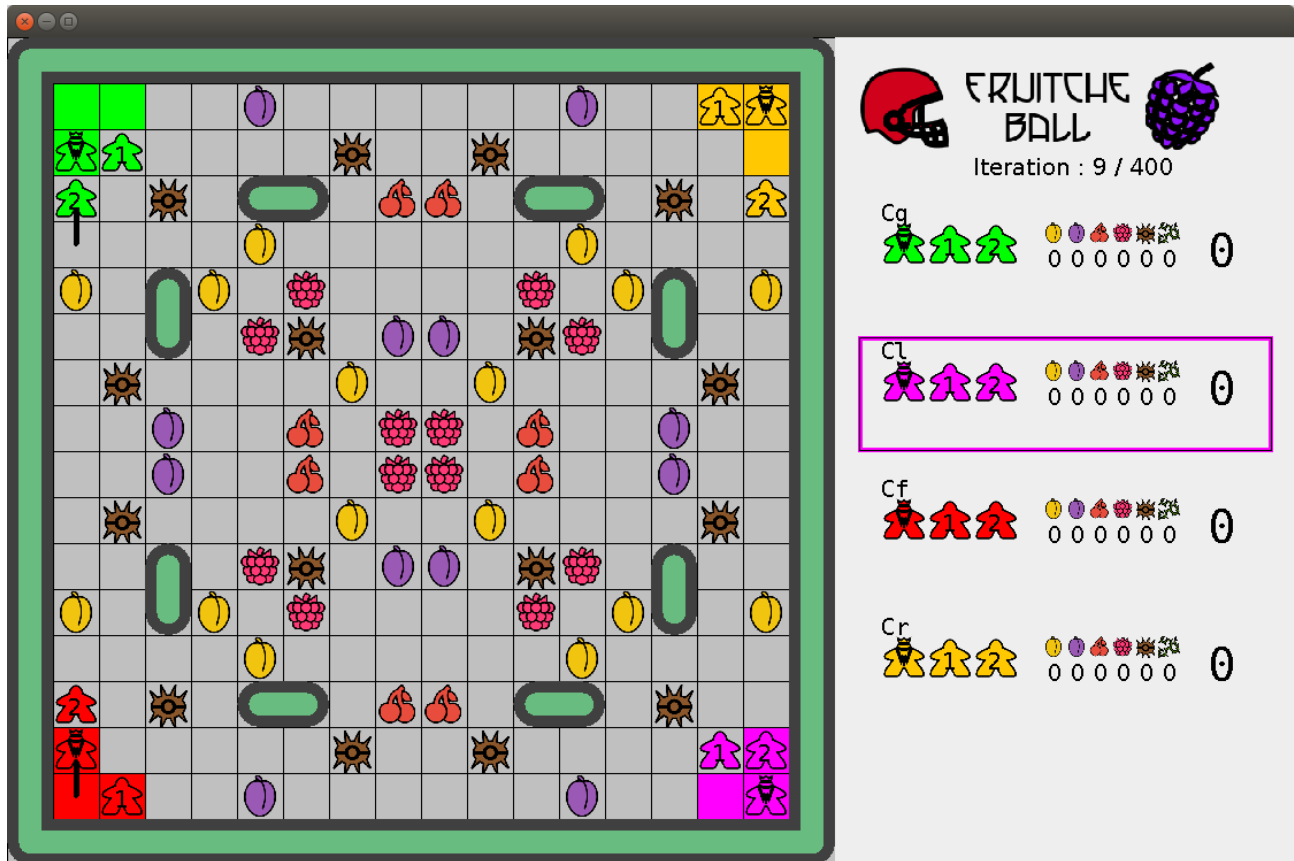


24h des IUT Informatique

Fruitcheball

25-26 Mai 2018 @ Nancy



UNIVERSITÉ
DE LORRAINE



nancy Charlemagne
Département Informatique

Table des matières

1	Présentation générale	3
2	Déroulement du concours	4
2.1	Organisation du tournoi	4
2.2	Développement des Intelligences Artificielles	4
2.3	Tournoi et évaluation des clients	5
3	Règles du jeu	6
3.1	Organisation du jeu	6
3.2	Structure de l'environnement	6
3.3	Constitution d'une équipe	7
3.4	Les actions possibles	8
3.4.1	Déplacement	8
3.4.2	Prise, pose ou échange d'un fruit	8
3.4.3	Envoi de fruit	9
3.4.4	Les châtaignes	10
3.5	Décompte des points	10
3.6	Éléments de stratégie	10
4	Protocole de communication avec le serveur	12
4.1	Organisation des échanges	12
4.2	Messages envoyés par le serveur	13
4.2.1	Descriptif de la carte	14
4.2.2	Descriptif des équipes	14
4.3	Messages attendus du client	15
5	Lancement de l'application	16
5.1	Lancement du serveur	16
5.2	Connexion des clients	16
5.3	Options du serveur	17
5.4	Version interactive	18
6	Résumé du sujet	19

1 Présentation générale

L'épreuve « Développement d'application et intelligence artificielle » des 24h des IUT informatique 2018 est dédiée au jeu fictif du **Fruitcheball**¹.



La présentation de tonton Charlemagne :

Lorsque l'été arrive et que les fruits commencent à faire ployer les branches des différents arbres fruitiers de la région, des équipes s'organisent pour le grand tournoi de **Fruitcheball**. C'est à travers une application simulant ce tournoi que nous vous proposons de vous initier au grand sport (fictif) qui mériterait d'être connu qu'est le Fruitcheball.

Le jeu voit s'affronter simultanément 4 équipes au sein d'un verger parsemé de fruits. L'objectif de chaque équipe est de ramener le plus de fruits sur sa zone de départ afin de pouvoir confectionner les plus belles tartes. Pour espérer gagner, les équipes devront utiliser au mieux les actions à disposition :

- lancer des fruits pour accélérer leur transport ;
- mais aussi intercepter les fruits des autres ;
- et lancer les châtaignes présentes sur le terrain pour perturber les autres équipes.

Chaque équipe est constituée de trois membres (aussi appelés *personnages*) : deux **Lanceurs** et un **Quetcherback**. Le **Quetcherback** possède un matériel spécifique et est le seul qui est capable de ramasser et de lancer des châtaignes sur ses adversaires. Enfin, le verger est constitué de clôtures qui constituent des obstacles infranchissables pour les personnages et les lanceurs de fruits.

Le jeu est entièrement contrôlé par une **application serveur fournie fruitcheball.jar** et les échanges se font à travers le réseau. Chaque groupe d'étudiants doit développer un client décidant les actions des trois membres de son équipe de **Fruitcheball** à chaque tour de jeu.

Le jeu se déroule **de manière séquentielle**. Une fois que tous les clients sont connectés, le serveur contacte le premier client, lui transmet le descriptif du jeu, lui demande les actions des membres de l'équipe et les exécute. Une fois que ces actions sont faites, il contacte le client suivant, lui transmet la nouvelle situation, lui demande les actions à effectuer, etc...

À chaque tour de jeu, le client doit transmettre **les actions des trois membres de son équipe**. L'action d'un personnage peut consister à **(1) se déplacer d'une case** dans une direction (Nord, Sud, Est ou Ouest) ou rester sur place, **(2) prendre ou poser le fruit** situé sur sa position ou **(3) lancer un fruit possédé** dans une direction donnée (à une distance de 4 cases). Si les fruits lancés traversent la case d'un autre personnage, ce personnage réussit automatiquement à intercepter le fruit.

Un match prend fin au bout de 400 tours de jeu (100 tours de jeu par client) et l'équipe qui a accumulé le plus de points remporte le match. **Les points marqués par une équipe** dépendent des fruits rapportés dans sa zone de départ. Chaque type de fruit rapporte des points en fonction du nombre de fruits rapportés : le premier fruit d'un type rapporte 1 point, le deuxième fruit du même type rapporte 2 points (pour un total de 3), le troisième fruit du même type rapporte 3 points (pour un total de 6), etc...

1. Note : toute ressemblance avec un jeu existant ou ayant existé serait purement fortuite.²

2. Signé le comité de réhabilitation des notes de bas de page.

2 Déroulement du concours

2.1 Organisation du tournoi

Au cours des huit heures que dure la compétition, vous devez programmer une intelligence artificielle contrôlant les actions des membres d'une équipe de **Fruitcheball**.

Pour 22h, vous devrez déposer l'intégralité de votre client (code source + ressources utiles + version compilées + un README expliquant comment compiler et lancer votre client) sur un serveur FTP dont l'adresse vous sera fournie. Une fois passé ce délai, aucune modification du client ne sera autorisée sous peine de disqualification.

À partir de 22h, les tournois se dérouleront au fil de l'eau dans une salle dédiée. En cas de soucis lors du tournoi, il est possible qu'un organisateur vous sollicite pour lancer votre client à partir de votre machine. Pour que cela puisse se faire correctement, il faut :

1. que vous **disposiez d'une machine** facilement transportable³ à partir de laquelle vous pourrez lancer votre client,
2. que vous **désigniez un membre** par équipe qui pourra se rendre rapidement disponible en soirée pour répondre aux questions concernant votre application et pour venir lancer le client en cas de nécessité.

2.2 Développement des Intelligences Artificielles

Pour développer votre client, prenez le temps de comprendre les règles du jeu détaillées dans ce document (cf. section 3) et la manière dont la communication se fait avec le serveur (cf. section 4).

Votre client peut être développé dans le langage de votre choix. Il faut cependant que votre client soit capable de communiquer avec le serveur via des sockets et que vous disposiez d'une machine permettant de lancer le client lors des tournois effectués en soirée. Les langages supportés par défaut sont C, C++, python, JAVA, C# et javascript sous node.js.

Vos clients doivent se lancer **en utilisant deux options via la ligne de commande** (cf. le client exemple fourni). La première option correspond à l'adresse IP du serveur, la seconde au port utilisé. Vous devrez fournir avec votre client un fichier **README.txt** qui explique comment compiler et lancer votre client.

IMPORTANT ! Comme indiqué, vous pouvez décider d'utiliser un autre langage mais, dans ce cas, il faudra prévenir l'équipe enseignante en début d'épreuve pour qu'elle puisse installer ce qu'il faut pour lancer votre client lors des tournois.



Le conseil avisé de tonton Charlemagne :

Avant de vous lancer dans une stratégie complexe, prenez soin de vérifier que votre client parvienne à communiquer avec le serveur de jeu et que vous interprétez convenablement les informations qu'il vous transmet !

3. Par exemple, cela ne devra pas être le serveur de votre application web.

2.3 Tournoi et évaluation des clients

Le tournoi qui déterminera le classement pour cette épreuve sera organisé en deux étapes : (a) une étape de qualification suivie (b) du tournoi lui-même.

L'**étape de qualification** a pour objectif de vérifier que les clients parviennent bien à se connecter et à échanger avec le serveur. Seules les équipes pour lesquelles ces échanges seront corrects participeront au tournoi.

Le tournoi s'organise ensuite par **poules tirées au hasard parmi les clients qualifiés**. Chaque poule sera constituée d'entre 8 et 13 clients. Ces clients cumuleront des points en affrontant les autres clients de la poule au cours de matchs créés au hasard. Les clients **les mieux classés dans chaque poule** seront conservés pour constituer de nouvelles poules conformément à la figure 1. Ils graviront ainsi les échelons du classement au fur et à mesure des parties jusqu'à déterminer les 4 clients classés pour la finale.

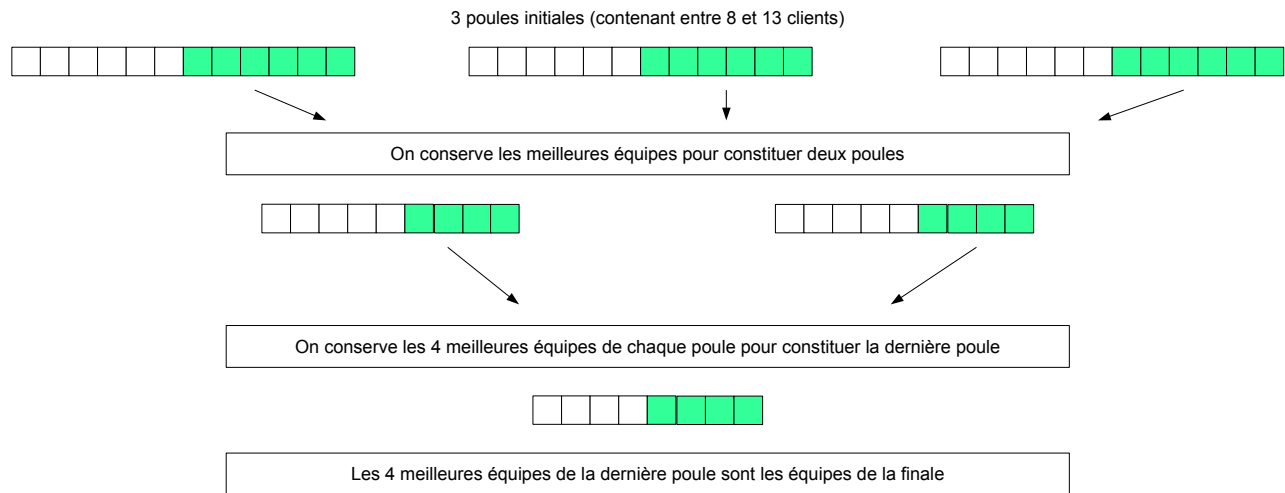


FIGURE 1 – Principe de sélection des équipes lors du tournoi.

La finale du tournoi déterminera le classement sur le podium. Elle sera enregistrée et diffusée devant l'ensemble des participants lors de la cérémonie de remise des prix.

À l'issue du classement, les équipes étant classées entre la première et la seizième position recevront **des points pour le classement général** conformément au règlement du concours.

3 Règles du jeu

Cette partie décrit en détail les règles du jeu qui sont rappelées en fin de sujet dans la section 6.

3.1 Organisation du jeu

Le jeu se déroule **de manière séquentielle, au tour par tour**. Lorsque c'est à son tour, le client reçoit le descriptif de la situation de jeu de la part du serveur et dispose de 3 secondes pour retourner une chaîne de caractères décrivant les actions des différents membres de son équipe. Dès que le serveur reçoit la réponse du joueur, il exécute ses actions, met à jour l'état du jeu et contacte le client suivant.

Une partie comprend **100 tours par joueur**. Pour une partie à 4 joueurs, le serveur s'arrête après 400 tours et les scores finaux pour cette partie correspondent aux points de chaque équipe.

Attention : si votre client ne répond pas assez vite ou si plusieurs ordres sont donnés entre 2 tours, votre client sera disqualifié pour le reste de la partie (pensez à attendre le descriptif du serveur avant d'envoyer un ordre).

3.2 Structure de l'environnement

L'environnement est représenté par un damier de cases sur lesquelles les personnages vont se déplacer. Pour éviter toute ambiguïté, les cases sont numérotées selon deux axes x et y dont l'origine se situe en haut à gauche de l'écran. Les indices débutent à l'indice 0 comme indiqué dans la figure 2.

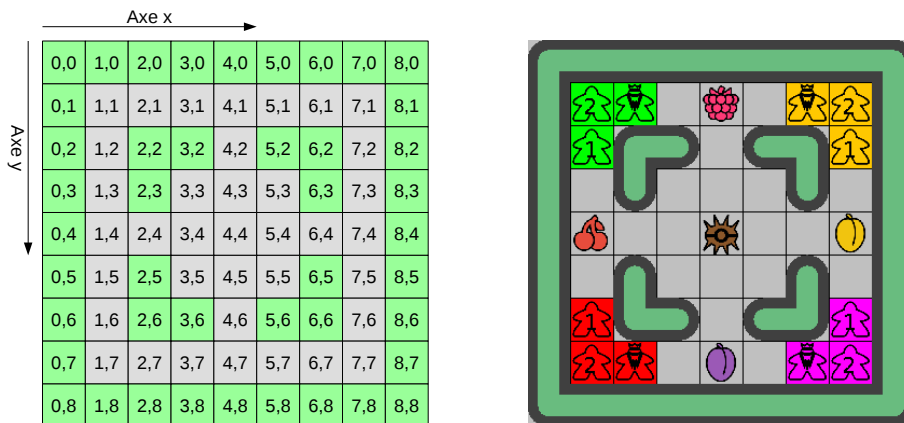


FIGURE 2 – Présentation des indices des cases de l'environnement (sur le labyrinthe mini exemple). Dans cet exemple, la framboise se trouve donc sur la case (4,1) et la cerise sur la case (1,4).

Les clôtures représentées en vert sur l'interface graphique constituent des obstacles infranchissables que ce soit par les personnages ou par les fruits. Tous les vergers utilisés disposent d'une clôture faisant le tour du verger comme présenté figure 2.



Le conseil avisé de tonton Charlemagne :

Les structures des clôtures des exemples et des environnements utilisés pour les tournois restent volontairement simples (par exemple sans cul-de-sac).

Les équipes débutent à partir de leur zone de départ constituées de 3 cases **toujours situées à un des coins du plateau**. Ces cases sont de la couleur de l'équipe correspondante dans l'interface graphique. Dans l'exemple de la figure 2, la zone de l'équipe verte est constituée des cases (2,1) (1,2) et (1,1). Dès qu'un fruit arrive sur le sol d'une de ces cases (qu'il y soit posé ou lancé), il est considéré comme ayant été rapporté par l'équipe à qui la zone appartient⁴.

Les fruits sont disposés dans le verger en fonction du verger sélectionné (cf. section 5 concernant le lancement de l'application). Une fois la partie lancée, il n'y a pas d'apparition de nouveaux fruits dans le verger (la tempête estivale de la veille les a déjà tous fait tomber). Les fruits sont de différents types et le nombre de points rapportés par un fruit dépend uniquement du nombre de fruits de ce type déjà rapportés par l'équipe.



Le conseil culinaire de tonton Charlemagne :

Pour confectionner les plus belles tartes lorraines, il vaut mieux, en effet, se limiter à un seul type de fruits (en Lorraine, de préférence jaune et issu du mirabellier).

3.3 Constitution d'une équipe

Une équipe est toujours constituée de trois membres :

- deux **Lanceurs** qui se déplacent dans l'environnement, peuvent récolter, lancer et intercepter des fruits ;
- un **Quetcherback** qui peut faire tout ce qu'un lanceur peut faire et qui dispose en plus d'un équipement spécifique lui permettant de ramasser les châtaignes et de les envoyer sur les adversaires.

Ces éléments sont représentés graphiquement de la manière suivante⁵ :

		
Quetcherback	Lanceur 1	Lanceur 2

À chaque tour de jeu, le client doit transmettre les actions des trois membres de l'équipe. Ces actions doivent toujours être fournies dans le même ordre : l'action du **Quetcherback** suivie des actions des deux **Lanceurs** (les lanceurs disposent de numéro pour les identifier).

Les actions sont ensuite exécutées de manière séquentielle : d'abord celle du **Quetcherback** puis celle du **Lanceur 1** et enfin celle du **Lanceur 2**.

4. Oui, même si c'est un adversaire qui l'y a amené, il arrive régulièrement que certains membres d'une équipe galvanisés par ce sport se trompent de camp.

5. La plupart des images utilisées pour l'interface graphique proviennent du site <http://game-icons.net/>

3.4 Les actions possibles

À chaque tour, chaque membre de l'équipe concernée doit effectuer une action parmi les actions disponibles. Cette action peut être :

1. soit un déplacement d'une case dans une direction donnée ;
2. soit une action de prise, de pose ou d'échange d'un fruit sur sa case ;
3. soit un lancer du fruit possédé dans une direction donnée ;
4. soit l'*action par défaut* consistant à ne rien faire.

Les actions exécutées sont représentées sur l'interface graphique soit par un trait plein lorsqu'il s'agit d'un déplacement soit par un trait en pointillé lorsqu'il s'agit d'un lancer de fruit.

3.4.1 Déplacement

Il existe 4 actions de déplacement différentes selon la direction cardinale suivie. Chaque action est représentée par une chaîne différente :

1. la chaîne **N** pour Nord ;
2. la chaîne **S** pour Sud ;
3. la chaîne **E** pour Est ;
4. la chaîne **O** pour Ouest ;
5. une action spécifique nommée **X** permet de rester sur place (action équivalente à ne rien faire).

Le déplacement consiste à aller sur la case adjacente dans la direction souhaitée. Si la case d'arrivée est un mur ou contient déjà un personnage, le déplacement échoue. Dans tous les autres cas, le déplacement réussit et le membre de l'équipe se déplace sur la case correspondante.

Règle d'or 1 : deux personnages ne peuvent jamais se trouver sur la même case !

3.4.2 Prise, pose ou échange d'un fruit

Chaque membre de l'équipe **dispose d'un inventaire** de taille limitée qui ne peut contenir qu'un seul fruit (ou chataigne). Une action particulière représentée par la chaîne **P** permet de prendre ou poser un fruit. Lorsque cette action est effectuée, le personnage échange le fruit qu'il possède dans son inventaire avec le fruit présent sur sa case.

De manière détaillée, pour cette action :

- si le personnage ne possède pas de fruit et que sa case est vide, rien ne se passe ;
- si le personnage ne possède pas de fruit et que sa case contient un fruit, le personnage ajoute le fruit dans son inventaire et le fruit disparaît de la carte ;
- si le personnage possède un fruit et que sa case est vide, le personnage dépose le fruit de son inventaire sur sa case ;
- si le personnage possède un fruit et que la case en contient un, le personnage échange son fruit avec celui présent sur sa case.

Des règles spécifiques s'appliquent dans deux cas particuliers :

1. si le fruit est déposé sur une zone de départ, il est comptabilisé comme un fruit rapporté par l'équipe. Un fruit déposé dans une zone de départ disparaît ensuite de la carte et n'est plus jamais accessible ;

- seul le **Quetcherback** peut manipuler des châtaignes. Si un **Lanceur** souhaite en prendre une, l'action échoue et le lanceur perd des points comme s'il avait été touché par un envoi de châtaigne.

Règle d'or 2 : deux fruits ne peuvent jamais être posés sur la même case ! La seule situation où deux fruits se situent sur une case est lorsqu'un fruit se trouve sur la case et qu'un second fruit est possédé par un joueur situé sur cette case.

Règle d'or 3 : un personnage ne peut porter qu'un seul fruit. Seul un **Quetcherback** peut porter une châtaigne.

3.4.3 Envoi de fruit

Afin d'accélérer le ramassage de fruits, il est possible de lancer le fruit de son inventaire dans une direction (Nord, Sud, Est ou Ouest). Le fruit est alors envoyé dans la direction donnée à une distance de 4 cases maximum.

Les chaines correspondant à ces actions sont constituées de deux caractères, un 'L' pour lancer suivi de la direction du lancer, à savoir :

- **LN** pour lancer vers le Nord ;
- **LS** pour lancer vers le Sud ;
- **LE** pour lancer vers l'Est ;
- **LO** pour lancer vers l'Ouest.

Les règles suivantes s'appliquent pour un lancer :

- par défaut, le fruit lancé se déplace de 4 cases et atterrit sur une case vide (sans fruit) s'il ne rencontre ni obstacle ni personnage ;
- si le fruit heurte un obstacle comme une clôture, il s'arrête sur la case précédente ;
- si un personnage (qu'il soit de la même équipe ou non) se trouve sur la trajectoire du fruit, il intercepte automatiquement le fruit lancé.
 - si l'inventaire de ce personnage est vide, il prend le fruit et le place dans son inventaire ;
 - si son inventaire est plein, le fruit est tout de même bloqué et tombe au sol sur sa case.

Dans tous les cas, quand un fruit doit arriver sur une case et qu'un autre fruit y est déjà présent, le fruit rebondit sur une case adjacente (choisie de manière aléatoire). Si cette case est à nouveau occupée, le fruit continue de rebondir jusqu'à ce qu'il arrive sur une case vide.

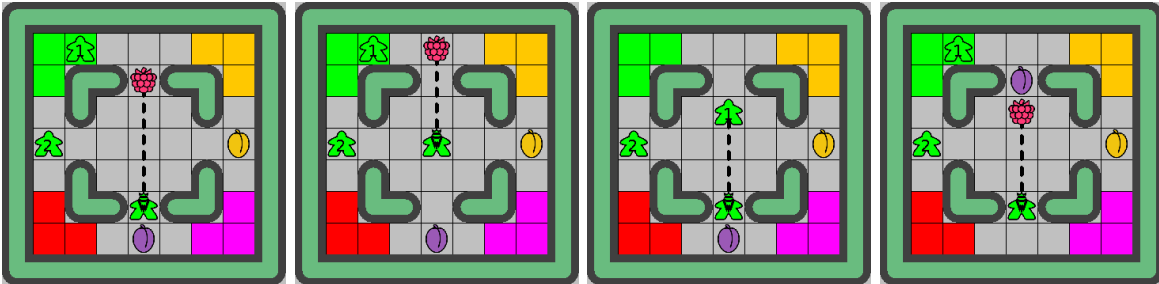


FIGURE 3 – (a) Envoi vers le Nord d'une framboise par le **Quetcherback** sans obstacle, (b) envoi vers le Nord avec un obstacle clôture, (c) envoi vers le Nord avec interception par un autre personnage (ici le **Lanceur** 1 de la même équipe) et (d) envoi vers le Nord sur une case qui contient déjà un fruit et rebond sur une case adjacente.

Si le fruit arrive sur le sol d'une zone de départ, il disparaît et il est considéré comme rapporté par l'équipe à qui la zone appartient.

3.4.4 Les châtaignes

La châtaigne est un fruit très particulier : elle peut servir à faire des tartes mais se révèle aussi très utile pour perturber les autres équipes. Cela nécessite cependant une combinaison adaptée.

Seul le **Quetcherback** peut prendre une châtaigne et la lancer. Lorsqu'une châtaigne est prise par un **Quetcherback**, elle occupe l'emplacement de l'inventaire comme les autres fruits.

Si une châtaigne arrive sur une **zone de départ**, elle est considérée comme étant rapportée mais elle occasionne des **points négatifs**, calculés selon la même logique que pour les autres types de fruits, à savoir 1 point pour la première, 2 points pour la deuxième, etc...

Si une **châtaigne est lancée**, les règles restent valides, mais si la châtaigne touche un **Lanceur** dans sa trajectoire, l'équipe du **Lanceur** touché perd des points comme si la châtaigne avait été déposée dans sa zone de départ. La châtaigne disparaît alors du terrain de jeu et le **Lanceur** fait tomber le fruit qu'il possédait (qui peut rebondir si la case où il se trouve possède déjà un fruit). Si la châtaigne touche un **Quetcherback**, comme il dispose du bon équipement, il peut l'intercepter comme tout autre fruit.

3.5 Décompte des points

Le décompte des points correspond aux fruits ramenés dans la zone de départ. Pour chaque type de fruits, le nombre de points récupérés correspond à la somme des points rapportés par chaque fruit pour un total de $1 + 2 + 3 + 4 + 5 + \dots + n$ si n fruits de ce type ont été ramenés⁶.

Concernant les châtaignes, on considère qu'un fruit est ramené s'il est déposé dans la zone de départ ou s'il a touché un personnage de son équipe. Les points correspondant sont ensuite calculés de la même manière mais comptabilisés **en négatif**.

Par exemple :

Points positifs			Points négatifs		
5 framboises ramenées	$1 + 2 + 3 + 4 + 5 =$	+15	2 châtaignes déposées	$1 + 2 =$	-3
0 mirabelle ramenée	$0 =$	+0	+ 3 châtaignes reçues	$3 + 4 + 5 =$	-12
3 prunes ramenées	$1 + 2 + 3 =$	+6			
1 cerise ramenée	$1 =$	+1			
Total : +7 =			+22 -15		

3.6 Éléments de stratégie

Le jeu ouvre volontairement à un grand nombre de stratégies très différentes. Afin de ne pas vous disperser, nous vous **conseillons fortement** :

6. Pour rappel, la somme d'une suite arithmétique $\sum_{i=1}^n i = \frac{n \cdot (n+1)}{2}$.

1. de commencer par des stratégies élémentaires (par ex. déplacements aléatoires, puis chercher les fruits et les ramener en marchant, ...);
2. de développer de nouveaux clients en intégrant progressivement de nouveaux éléments dans votre stratégie (par ex. utiliser les lancers, cibler les fruits, ...);
3. d'utiliser les premières stratégies que vous aurez codées comme adversaires pour évaluer vos stratégies plus évoluées;
4. de ne pas oublier que la structure du verger est simple en terme d'obstacles et qu'il est possible d'avoir des premiers clients sans nécessiter d'algorithmes évolués de recherche de chemin;
5. de prendre le temps de trouver les bonnes ressources bibliographiques (comme <https://www.redblobgames.com/pathfinding/a-star/introduction.html>) si vous souhaitez effectuer des recherches de chemin. En particulier, gardez à l'esprit que, pour ce problème, le verger est représenté simplement par un tableau régulier de cases.



Le conseil avisé de tonton Charlemagne :

Restez le plus simple possible et avancez progressivement.

Une stratégie sophistiquée peut être difficile à implémenter et elle n'est pas forcément synonyme de stratégie efficace. Il est possible qu'une stratégie simple correctement codée soit plus efficace que de nombreuses stratégies complexes.

N'hésitez pas à réfléchir collectivement et à vous répartir les tâches au sein de votre groupe. Intégrer du code développé en parallèle par plusieurs personnes peut aussi vite prendre beaucoup de temps et il peut être utile de désigner un membre de votre groupe en charge de le faire. **Attention aux mauvaises surprises de dernière minute !**

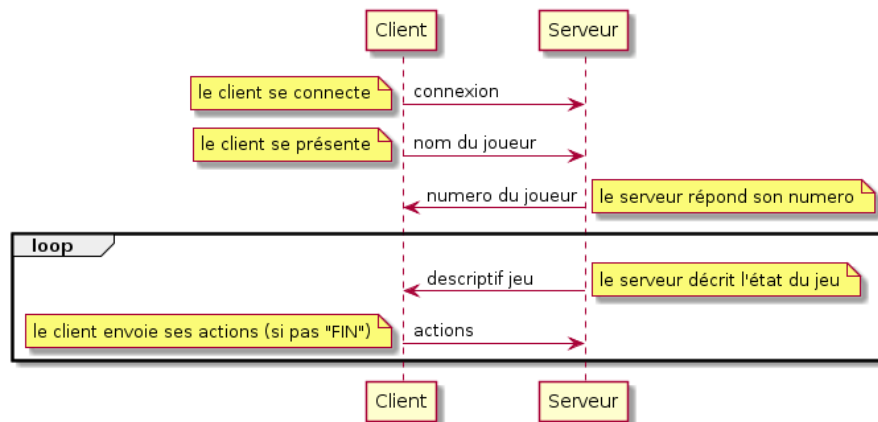
4 Protocole de communication avec le serveur

Cette partie présente en détail la manière dont les échanges doivent se faire avec le serveur. Elle peut être parcourue dans un premier temps et sera surtout utile aux personnes devant implémenter la communication avec le serveur.

4.1 Organisation des échanges

Les échanges avec le serveur doivent impérativement se faire en respectant la procédure décrite dans cette partie.

1. Une fois la connexion établie, la première étape de la procédure consiste à se présenter au serveur :
 - (a) **Envoi du nom** : à la connexion, le client doit d'abord envoyer son nom au serveur pour que le serveur puisse l'identifier. Pour que cela fonctionne correctement, le nom doit être suivi d'un retour chariot `\n`.
 - (b) **Réception du numéro de joueur** : à la réception du nom du client, le serveur répond au client en lui transmettant son numéro de joueur pour la partie.
2. Une fois qu'un nombre suffisant de clients s'est présenté au serveur, le jeu peut commencer. La communication entre le client et le serveur se fait alors de la manière suivante :
 - (a) **Réception du message du serveur** : le client attend que le serveur lui transmette un message. Le message reçu peut prendre deux formes :
 - i. Il peut s'agir d'un message `"FIN"` signifiant que le jeu est fini et le client peut alors se déconnecter.
 - ii. Sinon, ce message correspond au descriptif de la situation (cf. partie 4.2).
 - (b) **Émission de l'action** : le client envoie une chaîne de caractères construite selon la structure indiquée partie 4.3 et correspondant aux actions des membres de son équipe. **Cette chaîne de caractères doit impérativement se terminer par un retour chariot.**
 - (c) **Boucle** : le client se met en attente du prochain message du serveur qui décrira la prochaine situation et recommence le cycle.



Descriptif des échanges réseau.

4.2 Messages envoyés par le serveur

À chaque tour de jeu, le serveur envoie au client une chaîne de caractères (aussi appelée trame) décrivant la situation de jeu comme la chaîne suivante qui servira d'exemple :

```
0_4_13:13,XXXXXXXXXXXXX,X.....01...X,X...XX.XX...X,X.X.....4X.X,X1X.....X1X,X.....3....
.X,X...0...2...X,X0....3....0X,X1X.....X.X,X.X4.....4X.X,X...XX.XX...X,X...10.0....X,XXXX
XXXXXXXXXX_Equipe0,P,P0:3:5:4,P1:5:1:x,P2:4:6:2,Z,Z0:2:1,Z1:1:1,Z2:1:2,G,2,F,F0:1,F1:1,F2:0,
F3:0,F4:0_Equipe1,P,P0:8:11:x,P1:9:8:x,P2:11:8:x,Z,Z0:11:10,Z1:11:11,Z2:10:11,G,2,F,F0:0,F1
:2,F2:0,F3:0,F4:1_Equipe2,P,P0:1:11:x,P1:4:11:x,P2:3:8:x,Z,Z0:2:11,Z1:1:10,Z2:1:11,G,-3,F,F
0:0,F1:0,F2:0,F3:0,F4:2_Equipe3,P,P0:11:5:0,P1:7:5:x,P2:10:1:x,Z,Z0:11:1,Z1:10:1,Z2:11:2,G,
0,F,F0:0,F1:0,F2:0,F3:0,F4:0
```

Cette trame correspond à la situation de jeu de la figure 4

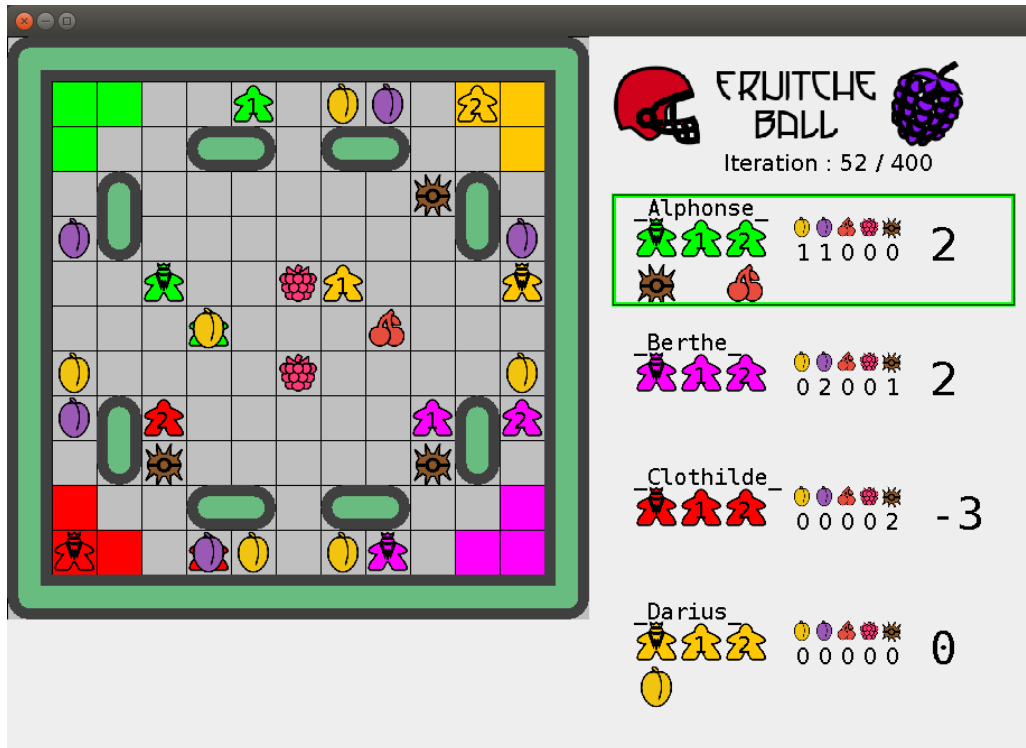


FIGURE 4 – Situation de jeu correspondant à la chaîne donnée en exemple

Cette chaîne de caractères est structurée par blocs séparés par des caractères *underscore* `_`. Ces blocs sont ordonnés de la manière suivante :

1. le numéro du joueur en cours (ici `0` - ce qui signifie que le client contrôle l'équipe numéro 0) ;
2. le nombre de joueurs dans la partie (ici `4`) ;
3. le descriptif complet de la carte (cf partie 4.2.1) ;
4. les descriptifs des équipes de chaque client l'une après l'autre (cf partie 4.2.2).

4.2.1 Descriptif de la carte

Le **descriptif de la carte** est structuré de sous-blocs séparés par des virgules `,`, avec

- la taille de la carte selon ses deux dimensions séparées par `:` (ici `13:13`)
- le descriptif de la carte ligne par ligne (les caractères utilisés sont décrits dans la suite)
 - la première ligne (ici `XXXXXXXXXXXXX`);
 - la deuxième ligne (ici `X.....01...X`);
 - et ainsi de suite (ici pour les 13 lignes de la carte)

Concernant le descriptif de la carte, chaque caractère décrit le contenu de la case correspondante :

- `X` désigne un mur ;
- `.` désigne une case vide ;
- un chiffre entre `0` et `4` désigne le type de fruit posé sur la case (qui ne contient donc pas de mur).

0 - Mirabelle 	1 - Prune 	2 - Cerise 	3 - Framboise 	4 - Châtaigne 
--	--	---	--	--

4.2.2 Descriptif des équipes

Le **descriptif d'une équipe** est constitué de sous-blocs séparés aussi par des virgules `,`, avec

- en position 0, `Equipe0`, le bloc d'entête avec le numéro de l'équipe (ici l'équipe `0`)
- en positions 1 à 4, `P,P0:3:5:4,P1:5:1:x,P2:4:6:2`, les blocs décrivant les 3 membres de l'équipe dans l'ordre avec leur numéro, leur position et leur inventaire, `x` signifiant que le personnage ne possède rien, soit :
 1. `P0:3:5:4` le **Quetcherback** (en (3,5) et possédant un fruit de type 4 - châtaigne) ;
 2. `P1:5:1:x` le **Lanceur 1** (en (5,1) et ne possédant rien) ;
 3. `P2:4:6:2` le **Lanceur 2** (en (4,6) et possédant un fruit de type 2 - Cerise).
- en positions 5 à 8, `Z,Z0:2:1,Z1:1:1,Z2:1:2`, les blocs décrivant les positions des 3 cases de la zone de départ de l'équipe (ici les cases (2,1), (1,1) et (1,2))
- en positions 9 à 10, `G,2` le bloc donnant le score global de l'équipe (parfois négatif cf. Equipe2)
- en position 11 à 15, `F,F0:1,F1:1,F2:0,F3:0,F4:0`, les blocs décrivant le nombre des fruits rapportés pour les 5 types de fruits possibles (`F0:1` 1 fruit de type mirabelle (0), `F1:1` 1 fruit de type 1,...). Le dernier type (`F4:0`) correspond aux châtaignes et malus reçus.

Le nombre de membres dans une équipe, le nombre de cases de la zone de départ et le nombre de types de fruits sont fixés et sont identiques pour tous les vergers proposés (respectivement, 3 membres, 3 cases de départ et 5 types de fruit), ce qui fait que les **indices indiqués des sous-blocs** dans la chaîne d'une équipe restent **identiques** pour tous les problèmes.



Le conseil avisé de tonton Charlemagne :

Pour analyser la chaîne de caractères, vous pourrez utiliser des méthodes permettant de décomposer une chaîne selon des séparateurs (par exemple la méthode `split` en Java).

4.3 Messages attendus du client

À chaque fois que le serveur demande les actions du joueur, le client doit envoyer une chaîne de caractères constituée de trois morceaux séparés par des tirets - :

- le premier morceau correspond à l'action du **Quetcherback** ;
- le deuxième morceau correspond à l'action du **Lanceur 1** ;
- le troisième morceau correspond à l'action du **Lanceur 2** ;
- enfin, un caractère de retour chariot `\n` est nécessaire pour valider la chaîne.

Les chaînes représentant les actions des personnages sont les mêmes quel que soit le personnage :

Déplacements	
N	se déplacer vers le Nord
S	se déplacer vers le Sud
E	se déplacer vers l'Est
O	se déplacer vers l'Ouest
X	rester sur place
Prise / Pose / Échange	
P	échanger l'objet de son inventaire avec la case
Lancer de fruit	
LN	lancer le fruit possédé vers le Nord
LS	lancer le fruit possédé vers le Sud
LE	lancer le fruit possédé vers l'Est
LO	lancer le fruit possédé vers l'Ouest

Ainsi, envoyer la chaîne `LN-E-P\n` consiste à :

- demander au **Quetcherback** de lancer son fruit vers le Nord ;
- demander au **Lanceur 1** de se déplacer vers l'Est ;
- demander au **Lanceur 2** d'échanger l'objet qu'il possède avec l'objet sur sa case.

Si la chaîne ne contient pas assez de morceaux comme `N-E\n`, les actions sont exécutées dans l'ordre (ici le **Quetcherback** et le **Lanceur 1**) et les membres qui n'ont pas d'action associée (ici le **Lanceur 2**) resteront sur place ce tour-ci. De la même manière, si la commande d'un personnage n'est pas une commande valide, la commande est simplement ignorée (comme si le personnage faisait l'action de rester sur place).



La sage remarque de tonton Charlemagne :

Attention, l'oubli des tirets conduira à une seule action qui sera mal interprétée.

5 Lancement de l'application

5.1 Lancement du serveur

Le serveur fourni est inclus dans le fichier `fruitcheball.jar`. Il s'agit d'une archive `jar` contenant des fichiers `.class` qu'il est possible de lancer avec une JVM⁷. Java doit donc être installé sur votre machine pour pouvoir lancer le serveur⁸.

Le lancement du serveur se fait simplement via la commande suivante (à lancer via une console dans le répertoire du fichier `.jar`) :

```
1 java -jar fruitcheball.jar
```

Cette commande lance le serveur qui ouvre une connexion et se met en attente de connexions des clients. Il est aussi nécessaire que votre machine autorise les connexions pour que les clients puissent se connecter au serveur.



Le conseil avisé de tonton Charlemagne :

Faites en particulier attention à bien autoriser le port associé au serveur (par défaut le port 1337) si vous avez un firewall sur votre machine.

5.2 Connexion des clients

Au lancement du serveur, une fenêtre s'ouvre pour attendre la connexion des clients avant le lancement de l'application (cf. figure 5).

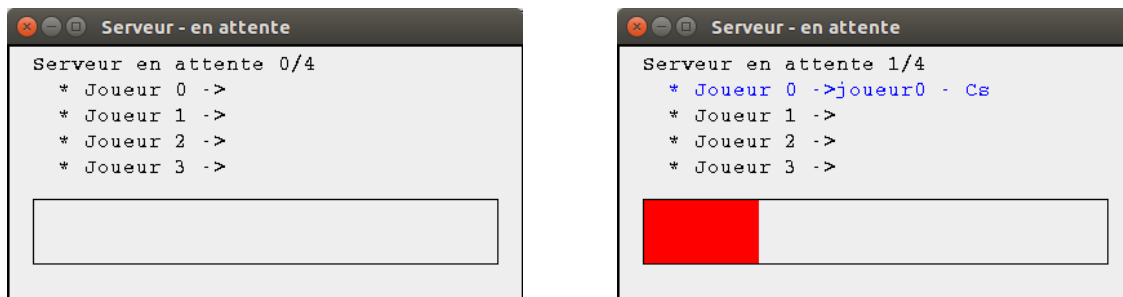
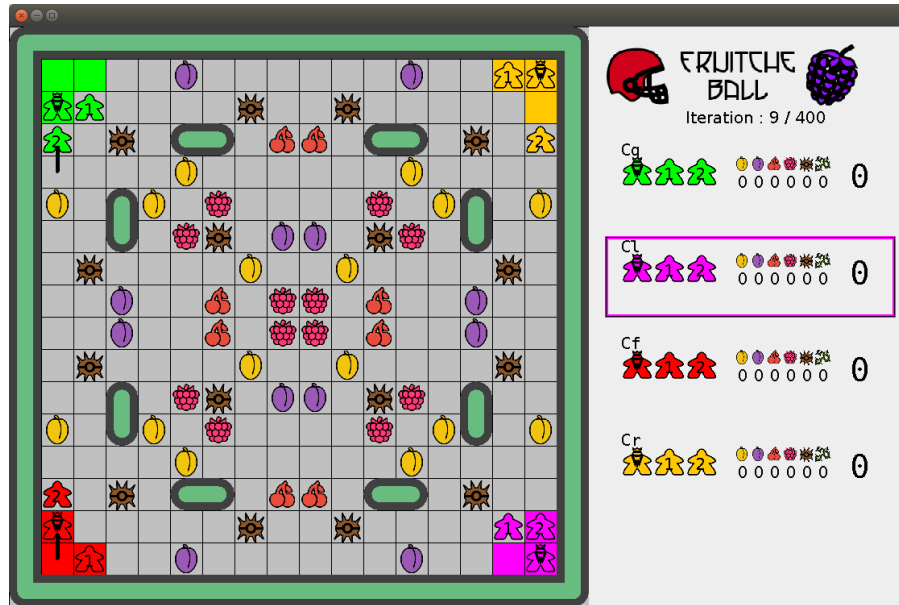


FIGURE 5 – (a) Le serveur est en attente de connexion des joueurs et (b) se met à jour en fonction des joueurs connectés.

Cette fenêtre indique le nombre de joueurs attendus (par défaut 4) et se met à jour au fur et à mesure des connexions des clients. Quand le nombre de clients connectés correspond au nombre de joueurs attendus, le tournoi se lance et le serveur commence les échanges avec les clients connectés (cf. section 4).

7. Java Virtual Machine

8. <https://www.java.com/fr/download/>



Dès que tous les joueurs se sont connectés, le jeu se lance.

5.3 Options du serveur

Plusieurs options sont disponibles au lancement du serveur. Il est possible d'accéder aux différentes options via la commande :

```
1 java -jar fruitcheball.jar --help
```

Le résultat de cette commande fournit les options possibles réunies par type :

```
1 Les options attendues doivent etre passees sous la forme
2   '-valeur=xx' pour donner la valeur xx a la variable valeur
3
4 Les options disponibles et leurs valeurs actuelles sont:
5 * <Reseau>
6   '-port=1337' le port sur lequel lancer le serveur
7 * <Graphique>
8   '-attente=500' temps attente entre chaque tour de jeu (ms)
9   '-taille=40' la taille des cases
10 * <Log>
11   '-log=1' stockage des logs de la partie : 0-fichier 1-ecran 2-vide
12 * <Jeu>
13   '-nbJoueurs=4' le nombre de joueurs (entre 1 et 4)
14   '-laby=0' le labyrinthe utilise (entre 0 et 8 - cf. sujet)
```

Il est possible de modifier chacune de ces options. Par exemple, si on souhaite changer le nombre de joueurs à 2 et accélérer l'exécution du serveur (attente de 100 ms plutôt que 500 ms entre chaque tour de jeu), la commande de lancement du serveur sera :

```
1 java -jar fruitcheball.jar -nbJoueurs=2 -attente=100
```

Il est aussi possible de changer la structure du verger utilisée avec l'option `laby`. Les vergers reconnus par le serveur sont les suivants

- laby=0 : verger standard par défaut (taille 13x13);
- laby=1 : second verger standard (taille 13x13);
- laby=2 : verger standard généré aléatoirement (taille 13x13);
- laby=3 : verger avec peu de fruits;
- laby=4 : verger avec 4 mirabelles pour expérimenter des cas simples;
- laby=5 : verger avec 4 châtaignes pour expérimenter des attaques;
- laby=6 : verger avec beaucoup de châtaignes et peu de fruits;
- laby=7 : verger sans mur;
- laby=8 : verger minimaliste (taille 7x7);
- laby=9 : verger de grande taille (18x18).

Les vergers utilisés lors du tournoi seront très proches des deux vergers par défaut (valeur de l'option `laby` à 0 ou 1) sans être exactement identiques. La structure des labyrinthes utilisée sera identique mais les fruits pourront être répartis différemment dans l'environnement et leur nombre pourra être différent. Le troisième verger (option `-laby=2`) qui pose les fruits de manière pseudo-aléatoire permettra de **tester votre client en conditions similaires à celles du tournoi**.

5.4 Version interactive

Il est aussi possible de lancer le client en mode interactif. Dans ce mode, les commandes se font via la console et non plus via le réseau. Il suffit simplement d'entrer à chaque tour de jeu la commande de l'équipe courante via le clavier (par exemple `N-N-N`).

Ce mode permet de tester des situations et d'expérimenter les règles du jeu pour être sûr d'avoir bien compris les actions associées aux personnages.

```
1 java -cp fruitcheball.jar jeux.fruitcheball.MainClavier
```

Il est possible de modifier les options de la même manière que cela est fait au lancement du serveur. Par exemple en modifiant le nombre de joueurs impliqués :

```
1 java -cp fruitcheball.jar jeux.fruitcheball.MainClavier -nbJoueurs=1
```

En utilisant la redirection de flux, il est aussi possible d'envoyer une suite de commandes textuelles au client en mode interactif.



Le dernier mot de tonton Charlemagne :

Comme le disait Virgile en parlant de la mirabelle, « *addam cerea pruna; honos erit huic quoque pomo.* »^a. En ce jour solennel du 25 Mai 2018, je déclare le tournoi de **Fruitcheball** officiellement lancé.

Bonne chance à tous et que le meilleur gagne !

^a. « J'ajouterai des prunes couleur de cire : ce fruit sera, lui aussi, à l'honneur. »

6 Résumé du sujet

Objectif L'objectif de cette épreuve est de coder un client contrôlant les membres d'une équipe de **Fruitcheball** se déplaçant dans un verger et devant amasser le plus de points possibles en récoltant des fruits.

Équipe Une équipe est constituée de trois membres : un **Quetcherback** et deux **Lanceurs**. Le **Quetcherback** est le seul membre du groupe capable de prendre, d'intercepter et de lancer des châtaignes.

Points En début de partie, les équipes ne possèdent aucun point.

- Une équipe **gagne des points** dès qu'un fruit est déposé ou lancé dans sa zone de départ. Le nombre de points récoltés pour ce fruit correspond au nombre de fruits du même type déjà déposés (1pt pour le premier fruit, 2pts pour le deuxième fruit, ...);
- Une équipe **perd des points** si une châtaigne est lancée ou déposée dans sa zone de départ ou si un de ses lanceurs reçoit une châtaigne selon le même principe (-1pt la première fois, -2pts la deuxième, ...)

Structure de la carte À un instant donné, la carte est décrite par ce qui se trouve sur ses différentes cases. Il peut s'agir :

- **X** : d'une clôture infranchissable et non déplaçable
- **.** : d'une case vide
- **0** à **5** : d'une case accessible contenant un fruit dont le type correspond au chiffre (5 correspond à une châtaigne).

Actions possibles À chaque tour de jeu, le client doit retourner une chaîne de caractères contenant les actions des 3 membres de l'équipe (dans l'ordre **Quetcherback**, **Lanceur 1** et **Lanceur 2**) séparés par des tirets **-**⁹.

L'action d'un membre d'une équipe peut consister :

- soit à se déplacer **N**, **S**, **E**, **O** ou à rester sur place **X**
- soit à échanger un fruit de son inventaire avec sa case **P**¹⁰
- soit à lancer le fruit possédé de 4 cases dans une direction donnée **LN**, **LS**, **LE**, **LO**

Règles de lancer Le fruit lancé avance dans la direction souhaitée de 4 cases jusqu'à rencontrer un obstacle (clôture ou personnage possédant un fruit) ou un personnage sans fruit qui l'intercepte automatiquement.

Lorsque le fruit tombe sur une case occupée par un fruit, il rebondit de case en case de manière aléatoire jusqu'à arriver sur une case vide de fruit.

Règles d'or Du fait des règles du jeu,

1. Aucun personnage ne peut se trouver sur une case occupée par un autre personnage;
2. Un personnage ne peut porter au plus qu'un fruit;
3. Une case ne peut contenir plus d'un fruit.

9. N'oubliez pas de terminer la chaîne par un retour chariot pour valider l'envoi.

10. Ce qui inclut la prise et la pose de fruits.