

Rédacteurs : Romuald Cari

Référence document : TOPSPORT-MAN-TECH

Contact : Cédric Dumas

Cedric . Dumas @ emn . fr

Version : 1.0

Révision : 11

Date : 04/01/2007

sous licence CREATIVE COMMONS Attribution-NonCommercial-ShareAlike 2.0 France  
<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

Romuald Cari  
Damien Coppéré  
Alexandre Moyrand

## Projet OSE Fi2



### **Rapport Technique** **(extrait du rapport final du projet)**

Sujet: Création d'un logiciel de saisie de scores pour une compétition sportive

# Sommaire

<b>INTRODUCTION.....</b>	<b>3</b>
<b>1 - LES OBJECTIFS DU PROJET.....</b>	<b>3</b>
A -) LE SITE WEB.....	3
B -) TOPSPORT.....	3
<b>2 - CONCEPTION DU PROGRAMME TOPSPORT.....</b>	<b>4</b>
A -) LES RESSOURCES MOBILISÉES.....	4
B -) L'INTERFACE HOMME-MACHINE.....	5
C -) LE MODÈLE DE NOTRE APPLICATION.....	10
1 - <i>Présentation</i> .....	10
2 - <i>Propriétés</i> .....	11
3 - <i>Fonctionnement</i> .....	11
<b>3 - LES RESULTATS DU PROJET.....</b>	<b>12</b>
A -) L'INSCRIPTION.....	12
B -) LE GÉNÉRATEUR DE TOURNOI.....	13
C -) LE MANAGER DE TOURNOI.....	14
<b>4 - AMÉLIORATION POSSIBLES.....</b>	<b>17</b>
<b>CONCLUSION.....</b>	<b>17</b>

## Introduction

Le logiciel que nous devons créer lors de ce projet informatique devait gérer la saisie des scores de tournois sportifs multidisciplinaires, et plus particulièrement le Cartel des Mines 2006. En effet cette manifestation assez spécifique était un prétexte pour réaliser un manager de tournoi très générique, portable et simple d'utilisation. Il devait s'appuyer sur le générateur de tournoi réalisé par une équipe de FI3.

Une adaptation sur Tablet PC était demandée, ce qui nécessitait de restreindre au maximum les saisies au clavier. Les fonctions essentielles demandées du logiciel étaient la saisie des scores et la diffusion de ceux-ci le plus clairement possible sur les différents sites sportifs.

Pour réaliser notre application, nous nous sommes répartis les tâches équitablement et nous avons développé toujours conjointement.

## 1 – Les objectifs du projet

### A -) Le site web

La première étape du projet TopSport fut de réaliser un site web ayant pour but de diffuser des informations relatives à projet sportif ' Cartel des Mines '. Ce site, en plus d'être purement informatif, devait permettre aux participants de s'inscrire par délégation. Tous les responsables de délégations avaient un identifiant et un mot de passe. Cet accès sécurisé permettait d'assurer la fiabilité des données saisies : Tout le monde n'avait pas accès à la partie 'inscription en ligne' du site web.

L'utilisateur devait avoir le choix d'inscrire son école dans un nouveau sport ou de modifier une précédente inscription. La procédure d'inscription devait comporter un certain nombre d'étapes permettant de récupérer toutes les informations nécessaires pour remplir correctement la base de données et respecter la structure.

### B -) TopSport

Le programme TopSport est un programme ayant pour but de gérer un tournoi sportif préalablement créé (avec le programme créé par l'équipe de FI3).

Durant les épreuves ou à la fin de chaque épreuve à l'aide d'un tablet PC (ou équivalent) l'utilisateur doit pouvoir entrer le résultat d'un match. En effet le programme lui doit donner la possibilité de choisir le sport pour lequel il veut faire une saisie.

Dans le cas où le sport considéré implique un tournoi en poules, l'utilisateur peut visualiser, pour chacune des poules, un tableau récapitulatif de tous les matchs ainsi que les scores.

Si le sport se joue en éliminations directes, il peut voir l'arbre des matchs et les résultats.

Dans les deux cas, l'utilisateur doit pouvoir facilement modifier les scores.

Le classement final ou la suite des épreuves doit donc être automatiquement généré par le logiciel.

Nous avons établi un certain nombre de spécifications fonctionnelles que devait pouvoir réaliser notre application :

- Sélection d'un tournoi à charger
- Sélection du sport dont les scores doivent être saisis.
- Les scores sont affichés dans un tableau qui reprend en abscisse et en ordonnée toutes les équipes du championnat.
- A côté de ce tableau, un classement de ce championnat est généré en temps réel.
- Pour saisir le score d'un match, on clique sur la case qui a pour ordonnée et abscisse chacune des deux équipes se disputant le match.
- Pour modifier le score il suffit de cliquer sur les boutons + et – prévus à cet effet.
- Si le match est terminé, appuyer sur le bouton "Match terminé". Ceci verrouillera le match et empêchera la modification du score final.
- Dans le cas où le match n'est pas terminé, appuyer sur le bouton "Actualiser". Ceci mettra le score du match à jour dans le tableau de la fenêtre principale.

Pour un sport qui se joue en coupe :

- La fenêtre principale est composée d'un arbre qui représente les différentes phases.
- Un match en particulier est représenté par une zone rectangulaire dans laquelle sont rappelés les noms des deux équipes avec, en face de chacune d'elles, le nombre de points gagnés, et un bouton permettant d'accéder à la saisie du score.
- Pour saisir le score d'un match, cliquer sur le bouton prévu à cet effet.
- Pour modifier le score il suffit de cliquer sur les boutons + et – prévus à cet effet.
- Si le match est terminé, appuyer sur le bouton “Match terminé”. Ceci verrouillera le match et empêchera la modification du score final.
- Dans le cas où le match n'est pas terminé, appuyer sur le bouton “Actualiser”. Ceci mettra le score du match à jour dans l'arbre de la fenêtre principale.
- Une fois le match verrouillé, le nom de l'équipe gagnant sera automatiquement reporté dans la case correspondant au match suivant.

Pour un sport qui se joue en Poules:

- La navigation entre les différentes interfaces qui affichent les scores des poules se fait à l'aide d'onglets. Se rajoute à ces onglets l'onglet “Phase finale” dont la première étape sera remplie une fois les phases de poules terminées.
- Pour chaque poule, les scores sont affichés dans un tableau qui reprend en abscisse et en ordonnée toutes les équipes de la poule.
- A côté de ce tableau, un classement de cette poule est généré en temps réel.
- Pour saisir le score d'un match, on clique sur la case qui a pour ordonnée et abscisse chacune des deux équipes se disputant le match.
- Pour modifier le score il suffit de cliquer sur les boutons + et – prévus à cet effet.
- Si le match est terminé, appuyer sur le bouton “Match terminé”. Ceci verrouillera le match et empêchera la modification du score final.
- Dans le cas où le match n'est pas terminé, appuyer sur le bouton “Actualiser”. Ceci mettra le score du match à jour dans le tableau de la fenêtre principale.
- Lorsque les matches de poules sont tous terminés, la phase finale est prête à démarrer et s'exécute comme une coupe.

Saisie des scores :

- Pour saisir un score, l'utilisateur rentre le nombre de points gagnés par les équipes (jeux pour le tennis)
- Pour ajouter un set au match, il faut cliquer sur le bouton + situé au niveau de la barre d'onglet. Pour modifier le score du set suivant, la procédure est la même que précédemment.
- Il y a un bouton “Saisir le résultat Final” pour éviter de rentrer les jeux un par un et directement inscrire le nombre de sets gagnants par équipe. La saisie se fait comme pour tous les autres matchs, avec des + et des -.

Nous avons aussi envisagé d'ajouter à notre projet une partie ‘ Affichage des résultats ’. Nos devons afficher de manière assez autonome les résultats de tous les sports. Pour les poules, l'application devait pouvoir afficher l'arbre des matchs dans le cas des phases finales et un tableau avec un classement dans le cas de poules.

## **2 – Conception du programme TopSport**

### **A -) Les Ressources mobilisées**

Pour la réalisation du site web nous avons utilisé le langage PHP pour la partie active : l'inscription en ligne. Les fonctions en PHP remplissaient une base de données MySql gérée grâce à EasyPHP.

L'application TopSport est basée sur la même base donnée. Elle utilise les informations qui y ont été rentrées par le générateur de tournoi réalisé par l'Equipe de Fi3. Aussi nous avons utilisé Eclipse pour

implémenter ce programme. Nous avons utilisé une library pour que notre projet prenne en charge les requêtes SQL. Ce fichier était mysql-connector-java.jar. Notre programme a pour but d'être utilisé sur un tablet PC. Aussi, nous avons testé notre application sur un de ces appareils empreinté au département informatique de l'école.

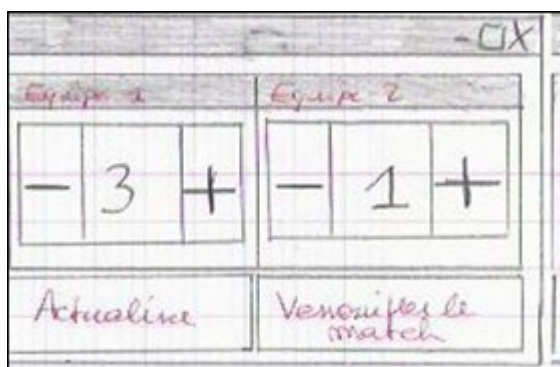
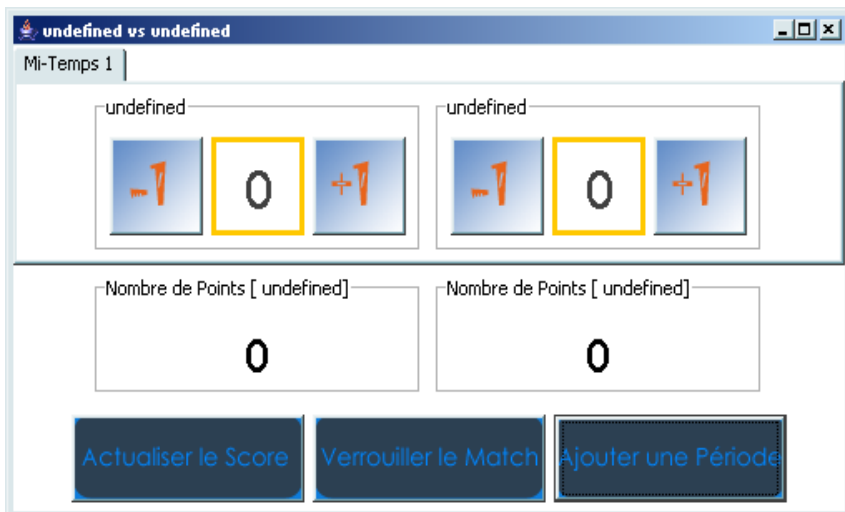
## B -) L'Interface Homme-Machine

Lorsque nous avons commencé à penser à l'interface de notre programme, nous avons imaginé nos fenêtres sur papier. Cette première étape a pour but de laisser place à l'imagination et à concevoir un grand nombre de fenêtres, qui permettent par la suite d'avoir un plus large choix. Nous avons choisi les prototypes que nous pensions être les plus intéressants, en considérant deux points: L'ergonomie et la faisabilité.

En effet, dans un premier temps, nous avons voulu que notre interface graphique soit utilisable au stylet sur un tablet PC. Nous avons donc banni toutes les saisies au clavier pour n'utiliser que des JButton. Par exemple pour la modification des scores, nous avons fait des fenêtres utilisant exclusivement des JButton pour augmenter ou diminuer les scores, pour valider la saisie et pour verrouiller le match. Par ailleurs nous avons utilisé des JLabels pour l'affichage en temps réel du score. Des variantes ont été apportées une fois les fenêtres réalisées en swing. Nous avons conçu des fenêtres différentes pour les sports tels que Tennis, Football, ou encore Basket.

Notre application devait prendre en compte le fait qu'un match a plusieurs périodes. Nous avons donc inséré des onglets (JTabbedPane) dans nos fenêtres pour naviguer facilement dans les steps.

Voici un aperçu de la différence entre notre prototype et la version finale de la fenêtre (Figure 1):



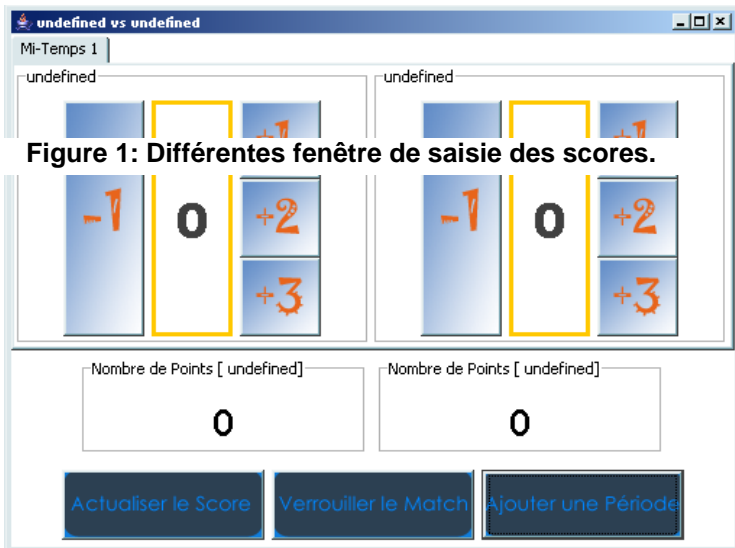


Figure 1: Différentes fenêtre de saisie des scores.

A partir de cette fenêtre nous avons créé les cas particuliers pour le basket et le tennis. En effet, au basket on doit pouvoir entrer les points 1 par 1, 2 par 2, ou 3 par 3. Ceci nous a donc obligés à envisager tous ces cas. Pour le tennis, le problème venait du nombre total de points qui devait être exprimé en sets. Nous avons donc conçu un cas particulier avec un algorithme qui calcul de nombre de sets en fonction des jeux.

La fenêtre de base du programme est une JFrame qui contient autant de JTabbedPane que de sports différents. En effet, chaque sport a son onglet. Cela permet de naviguer plus facilement et plus rapidement. Une fois chaque onglet créé nous devons différencier trois types de tournois possibles: Soit le tournoi était un championnat, soit en coupe ou soit en poule suivies des phases finales. Ces trois cas ont beaucoup compliqué notre interface graphique.

Dans le cas du championnat, nous devons afficher un tableau pour chaque groupe ainsi que le classement général du tournoi. Pour une coupe, nous devons afficher un arbre qui était de taille très variable dépendant du nombre d'équipes inscrites dans la base de données pour le sport considéré. Enfin, dans le cas d'un tournoi en poules+phases finales, nous devons séparer ces deux étapes de manière assez claire. Nous avons donc pensé une fois de plus au JTabbedPane. Notre fenêtre principale devait donc ressembler à cela (Figure 2):

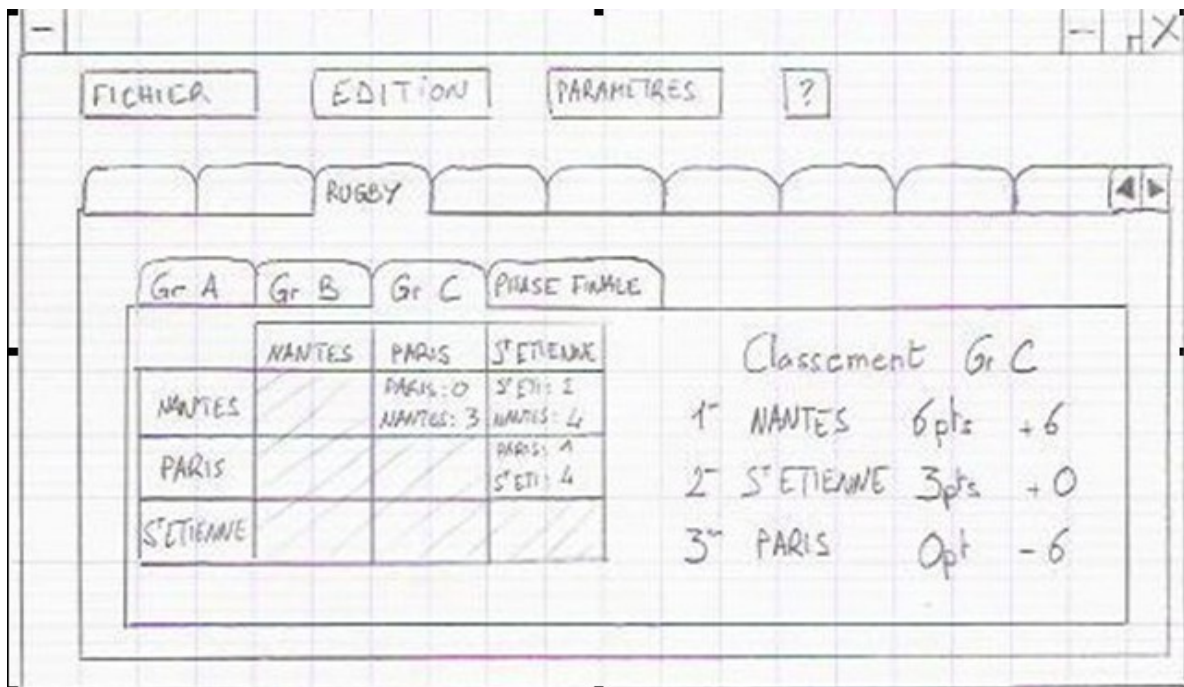


Figure 2: Widget de l'interface graphique associée à la gestion des Groupes

Cependant nous avons préféré mettre, pour ce cas ci, les tableaux de toutes les poules dans un même JTabbedPane et l'arbre des phases finales dans un autre pour avoir une vision plus globale du tournoi. Ceci nous a donc amené à concevoir une fenêtre plus ergonomique (Figure 3 et 4):

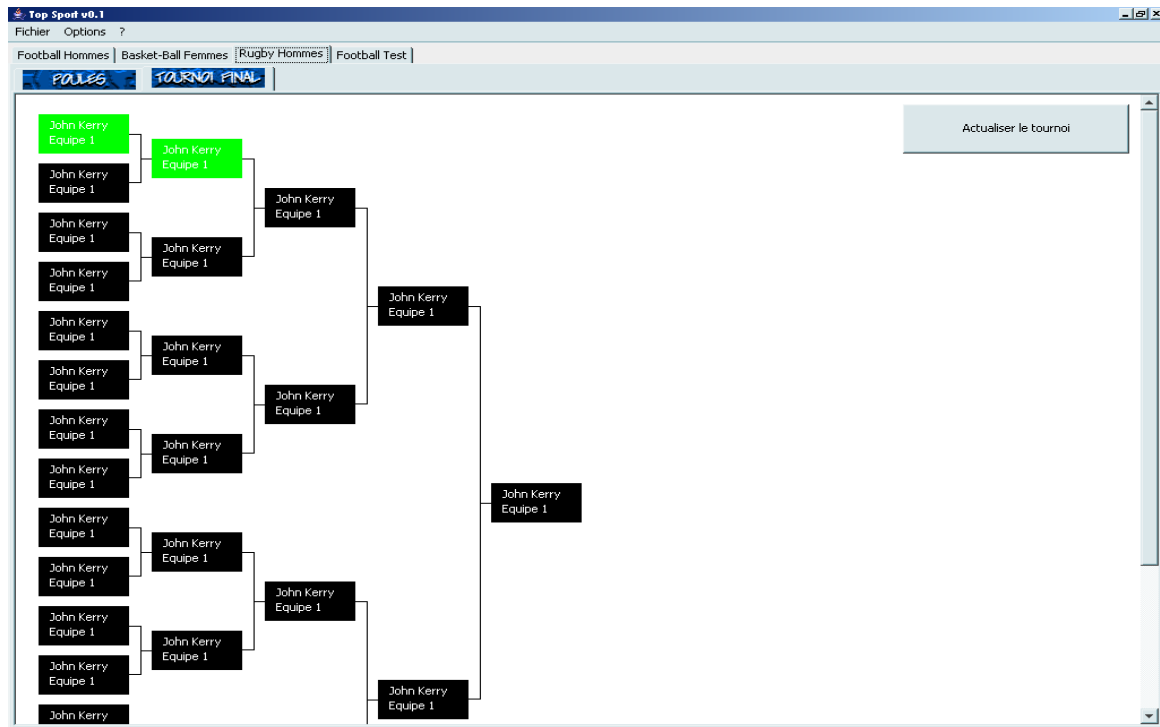


Figure 3: Screenshot de notre application pour la gestion d'une coupe

Groupe 1	John Kerry	Equipe 1	Equipe 2	ah!?
John Kerry				
Equipe 1	John Kerry : 9 Equipe 1 : 7			
Equipe 2	John Kerry : 4 Equipe 2 : 5	Equipe 1 : 0 Equipe 2 : 0		
ah!?	John Kerry : 7 ah!?: 11	Equipe 1 : 3 ah!?: 5	Equipe 2 : 5 ah!?: 3	

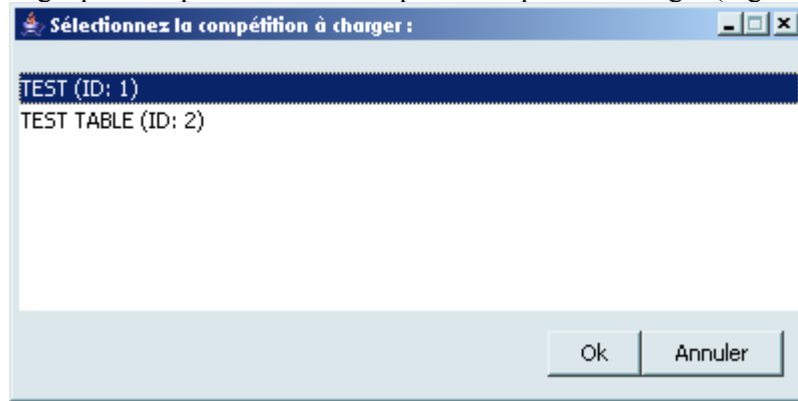
Classement des Groupes	
<b>Groupe 1</b>	
1	John K... 20
2	ah!? 19
3	Equipe 2 10
4	Equipe 1 10
<b>Groupe 2</b>	
1	Equipe 1 7
2	John K... 5

Figure 4: Screenshot de notre application pour la gestion d'un groupe

La particularité de ces représentations en tableau et en arbre est qu'elles sont réalisées

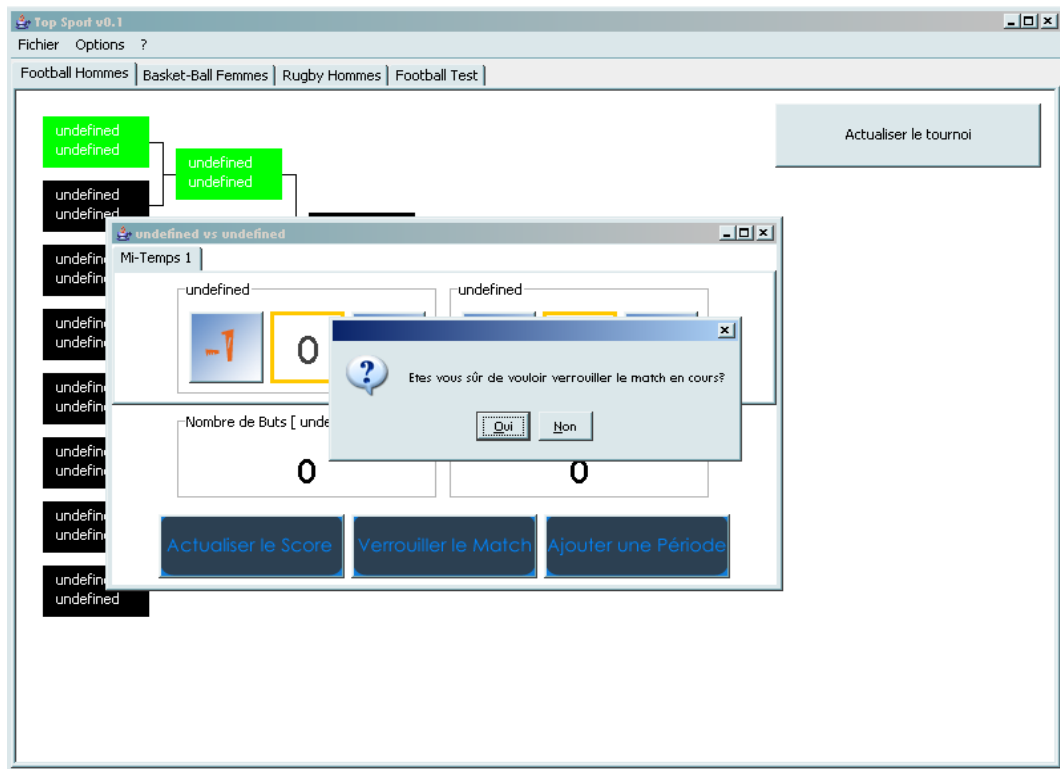
graphiquement. En effet dans les deux cas, les cellules avec les noms des deux équipes concernées par le match sont des JPanel contenant chacun un JLabel. Nous avons ajouté des MouseListener pour qu'une fenêtre de saisie des scores s'affiche sur un double click. Le principe de ces deux représentations est identique. Il n'y a que l'algorithme de placement qui change et les liaisons entre les cellules. Nous avons aussi ajouté une fonction qui colore la cellule en vert lorsque le match a été verrouillé et qui reporte le nom du gagnant dans la cellule de l'étape suivante.

Au cours de la conception de notre interface graphique nous nous sommes rendu compte que nous devions ajouter des fenêtres auxquelles nous n'avions pas pensé initialement. Par exemple, nous avons ajouté une fenêtre de démarrage qui nous permet de choisir quelle compétition charger (Figure 5).



**Figure 5: Screenshot de la fenêtre "Competition Selection Window"**





**Figure 6: Validation d'une action**

Aussi, nous avons ajouté des fenêtres de dialogue pour demander des confirmations à l'utilisateur (Figure 6). Nous avons jugé que certaines actions, telles que le verrouillage d'un match, devaient être réfléchies puisque irréversibles dans le cas d'un utilisateur lambda.

D'une manière générale, la réalisation de l'IHM a été une très grosse partie de notre travail sur ce projet. Nous avons essayé de créer, autant que possible, un modèle le plus général possible.

Lorsque nous avons commencé à implémenter l'Interface Graphique, nous avons créé une classe BaseWindow dont découlent toutes les autres fenêtres de notre application. Dans notre modèle les seules fenêtres pour lesquelles nous avons jugé nécessaire de créer une relation d'héritage sont les fenêtres de saisie des scores. En effet une ScoreWindow est constituée d'un ScorePanel qui est en fait une classe abstraite. Nous avons trois classes TennisPanel, BasketPanel et FootPanel qui implémentent ScorePanel.

Par ailleurs, nous avons créé des classes pour chacun des objets constituant les tableaux ou les arbres de la fenêtre principale. Par exemple un tableau (Table) est constitué de TableCell et de TableEmptyCell dans lesquelles il y a des TableLabel. Les Table de chaque groupe d'un tournoi sont tous rassemblés dans un MasterTable. La structure est sensiblement la même pour les arbres.

## C -) Le modèle de notre application

TopSport repose sur un modèle objet qui se veut avant tout générique : une des spécifications fonctionnelles de TopSport était de pouvoir gérer une grande variété de tournois. La structure de données associée devait donc être compatible avec la plus grande diversité de tournois possible. De plus, dans une logique de portabilité du code, celui-ci a été organisé de manière à isoler la structure de données et l'interface homme machine.

### 1 – Présentation

Cette structure (appelée « core ») est organisée en deux packages, `core.competition` et `core.tournament`.

Une compétition (`core.competition.Competition`) est l'élément de base de la structure, elle regroupe les différents tournois. Cette classe `Competition` correspond à la notion d'événement sportif.

La structure des tournois doit permettre à la fois de gérer les compétitions de type championnat, de type groupes puis arbre d'élimination ou simplement de type élimination. Elle est donc organisée de la sorte (voir Figure 7) :

- un tournoi est décomposé en étapes ou `Step` (`core.tournament.Step`)
- chaque étape est elle-même décomposée en groupes ou `Group` (`core.tournament.Group`)
- chaque groupe est quant à lui composé de matchs (`core.tournament.Match`)
- un match est finalement découpé en Rounds (`core.tournament.Round`).

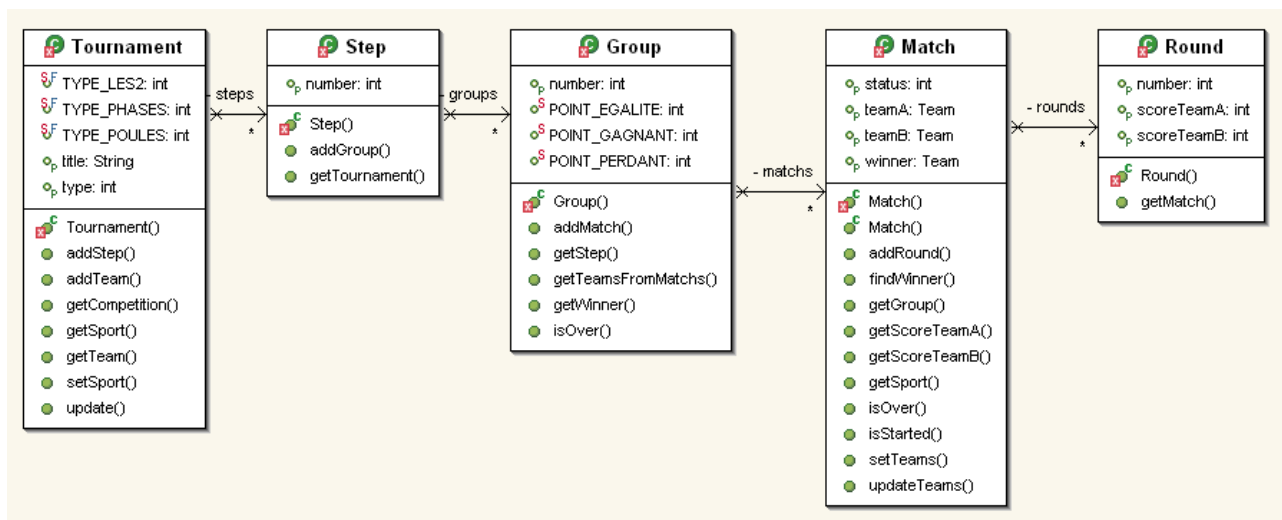


Figure 7: Structure d'un Tournoi

Grâce à cette structuration des données, tous les types de tournois énoncés précédemment peuvent être représentés.

Dans le cas d'un championnat simple, le tournoi s'apparente en réalité à une seule poule réunissant plusieurs équipes. Dans ce cas là un tournoi contenant une seule étape est généré et cette étape ne contient qu'un seul groupe. Dans ce groupe le nombre de matchs dépend de facteurs comme le nombre d'équipe ou comme la configuration des matchs en aller/retour ou en aller simple.

Dans le cas d'un tournoi fait de poules puis d'un arbre éliminatoire, la première étape est composée d'autant de groupes qu'il y a de poules : chaque groupe représente une des poules du tournoi. L'étape suivante est elle composée de N groupes (avec  $N = \text{nombre de poules de la première étape divisé par deux}$ ). Chacun des groupes ne contient qu'un seul match. L'étape suivante sera composée de façon similaire à cette étape, à la différence que le nombre de groupes est de N divisé par deux. Ceci est réitéré jusqu'à ce qu'il n'y ait plus qu'un seul match, indiquant le dernier groupe.

Enfin, pour le cas d'un tournoi simple en arbre, le principe est exactement le même que pour le cas précédent, en omettant la première étape de poules.

## 2 – Propriétés

Toutes les classes représentant les éléments de base de la compétition : équipes (Team), sport (Sport), délégations (Delegation) et compétition (Competition) sont définis dans le package core.competition. Ils sont tous des extensions de la classe CompetitionElement puisqu'ils partagent tous certaines propriétés de base : un identifiant unique (au sein de leur type respectif) et un nom (Figure 8).

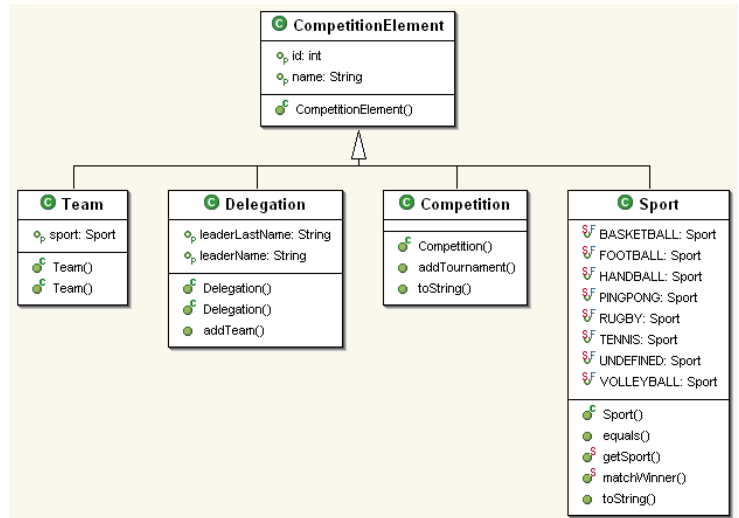


Figure 8: Diagramme de classe du package Competition

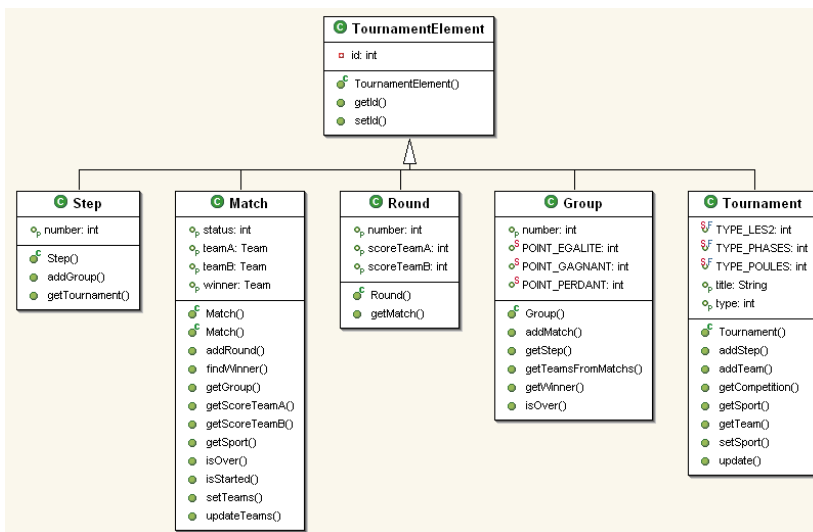


Figure 9: Diagramme de classe du package Tournament

Toutes les classes de base d'un tournoi : étape (Step), groupe (Group), match (Match) et round (Round) sont déclarés dans le package core.tournament. Ils héritent tous de la classe TournamentElement puisqu'ils partagent les propriétés de base qu'est un identifiant unique (Figure 9).

Cette structure a l'avantage de pouvoir être parcourue de manière bidirectionnelle. On peut donc descendre depuis une compétition jusqu'à un round de match et remonter de ce même round vers la compétition englobante. Les relations descendantes sont généralement de type « un vers plusieurs » et les classes englobantes disposent de variables d'instances vers les instances des classes englobées stockées sous la forme de collections (Vector paramétrisés pour la plupart, ou Hashmap paramétrisés également). Les relations montantes sont elles de type « un vers un », pour les caractériser, chacune des classes énoncées précédemment (à l'exception de Competition) dispose d'une variable d'instance qui pointe vers l'instance de la classe qui les englobe.

## 3 – Fonctionnement

L'IHM interagit avec cette structure principalement au niveau des matchs avec des méthodes telles que getWinner() ou setStatus()

### 3 – Les resultats du projet

TopSport permet de gérer tous types de tournoi sportif. Afin de pouvoir manager une compétition 3 étapes sont nécessaires :

- l'inscription
- la génération du tournoi (suivant son type : poule, coupe, championnat, course unique)
- le management à proprement parler (scores, classements...)

#### A -) L'inscription

L'inscription des équipes des différentes délégations se fait via le site Internet développé ici spécialement pour le Cartel des Mines (Figure 10) :



Figure 10: Screenshot du site web '[www.cartelnantes2006.com](http://www.cartelnantes2006.com)'

Le site est destiné à donner des informations pour les participants de cet évènement sportif avec les rubriques : Accueil, Présentation, Informations et Partenaires. Mais dans le cadre effectif de TopSport c'est la partie Inscriptions qui est le départ du management de tournoi.

Chaque délégation présente au Cartel s'est vue fournir un mot de passe et un login nécessaire pour s'authentifier et accéder à l'inscription des différentes équipes. Un responsable par délégation était responsable de l'inscription de tous les membres de sa délégation. En page d'accueil de l'inscription, il y a le choix d'ajouter une équipe dans un sport où l'on n'est pas encore inscrit ou de modifier un sport où l'on est déjà inscrit. L'inscription d'une équipe se fait alors en 6 étapes :

- Choisir le sport
- Entrer le nom et prénom du responsable du sport
- Saisir le nom de l'équipe et le nombre de joueurs
- Renseigner les noms et prénoms des joueurs
- Valider la feuille de renseignement

Une fois toutes ces étapes effectuées le chef de délégation est redirigé sur la page d'accueil d'inscription où il peut modifier une équipe ou en inscrire une nouvelle (Figure 11).



Figure 11: Screenshot de la partie inscription

## B -) Le générateur de tournoi

La base de données MySQL produite par l'inscription du site web est décomposée en 4 parties : délégations, teams, participants et sports. C'est avec cette base que le générateur crée la structure du tournoi suivant le type de compétition voulu : Coupe, Championnat, Poules puis phase finale, Course (Figure 12).

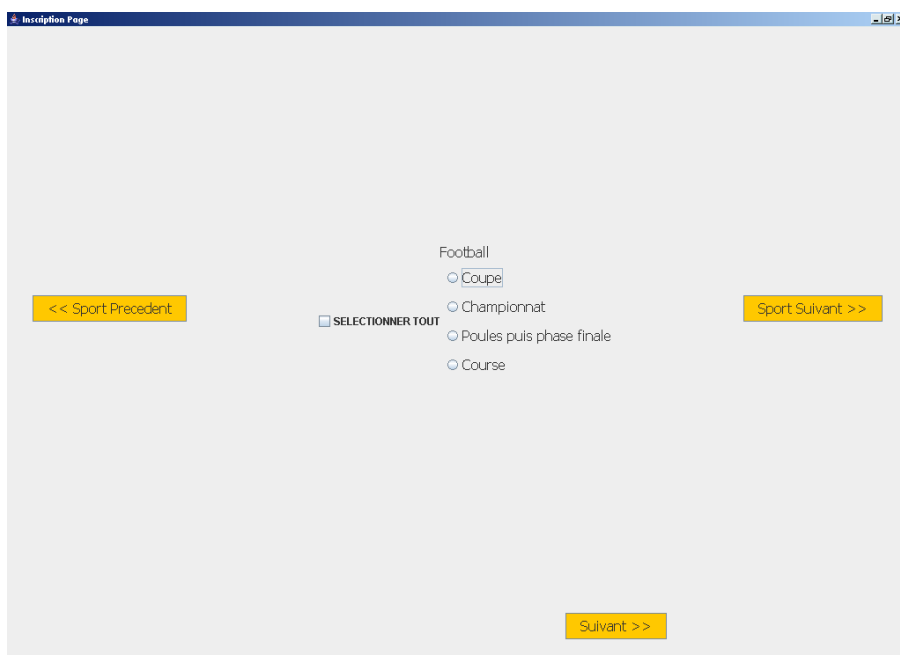


Figure 12: Screenshot de la première fenêtre du générateur de tournoi

Par la suite l'utilisateur a la possibilité de choisir le nombre d'équipes par poule, le nombre de poule... Ce programme développé par l'équipe de FI3 modifie la base de données du site web pour qu'elle soit utilisable par le manager de tournoi en formant dans la compétition générale les différents tournois (si différents sports ou si différentes catégories par exemple), les groupes et les matchs.

## C -) Le manager de tournoi

Le manager prend en paramètre la base de données générée auparavant. Il peut y avoir plusieurs compétitions c'est pourquoi il est proposé au lancement de choisir la base de données correspondant à la compétition voulue (Figure 13).

De même durant la compétition il est possible de sauvegarder l'évolution puis d'accéder à une autre base de données.

Une fois la base de données chargée, chaque tournoi de la compétition est représenté dans des onglets. A l'intérieur de chaque onglet il y a l'affichage correspondant au tournoi (coupe, groupes...). Dans le seul cas où des groupes précèdent la période de coupe finale il y a encore des onglets pour séparer la phase groupe de la phase finale (Figure 14) :

Configuration de la Base de Donnée

Veillez saisir les informations relatives à la Base de Donnée

Hostname : localhost

Host : 3306

Database name : cartel

Login : root

Pass : \*\*\*\*\*

Repeat pass : \*\*\*\*\*

Actualiser

Figure 13: Fenêtre de configuration de la base de donnée

Top Sport v0.0.0.0.0.1

Fichier Options ?

Football Hommes | Basket-Ball Femmes | Rugby Hommes | Football Test

Poules | Tournoi Final

Equipe 1	Equipe 2	Score
Equipe 1	Equipe 1	John Kerry : 26 Equipe 1 : 0
Equipe 1	Equipe 1	John Kerry : 4 Equipe 1 : 0
Equipe 1	Equipe 1	John Kerry : 0 Equipe 1 : 0
Equipe 1	Equipe 1	John Kerry : 0 Equipe 1 : 0
Equipe 1	Equipe 1	John Kerry : 0 Equipe 1 : 0
Equipe 1	Equipe 1	John Kerry : 0 Equipe 1 : 0
Equipe 1	Equipe 1	John Kerry : 0 Equipe 1 : 0
Equipe 1	Equipe 1	John Kerry : 0 Equipe 1 : 0
Equipe 1	Equipe 1	John Kerry : 0 Equipe 1 : 0
Equipe 1	Equipe 1	John Kerry : 0 Equipe 1 : 0
Equipe 1	Equipe 1	John Kerry : 0 Equipe 1 : 0
Equipe 1	Equipe 1	John Kerry : 0 Equipe 1 : 0
Equipe 1	Equipe 1	John Kerry : 0 Equipe 1 : 0
Equipe 1	Equipe 1	John Kerry : 0 Equipe 1 : 0
Equipe 1	Equipe 1	John Kerry : 0 Equipe 1 : 0

Classement des Groupes

- Groupe 1: 1 John Kerry 3, 2 Equipe 1 0
- Groupe 2: 1 John Kerry 3, 2 Equipe 1 0
- Groupe 3: 1 John Kerry 3, 2 Equipe 1 0
- Groupe 4: 1 John Kerry 0, 2 Equipe 1 0
- Groupe 5: 1 John Kerry 0, 2 Equipe 1 0
- Groupe 6: 1 John Kerry 0, 2 Equipe 1 0
- Groupe 7: 1 John Kerry 0, 2 Equipe 1 0
- Groupe 8: 1 John Kerry 0, 2 Equipe 1 0
- Groupe 9: 1 John Kerry 0, 2 Equipe 1 0
- Groupe 10: 1 John Kerry 0, 2 Equipe 1 0
- Groupe 11: 1 John Kerry 0, 2 Equipe 1 0
- Groupe 12: 1 John Kerry 0, 2 Equipe 1 0

Figure 14: Screenshot de l'application

Ainsi tous les matchs de tous les tournois sont représentés par des cellules soit dans un arbre soit dans des tableaux. Dans le cas d'un championnat où d'une phase de poule, l'écran d'affichage est séparé en deux avec d'un côté les tableaux représentant les groupes et de l'autre le classement de chaque poule. Pour les tournois types coupe, dès qu'un match est terminé, un algorithme de placement permet de directement diriger l'équipe gagnante dans l'étape suivante.

Dans les cellules de match est affiché le score. Pour modifier le score il suffit de double-cliquer sur la cellule correspondante. Une fenêtre de saisie du score s'ouvre donc en fonction du sport :

Pour le tennis un algorithme définit automatiquement si le set est terminé ou non.  
Pour le basket, il y a possibilités d'ajouter 1, 2 ou 3 points à la fois et de retirer 1 point à la fois.  
Pour les autres sports, on ajoute ou on retire un point.

Il est aussi possible d'ajouter des périodes (mi-temps). Le score affiché sur l'écran des arbres ou tableaux sera la somme des points (buts) de chaque période excepté pour le tennis où il est affiché le nombre de sets gagné. Des couleurs renseignent sur le statut du match. En effet la couleur noire est attribuée à une cellule dont le match n'a pas encore commencé ou est toujours en cours. Un match verrouillé (le verrouillage d'un match est une option disponible dans la fenêtre de saisie de score) est alors affiché en vert (Figure 15).

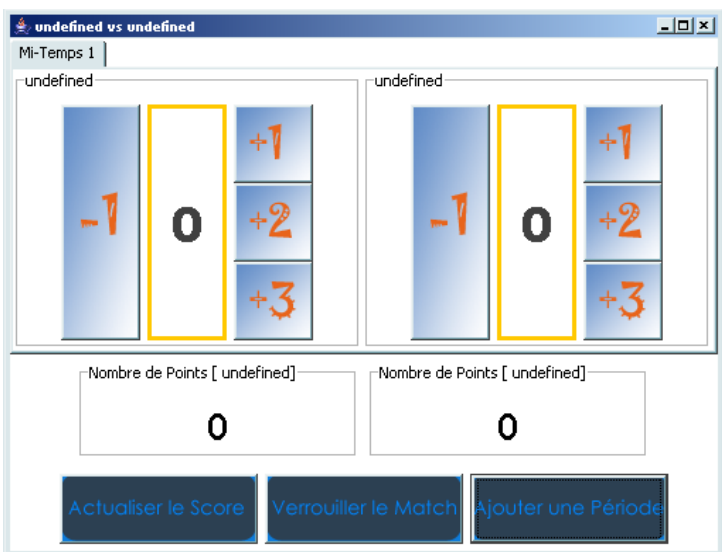
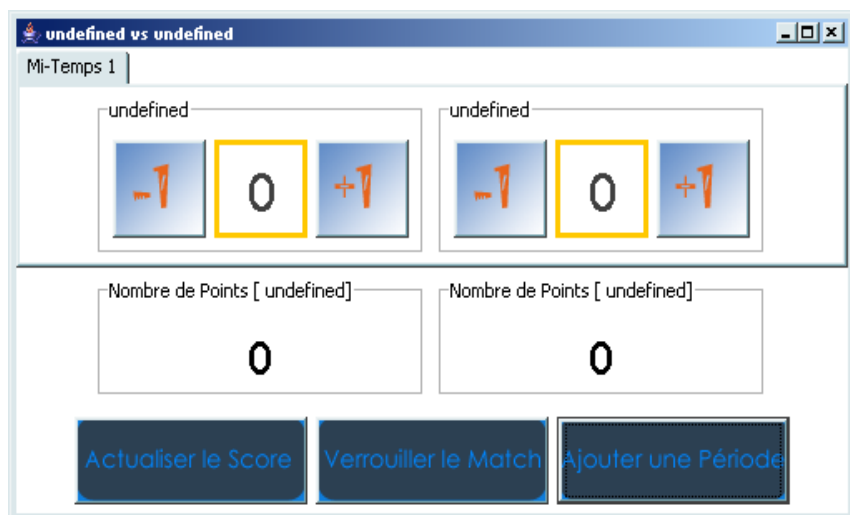


Figure 15: ScoreWindows





## 4 – Amélioration possibles

L'application que nous avons réalisé peu être améliorée sur de nombreux points. En effet, dans un premier temps, il faudrait mettre en place plusieurs niveaux de sécurité. Il serait judicieux de faire en sorte que qu'un simple utilisateur, identifié avec un mot de passe, ne puisse que modifier les scores. Un administrateur pourrait créer des utilisateurs (fournir identifiant et mot de passe) et modifier les scores. Enfin un super-administrateur pourrait accéder à toutes les fonctions du programme et notamment modifier les scores de matchs verrouillés.

Notre programme pourrait avoir la possibilité de se verrouiller après un certain temps d'inactivité.

Dans l'idéal, si notre application était utilisée sur plusieurs sites simultanément, il faudrait que toutes les instances de cette application soient connectées à un même server de base de données. Nous devrions, dans ce cas là, faire en sorte qu'un seul utilisateur puisse modifier les données à chaque fois.

D'un point de vue esthétique nous pourrions créer un menu nous permettant de choisir des thèmes de couleurs ou encore la disposition des JTabbedPane (Gauche, Droite, Haut, Bas). Il faudrait penser a tout ce qui peut faciliter l'utilisation de notre application.

Des améliorations pourraient être faites au niveau du passage de phase de groupe à phase finale. En effet il serait intéressant de pouvoir choisir le nombre d'équipes qui passent au tour suivant

## Conclusion

Lors de ce projet OSE FI2, nous avons appris beaucoup de choses, d'un point de vue organisationnel, mais aussi d'un point de vue technique.

Nous avons mis en œuvre une méthode de gestion de projet informatique qui nous a permis de mieux cibler les objectifs de notre application. Cette méthode, qui contenait un certain nombre d'étapes successives, nous a permis d'aboutir a un modèle objet avec un noyau indépendant qui est certainement le point fort de notre programme.

Aussi, même si au début le fait que nous ne soyons que trois nous a paru être un point négatif, nous avons su tirer profit de cet effectif pour nous répartir les tâches assez équitablement. Nous nous étions fixés jusqu'à janvier pour réaliser le site et ensuite commencer la réalisation de l'application. Nous avons réussi à respecter ce planning. De plus aux vues de l'application terminée et utilisable, nous en concluons que l'approximation des délais de développement étaient assez justes.

Bien entendu, l'application finale de possède pas toutes les fonctionnalités envisagées lors du listage des spécifications fonctionnelles, notamment la partie affichage des résultats. Cependant, nous pensons que si nous avons essayé d'en faire plus et de démarrer plusieurs parties en même temps, nous nous serions égarés et l'application finale n'aurait peut être pas été utilisable.

D'un point de vue technique, nous avons appris beaucoup de chose en programmation Java. En effet, n'ayant eu aucun cours d'IHM en première ou deuxième année, nous avons du apprendre par nous même Il en est de même pour toutes les autres parties. Cela a parfois été un bon moyen d'appliquer ce que nous avons appris en cours de FI2, notamment le cours de Structure de Données. Nous avons réussi à prendre un certain recul sur la programmation ce qui nous rend maintenant plus à l'aise avec le langage Java.

En fin, même si notre programme n'a pas pu être utilisé pendant le Cartel des Mines de Nantes, nous pensons qu'il pourra être facilement utilisé dans le cadre d'autre tournoi puisque nous avons n'avons pas spécialisé l'application au Cartel. Nous l'avons d'ailleurs utilisé à des fins personnelles pour créer un tournoi de tennis. Cela a été un très bon test puisqu'il nous a permis de corriger certains bugs.