
SA1.01 IMPLÉMENTATION
SUJET N°2
CyberAttack@IUT'O

1 Introduction

Une puissance extra-terrestre de la galaxie d'Andromède souhaite envahir la Terre afin d'en faire un parc d'attraction. Pour arriver à ses fins, elle a détecté qu'il fallait anéantir l'élite de l'informatique sur Terre, c'est-à-dire les étudiants de l'IUT'O ! Grâce à un rayon intergalactique pointé sur le département Informatique de l'IUT'O, les extra-terrestres ont réussi à modifier le comportement des étudiants qui n'ont plus qu'une obsession : saboter la SAÉ *Implémentation* des autres groupes !

En effet, chaque groupe a à sa disposition un serveur 🖥️ et 5 ordinateurs 💻 qu'il peut protéger grâce à des dispositifs comme des *firewalls* 🔥 ou des *antivirus*¹ 🦋. Pour attaquer les autres groupes, les étudiants produisent des chevaux de Troie 🐘 (appelés *trojans*) qu'ils envoient détruire les équipements des autres groupes !

Quel groupe arrivera à sauver ses équipements pour rendre sa SAÉ en temps et en heure ? Ce groupe d'élite saura-t-il sauver la terre de l'invasion extra-terrestre ? Nous le saurons à la fin du semestre lors de la CyberAttack finale !

2 Présentation du jeu

CyberAttack@IUT'O est un jeu librement inspiré de Bad Bones ([https://fr.wikipedia.org/wiki/Bad_Bones_\(jeu\)](https://fr.wikipedia.org/wiki/Bad_Bones_(jeu))) dans lequel chaque joueur possède un plateau constitué d'une matrice de 5×5 (voir figure 1) cases au milieu duquel est disposé un serveur. En bas de ce plateau, le joueur possède 5 ordinateurs. Ces deux équipements sont les éléments à préserver des cyber attaques des autres joueurs.

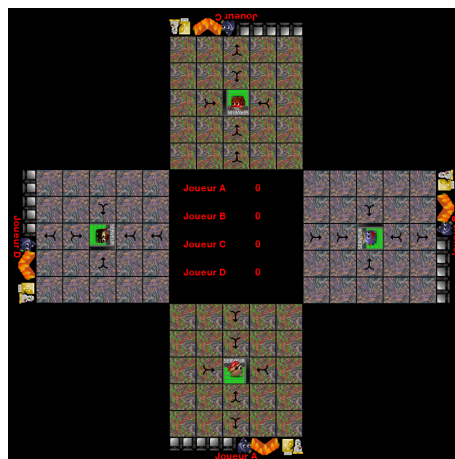






FIGURE 1 – Le plateau des quatre joueurs

1. La liste complète sera donnée par la suite

Pour attaquer, le joueur produit des chevaux de Troie qu'il va envoyer sur le plateau des trois autres joueurs. Un cheval de Troie se déplace tout droit sur le plateau sauf s'il se trouve sur une case où est dessinée une flèche, dans ce cas il s'oriente dans le sens de la flèche. Lorsqu'un cheval de Troie arrive sur le serveur il lui inflige des dégâts. De même si un cheval de Troie sort par le bas du plateau, il inflige des dégâts aux ordinateurs.

Pour se défendre chaque joueur possède un avatar qui peut détruire les chevaux de Troie. Cet avatar est placé au milieu du plateau au début du jeu et doit se déplacer d'une case à chaque tour de jeu. En défense le joueur possède aussi un certain nombre de protections qu'il peut mettre en place afin de détourner les chevaux de Troie. Ces protections sont les suivantes :

-  *Data Protection Officer (DPO)* qui permet de faire faire demi-tour aux chevaux de Troie qui s'approchent.
-  *Firewall* qui permet de faire tourner un cheval de Troie d'un quart de tour. Il y a deux types de firewall expliqués dans la suite.
-  *Données personnelles* qui permet d'attirer les chevaux de Troie vers une case.
-  *Antivirus* qui permet de renvoyer le cheval de Troie à son expéditeur.

Les protections ne peuvent pas se placer sur une case déjà occupée (par l'avatar, le serveur, des chevaux de Troie ou une autre protection). Elles ne persistent que sur deux tours de jeu.

Le jeu se joue à quatre joueurs qui possèdent tous un plateau tel que décrit ci-dessus. Un tour de jeu se déroule en trois phases :

1. Choix stratégiques des joueurs (et application de ces choix).
2. Orientation des chevaux de Troie attirés par des données personnelles.
3. Déplacement des chevaux de Troie.

2.1 Choix stratégiques des joueurs

Chaque joueur a deux actions à faire :

1. Le joueur **DOIT** choisir entre placer une protection ou créer des chevaux de Troie.
 - Pour placer une protection, il choisit la protection qu'il veut mettre, la case de son plateau où il veut la placer avec la contrainte que cette case ne doit pas être occupée.
 - Pour créer des chevaux de Troie, le joueur doit choisir pour chacun de ses adversaires (gauche, droite et haut) le numéro de la ligne ou de la colonne où le cheval de Troie va apparaître sachant que la direction du cheval sera déterminée par la destination. Un cheval de Troie envoyé au voisin de droite apparaîtra sur la gauche du plateau de ce voisin et ira vers la droite. Un cheval de Troie envoyé au voisin de gauche apparaîtra sur la droite du plateau de ce voisin et ira vers la gauche. Enfin, un cheval de Troie envoyé au voisin du haut apparaîtra en haut du plateau de ce voisin et ira vers le bas.
2. Le joueur **DOIT** déplacer son avatar d'une case en horizontal, vertical ou diagonal. L'avatar peut se déplacer sur une case déjà occupée. Si des chevaux de Troie étaient sur la case d'arrivée, ceux-ci sont détruits ce qui rapporte des points au joueur.

Ces deux actions seront exécutées dans cet ordre. En d'autres termes, le joueur peut poser une protection puis déplacer son avatar sur cette protection.

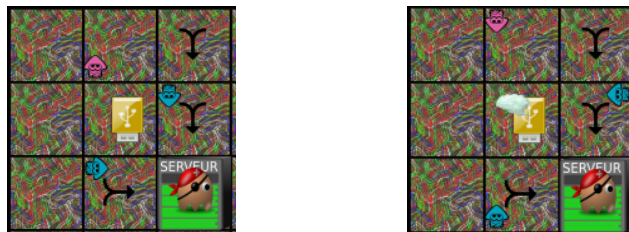


FIGURE 2 – Effet des données personnelles sur les chevaux de Troie

2.2 Orientation des chevaux de Troie voisins des données personnelles

Sur chaque plateau, si une case possède la protection *données personnelles*, tous les chevaux de Troie qui sont sur une case adjacente horizontalement ou verticalement à cette case seront réorientés vers elle (voir figure 2). Le joueur gagne autant de points qu'il y a de chevaux de Troie dans les cases adjacente. Notez qu'un cheval de Troie qui se trouve à proximité de deux protections *données personnelles* sera touché par les deux successivement dans l'ordre de parcours du plateau.

2.3 Déplacement des chevaux de Troie

Les chevaux de Troie vont se déplacer dans la case adjacente correspondant à leur direction. Un certain nombre de cas sont à considérer :

- La case d'arrivée contient l'avatar, le cheval de Troie est détruit et le joueur gagne 5 points
- La case d'arrivée contient le serveur, le cheval de Troie est détruit, le joueur perd 5 points et le serveur une resistance
- Le cheval de Troie sort du plateau :
 - si c'est par le bas, le cheval de Troie est détruit, le joueur perd 5 points et les ordinateurs une resistance
 - si c'est par un autre coté, le cheval de Troie devient la propriété du joueur à qui appartient le plateau et il est envoyé sur le plateau du joueur correspondant à la direction. Attention, lorsque le cheval de Troie sort par le haut, sa direction est inversée avant qu'il soit envoyé au voisin du haut.
- La case d'arrivée contient une protection autre que *données personnelles*, l'action de la protection est exécutée
 - *Data Protection Officer* : Le cheval de Troie retourne sur sa case initiale et sa direction est inversée.
 - *Firewall BasDroit-HautGauche* : Le cheval de Troie est réorienté de la manière suivante : gauche→haut, droite→bas, haut→droite, bas→gauche et envoyé vers la case correspondante (voir figure 3).
 - *Firewall BasGauche-HautDroit* : Le cheval de Troie est réorienté de la manière suivante : gauche→bas, droite→haut, haut→gauche, bas→droite et envoyé vers la case correspondante (voir figure 4).
 - *Antivirus* : Le cheval de Troie est renvoyé à son propriétaire actuel et devient la propriété du joueur sur lequel se trouve l'antivirus.

Dans tous les cas, le joueur gagne autant de points qu'il y a eu de chevaux de Troie affectés.

- Dans les autres cas le virus arrive sur la case et se réoriente si la case contient une flèche.

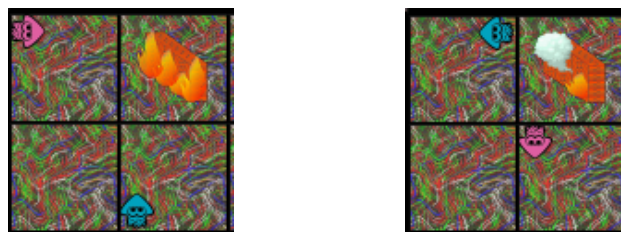


FIGURE 3 – Effet du firewall BasDroit-HautGauche : le cheval de Troie rose se dirige vers la droite et le bleu vers le haut, le firewall envoie le cheval de Troie rose dans la case sous le firewall et il se dirige maintenant vers le bas tandis que le bleu est envoyé sur la case de gauche et se dirige maintenant vers la gauche

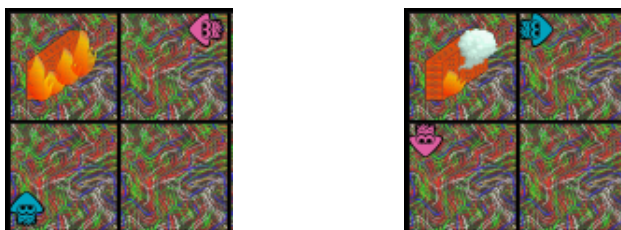


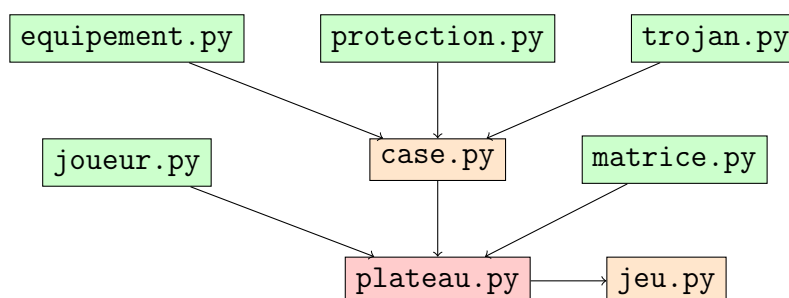
FIGURE 4 – Effet du firewall BasGauche-HautDroit : le cheval de Troie rose se dirige vers la gauche et le bleu vers le haut, le firewall envoie le cheval de Troie rose dans la case sous le firewall et il se dirige maintenant vers le bas tandis que le bleu est envoyé sur la case de droite et se dirige maintenant vers la droite

Chaque joueur gagne autant de points qu'il y a de virus restant sur les plateaux à la fin du tour et les protections présentes sur les plateaux perdent une résistance. Celles qui arrivent à un niveau 0 de résistance disparaissent.

Le jeu se termine si au moins 3 joueurs ont perdu leur serveur et tous leurs ordinateurs, ou si on a effectué un nombre de tour de jeu déterminé. Le vainqueur est le joueur qui a le plus de points à la fin de la partie.

3 Implémentation

Pour implémenter ce jeu, on vous demande d'implémenter les API dont voici un graphe indiquant leurs dépendances. Les modules verts sont principalement des structures de données pour lesquelles on veut stocker ou consulter les informations qui sont dedans. Les autres contiennent des fonctions plus complexes. Le module principal étant `plateau.py` qui contient les fonctions implémentant le principe du jeu.



Voici quelques détails sur les différents modules. La spécification des fonctions est donnée sous la forme de docstring dans les fichiers Python. Pour faire votre implémentation vous pouvez choisir les structures de données qui vous semblent le plus adéquat mais vous devez respecter la spécification des fonctions.

3.1 Equipement

Un équipement est contient deux informations : son type et sa résistance. Le type est un entier qui vaut soit `SERVEUR`, soit `ORDINATEURS`. La resistance est un entier indiquant au bout de combien d'attaques l'équipement est hors service. Notez que les ordinateurs sont considérés comme un seul équipement.

3.2 Protection

Une protection possède un type et une resistance. Comme pour les équipements le type est un entier indiquant le type de la protection. La résistance est un entier indique au bout de combien de tour l'équipement est détruit.

3.3 Trojan (cheval de Troie)

Un trojan possède trois informations :

- un créateur : un entier indiquant le numéro du joueur qui a créé le cheval de Troie,
- un type qui est un nombre entre 0 et 4 indiquant le numéro de la ligne ou de la colonne par où le cheval de Troie va entrer sur le plateau,
- une direction qui est un des caractères `D,G,H,B` indiquant dans quelle direction le cheval de Troie se dirige.

3.4 Case

Cette API sert à gérer les cases du plateau. Une case du plateau peut contenir un certain nombre d'objets :

- une flèche redirigeant les chevaux de Troie qui arrivent sur elle
- l'avatar
- une protection
- un équipement
- une liste de chevaux de Troie

Si la case possède une flèche, celle-ci sera représentée par un des caractères `D,G,H,B`. L'avatar n'a pas de caractéristiques particulières, donc il est juste nécessaire de savoir s'il est présent ou non sur la case. L'équipement ou la protection seront stockés sur la case. Enfin les chevaux de Troie sont stockés sous la forme de listes. **Attention !** Afin de gérer les actions provoquées par les objets présents sur la case, nous allons avoir besoin de deux listes de chevaux de Troie :

- la liste des trojans présents effectivement sur la case,
- la liste des trojans arrivants sur la case : il se peut qu'un trojan arrivant sur la case soit finalement détruit (par l'avatar ou le serveur), ou repousser vers une autre case.

La fonction `mettre_a_jour_case()` va justement permettre de passer les trojans arrivants sur la case à la liste des trojans présents sur la case. Cette fonction permettra de compter combien de trojans de chaque joueur sont arrivés sur la case.

Notez aussi que la fonction `poser_avatar()` va détruire les trojans qui sont présents sur la case.

3.5 Matrice

Cette API contient une implémentation des matrices. Quelques fonctions supplémentaires vous sont proposées pour vous aidez à faire une IA par la suite

3.6 Joueur

Un joueur possède trois informations : son identifiant (un entier entre 1 et 4), son nom et son nombre de points Dans ce module, on gère la position relative des joueurs les uns par rapport aux autres.

3.7 Plateau

Ce module gère le plateau d'un joueur. Un plateau possède

- un joueur
- une matrice 5×5 de cases qui sont initialisées comme indiqué figure 1
- un serveur
- des ordinateurs
- La liste des chevaux de Troie reçus de la part des autres joueurs
- Les chevaux de Troie à envoyer aux autres joueurs

Ce module commence par une série de fonctions (getters et setters) nécessaires aux fonctions de sauvegarde et restauration d'un plateau. Ces fonctions sont à implémenter en priorité. Ensuite viennent les fonctions qui permettent de gérer les phases de jeu, puis les fonctions de sauvegarde et restauration du plateau. Les fonctions qui gèrent les déplacements sur le plateau sont organisées de la manière suivante

- `deplacer_avatar()` qui permet d'indiquer vers où doit se déplacer l'avatar. La principale particularité de cette fonction est que les directions sont notées par deux lettres car l'avatar peut se déplacer en diagonal.
- `diriger_trojans()` qui permet d'orienter tous le chevaux de Troie présents sur une case adjacente à une protection de type données personnelles.
- `deplacement_phase1()` qui va faire entrer les chevaux de Troie entrants du plateau et déplacer tous les chevaux de Troie présents sur les cases, vers les listes entrantes des cases destination. Certains chevaux de Troie sortiront du plateau, ils seront alors ajoutés aux trojans sortants du plateau.
- `deplacement_phase2()` qui va appliquer les effets des protections autres que données personnelles. Certains chevaux de Troie seront transférés vers la liste des entrants d'une autre case (ou vers les sortants du plateau). A la fin de cette phase, la finalisation des déplacements sera effectuée, permettant de transférer les trojans entrants vers les trojans présents.

Notez que les transferts de chevaux de Troie d'un plateau à l'autre sont effectués dans le module `jeu.py`

Les fonctions de déplacements seront appelées dans la fonction `executer_ordres()` qui prend en paramètres les ordres donnés par le joueur sous la forme d'une chaîne de caractères dont les deux premiers indiquent le déplacement de l'avatar le troisième caractère est

- soit un A pour une attaque : dans ce cas les caractères suivants sont `GxHyDz` où `x` `y` et `z` sont des chiffres entre 0 et 4 indiquant le numéro de la ligne ou de la colonne où sera envoyé le trojan

- soit un P pour une protection : dans ce cas les caractères suivants seront trois chiffres `tlc` où `t` est le type de la protection, `l` la ligne où poser la protection et `c` la colonne où poser la protection

Par exemple, la chaîne `"HGAG2H1D4"` indique que le joueur veut déplacer son avatar en diagonal vers le haut gauche, il veut effectuer une attaque en envoyant un cheval de Troie à gauche en ligne 2 en haut en colonne 1 et à droite en ligne 4

La chaîne `"DDP241"` indique que le joueur veut déplacer son avatar vers la droite et poser une protection de type 2 en ligne 4 colonne 1.


Dans la phase finale de cette SAÉ, votre I.A devra produire une telle chaîne de caractères en fonction de l'état du jeu.

3.8 Jeu

Ce module gère la création du jeu avec la création des plateaux des quatre joueurs. La fonction `echange_trojans()` permet de transférer les trojans sortants d'un plateau aux trojans entrants des autres plateaux. On retrouve les différentes phases du jeu. L'échange de trojans se fait à la fin de l'exécution des ordres des joueurs et à la fin de la phase 2.

C'est dans ce module qu'est implémenté un joueur aléatoire et un joueur humain. Vous pourrez aussi implémenter des I.A. dans ce module afin de les tester.

3.9 Interfaces graphiques

Il y a trois programmes permettant de faire des affichages graphiques. Ces programmes reposent sur la librairie  <https://www.pygame.org>. Pour installer cette librairie, il suffit de taper la commande `python3 -m pip install -U pygame --user`.

Le programme `affichage_case` permet d'afficher une case que vous pouvez configurer dans le programme principal la case que vous voulez afficher en dé-commentant les lignes que vous souhaitez à partir de la ligne 226.

Le programme `affichage_plateau` permet d'afficher un plateau de manière statique (pas d'évolution). Un certain nombre de fichiers d'exemples de plateaux vous sont fournis dans le répertoire `donnees`. Vous pouvez tester l'affichage avec ces exemples en modifiant le programme principal en ligne 255.

Le programme `cyber_attack_at_iuto` permet de jouer au jeu en mode interactif où le joueur 1 est un être humain et les autres joueurs sont aléatoires. Vous pourrez tester vos I.A en modifiant la fonction `action_joueur` du module jeu.

4 Travail à faire et organisation

La SAÉ va se dérouler en trois grandes étapes :

1. 19 décembre minuit : de manière individuelle vous rendez

- L'implémentation des modules `trojan`, `equipement`, `joueur`, `protection`, `case`,

- item Un jeu de test pour chacun de ces modules (sous forme de fichier `test_xxx.py`) où `xxx` est le nom du module que vous testez
2. **16 janvier minuit : de manière individuelle** vous rendez
 - L'implémentation des modules `matrice` ainsi que les fonctions qui se trouvent avant le commentaire `# FIN DES FONCTIONS A IMPLEMENTER AVANT SEMAINE DE PROJET` du module `plateau`
 - Faire un rapport sur ce début d'implémentation suivant le modèle fourni sur CELENE.
 3. **20 janvier minuit : Une étape à faire en groupe** où vous aurez à finir l'implémentation du jeu et écrire une I.A. pour participer au tournoi qui aura lieu le 21 janvier. Des détails vous seront fournis par la suite sur ce que vous aurez à rendre. Les groupes de quatre seront formés par l'équipe enseignante.