Avant-projet du projet informatique

Matthieu Denoux - Groupe 1

26 décembre 2013

Table des matières

Chapitre 1

Présentation du sujet

1.1 Principe général

Le sujet choisi a pour intitulé **Babyfoot en réseau**. Il s'agit de concevoir un système complet de jeu en réseau. Le système serait donc séparé en deux parties, une serveur et une client. Chaque joueur pourrait donc se connecter à une partie n'ayant pas encore comméncé et une fois le nombre de joueurs réunis (*2 ou 4*), la partie serait lancée. Il faut donc réaliser à la fois le système réseau, l'interface graphique et imaginer un gameplay qui rende le jeu agréable.

1.2 L'interface graphique

Au niveau de l'interface graphique, il faudra réaliser plusieurs fenêtres successives de menus pour parvenir jusqu'au jeu lui-même.

1.2.1 Menu principal

Un premier menu, dit *menu principal*, permettra de commencer une nouvelle partie en ligne, de rejoindre une partie déjà en cours, de modifier les options ou bien de quitter le jeu.

1.2.2 Commencer une partie

Si l'on commence une nouvelle partie en ligne, on se retrouve dans une « salle d'attente ». Là, il est possible de configurer la partie que l'on souhaite lancer, l'ouvrir à d'autres joueurs puis attendre que d'autres joueurs rejoignent la partie. Une fois que les équipes sont complètes, on peut lancer le jeu.

1.2.3 Rejoindre une partie

On peut aussi sélectionner une partie dans la liste des parties déjà commencées, dans la limite des places disponibles. On rejoint alors un salon similaire à celui décrit dans la section précédente où l'on attend que la partie soit complète avant que le meneur, celui qui a créé la partie, ne lance le jeu.

1.2.4 Déroulement d'une partie

Il y aurait deux gameplays différents :

— l'un entièrement manuel verrait le joueur maître de ses possibilités. Ainsi, les touches « A », « Z », « E », « R »permettraient de sélectionner la canne que l'on souhaiterait manier. Les flèches « Haut »et « Bas »permettraient quant à elle de déplacer les cannes tandis que la barre espace commanderait le tir. Je ne pense pas réaliser plusieurs puissances de tir possibles mais cela pourrait consister en une amélioration de la richesse du gameplay.
— l'autre, principalement rencontrée dans les autres jeux de babyfoot trouvés en ligne, serait plus automatisée : les cannes se déplaceraient toutes ensemble de manière synchronisée avec les flèches « Haut »et « Bas ». Le tir pourrait être ou non automatisé selon que j'aurais le temps pour configurer cette fonctionnalité supplémentaire.

1.2.5 Options

Un petit menu sera consacré aux options, notamment le type de jeu que l'on souhaite utiliser (manuel ou automatisé, auquel cas jusqu'à quel point) et d'autres options.

Chapitre 2

Analyse de la solution envisagée

2.1 Découpage en modules

Je prévois de découper le code en trois modules principaux :
- L'interface graphique (rangée dans /gui)
- La partie réseau de l'application (rangée dans /network)
- Le cœur algorithmique de l'application (rangée dans /core)

Chaque partie comportera donc plusieurs classes qui se chargeront chacune d'une des tâches du module parent.

2.2 L'interface graphique

2.2.1 MainFrame.java

Gère la fenêtre qui englobe tout le reste. On utilisera en fait des JPanel pour modifier le contenu de cette fenêtre. J'empêcherai dans un premier temps de modifier la taille de la fenêtre pour éviter d'avoir des problèmes de dessin du terrain de babyfoot à gérer. Cette classe contient le **main** du programme.

2.2.2 BPanel.java

Classe abstraite présentant certaines actions mécaniques quant aux caractéristiques des JPanel utilisés (header, background color, taille, layoutmanager, etc.). La plupart des panels ci-dessous en héritent.

2.2.3 MenuPanel.java

Gère le menu principal affiché à l'ouverture du jeu. Contient les boutons qui mèneront vers les principales actions citées plus haut (nouvelle partie, rejoindre, options, quitter).

2.2.4 NewPanel.java

Lorsque l'on lance une nouvelle partie, on obtient cet écran qui contiendra les principales options nécessaires pour configurer une partie puis l'ouvrir à des joueurs extérieurs. Une fois cette partie ouverte, on aboutit à une *salle d'attente*.

2.2.5 ServersPanel.java

Gère la liste des parties actuellement en recherche de joueurs lorsque l'on cherche à rejoindre une partie. On sélectionne une partie dans cette liste puis on aboutit à une *salle d'attente*.

2.2.6 WaitingRoomPanel.java

Une fois la partie configurée et ouverte, on aboutit dans cette salle d'attente qui attendra que certaines conditions soient réunies pour permettre au jeu d'être commencé. On peut *aussi* y accéder depuis la partie *rejoindre une partie*.

2.2.7 GamePanel.java

La partie à proprement parler. Donc le conteneur du dessin du terrain qui gère les éléments extérieurs, les événements et toute autre interaction avec le reste du code. Inclut une zone de dessin.

2.2.8 GameZone.java

Zone de dessin chargée de représenter le terrain, les joueurs, etc.