



Les Citrons

Sujet de la finale du 09-12 Mai 2002

Table des matières

1	Introduction	3
2	Le jeu	5
2.1	Le terrain	5
2.2	Les Citrons	5
2.3	Les ponts	5
2.4	Placement des Citrons et des ponts sur une case	5
2.5	Cycle de vie	6
2.6	Equipes de Citrons	6
2.7	Contrôle d'une case par un Citron	6
2.8	Bloquage de Citrons	6
2.9	Déplacement des Citrons	6
2.10	Fusion des Citrons	7
2.11	Chute des Citrons	7
2.12	Sources d'énergie	7
2.13	Les actions des Citrons	7
2.13.1	Déplacer un Citron	7
2.13.2	Créer/Réduire un pont	7
2.13.3	Diviser un Citron	8
2.13.4	Sauver un Citron	8
2.13.5	Presser un Citron	8
2.13.6	Transporter un Citron	9
2.13.7	Ejecter un Citron	9
2.14	Ordre de jeu chaque jour	9
2.15	Fonctionnement détaillé des ponts	9
2.15.1	Déplacement sur des ponts	9
2.15.2	Charge maximale d'un pont	10
2.15.3	Déplacement sur un pont en construction	10
2.15.4	Ecroulement d'un pont	10
2.16	Visibilité	10
2.17	Energie	10
2.18	Disparition d'un Citron	11
2.19	Ordre des traitements	11
2.20	Début de la partie	12
2.21	Fin de la partie	12
2.22	But du jeu	12
2.23	Résumé des actions possibles d'un Citron	12
2.24	Limites de temps et de mémoire	13
3	L'API	14
3.1	L'interface	14
3.2	Fonctions à écrire	14
3.2.1	player_init	14
3.2.2	player_new_day	14
3.2.3	citron_day	14
3.3	Fonctions d'obtention d'informations	15

3.3.1	game_get_my_team	15
3.3.2	map_get_size_x	15
3.3.3	map_get_size_y	15
3.3.4	map_get_nb_teams	15
3.3.5	map_get_max_citrons	15
3.3.6	map_is_source	15
3.3.7	map_get_type	16
3.3.8	map_get_nb_citrons	16
3.3.9	map_get_nb_bridges	16
3.3.10	map_get_citron_team	16
3.3.11	map_get_citron_id	16
3.3.12	map_get_bridge_team	16
3.3.13	map_get_bridge_id	17
3.3.14	map_get_next_citron_id	17
3.3.15	citron_get_state	17
3.3.16	citron_get_energy	17
3.3.17	citron_get_cur_x	17
3.3.18	citron_get_cur_y	17
3.3.19	citron_get_cur_z	18
3.3.20	citron_get_origin_x	18
3.3.21	citron_get_origin_y	18
3.3.22	citron_get_action_result	18
3.4	Fonctions permettant d'effectuer des actions	18
3.4.1	move_citron	18
3.4.2	stretch_citron	19
3.4.3	retract_citron	19
3.4.4	press_citron	19
3.4.5	capt_and_move_citron	19
3.4.6	drop_citron	19
3.4.7	pull_citron	20
3.4.8	split_citron	20
3.5	Autres fonctions utiles	21
3.5.1	print_map	21
3.5.2	time_left	21
3.5.3	time_reserve_left	21
3.5.4	print_action_result	22

4 Conclusion

23

1 Introduction

Il y a longtemps, très longtemps, dans une galaxie lointaine, très lointaine, vivaient paisiblement une race fort aimable de doux Citrons, d'adorables petits êtres gentils comme tout, les Citrons ! Ceux-ci, totalement incolores, passaient leur temps à végéter dans l'eau dans un calme total. Bref, leur monde était un véritable petit paradis, fade et inintéressant comme il se doit ...

Heureusement, pour le plaisir de tous, les 'Romechextratraistr, provenant du fin fond de l'univers, à cent mille millions d'années lumières qui, ayant pris la mauvaise sortie de la voie rapide intergalactique (c'est de la faute de Glop qui était aux commandes ! (d'ailleurs c'est toujours la faute de Glop !)) se retrouvèrent dans ce coin paumé, ... euh ... cette charmante petite planète aux goûts champêtres, et commencèrent à taquiner dans un esprit bon enfant la population locale, bref, à créer d'immenses catastrophes naturelles (ou pas d'ailleurs, il ne faut jamais sous-estimer la puissance du "ou pas"), à enlever de nombreux Citrons pour effectuer diverses expériences indiscibles, à semer la zizanie et enfin à propager le dernier gadget de la technologie 'Romechextratraistr (très avancée d'ailleurs, c'est un fait reconnu dans tout l'univers que les 'Romechextratraistr ont la technologie avancée livrée en série!), un virus transgénique radioactif en état de décomposition avancée. Pour le malheur des 'Romechextratraistr, ceux-ci furent repoussés par les Citrons suite à la bourde de Glop (puisque l'on vous dit que c'est TOUJOURS de sa faute ...) qui a révélé aux Citrons qu'ils n'aimaient pas les couleurs trop vives, mais ceci est une autre histoire ... (En effet, la plupart des documentaires scientifiques (Alien, Indépendance Day...) démontrent que ces créatures se doivent d'avoir des goûts esthétiques des plus ternes (on oubliera bien sûr le délire phantasmagorique de Tim Burton, car il est impensable que les 'Romechextratraistr soient restés insensibles au signe interplanétaire du donuts!))

Bref, cela risque de vous étonner, mais l'ensemble de ces blagues de collégiens modifièrent sensiblement l'environnement de la planète Citronienne !

Seule une petite partie de la planète restante pouvait désormais recueillir les Citrons survivants. La plupart de l'eau qui composait la planète fut réquisitionnée en vue de satisfaire une commande annuelle faite par une certaine association d'une planète bleutée, afin de ... (hum ... nan nan rien ... héhéhé).

De nombreuses montagnes émergèrent à la surface de la planète, ainsi que des plaines et des trous remplis d'eau. Les Citrons eux-même furent changés du tout au tout ! (Enfin nous vous rassurons, ils sont tout de même restés d'une fainéantise admirable (ici rajouter un bon vieux proverbe campagnard disant qu'il ne faut pas se presser, ou alors on laisse comme ça) ... (C'est vraiment une chose qui dépasse l'entendement, que des Citrons refusent de se presser quand même ...)) Tout d'abord ils se colorèrent, ce qui devint leur première source de fierté, ensuite de créature aquatique ils devinrent aquaphobes (ce qui est logique ! les Citrons déteignent sous l'eau ! (d'ailleurs ils se lavent à sec ... Sinon ils perdent leur couleur et on n'aime pas ça)). Enfin leur goûts esthétiques se firent de plus en plus intransigeants, et toute couleur autre que la leur devint une injure à leur yeux. Leur but devint donc rapidement de ne plus avoir de Citrons différemment colorés à proximité (ils sont bêtes ... Forrest, qui certes est quelque part un peu "différent" est pourtant très sympa (et je n'évoque même pas les orgas de prologin ...)) Une guerre longue, terrible et sans pitié s'ensuivit ...

Dans ce carnage atroce, tous les coups sont permis ! De pousser les ennemis dans les trous d'eau (et ça, vraiment, tous les organisateurs de prologin vous diront que c'est vraiment Mal (tm)), à presser

les Citrons ennemis jusqu'à la dernière goutte, en passant par la multiplication des Citrons afin de se répandre sur tout leur espace vital en vrai petit parasite, ou alors en se fusionnant pour former un être plus puissant (pour parler de manière à ce qu'un bitmap 1.0 puisse comprendre, cela ressemble énormément à bioman et à la méga-transformation des cinq sauveurs de l'univers en force arc-en-ciel). D'ailleurs les Citrons aussi peuvent se transformer en arc-en-ciel, mais comme le rôle de sauveur de l'univers était déjà pris, cette version ressemble plus à celle des bisounours (et oui, comme toujours références références ...) et donc ces arc-en-ciels sont plus considérés comme des ponts par leur congénères (sauf bien sûr lorsque ceux-ci se cassent la figure dans les précipices avec les malheureux reposant dessus ... Mais bien entendu, ce genre de choses est dû à "pas d'chance ..." (oups, pardon, à Glop))

L'importance d'un Citron est proportionnelle à l'intensité de sa couleur. Tous les Citrons y sont sensibles, qu'ils soient ou non de couleur identique. Ainsi lorsqu'ils sont trop proches de magnifiques Citrons très colorés, les autres Citrons en restent béats d'admiration ! (oui, ce n'est vraiment pas malin les Citrons, mais bon, il faut dire que ceux qui s'occupent de leur "intelligence" ne sont pas bien malins ... (mais naaannn c pas de votre faute ... de toute manière vous pouvez toujours accuser Glop)). Pendant ce temps, les Citrons béats peuvent se faire presser comme des citrons pour que leur prédateur puisse se délecter de leur substantifique moëlle ... (l'abus de jus de Citron est dangereux pour la santé, à consommer avec modération)

Certes, les Citrons d'une même équipe coopèrent entre eux, mais il ne faut pas les croire si solidaires que ça : Autant ils ont une confiance totale envers leurs supérieurs hiérarchiques, autant ils ont vraiment tort ... Un Citron approchant de trop près son supérieur se fera instantanément presser jusqu'à ce que mort s'en suive (les détails sur la manière dont le cadavre de Citron se fait ingérer ne sera pas décrit afin de ne pas choquer les âmes sensibles ...) De même un Citron sur le point de se noyer dans un trou d'eau ne devrait pas faire confiance aux mains charitables qui pourraient lui être tendues afin de le "sauver" ... le même sort les attendrait impitoyablement ...

Tout cela pour dire que la vie d'un Citron n'est vraiment pas rose ... Enfin bon, cela aurait pu être pire ... Imaginez qu'ils soient réincarnés en bitmaps ... (enfin là ça serait quand même vraiment abusé de la part de Glop quand même ...)

2 Le jeu

2.1 Le terrain

La planète sur laquelle vivent les Citrons est immense, mais ils vivent tous dans une unique zone habitable, de taille plutôt réduite, et de forme carrée, représentée par une grille de 64*64 cases carrées. Le relief de la zone habitable est constitué de montagnes, de plaines, et de trous, le plus souvent remplis d'eau.

Une case de la grille peut donc être de chacun des 3 types suivants :

- **plaine** (niveau 0)
- **trou** (niveau -1)
- **montagne** (niveau 1)

2.2 Les Citrons

Un Citron est une créature en forme de citron, qui se déplace en rampant d'une case à l'autre. Un Citron possède une énergie, qui peut varier au cours de sa vie et qui lui est très utile pour se défendre contre des Citrons ennemis.

La position d'un Citron dans son état normal est définie par 3 paramètres :

- **x** : sa position dans la direction ouest-est
- **y** : sa position dans la direction nord-sud
- **z** : son niveau, qui vaut -1 si le Citron est dans un trou, 0 sinon

2.3 Les ponts

Un Citron peut prendre la forme d'un pont qui s'étend sur plusieurs cases, dans la direction ouest-est, ou dans la direction nord-sud. Un pont part d'une case de plaine, pour arriver vers une autre case de plaine, en passant éventuellement au dessus de cases trous. Un pont ne peut jamais passer par une case de montagne.

Il est également possible de construire un pont qui part d'une case trou, si celle-ci est déjà recouverte par un pont. Idem pour l'autre extrémité du pont.

2.4 Placement des Citrons et des ponts sur une case

Une case de plaine (niveau 0) peut contenir un Citron par équipe au maximum, plus un maximum de 4 ponts au total.

Une case de trou (niveau -1) peut contenir un Citron par équipe au niveau -1, un maximum de 4 ponts au total, au niveau 0, et également un Citron par équipe au niveau 0, donc au dessus du trou.

Un pont recouvre au moins 2 cases, mais reste un unique Citron avec son id, qui a simplement changé d'état. Les ponts se trouvent entre le niveau -1 et le niveau 0. Les Citrons de niveau 0 se trouvent donc au dessus des éventuels ponts de la case, et les Citrons de niveau -1 se trouvent en dessous.

2.5 Cycle de vie

Les Citrons ne sont vraiment pas pressés : chaque jour, ils ne peuvent faire qu'une seule chose. Une fois cette chose effectuée, ils se reposent jusqu'au lendemain. Quelle que soit son énergie, un Citron est aussi fainéant. Tous les Citrons font donc chacun une action par jour. Cela vaut aussi pour les Citrons qui sont devenus des ponts.

Notez que la durée d'un jour sur cette planète est plutôt courte : elle est de l'ordre du dixième de nos secondes. On peut ainsi les regarder évoluer sans trop s'ennuyer.

2.6 Equipes de Citrons

Chaque Citron fait partie d'une équipe. Vous contrôlez tous les Citrons d'une équipe, votre programme devant décider de ce qu'ils doivent faire chaque jour. Pour ce faire, vous pouvez obtenir diverses informations sur chacun des Citrons de votre équipe, mais aussi sur ceux de l'équipe adverse qui sont visibles par l'un au moins de vos Citrons.

A aucun moment un Citron ne peut changer d'équipe.

2.7 Contrôle d'une case par un Citron

Lorsque des Citrons d'équipes différentes se trouvent sur la même case, au même niveau, c'est celui qui a le plus d'énergie qui a le contrôle de la case. Si deux Citrons ont la même énergie, c'est celui qui est arrivé le plus récemment sur la case qui en a le contrôle.

2.8 Bloquage de Citrons

Tant qu'il reste sur une case, le Citron qui en a le contrôle est le seul à pouvoir agir. Tous les autres sont bloqués, et ne peuvent absolument rien faire. Le jour même où le Citron qui a le contrôle d'une case la quitte, il en perd le contrôle, ce qui veut dire qu'un autre Citron de cette case aura le contrôle ce jour là. S'il y a 4 Citrons sur une case, et qu'ils essayent tous de se déplacer, ils pourront ainsi le faire. Si par contre celui qui a le contrôle ne bouge pas, et que les 3 autres essaient de bouger, rien ne se passera ce jour là.

Lorsqu'un Citron se déplace sur une case de quelque manière que ce soit, il bloque du même coup tout Citron d'énergie inférieure ou égale présent sur cette case, quelle que soit son équipe, et l'empêche donc de faire ce qu'il avait prévu de faire ce jour là. Deux Citrons d'énergie égale peuvent ainsi se bloquer mutuellement.

2.9 Déplacement des Citrons

Les Citrons ne sont pas très mobiles, ils sont incapables de se déplacer sur les montagnes, et ne s'y trouvent donc jamais. Ils peuvent se déplacer sur les plaines, ou dans les trous, case par case, uniquement sur l'une des 4 cases adjacentes à leur case courante, dans la direction ouest-est ou nord-sud (jamais en diagonale).

Un Citron peut tomber dans un trou, mais il ne peut en aucun cas remonter seul d'un trou vers une case de plaine adjacente.

Lorsqu'il est au niveau 0, un Citron se déplace sur un pont exactement comme s'il s'agissait d'une case de plaine. S'il est dans un trou, il peut passer sous le pont.

2.10 Fusion des Citrons

Lorsque deux Citrons de la même équipe se déplacent vers la même case et au même niveau, ou qu'un Citron se déplace vers une case contenant un Citron de son équipe, et qui ne part pas, ils fusionnent automatiquement. Le Citron restant se retrouve avec une énergie égale à la somme des énergies des deux Citrons fusionnés. Ceci ne fonctionne pas pour les ponts.

2.11 Chute des Citrons

Lorsqu'un Citron se déplace d'une case de plaine, ou d'un pont vers une case de trou sans pont, ou bien qu'il se retrouve subitement au dessus du vide, suite à l'écroulement d'un pont, il se retrouve dans le vide, mais reste au niveau 0 pendant un jour. Un Citron dans le vide ne peut rien faire ce jour là. Le jour suivant, il ne pourra rien faire d'autre que tomber, et se retrouvera à la fin de la journée au fond du trou, donc au niveau -1.

Si un Citron se retrouve au dessus du vide à la fin d'un jour, rien ne pourra l'empêcher de tomber, et de se retrouver au fond du trou au jour d'après.

2.12 Sources d'énergie

Une case de plaine (niveau 0) peut contenir une source infinie d'énergie, qui peut fournir 8 (SOURCE_ENERGY_FLOW_PARAM) unités d'énergie par jour à un Citron. Le Citron qui contrôle la case contenant la source absorbe automatiquement l'énergie de la source, chaque jour où il ne fait pas d'action.

Attention, un pont, ou un pont en construction ne peut pas absorber d'énergie d'une source. Une unité se trouvant sur un pont, sur une case contenant une source, peut par contre absorber de l'énergie de la source, malgré la présence du pont.

Le terrain contient exactement 20 sources d'énergie.

2.13 Les actions des Citrons

Chaque jour, un Citron choisit de ne rien faire, ou de faire une, et une seule action. Cette action pourra être réalisée si toutes les conditions sont réunies, comme par exemple le fait de ne pas être bloqué par un autre Citron.

2.13.1 Déplacer un Citron

Un Citron peut choisir de se déplacer simplement vers l'une des 4 cases adjacentes.

2.13.2 Créer/Réduire un pont

Un Citron qui se trouve sur une plaine peut se transformer en pont. Cela consiste à s'allonger dans une direction donnée, sur l'axe ouest-est ou nord-sud, d'une case par jour, éventuellement au dessus d'un trou. Un pont ne peut faire plus de 5 (MAX_BRIDGE_LENGTH_PARAM) cases au total.

La case où se trouvait le Citron au moment où il a commencé à s'allonger constitue l'origine du pont. La position courante représente l'autre extrémité.

Tant que la position courante est au dessus d'un trou, il s'agit d'un pont en construction, sinon c'est un pont complet. Il peut cependant continuer à s'agrandir, et éventuellement redevenir un pont en construction.

On peut réduire un pont en faisant revenir la position courante case par case vers la position d'origine, jusqu'à ce qu'il ne fasse plus qu'une case.

Notez qu'un Citron ne peut pas traverser un trou tout seul, en faisant un pont et en se réduisant de l'autre côté.

On ne peut pas avoir plus de 4 ponts passant par la même case.

2.13.3 Diviser un Citron

Un Citron peut à tout moment se diviser en 2, en créant un nouveau Citron contenant une partie de son énergie, et en le plaçant sur l'une des 4 cases adjacentes.

2.13.4 Sauver un Citron

Un Citron au fond d'un trou (niveau -1) ne peut remonter de lui-même au niveau 0. Par contre, un autre Citron de la même équipe qui se trouve au niveau 0 peut l'aider à remonter, en se plaçant sur une case adjacente, et en le tirant vers lui. Les deux Citrons se retrouvent le jour suivant sur la même case, et fusionnent.

Ceci n'est possible que s'il n'y a pas de pont au dessus du trou, et que le Citron à tirer n'est pas bloqué par un Citron ennemi plus gros.

Un Citron ne peut en sauver un autre que s'il a plus d'énergie que lui.

2.13.5 Presser un Citron

Un Citron peut en presser un autre, s'il est d'une équipe ennemie, selon certaines conditions, pour absorber une partie de son énergie. Pour presser un Citron, on indique une case, qui peut être soit sa propre case, soit l'une des 4 cases adjacentes. Dans tous les cas, le Citron pressé doit se trouver au même niveau que le Citron qui presse.

Un Citron se trouvant au niveau 0, et qui essaie de presser un Citron sur une case qui n'en contient pas, pressera dans ce cas le dernier pont construit sur cette case, si celle-ci en contient.

S'il presse un Citron se trouvant sur sa propre case, la quantité d'énergie absorbée est 12 (CAPT_ENERGY_CITRON_PARAM). Le Citron pressé est celui qui aurait le contrôle de la case, si le Citron qui presse n'était pas là.

S'il presse un Citron se trouvant sur une case adjacente, la quantité d'énergie absorbée est 8 (CAPT_ENERGY_NEXT_CITRON_PARAM). Le Citron pressé est celui qui a le contrôle de la case

adjacente choisie.

Dans les deux cas, l'énergie absorbée est retirée du Citron pressé, pour être ajoutée au Citron qui presse. Si le Citron pressé a moins d'énergie que ce que l'on essaie de lui retirer, son énergie passe à 0, et l'énergie du Citron qui presse n'est augmentée que de ce qui reste.

Notez qu'un Citron ayant le contrôle d'une case sur laquelle se trouvent d'autres Citrons peut presser ceux-ci tranquillement, tous les jours, jusqu'à ce qu'ils meurent, puisqu'ils ne peuvent pas s'échapper.

Notez également qu'un Citron qui en presse un autre qui se trouve sur une case adjacente ne se déplace pas : il reste sur sa case.

2.13.6 Transporter un Citron

Un Citron peut transporter un autre Citron se trouvant sur sa case, en se déplaçant avec lui sur l'une des 4 cases adjacentes. Le Citron déplacé est celui qui aurait le contrôle de la case si le celui qui le transporte n'y était pas.

2.13.7 Ejecter un Citron

Un Citron peut éjecter un autre Citron se trouvant sur sa case, vers l'une des cases adjacentes, sans avoir à se déplacer lui-même. Le Citron éjecté est celui qui aurait le contrôle de la case si le Citron qui l'éjecte n'y était pas.

Le Citron qui éjecte doit bien sûr, comme pour toute action, avoir le Contrôle de sa case le jour où il l'exécute.

2.14 Ordre de jeu chaque jour

Chaque jour, les Citrons des différentes équipes se déplacent dans un ordre donné. Cet ordre est déterminé aléatoirement chaque jour.

La seule conséquence de cet ordre se situe au niveau de l'ordre d'arrivée des Citrons d'énergie égale sur une même case : si deux Citrons ennemis de même énergie arrivent sur une même case le même jour, c'est celui qui arrive le dernier, c'est-à-dire celui dont l'équipe joue la dernière ce jour là qui obtient le contrôle de la case.

2.15 Fonctionnement détaillé des ponts

2.15.1 Déplacement sur des ponts

Un Citron se trouvant au niveau 0 peut se déplacer sur une case adjacente contenant un pont, que ce soit l'origine du pont, sa position courante, ou n'importe-quelle case intermédiaire. Ceci quelle que soit l'équipe à laquelle appartient le pont.

Il peut ensuite se déplacer sur ce pont comme il le ferait sur une case de plaine. Il peut également passer d'un pont à un autre, ou créer un pont partant d'une case du pont.

2.15.2 Charge maximale d'un pont

Le déplacement sur un pont a cependant une limite : le poids total supporté par un pont, c'est-à-dire la somme des énergies des Citrons, normaux ou ponts qui se trouvent dessus ne doit pas dépasser l'énergie du pont lui-même. S'il y a surcharge, le pont s'écroule.

Le poids d'un Citron posé sur un pont est supporté entièrement par ce pont, c'est-à-dire que tout le poids du Citron est ajouté à la charge que le pont supporte. Le poids du Citron est également propagé sur les cases qui soutiennent le pont.

Les cases qui soutiennent un pont sont ses deux extrémités, c'est-à-dire son origine et sa position courante. Chacune de ces deux cases doit supporter la moitié de la charge totale du pont. Si l'une des extrémités du pont se trouve sur un deuxième pont, ce deuxième pont se verra donc ajouter la moitié de la charge du premier à sa propre charge.

Si une case de niveau 0, ou un autre pont se trouve en dessous du pont, juste à l'endroit où se trouve le Citron, tout le poids de ce dernier est supporté par cette case, ou cet autre pont.

2.15.3 Déplacement sur un pont en construction

Si un Citron tente de se déplacer sur un pont en construction, il se déplace, mais le pont s'écroule immédiatement.

Un pont ne peut pas avoir le contrôle d'une case, donc ne peut pas empêcher un Citron de faire une action. Seul le sauvetage d'un Citron peut être bloqué par un pont qui se trouverait au dessus du Citron à sauver.

2.15.4 Ecrroulement d'un pont

Lorsqu'un pont s'écroule, il est retransformé en un Citron normal, qui se retrouve au milieu de l'ancien pont (si la longueur du pont était paire, parmi les 2 cases du milieu, on choisit celle qui est la plus proche de l'origine du pont). Aucune énergie n'est perdue par le Citron lors de l'écroulement.

2.16 Visibilité

Chaque Citron voit une zone de terrain en forme de losange, centrée sur sa position courante. Ce losange contient toutes les cases auxquelles on pourrait accéder en se déplaçant 5 (VIEW_DISTANCE_PARAM) fois ou moins.

Le programme d'un joueur peut ainsi voir toute case se situant dans le losange de visibilité d'un de ses Citrons, même si elle se trouve derrière un obstacle.

Les ponts voient de la même manière, mais uniquement autour de l'extrémité du pont qui peut se déplacer (sa position courante). La case d'origine et les cases intermédiaires ne permettent pas de se voir.

2.17 Energie

Le Citron de départ possède 100 points d'énergie.

Cette énergie peut augmenter :

- si le Citron absorbe de l'énergie d'une source d'énergie
- si le Citron absorbe de l'énergie d'un Citron ennemi
- si un autre Citron de la même équipe fusionne avec lui

Elle peut diminuer :

- si le Citron se trouve dans un trou, d'une unité d'énergie par jour
- si le Citron se fait absorber son énergie par un autre Citron
- si le Citron se divise en deux, et crée un autre Citron

L'énergie d'un Citron ne peut en aucun cas dépasser 200 (MAX_CITRON_ENERGY_PARAM). Si un Citron a atteint cette limite, mais continue d'absorber de l'énergie, que ce soit d'un autre Citron ou d'une source, l'énergie en trop est perdue.

2.18 Disparition d'un Citron

Un Citron peut disparaître dans les cas suivants :

- Son énergie descend à 0, et il meurt : un Citron mort reste sur le terrain pendant quelques jours (voir les constantes), pour se décomposer jusqu'à sa disparition complète. Un Citron mort ne peut absolument rien faire, ni être l'objet d'aucune action.

Un Citron mort disparaît complètement dès qu'un autre Citron de la même équipe se place à la même position (il fusionne avec le cadavre ! beurk !)

Lorsqu'un pont meurt, il se rétracte instantanément.

Un Citron mort est soumis aux lois de la gravité, et tombe donc lorsqu'il est au dessus d'un trou.

- Il fusionne avec un autre Citron.

Lorsqu'un Citron se déplace vers un autre Citron de la même équipe, il fusionne avec lui. Le Citron qui s'est déplacé disparaît, en laissant toute son énergie à l'autre Citron.

2.19 Ordre des traitements

Les actions des Citrons ne sont pas effectuées immédiatement après que vous les ayez choisies. Tout le monde décide au début de la journée de ce qu'il va faire, et elles sont ensuite gérées toutes ensemble.

A titre indicatif, voici le déroulement précis d'une journée.

- Décès des Citrons ayant perdu toute leur énergie la veille.
- Réduction de l'énergie des Citrons se trouvant dans un trou.
- Marquage des Citrons qui sont dans le vide, comme devant tomber à ce jour.
- Ecoulement de tous les ponts qui sont trop chargés.
- Annulation des déplacements des Citrons qui se retrouvent dans le vide.
- Annulation des déplacements impossibles. (Vers un niveau plus élevé)

- Annulation de l'action des Citrons se trouvant sur une case vers laquelle un Citron d'énergie égale ou supérieure se dirige.
- Pressage des Citrons, et transferts d'énergie de tous types.
- Déplacement de toutes sortes des Citrons, et chute des Citrons marqués.

2.20 Début de la partie

Au début de la partie, chaque équipe dispose d'un unique Citron, d'identifiant 0, placé sur une position arbitraire du terrain (sur une plaine), et ayant 100 points d'énergie.

2.21 Fin de la partie

La partie se termine au bout d'une année, c'est-à-dire 242 jours.

2.22 But du jeu

Le gagnant est celui dont l'équipe possède un maximum de cases de plaine à la fin de la partie. Une case de plaine est possédée par une équipe si elle est dans le losange de vision de l'un de ses Citrons (normal ou pont) qui ne soit pas dans un trou, et qu'aucun Citron ennemi n'étant pas dans un trou ne l'ait dans le sien.

En cas d'égalité du nombre de cases possédées par deux équipes, c'est l'équipe dont les Citrons totalisent le maximum d'énergie qui l'emporte. S'il y a toujours égalité, c'est l'équipe qui contient le plus de Citrons qui gagne.

2.23 Résumé des actions possibles d'un Citron

Chaque jour, un Citron peut faire une chose et une seule parmi les suivantes :

- ne rien faire
- se déplacer sur une des 4 cases adjacentes.
Condition : se déplacer vers une case de niveau inférieur ou égal.
- s'étirer, et former un pont, ou le réduire.
Condition : ne pas avoir atteint la longueur maximale de 5 (MAX_BRIDGE_LENGTH_PARAM) cases, se trouver au niveau 0.
- s'écrouler.
Condition : être un pont, terminé ou en cours de construction.
- presser un autre Citron, sur sa case, ou l'une des cases adjacentes.
Condition : on ne peut presser un Citron d'une case adjacente que s'il a le contrôle de sa case.
- capturer et se déplacer avec un Citron de sa case, sur l'une des cases adjacentes.
- lâcher un Citron de sa case, vers une case adjacente.
- sauver un Citron de son équipe, en le tirant d'un trou adjacent vers sa case.
Conditions : être sur une case de plaine, et avoir plus d'énergie que le Citron sauvé.
- créer un nouveau Citron sur une case adjacente, en se divisant en deux.

En plus de toutes les conditions indiquées, pour toute action autre que "ne rien faire", pour pouvoir effectuer une action, il faut que le Citron ait le contrôle de la case sur laquelle il se trouve, le jour où il fait son action.

Tout déplacement se fait d'une case vers une case de niveau inférieur ou égal, sauf dans le cas du sauvetage.

Si une action d'un Citron ne peut pas être entièrement réalisée, elle est complètement annulée.

2.24 Limites de temps et de mémoire

Votre programme ne peut à tout moment utiliser plus de 16 Mo de mémoire, que ce soit sous la forme de variables globales, de pile, ou d'utilisation du tas.

Chaque jour, vous disposez de 0.25 secondes pour faire vos traitements. Si jamais ce temps est dépassé, le temps en trop utilisé est retiré d'une réserve de temps de 60 secondes, dont vous disposez au départ, et qui diminue tous les jours où vous dépassez la limite de temps.

Le total du temps CPU utilisé par votre programme pendant la partie ne pourra ainsi pas dépasser $60 + 0.25 * 242 = 2$ minutes.

Si votre réserve arrive à 0 pendant la partie, votre programme est interrompu, et vos Citrons ne feront plus rien de toute la partie.

Votre programme sera exécuté sur une machine à 500 Mhz. Si jamais ce n'était pas le cas, les limites de temps seraient augmentées en conséquence automatiquement par le serveur.

3 L'API

3.1 L'interface

Votre programme se présentera sous forme d'une librairie, qui communique via un certain nombre de fonctions, définies ci-dessous.

Dans toutes ces fonctions, un Citron sera identifié par :

- un identifiant d'équipe, de 0 au nombre d'équipes moins 1.
- un identifiant de Citron, de 0 au nombre maximum d'unités par équipe (200) moins 1.

L'identifiant de Citron l'identifie parmi les Citrons d'une équipe donnée. Deux Citrons d'équipes différentes peuvent donc avoir le même identifiant de Citron.

A chaque fois qu'un Citron est créé, il se voit attribué le premier identifiant disponible pour son équipe. Cet identifiant ne change pas, mais il peut ne plus être attribué, si le Citron meurt, ou fusionne avec un autre Citron lors d'un déplacement. Il sera alors réattribué plus tard, si un nouveau Citron est créé. Il peut donc y avoir des trous dans les identifiants attribués, mais à aucun moment, un identifiant ne pourra avoir une valeur supérieure ou égale au nombre maximal de Citrons par équipe.

Les prototypes des fonctions sont donnés ici en langage C, à titre indicatif. Vous trouverez en annexe les prototypes des fonctions pour tous les langages, ainsi que la définitions des différentes constantes utilisées.

3.2 Fonctions à écrire

3.2.1 `player_init`

prototype : `void player_init();`

Cette fonction est appelée au début de la partie, avant le premier jour. Elle permet de faire toutes sortes d'initialisations, de lire les zones visibles de la carte, etc.

On ne peut pas faire d'action lors de l'appel de cette fonction (elles seraient ignorées).

3.2.2 `player_new_day`

prototype : `void player_new_day();`

Cette fonction est appelée chaque jour. C'est dans cette fonction que votre client devra faire tout son traitement, et exécuter toutes les actions des Citrons.

3.2.3 `citron_day`

prototype : `void citron_day(int citron_id);`

Cette fonction est appelée chaque jour, pour chacun des Citrons de votre équipe, après l'appel à `player_new_day`. Le numéro du Citron en question est passé en paramètre. Vous pouvez l'utiliser si vous souhaitez calculer une par une les actions de vos Citrons, mais n'êtes pas du tout obligé de l'utiliser, si vous souhaitez calculer toutes les actions dans `player_new_day`. (Ce qui est sans doute préférable).

3.3 Fonctions d'obtention d'informations

3.3.1 `game_get_my_team`

Prototype : `int game_get_my_team()`
Paramètres : aucun
Retour : Un entier positif ou nul identifiant son équipe.
Remarques :

3.3.2 `map_get_size_x`

Prototype : `int map_get_size_x()` ;
Paramètres : aucun
Retour : Un entier indiquant la largeur de la carte.
Remarques : Renvoie toujours 64 lors des matchs officiels.

3.3.3 `map_get_size_y`

Prototype : `int map_get_size_y()` ;
Paramètres : aucun
Retour : Un entier indiquant la hauteur de la carte.
Remarques : Renvoie toujours 64 lors des matchs officiels.

3.3.4 `map_get_nb_teams`

Prototype : `int map_get_nb_teams()` ;
Paramètres : aucun
Retour : Un entier indiquant le nombre d'équipes de la partie.
Remarques : Renvoie 1, 2, ou 4 équipes lors des matchs officiels.

3.3.5 `map_get_max_citrons`

Prototype : `int map_get_max_citrons()` ;
Paramètres : aucun
Retour : Un entier indiquant le nombre de Citrons maximal qu'une équipe peut avoir pendant la partie. Les identifiants des Citrons sont toujours compris entre 0 et cette valeur - 1.
Remarques : Renvoie 200 lors des matchs officiels

3.3.6 `map_is_source`

Prototype : `int map_is_source(int x, int y)` ;
Paramètres : Les coordonnées x et y d'une case.
Retour : Un entier valant 1 ou 0, suivant si la case contient une source d'énergie ou non.
Remarques :

3.3.7 map_get_type

Prototype : `int map_get_type(int x, int y);`
Paramètres : Les coordonnées x et y d'une case.
Retour : Un entier valant -1, 0, ou 1 suivant si la case est un trou, une plaine, ou un obstacle.
Remarques :

3.3.8 map_get_nb_citrons

Prototype : `int map_get_nb_citrons(int x, int y, int z);`
Paramètres : Les coordonnées x, y et z d'une position.
Retour : Un entier représentant le nombre de Citrons présents à cet endroit.
Remarques :

3.3.9 map_get_nb_bridges

Prototype : `int map_get_nb_bridges(int x, int y);`
Paramètres : Les coordonnées x et y d'une case.
Retour : Un entier représentant le nombre de ponts présents sur cette case.
Remarques :

3.3.10 map_get_citron_team

Prototype : `int map_get_citron_team(int x, int y, int z, int count);`
Paramètres : Les coordonnées x, y et z d'une position, et un compteur entier.
Retour : Un entier représentant l'identifiant d'équipe du Citron à cette position. S'il y a plusieurs Citrons à cette position, le compteur indique lequel choisir, de 0 au nombre de Citrons de la case moins 1. Ils sont classés par ordre de contrôle, 0 pour celui qui a le contrôle de la case, puis 1 pour celui qui aurait le contrôle si le 0 partait, etc.
Remarques :

3.3.11 map_get_citron_id

Prototype : `int map_get_citron_id(int x, int y, int z, int count);`
Paramètres : Les coordonnées x, y et z d'une position, et un compteur entier.
Retour : L'identifiant du Citron à cette position et d'indice le compteur, parmi les Citrons sur cette position.
Remarques :

3.3.12 map_get_bridge_team

Prototype : `int map_get_bridge_team(int x, int y, int count);`
Paramètres : Les coordonnées x et y d'une case, et un compteur entier.
Retour : L'identifiant de l'équipe du pont à cette position, et d'indice le compteur. L'ordre est celui de création des ponts, l'indice 0 correspond au pont présent sur la case depuis le plus longtemps.
Remarques :

3.3.13 map_get_bridge_id

Prototype : `int map_get_bridge_id(int x, int y, int count);`
Paramètres : Les coordonnées x et y d'une case, et un compteur entier.
Retour : L'identifiant du pont à cette position, et d'indice le compteur.
Remarques :

3.3.14 map_get_next_citron_id

Prototype : `int map_get_next_citron_id(int team, int citron_id);`
Paramètres : Un entier identifiant l'équipe, et un entier identifiant un Citron de cette équipe.
Retour : Un entier représentant le premier identifiant de Citron valide qui soit strictement supérieur à celui passé en paramètre. S'il n'y a plus de Citron ayant un id supérieur à l'id passé en paramètre, cette fonction retourne -1.
Remarques :

3.3.15 citron_get_state

Prototype : `int citron_get_state(int team_id, int citron_id);`
Paramètres : Un entier identifiant l'équipe, et un entier identifiant un Citron de cette équipe.
Retour : Un entier représentant l'état du Citron. (voir les constantes dont le nom se termine par "STATE").
Remarques :

3.3.16 citron_get_energy

Prototype : `int citron_get_energy(int team_id, int citron_id);`
Paramètres : Un entier identifiant l'équipe, et un entier identifiant un Citron de cette équipe.
Retour : Un entier représentant l'énergie du Citron.
Remarques :

3.3.17 citron_get_cur_x

Prototype : `int citron_get_cur_x(int team_id, int citron_id);`
Paramètres : Un entier identifiant l'équipe, et un entier identifiant un Citron de cette équipe
Retour : Un entier représentant la position en x du Citron
Remarques :

3.3.18 citron_get_cur_y

Prototype : `int citron_get_cur_y(int team_id, int citron_id);`
Paramètres : Un entier identifiant l'équipe, et un entier identifiant un Citron de cette équipe.
Retour : Un entier représentant la position en y du Citron.
Remarques :

3.3.19 citron_get_cur_z

Prototype : `int citron_get_cur_z(int team_id, int citron_id);`
Paramètres : Un entier identifiant l'équipe, et un entier identifiant un Citron de cette équipe.
Retour : Un entier représentant le niveau (z) du Citron.
Remarques :

3.3.20 citron_get_origin_x

Prototype : `int citron_get_origin_x(int team_id, int citron_id);`
Paramètres : Un entier identifiant l'équipe, et un entier identifiant un Citron de cette équipe.
Retour : Un entier représentant la coordonnée x de l'extrémité d'origine du pont.
Remarques :

3.3.21 citron_get_origin_y

Prototype : `int citron_get_origin_y(int team_id, int citron_id);`
Paramètres : Un entier identifiant l'équipe, et un entier identifiant un Citron de cette équipe.
Retour : Un entier représentant la coordonnée y de l'extrémité d'origine du pont.
Remarques :

3.3.22 citron_get_action_result

Prototype : `int citron_get_action_result(int team_id, int citron_id);`
Paramètres : Un entier identifiant l'équipe, et un entier identifiant un Citron de cette équipe.
Retour : un entier identifiant le résultat de la dernière action du Citron. (voir la liste des constantes dont le nom se termine par "RESULT").
Remarques :

3.4 Fonctions permettant d'effectuer des actions

Vous pouvez appeler l'une de ces fonctions, au choix, une fois pour chacun de vos Citrons, chaque jour. Si jamais vous appelez deux fonctions d'action pour un même Citron, seule la dernière action serait prise en compte. Mais en appeler trop peut poser un problème (voir le retour de ces fonctions, plus bas).

3.4.1 move_citron

Prototype : `int move_citron(int citronId, int x, int y);`
Paramètres : Un entier identifiant le Citron, et deux entiers x et y identifiant la case vers laquelle il se déplace.
Retour : ORDER_OK ou un code d'erreur
Remarques : Permet de déplacer un Citron.

3.4.2 stretch_citron

Prototype : `int stretch_citron(int citronId, int x, int y) ;`
Paramètres : Un entier identifiant le Citron, et deux entiers x et y identifiant la nouvelle case vers laquelle doit se trouver l'extrémité du pont.
Retour : ORDER_OK ou un code d'erreur
Remarques : Permet de former un pont, en étirant un Citron ou un pont existant d'une case. Permet également de réduire un pont existant en le raccourcissant d'une case. La case spécifiée doit dans tous les cas être l'une des 4 cases adjacentes à la position courante, et être dans l'axe du pont.

3.4.3 retract_citron

Prototype : `int retract_citron(int citronId) ;`
Paramètres : Un entier identifiant le Citron (pont) que l'on veut faire écrouler.
Retour : ORDER_OK ou un code d'erreur
Remarques : Permet de faire s'écrouler un pont, pour le retransformer en Citron.

3.4.4 press_citron

Prototype : `int press_citron(int citronId, int x, int y) ;`
Paramètres : Un entier identifiant le Citron qui presse, et deux entiers x et y définissant la case où se trouve le Citron pressé.
Retour : ORDER_OK ou un code d'erreur
Remarques : Permet d'absorber de l'énergie d'un autre Citron, sur la même case, ou sur une case adjacente.

3.4.5 capt_and_move_citron

Prototype : `int capt_and_move_citron(int citron_id, int x, int y) ;`
Paramètres : Un entier identifiant le Citron qui fait la capture, et deux entiers x et y définissant la case adjacente vers laquelle on veut se déplacer avec le Citron capturé.
Retour :
Remarques : Permet de se déplacer tout en déplaçant un Citron ennemi, se trouvant sur la même case.

3.4.6 drop_citron

Prototype : `int drop_citron(int citron_id, int x, int y) ;`
Paramètres : Un entier identifiant le Citron qui fait l'action, et deux entiers x et y définissant la case adjacente vers laquelle le Citron ennemi est expulsé.
Retour : ORDER_OK ou un code d'erreur
Remarques : Permet d'éjecter un Citron ennemi se trouvant sur la même case, vers une case adjacente.

3.4.7 pull_citron

Prototype : `int pull_citron(int citron_id, int x, int y) ;`
Paramètres : Un entier identifiant le Citron qui fait le sauvetage, et deux entiers x et y définissant la case trou adjacente où se trouve le Citron capturé.
Retour : ORDER_OK ou un code d'erreur
Remarques : Permet de sauver un Citron ami se trouvant dans une case trou adjacente.

3.4.8 split_citron

Prototype : `int split_citron(int citron_id, int x, int y, int energy) ;`
Paramètres : Un entier indiquant le Citron qui fait la fission, deux entiers x et y indiquant les coordonnées de la case adjacente sur laquelle se déplace le Citron d'origine après la division, et un entier spécifiant l'énergie qui est attribuée au nouveau Citron.
Retour : ORDER_OK ou un code d'erreur
Remarques : Permet de diviser un Citron, et de créer ainsi un nouveau Citron, avec une partie de son énergie. Le Citron créé reste sur la case de départ, l'autre se déplace pendant la fission vers une case adjacente.

Toutes ces fonctions renvoient une valeur parmi les constantes suivantes :

- ORDER_OK : l'ordre a bien été transmis au serveur, ce qui ne veut pas forcément dire qu'il sera effectué avec succès.
- TOO_MANY_ORDERS_ERROR : vous avez transmis beaucoup trop d'ordres ! Vous ne devriez pourtant pas en envoyer plus d'un par Citron.
- INVALID_ARGS_ERROR : les paramètres de votre ordre sont invalides.
- INVISIBLE_POSITION_ERROR : la position que vous indiquez n'est visible par aucun des Citrons, il doit y avoir un problème...
- NON_EXISTENT_CITRON_ERROR : le citron que vous spécifiez n'existe pas.
- INVALID_POSITION_ERROR : la position que vous spécifiez n'est pas dans la carte.

Toutes ces erreurs signifient qu'il y a un problème dans votre programme. Un programme valide devrait toujours obtenir ORDER_OK comme résultat, lors de l'appel de ces fonctions.

Pour savoir si une action a été exécutée correctement, vous devez utiliser la fonction `citron_get_action_result`. Elle renvoie une valeur parmi toutes les constantes se terminant par `_RESULT`. Attention, il ne faut pas l'appeler juste après avoir fait l'action, mais au jour suivant. Les actions sont en effet exécutées tout à la fin de la journée, une fois que vous avez terminé tous vos calculs.

Au début d'un jour, vous pouvez donc appeler `citron_get_action_result` pour chacun de vos Citrons, pour vérifier s'ils ont réussi ce qu'ils essayaient de faire. Toute valeur différente de `SUCCESS_RESULT` signifie que l'action a échoué. Les autres valeurs vous donnent une idée de ce qu'il s'est passé.

3.5 Autres fonctions utiles

3.5.1 print_map

Prototype : `int print_map(nt result) ;`

Paramètres :

Retour : aucun

Remarques : Cette fonction permet d'afficher le contenu de l'ensemble de la carte sur la sortie standard, y compris les zones non visibles par votre équipe. Cela peut vous aider à déboguer, mais ne peut pas aider votre programme à obtenir des informations.

Sur la carte affichée, '.' représente une plaine, '-' représente un trou, '+' représente une montagne. '*' représente une source d'énergie. 'a' à 'd' représentent des Citrons des équipes 0 à 3, respectivement. Ces lettres sont en majuscules si une autre unité se trouve en dessous. (La lettre affichée correspond ainsi à l'équipe du Citron ayant le contrôle de la case).

Les chiffres de 0 à 3 représentent des cases contenant un pont, de l'équipe ayant ce numéro. L'origine et la position courante du pont sont indiqués par '@'. La carte est vue de dessus, donc un pont cache les éventuelles unités qui sont en dessous, et un Citron au niveau 0 cache les éventuels ponts se trouvant en dessous.

Pour les éventuels petits malins qui essaieraient de détourner la sortie standard pour récupérer les infos sur toute la map par ce moyen, oui, ok ça peut marcher, mais cette fonction sera désactivée pendant les tournois pour faire le classement, donc ne comptez pas dessus !

D'une manière générale, ils vous est interdit d'appeler des fonctions systèmes, du style fork, kill, etc...

3.5.2 time_left

Prototype : `int time_left() ;`

Paramètres : aucun

Retour : un entier indiquant le nombre de milli-secondes qu'il vous reste dans la journée pour faire vos traitements, avant d'être obligé de puiser dans la réserve de temps.

Remarques : Cette fonction renvoie 0 si vous êtes déjà en train de puiser dans la réserve.

3.5.3 time_reserve_left

Prototype : `int time_reserve_left() ;`

Paramètres : aucun

Retour : un entier indiquant le nombre de milli-secondes qu'il vous reste dans la réserve de temps supplémentaire, et que vous pourrez utiliser avant la fin de la partie.

Remarques :

3.5.4 print_action_result

Prototype : `int print_action_result(int result) ;`

Paramètres : un entier spécifiant un résultat d'action, retourné par la fonction `citron_get_action_result`

Retour : aucun

Remarques : Cette fonction permet d'afficher en français la signification du résultat fourni en paramètre.

4 Conclusion

Il est conseillé aux candidats de bien veiller à s'assurer de bien comprendre le sujet avant de commencer à coder quoi que ce soit, ou même de rechercher une stratégie de jeu. (En fait ce n'est pas si important, de toute façon il y aura un vainqueur, si ce n'est pas vous, ça sera un autre ...) Après avoir lu et relu le sujet, si vous ne comprenez pas bien le sujet n'hésitez pas à poser vos questions à l'équipe serveur (badges jaunes). (de toute manière, si celui-ci n'est pas clair, c'est de la faute de Glop!)

Que le plus chanceux gagne

Plus on rate, plus on a de chances de réussir

42