

# Projet WEB : WebArena

## Spécification fonctionnelle

### Objectif

Créer un site web de jeu en ligne, type combat d'arène multijoueurs. On pourra, par exemple :

- ➔ s'inscrire
- ➔ créer un combattant
- ➔ faire réaliser diverses actions à ce combattant : se déplacer, attaquer, etc.
- ➔ avoir diverses informations sur ce combattant : sa position, ses caractéristiques, ses perceptions, etc.

### Détail

Les informations qui devront être gérées par le système sont, à minima, les suivantes. :

- ✓ Joueurs : id, email, mot de passe crypté (<http://www.php.net/manual/en/faq.passwords.php>)
- ✓ Combattants : id, nom, joueur, position, niveau, expérience, portée de la vue, force, points de vie maximaux, points de vie actuels, une image avatar ; guildes\*, date d'action\*,
- ✓ Événements : date, position, description.
- ✓ Éléments de décors\* : id, nom, type, position.
- ✓ Objets\* : id, nom, position, combattant, type, bonus.
- ✓ Messages\* : id, expéditeur, destinataire, message
- ✓ Guildes\* : id, nom
- ✓ Paramètres : cette table clef valeur permettra de stocker les données de jeu (taille du terrain, pv de départ, vue de départ, etc.)

\* Pour les options.

Une base de donnée commune sera fournie.

## Spécification technique

### Techniques

Le projet devra utiliser correctement les technologies suivantes :

- CakePHP, version 3.x.
- HTML5
- CSS3

- JQuery, si du javascript est utilisé
- PHP 5.5.9 ou plus
- Mysql 5.1.10 ou plus
- Linux, attention à la casse des fichiers. (Windows/Mac sont excessivement permissifs.)

## Pages

Le projet devra présenter, à minima, les pages suivantes :

- A) Une page « Accueil » qui présentera le jeu.
- B) Une page « Connexion ». Elle devra offrir les actions (formulaires) suivantes : s'inscrire, se connecter, récupérer son mot de passe.
- C) Une page « Combattant ». Elle devra présenter la feuille de personnage. Elle devra offrir les actions suivantes : passer de niveau, choisir un avatar, ou en cas de mort : recréer un combattant.
- D) Une page « Vision ». Elle affichera les combattants et les objets du décors en vue classés par distance croissante. Action : se déplacer, attaquer (dans une direction).
- E) Une page « Journal ». Elle présentera les événements à portée de vue de moins de 24h.

Chaque page devra offrir une ergonomie optimale et l'ensemble des pages devront respecter une charte graphique commune, avec un menu et une navigation claire. En particulier, toutes les pages du site devront être accessibles par la navigation, et aucun lien ne doit être brisé. Le nom du groupe et les noms de famille des membres, ainsi que les options choisies devront apparaître dans un footer présent sur toutes les pages.

Lors du développement il sera **obligatoire** d'utiliser un système de suivi de version, type subversion(svn) ou (git par exemple). Il faudra livrer avec le projet le fichier de log du gestionnaire avec mention des fichiers modifiés (pour svn, il s'agit du résultat de `svn log -v`, pour git, il s'agit du résultat de `git log`). Le fichier devra s'appeler versions.log et un lien dans le footer de toutes les pages devra y mener.

## Règles du jeu

Les règles suivantes seront implémentées :

- Un combattant se trouve dans une arène en damier à une position X,Y. Cette position ne peut pas se trouver hors des dimension de l'arène. Un seul combattant par case. Une arène par site.
- Un combattant commence avec les caractéristiques suivantes : vue= 2, force=1, point de vie=5 (ces valeurs doivent être paramétrées). Il apparaît à une position libre aléatoire.
- Constantes paramétrées et valeurs de livraison : largeur (x) de l'arène (15), longueur (y) de l'arène (10) (ces valeurs doivent être paramétrées dans un modèle).
- La caractéristique de vue détermine à quelle distance (de Manhattan =  $x+y$ ) un combattant peut voir. Ainsi seuls les combattants et les éléments du décor à portée sont affichés sur la page concernée. 0 correspond à la case courante.

- La caractéristique de force détermine combien de point de vie perd son adversaire quand le combattant réussit son action d'attaque..
- Lorsque le combattant voit ses points de vie atteindre 0, il est retiré du jeu. Un joueur dont le combattant a été retiré du jeu est invité à en recréer un nouveau.
- Une action d'attaque réussit si une valeur aléatoire entre 1 et 20 (d20) est supérieur à un seuil calculé comme suit :  $10 + \text{niveau de l'attaqué} - \text{niveau de l'attaquant}$ .
- Progression : à chaque attaque réussie le combattant gagne 1 point d'expérience. Si l'attaque tue l'adversaire, le combattant gagne en plus autant de points d'expériences que le niveau de l'adversaire vaincu. Tous les 4 points d'expériences, le combattant change de niveau et peut choisir d'augmenter une de ses caractéristiques : vue +1 ou force+1 ou point de vie+3. En cas de progression, les points de vie maximaux augmentent ET les points de vie courants remontent au maximum.
- En pratique, on incrémentera le niveau que lorsqu'une augmentation sera prise, et on utilisera  $(xp/4) - \text{niveau}$  pour savoir s'il reste des augmentations à prendre. Le niveau commence et les points d'expérience commencent à 0.
- Chaque action provoque la création d'un événement avec une description claire. Par exemple : « john attaque bill et le touche ».

## Options supplémentaires

Vous devez intégrer dans votre projet 3 des options listées ci-dessous (au moins 1 parmi les améliorations et 1 parmi les bibliothèques externes). Vous pouvez en implémenter davantage, cela donnera lieu à un bonus. Les options choisies devront être indiquées clairement dans le footer de toutes les pages. Attention, il sera fait en sorte que toutes les options soient prises donc indiquez vos préférences en donnant une liste ordonnée de toutes les options, premier arrivé premier servi.

### ***Option A : Gestion avancée des combattants et de leur équipement (amélioration)***

Un joueur doit pouvoir avoir plusieurs combattants. Une page particulière lui permet de choisir lequel il joue.

Le système doit pouvoir gérer une nouvelle catégorie d'objets : les pièces d'équipements. Ces pièces doivent pouvoir se trouver dans l'arène, une action ramasser permet de les équiper. Chaque pièce augmente une caractéristique (l'équivalent de une à trois progression) quand elle est portée. Les valeurs possibles sont V(vue) D(force) ou L(point de vie). En fonction du type et du bonus indiqué en base de donnée, votre programme pourra proposer un nom inventif (force+1 : dague ; force+2 : épée, etc.). Une action spécifique (un bouton) permettra de créer une dizaine d'objets aléatoires dans le terrain de jeu (pour des raisons de simplicité dans ce projet le bouton sera disponible sur la page de vue.).

### ***Option B : Gestion de la communication et de guilde (amélioration)***

Le système doit permettre d'envoyer un message à un autre combattant. Il doit aussi ajouter l'action crier qui permet de créer un événement avec une description à saisir.

Le système doit gérer la possibilité de créer et ou de rejoindre une guild. Un combattant qui attaque une cible gagne +1 en attaque par autre membre de sa guild au contact de sa cible.

### **Option C : Gestion d'une limite temporelle (amélioration)**

Le système doit limiter le nombre d'actions par unité de temps. Pour cela on utilisera un concept de points d'action. Chaque action(attaquer/se déplacer) consommera un point d'action. On récupérera un point d'action par durée donnée (temps de récupération), et on pourra en accumuler maximum un certain nombre (maximum de points d'action). Ces deux valeurs (temps de récupération en seconde et maximum de points d'action) devront être définies par des constantes facilement modifiables par le correcteur. Valeur de livraison : temps de récupération = 10 secondes , maximum de points d'action=3.

On utilisera une seule valeur en base indiquant la date correspondant à 0 points d'action (date de référence). On en déduira par rapport à la date actuelle le nombre de points d'action disponibles, en appliquant un maximum. Lors de la dépense d'un point, soit on augmentera la date de référence du temps de récupération, soit, si l'on est en dépassement, on calculera une nouvelle date de référence en fonction de la date actuelle et du nombre de points qui devraient rester.

### **Option D : Gestion d'éléments du décors (amélioration)**

Les éléments du décors sont créés aléatoirement lors de l'initialisation d'une arène. Un bouton dans la page « vue » (pour des raisons de simplicité) permettra de régénérer les éléments de décors.

Les éléments de décors seront de trois types (vous pouvez en ajouter) :

- colonne : un combattant ne peut pas passer sur la case. Une colonne pour 10 cases. La colonne sera notée « P » en base.
- Piège invisible : Le combattant meurt automatiquement quand il passe sur cette case. Le piège n'est pas affiché dans la vue du combattant. En revanche, lorsqu'il se trouve à 1 case de distance, une mention « Brise suspecte » apparaît dans la vue sur la case où se trouve le combattant. Un piège pour 10 cases. Le piège sera noté « T » en base.
- Monstre invisible : Le combattant meurt automatiquement quand il passe sur cette case, mais une attaque le détruit automatiquement. Il n'est pas affiché dans la vue du combattant. En revanche, lorsqu'il se trouve à 1 case de distance, une mention « Puanteur » apparaît dans la vue sur la case où se trouve le combattant. Un seul monstre invisible. Le monstre sera noté « W » en base

### **Option E : Interface avancée (bibliothèque externe)**

Extra : Mettre en place des librairies avancées de représentation des données.

- Utilisez le plugin datatables (<http://datatables.net/>) pour jquery pour **tous** les tableaux du site (presque toutes les pages) pour offrir un classement dynamique, coté client.
- Ajoutez une page « hall of fame » en espace public où vous présentez une liste de statistiques sur les caractéristiques, les dates de connexion etc, en utilisant au moins 4 « charts » de jqplot.
- Ajoutez sur la page d'accueil un slider des avatars

- Représentez la vue sous la forme d'un damier avec les images qui vont bien et faites apparaître un tooltip au survol.

Normal : Sans ce bonus, les tables sont classées par date ou par nom selon votre choix, et les relevés temporels sont simplement classés dans un tableau ordonné par date.

Références techniques : <http://dev.mysql.com/doc/refman/5.0/fr/group-by-functions.html>, <http://jquery.com/>, <http://datatables.net/>, <http://www.jqplot.com/>

### ***Bonus F : Utilisation de Bootstrap (bibliothèque externe)***

Extra : Utilisez bootstrap 3 (attention c'est une version récente, beaucoup de tutoriels en ligne concernent la version 2) pour la composition de vos pages. Respectez les conventions bootstrap, utilisez les classes et les id de bootstrap dans votre HTML et utilisez la css personnalisée **la plus restreinte possible** (10 lignes max).

Si vous voulez vous pouvez utiliser la v4 qui est en cours de stabilisation mais vérifiez bien le résultat sur Firefox.

Le site doit alors être *responsive*. Installez vraiment les fichiers bootstrap dans votre projet, ne passez pas par le CDN (ou juste pour récupérer les sources).

Normal : vous utilisez votre propre charte graphique, avec vos CSS et vos Javascript (Jquery si utile).

Références techniques : <http://getbootstrap.com/>

### ***Option G : Utilisation de Foundation***

Extra : Utilisez Foundation 6 pour la composition de vos pages. Respectez l'esprit du Framework et utilisez la css personnalisée **la plus restreinte possible** (30 lignes max). Le site doit alors être *responsive*.

Normal : vous utilisez votre propre charte graphique, avec vos CSS et vos Javascript (Jquery si utile).

Références techniques : <https://foundation.zurb.com/sites.html>

## **Bonus**

En plus des options ci-dessus, les groupes qui mettront en places les réalisations suivantes gagneront des points de bonus. Ces aspects ne seront pas approfondis en cours, il faudra donc mener vos propres recherches pour les implémenter. C'est la raison pour laquelle ils sont hors barème.

### ***Mise en ligne sur Internet (2 points)***

Le site devra être disponible et fonctionnel sur un hébergement accessible sur Internet. Il existe des hébergements gratuits. L'école propose également des hébergements, mais la compatibilité avec notre Framework est inconnue.

L'adresse d'hébergement devra être indiquée dans le footer de toutes les pages.

## **Recommandations méthodologique**

C'est un projet complet, et le risque est de perdre de vue l'objectif global et de se retrouver bloqué dans les méandres. On peut, par exemple, passer deux jours à « jouer » avec les css et manquer de temps pour implémenter les traitements vraiment importants.

Voici quelques recommandations de travail :

- Votre premier objectif doit être que toutes les fonctionnalités soient implémentées. La qualité de leur implémentation (ergonomie, optimisation, beauté du code) peut attendre un second passage.
- Travaillez en équipe : répartissez-vous le travail, mais dès que vous bloquez plus d'une demie heure sur un problème passez-le à un autre membre de l'équipe et passez à un autre sujet. Alternativement, travaillez à deux, un qui code, l'autre qui relit en temps réel et cherche dans la documentation. Changez de poste toutes les demie heures.
- Même si vous prenez de l'avance sur les séances respectez l'ordre de celles-ci :
  1. Apprenez les conventions du framework (vous gagnerez énormément de temps).
  2. Navigation entre les pages vides (pour pouvoir travailler ensuite en parallèle).
  3. Mise en place des modèles pour extraire les informations de la base de donnée et remplir les pages avec du contenu.
  4. Mise en place des formulaires d'action utilisateur, des pages de traitement et des modèles pour modifier la base de données.
  5. Mise en place de la charte graphique de toutes les pages et utilisation des Templates de vue (view, element, layout, helpers). Si vous vous concentrez sur la charte graphique trop tôt vous risquez de devoir revenir dessus lorsque vous intégrerez vos contenus et formulaires.
  6. Mise en place du login/password, le plus tard possible. Une fois que l'on doit se connecter pour chaque action et que certaines pages nous sont interdites, on perd beaucoup de temps pendant le développement.