



Davide Carboni et Quentin Neves

2018 – CPNV

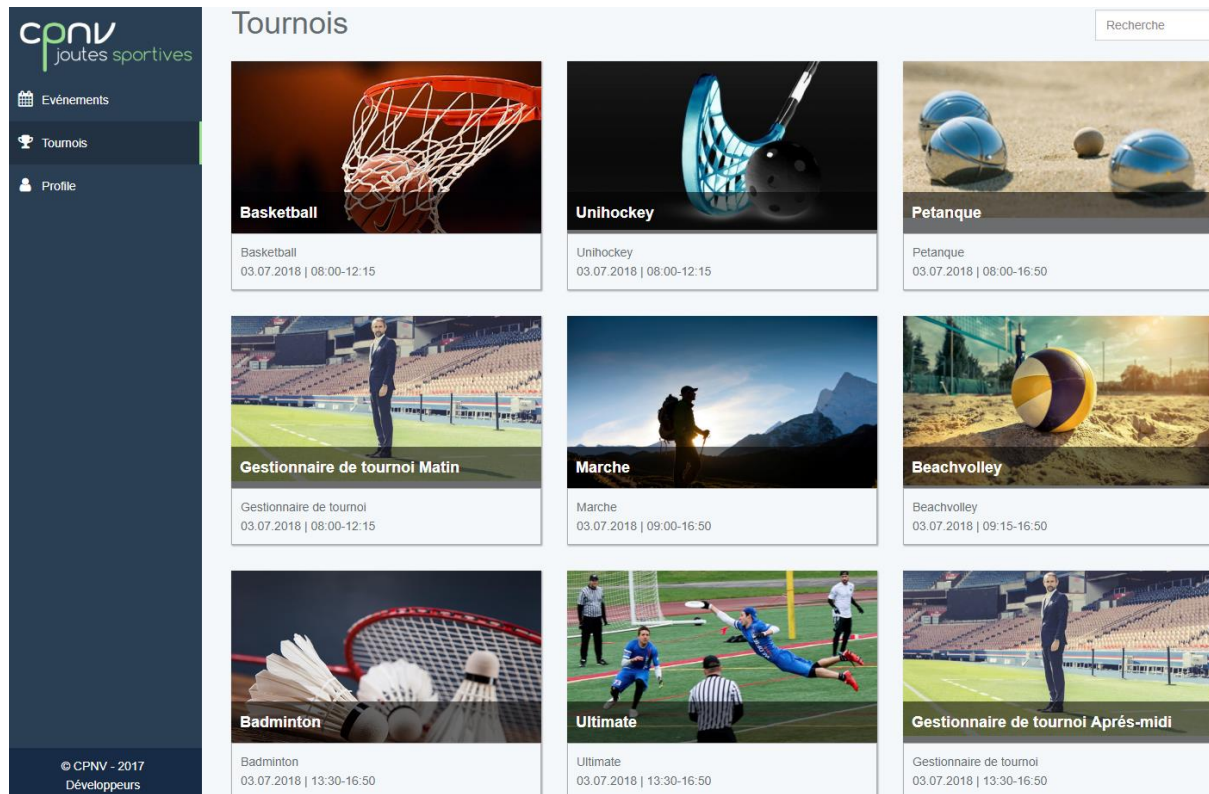
joutes.mycpnv.ch

Table des matières

Introduction	4
Objectif du projet.....	4
Description.....	4
Fonctionnalités.....	5
Le projet	6
Technologie	6
Étude de la logique	6
Analyse.....	6
Définition des objectifs	7
Login de l'utilisateur.....	7
SAML.....	7
SAML dans le contexte de l'École.....	7
Comment elle fonctionne	8
Intégration dans le projet	8
Problèmes relevés	8
Connexion Hors Ligne	8
Paramètres	8
Comment elle fonctionne	9
Problèmes relevés	9
Le participant.....	9
Profil Utilisateur.....	10
Inscription	10
Problèmes relevés	10
Le menu Profil.....	11
La fonction "afficher équipe"	11
Problèmes relevés.....	11
La fonction New.....	11
Problèmes relevés.....	12
Le menu de Changement d'Équipe	12

Modifications concernant l'administration	12
Limite d'équipe.....	12
Limite de tournoi	12
Heure de fin du tournoi	12
Création d'équipe.....	13
Génération de tournoi.....	13
Code.....	13
Création des contendres	14
Création des games (matches).....	14
Difficultés particulières	14
Compréhension de la logique d'un tournoi.....	14
Complexité de la génération de tournoi	14
Améliorations restantes	15
Bugs restants	15
Conclusion	16

Introduction



Le projet Joutes est une application dédiée à la gestion d'événements sportifs, conçue et développée au sein de l'école CPNV. Aujourd'hui, après trois ans de mise au point et de réalisation de tests, l'application a finalement été mise à la disposition de l'école pour la première fois, de sorte qu'elle puisse être utilisée par les étudiants et les professeurs.

Objectif du projet

Le développement de l'application est réalisé dans le cadre de MAW¹ 1.2. L'idée de base est de développer des applications de nature formatrices mais qui soient en même temps utiles à l'école.

Description

Chaque année, l'école organise un événement sportif appelé Les Joutes Sportives. Les étudiants en CFC sont obligés de participer à toute la durée d'un événement qui se déroule au cours d'un jour scolaire. L'autre partie des étudiants (Technicien ES et FPA de la deuxième année) et les professeurs peuvent aussi y participer, mais sans obligation. Chaque événement comporte plusieurs tournois, un pour chaque discipline sportive. Chaque participant doit s'inscrire à un tournoi en présentant une équipe, à l'exception de la marche et du groupe de gestion des tournois. Le déroulement d'un tournoi de ces joutes diffère de celui d'un tournoi classique, les équipes qui devraient être éliminées se retrouvent au tour suivant avec des équipes ayant obtenu un résultat similaire. Ce faisant tout le monde joue le même nombre de matchs.

¹ Module d'Application Web

Fonctionnalités

L'application doit permettre d'automatiser l'ensemble du système sous-tendant un événement, à la fois du point de vue de l'administration et du point de vue de l'utilisateur.

En particulier, l'application permet à l'administrateur du système de :

- Créer un événement
- Créer un tournoi
- Définir les terrains où les matchs auront lieu
- Créer les différents types de sports
- Vérifier et gérer les équipes inscrites
- Vérifier et gérer les participants inscrits

Du point de vue de l'utilisateur, l'application permet de :

- Effectuer un enregistrement sur le système
- Créer des équipes dans un tournoi
- S'inscrire dans une équipe existante
- Gérer son équipe
- Changer d'équipe
- Voir le résultat des différentes rencontres

Du point de vue de la logique des tournois, l'application doit :

- Créer automatiquement les différentes pools en fonction du nombre d'équipes
- Créer les liens entre les équipes et les pools
- Créer les matchs d'après ces liens
- Calculer les résultats des pools et placer les équipes en fonction au tour suivant

De plus, l'application permet de considérer une troisième figure : celle du Writer. De son point de vue, l'application permet de :

- Inscrire les scores des matchs
- Change l'heure d'un match

Il est également possible d'utiliser le mode plein écran (écran géant) pour que le public puisse visualiser en temps réel les résultats de différents tournois.

Le projet

Le projet Joutes est né officiellement en 2015. Tout au long de son développement, il a été suivi par trois équipes différentes de programmeurs. Bien que la logique de développement soit largement dictée par le même framework Laravel, il est toujours normal de trouver des différences de style dans le code source. Chaque programmeur a sa façon de penser et de concevoir le code. C'est pourquoi il n'est pas facile de reprendre et de comprendre le code source d'un projet développé par des tiers sans documentation expliquant le fonctionnement de l'application. Une analyse s'est alors rendue nécessaire au moment de la livraison du projet.

Technologie

L'application a été développée en utilisant le framework Laravel. Écrit en PHP, Laravel permet de créer des applications web de façon rapide et facile. Les autres technologies web usuelles (HTML 5, CSS 3 et JavaScript) sont aussi utilisées pour la réalisation de ce projet.

Étude de la logique

La première étape utile pour la reprise de l'application a été celle de l'analyse du code. Elle permet d'identifier et de comprendre comment les différentes fonctions de l'application ont été regroupées et implémentées. De plus, cela permet de voir ce qui reste encore à faire.

Analyse

Dans une première analyse, il a été immédiatement constaté que l'application :

Pour ce qui est de la partie utilisateur :

- ne permet pas d'effectuer une inscription au système
- ne permet pas de s'inscrire à un événement ou à un tournoi
- ne permet pas de créer, personnaliser et gérer une équipe

Pour la partie administrateur :

- ne permet pas de créer des équipes
- ne permet pas de définir une limite minimale et maximale dans les équipes
- ne permet pas de définir une limite maximale de participation aux tournois
- ne permet pas de vérifier l'inscription pour les étudiants lorsqu'il existe une obligation de participation
- ne donne pas d'informations claires sur la situation en ce qui concerne l'état de l'équipe
- ne permet pas d'interagir de façon directe avec les participants

Au niveau de la logique :

- la logique de création des pools ne reflète pas celle requise par l'organisation des événements
- il n'y a pas de fonctionnalité permettant d'automatiquement générer un tournoi, il faut le faire manuellement
- elle ne prévoit pas la figure du participant en tant qu'entité active au sein du système

Définition des objectifs

Le résultat de l'analyse a permis de définir les priorités et les fonctions manquantes à développer. Pour résumer, pour que l'application puisse être complétée, il faudra :

Utilisateur :

- Mettre à disposition un système de Login de l'utilisateur utilisant les comptes des élèves du CPNV
- Créer un système d'inscription aux événements pour les utilisateurs
- Créer une section "Profil" utilisateur
- Donner à l'utilisateur la possibilité de gérer l'équipe à laquelle il appartient

Administrateur

- Possibilité de créer un sport
- Corriger les limites actuelles dans la création et la gestion des tournois et des équipes
- Mettre à disposition de l'administrateur un système de vérification

Logique de l'application

- Redéfinir la logique de création et de gestion des tournois
- Définir la figure du participant en tant que partie active au sein du système

Login de l'utilisateur

SAML

C'est un standard informatique pour l'échange de données d'authentification et d'autorisation (appelées assertions) entre des domaines de sécurité distincts, typiquement un fournisseur d'identité ou identity provider (une entité fournissant des informations d'identité) et un fournisseur de services ou service provider (une entité fournissant des services). Le format des assertions SAML est basé sur XML. SAML est maintenu par OASIS Security Services Technical Committee (comité technique des services de sécurité).

Le principal problème que SAML cherche à résoudre est celui du Web Single Sign-On (SSO) (authentification unique web) entre des entités appartenant à des organisations et à des domaines de sécurité différents.

SAML dans le contexte de l'École

L'application Joutes met à disposition une authentification pour les utilisateurs du type de base. Cela signifie que l'authentification est effectuée internement par l'application elle-même, c.à.d. que le nom d'utilisateur et le mot de passe sont gérés par l'application et stockés dans la base de données. Autrement dit, on ne peut pas utiliser les mêmes identifiants d'accès à l'école pour utiliser l'application. Mais comment faire pour utiliser les mêmes identifiants d'accès à l'école et avoir accès à l'application ? La seule possibilité est d'envoyer une demande au serveur de l'école pour obtenir les informations d'accès de chaque utilisateur, d'où l'utilité de SAML.

Comment elle fonctionne

SAML exige que l'utilisateur (appelé le "principal") soit enregistré auprès d'au moins un fournisseur d'identité ou identity provider (dans ce cas, l'école). L'identity provider procède à authentifier l'utilisateur. À ce point-là, le service provider s'en remet à l'identity provider pour identifier le principal. À la demande du principal, l'identity provider passe une assertion SAML au service provider sur la base de laquelle celui-ci décide si autoriser ou refuser l'accès à ses services par le principal.

Intégration dans le projet

L'intégration du système SAML a requis quelques petites modifications dans la structure de la base de données, dans la table des utilisateurs. La structure précédente ne permettait pas d'ajouter des informations telles que le courriel, le prénom, le nom et le rôle de l'utilisateur à l'intérieur de l'application, des informations essentielles non seulement pour attribuer une entité plus précise à l'utilisateur, mais aussi pour le développement des fonctions requises.

Problèmes relevés

La procédure d'implémentation du système SAML n'a créé aucun type de problème. La documentation technique se trouve sur GitHub Wiki.

Connexion Hors Ligne

La connexion SAML permet d'unifier la connexion, mais s'empêche en même temps tout utilisateur d'accéder à l'application sans pourtant disposer des identifiants d'accès valables. Cela signifie que lors de la phase de développement et d'exécution des tests l'utilisateur – dans ce cas, le programmeur – doit toujours être authentifié via SAML. Dans ce cas, il a été utile de développer un système permettant la connexion à l'application sans avoir besoin d'utiliser une authentification SAML.

Paramètres

L'idée de base est de mettre à disposition du développeur deux types d'authentification, à savoir : une Hors Ligne et une SAML. Le choix doit être défini par l'utilisateur de manière facile et rapide. Pour répondre à ce type d'exigence, on a implémenté un système qui permet d'établir le type de connexion choisie en passant par le fichier .ENV. Il suffit d'ajouter la ligne "OFFLINE=YES" ou "OFFLINE=NO" pour définir le type d'authentification.

L'authentification Hors Ligne permet en outre de choisir le type de rôle "administrateur-participant-writer" au moment de la connexion. Cela permet au développeur d'exécuter des tests sur les trois types de profils requis.

En ce qui concerne la gestion de la base de données, les utilisateurs et leurs dépendances dans les tables sont créés et détruits lors de la déconnexion. Les utilisateurs qui sont utilisés pour différents profils sont les suivants :

- ADMIN Tester
- PARTICIPANT Tester
- WRITER Tester

Comment elle fonctionne

En fait, le principe est très simple. La barre de navigation de l'application met à disposition un bouton de "Connexion". En fonction de l'authentification définie dans le fichier ENV par le programmeur, la "View" (l'"Affichage") de la page principale décide le type d'Url à associer au bouton : SAML ou Hors Ligne. Cela permet de sélectionner les deux contrôleurs différents, à savoir : SAML Controller pour l'authentification SAML et Session Controller pour la connexion Hors Ligne

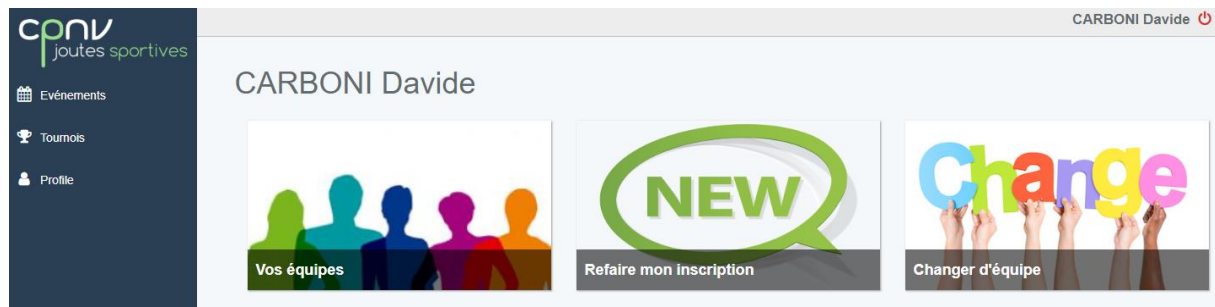
Problèmes relevés

Dans l'ensemble, aucun problème majeur ne s'est produit qui mérite d'être relevé en particulier. Ce qui a requis qu'on lui accorde plus d'attention c'est la vérification de l'existence ou non de la directive "OFFLINE=YES/NO" ("HORS-LIGNE=OUI/NON") dans le fichier .ENV. Les premiers tests ont montré comment une erreur se produisait dans l'application en l'absence de la directive. Pour résoudre ce problème, on a ajouté dans le contrôleur "Session Controller" la vérification de l'existence ou non de la directive. Cela permet au développeur web de décider d'inclure ou non la directive, sans pourtant comporter un bloc de l'ensemble de l'application.

Le participant

La structure logique de l'application ne prévoit pas le participant en tant que partie active au sein du système. Il y a une table des "Participants" dans la base de données, mais rien qui leur permette de vivre dans l'application. La première étape nécessaire a été de lier la table des participants à celle des utilisateurs. En fait, les utilisateurs enregistrés sont également des participants, à l'exception de l'administrateur et du writer. La meilleure solution eût été de fusionner les deux tables en une seule, car cela n'a aucun sens d'avoir deux tables contenant les mêmes informations. Cependant, cette solution a été écartée car l'élimination de la table des participants aurait créé des problèmes pour l'application mobile créée et en cours de développement par une équipe informatique de deuxième année. Pour résoudre ce problème, un champ "user_id" a été ajouté dans la table qui permet de lier les utilisateurs et les participants.

Profil Utilisateur



Le deuxième élément utile pour donner naissance à l'utilisateur a été de créer une page de "Profil de l'utilisateur". Ladite page permet de faire interagir l'utilisateur avec l'application en lui offrant la possibilité de réaliser et de gérer ses inscriptions aux tournois.

Inscription

L'inscription est obligatoire pour chaque utilisateur enregistré. Cela n'a aucun sens, dans le contexte scolaire, de mettre à disposition une application permettant d'effectuer le login sans disposer d'une quelconque inscription à un tournoi. Ce choix a donc été imposé. La page du profil n'est accessible et visible qu'après l'accès à l'application. La page propose un formulaire d'inscription aux tournois pour les utilisateurs qui ne sont pas encore inscrits ou un menu de sélection pour ceux qui sont déjà inscrits.

Problèmes relevés

La création du système d'inscription a posé quelques problèmes du point de vue de la validation. Le système de validation déjà présent dans le code est divisé en deux parties, à savoir : une partie en JavaScript, qui vérifie la présence ou non des informations dans le "champ de texte", et une partie en PHP confiée au système de validation de Laravel, qui vérifie le contenu des informations de chaque domaine individuel. La reprise de ce type de logique a été, dans un premier moment, avantageuse, mais ensuite elle s'est avérée inconfortable et difficile à gérer. Le formulaire d'inscription doit, au départ, contenir uniquement la liste des événements disponibles. Ce n'est qu'à un moment ultérieur, en fonction du choix de l'utilisateur, que les tournois et les équipes peuvent être proposés par

un choix cohérent avec la sélection elle-même. Dans u premier moment, on a opté pour utiliser la requête de type Ajax, qui permet de préparer les champs du formulaire en fonction des choix effectués, laissant la partie de la validation au contrôleur chez Laravel, qui vérifie que le choix soit correct uniquement si certaines conditions sont remplies. Par exemple, on ne peut pas accepter des inscriptions pour des tournois qui se sont déjà terminés ; on ne peut pas choisir une équipe déjà complète ; et on ne peut pas créer une équipe déjà existante ni participer à deux tournois en même temps. La validation Laravel est pratique, mais, une fois effectuée, le contrôleur exécute une redirection vers la page du "profil de l'utilisateur" si tout s'est bien passé, sinon il propose à nouveau le formulaire d'inscription pour répéter le choix en cas d'erreur. La redirection vers le formulaire en cas d'erreur présente un désavantage : les données précédemment chargées dans les champs respectifs seront perdues. Pour obvier à ce problème, il faut reprendre les valeurs précédemment sélectionnées et passées au contrôleur pour les réinsérer dans le formulaire, de façon à ramener celui-ci à l'état précédent dans lequel il était avant la validation, tout en restant dans le contrôleur. Cela a généré deux systèmes qui sont essentiellement identiques, sur deux fronts différents, à savoir : un en JavaScript qui, à travers les requêtes Ajax, remplit les champs du formulaire, et un en PHP, qui fait la même chose, mais après la validation. Pour résoudre ce problème, il a été décidé d'intégrer la partie de validation en JavaScript. L'avantage est que le formulaire n'est jamais envoyé, et donc abandonné, jusqu'à ce que les conditions nécessaires pour sa validation ne soient remplies.

Le menu Profil

Le menu concernant le profil de l'utilisateur est affiché une fois l'inscription complétée lors d'un événement. Elle permet de :

- Obtenir des informations sur l'inscription
- Réintroduire les paramètres de l'inscription
- Effectuer un changement d'équipe

La fonction “afficher équipe”

Cette fonction permet d'afficher les détails de son équipe, comme, par exemple, le nombre de participants, leurs coordonnées de contact, tel que courriel, etc. En particulier, elle permet seulement au propriétaire, qui par définition est celui qui a créé l'équipe, d'effectuer des changements, tels que l'élimination de participants de l'équipe.

Problèmes relevés

Il n'y avait pas eu de problèmes majeurs dans la mise en œuvre de cette fonction. Le seul problème relevé était que, dans un premier moment, il n'était pas possible d'implémenter la fonction de gestion d'équipe pour le propriétaire, du fait qu'il n'était pas défini dans la base de données. Pour résoudre ce problème, il a suffi d'ajouter le champ "owner_id" dans la table "teams" ("équipes"), qui fait référence justement à l'utilisateur qui a créé l'équipe.

La fonction New

Le menu “new” permet de réintroduire les paramètres de l'inscription. Il peut être nécessaire dans le cas où un utilisateur aurait besoin, par exemple, d'annuler son inscription à plusieurs tournois pour s'inscrire à un seulement.

Problèmes relevés

Le seul problème relevé s'est produit dans le cas d'une réinitialisation pour les propriétaires d'équipe. Une équipe sans propriétaire ne peut plus être gérée. Il est donc nécessaire de redéfinir un propriétaire lors de l'abandon. Pour résoudre le problème, le système fournit à la première équipe dans la liste le rôle de propriétaire. Si l'équipe ne contient pas de participants, et qu'elle est donc vide, alors l'équipe est éliminée de la base de données.

Le menu de Changement d'Équipe

Le menu de Changement d'Équipe permet à l'utilisateur de changer d'équipe. L'implémentation s'est avérée nécessaire pour faire en sorte que l'utilisateur ne soit pas obligé de réinitialiser l'intégralité de l'inscription lorsque le transfert est fait par une équipe d'un tournoi vers une autre équipe d'un autre tournoi. Le formulaire utilise la même logique utilisée pour l'inscription.

Modifications concernant l'administration

En ce qui concerne le côté "administrateur" de système, il a été nécessaire d'intervenir sur trois éléments importants, à savoir :

- Ajouter une limite à l'équipe au moment de la création
- Ajouter une limite au tournoi au moment de la création
- Ajouter une heure de fin du tournoi au moment de la création
- Mettre à disposition la possibilité de créer une équipe

Limite d'équipe

Le choix de la limite maximale pour une équipe a été une exigence dictée par le client. Il s'agit donc d'une fonction requise. La création n'a entraîné aucun type de problème. Il a suffi de modifier le formulaire de base déjà existant et le contrôleur correspondant pour résoudre le problème. De plus, le champ "max" a été ajouté à la table "teams" pour accueillir les nouvelles modifications.

Limite de tournoi

Dans ce cas aussi, il s'agit d'une exigence du client. Pour qu'elle soit implémentée, il a fallu modifier le formulaire déjà existant pour résoudre le problème. De plus, le champ "max" a été ajouté à la table des "tournaments" ("tournois") pour accueillir les nouvelles modifications.

Heure de fin du tournoi

L'heure de fin du tournoi naît d'une exigence du client, mais en même temps d'un besoin technique. En fait, il n'est pas possible d'établir si un tournoi se déroule seulement le matin, seulement l'après-midi ou tout au long de la journée. Or, pour résoudre ce problème, il a fallu apporter des modifications dans le formulaire de création du tournoi en ajoutant le champ "date de fin" et dans le contrôleur correspondant. La base de données a également été modifiée pour accueillir les nouveaux changements, en ajoutant le champ "date_fin" dans la table "tournaments" ("tournois").

Création d'équipe

La fonction Équipe n'avait pas encore été implémentée. Il s'agit de l'une des fonctionnalités encore laissées à implémenter. La création n'a pas requis d'efforts particuliers.

Génération de tournoi

Cette fonctionnalité soulagerait grandement les personnes gérant le tournoi en créant pour eux toutes les pools et les matchs à partir des équipes inscrites à un tournoi.

Le remplissage d'un tournoi dans la base de données touche à plusieurs tables, donc plusieurs modèles. Il aurait été très complexe et pas très correct de modifier ces modèles pour implémenter des méthodes supplémentaires qui ne font pas partie du crud.

Pour simplifier le traitement, mais aussi la compréhension d'un tournoi, il a été décidé de créer un modèle spécifique à leur création : `TournamentSetup.php` comme expliqué auparavant.

La majeure partie de la logique d'un tournoi se trouve entre les **pools** et les **contenders** (lien entre les pools et les équipes). Le passage d'une pool à une autre pour une team se décide en fonction du classement de la pool et du champ `rank_in_pool` dans la table **contenders**.

Les contenders permettent de mettre en place le tournoi en entier sans avoir à connaître l'id des équipes pour les stages sauf le premier. Malheureusement, cela complique beaucoup les choses, par exemple si le nombre d'équipes inscrites n'est pas un multiple du nombre d'équipes par pool, certaines pools manqueront d'équipes pour être complètes. Une solution discutée avec le client serait de créer une équipe fantôme qui perdrait automatiquement tous ses matchs.

Mais cette solution apporte un autre problème. S'il manque 3 équipes pour terminer la dernière pool de 4 équipes, cela fera donc une pool composée d'une vraie équipe qui sera propulsée directement en première place du stage suivant, et l'équipe se retrouvant dernière se retrouvera dans cette pool, puis propulsée à la pool des premiers du stage suivant.

L'algorithme se complexifiant rapidement, il n'a pas pu être terminé à temps et c'est pourquoi il reste dans la feature **Add_generateTournament**.

Code

Il est important de détailler certaines parties du code de ce modèle, ou plutôt les raisonnements qui expliquent les choix faits dans celui-ci.

Puisque pour chaque élément il y en a plusieurs à créer, la boucle foreach serait idéale, mais puisque la plupart des éléments n'existent pas encore, il faut connaître à l'avance ces informations :

- Nombre d'équipes inscrites au tournoi
- Taille des pools
- Nombre de stages

À partir de ça, on peut générer tout le tournoi en une seule fois. Il suffit d'avoir l'**id** du tournoi à générer pour les récupérer et ainsi calculer le reste des informations.

Création des contendere

La création des contendere est un peu plus complexe. Représentant le lien entre une équipe et une pool et se référant aux pools du stage précédent et de l'actuel, chaque champ est rempli dynamiquement.

```
if ($c % $nbPoolsPerStage == 0) $rank++;
$contender->rank_in_pool = ($s) ? $rank : null;
```

Cette condition ne fonctionne que si `$nbPoolsPerStage` est un multiple de `$nbTeamsPerPool` puisqu'elle se base sur le seed du tournoi de badminton.

La condition "compte" le nombre de contendere traité, dans le cas du seed utilisé, il y a 2 pools pour chaque rang de 1 à 4, donc le rang changera tous les 8 contendere. Ça ne s'applique pas au premier stage du tournoi puisqu'il n'y a pas encore de classement défini.

Création des games (matches)

La création des games est relativement simple, seule l'attribution des contendere complique le processus.

```
$game->contender1_id = $contenders[$s][$c+($p * $nbTeamsPerPool)]->id;
$game->contender2_id = $contenders[$s][$c+$g+($p * $nbTeamsPerPool)]->id;
```

Ici, `$p * $nbTeamsPerPool` sert à changer de pool pour avoir le "set" suivant de contendere à utiliser. On ajoute aussi `$g` pour "sauter" les équipes avec lesquelles les matches ont déjà été créés.

Difficultés particulières

Compréhension de la logique d'un tournoi

N'ayant pas de documentation expliquant ceci, il a fallu parcourir le code manuellement pour comprendre les raisons de la modélisation de la base de données et la logique d'un tournoi pour pouvoir commencer à travailler sur la génération de tournoi.

Complexité de la génération de tournoi

L'analyse avait relevé une grande complexité pour cette tâche qui semblait réalisable. Mais les contraintes techniques qui sont apparues et les spécifications quant aux demandes du client ont rendu la réalisation de cette fonctionnalité trop complexe pour les compétences disponibles. Une réanalyse au milieu de la réalisation aurait permie d'orienter les efforts sur d'autres fonctionnalités réalisables.

Améliorations restantes.

Certaines améliorations restent à faire, telles que des fonctions qui ont été expressément demandées par le client, mais qui n'ont pas été implémentées pour des raisons de temps :

Partie administrateur

- Possibilité de vérifier les inscriptions obligatoires avec la notification relative
- Possibilité d'envoyer des notifications à un groupe d'utilisateurs ou à des équipes individuelles
- Possibilité d'éliminer un tournoi ou une équipe directement à partir de la page index des Affichages (Views) respectifs

Partie utilisateur

- Implémenter un système de notification au moment de l'élimination d'une équipe
- Implémenter un système de messagerie au sein de l'application
- Un système d'affichage des notifications faites par l'administrateur

Logique de l'application

- Possibilité d'attribuer un délai maximal aux inscriptions (date d'expiration)
- Permettre aux utilisateurs n'ayant pas l'obligation d'inscription complète d'opter pour le matin ou le soir seulement.
- Date et heure des matchs générées automatiquement

Bugs restants

Il y a plusieurs bugs à résoudre :

- Lors de l'élimination d'une équipe d'un tournoi par l'administrateur, l'impression d'écran des équipes et des participants génère une erreur.
- Aucune limite maximale des caractères n'est pas définie en ce qui concerne la création du nom de l'équipe, ce qui signifie que, au moment de la création, une chaîne de 3000 caractères génère une erreur dans le système.
- Aucune limite minimale n'est prévue en ce qui concerne la longueur du nom d'une équipe.
- La redéfinition d'un propriétaire d'équipe n'est pas prise en compte lorsque l'administrateur le supprime d'une équipe.
- Les accents français ne sont pas acceptés dans les différents formulaires (par exemple, le caractère « è »).

Conclusion

L'application Joutes est déjà une réalité. Pour la première fois, il a été utilisé par l'école avec un énorme succès. Tous les utilisateurs ont pu effectuer l'inscription de manière rapide et pratique et sans problème. L'objectif principal a donc été atteint : automatiser le système d'inscription des jeux scolaires. L'autre objectif n'a cependant pas pu être réalisé, mais le travail investi ne sera pas perdu puisqu'il permettra aux étudiants suivants de mieux comprendre ce qu'implique la génération d'un tournoi. Cela a été une grande satisfaction pour toute l'équipe, qui a travaillé dur pour donner vie au projet. C'est un projet qui, dans son ensemble, représente quatre mois d'étude et de recherche, et qui nous a permis d'approfondir nos connaissances et nos compétences techniques. Le travail en groupe a mis en évidence l'importance de la collaboration et de la communication à chaque étape du projet. De plus, il nous a également permis d'entrer en contact avec une entité majeure d'un projet : un client réel ne possédant que peu de connaissances techniques. Un projet qui s'inscrit donc dans un contexte réel, où le calendrier et les besoins de développement doivent être respectés pour répondre aux attentes du client. Un facteur très important qui nous a rendus bien plus conscients du développement du projet.