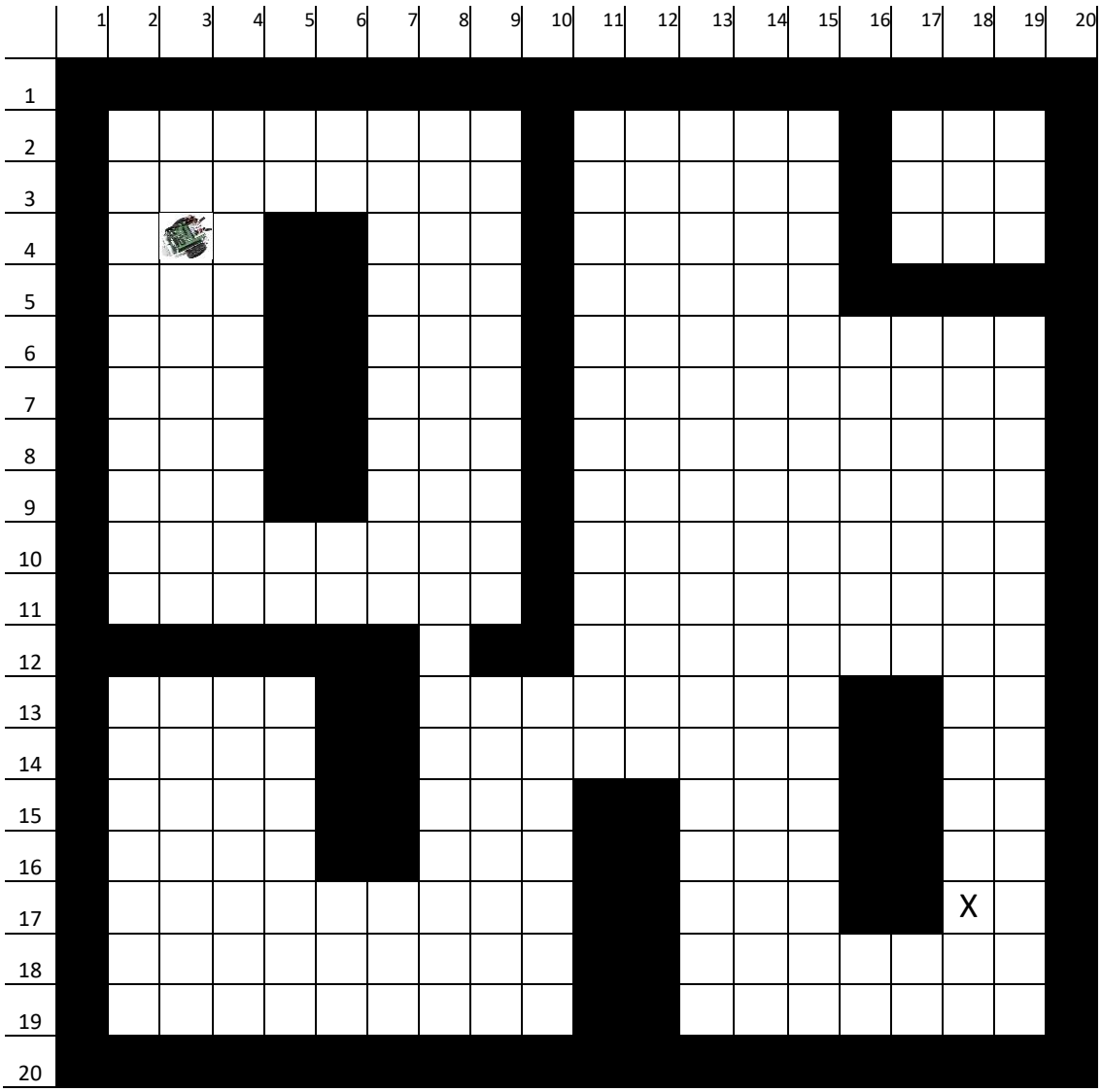
ENSC 2ème année, module d'Intelligence artificielle

Responsable : Jean-Marc Salotti

Projet d'intelligence artificielle

Ce projet est à réaliser par groupe de 3 élèves maximum.

On considère un domaine carré de 20 mètres par 20 mètres représenté par une grille. Un robot de 1 mètre carré se déplace à l'intérieur. Les obstacles sont représentés par des cases marquées en noir.



Ecrire un programme informatique qui exploite l'A* et résout les problèmes suivants :

a) Soit une position initiale I (Xi,Yi) choisie par l'utilisateur (case libre) et une position F (Xf, Yf) également choisie par l'utilisateur, déterminez la suite de cases permettant d'aller de I à F par le chemin le plus court (selon la distance euclidienne). On suppose que le déplacement du robot peut se faire horizontalement, verticalement ou en diagonale. Exploitez au mieux les heuristiques de votre choix pour minimiser l'arbre de recherche tout en étant assuré de trouver la solution optimale. Testez en particulier les cas suivants et indiquez pour chacun les cases de l'arbre d'exploration.
   - a.1) I(3,4) ; F(18,17) (exemple de la figure)
   - a.2) I(3,4) ; F(4,13)
   - a.3) I(3,4) ; F(18,3)
   - a.4) I(18,17) ; F(3,4)

b) Soit un ensemble E de 4 cases par lesquelles le robot doit obligatoirement passer dans un ordre quelconque, puis revenir à sa position initiale. Déterminez la suite de cases permettant de minimiser le trajet complet. Le programme doit trouver le plus court chemin pour toute position initiale du robot et tout ensemble E comportant 4 cases distinctes et accessibles. Testez en particulier les cas suivants et indiquez pour chacun le nombre de nœuds de l'arbre d'exploration :
   b.1) I(3,4) E={ (2,8) ; (2,9) ; (3,9) ; (4,9)}
   b.2) I(3,4) E={ (3,10) ; (8,10) ; (8,3) ; (3,3) }
   b.3) I(3,4) E={ (3,10) ; (3,17) ; (13,4) ; (18,17) }

Annexe :
Vous intégrerez dans votre projet, **sans les modifier**, les 2 fichiers GenericNode.cs et Graphcs.

- GenericNode, classe abstraite, comporte notamment :
  - un double GCost pour mémoriser le coût du chemin qui part du nœud initial et qui arrive à ce nœud (g(n) dans Dijkstra/A*)
  - un double HCost pour mémoriser l'estimation heuristique du coût du chemin restant pour atteindre le nœud final (h(n) dans A*)
  - un double TotalCost pour mémoriser le coût total associé au nœud, en fait la somme des 2 valeurs précédentes, c'est-à-dire g(n) + h(n)
  - un GenericNode ParentNode qui définit l'état parent
- Graph, qui comporte notamment :
  - la liste des ouverts
  - la liste des fermés
  - public List<GenericNode> RechercheSolutionAEtoile(GenericNode N0)

Il est important de noter que RechercheSolutionAEtoile est la seule méthode publique de Graph. L'exploitation de cette classe est donc très contrainte. La liste retournée correspond au chemin solution pour aller du nœud initial au nœud final proposé.

La classe GenericNode est plus complexe. Elle comporte notamment les méthodes **abstraites** suivantes :

```
// Méthodes abstrates, donc à surcharger obligatoirement avec override dans une classe dérivée
    public abstract double GetArcCost(GenericNode N2);
    public abstract bool EndState();
    public abstract List<GenericNode> GetListSucc();
    public abstract void CalculeHCost();
```

Les classes Graph et GenericNode sont complètes, vous ne devez pas les modifier. Un exemple de classe dérivée est proposée pour résoudre le problème du taquin. Cette dernière classe ne doit pas faire partie de votre projet.

Travail à rendre au plus tard le 30 mars 2018 :

Les sources C# et un rapport dans lequel vous expliquerez votre modélisation (quels attributs pour la classe dérivée de GenericNode, notamment) ainsi que les choix et justifications de vos heuristiques, avec des tests et résultats pertinents.

Critères de notation : Réalisation du travail demandé, efficacité et justification des heuristiques retenues, justesse du code, adéquation des commentaires, pertinence des exemples retenus pour les tests, clarté des explications et présentation.