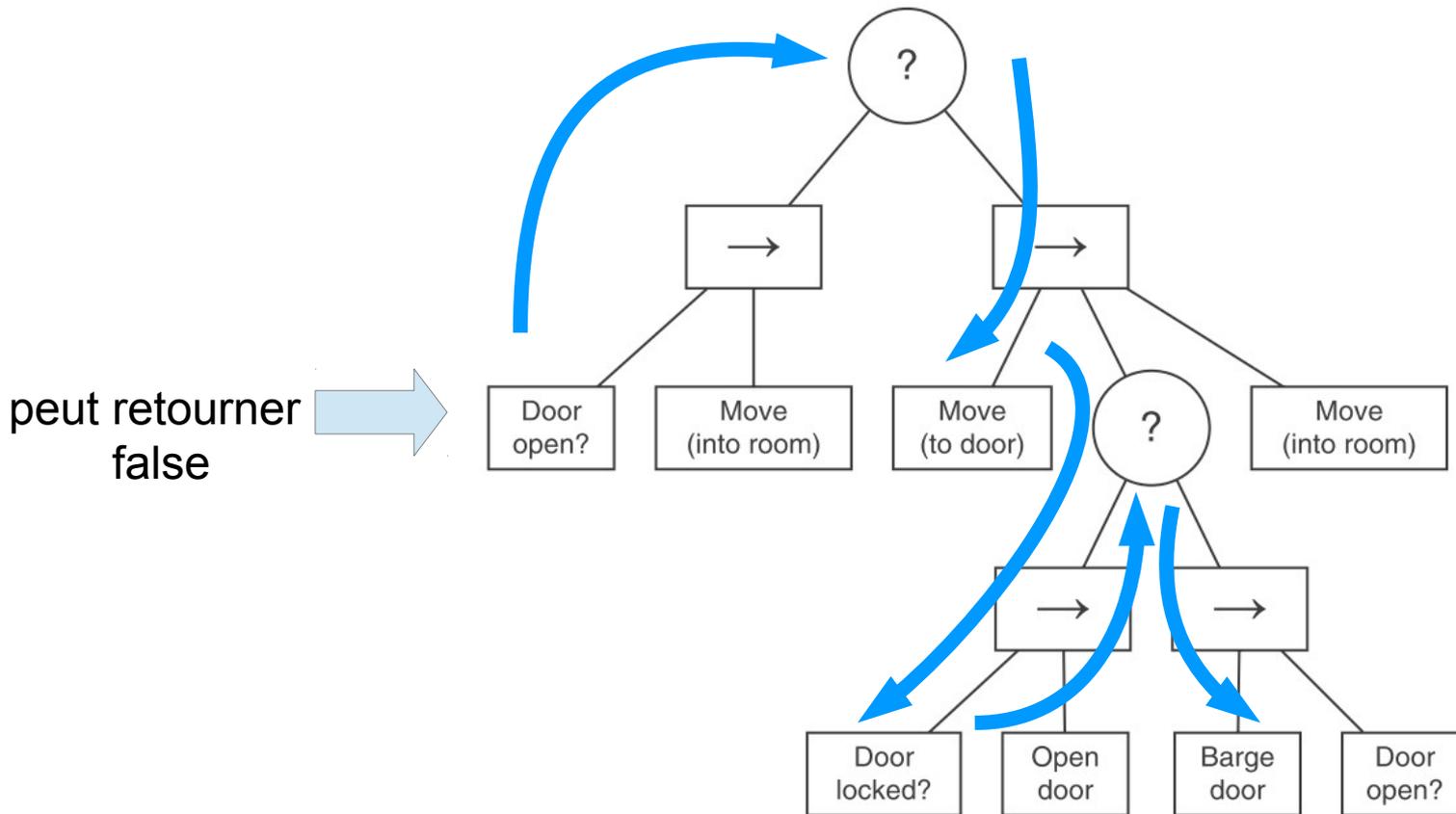
- Coder une suite d'actions
 - Behavior Trees



« Artificial intelligence for games » (Millington / Funge)

3.2 - Sélection d'objectifs

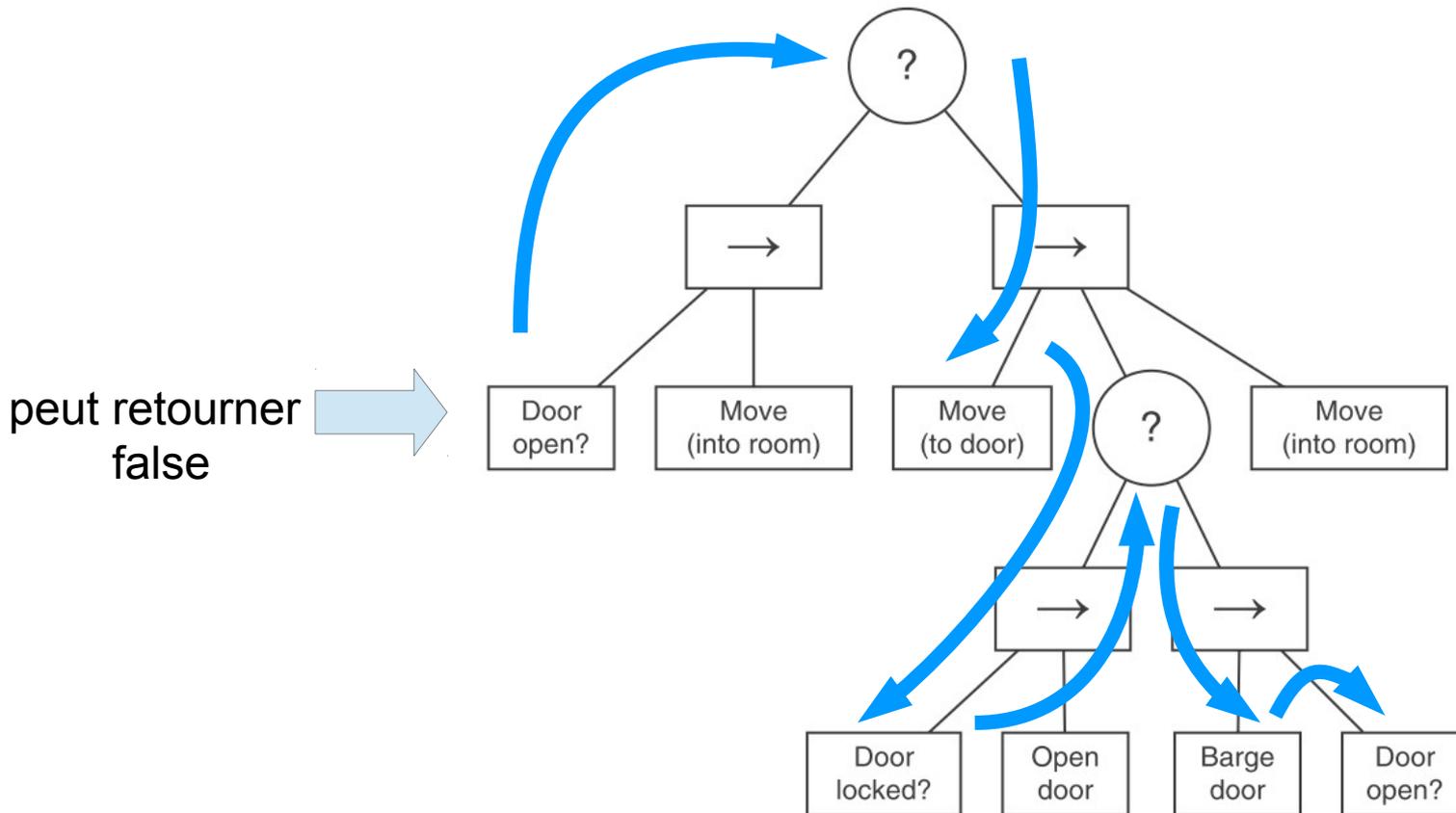
- Coder une suite d'actions
 - Behavior Trees



« Artificial intelligence for games » (Millington / Funge)

3.2 - Sélection d'objectifs

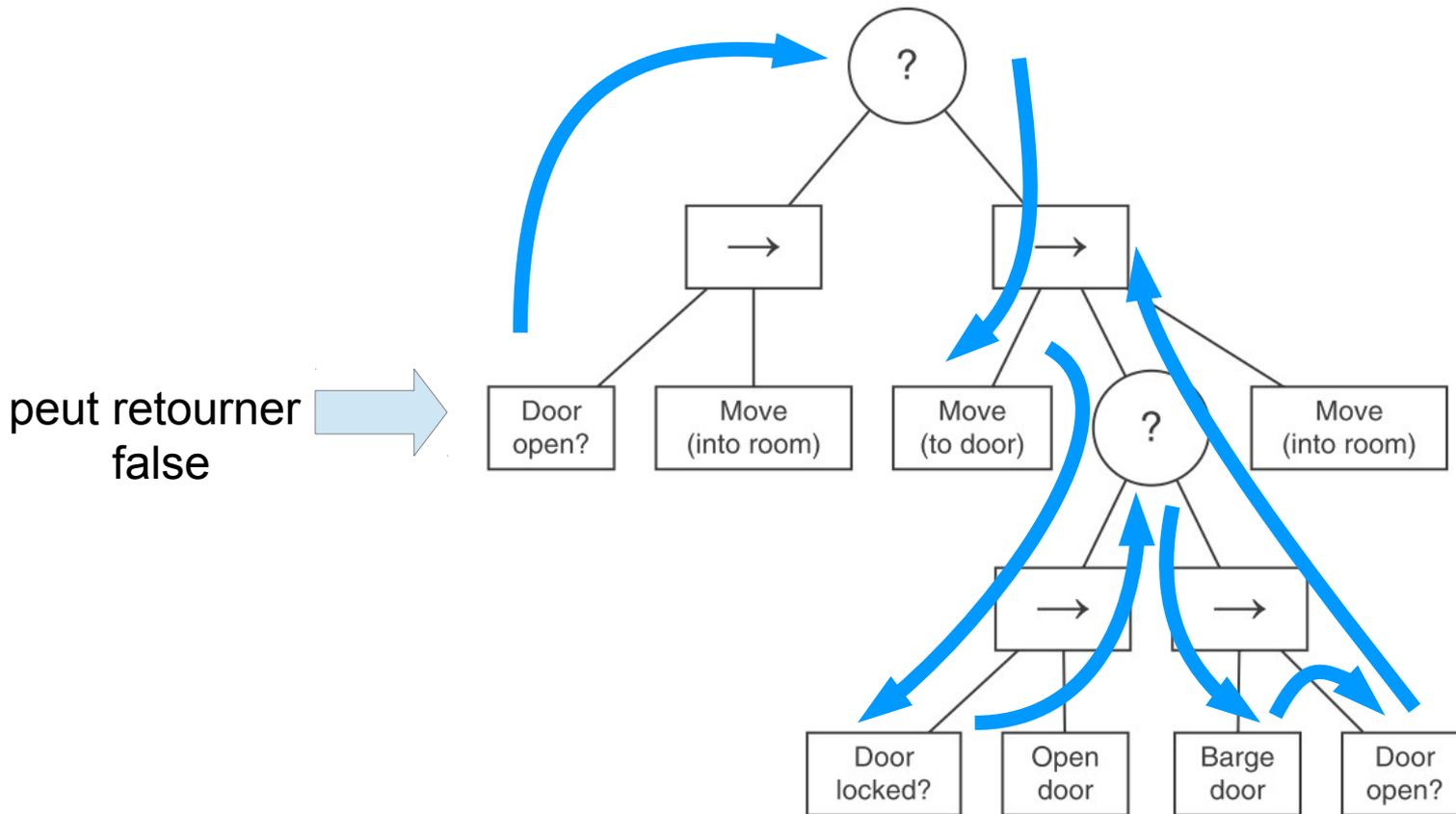
- Coder une suite d'actions
 - Behavior Trees



« Artificial intelligence for games » (Millington / Funge)

3.2 - Sélection d'objectifs

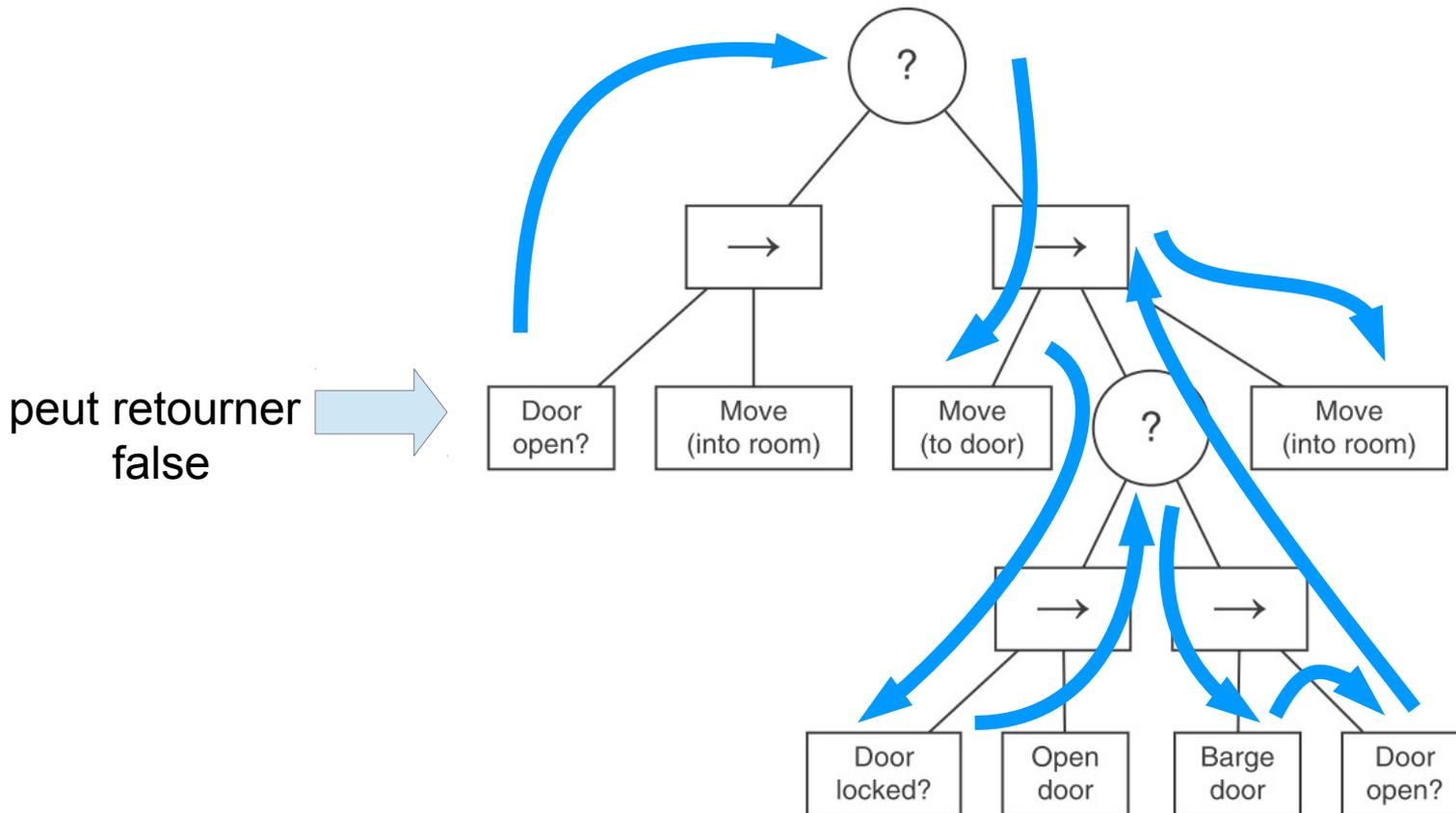
- Coder une suite d'actions
 - Behavior Trees



« Artificial intelligence for games » (Millington / Funge)

3.2 - Sélection d'objectifs

- Coder une suite d'actions
 - Behavior Trees



« Artificial intelligence for games » (Millington / Funge)

3.2 - Sélection d'objectifs

- **F.E.A.R**

- <http://web.cs.wpi.edu/~rich/courses/imgd4100-b11/lectures/L-Planning.pdf>
- http://web.media.mit.edu/~jorkin/gdc2006_orkin_jeff_fear.pdf



(2005)

- Halo2

- <http://aigamedev.com/open/review/halo-ai/>
- "Managing Complexity in the Halo 2 AI System" (D Isla)



(2004)

- Coder une suite d'actions

- Behavior Trees

- Arbre

- Conditionnelles

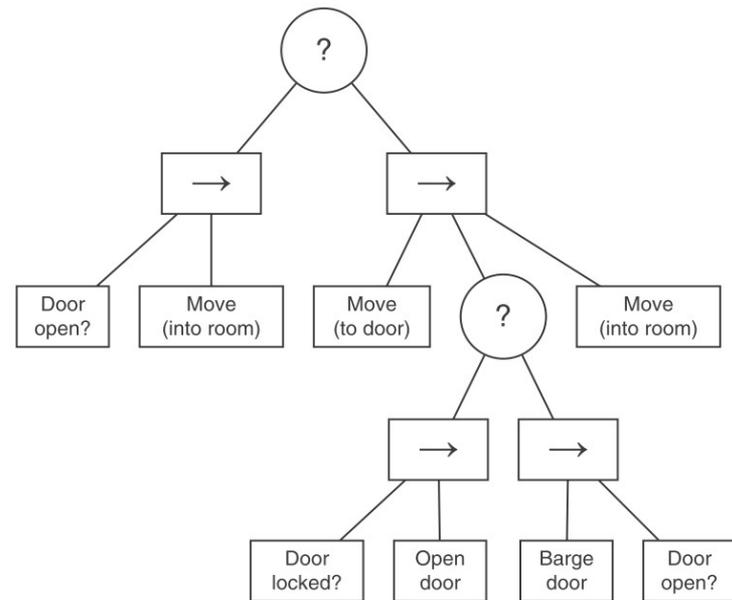
- Séquences d'actions

- Intérêts

- Langage expressif

- Comportement flexible

- Facilement lisible



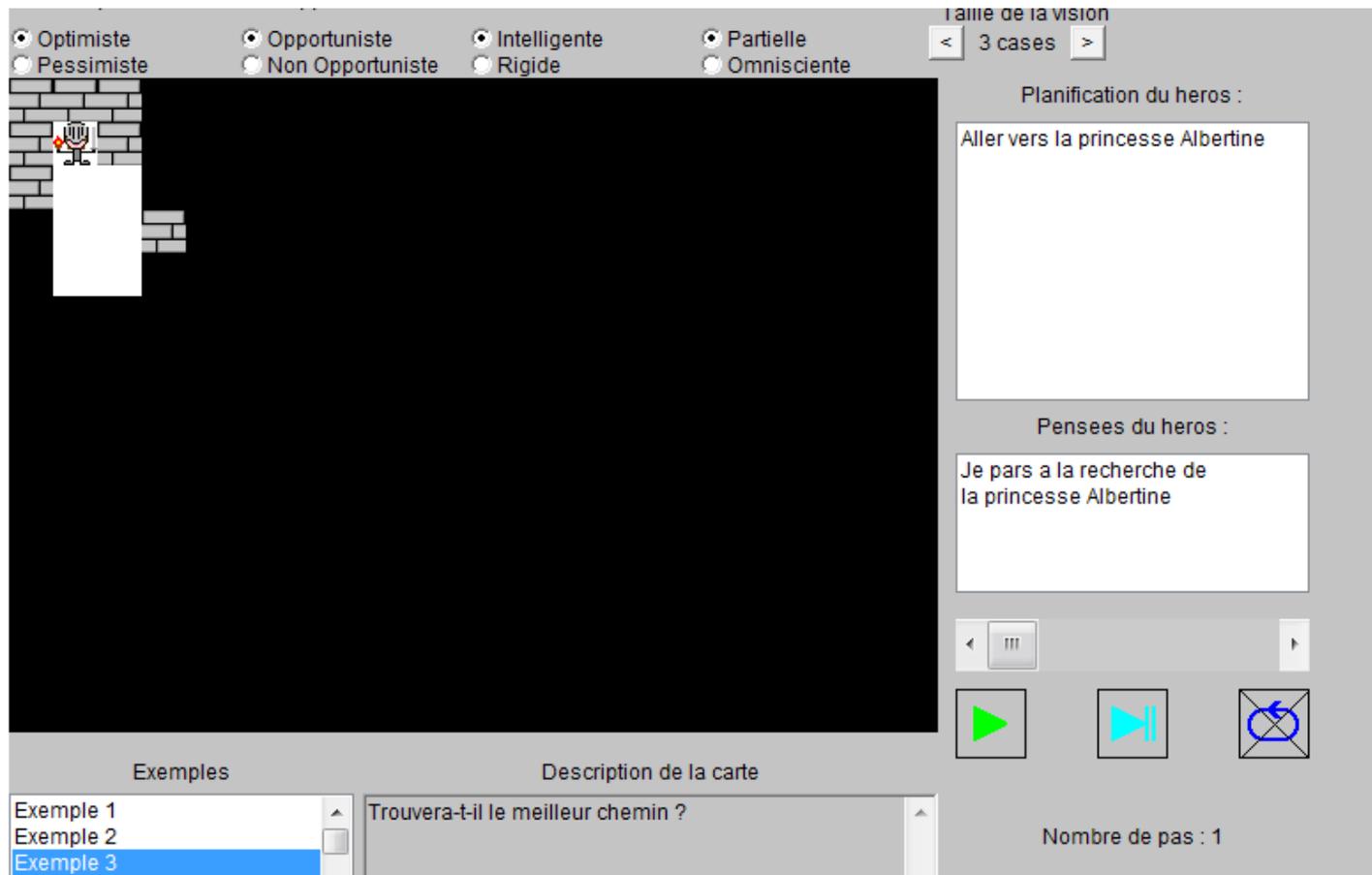
- Scripts
- Automates
- Behavior Tree
- Practical reasoning – BDI
- Planification haut niveau

- "Practical reasoning"(brattman)
 - Raisonnement humain

- "Practical reasoning"(brattman)
 - Raisonnement humain
- Exemple: Donner cours
 - Donner cours → Aller à l'université
 - Aller université → Prendre le véhicule
 - Prendre le véhicule → Trouver les clefs
- Trois éléments
 - Désir, Intentions, Croyances

- **Probleme de la princesse**

<http://www.lifl.fr/SMAC/projects/cocoa/princesse.html>



- **Probleme de la princesse**

<http://www.lifl.fr/SMAC/projects/cocoa/princesse.html>

The screenshot displays a game interface for 'Le Probleme de la Princesse'. At the top, there are radio buttons for character traits: Optimiste, Opportuniste, Intelligente, Partielle, Pessimiste, Non Opportuniste, Rigide, and Omnisciente. A 'Taille de la vision' (View Size) control is set to '3 cases'. The main area shows a maze with a character on a platform and a key on the floor. On the right, the 'Planification du heros' (Hero Planning) section contains the tasks: 'Trouver la princesse Albertine' and 'Aller vers la princesse Albertine'. Below it, the 'Pensees du heros' (Hero Thoughts) section shows the text: 'Je pars a la recherche de la princesse Albertine'. At the bottom left, there is a list of 'Exemples' (Examples) with 'Exemple 3' selected. The 'Description de la carte' (Map Description) section contains the question: 'Trouvera-t-il le meilleur chemin ?'. At the bottom right, the 'Nombre de pas : 12' (Number of steps: 12) is displayed. Navigation buttons for play, pause, and reset are also visible.

- **Probleme de la princesse**

<http://www.lifl.fr/SMAC/projects/cocoa/princesse.html>

The screenshot displays a game interface for 'Le Probleme de la Princesse'. At the top, there are several radio buttons for character traits: Optimiste, Opportuniste, Intelligente, Partielle, Pessimiste, Non Opportuniste, Rigide, and Omnisciente. A 'Taille de la vision' (Vision Size) control is set to '3 cases'. The main area shows a maze with a character (a robot) and a key. On the right, a 'Planification du heros' (Hero Planning) window contains the text: 'Trouver la princesse Albertine' and 'Aller vers la princesse Albertine'. Below it, a 'Pensees du heros' (Hero Thoughts) window shows: 'Peut-être que cette cle verte va me servir plus tard.' and 'Je pars a la recherche de la princesse Albertine'. At the bottom, there are controls for 'Exemples' (Exemple 1, Exemple 2, Exemple 3), a 'Description de la carte' (Map Description) window with the text 'Trouvera-t-il le meilleur chemin ?', and a 'Nombre de pas : 20' (Number of steps: 20) indicator.

- **Probleme de la princesse**

<http://www.lifl.fr/SMAC/projects/cocoa/princesse.html>

The screenshot shows a game interface for 'Le Probleme de la Princesse'. At the top, there are character traits: Optimiste, Opportuniste, Intelligente, Partielle, Pessimiste, Non Opportuniste, Rigide, and Omnisciente. A 'Taille de la vision' (vision size) control is set to '3 cases'. The main area is a maze with a character on the left and a key on the right. Below the maze, there are 'Exemples' (Exemple 1, Exemple 2, Exemple 3) and a 'Description de la carte' (Description of the map) section. On the right, there is a 'Planification du heros' (Hero's plan) section with the following steps: 'Prendre la cle verte', 'Actionner la porte verte', 'Trouver la princesse Albertine', and 'Aller vers la princesse Albertine'. Below that is a 'Pensees du heros' (Hero's thoughts) section with the text: 'Peut-être que cette cle verte va me servir plus tard.' and 'Je pars a la recherche de la princesse Albertine'. At the bottom right, there are navigation buttons (play, pause, reset) and a 'Nombre de pas : 33' (Number of steps: 33) indicator.

- **Probleme de la princesse**

<http://www.lifl.fr/SMAC/projects/cocoa/princesse.html>

The screenshot shows a game interface for 'Le Probleme de la Princesse'. The main area is a maze with a hero character (a small figure with a red key) and a princess (a small figure with a crown). The maze has two large black areas representing obstacles. The interface includes several panels:

- Top Panel:** Character traits: Optimiste, Opportuniste, Intelligente, Partielle, Pessimiste, Non Opportuniste, Rigide, Omnisciente. A 'Taille de la vision' (Vision Size) control is set to 3 cases.
- Planification du heros (Hero Planning):**
 - Aller vers la cle rouge
 - Prendre la cle rouge
 - Trouver la princesse Albertine
 - Aller vers la princesse Albertine
- Pensees du heros (Hero Thoughts):**
 - Peut-être que cette cle rouge va me servir plus tard.
 - J'ai reussi a actionner la porte verte grace a la cle
- Bottom Panel:** 'Exemples' (Exemples) and 'Description de la carte' (Description de la carte). The description is 'Trouvera-t-il le meilleur chemin ?' (Will he find the best path?).
- Bottom Right:** 'Nombre de pas : 34' (Number of steps: 34).

3.2 - Sélection d'objectifs

197
244

- **Probleme de la princesse**

<http://www.lifl.fr/SMAC/projects/cocoa/princesse.html>

The screenshot displays a game interface for 'Probleme de la princesse'. At the top, there are radio buttons for character traits: Optimiste, Opportuniste, Intelligente, Partielle, Pessimiste, Non Opportuniste, Rigide, and Omnisciente. A 'Taille de la vision' (Vision Size) control is set to '3 cases'. The main area shows a maze with a character (a knight) and a yellow key. On the right, a 'Planification du heros' (Hero Planning) box lists goals: 'Aller vers la cle jaune', 'Prendre la cle jaune', 'Trouver la princesse Albertine', and 'Aller vers la princesse Albertine'. Below it, a 'Pensees du heros' (Hero Thoughts) box contains the text: 'Peut-être que cette cle jaune va me servir plus tard.' and 'J'ai réussi a actionner la porte rouge grace a la cle'. At the bottom, there are navigation buttons (play, pause, reset) and a 'Nombre de pas : 54' (Number of steps: 54) indicator. A bottom panel shows 'Exemples' (Examples) with 'Exemple 3' selected and a 'Description de la carte' (Map Description) box containing the text 'Trouvera-t-il le meilleur chemin ?'.

- **Probleme de la princesse**

<http://www.lifl.fr/SMAC/projects/cocoa/princesse.html>

The screenshot shows a game interface for 'Le Probleme de la Princesse'. The main area is a maze with a hero character in the center and a princess character in a room below. The interface includes several panels:

- Top Panel:** Character traits: Optimiste, Opportuniste, Intelligente, Partielle, Pessimiste, Non Opportuniste, Rigide, Omnisciente. A 'Taille de la vision' slider is set to '3 cases'.
- Planification du heros :** A list of goals: 'Trouver la princesse Albertine' and 'Aller vers la princesse Albertine'.
- Pensees du heros :** A text box containing: 'J'ai reussi a actionner la porte jaune grace a la cle jaune' and 'Peut-être que cette cle jaune'.
- Bottom Left:** 'Exemples' list with 'Exemple 1', 'Exemple 2', and 'Exemple 3' (selected).
- Bottom Center:** 'Description de la carte' with the text 'Trouvera-t-il le meilleur chemin ?'.
- Bottom Right:** 'Nombre de pas : 71' and navigation buttons (play, pause, reset).

- **Probleme de la princesse**

<http://www.lifl.fr/SMAC/projects/cocoa/princesse.html>

The screenshot shows a game interface for 'Princesse'. At the top, there are radio buttons for character traits: Optimiste, Opportuniste, Intelligente, Partielle, Pessimiste, Non Opportuniste, Rigide, and Omnisciente. A 'Taille de la vision' slider is set to '3 cases'. The main area is a maze with a hero (a knight) and a blue key. A yellow door is visible. On the right, the 'Planification du heros' panel lists: 'Aller vers la cle bleue', 'Prendre la cle bleue', 'Trouver la princesse Albertine', and 'Aller vers la princesse Albertine'. Below it, the 'Pensees du heros' panel shows: 'Peut-être que cette cle bleue va me servir plus tard.' and 'J'ai reussi a actionner la porte jaune grace a la cle'. At the bottom, there are 'Exemples' (Exemple 1, Exemple 2, Exemple 3) and a 'Description de la carte' (Trouvera-t-il le meilleur chemin ?). The 'Nombre de pas' is 83.

3.2 - Sélection d'objectifs

200
244

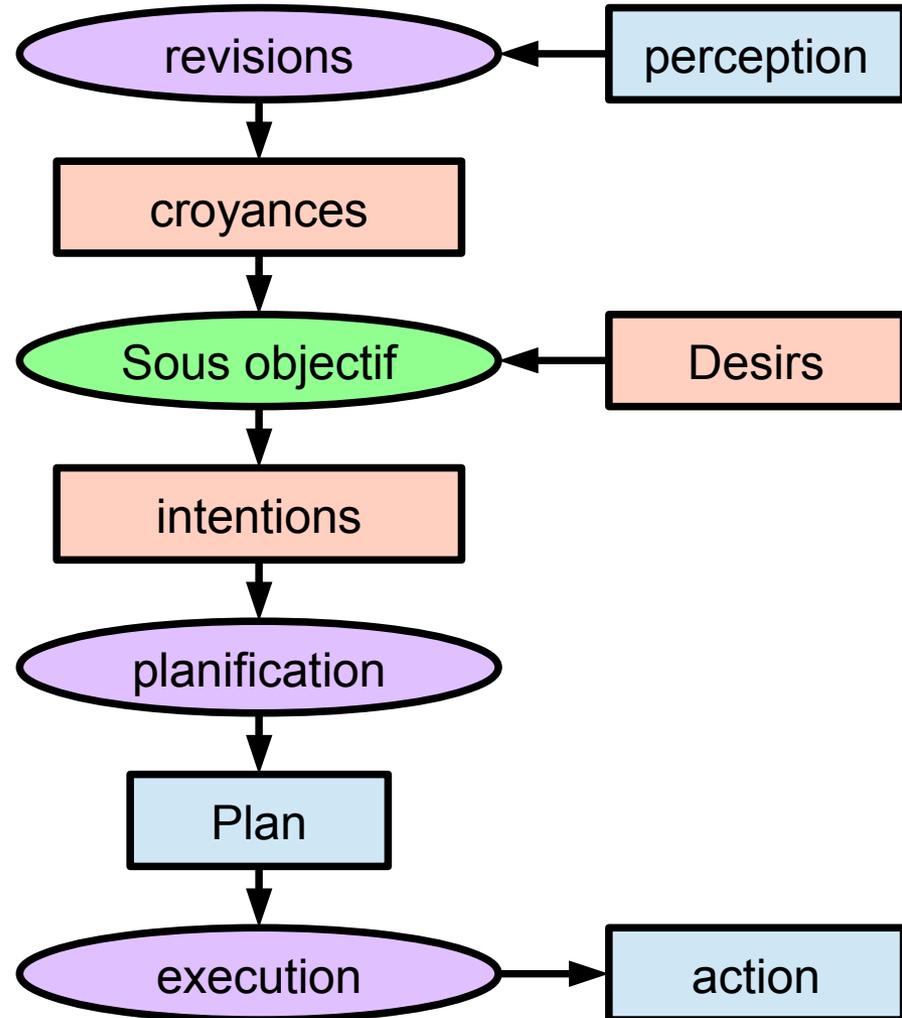
- **Probleme de la princesse**

<http://www.lifl.fr/SMAC/projects/cocoa/princesse.html>

The screenshot shows a game interface for 'Le Probleme de la Princesse'. At the top, there are several radio buttons for character traits: Optimiste, Opportuniste, Intelligente, Partielle, Pessimiste, Non Opportuniste, Rigide, and Omnisciente. A 'Taille de la vision' (Vision Size) control is set to '3 cases'. The main area is a maze with a princess character and a knight. On the right, there is a 'Planification du heros' (Hero Planning) section with the text 'Aller vers la princesse Albertine'. Below it is a 'Pensees du heros' (Hero Thoughts) section with a text area containing 'J'ai reussi a actionner la porte bleue grace a la cle bleue' and 'Peut-être que cette cle bleue'. At the bottom, there are navigation buttons (play, pause, reset) and a 'Nombre de pas : 90' (Number of steps: 90) indicator. A bottom panel shows 'Exemples' (Examples) with a list containing 'Exemple 1', 'Exemple 2', and 'Exemple 3', and a 'Description de la carte' (Map Description) section with the text 'Trouvera-t-il le meilleur chemin ?'.

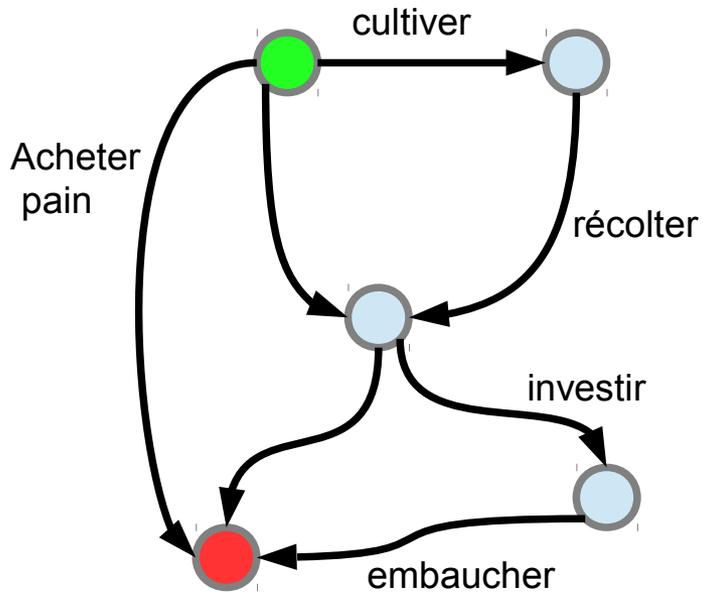
3.2 - Sélection d'objectifs

- Architecture BDI



- Scripts
- Automates
- Behavior Tree
- Practical reasoning – BDI
- Planification haut niveau

- Planification haut-niveau



Settlers 7

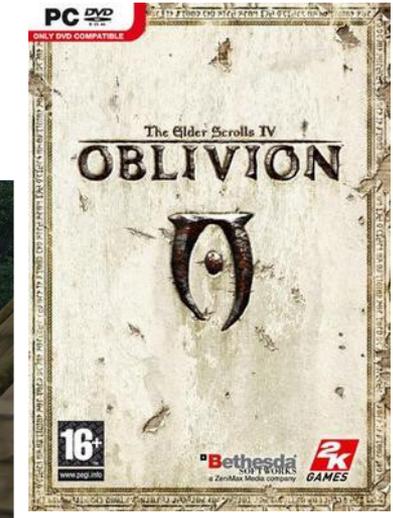
3.2 - Sélection d'objectifs

- Imaginer mondé cohérent ?
 - Radiant AI – horaires et vie autonome



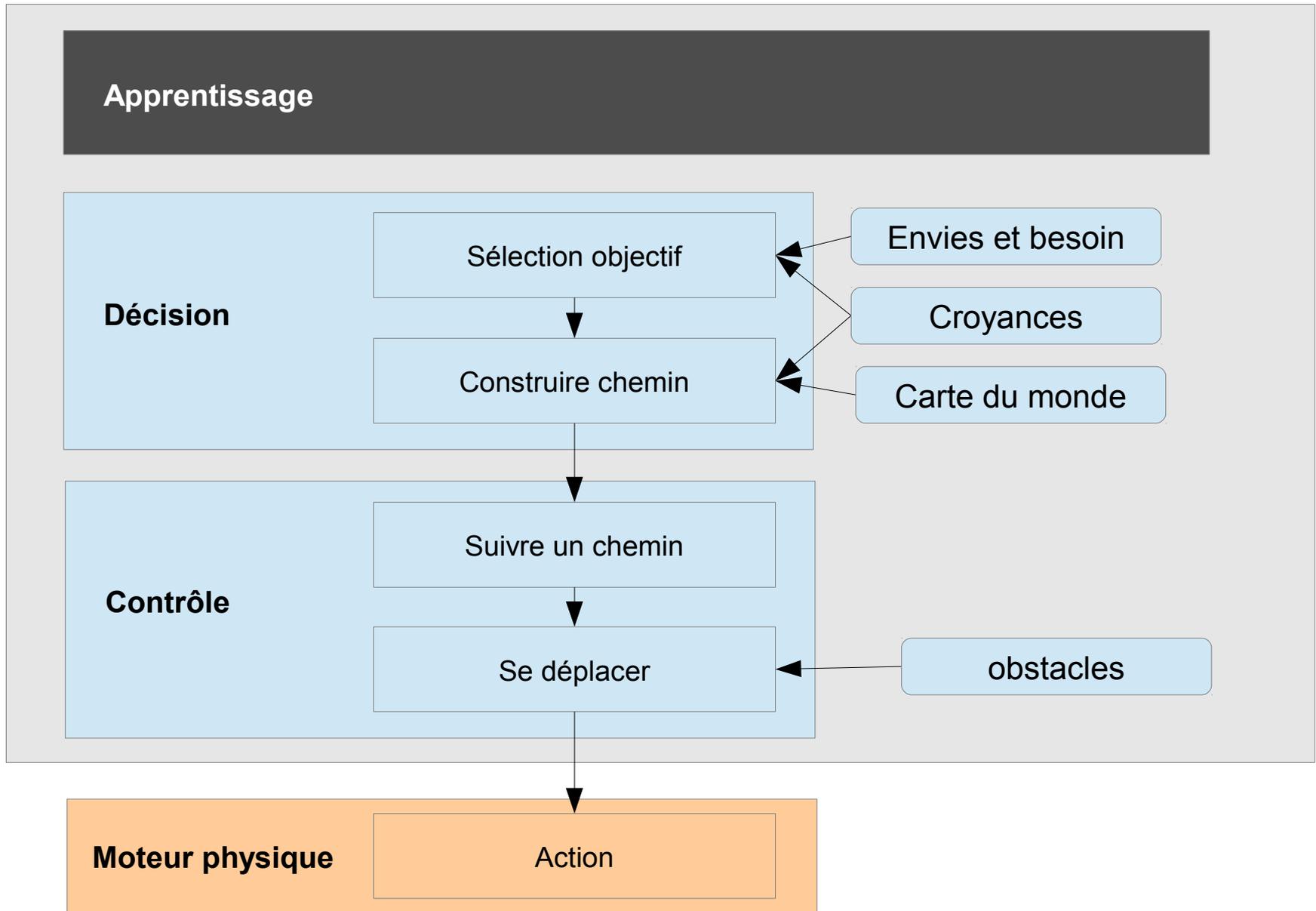
3.2 - Sélection d'objectifs

- Imaginer mondé cohérent ?
 - Radiant AI – horaires et vie autonome

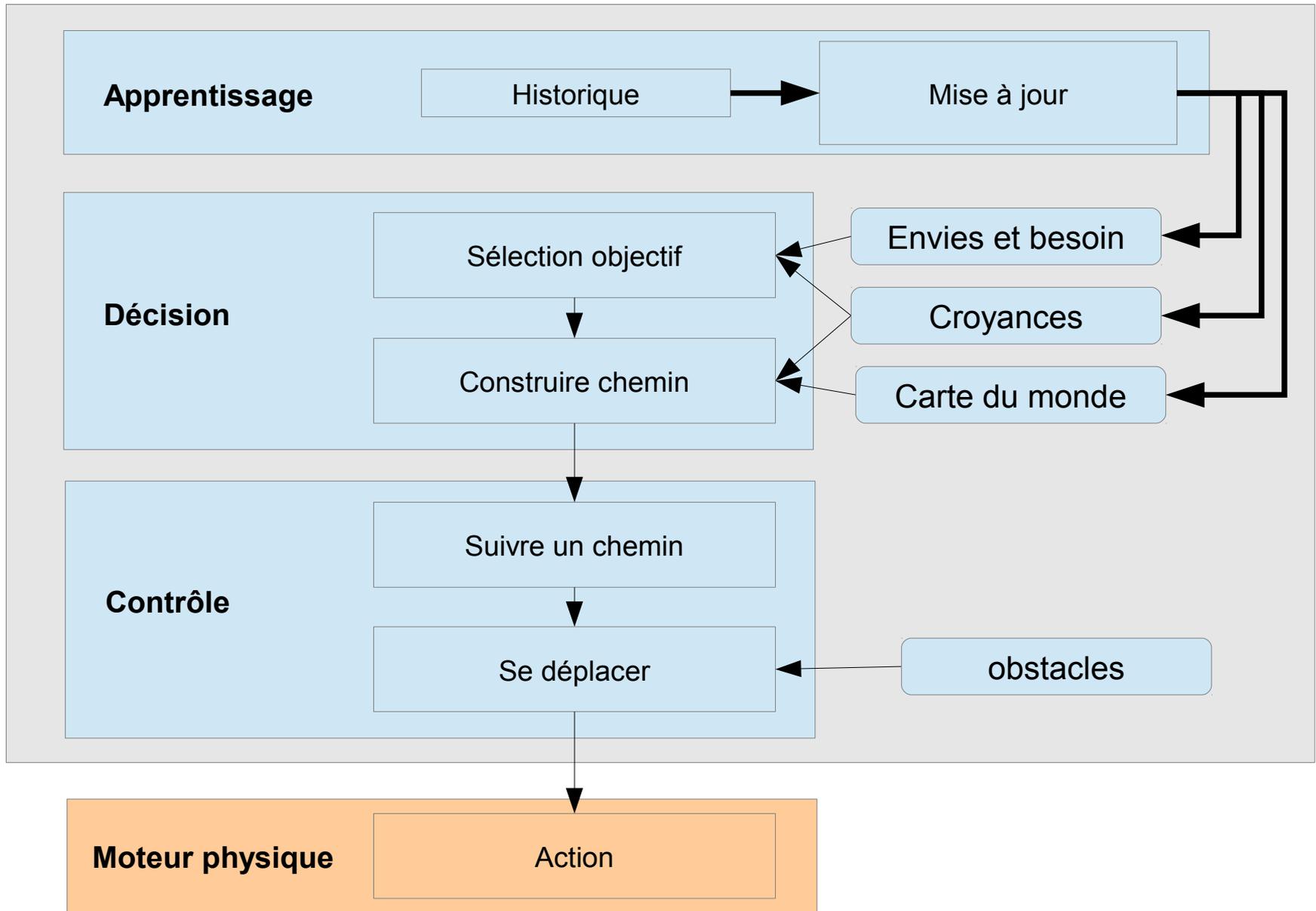


Shadow over Hackdirt

3 – couche Décision



4 – Couche apprentissage



- **Question**

- Pourquoi apprendre dans un jeu ?
- Temps de réponse ?

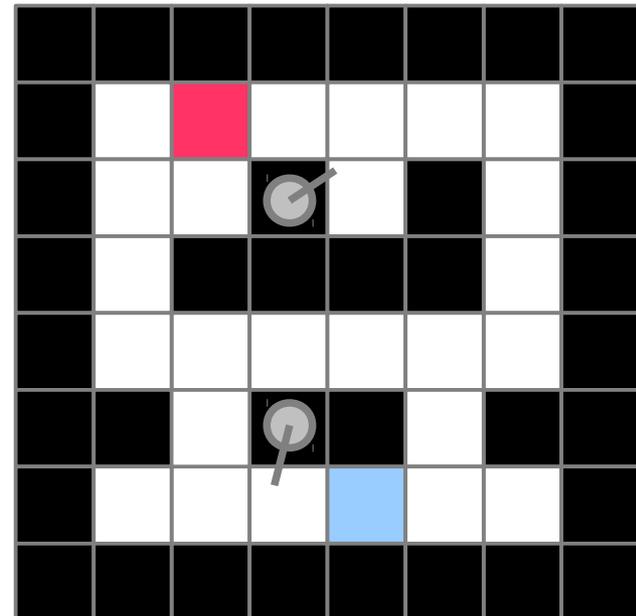
- **Objectifs**

- Mettre à jour connaissance du monde
- S'adapter aux actions du joueurs
- Simuler le vivant

- **Tower defense intelligent**
 - Mise à jour zone de risque
 - Retour vers planification



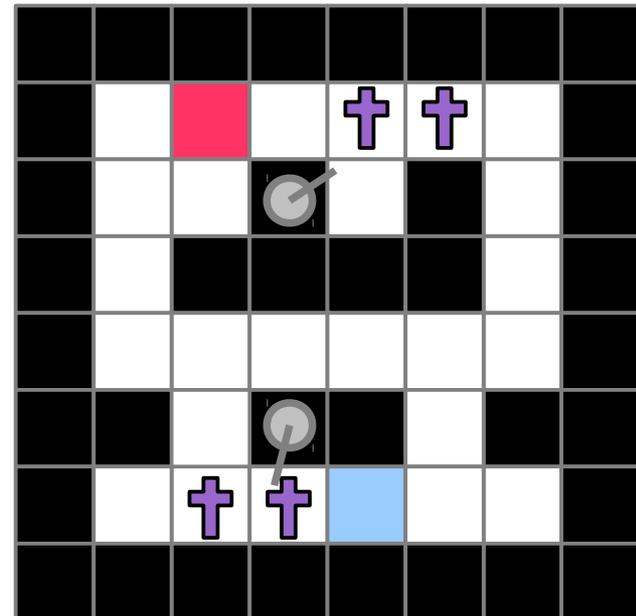
Desktop tower defense



- **Tower defense intelligent**
 - Mise à jour zone de risque
 - Retour vers planification



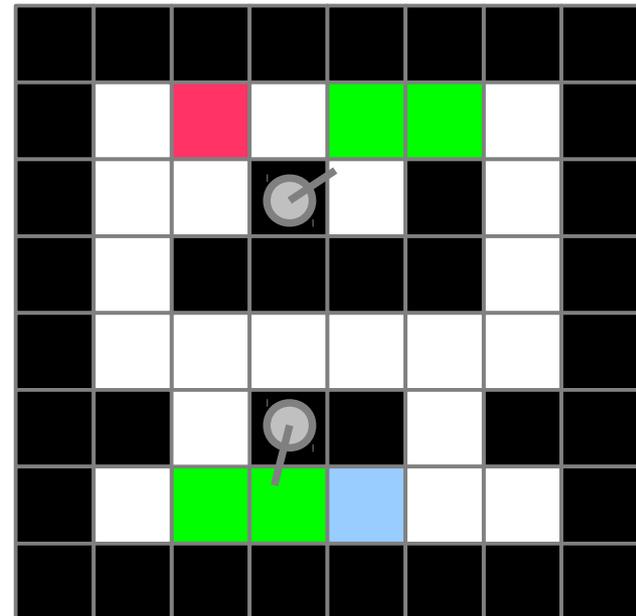
Desktop tower defense



- **Tower defense intelligent**
 - Mise à jour zone de risque
 - Retour vers planification



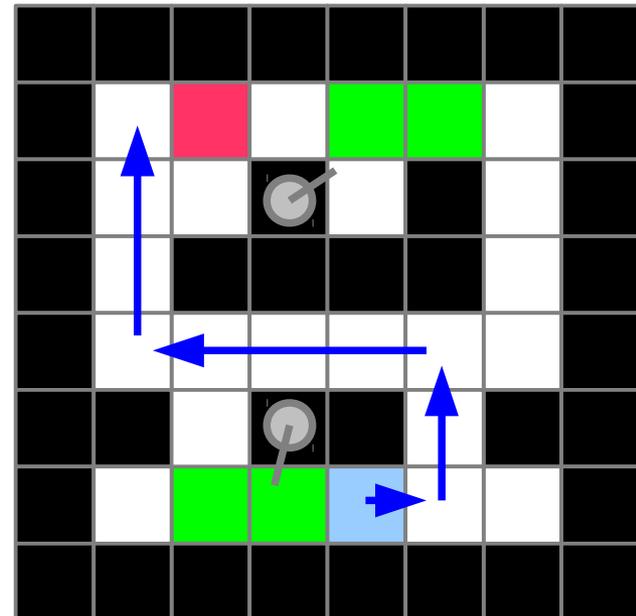
Desktop tower defense



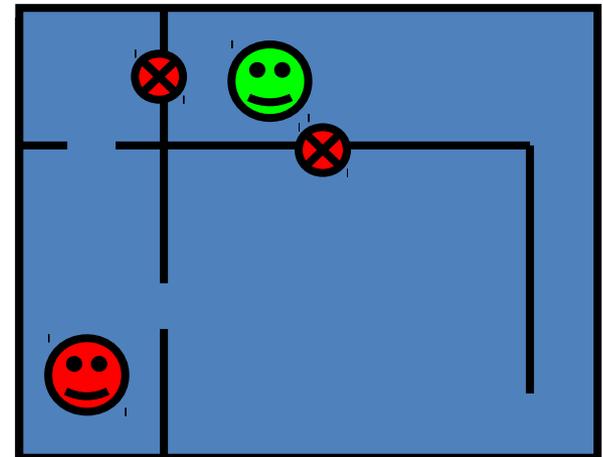
- **Tower defense intelligent**
 - Mise à jour zone de risque
 - Retour vers planification



Desktop tower defense



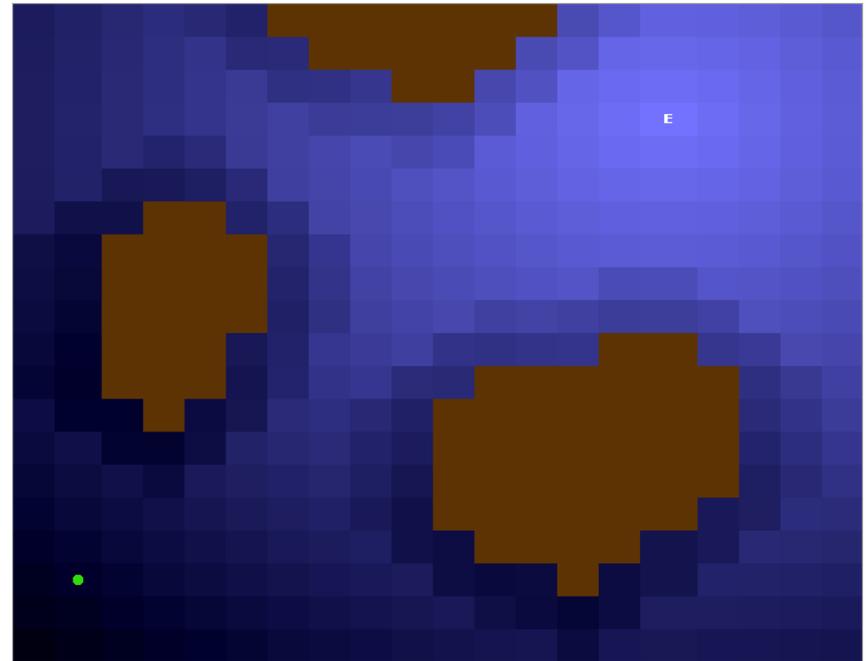
- **Gestion de l'information partielle**
 - Information inconnue
 - Gestion connaissance
 - Re-planifier quand contradiction
 - Proie a disparu
 - Ex: Porte fermée non attendue
 - Soit contourner
 - Soit detruire accès avec hache



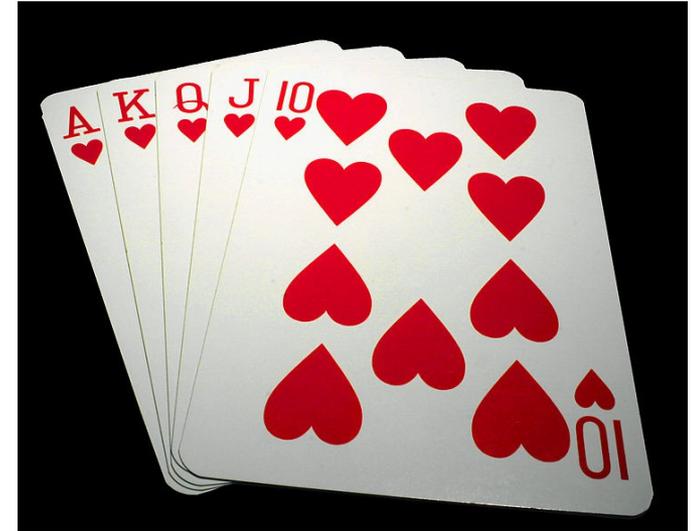
- Champs de potentiel

- <http://aigamedev.com/open/tutorial/potential-fields/>

0	2	6	2	0	0	0	0	0
2	6	8	6	2	0	0	0	0
6	8	10	8	6	2	0	0	0
2	6	8	6	2	0	-2	0	0
0	2	6	2	0	-2	-6	-2	0
0	0	2	0	-2	-6	-8	-6	-2
0	0	0	-2	-6	-8	-10	-8	-6
0	0	0	0	-2	-6	-8	-6	-2



- **Poker**
- **Principe**
 - Jeu de bluff
 - Comprendre habitudes
- **Apprendre**
 - Le comportement du joueur
 - Timoré, Casse-cou, bluffeur, ...
 - Prédire son jeu



- **Pierre feuille ciseaux**

- Stratégie optimale
- Jouer au hasard



- **Application web (NY Times)**

- Statistiques (200 000 rounds)
- Feuille 29,6 % Pierre 35,4 % Ciseaux 35 %

- **Pierre feuille ciseaux**

- Stratégie optimale
- Jouer au hasard



- **Robot université Tokyo (2012)**

- Gagne 100 % des cas
- Cf vidéo



4.3 – Simuler le vivant

219
244

- Black and white



(2001)

- Black and white
- Apprendre comportement
 - Créature observe
 - Créature répète
 - Joueur renforce





4.3 – Simuler le vivant



- Black and white
- Apprendre comportement
 - Créature observe
 - Créature répète
 - Joueur renforce
- Techniques
 - Architecture BDI
 - Apprentissage par renforcement / Réseau neurone
 - Valeur associée à couples (Situation → Action)



- **Comprendre le joueur**
 - Ses attentes
 - Ses principes de résolutions

- **Construire une histoire adaptée**
 - Aide à la scénarisation

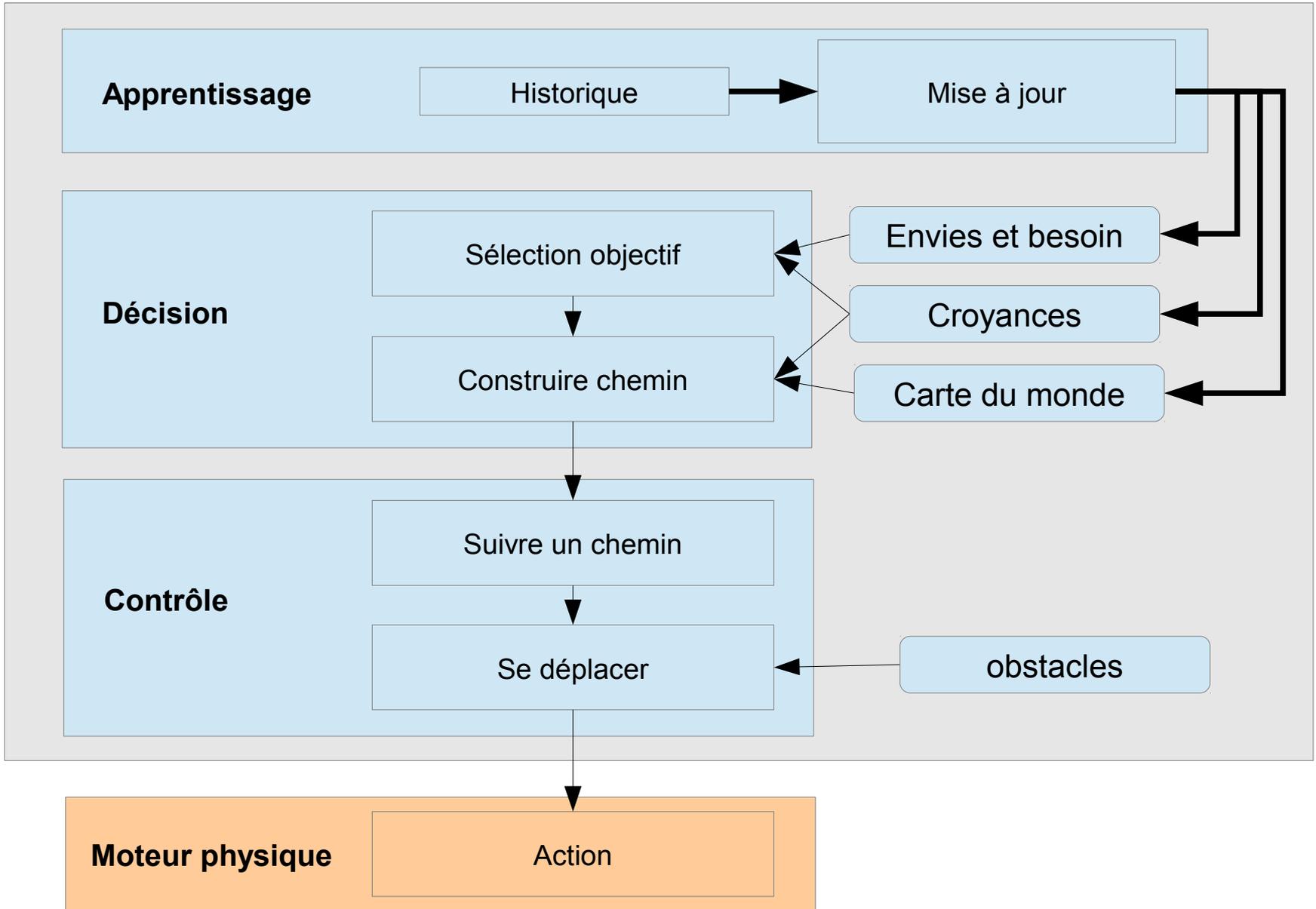
- **IA Director**
 - Construit l'histoire
 - Cent mille milliards poèmes
- **Exemple**
 - Left for dead
- **Thèses sur le sujet**
 - Machines à histoires
 - Guylain Delmas



- **Skyrim**
 - Radiant AI - Quêtes adaptées, Templates
 - {PNJ ami} enlevé dans {lieu inconnu}

Conclusion

226
244



- **Objectifs de départ**
 - Proposer un adversaire au joueur
 - Renouveler le jeu
 - Proposer des défis au joueur
 - Assurer une cohérence
 - Construire un système complexe

- **Différents niveaux**
 - IA complexe / simple
 - IA fixe / adaptative
 - IA peut tricher

Comportements de groupe

228
244



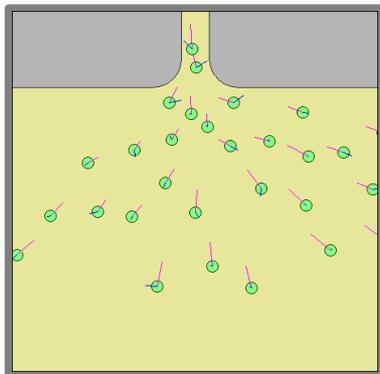
- **De nombreuses questions**
 - Déplacement en formation
 - Coordination pour attaque
 - Squads et couverture mutuelle



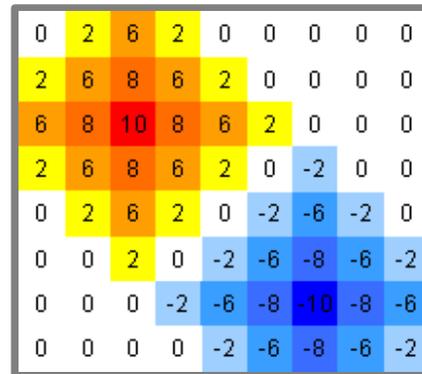
- De nombreuses questions
 - Déplacement en formation
 - Coordination pour attaque
 - Squads et couverture mutuelle



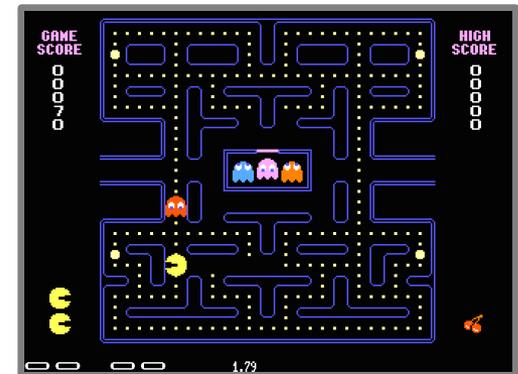
- Des pistes de solution



Boids



Champs de potentiel



Autres comme obstacles

3.2 - Sélection d'objectifs

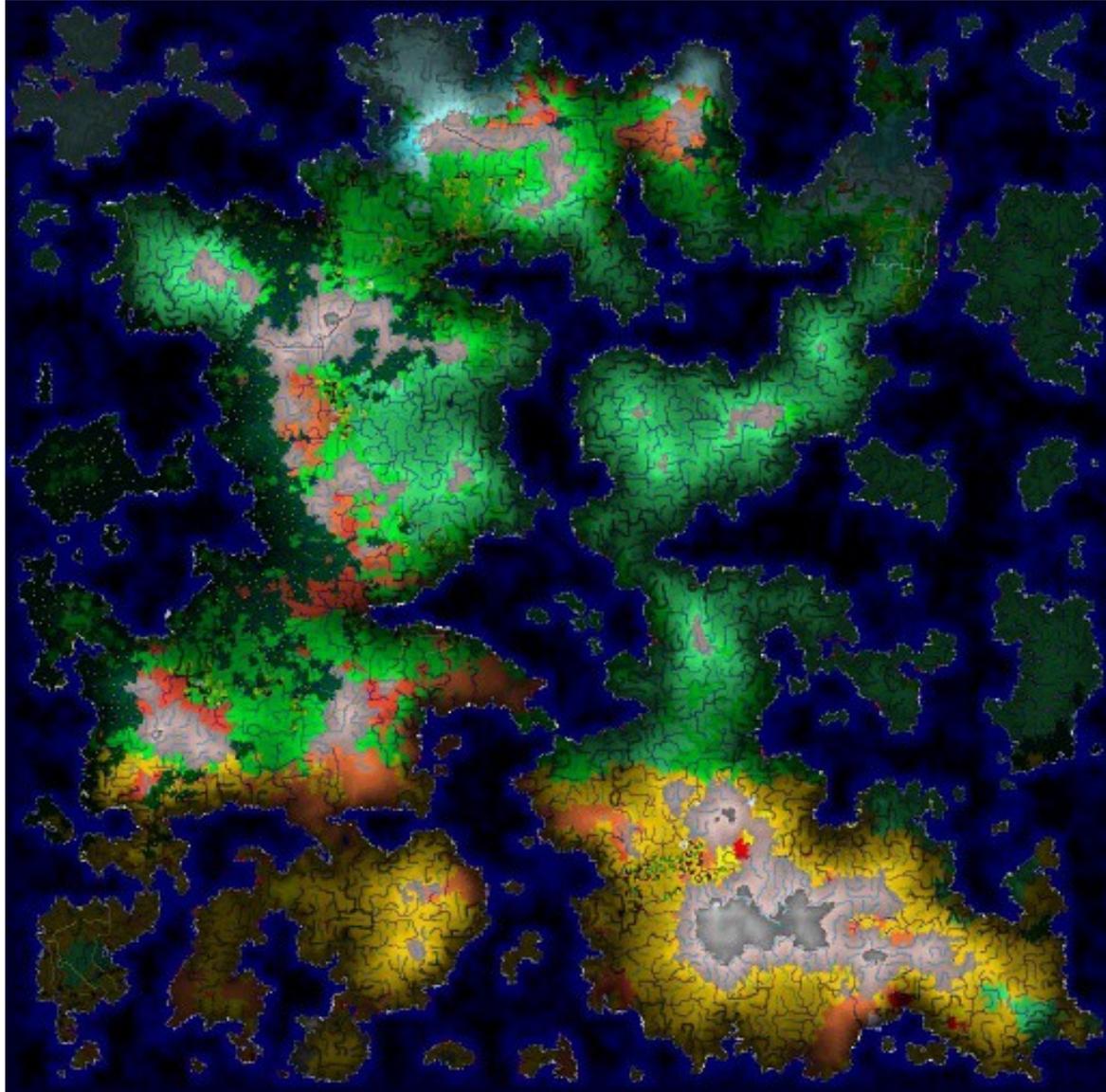
- **F.E.A.R**

- <http://web.cs.wpi.edu/~rich/courses/imgd4100-b11/lectures/L-Planning.pdf>
- http://web.media.mit.edu/~jorkin/gdc2006_orkin_jeff_fear.pdf



Terraforming

232
244

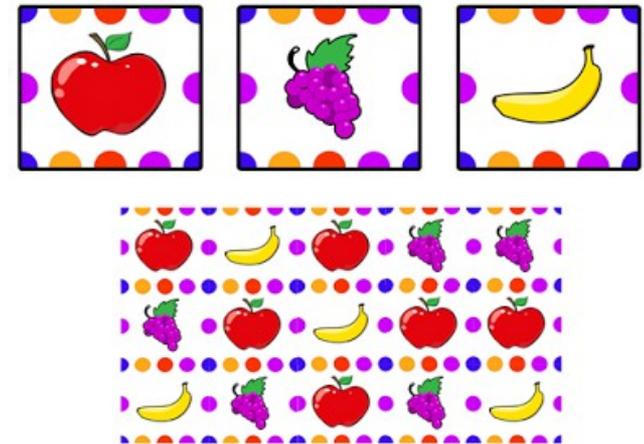
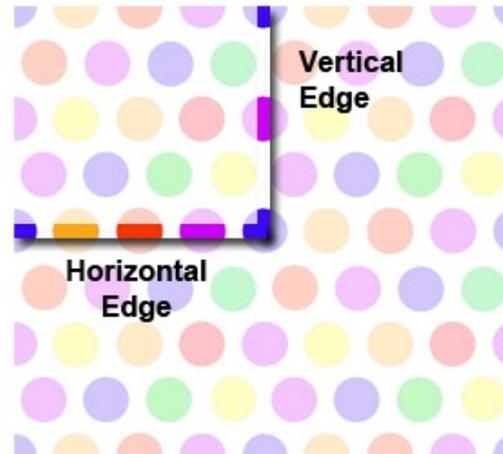


Terraforming

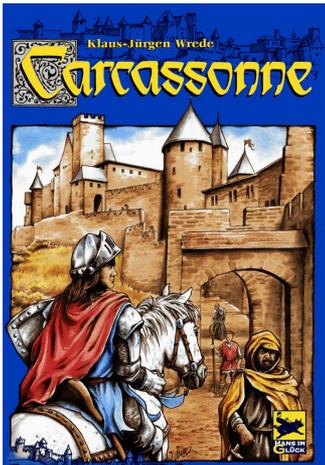
233
244



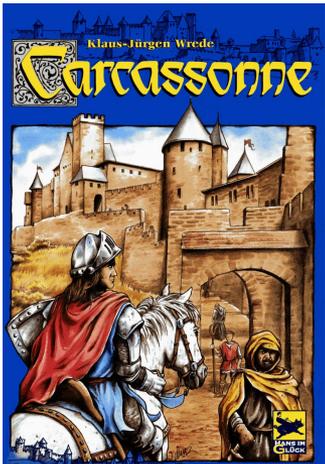
- Construire des reliefs
 - Vastes et réalistes
- Approches
 - Wang tiles



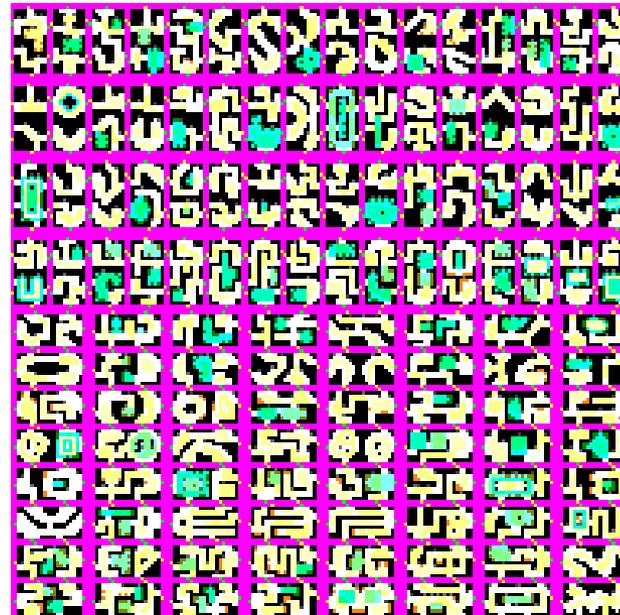
- Construire des reliefs
 - Vastes et réalistes
- Approches
 - Herringbone Tiles

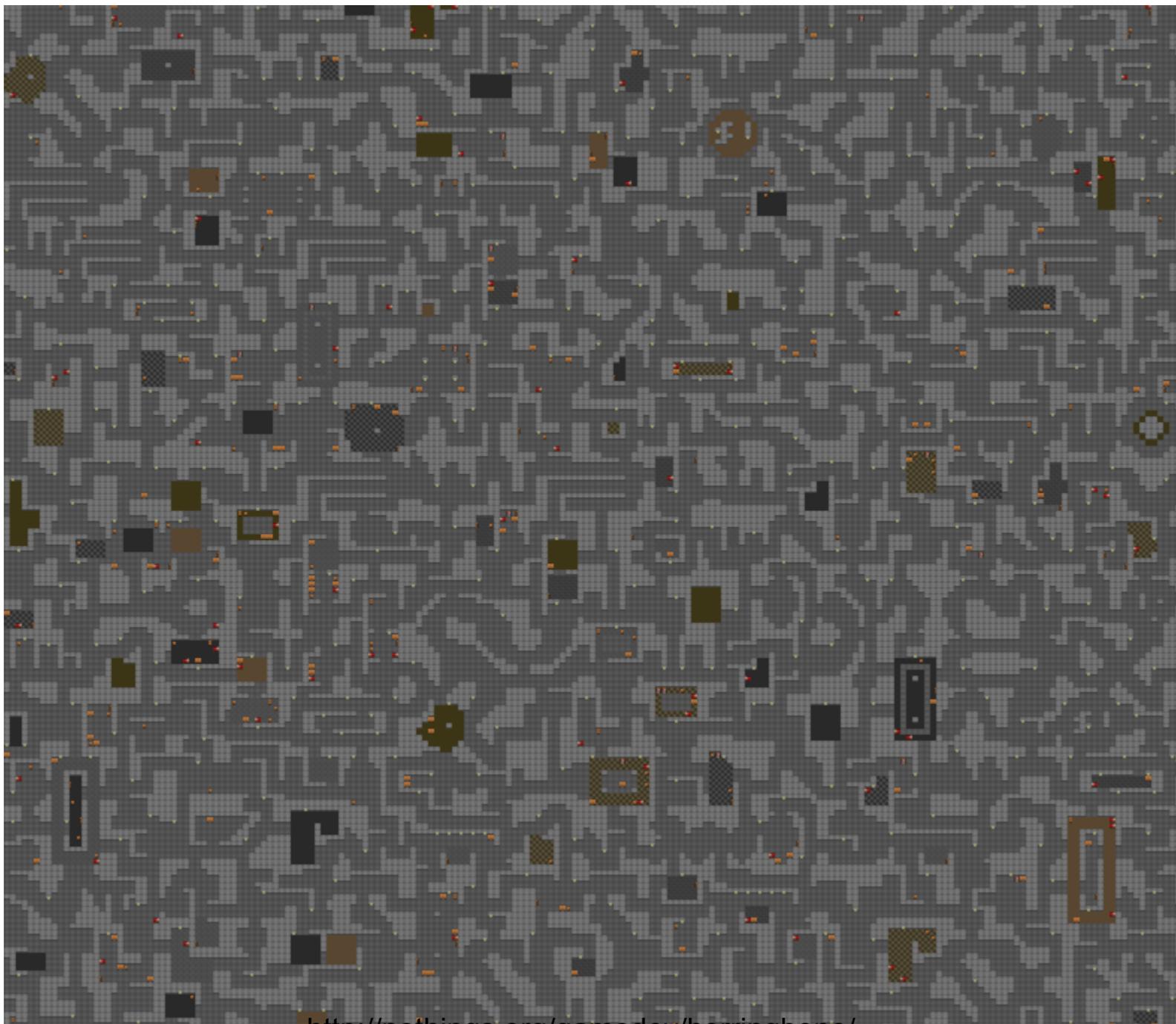


- Construire des reliefs
 - Vastes et réalistes
- Approches
 - Herringbone Tiles



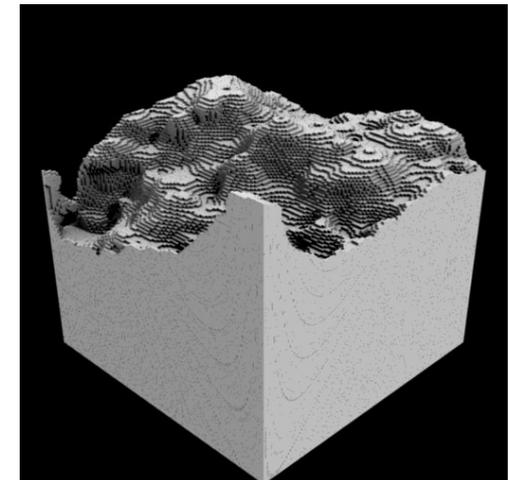
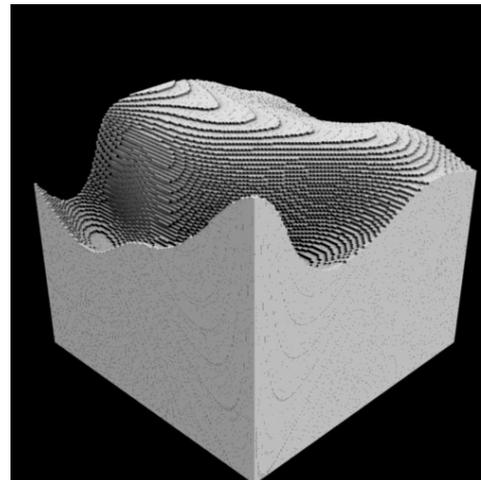
- Construire des reliefs
 - Vastes et réalistes
- Approches
 - Herringbone Tiles





<http://nothings.org/gamedev/herringbone/>

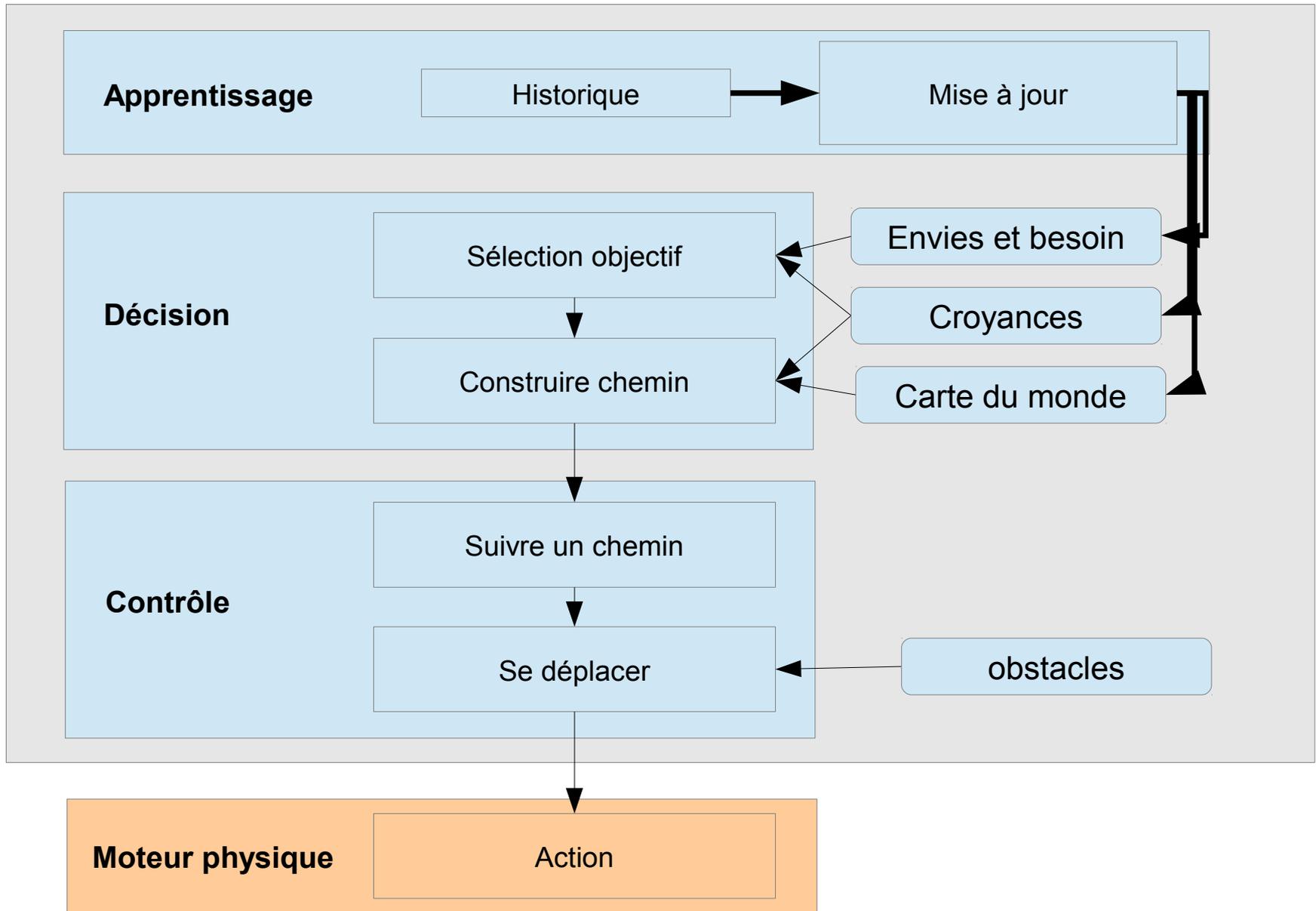
- **Construire des reliefs**
 - Vastes et réalistes
- **Approches**
 - Perlin noise
 - Bruit cohérent
 - Sans discontinuité



- **Ai Game developpement** <http://aigamedev.com>
 - 10 jeux marquants pour leur IA
<http://aigamedev.com/open/review/top-ai-games/>
 - Conférences sur IA dans les jeux
- **Game programming** <http://programming4game.wordpress.com/>
 - Général mais sections sur IA
- **Craig Reynolds** <http://www.red3d.com/cwr/steer/>
 - Inventeur "steering behaviour"

Organisation d'une IA complexe

241
244



Une partie de l'IA

- **Centrée sur le comportement**
- **Mais IA plus que cela**
 - Interaction utilisateur
 - Dialogue naturel, Vision
 - Scenarisation
 - IA metteur en scene du jeu
 - Scenario évolutif à partir de brique de base
 - Création de levels
 - Terraforming, Labyrinthe
 - Emotions et ressenti ?
- **Bien plus encore**

Conclusion

- IA est avant tout pour le joueur
- Elle doit servir quelque chose en terme de gameplay
- Elle doit sembler intelligente
 - Parfois une ia simple est tres efficace
- Ne pas hésiter à tricher
 - depend de l'objectif de l'IA
- Trouver un compromis
 - Entre le réalisme
 - La facilité de mise en oeuvre