Programmer's Nighmare (v3.1)

Cauchemard programmable

Auteur : Tom Jolly

Editeur : Jolly Games

Nombre de joueurs : de 2 à 6

Durée : 20 minutes

Traduit de l'anglais par : Jean-Marc Tribet

dessin original (Jolly Games)

Il s'agit d'un jeu de stratégie chaotique mettant les joueurs dans la peau de programmeurs au prise avec un programme meurtrier. Comme dans WIZWAR, il existe deux façons de gagner, soit vous accumulez assez de points de victoire (20), soit vous êtes le dernier programmeur en vie (sigh!).

Il existe deux versions du jeu : la première consiste à sortir vainqueur d'un programme déjà construit, la seconde intègre la construction du programme par les joueurs et donc quelques paramètres supplémentaires... Dans ce jeu, les cartes sont des instructions d'un unique programme que l'on peut se représenter comme une grande boucle, et les jetons des joueurs sont des BITs et décrivent leur capacité à réaliser ou non l'instruction sur laquelle chacun sera placé.

Version 1 : tout est connu à l'avance... ou presque

Retirer du jeu les cartes suivantes (entre parenthèse le nombre d'occurence de ces cartes dans le paquet) : BREAK (3), RUN (4), SECRET (2) et INSERT (1). Note : pour un tri rapide, vous pouvez colorier l'intérieur des cartes afin de les identifier plus rapidement lorsque vous voulez passer d'une version à l'autre. En fonction du nombre de joueurs, utilisez le nombre de cartes indiqué dans le tableau ci-dessous pour constituer le programme en un cercle d'instruction (cartes), à l'instar d'une grande boucle. On place le marqueur d'avancée du programme à côté de la première instruction posée.

Nb de joueurs	***2***	***3***	***4***	***5***	***6***
Cartes formant le programme	14	18	18	18	21
bits par joueur	6	5	4	3	3

Chaque joueur dispose de 10 points et d'un quota de BITs comme indiqué dans le tableau qu'ils vont placer alternativement dans la boucle sur une instruction encore vièrge de BIT. Une fois cette phase d'initialisation réalisée, le programme démarre par la première instruction et suivra son cours (en déplaçant le marqueur d'avancée de programme d'une instruction dans le sens de la flèche) jusqu'à ce qu'un joueur atteigne une condition de victoire : 20 points ou bien élimination de tous ses adversaires.

3 cas peuvent se produire lorsque le marqueur d'avancée de programme atteint une instruction

- Elle ne dispose **pas de BIT** : l'instruction n'est pas exécutée et on passe à la suivante (lisez tout de même le texte de la carte car il se peut que l'absence de bits provoque un résultat...)
- Elle dispose d'**un BIT** : le possésseur de ce BIT décide si oui ou non on exécute l'instruction.
- Elle dispose de **plusieurs BITs** : elle peut alors être exécutée autant de fois qu'il y'a de BIT, et à chaque essai (par ordre d'arrivée des BITs sur l'instruction), le possésseur du BIT décide si oui ou non on exécute la dite instruction. Suivant les cas, une première exécution de l'instruction peut tout à fait empêcher des éxécutions suivantes ou tout au moins les modifier...

Tout joueur parvenant à réaliser une boucle infinie ne permettant ni de gagner ni de perdre des points, ne peut exécuter cette boucle que trois fois de suite avant de passer à la carte suivante à la sortie de sa boucle. Par contre, tant que la boucle permet à un joueur de perdre ou de gagner des points, elle est exécutée et cela peut tout à fait conclure la partie. Notons que cette règle ne s'applique pas dans la version 2, où il serait assez trivial de construire sa propre « boucle victoire ».

Un programmeur peut être éliminé de deux manières : soit il n'a plus aucun BITs sur le plateau, soit il atteint 0 points ou moins auquel cas on retire aussi tous ses BITs du plateau et on laisse continuer les survivants...

Version 2 : Les programmeurs réalisent le programme qui causera leur perte !!!

Quasiement identique à la version 1 mais cette fois ci, on utilise l'ensemble des cartes, et la différence vient du fait que ce sont les joueurs -les programmeurs- qui vont fabriquer, lancer puis probablement stoper le programme, le relancer et ainsi de suite... On dessinera un programme linéaire (une file de cartes) plutôt qu'un cercle pour faciliter l'ajout et la suppression d'instructions, mais globalement, cela revient au même que le cercle car lorsque le pointeur d'avancée du programme atteint une extrémité, il continue dans le même sens depuis l'extrémité opposée.

Pour initialiser le jeu, on distribue à chacun 4 cartes, l'ensemble de ses BITs, puis on place une instruction au centre avec le marqueur de programme orienté dans une direction qui sera le sens d'exécution du programme. Chaque joueur dispose de 10 points et les conditions de victoire ou de défaite sont les mêmes, il existe aussi une condition de victoire supplémentaire : Si un programmeur parvient à faire exécuter une boucle dans laquelle il est le seul à posséder des instructions, il gagne la partie !

Programmer : chacun à son tour en commençant pas le plus jeune joueur, on peut placer une instruction avec un de ses BITs dessus et en piocher une nouvelle sur la pile de cartes restantes. Les cartes d'instruction peuvent être jouées indifférement après la dernière instruction ou bien avant la première, mais en aucun cas elles ne peuvent s'intercaller entre deux instruction déjà posées. Le marqueur d'avancée du programme n'est pas déplacé lors de cette phase, de même les instructions ne sont pas éxécutées, puisque le programme n'a pas encore démarré; On se contente donc de poser des cartes qui ne prendront effet que lorsque le programme sera exécuté, suivant les BITs qui contrôlent les instructions au moment où le marqueur d'avancé de programme les atteint...

Exécuter le programme consiste à jouer une carte RUN lors de son tour à la place d'une isntruction. La séquence d'instruction s'exécute de la même façon que dans la première version du jeu, plus personne ne peut jouer de cartes hormis un BREAK qu'il défaussera. Notons que si le programme atteint une longueur de 20 instructions, il démarre automatiquement et continuera tant qu'il n'y aura pas un vainqueur : aucun BREAK ne peut être joué une fois que le programme à atteint cette taille critique. En deça des 20 instructions, le programme peut aussi être arrêté verbalement soit par le programmeur l'ayant démarré, soit par un autre programmeur pour un coût de 3 points. Enfin, une carte BREAK peut aussi bien être jouée comme instruction du programme...

Arrêter le programme qui s'exécute par une carte BREAK ou verbalement n'est pas si facile, puisqu'il faut que le marqueur d'avancée du programme arrive sur une instruction non exécutée soit par choix, soit par absence de BITs, de plus, aucun BREAK d'aucune sorte ne peut être lancé sur le programme tant qu'une instruction n'est pas sur le point d'être exécutée pour la seconde fois. Une fois l'arrêt effectué, les programmeurs reprennent leur savante construction en commençant par le joueur ayant en dernier démarré le programme.

Enfin, il existe une série de cartes comme la carte RUN qui se joue en dehors du programme. Elles portent toutes la mention « ONE-SHOTS », ce qui signifie qu'elles sont jouées une fois avant d'être défaussées. Si le jeu d'une de ces cartes vous en fait placer plusieurs, à la fin de votre tour vous repiocher suffisement de cartes pour en avoir toujours quatre en main.

Précisions

Incrément et Décrément : certaines instructions modifient les nombres indiqués sur les cartes, elles agissent uniquement sur les NOMBRES écrits en CHIFFRE et ceci donne une certaines nuance entre une instruction annonçant « Enlever 1 BIT » et une autre « Enlever un BIT », cette dernière étant insensible à ce genre de modifications... Dans le même ordre d'idée, si les nombres peuvent être rammenés à zéro, il ne peuvent devenir négatifs quelque soit le nombre de marqueur -1 que l'on place sur la carte. Vous pouvez toujours jouer sans cette règle, mais cela risque de donner un cocktaïl encore plus confus que celui que l'on vous propose...

Liste des cartes

NdT : j'ai tenté de respecter la taille originale du texte des cartes pour vous permettre de les coller par dessus la partie anglaise de votre jeu, pour ce faire, j'ai du utiliser l'abréviation « pgm » pour indiquer « programme ». Par contre, pour d'obscures raisons liées aux languages de programmation actuels, passés et sans doute futurs, j'ai conservé les titres des instructions dans la langue de Sheakspeare... Merci de ne pas utiliser la présente règle pour vous fabriquer des exemplaires « pirates », la duplication est un droit réservé au possésseur d'un exemplaire original !

ACQUIRE ***Vous gagnez 2 points !!!***	***BIT MAKER*** ***Place un nouveau BIT adverse (i.e. qui n'est pas de votre couleur) sur n'importe quelle instruction***	***BIT MOVER*** ***Déplace 1 BIT depuis n'importe quelle carte vers l'une des deux instructions adjacentes***
BIT SWAPER ***Echange la place de 2 BITs qui ne sont ni de la même couleur, ni sur la même carte***	***BUG*** ***Tous les joueurs prennent 1 point de dégât.***	***BREAK (V2)*** ***Stoppe le programme à la prochaine instruction non exécutée à partir du moment où une instruction a été atteinte au moins deux fois par le marqueur d'avancée de programme***
COPY ***Reexécute les 2 cartes précédentes dans l'ordre sans déplacer le marqueur d'avancée du programme***	***DECREMENT*** ***Réduit d'un point un chiffre d'une instruction de façon permanente*** ***utiliser un marqueur « -1 »***	***DEFEND*** ***La perte d'1 point vous permet d'éviter tout dommage causé par les 2 prochaines instructions exécutées***
DELETE ***Retire un BIT d'une carte et le remet dans le stock de son propriétaire***	***ERASE*** ***Retire la carte d'instruction précédente dans le programme (suivant son sens d'exécution)***	***FUTURE*** ***Exécute les 2 instructions suivantes dans l'ordre sans déplacer le marqueur d'avancée du programme***
FLOATER ***Déplace cette carte en avant ou en arrière d'1 instruction, puis execute l'instruction suivante en laissant le marqueur d'avancée de programme sur la présente carte***	***GO TO*** ***Avance le marqueur d'avancée du programme de 6 instructions et place un marqueur GOTO sur la carte d'arrivée qui sera toujours celle-ci même si elle se déplace par la suite***	***INCREMENT*** ***Augmente d'un point un chiffre d'une instruction hormis celle-ci*** ***utiliser un marqueur « +1 »***

ACQUIRE ***Vous gagnez 2 points !!!***	***BIT MAKER*** ***Place un nouveau BIT adverse*** ***(i.e. qui n'est pas de votre couleur)*** ***sur n'importe quelle instruction***	***BIT MOVER*** ***Déplace 1 BIT depuis n'importe quelle carte vers l'une des deux instructions adjacentes***
INSERT (one-shot) ***Tire une carte de la pioche, puis, en connaissance de cause, la place n'importe où dans le programme avec un de ses BITs au-dessus***	***MULTIPLIER*** ***Tout joueur ayant au moins un BIT sur cette carte en ajoute un nouveau*** ***Quelque soit le nombre de BIT sur cette carte, elle n'est exécutée qu'une fois par passage du marqueur pgm***	***OVERSIGHT*** ***A EXECUTER S'IL N'Y A AUCUN BIT AU DESSUS*** ***Tous les joueurs perdent 1 point, mais s'il y'a au moins un BIT personne n'est impacté***
OVERWRITE ***Remplace un des BIT posé sur une instruction du programme par un des votres, puis remplacer cette carte par une nouvelle tirée dans la pioche***	***POINTER*** ***Redirige sur un joueur de votre choix tout dommage fait sur vous par les deux cartes précédentes***	***POWER SURGE*** ***Les joueurs n'ayant pas de BIT sur cette carte perdent 3 points, ceux qui en ont ne perdent qu'1 point***
PROGRAM ERROR ***Lorsqu'on pose cette carte (en V1 quand on va en prendre le contrôle), placer un BIT de la reserve adverse au lieu du sien. Tout possesseur de BIT sur cette carte perd 1 point sans choisir si elle s'exécute***	***REPLACE*** ***Remplace une carte déjà posée dans le programme par une autre tirée dans la pioche. La couleur et l'ordre des BITS sur cette carte rete inchangé***	***REVERSE PROGRAM*** ***Change la direction d'exécution du programme en inversant le sens du marqueur d'avancée de programme***
RUN (one-shot) ***Le programme démarre en commençant par l'instruction sur laquelle se trouve le marqueur d'avancée de programme*** *Tant que le programme s'exécute conservez cette carte face visible devant vous pour indiquer que vous l'avez lancé*	***SECRET (one-shot)*** ***Placez en tête ou en queue de programme une carte de votre main face cachée avec un de vos BIT dessus. Le premier joueur décidant de son exécution retournera la carte***	***SELF DESTRUCT*** ***Cause 4 points de dégâts à un joueur avant de défausser cette carte ainsi que tous les BITs qu'elle comporte***
SEQUENCE MOD ***Déplace une instruction dans le programme avec les BITs qu'elle comporte. Si cette carte se déplace, le marqueur d'avancée de pgm la suit***	***SWAP INSTRUCTIONS*** ***Inverse les 2 instructions précédentes ou bien les 2 suivantes.*** *Note : si cette carte est modifiée d'un +1, vous pouvez changer l'ordre des 3 précédentes ou bien des 3 suivantes...*	***TIME-DELAY*** ***Donne un marqueur « SKIP » qui pourra être utilisé par la suite à la volée pour sauter une instruction (même si cette carte ou les BITs la contrôlant ont été défaussés)***
UPGRADE ***Après avoir choisi un emplacement dans le programme, placez y une carte tirée de la pioche ainsi que votre BIT contrôlant l'UPGRADE que vous remplacerez par un BIT de la réserve de votre voisin de droite***	***WORM*** ***Ajoute 1 de vos BITs sur une autre instruction que celle ci...***	***ZAP*** ***Cause 1 point de dégât à un autre joueur.***

http://www.silcom.com/~tomjolly/pnfaq.htm est une FAQ (Foire aux questions) en anglais sur ce jeu, toutes les situations tordues pouvant résulter de la combinaison de cartes n'étant pas aisées à faire tenir sur une simple règle de jeu. Les cartes traduites ci-dessus, lues posément, doivent permettent de résoudre la majeure partie des conflits, cependant, s'il en subsistait, n'hésitez pas à me contacter par mail : jmtribet@free.fr