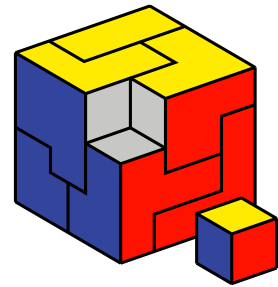


Prologuin 2018



PENGUINS IN BLACK

TOP SECRET//SI//ORCON//NOFORN

Sujet de la finale du Concours National d'Informatique
Samedi 19 mai 2018



Table des matières

1	Contexte	3
1.1	Épisode 1 : Crise	3
1.2	Épisode 2 : Les PiB	4
1.3	Épisode 3 : Les joies de l'administration française	5
2	Instructions de mission	6
2.1	Banquise	6
2.2	Déroulement d'un tour	8
3	Tournois	11
3.1	Tournois intermédiaires	11
3.2	Rendu final	11
4	Considérations techniques	12
5	API	13
6	Notes sur l'utilisation de l'API	20
6.1	C	20
6.2	C++	20
6.3	C#	20
6.4	Haskell	20
6.5	Java	21
6.6	OCaml	21
6.7	PHP	21
6.8	Python	22
6.9	Rust	22

PROLOGIN PRESENTS

PIB

PENGUINS IN BLACK



2018 EDITION

1 Contexte

1.1 Épisode 1 : Crise

Maubeuge, Nord, jeudi 17 mai 2018

Cela fait maintenant trois ans que vous êtes employé dans le bureau des IeN¹. Affecté depuis la moitié de ce temps dans les bureaux grisâtres du Nord Pas-de-Calais², vous pensiez que votre carrière se limiterait à servir le café à des quadragénaires pédants.

Pourtant, ce matin, vous notez une agitation certaine dans les locaux du 4, Rue du Trieu Mouton, Maubeuge, 59600, FRANCE. Effectivement, alors que vous naviguez entre les bureaux Ikea pour déposer un latte - tiède, avec deux sucres - sur la grosse patte gélatineuse de Beauté, l'alien aux télécommunications, vous surprenez des cris provenant directement du bureau du Directeur, D. Les têtes se dressent, curieuses, d'un peu partout de l'open space. Malgré votre professionnalisme à toute épreuve, vous ne pouvez vous empêcher de jeter un œil, vous aussi. À travers les stores entrouverts du bureau, vous voyez le directeur et J³, un de vos collègues affectés au terrain⁴, engagés dans une discussion animée dont vous entendez quelques bribes : apparemment, il se passe quelque chose de très grave impliquant le Pôle Sud, des aliens, et le menu de la cantine.

Le ton monte et, inquiet, vous fixez franchement les deux hommes - cela fait maintenant trop longtemps que le cuisinier essaye de tous vous empoisonner - quand ceux-ci se détournent brusquement vers vous. Ils vous regardent, vous les regardez, puis vous vous voyez soudainement convoqué par un signe sec du Directeur. Avec un courage que vos collègues saluent en murmurant, vous entrez.

- Un petit café ?
- Ça vous dirait de diriger une équipe ?
- Ah ben oui, pourquoi pas.

Silence.

- Bon, j'ai une idée.

Et sur cette déclaration étonnante, J plonge les mains dans la poubelle et en extirpe une affiche recouverte de manchots⁵, où s'étale en grand le mot suivant : *Prologin*.

1. *Individus en Noir*TM.

2. La délocalisation affecte tous les secteurs.

3. De son vrai nom Joseph Marchand.

4. Promotion dont vous rêvez chaque soir : ils ont beaucoup plus de RTT.

5. Et non pas de pingouins.

1.2 Épisode 2 : Les PiB

Kremlin-Bicêtre, Île-de-France, samedi 19 mai 2018

Voilà comment vous vous êtes retrouvés dans le bureau des PiB⁶, sous-équipe du sous-sous-secteur du Kremlin-Bicêtre.

Le Président, B, vous explique la situation.

- Le Département Administratif des Déplacements Aliens⁷ est formel : une large invasion alien se prépare.
- Mais pourquoi ?
- On ne le sait pas vraiment.⁸

B balaye toutes vos protestations dubitatives d'un mouvement ample de la main.

- En tout cas, une chose est certaine : un groupe d'aliens débarquera en repérage au Pôle Sud. Il faut absolument en capturer avant qu'ils ne repartent pour déterminer quel est leur plan, et empêcher l'invasion. C'est pour cela que nous avons décidé de mettre nos meilleurs agents dessus : les manchots.
- Les quoi ?
- Les manchots.
- Pardon ?

B soupire et tire de sa poche une petite photographie froissée. Vous discernez dessus une silhouette replète, ronde et toute douce, ornée d'une paire de lunettes de soleil sur son adorable petit bec.

- Aaaaaw.
- Détrompez-vous, ils sont redoutables. Leur vitesse n'a aucun égal sur la glace.
- Bon, dans ce cas, pourquoi on est là ?
- Les manchots ont un problème majeur, explique B en ajustant ses lunettes. Ils sont très, très stupides. Ils auront besoin de quelqu'un pour leur expliquer quoi faire.

Et d'un geste dramatique, B montre de la main les recrues pullulant dans la cour.

6. *Prologin in Black*TM

7. Le *DADA* est très à cheval sur les acronymes.

8. D'après les rumeurs, une sombre histoire de clafoutis.

1.3 Épisode 3 : Les joies de l'administration française

Pôle Sud, samedi 19 mai 2018

Vous sortez vos jumelles et les pointez droit vers la banquise jusqu'à retrouver les petits points noirs : les quatre manchots qui composent votre équipe. Cependant, vous déchantez vite. Un peu plus haut, vous repérez quatre petits points noirs supplémentaires : les aliens, déjà!? Non. D'autres pingouins⁹. Mais d'où?

Irrité, vous appelez aussitôt votre supérieur hiérarchique direct.

- À quoi ça rime? Pourquoi on est deux équipes au même endroit?
- Euuuuuh... Attendez.

Petite musique d'ascenseur.

- Bah en fait, euh, ben, il y a eu une erreur.
- Une erreur?
- On a peut-être, ou peut-être pas, envoyé deux équipes sur la même partie de la banquise...¹⁰

Vous raccrochez, et réfléchissez. Que devriez-vous faire? Laisser l'autre équipe s'en occuper, quitte à perdre votre travail? Ou profiter de cette compétition pour briller aux yeux des PiB? Le choix est vôtre.¹¹

9. Pardon. D'autres *manchots*.

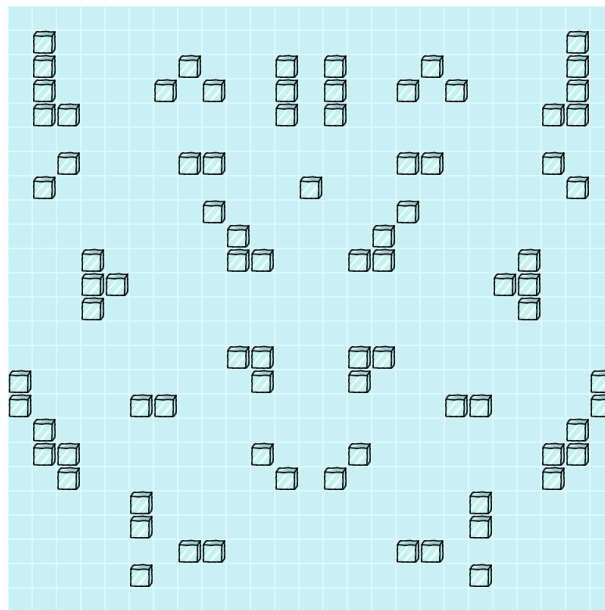
10. Cf. le titre

11. En fait, si vous êtes toujours là, vous n'avez plus vraiment le choix.

2 Instructions de mission

2.1 Banqueise

Dans le cadre de votre mission, vous serez envoyé au Pôle Sud, sur une partie de la banqueise où nos services de renseignements ont indiqué que les premiers repérages aliens auront lieu. La banqueise est représentée par une grille carrée de 25 cases de côté.



Une partie de la banqueise.

Une position sur la banqueise est notée à l'aide d'un couple (*ligne*, *colonne*), où $0 \leq \text{ligne}, \text{colonne} < 25$.

2.1.1 Cases

Chaque case de la banqueise est soit libre, soit un mur de glace. Les murs sont des obstacles et bloquent tout déplacement sur la case.

Une case libre peut contenir un alien ainsi qu'un agent. Un agent peut être sur la même case qu'un alien, en revanche il est impossible d'avoir plusieurs agents ou plusieurs aliens sur une même case.

2.1.2 Agents

Les deux recrues PiB ont à leur disposition quatre agents, numérotés de 0 à 3. Ces derniers sont considérés comme des obstacles, et bloquent donc tout déplacement sur la case.



Un agent PiB.

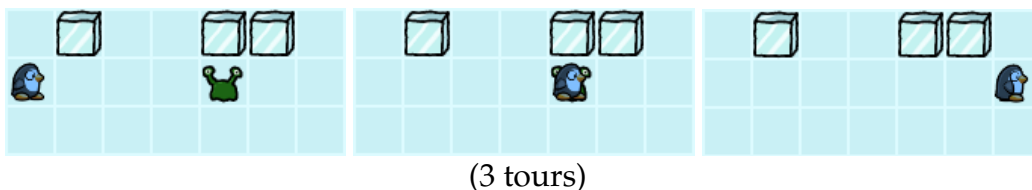
2.1.3 Aliens

Des aliens débarqueront sur la banquise à des positions précises de la carte, pendant un certain nombre de tours afin d'accomplir leur mission de reconnaissance, avant de repartir sur leur planète d'origine. De plus, les aliens n'envahissent jamais plusieurs fois le même endroit sur la banquise.



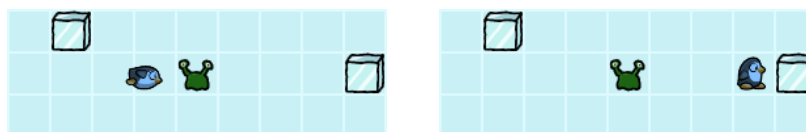
Un alien menaçant.

Pour capturer un alien, un agent doit être sur la case pendant au moins 3 tours. L'alien capturé disparaît de la banquise, et des échantillons d'analyse sont envoyés instantanément au QG des PiB, l'agent peut donc continuer sa mission. Si l'agent quitte la case, ne serait-ce qu'un instant (en se déplaçant ou alors en étant poussé par un agent), la capture devra reprendre de **zéro**.



Un agent qui capture un alien, puis continue sa mission.

Les aliens ne sont pas assez habitués à la glace pour se déplacer sur la banquise. Ils se contenteront donc pour leur mission de repérage de rester fixes par rapport à leurs lieux d'invasion. En revanche, les aliens ne sont pas des obstacles : faisant des efforts admirables pour éviter les agents - contrairement aux murs, qui sont davantage récalcitrants - ils se contorsionneront et esquiveront de leur mieux : ils ne bloquent donc pas le déplacement des agents.



Les aliens ne sont pas des obstacles.

2.2 Déroulement d'un tour

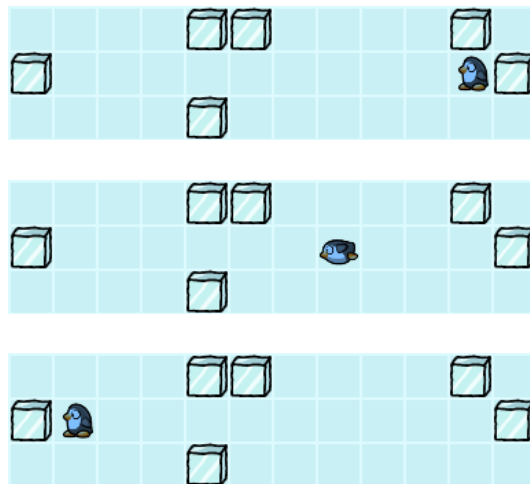
Il y a 100 tours par partie, numérotés de 0 à 99. Pendant un tour les recrues jouent alternativement. Les invasions ou départs d'aliens ont toujours lieu en début de tour avant les actions des joueurs. Par exemple, si un alien envahit la banquise au tour 10, pour une durée de 3 tours, alors il sera présent aux tours 10, 11 et 12 et repartira au tout début du tour 13. En revanche, la capture des aliens se fait toujours à la fin du tour, lorsque les deux recrues ont fini de jouer.

Tous les agents se voient attribuer 8 points d'action au début de chaque tour. Ces points ne sont utilisables que durant ce tour et sont spécifiques à un agent (il est donc impossible de transférer des points d'un agent à un autre). Les points vous permettent d'effectuer les actions ci-dessous.

2.2.1 Actions

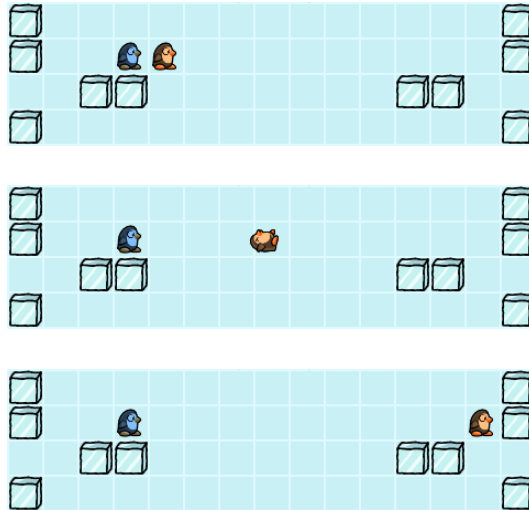
Déplacer Vous pouvez déplacer un agent vers une case libre adjacente dans la direction de votre choix (nord, sud, est, ouest). Cette action coûte 1 point d'action à l'agent.

Glisser Un agent peut s'élancer fougueusement sur la banquise, directement sur le ventre, dans une certaine direction, ce qui le propulse jusqu'à ce qu'il heurte un obstacle (un autre agent ou un mur). L'action coûte 3 points d'action.



Un agent qui glisse vers l'ouest.

Pousser Il est possible de pousser un autre agent (allié ou ennemi) si ce dernier est sur une case adjacente à l'un de vos propres agents. Le pousser dans une direction le fait glisser jusqu'à ce qu'il rencontre un obstacle. Pousser un agent coûte 5 points d'action.



L'agent bleu pousse l'agent rouge.

À nouveau, les aliens font tout leur possible pour esquiver les agents, vous ne pouvez donc pas les pousser (ce n'est pas pour rien qu'il faut 3 tours pour les capturer!).

Débug Pour vous permettre de déboguer votre intelligence artificielle, il est possible de placer des drapeaux de débog de trois couleurs différentes sur la carte que vous pourrez ainsi voir dans l'interface de la simulation. Cette action ne coûte aucun point d'action.



Drapeaux de débog.

2.2.2 Score

Chaque alien capturé vous rapporte un certain nombre de points en fonction de l'alien, selon son espèce, le danger brut qu'il représente, et ses opinions politiques. La recrue ayant accumulé le plus de points à la fin de la partie rejoindra les rangs des *Prologin in Black*TM pour lutter contre les invasions intergalactiques.

2.2.3 Format de la carte

La carte de la banquise est représentée dans un fichier texte qui suit le format suivant :

```
banquise ASCII
positions de depart agents joueur 1
positions de depart agents joueur 2
description des aliens
```

La représentation ASCII de la banquise est constituée de « . » pour une case libre et « X » pour un mur.

Pour chaque joueur, quatre lignes, une par agent, indiquent la position de départ d'un agent sous la forme ligne colonne.

La description des aliens commence par un nombre sur une seule ligne indiquant le nombre d'aliens qui envahiront la banquise durant la partie. Chaque ligne précise ensuite les caractéristiques d'un alien : position_ligne position_colonne points_capture tour_invasion duree_invasion

3 Tournois

3.1 Tournois intermédiaires

Afin de vous aider à perfectionner vos algorithmes, des tournois intermédiaires vous seront proposés toutes les six heures environ. Ces matchs n'ont absolument aucune influence sur le classement final, mais sont néanmoins à prendre au sérieux, car ils vous permettront de vous situer par rapport aux autres joueurs, de connaître vos ennemis, vos points forts et vos faiblesses, et vous donneront des pistes pour vous améliorer pendant la finale.

Les tournois se dérouleront aux horaires suivants :

- Samedi 15 h 42 (tournoi de test)
- Samedi 17 h 42
- Samedi 23 h 42
- Dimanche 5 h 42
- Dimanche 11 h 42
- Dimanche 17 h 42
- **Lundi 00 h 42 (rendu final)**

À chacun des horaires indiqués ci-dessous, nous prendrons le dernier champion que chaque candidat aura envoyé sur le site de soumission pour le faire participer au tournoi, et nous vous donnerons les résultats ainsi que votre progression dès que les tournois se seront terminés, avec un récapitulatif de votre progression globale.

Les tournois seront exécutés sur des cartes officielles de notre choix, qui seront potentiellement amenées à changer au fur et à mesure.

3.2 Rendu final

Le rendu final est le seul rendu qui comptera pour le classement. Les mêmes règles s'appliquent : le dernier champion soumis à l'heure du début du tournoi sera le champion utilisé pour le tournoi final.

Lors du tournoi final, plusieurs cartes seront ajoutées. Celles-ci resteront inconnues de tous les joueurs jusqu'à la fin du concours, afin de mesurer l'adaptabilité de vos algorithmes à des situations inconnues.

Pour le rendu final, nous vous demandons de rajouter des commentaires qui résument le fonctionnement des différents blocs logiques de votre code, ainsi qu'un **commentaire global en haut de votre fichier principal** qui détaille votre stratégie ainsi que les différents algorithmes que vous avez employés pour l'implémenter.

4 Considérations techniques

Vous disposez d'une seconde (temps réel!) à chaque fois qu'une de vos fonctions est appelée pour rendre la main. Passé ce délai, votre programme est tué, le match continue sans vous et vos fonctions ne sont plus appelées. Il n'est pas possible de revenir en jeu tout simplement parce qu'il n'y a aucun moyen de rétablir l'état des environnements des langages après une interruption. Les limites de mémoire sont faites avec des cgroups, ce qui fait que l'allocation échouera si vous essayez de dépasser la limite qui vous est accordée. Cette limite compte aussi la taille de la pile.

D'autres limitations sont appliquées :

- le système de fichiers est entièrement en lecture seule;
- seuls /usr, /var et /tmp sont montés;
- vous n'avez pas le droit d'utiliser des processus en parallèle;
- la mémoire est limitée à 500 Mio;
- la taille totale de votre output ne doit pas dépasser 256 Kio (elle sera tronquée à partir de cette limite);
- le temps d'exécution total du processus est limité à 300 secondes de temps réel;
- chaque appel de fonction est limité à une seconde de temps réel plus 500 millisecondes de marge pour prendre en compte le surcoût de sérialisation/désérialisation des valeurs depuis et vers les langages cibles.

5 API

Constante : TAILLE_BANQUISE

Valeur : 25

Description : Taille de la banquise (longueur et largeur).

Constante : NB_TOURS

Valeur : 100

Description : Nombre de tours à jouer avant la fin de la partie.

Constante : NB_POINTS_ACTION

Valeur : 8

Description : Nombre de points d'action par tour par agent.

Constante : COUT_DEPLACEMENT

Valeur : 1

Description : Nombre de point d'action que coûte un déplacement.

Constante : COUT_GLISSADE

Valeur : 3

Description : Nombre de points d'action que coûte une glissade.

Constante : COUT_POUSSER

Valeur : 5

Description : Nombre de points d'action nécessaires pour pousser un agent.

Constante : NB_AGENTS

Valeur : 4

Description : Nombre d'agents par joueur.

Constante : NB_TOURS_CAPTURE

Valeur : 3

Description : Nombre de tours nécessaires pour capturer un alien.

• case_type

Description : Types de cases

Valeurs : *LIBRE* : Case libre
MUR : Mur
ERREUR : Erreur

• direction

Description : Points cardinaux

Valeurs : *NORD* : Direction : nord
EST : Direction : est
SUD : Direction : sud
OUEST : Direction : ouest

• erreur

Description : Erreurs possibles

Valeurs : *OK* : L'action s'est effectuée avec succès.
PA_INSUFFISANTS : Votre agent ne possède pas assez de points d'action pour réaliser cette action.
POSITION_INVALIDE : La position spécifiée n'est pas sur la banquise.
OBSTACLE_MUR : La position spécifiée est un mur.
OBSTACLE_AGENT : La position spécifiée est un agent.
DEPLACEMENT_HORS_LIMITES : Ce déplacement fait sortir un agent des limites de la banquise.
DIRECTION_INVALIDE : La direction spécifiée n'existe pas.
ID_AGENT_INVALIDE : L'agent spécifié n'existe pas.
RIEN_A_POUSSER : Aucun agent à pousser dans la direction indiquée.
DRAPEAU_INVALIDE : Le drapeau spécifié n'existe pas.

• action_type

Description : Types d'actions

Valeurs : *ACTION_DEPLACER* : Action "déplacer"
ACTION_GLISSER : Action "glisser"
ACTION_POUSSER : Action "pousser"

• debug_drapeau

Description : Types de drapeaux de debug

Valeurs : *AUCUN_DRAPEAU* : Aucun drapeau, enlève le drapeau présent
DRAPEAU_BLEU : Drapeau bleu
DRAPEAU_VERT : Drapeau vert
DRAPEAU_ROUGE : Drapeau rouge

- position

```
struct position {  
    int ligne;  
    int colonne;  
};
```

Description : Position sur la banquise, donnée par deux coordonnées.

Champs : *ligne* : Coordonnée : ligne
colonne : Coordonnée : colonne

- alien_info

```
struct alien_info {  
    position pos;  
    int points_capture;  
    int tour_invasion;  
    int duree_invasion;  
    int capture_en_cours;  
};
```

Description : Alien à capturer durant la mission.

Champs : *pos* : Position de l'alien
points_capture : Nombre de points obtenus pour la capture de cet alien
tour_invasion : Tour où l'alien débarque sur la banquise
duree_invasion : Nombre de tours où l'alien reste sur la banquise
capture_en_cours : Nombre de tours de la capture en cours, l'alien est considéré comme capturé si la variable atteint NB_TOURS_CAPTURE

- action_hist

```
struct action_hist {  
    action_type atype;  
    int id_agent;  
    direction dir;  
};
```

Description : Action de déplacement représentée dans l'historique.

Champs : *atype* : Type de l'action
id_agent : Numéro de l'agent concerné par l'action
dir : Direction visée par l'agent durant le déplacement

- deplacer

erreur deplacer(int id_agent, direction dir)

Description : Déplace l'agent "id_agent" d'une case dans la direction choisie.

Parametres : *id_agent* : Numéro de l'agent
dir : Direction visée

- glisser

erreur glisser(int id_agent, direction dir)

Description : Propulse l'agent "id_agent" dans la direction choisie jusqu'à ce qu'il heurte un obstacle, c'est-à-dire soit un mur soit un autre agent.

Parametres : *id_agent* : Numéro de l'agent
dir : Direction visée

- pousser

erreur pousser(int id_agent, direction dir)

Description : L'agent "id_agent" pousse tout autre agent se trouvant sur la case adjacente dans la direction indiquée. Ce dernier est propulsé jusqu'à ce qu'il rencontre un obstacle, c'est-à-dire soit un mur soit un autre agent.

Parametres : *id_agent* : Numéro de l'agent
dir : Direction visée

- debug_afficher_drapeau

erreur debug_afficher_drapeau(position pos, debug_drapeau drapeau)

Description : Affiche le drapeau spécifié sur la case indiquée.

Parametres : *pos* : Case choisie
drapeau : Drapeau à afficher sur la case

- points_action_agent

int points_action_agent(int id_agent)

Description : Renvoie le nombre de points d'action de l'agent "id_agent" restants pour le tour. Si le numéro d'agent est invalide, la fonction renvoie -1.

Parametres : *id_agent* : Numéro de l'agent

- chemin

direction array chemin(position pos1, position pos2)

Description : Renvoie le plus court chemin entre deux positions de la banque sous la forme d'une suite de direction à emprunter. Ce chemin ne contient pas de glissade, uniquement des déplacements simples. Si la position est invalide ou que le chemin n'existe pas, le chemin renvoyé est vide.

Parametres : *pos1* : Position de départ
pos2 : Position d'arrivée

- type_case

case_type type_case(position pos)

Description : Renvoie le type d'une case donnée.

Parametres : *pos* : Case choisie

- agent_sur_case

int agent_sur_case(position pos)

Description : Renvoie le numéro du joueur à qui appartient l'agent sur la case indiquée. Renvoie -1 s'il n'y a pas d'agent ou si la position est invalide.

Parametres : *pos* : Case choisie

- alien_sur_case

bool alien_sur_case(position pos)

Description : Indique si un alien se trouve sur une case donnée. Renvoie vrai si l'alien est en train d'envahir la banque et qu'il n'a pas encore été capturé. Renvoie faux autrement, ou si la position est invalide.

Parametres : *pos* : Case choisie

- position_agent

position position_agent(int id_joueur, int id_agent)

Description : Indique la position de l'agent sur la banque désigné par le numéro "id_agent" appartenant au joueur "id_joueur". Si la description de l'agent est incorrecte, la position (-1, -1) est renvoyée.

Parametres : *id_joueur* : Numéro du joueur
id_agent : Numéro de l'agent

- info_alien

alien_info info_alien(position pos)

Description : Renvoie la description d'un alien en fonction d'une position donnée. Si l'alien n'est pas présent sur la carte, ou si la position est invalide, tous les membres de la structure "alien_info" renvoyée sont initialisés à -1.

Parametres : *pos* : Case choisie

- liste_aliens

alien_info array liste_aliens()

Description : Renvoie la liste de tous les aliens présents durant la partie.

- historique

action_hist array historique()

Description : Renvoie la liste des actions effectuées par l'adversaire durant son tour, dans l'ordre chronologique. Les actions de debug n'apparaissent pas dans cette liste.

- **score**

```
int score(int id_joueur)
```

Description : Renvoie le score du joueur "id_joueur". Renvoie -1 si le joueur est invalide.

Parametres : *id_joueur* : Numéro du joueur

- **moi**

```
int moi()
```

Description : Renvoie votre numéro de joueur.

- **adversaire**

```
int adversaire()
```

Description : Renvoie le numéro de joueur de votre adversaire.

- **annuler**

```
bool annuler()
```

Description : Annule la dernière action. Renvoie faux quand il n'y a pas d'action à annuler ce tour-ci.

- **tour_actuel**

```
int tour_actuel()
```

Description : Retourne le numéro du tour actuel.

6 Notes sur l'utilisation de l'API

6.1 C

- Les booléens sont représentés par le type `bool`, défini par le standard du C99, et que l'on retrouve dans le header `stdbool.h`;
- Les fonctions prenant des tableaux en paramètres et retournant des tableaux utilisent à la place de ces tableaux une structure `type_array`, où `type` est le type des données dans le tableau. Ces structures contiennent deux éléments : les données, `type* datas`, et la taille, `size_t length`. Dans tous les cas, la libération des données est laissée au soin du candidat;
- Tout le reste est comme indiqué dans le sujet.

6.2 C++

- Les tableaux sont représentés par des `std::vector<type>`;
- Le reste est identique au sujet.

6.3 C#

- Les fonctions à utiliser sont des méthodes statiques de la classe `Api`. Ainsi, pour utiliser la fonction `Foo`, il faut faire `Api.Foo`;
- Les noms des fonctions, structures et énumérations sont en `CamelCase`. Ainsi, une fonction nommée `foo_bar` dans le sujet s'appellera `FooBar` en C#.

6.4 Haskell

- L'API est fournie par le module `Api`.
- Les énumérations sont représentées par des types sommes, les structures par des records. Seule la première lettre des noms de types et de constructeurs est en majuscule. Le nom du constructeur d'une structure est son nom de type.
- La commande `make doc` permet de générer la documentation dans le fichier `doc/index.html` pour votre code ainsi que pour l'API.
- Pour pouvoir conserver des valeurs entre différents appels à vos fonctions à compléter, il faut utiliser des variables mutables :

```
import Data.IORef
import System.IO.Unsafe (unsafePerformIO)

-- La pragma NOINLINE est importante !
-- MonType ne doit pas être polymorphe !
{-# NOINLINE maVariable #-}
```

```
maVariable :: IORef MonType
maVariable = unsafePerformIO (newIORef maValeurInitiale)

fonctionACompleter :: IO ()
fonctionACompleter = do
  maValeur <- readIORef maVariable
  ...
  writeIORef maVariable maValeur'
```

6.5 Java

- Les fonctions à utiliser sont des méthodes statiques de la classe Interface. Ainsi, pour utiliser la fonction foo, il faut faire Interface.foo;
- Les structures sont représentées par des classes dont tous les attributs sont publics.

6.6 OCaml

- L'API est fournie par le fichier api.ml, qui est open par défaut par le fichier à compléter;
- Les énumérations sont représentées par des types sommes avec des constructeurs sans paramètres. Seule la première lettre des noms des constructeurs est en majuscule;
- Les structures sont représentées par des records, sauf pour la structure position qui est représentée par un couple int * int;
- Les tableaux sont représentés par des array Caml classiques.

6.7 PHP

- Les constantes sont définies via des define et doivent donc être utilisées sans les précéder d'un signe dollar;
- Les énumérations sont définies comme des séries de constantes. Se référer à la puce au-dessus;
- Les structures sont gérées sous forme de tableaux associatifs. Ainsi, une structure contenant un champ x et un champ y sera créée comme ceci : array('x' => 42, 'y' => 1337).

6.8 Python

- L'API est fournie par le module `api`, dont tout le contenu est importé par défaut par le code à compléter ;
- Les énumérations sont représentées par des `IntEnum` Python, qui peuvent-être utilisés comme ceci : `nom_enum.CHAMP` ;
- Les structures sont représentées par des `namedtuple` Python, dont on peut accéder aux champs via la notation pointée habituelle, et qui peuvent être créés comme ceci : `foo(bar=42, x=3)`, sauf pour la structure `position` qui est représentée par un couple `(x, y)`.

6.9 Rust

- L'API est fournie par le module `api`, dont tout le contenu est importé par défaut par le code à compléter ;
- Les noms des structures et énumérations sont en `CamelCase`. Ainsi, une structure nommée `foo_bar` dans le sujet s'appellera `FooBar` en Rust.
- Les tableaux sont représentés par des `Vec<T>`.

Vous **devez** réussir cette mission, autrement, si l'invasion alien a lieu alors

