Projet Qawi – Jeu de guerre galaxique

Table des matières

1. Présentation 2

1.1. Générale 2

1.2. Objectif du jeu 2

1.3. Description des fonctionnalités 2

1.4. Limite du périmètre fonctionnel 3

2. Gestion de projet 4

2.1 Diagramme de Gantt 4

2.2. Versioninng avec Git Hub 5

2.3. Prévision des derniers travaux à réaliser : 5

2.4. Contraintes techniques : 5

3. Modélisation 6

3.1 Interface Human Machine : 6

3.2. Diagramme de cas d’utilisation 8

3.3. Descriptifs cas d’utilisation 9

3.5. Algorithmes Complexes Imaginés 14

3.6. MCD 16

# 1. Présentation

## 1.1. Générale

|  |  |
| --- | --- |
| **Nom du Projet** | **QAWI** |
| **Logo** |  |

## 1.2. Objectif du jeu

Le jeu consiste à gagner des ressources pour construire des bâtiments et/ou usines. Grâce aux ressources produites en temps réels, les joueurs peuvent construire des vaisseaux pour attaquer une planète ennemie ou encore défendre sa propre planète.

## 1.3. Description des fonctionnalités

Un utilisateur est un joueur ou un administrateur.  
Un joueur est une personne ayant un nom de joueur, une adresse mail et un nombre de points initialisé à 0.  
Un utilisateur s’inscrit via une page d’inscription dédié et saisie son adresse mail, nom de joueur, mot de passe et choisie une Galaxie existante.  
Un utilisateur peut se connecter en sélectionnant une galaxie dans laquelle il a créée son compte.  
  
Un joueur peut modifier son adresse mail et son nom d’utilisateur.  
Un joueur peut voir son classement uniquement dans la galaxie dans laquelle il se trouve.  
  
Un administrateur peut voir le classement des utilisateurs en choisissant une galaxie.  
Un administrateur peut modifier le nom du jouer, sa galaxie et son mot de passe.

Une galaxie possède un identifiant et un nom.  
  
Un vaisseau peut être d’attaque ou de défense.

Un vaisseau possède un nom, un prix en ressources, une valeur défensive et une valeur d’attaque.  
  
Un regroupement de vaisseaux constitue une défense.  
Un regroupement de vaisseaux constitue une attaque.

Une usine contient un nom et un niveau.  
Un type d’usine produit un type de ressource.  
  
Le niveau d’une usine peut être amélioré par un joueur en échange d’un nombre de ressources dans le cas où le joueur n’a pas assez de ressources alors il ne peut pas améliorer l’usine et doit attendre d’avoir assez de ressources.  
  
Une ressource peut être de l’or, de l’argent, de la pierre ou de la nourriture.  
  
Un récaputalif est un tableau récapitulant les deux dernières activités sur le compte d’un joueur (Amélioration et/ou attaques).  
  
Pour passer d’un niveau usine à un autre, l’on doit multiplier le niveau actuel par la somme du niveau actuel de l’usine plus 3.  
La création d’un vaisseau d’attaque ou de défense coûte des ressources. (Cela dépend des types de vaisseaux).

## 1.4. Limite du périmètre fonctionnel

Le jeu est limité par le fait que l’utilisateur n’a pas beaucoup de fonctionnalité à faire si ce n’est améliorer des usines, créer des vaisseaux et attaquer des joueurs avec une flotte.  
Le joueur est également limité aux temps d’attentes d’amélioration d’une usine.  
Le joueur ne peut pas faire plusieurs choses comme par exemple créer un vaisseau et en même temps améliorer une usine.

|  |
| --- |
|  |

# 2. Gestion de projet

## 2.1 Diagramme de Gantt

Tâches et sous tâches :

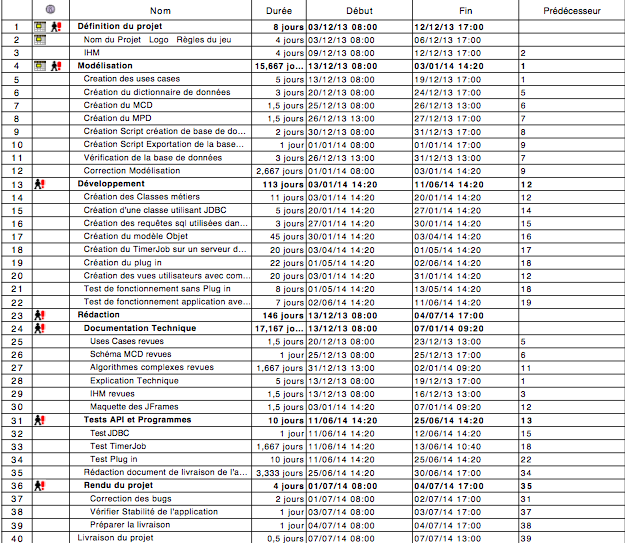
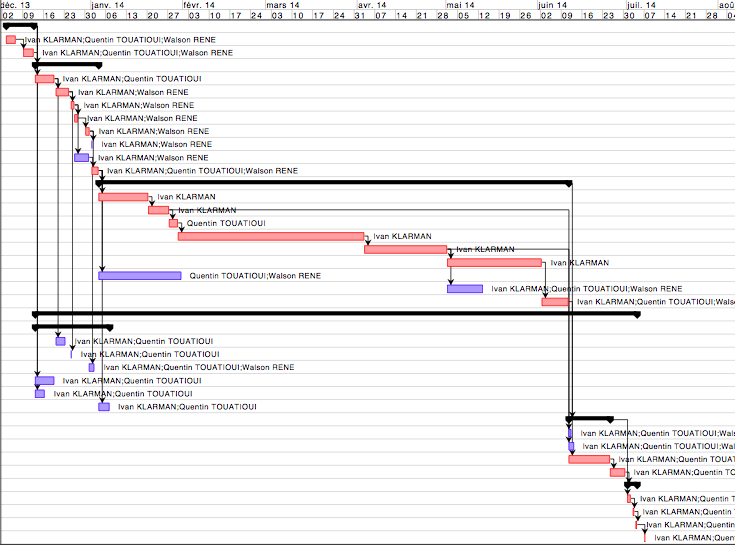


Diagramme de Gantt :



## 2.2. Versioninng avec Git Hub

Nous avons utilisé Git Hub pour la partie versionning, cela nous a permis de travailler sur nos fichiers de manières autonomes, nous avons pu partager également les fichiers mis par les différentes personnes du groupe pour s’assurer d’avoir les dernières versions des fichiers. De plus en terme de management, cela à permis de veiller à l’avancement du projet notamment sur ce qui as été fait.

Au 24/06/2014  **🡪 43 Commits**dont : **Ivan Klarman 🡪** 20 **Commits**

**Quentin Touatioui** 🡪 8 **Commits**

**Walson RENE 🡪** 15 **Commits**

## 2.3. Prévision des derniers travaux à réaliser :

Les futurs rendues à faire par l’ensemble des collaborateurs de l’équipe projet sont :  
1 – Création des plug-in  
2 – Tests des plug-in  
3 – Implémentation des JFrames créaient avec les contrôleurs  
4 – Assemblage des différentes parties précédemment réalisées   
5– Le cahier de livraison  
6 – Préparation à la soutenance

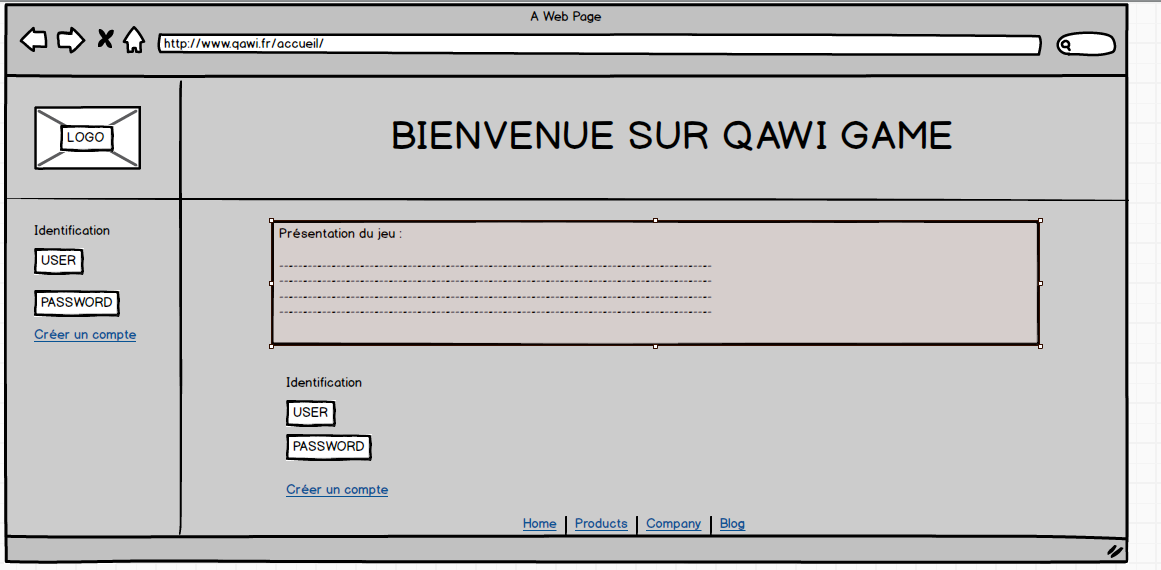
## 2.4. Contraintes techniques :

Les contraintes techniques ont été identifiées dans le syllabus du projet annuel fournis par notre professeur Mr Sananes.

# 3. Modélisation

## 3.1 Interface Human Machine :

PAGE D'ACCUEIL



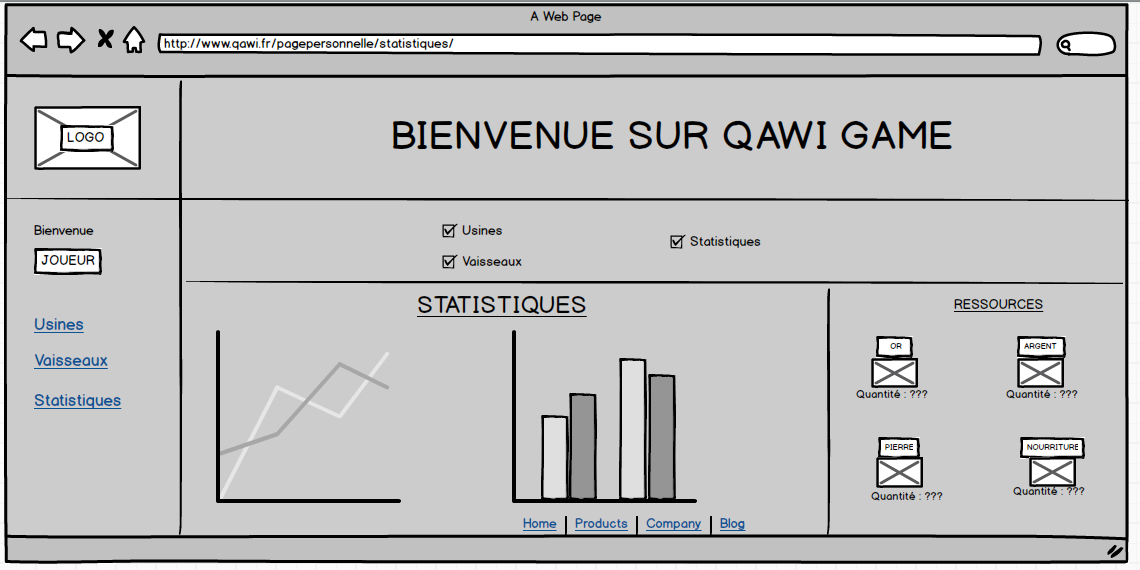
PAGE CREATION D'UN NOUVEAU COMPTE



PAGE "RESSOURCES"



CONSULTATION DES STATISTIQUES DU JOUEUR



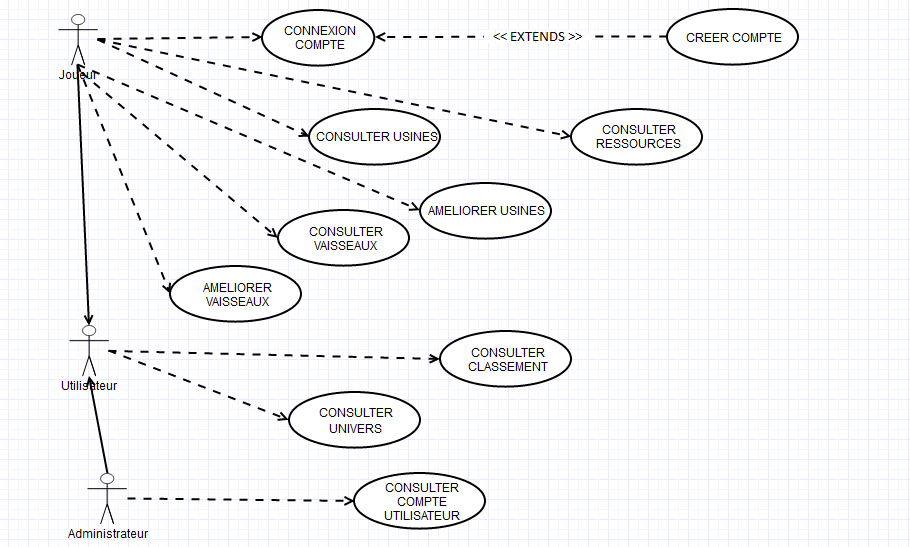
PAGE "USINES"



PAGE "VAISSEAUX"



## 3.2. Diagramme de cas d’utilisation

****

## 3.3. Descriptifs cas d’utilisation

**CAS : « CREER UN COMPTE » :**Auteur : Quentin TOUATIOUI  
Date Modification : 03/04/2014  
Démarrage :   
 A l’initiative du visiteur  
Pré-Condition :

Le visiteur se trouve sur la page du jeu.  
Dialogue :   
Scénario nominal :

|  |  |
| --- | --- |
| **UTILISATEUR** | **SYSTEME** |
|  | 1) Affichage de la page "Création d'un compte" |
| 2) Création d'un identifiant et Mot de passe |  |
|  | 3) Vérification identifiant non existant dans BDD |
|  | 4) Nouveau compte utilisateur créé |

Scénarios Alternatifs :   
A1. Après 2) ◊ Identifiant déjà existant - Retour au point 1)

**CAS : « CONNEXION COMPTE »**:   
Auteur : Ivan KLARMAN  
Date Modification : 02/04/2014  
Démarrage :   
 A l’initiative du visiteur  
Pré-Condition :   
 Le visiteur doit se connecter

Dialogue :   
Scénario nominal :

|  |  |
| --- | --- |
| **UTILISATEUR** | **SYSTEME** |
|  | 1) Affiche la page de connexion |
| 2) Saisir les identifiants |  |
|  | 3) Vérifie si l’utilisateur est en BDD |
|  | 4) Renvois vers la page personnel |
|  | 5) Affichage message de bienvenue |

Scénarios Alternatifs :   
A1. Après 3) 🡪 Aucun utilisateur n’est trouvé - Retour au point 1)

**CAS : « CONSULTER UNIVERS»**:   
Auteur : Ivan KLARMAN  
Date Modification : 02/04/2014  
Démarrage :   
 A l’initiative de l’utilisateur  
Pré-Condition :   
 L’utilisateur doit se connecter   
Dialogue :   
Scénario nominal :

|  |  |
| --- | --- |
| **UTILISATEUR** | **SYSTEME** |
| 1) Cliquer sur le lien « Galaxie » |  |
|  | 2) Vérifier toutes les Galaxies en BDD |
|  | 3) Renvoyer vers la page « Univers » |
|  | 4) Affichage des Galaxies |
| 5) Filtrer les Galaxies |  |
|  | 6) Vérifier le filtre en BDD |
|  | 7) Affichage des Galaxies avec le filtre |

Scénarios Alternatifs :   
A1. Aucune Galaxie après 2), Affichage message « Aucune Galaxie »

**CAS : « CONSULTER CLASSEMENT»**:   
Auteur : Ivan KLARMAN  
Date Modification : 02/04/2014  
Démarrage :   
 A l’initiative de l’utilisateur  
Pré-Condition :   
 L’utilisateur doit se connecter.  
Dialogue :   
Scénario nominal :

|  |  |
| --- | --- |
| **UTILISATEUR** | **SYSTEME** |
| 1) Cliquer sur le lien « Classement » |  |
|  | 2) Vérifier tous les utilisateurs en BDD avec leurs points gagnés. |
|  | 3) Renvoyer vers la page « Classement » |
|  | 4) Affichage du Classement par ordre de nombre de points. |
| 5) Filtrer le Classement |  |
|  | 6) Vérifier le filtre en BDD |
|  | 7) Affichage des Utilisateurs avec le filtre |

**CAS : « CONSULTER COMPTE UTILISATEUR»**:   
Auteur : Ivan KLARMAN  
Date Modification : 02/04/2014  
Démarrage :   
 A l’initiative de l’administrateur  
Pré-Condition :   
 L’administrateur doit se connecter  
Dialogue :   
Scénario nominal :

|  |  |
| --- | --- |
| **UTILISATEUR** | **SYSTEME** |
| 1) Cliquer sur le lien « Administration Comptes » |  |
|  | 2) Vérifier tous les utilisateurs en BDD. |
|  | 3) Renvoyer vers la page « Administration Comptes» |
|  | 4) Affichage des comptes utilisateurs avec leurs identifiants et mots de passe par ordre alphabétique du nom d’utilisateur. |
| 5) Filtrer par nom d’utilisateur ou id |  |
|  | 6) Vérifier le filtre en BDD |
|  | 7) Affichage des Utilisateurs avec le filtre |
| 8) Modification des données |  |
|  | 9) Changement des données en BDD |
|  | 10) Message de modification effectué |
| 10) Fermer le message de modification |  |
|  | 11) Retour page personnel |

**CAS : « CONSULTER RESSOURCES »**:   
Auteur : Quentin TOUATIOUI  
Date Modification : 03/04/2014  
Démarrage :   
 A l’initiative du visiteur  
Pré-Condition :   
 L'utilisateur doit être connecté.   
Dialogue :   
Scénario nominal :

|  |  |
| --- | --- |
| **UTILISATEUR** | **SYSTEME** |
|  | 1) Vérifier ressources utilisateur dans BDD |
|  | 2) Affichage des ressources |
| 3) Consultation des ressources |  |

Scénarios Alternatifs :   
(Vide ◊ L'utilisateur peut toujours consulter ses ressources)  
**CAS : « CONSULTER USINES »**:   
Auteur : Quentin TOUATIOUI  
Date Modification : 03/04/2014  
Démarrage :   
 A l’initiative du visiteur  
Pré-Condition :   
 L'utilisateur doit être connecté.

Dialogue :   
Scénario nominal :

|  |  |
| --- | --- |
| **UTILISATEUR** | **SYSTEME** |
| 1) Choix : Visualisation "Usines" |  |
|  | 2) Vérifier niveaux des usines dans BDD |
|  | 3) Affichage des usines |
| 4) Consultation des usines |  |

Scénarios Alternatifs :

(Vide ◊ Il est toujours possible de consulter ses usines)

**CAS : « AMELIORER USINES »**:

Auteur : Quentin TOUATIOUI  
Date Modification : 03/04/2014  
Démarrage :   
 A l’initiative du visiteur  
Pré-Condition :

Le visiteur se trouve sur la page du jeu.  
Dialogue :   
Scénario nominal :

|  |  |
| --- | --- |
| **UTILISATEUR** | **SYSTEME** |
| 1) Choix : Visualisation "Usines" |  |
|  | 2) Vérifier niveaux des usines dans BDD |
|  | 3) Affichage des usines |
| 4) Amélioration des usines |  |
|  | 5) Usines passent au niveau supérieur |

Scénarios Alternatifs :

A1. Après 4) ◊ Manque de ressources, Amélioration impossible - Retour au point 3)

**CAS : « CONSULTER VAISSEAUX »**:   
Auteur : Quentin TOUATIOUI  
Date Modification : 03/04/2014  
Démarrage :   
 A l’initiative du visiteur  
Pré-Condition :   
 L'utilisateur doit être connecté.

Dialogue :   
Scénario nominal :

|  |  |
| --- | --- |
| **UTILISATEUR** | **SYSTEME** |
| 1) Accéder à la page Vaisseaux |  |
|  | 2) Récupérations informations Vaisseaux |
|  | 3) Affichage des vaisseaux |

Scénarios Alternatifs :   
A1. Après 2) ◊ Aucun vaisseau disponible, affichage impossible - Retour au point 1)

**CAS : « ACHETER VAISSEAUX »**:   
Auteur : Quentin TOUATIOUI  
Date Modification : 03/04/2014  
Démarrage :   
 A l’initiative du visiteur  
Pré-Condition :   
 L'utilisateur doit être connecté.

Dialogue :   
Scénario nominal :

|  |  |
| --- | --- |
| **UTILISATEUR** | **SYSTEME** |
| 1) Choix : Visualisation "Vaisseaux" |  |
|  | 2) Vérifier niveaux des vaisseaux dans BDD |
|  | 3) Affichage des vaisseaux |
| 4) Achat de nouveaux vaisseaux |  |
|  | 5) Obtention de nouveaux vaisseaux |

Scénarios Alternatifs :   
A1. Après 4) ◊ Ressources insuffisantes - Retour au point 3)

## 3.5. Algorithmes Complexes Imaginés

A) Temps d’attentes des améliorations/constructions des usines/vaisseaux :   
  
Nous étudierons un cas avec les Usines.  
  
// Deux constructeurs possibles   
 // Premier constructeur   
Classe Usine (String nom,,int tempsAttentes, int niveau, int nbRessourcesA, int RessourcesB)  
 // Deuxième constructeur   
Classe Usine (String nom, int tempsAttentes, int niveau, int nbRessourcesA, int RessourcesB, int nbRessourceC)  
// Calcul de l’attente en fonction du niveau de l’usine  
uneUsine.getTempsAttentes = uneUsine.getNiveau()\*(uneUsine.getNiveau() + 3);  
  
***Exemple :***  
Soit si une usine est de niveau 1 pour passer au niveau 2 en l’améliorant, le calcul sera :   
Attentes = 1\*(1 + 3)  
Attentes = 4 minutes  
Soit si une usine est de niveau 2 pour passer au niveau 3 en l’améliorant, le calcul sera :  
 Attentes = 2\*(2 + 3)  
Attentes = 10 minutes  
Soit si une usine est de niveau 2 pour passer au niveau 3 en l’améliorant, le calcul sera :  
Attentes = 3\*(3+ 3)  
Attentes = 18 minutes  
ETC …

// Vérifier possibilité d’amélioré ou non  
 Boolean faisable = false ;

int Compteur = uneUsine.getTempsAttentes() ;  
// Condition de faisabilité ou non ( améliorations ou constructions)   
 SI Compteur == 0  
 ALORS faisable = true ;  
 SINON faisable = false ;  
// Si faisable Que faire ?, si non faisable Que faire ?  
 SI faisable == true  
 ALORS uneUsine.upNiveau() ;  
 SINON alert(« Une amélioration n’est pas terminée »  ) ;

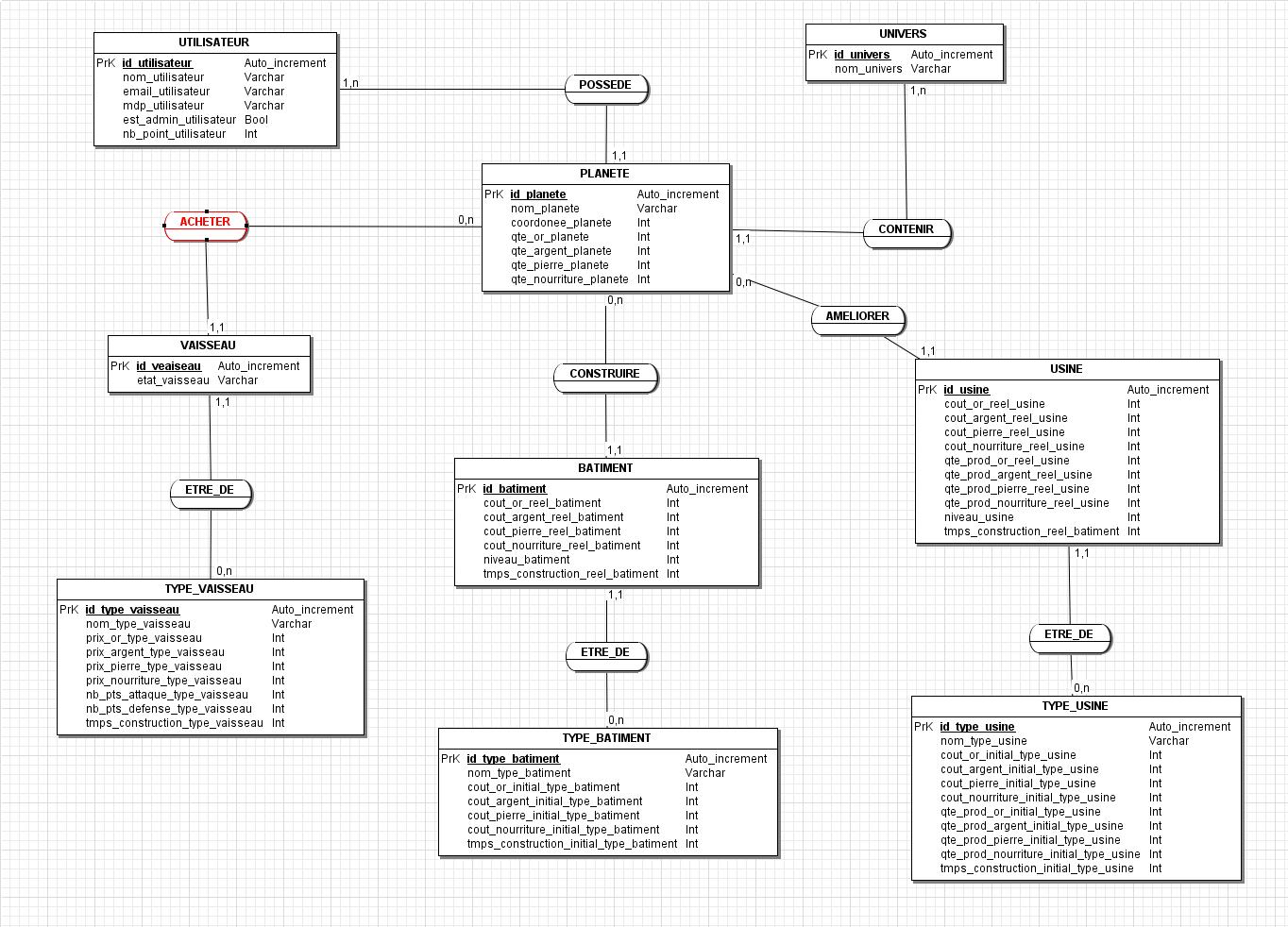
B) Algorithme d’attaque d’un joueur à un autre :   
// Méthode prévue pour l’attaque   
// @param : une liste de vaisseaux  
// @param : Un objet Joueur  
Classe Joueur avec une méthode attaquer (List<Vaisseaux> uneListeV, Joueur unJoueur) ;   
Classe Joueur avec une méthode estAttaquer (List<Vaisseaux> uneListeV, Joueur unJoueur) ;

// Dans cette méthode un calcul décrit la valeurd’attaque de chaques vaisseaux envoyé au combat et on a la valeur d’attaque final.  
 int valeurAttaqueFinal = 0 ;  
 foreach(Vaisseaux unVaisseau as uneListeV){

valeurAttaqueFinal += unVaisseau.getValeurAttaque() ;  
 }  
// Pour savoir si la personne est gagnante, on vérifie dans une autre méthode la valeur défensive de l’adversaire représenté par des vaisseaux également mais défensif  
 int valeurDéfensiveFinal = 0 ;  
 foreach(Vaisseaux unVaisseauDéfensif as uneListeV){

valeurDéfensiveFinal += unVaisseauDéfensif.getValeurDéfense() ;  
 }

## 3.6. MCD

****