

Projet de programmation

-

Mise en place du "Le Jeu de la Guerre" écrit par Guy
DEBORD

Émile BARJOU, Christophe CAUBET, Adrien HALNAUT, Romain ORDONEZ

23 janvier 2018

Résumé

Le jeu de la guerre est un jeu sur plateau pensé par Guy DEBORD qui oppose deux armées ayant pour objectif de détruire les troupes adverses ou ses moyens de communication. Ce sujet nous a été proposé par Philippe NARBEL, avec qui nous avons pu préciser les besoins autour du projet. Ainsi, ce document constitue la première version du cahier des besoins.

1

Description et analyse de l'existant

Il existe d'autres projets autour du Jeu de la guerre, la plupart d'entre eux viennent d'anciens projets de programmation dans le cadre de cette UE.

PdP 2015[\[1\]](#)

Ce projet fournit une batterie de tests non-exhaustive de l'implémentation des règles du jeu de la guerre du groupe. Le projet a subi des problèmes au sein du groupe et est resté inachevé.

PdP 2014[\[2\]](#)

Ce projet permet la visualisation d'une instance d'une partie sans interaction possible avec le plateau. Ainsi, l'interface permet la visualisation des lignes de communications directes, mais pas indirectes, la portée de déplacement d'une unité ainsi que le potentiel offensif et défensif de chaque camp sur chaque case. Ce projet pourra nous donner une idée de fonctionnalités à implémenter si on veut développer une interface graphique pour l'utilisateur.

Kriegspiel par RSG[\[4\]](#)

Ce projet n'est plus disponible sur leur site, mais reste cependant trouvable sur d'autres. Il permettait de jouer en réseau contre un autre joueur au Jeu de la guerre, cependant des règles ont été simplifiées. Il semblait bien fonctionnel, mais il serait apparemment en refonte.

Conclusion

Nous pouvons voir après l'analyse de ces cas qu'il n'existe pas, à notre connaissance, d'application permettant de jouer au jeu de Debord.

2

Besoins fonctionnels

2.1 Définition des priorités

2.1.1 Besoins primaires

L'objectif premier à réaliser et le grand pilier du projet consiste à implémenter de façon informatique les règles du Jeu de la guerre en respectant autant que possible les idées de Guy Debord, dans une sorte de moteur de règles qui permet d'affirmer si un coup donné par un joueur à un moment donné d'une partie est valide ou non. Cette implémentation devra pouvoir rendre possible l'extension du moteur pour les besoins secondaires et tertiaires ainsi que posséder une interface permettant à un utilisateur d'interagir avec le programme.

Ainsi les règles sont les suivantes :

⇒ Les éléments du plateau :

- Le plateau fait 25 cases de long, et 20 cases de hauteur.
- Le plateau est divisé en deux parties par une ligne parallèle au coté le plus long, chaque partie constituant d'un des deux camps adverses.
- Chaque moitié possède 3 forteresses, 2 arsenaux, 9 montagnes et 1 col placé entre deux cases montagnes.
- Ces éléments ne peuvent pas se superposer sur une même case.
- Chaque camp possède également plusieurs types d'unités, définis par 4 caractéristiques : le coefficient d'attaque, le coefficient de défense, la portée et les points de mouvement.
- Les montagnes ne peuvent pas être occupées ou traversées par une unité, et une attaque ne peut être portée à travers celle-ci.
- Une forteresse ou un col de montagne apportent un bonus défensif à l'unité l'occupant, sans se soucier du côté du terrain où ils sont situés.
- Une forteresse ne peut pas être détruite.
- Un arsenal est considéré comme détruit lorsqu'une unité ennemie a occupé cette case.
- Si une unité se trouve sur une case arsenal, une unité ennemie doit d'abord la détruire avant de pouvoir s'y déplacer.

⇒ Initialisation de la partie :

- Chacun leur tour, les joueurs devront placer leurs bâtiments et leurs unités sur le terrain.
- Pendant cette phase, aucun des deux joueurs n'aura conscience du placement de l'autre joueur.
- Les forteresses doivent être posées sur des cases en communication directe (voir partie communications) avec au moins un arsenal allié.
- Le joueur qui commence en premier la partie est désigné de manière aléatoire.

⇒ Déroulement d'un tour :

- Chaque joueur peut déplacer jusqu'à 5 de ses unités par tour. Il ne peut pas déplacer 2 fois une même unité pendant son tour.
- À l'issue de chaque mouvement, le joueur peut attaquer une unité ennemie à portée d'attaque de l'unité déplacée.
- Une unité ne peut se déplacer, attaquer ou défendre que si elle est en communication directe ou indirecte avec l'un de ses arsenaux.

- ⇒ Conditions de victoire (au moins une de ces conditions est nécessaire) :
 - Avoir détruit les deux arsenaux de l'adversaire.
 - Avoir détruit toutes les unités combattantes de l'adversaire.
- ⇒ Les communications :
 - Les cases-Arsenal diffusent les lignes de communication dans les 8 directions (horizontales, verticales et diagonales), sans limitation de portée. La diffusion dans l'une des directions est stoppée dès lors qu'elle croise une unité ennemie combattante ou une montagne.
 - Quand une unité combattante est alimentée par cette communication, elle diffuse l'information sur ses 8 cases adjacentes, ainsi les unités se trouvant sur ces cases diffusent à leur tour l'information de la même manière.
 - Les unités non-combattantes (les unités relais) alimentées par cette communication diffusent l'information de la même manière que les arsenaux.
- ⇒ Attaque et Défense :
 - L'attaque se déroule sur une case définie par le joueur attaquant, à savoir la case où se trouve l'unité attaquée.
 - Toutes les unités du joueur à portée d'attaquer cette même case additionnent leurs coefficients offensifs. Les unités du joueur défendant à portée d'attaquer cette même case additionnent leurs coefficients défensifs.
 - La différence entre le total des coefficients offensifs et le total des coefficients défensifs permet trois situations :
 - La différence est inférieure ou égale à 0, il ne se passe rien.
 - La différence est supérieure ou égale à 2, l'unité sur la case ciblée est détruite.
 - La différence est égale à 1, lors du prochain tour, l'adversaire devra en premier utiliser un de ses coups pour déplacer l'unité sur la case ciblée. Cette unité ne pourra participer à aucune offensive pendant ce prochain tour.
 - Une unité ayant attaqué ne peut plus se déplacer jusqu'à la fin du tour.
 - Une unité de cavalerie peut, à l'issue de son déplacement, initier une attaque contre une unité ennemie se trouvant sur une des 8 cases adjacentes à cette unité de cavalerie afin d'utiliser sa charge. La charge de cavalerie donne un coefficient offensif de 7 à cette unité de cavalerie et permet de faire participer à l'offensive les autres unités de cavalerie alliées se trouvant dans le même alignement de cases sans discontinuité de cases entre elles. Ainsi les unités de cavalerie participant à la charge voient leurs coefficients offensifs montés à 7.
 - Une charge ne peut être initiée contre une unité se situant sur un col de montagne.

2.1.2 Besoins secondaires

Le projet se divise donc en deux parties, après avoir une version jouable du jeu de la guerre entre deux utilisateurs sur une même console, il est désormais question d'ajouter une intelligence artificielle pouvant jouer au jeu.

Il s'agit d'étendre l'implémentation actuelle pour permettre de pouvoir jouer une partie complète sur le programme en utilisant des aides à la compréhension d'une situation, ainsi qu'une ébauche pour un joueur artificiel. Ces aides permettront à l'utilisateur d'appréhender plus facilement et rapidement la situation de la partie, à savoir par exemple, mais peut-être pas limité à, le potentiel offensif et défensif de chaque case du plateau, la portée des lignes de communications, la portée d'attaque des unités. Ces aides ne devront pas être exclusivement visuelle et pourront être utilisées par le joueur artificiel plus tard. Le joueur artificiel devra reconnaître l'ensemble des coups possibles à son tour, et pouvoir les jouer en pouvant utiliser des heuristiques qui seront détaillées dans les besoins tertiaires. Dans ce premier temps, on pourra par exemple, limiter ces heuristiques et plutôt jouer les coups de façon aléatoire.

1. Permettre à deux utilisateurs de jouer dans une même console à tour de rôle.
2. À son tour, l'utilisateur devra proposer une action en utilisant une syntaxe à définir.
3. Le programme générera et affichera un aperçu de la situation représentant l'état du plateau après chaque coup valide joué (nature de l'aperçu à définir ultérieurement).
4. Toujours avec cette syntaxe, un joueur devra pouvoir annuler les derniers coups qu'il a joué, jusqu'à revenir au début de son tour s'il le souhaite. (Ce mécanisme pourra être implémenter à l'aide d'une pile des instances du plateau pendant ce tour).

5. Un joueur pourra sauvegarder l'état du plateau dans un fichier en respectant un ordre logique des informations stockées, lisible par l'utilisateur.
6. Un joueur pourra générer une instance du jeu à partir d'un fichier suivant un format défini.
7. Une première IA sera implémentée. Elle sera en lien avec le moteur de règle et pourra récupérer la liste des coups valides, pourra proposer des actions au jeu et devra pouvoir être étendable.

2.1.3 Besoins tertiaires

Cette dernière partie du projet se focalise sur le joueur artificiel exclusivement, afin de pouvoir le faire jouer de façon efficace en utilisant des notions tactiques et stratégiques [3]. Il faudra probablement faire interpréter une situation donnée au joueur artificiel pour que celui-ci prenne une décision pertinente dans un temps réduit. On pourra penser par exemple à représenter les unités des deux joueurs par des "ensembles d'unités à un certain potentiel".

M. Narbel nous a notamment indiqué ces deux besoins concernant une stratégie potentielle pour l'IA :

- Évaluation ("comprehension") d'une position (statique), (Spatial Reasoning - Potential fields - PathFinding) Détermination de "points décisifs". Représentation graphique de cette évaluation (attaque, défense, communications).
- Évaluation ("comprehension") d'une suite de mouvements (dynamique). Prise en compte de techniques de planification (Planning). Représentation graphique de cette évaluation.

3

Besoins non-fonctionnels

1. Le code du moteur de règles devra avoir un certain degré de généricité pour pouvoir altérer les règles du jeu lors du développement ou même après la release finale.
2. Le temps d'attente entre un coup proposé et sa validité évaluée par le moteur de règle devra être de l'ordre de la seconde.
3. Le temps d'attente entre le début et la fin du tour de l'IA ne devra pas excéder 10 secondes.
4. Une couverture des tests sur les instances du jeu est attendue lors du développement.
5. Le jeu doit fonctionner sur un système UNIX.

Bibliographie

- [1] Antoine Gamelin, Matthieu Desplat, Aymeric Miville, Nicolas Endredi. Jeudelaguerre-pdp2015, <https://services.emi.u-bordeaux.fr/projet/savane/projects/jeudelaguerre/>, 2015.
- [2] Guillaume Desbieys, David Cheminade, Hubert Mondon, Quentin Michaud. Jeu de la guerre - kriegspiel, <https://services.emi.u-bordeaux.fr/projet/savane/projects/kriegspiel/>, 2014.
- [3] Funge Millington. *AI for Games : chapters 5 - 6 : Decision Making - Tactical and Strategic AI*. Morgan Kaufmann Publishers, 2009.
- [4] Radical Software Group. Kriegspiel, <http://r-s-g.org/kriegspiel/index.php>, 2009.