INTELLIGENCE ARTIFICIELLE POUR LE JEU VIDEO (GAMEAI)

Rémi Ronfard, Novembre 2014

Cette présentation va s'intéresser aux différentes techniques d'intelligence artificielle utilisées dans les moteurs de jeu. Surtout sur le plan des méthodes génériques d'utilisation courante.

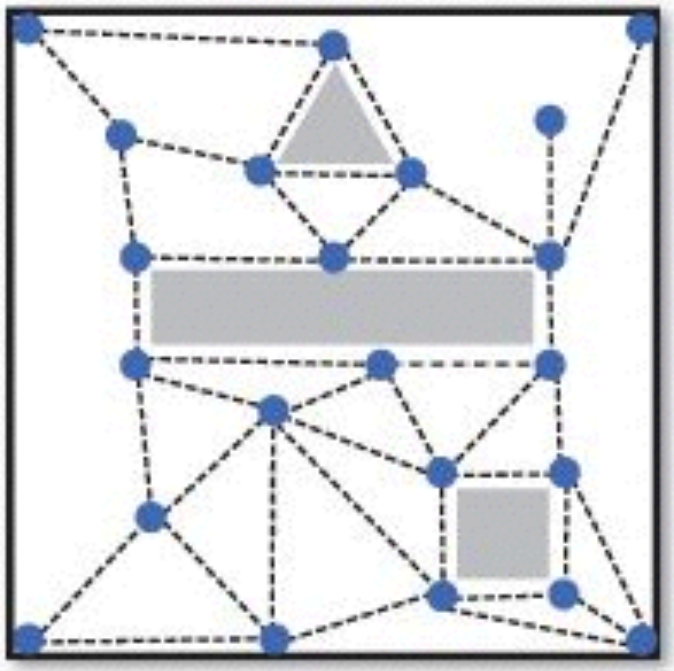
INTELLIGENCE ARTIFICIELLE

SOMMAIRE

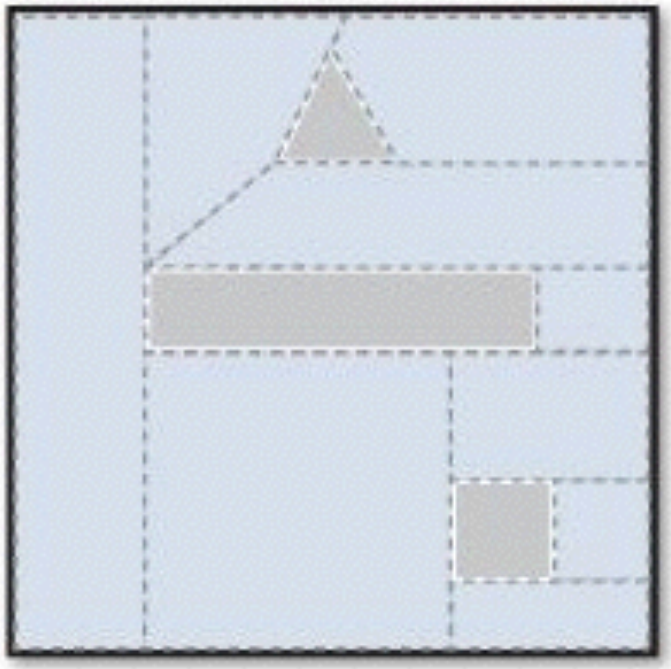
1. Recherche de chemins
2. Machines d'états finis
3. Arbres de comportements
4. Etude de cas : caméras intelligentes
5. Etude de cas : Les SIMS

RECHERCHE DE CHEMINS OPTIMAUX

Path nodes and navigation meshes

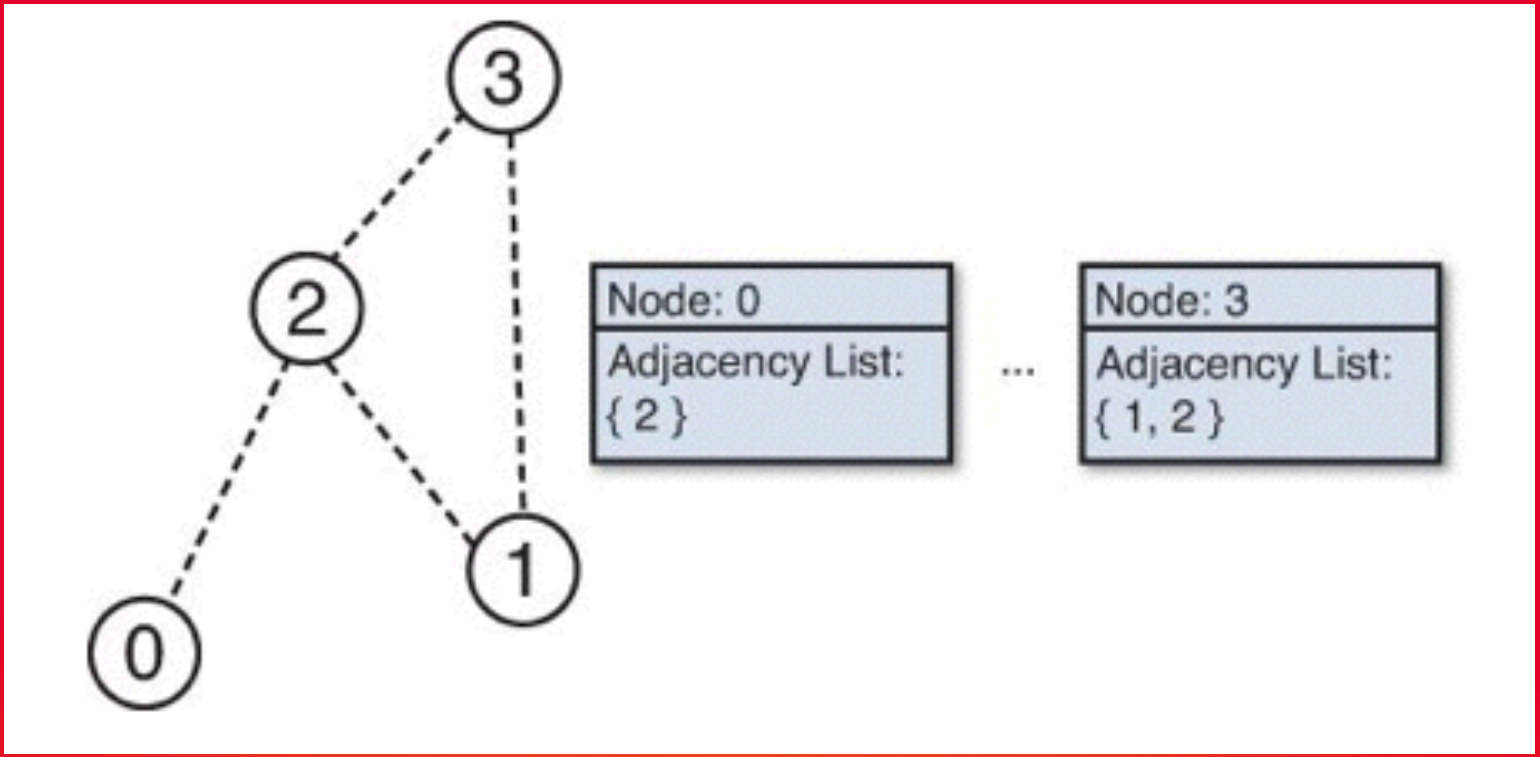


Path nodes
(22 nodes, 41 edges)

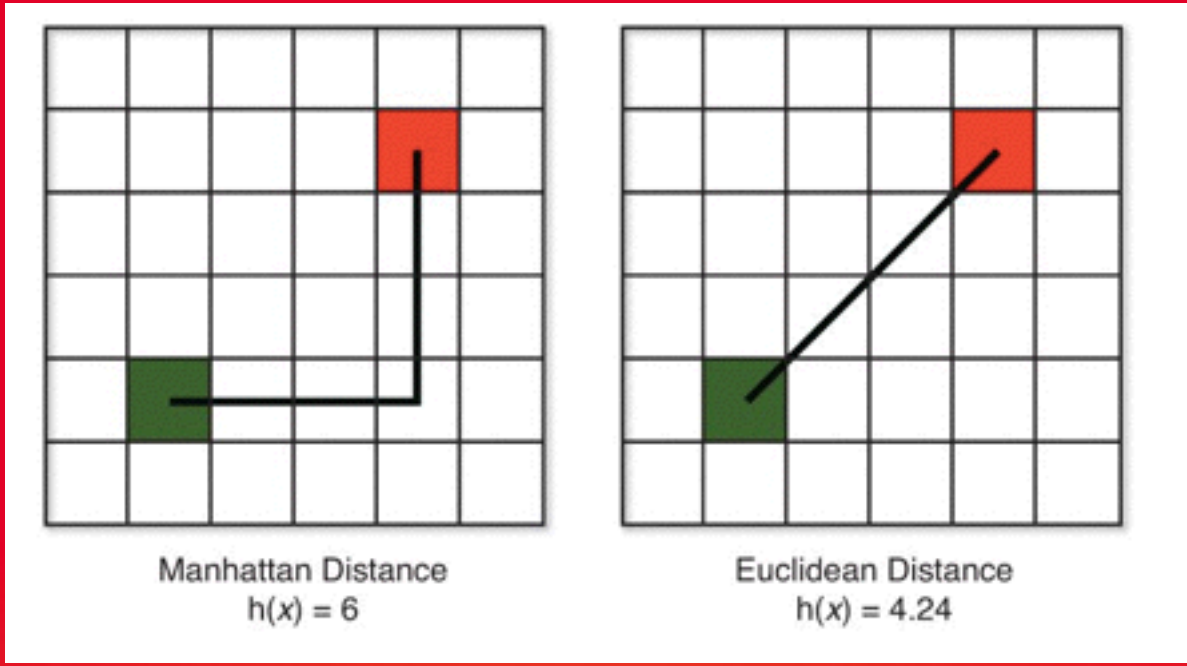


Navigation Mesh
(9 nodes, 12 edges)

Graphe de parcours



Heuristique : distance Manhattan ou Euclidienne



- Longueur de parcours >= heuristique

Arbre de parcours

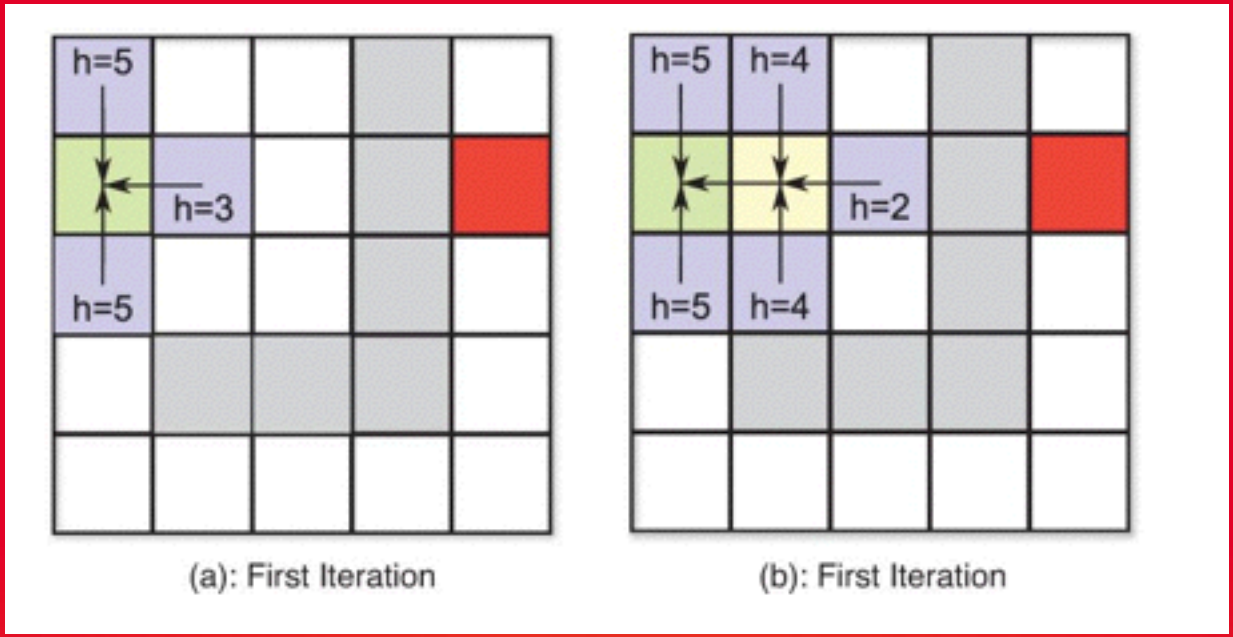
```
struct Node
    Node parent
    float h
end
```

- OpenSet : Noeuds à évaluer, par ordre de distance croissante (file de priorité)
- ClosedSet : Noeuds déjà évalués (arbre binaire)

Algorithme de parcours « best first »

```
do
    foreach Node n adjacent to currentNode
        if closedSet contains n
            continue
        else
            n.parent = currentNode
            if openSet does not contain n
                compute n.h
                add n to openSet
            end
        end
until currentNode == endNode //end main do...until loop
```

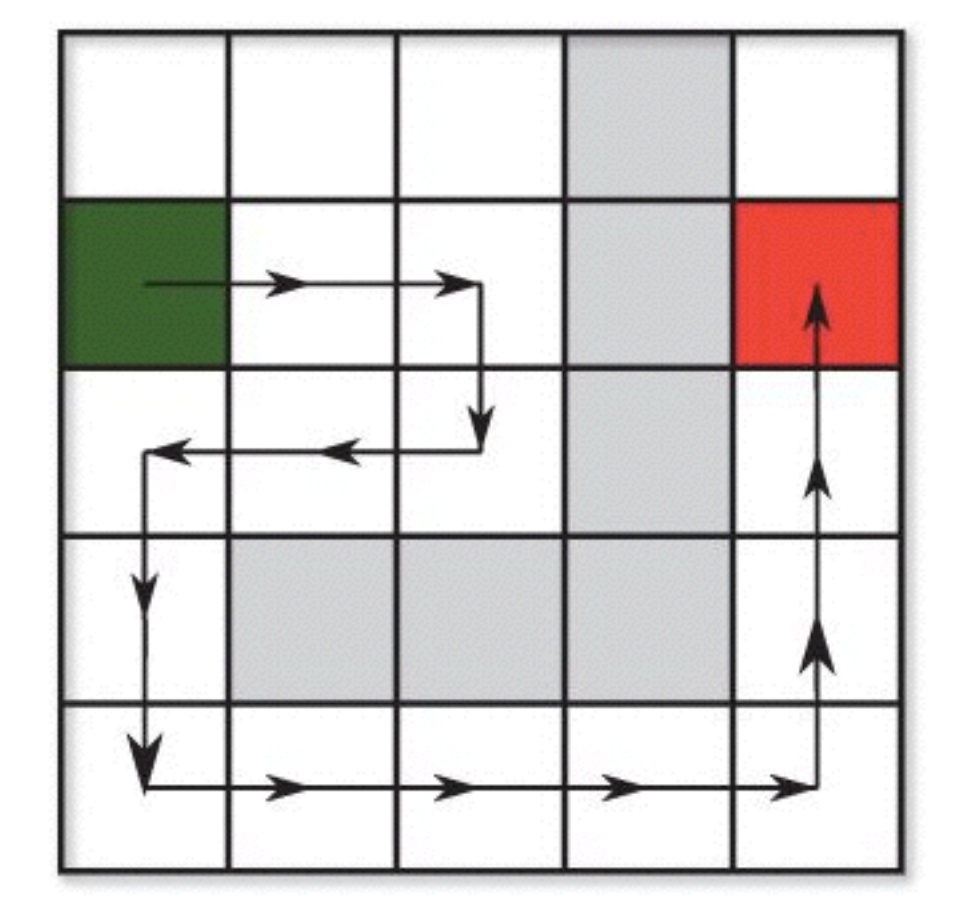
Best-first search



(a): First Iteration

(b): First Iteration

Résultat de la recherche heuristique



A* algorithm

- Meilleure estimation de la distance totale

$$f(x) = g(x) + h(x)$$

- Révision du meilleur chemin

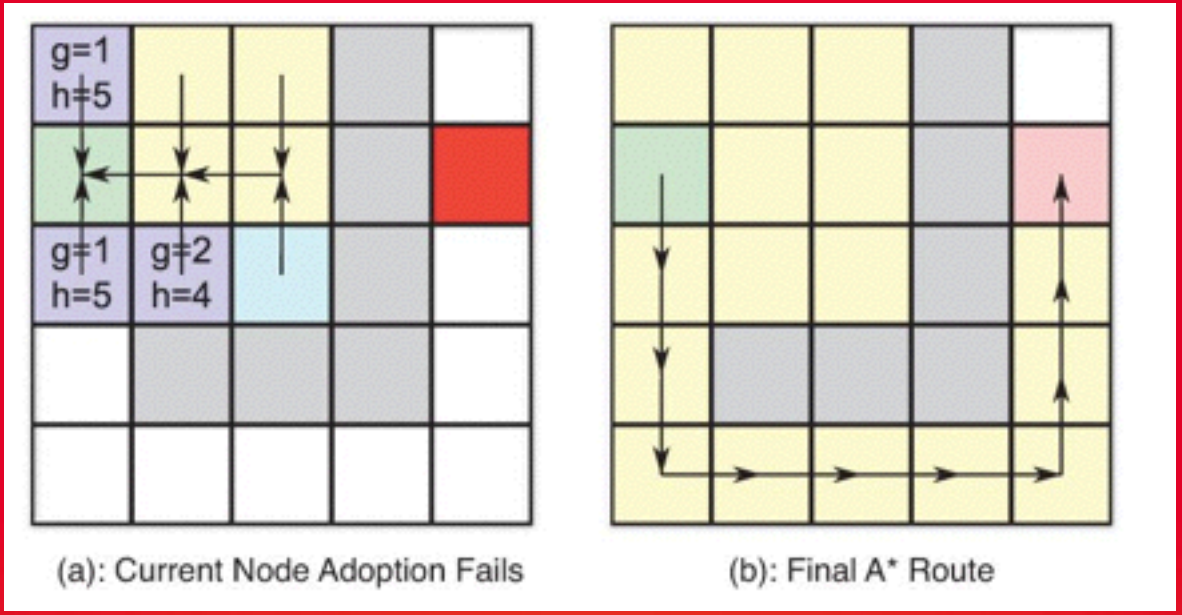
```
struct Node
    Node parent
    float f
    float g
    float h
end
```

A*

```
currentNode = startNode
add currentNode to closedSet
do
   foreach Node n adjacent to currentNode
      if closedSet contains n
         continue
      else if openSet contains n // Check for adoption
         compute new_g // g(x) value for n with currentNode as parent
         if new_g < n.g
            n.parent = currentNode
            n.g = new_g
            n.f = n.g + n.h // n.h for this node will not change
         end
      else
         n.parent = currentNode
         compute n.h
         compute n.g
         n.f = n.g + n.h
         add n to openSet
      end
   loop

   if openSet is empty
      break
   end
   currentNode = Node with lowest f in openSet
   remove currentNode from openSet
   add currentNode to closedSet
until currentNode == endNode
// Path reconstruction from Listing 9.1.
...
```

A*



(a): Current Node Adoption Fails

(b): Final A* Route

Variante sans heuristique : Dijkstra

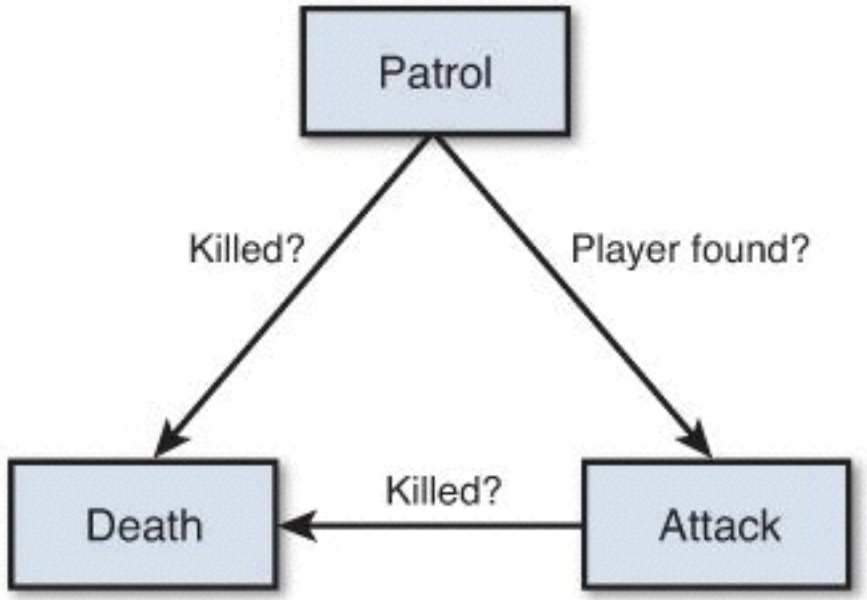
$$f(x) = g(x) + h(x)$$
$$h(x) = 0$$
$$\therefore f(x) = g(x)$$

- Avantage : trouve toutes les solutions
- Inconvénient : parcourt tous les chemins possibles

MACHINES D’ETATS FINIS

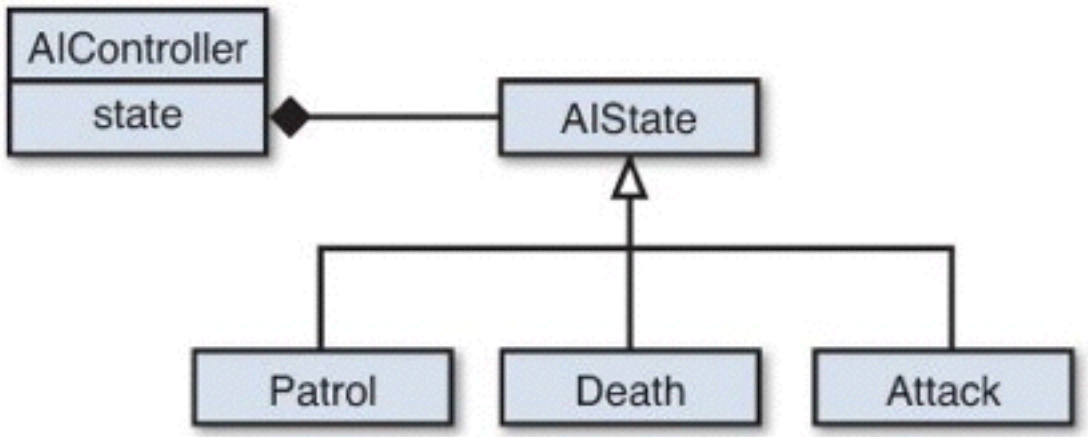
Machines d'états finis

```
function AIController.Update(float deltaTime)
    if state == Patrol
        // Perform Patrol actions
    else if state == Death
        // Perform Death actions
    else if state == Attack
        // Perform Attack actions
    end
end
```



Machine d'états finis

```
class AIState
    AIController parent
    function Update(float deltaTime)
     function Enter()
     function Exit()
end
```



Machine d'états finis

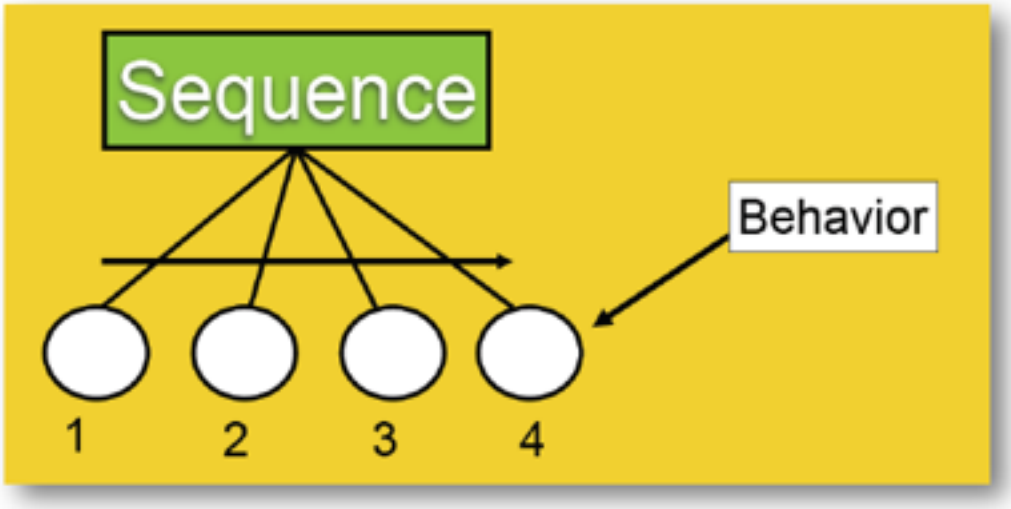
```
class AIController
    AIState state
    function Update(float deltaTime)
    function SetState(AIState newState)
end
```

```
function AIController.SetState(AIState newState)
    state.Exit()
    state = newState
    state.Enter()
end
```

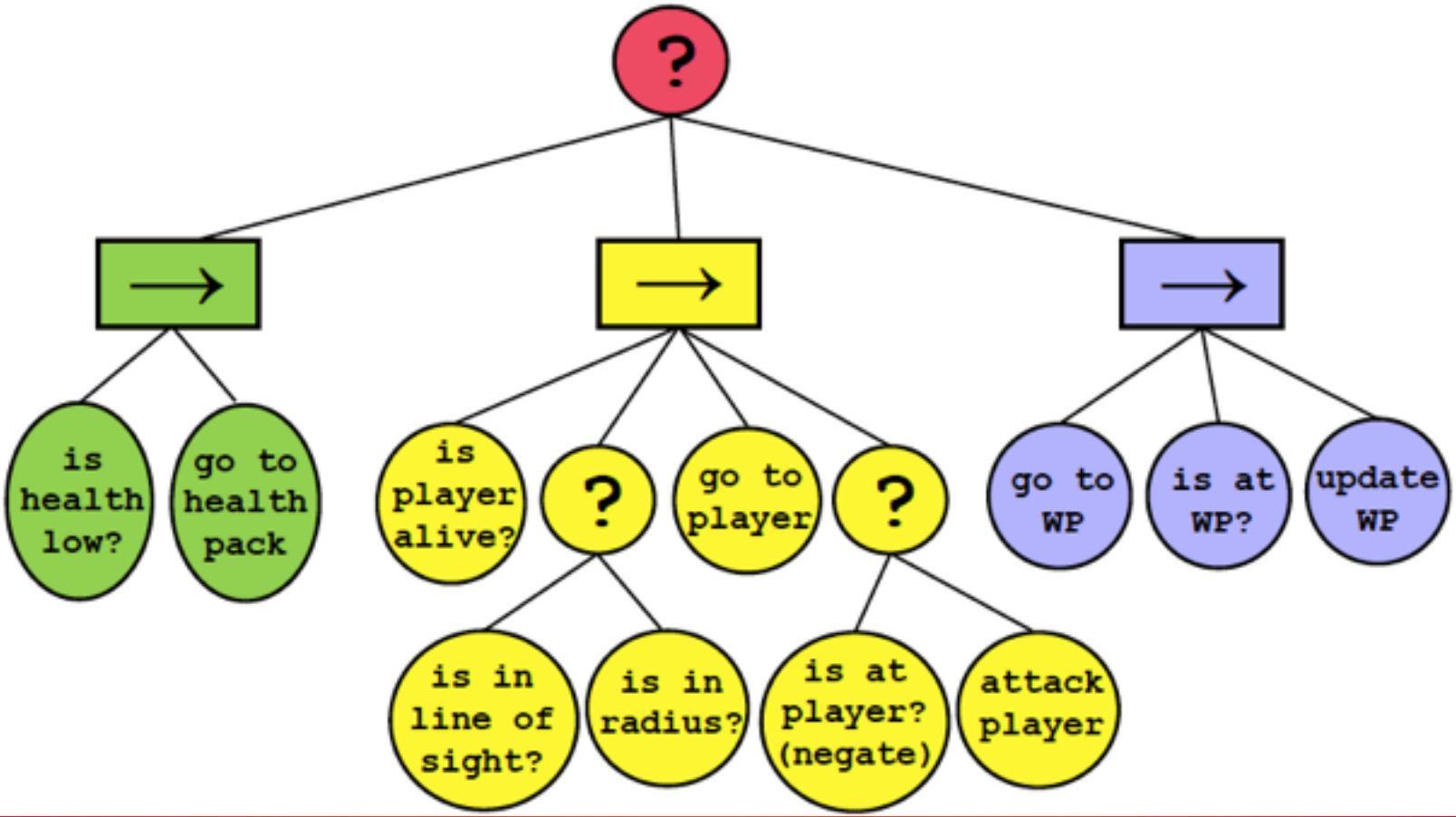
ARBRES DE COMPORTEMENTS

Behavior Tree - Sequence

- Basic functionality:
- Child behavior succeeds → sequence continues
- Child behavior fails → sequence fails



Example Behavior Tree



1ERE ETUDE DE CAS : CAMERAS INTELLIGENTES

Virtual Cinematography

Theory and Practice for Automatic Real-Time Camera Control and Directing

Liwei He

Microsoft Research

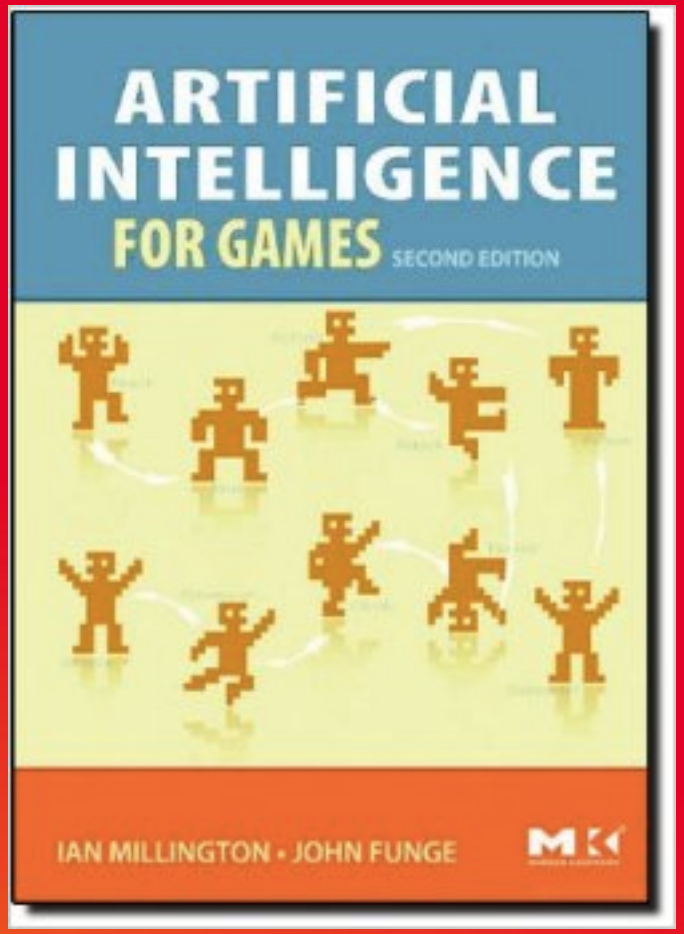
http://research.microsoft.com/users/lhe

2EME ETUDE DE CAS : LES SIMS

Artificial intelligence in the SIMS series (Yoann Bourse)

Références


A PLATFORM-AGNOSTIC APPROACH
Game Programming
ALGORITHMS
and
TECHNIQUES
Sanjay MADHAV


ARTIFICIAL
INTELLIGENCE
FOR GAMES
SECOND EDITION
IAN MILLINGTON • JOHN FUNGE