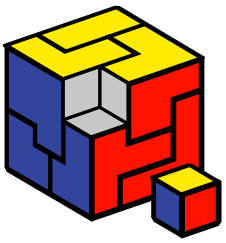
Prologin 2014



Tours de magie

« Windowsium Levista ! »

Sujet de la finale du Concours National d'Informatique
Jeudi 8 mai 2014

Table des matières


Prologin
Concours national d'informatique

1 Introduction

Ce mercredi soir, après avoir joué à des jeux de plateau avec les autres participants, le candidat [1] se rendit dans le dortoir. Sur son chemin, il tomba nez à nez avec une bestiole toute mignonne [2] qui se mit à lui parler :

« Ça t'intéresserait d'avoir des pouvoirs magiques ? »

Le candidat s'arrêta. Était-ce vraiment cet animal qui lui avait parlé ? Ses lèvres n'avaient pourtant pas bougé. La bestiole continua.

« Oui, c'est bien à toi que je parle. Ça ne te dirait pas d'être capable en un clin d'œil de construire des tours de magie ? »

Le candidat chercha aussitôt un bouton « *Report Spam* » sur les murs mais, n'en trouvant pas, il se décida à répondre.

« Attends, par tour de magie, tu veux dire genre lapin, chapeau et tout ?
— Je parle de tourelles.
— Pour quoi faire ?
— Pour tuer des sorciers.
— Pour quoi faire ?
— Pour pouvoir explorer plus librement la carte.
— Pour quoi faire ?
— Pour obtenir le PLUS GROS SCORE.
— Ah, cool ! »

Profitant de l'émerveillement du candidat, le mystérieux animal ajouta :

« Vendredi soir, ce sera la *nuit de Walpurgis*. Samedi à 00 h 42, le potentiel magique sera à son paroxysme [3] et c'est pourquoi nous devons d'ici là déterminer les dix meilleurs magiciens, qui auront l'honneur de passer devant un grand conseil d'archimages à l'aube. »

La bestiole laissa passer un silence, puis reprit :

« Tu as le pouvoir de changer le cours des choses. Tu peux devenir le meilleur développeur de France [4].
— Même si je ne sais coder qu'en PHP ? »

La bestiole resta muette.

1. Celui qui lit présentement ce texte.
2. ／人•‿‿•人＼.
3. Le sommeil aussi, d'ailleurs.
4. Et gratuitement.

« … Et comment puis-je acquérir ces pouvoirs ?
— Tu as juste à faire un pacte avec moi [5]. Si tu es mineur, tes parents ont déjà signé l'autorisation parentale.
— Et qu'en est-il des autres candidats, alors ?
— Tu sais, tous les autres candidats ont effectivement conclu ce pacte avec moi, et ont accepté les conditions d'utilisation [6]. Il ne reste plus que toi.
— Ah, ben d'accord alors [7]. »

1.1 L'initiation

La bestiole emmena le candidat dans une pièce carrée, dont trois des coins étaient déjà occupés par des candidats. Au centre, un superbe artefact brillait de mille feux. À mi-chemin de chacun des quatre murs, une fontaine magique.

« Pas terrible, la déco.
— Je te trouve bien impertinent ! » s'écria la bestiole, sans s'arrêter de sourire. « Tous ces objets que tu vois sont des gisements de magie. »

« Choisis un endroit pour construire ta première tourelle.
— Ici ? » demanda le candidat, en désignant une dalle de la salle. À cet instant précis, une tourelle s'y éleva. Le candidat recula de surprise.

« Tu peux aussi la détruire, en prononçant la formule magique.
— « S'il te plaît » ?
— Mais non. Tu ne connais pas d'incantation ?
— Je ne connais que « **Wingardium Leviosa**. »
— Non, mais c'est presque ça. Qu'est-ce qui correspond à l'effondrement d'une architecture, selon toi ? »

Le candidat se rappela effectivement qu'un de ses souvenirs semblait correspondre à la description.

« **Windowsium Levista** ! » À ces mots, la tour s'effondra brusquement.

La voix continua.

« Détruire une tour te rapporte des points de magie. Tu peux également créer des sorciers depuis ta base, qui se déplacent sur la carte pour attaquer d'autres sorciers ainsi que des tourelles adverses, ou récolter des points de magie sur l'artefact ou les fontaines.

5. NON C'EST UN PIÈGE
6. Cf. *Terms of Service ; Didn't Read*, `http://tosdr.org`
7. Voyons. Ne sois pas candide.

— Pour quoi faire ?
— Pour faire progresser tes sorciers jusqu'à un coin de la salle, ce qui anéantira le joueur adverse correspondant !
— Pour quoi faire ?
— Pour que le candidat soit mort.
— Pour quoi faire ?
— Pour avoir du rab' au banquet du samedi.
— Ah, cool ! »

1.2 Dénouement

« Dans les clauses du pacte, je peux exaucer ce que tu veux. Quel est ton vœu ?
— J'aimerais avoir deux vœux.
— Les explosions combinatoires ne sont pas acceptées. Que désires-tu le plus au monde ? »

Le candidat s'arrêta et réfléchit.

« … Un… un ordinateur portable.
— Eh bien, combats tous les candidats, et tu l'obtiendras. Bonne chance ! »

La bestiole s'éclipsa.

« Attends ! Qui es-tu vraiment ?
— Je suis… »

La bestiole se retourna et amplifia son sourire.

« … un incubateur. »

2 Sujet

2.1 Généralités

Le sujet de la finale de Prologin 2014 est un jeu de stratégie divisé en plusieurs phases. Le but du jeu est de vaincre les joueurs ennemis et de prendre possession de l'artefact central et des fontaines magiques.

2.1.1 Nombre de joueurs

Une partie voit s'affronter 4 joueurs à la fois (1-4). Chaque joueur possède une base placée dans un coin d'une carte carrée. Au centre de la carte se trouve l'artefact magique (A) et des fontaines magiques (F) sont disposées aux points cardinaux (nord, sud, est et ouest) de la salle.

2.1.2 Carte

La carte a une taille constante (longueur et largeur) : TAILLE_TERRAIN par TAILLE_TERRAIN.

```
1 . . . . F . . . . 2
. . . . . . . . . . .
. . . . . . . . . . .
. . . . . . . . . . .
. . . . . . . . . . .
F . . . . A . . . . F
. . . . . . . . . . .
. . . . . . . . . . .
. . . . . . . . . . .
. . . . . . . . . . .
3 . . . . F . . . . 4
```

Légende :

- `1..4` : Joueurs 1 à 4
- `.` : Case vide
- `A` : Artefact
- `F` : Fontaine

(Le schéma n'est pas à l'échelle.)

2.1.3 But du jeu

Le but du jeu est de totaliser un maximum de points à la fin d'une partie. Les différents moyens de gagner des points sont :

- survivre à la fin de la partie (1 point) ;
- conquérir le camp d'un joueur ennemi (1 point) ;
- contrôler une fontaine de magie à la fin de la partie (1 point), c'est-à-dire avoir un de ses soldats sur la case ;
- contrôler l'artefact à la fin de la partie (4 points).

2.1.4 Magie

La magie est l'unité de base du jeu. À chaque tour, vous gagnez MAGIE_TOUR points de magie de base, et MAGIE_FONTAINES par fontaine

contrôlée. Vous pouvez gagner des points de magie en détruisant des soldats adverses ou en supprimant une de vos tourelles. Les points de magie permettent de construire des tourelles et de produire des sorciers.

2.1.5 Déroulement du jeu

Chaque phase se déroule de manière simultanée pour tous les joueurs : tout le monde exécute ses actions en même temps. À la phase suivante, vous observez le résultat des actions des autres joueurs. Il n'existe aucun conflit d'action possible.

2.2 Phases

Chaque tour se déroule en 4 phases :

1. construction ;
2. déplacement ;
3. tir des tourelles ;
4. siège.

2.2.1 Construction

Lors de la phase de construction, il est possible de construire des tourelles et des sorciers pour des coûts COUT_TOURELLE et COUT_SORCIER. Les sorciers sont toujours construits dans la base du joueur.

Une tourelle peut être construite sur une case sous certaines conditions :
— ce ne doit être ni une base, ni une fontaine, ni un artefact ;
— cette case doit être proche d'une autre tour du même joueur (être dans la « portée de construction » d'une tour) ;
— le joueur doit avoir une tour qui est strictement plus proche de cette case que n'importe quelle tour d'un joueur ennemi (c'est pourquoi deux joueurs ne risquent pas de construire une tour au même endroit).

Les tours ont une portée d'attaque minimale PORTEE_TOURELLE, qui peut être augmentée lors de la construction : pour l'augmenter de N cases, on paie un surcoût de COUT_PORTEE + N*N. On paie donc au total COUT_TOURELLE + COUT_PORTEE + N*N.

Il est également possible de détruire vos propres tourelles pendant cette phase, si elles vous bloquent le passage. Vous récupérez MAGIE_SUPPRESSION points de magie pour chaque tourelle vous appartenant que vous détruisez.

2.2.2 Déplacement

Lors de la phase de déplacement, on peut choisir de bouger un certain nombre de sorciers d'une distance PORTEE_SORCIER. Cependant, un sorcier ne peut pas se rendre sur une case où il y a une tourelle.

Attaque des sorciers À la fin de chaque phase de déplacement, lorsque plusieurs sorciers ennemis se retrouvent sur la même case, ils s'attaquent automatiquement. Le nombre de sorciers du deuxième groupe en plus grand nombre est retiré à l'ensemble des groupes, ce qui fait qu'il ne reste plus que les soldats d'un unique joueur sur la case, ou bien aucun soldat (en cas d'égalité des deux groupes les plus nombreux).

Exemple : si A, B, C, D ont respectivement 1, 3, 3 et 7 soldats, à l'issue du combat :

— A n'a plus que 7 - 3 = 4 unités ;
— B, C et D : 0 unité.

Le joueur restant, s'il existe, gagne MAGIE_COMBAT points de magie pour chaque sorcier retiré aux autres joueurs (ici, (1 + 3 + 3) * MAGIE_COMBAT).

2.2.3 Tir des tourelles

Lors de la phase de tir, les tourelles peuvent répartir leurs points d'attaque (ATTAQUE_TOURELLE) sur un ensemble de cases. Chaque point d'attaque utilisé correspond à un sorcier en moins sur la case choisie. Il n'est pas possible d'attaquer des tourelles avec cette technique.

Tuer des sorciers à distance ne rapporte aucun point de magie.

2.2.4 Siège

Lors de la phase de siège, les sorciers peuvent attaquer les tourelles qui se trouvent sur une case adjacente (haut, bas, gauche, droite). Chaque tourelle a VIE_TOURELLE points de vie à sa création, et en perd un par nombre de sorciers qui l'attaquent à chaque tour. Elle ne peut pas en regagner.

Lorsqu'elle n'a plus aucun point de vie, la tourelle est détruite et laisse la voie libre aux sorciers.

Capture À la fin de chaque tour[8] :

— si un sorcier se trouve sur la base d'un ennemi, ce dernier est vaincu, et toutes ses unités (tourelles et sorciers) sont supprimées de la carte ;
— si un sorcier se trouve sur une fontaine, il fait gagner MAGIE_FONTAINES points de magie au joueur qui le contrôle.

2.3 Fin de la partie

La partie s'arrête au bout de MAX_TOUR tours.

8. Et non tourelle, vous suivez ?

3 API

Constante : TAILLE_TERRAIN
Valeur : 31
Description : Taille du terrain (longueur et largeur)

Constante : NB_JOUEURS
Valeur : 4
Description : Nombre de joueurs dans la partie

Constante : MAX_TOUR
Valeur : 100
Description : Nombre maximum de tours à jouer avant la fin de la partie

Constante : MAGIE_TOUR
Valeur : 20
Description : Magie gagnée à chaque tour

Constante : MAGIE_FONTAINES
Valeur : 15
Description : Magie gagnée à chaque tour pour chaque fontaine possédée

Constante : MAGIE_COMBAT
Valeur : 1
Description : Magie gagnée à chaque sorcier tué

Constante : MAGIE_SUPPRESSION
Valeur : 10
Description : Magie récupérée à chaque tourelle supprimée

Constante : COUT_SORCIER
Valeur : 2
Description : Nombre de points de magie par sorcier

Constante : COUT_TOURELLE
Valeur : 20
Description : Nombre de points de magie par tourelle

Constante : COUT_PORTEE
Valeur : 4
Description : Coût de base pour chaque case de portée supplémentaire

Constante : PORTEE_SORCIER
Valeur : 4
Description : Nombre maximum de cases qu'un sorcier peut franchir à chaque tour

Constante : PORTEE_TOURELLE
Valeur : 3
Description : Portée de base d'une tourelle

Constante : CONSTRUCTION_TOURELLE
Valeur : 3
Description : Portée de construction des tourelles

Constante : VIE_TOURELLE
Valeur : 10
Description : Points de vie d'une tourelle à sa création

Constante : ATTAQUE_TOURELLE
Valeur : 10
Description : Points d'attaque d'une tourelle au début d'un tour

Constante : POINTS_SURVIVRE
Valeur : 1
Description : Points gagnés pour avoir survécu à la fin de la partie

Constante : POINTS_VAINQUEUR
Valeur : 1
Description : Points gagnés pour avoir vaincu un adversaire

Constante : POINTS_CONTROLE_FONTAINE
Valeur : 1
Description : Points gagnés pour contrôler une fontaine à la fin de la partie

Constante : POINTS_CONTROLE_ARTEFACT
Valeur : 4
Description : Points gagnés pour contrôler un artefact à la fin de la partie

- **case_info**

Description : Information sur les cases
Valeurs :

CASE_SIMPLE :	Case simple
CASE_TOURELLE :	Tourelle
CASE_BASE :	Base du joueur
CASE_FONTAINE :	Fontaine magique
CASE_ARTEFACT :	Artefact magique
CASE_ERREUR :	Erreur

- **erreur**

Description : Erreurs possibles

Valeurs :

OK :	L'action s'est effectuée avec succès
ANNULER_IMPOSSIBLE :	Aucune action à annuler
CASE_IMPOSSIBLE :	Cette case n'existe pas
CASE_ADVERSE :	Vous ne contrôlez pas cette case
CASE_UTILISEE :	Cette case n'est pas libre
CASE_VIDE :	Cette case est vide
VALEUR_INVALIDE :	Cette valeur est invalide
MAGIE_INSUFFISANTE :	Vous n'avez pas assez de magie
SORCIERS_INSUFFISANTS :	Vous n'avez pas assez de sorciers
ATTAQUE_INSUFFISANTE :	Vous n'avez pas assez de points d'attaque
PHASE_INCORRECTE :	Cette action ne peut pas être utilisée lors de cette phase du jeu.
PORTEE_INSUFFISANTE :	Vous n'avez pas assez de portée pour effectuer cette action
PERDANT :	Vous avez perdu et ne pouvez pas effectuer d'actions

- **position**

```
struct position {
    int x;
    int y;
};
```

Description : Représente la position sur la carte

Champs :

x :	Coordonnée en X
y :	Coordonnée en Y

- **tourelle**

```
struct tourelle {
    position pos;
    int portee;
    int joueur;
    int vie;
    int attaque;
};
```

Description : Représente une tourelle
Champs :
- *pos* : Position de la tourelle
- *portee* : Portée de la tourelle
- *joueur* : Joueur qui possède la tourelle
- *vie* : Nombre de points de vie de la tourelle
- *attaque* : Nombre de points d'attaque de la tourelle

• **info_case**

```
case_info info_case(position pos)
```

Description : Retourne le type de la case à l'emplacement 'pos'
Parametres : *pos* : Position de la case

• **tourelles_joueur**

```
tourelle array tourelles_joueur(int joueur)
```

Description : Retourne la liste des tourelles qui appartiennent au joueur "joueur"
Parametres : *joueur* : Identifiant du joueur

• **magie**

```
int magie(int joueur)
```

Description : Retourne la magie que possède le joueur "joueur"
Parametres : *joueur* : Numéro du joueur

• **nb_sorciers**

```
int nb_sorciers(position pos, int joueur)
```

Description : Retourne le nombre de sorciers du joueur "joueur" sur la case "pos"
Parametres :
- *pos* : Case
- *joueur* : Identifiant du joueur

• **nb_sorciers_deplacables**

```
int nb_sorciers_deplacables(position pos, int joueur)
```

Description : Retourne le nombre de sorciers du joueur "joueur" déplaçables sur la case "pos"
Parametres : *pos* : Case
joueur : Identifiant du joueur

• **joueur_case**

```
int joueur_case(position pos)
```

Description : Retourne le numéro du joueur qui contrôle la case "pos"
Parametres : *pos* : Case

• **tourelle_case**

```
tourelle tourelle_case(position pos)
```

Description : Retourne la tourelle située sur la case "pos"
Parametres : *pos* : Case de la tourelle

• **base_joueur**

```
position base_joueur(int joueur)
```

Description : Retourne la position de la base du joueur "joueur"
Parametres : *joueur* : Identifiant du joueur

• **constructible**

```
bool constructible(position pos, int joueur)
```

Description : Retourne vrai si l'on peut construire sur la case "pos"
Parametres : *pos* : Case
joueur : Identifiant du joueur

• **chemin**

```
position array chemin(position pos1, position pos2)
```

Description : Retourne la liste des positions constituant le plus court chemin allant de la case “pos1“ à la case “pos2“. Attention : Cette fonction est lente.
Parametres : *pos1* : Case de départ
pos2 : Case d’arrivée

• **construire**

```
erreur construire(position pos, int portee)
```

Description : Construire une tourelle à la position “pos“
Parametres : *pos* : Position
portee : Portée

• **supprimer**

```
erreur supprimer(position pos)
```

Description : Supprimer une tourelle à la position “pos“
Parametres : *pos* : Position

• **tirer**

```
erreur tirer(int pts, position tourelle, position cible)
```

Description : Tirer avec “pts“ points de dégât depuis la tourelle “tourelle“ sur la position “cible“
Parametres : *pts* : Nombre de points de dégât
tourelle : Position de la tourelle
cible : Position de la cible

• **creer**

```
erreur creer(int nb)
```

Description : Créer “nb“ sorciers dans la base
Parametres : *nb* : Position d’arrivée

• **deplacer**

```
erreur deplacer(position depart, position arrivee, int nb)
```

Description : Déplace “nb“ sorciers de la position “depart“ jusqu’à la position “arrivee“.

Parametres :
depart : Position de départ
arrivee : Position d’arrivée
nb : Nombre de sorciers à déplacer

• **assieger**

```
erreur assieger(position pos, position cible, int nb_sorciers)
```

Description : Attaquer la tourelle à la position “cible“ depuis la position “pos“

Parametres :
pos : Position des sorciers
cible : Position de la tourelle à attaquer
nb_sorciers : Nombre de sorciers attaquant la tourelle

• **moi**

```
int moi()
```

Description : Retourne le numéro de votre joueur

• **adversaires**

```
int array adversaires()
```

Description : Retourne la liste des numéros de vos adversaires

• **tour_actuel**

```
int tour_actuel()
```

Description : Retourne le numéro du tour actuel

• **distance**

```
int distance(position depart, position arrivee)
```

Description : Retourne la distance entre deux positions
Parametres : *depart* : Départ
arrivee : Arrivée

• **annuler**

```
erreur annuler()
```

Description : Annule la dernière action

4 Notes sur l'utilisation de l'API

4.1 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 size`. Dans tous les cas, la libération des données est laissée au soin du candidat ;
— Tout le reste est comme indiqué dans le sujet.

4.2 C++

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

4.3 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#.

4.4 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.

4.5 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.

4.6 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).`

4.7 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 représentés 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).

4.7.1 Python 2

La version de Python par défaut est Python 3. Cependant, si vous souhaitez toujours coder en Python 2, vous pouvez utiliser le dossier *python2* à la place du dossier *python*.

Les différences par rapport à Python 3 sont les suivantes :

— les constantes des énumérations sont représentées par de simples entiers ;
— les gens dans la rue vous jettent des tomates et crient au scandale.