

Projet Nemesis

Cahier Des Charges

1ère Version

PIERRAT Lucas

CALITSTO Vincent

MILLERET Jehan

PEREIRA Rémi

POURCHOT Mathieu

Projet tuteuré S3

2015-2016

[**Contexte** 2](#_Toc434517777)

[Description du projet 2](#_Toc434517778)

[Définition du problème 2](#_Toc434517779)

[**Objectif** 2](#_Toc434517780)

[Partie Client 2](#_Toc434517781)

[Partie Serveur 2](#_Toc434517782)

[Partie base de données 3](#_Toc434517783)

[**PERIMETRE** 3](#_Toc434517784)

[**Description fonctionnelle** 3](#_Toc434517785)

[La partie niveau client 3](#_Toc434517786)

[Écran de connexion 3](#_Toc434517787)

[Écran Mot de passe oublié 3](#_Toc434517788)

[Écrande changement de mot de passe 4](#_Toc434517789)

[Écran d'inscription 4](#_Toc434517790)

[Écran d'accueil 5](#_Toc434517791)

[Écran de création d'une partie 5](#_Toc434517792)

[ECRAN REJOINDRE UNE PARTIE 5](#_Toc434517793)

[Ecran de jeu pour le joueur 6](#_Toc434517794)

[ECRAN DE JEU POUR LE MJ 7](#_Toc434517795)

[ACTIONS SUPPLEMENTAIRES DU MJ 7](#_Toc434517796)

[LA PARTIE NIVEAU SERVEUR 7](#_Toc434517797)

[Partie creation du serveur 7](#_Toc434517798)

[Partie connexion à la base de données 8](#_Toc434517799)

[Partie gestion des connexions avec les clients 8](#_Toc434517800)

[Partie gestion des threads clients 8](#_Toc434517801)

[Base de données 8](#_Toc434517802)

[BASE DE DONNÉES 8](#_Toc434517803)

**Contexte**

Description du projet

Le projet de nom de code : Nemesis, est un projet initié dans le cadre des projets tuteurés de l'IUT de Belfort-Montbéliard. Ce projet est mené par un groupe de 5 étudiants :

* Lucas PIERRAT ;
* Rémi PEREIRA ;
* Mathieu POURCHOT ;
* Jehan MILLERET ;
* Vincent CALISTO.

Le projet porte sur la création d’un jeu de rôle ainsi que la programmation Client-Serveur.

D'une manière générale, le jeu se présentera sous la forme suivante. Un serveur sera préalablement lancé sur une machine distante. Quant au client, il sera distribué sur d'autres machines. Le joueur lance son jeu qui cherche immédiatement à joindre le serveur. Le client propose alors une salle d'attente où deux possibilités sont proposées au joueur :

* Créer une partie ;
* Rejoindre une partie en cours de création.

Une fois le nombre de joueur souhaité atteint, le MJ lance la partie. Les joueurs évolueront donc dans un univers RP / Donjon selon l'inspiration.

Définition du problème

**Objectif**

Partie Client

L'objectif de l'application côté client est de gérer un affichage du jeu. L'application côté client devra être capable de récupérer les données envoyées par le serveur. Afin de libérer le serveur de tâche lourde, les différents calculs du jeu devront être effectués côté client. Une version différente du client devra être disponible suivant le type de personne : MJ ou joueur. L'interface client du MJ devra être capable de proposer des actions prédéfinies au MJ afin que celui-ci puisse diriger la partie. L'interface client des joueurs devra s'adapter suivant les actions des joueurs et du MJ.

Partie Serveur

L'objectif de l'application côté serveur devra être capable de gérer les interactions entre les joueurs des différentes parties. En effet, plusieurs parties différentes devront pouvoir être lancées en parallèle. Afin de gérer correctement l'affichage côté client, le serveur devra être capable de communiquer avec une base de données.

Partie base de données

La base de données contiendra toute les informations nécessaires au jeu. Par exemple, on devra trouver dans la base de données les renseignements des joueurs comme le pseudo et le mot de passe. On trouvera également toute les données du jeu. Voir annexe 1 pour plus de détails.

**PERIMETRE**

L'application client sera disponible sur Windows, Linux et MacOS. Le serveur tournera sous Windows uniquement. Il y aura un serveur unique. Il n'y aura pas de limitation géographique. L'application client devra donc proposer un affichage multilingue.

**Description fonctionnelle**

La partie niveau client

Écran de connexion

Au lancement de l'application, un écran de connexion sera proposé au joueur. Cet écran comporta les éléments suivants :

* Champ 'Login' ;
* Champ 'Mot de passe' ;
* Bouton 'Se connecter' ;
* Bouton 'Mot de passe oublier' ;
* Bouton 'S'inscrire' ;
* Check-box 'Se souvenir de moi'.

|  |  |
| --- | --- |
| Objectif | Se connecter au jeu / Accéder à l'écran d'inscription / Accéder à l'écran de récupération du mot de passe |
| Description | Il faut renseigner les champs login et mot de passe. Lorsqu'on clique sur 'Se connecter', une requête est envoyer au serveur afin d'identifier le joueur qui se connecte. Une check-box permettra d'enregistrer ses identifiants afin de se connecter plus rapidement. |
| Contraintes | L'envoi du mot de passe au serveur sera crypté |
| Niveau de priorité | moyen |

Écran Mot de passe oublié

Au clique sur le bouton 'Mot de passe oublié', cet écran sera affiché. Cet écran comportera les éléments suivants :

* Champ 'Question secrète' ;
* Champ 'Réponse' ;
* Bouton 'Envoyer' ;
* Bouton 'Annuler'.

|  |  |
| --- | --- |
| Objectif | Permettre au joueur de retrouver son mot de passe. |
| Description | Il devra répondre à la question secrète qu'il aura renseignée lors de la création de son compte. Si la réponse est correcte, il pourra modifier son mot de passe directement |
| Contraintes | L'envoi du mot de passe et de la réponse secrète au serveur sera crypté |
| Niveau de priorité | faible |

Écrande changement de mot de passe

Lorsque le joueur aura renseigné sa réponse secrète, et si celle-ci est correcte, il arrivera sur cet écran. Il pourra alors renseigner son mot de passe. Cet écran comportera les éléments suivants :

* Champ 'Mot de passe' ;
* Champ ''Confirmation du mot de passe' ;
* Champ 'Confirmer'.

|  |  |
| --- | --- |
| Objectif | Permettre au joueur de changer son mot de passe |
| Description | Le joueur pourra renseigner son nouveau mot de passe et le confirmer. |
| Contraintes | L'envoi du mot de passe au serveur sera crypté |
| Niveau de priorité | faible |

Écran d'inscription

Cet écran sera affiché lorsque le joueur aura cliqué sur 'S'inscrire' sur l'écran de connexion. Il comportera les éléments suivants :

* Champ 'Login' ;
* Champ 'Mot de passe' ;
* Champ 'Confirmation de mot de passe' ;
* Champ 'Question secrète' ;
* Champ 'Réponse secrète' ;
* Champ 'Confirmation de réponse secrète' ;
* Bouton 'S'inscrire' ;
* Bouton 'Annuler'.

|  |  |
| --- | --- |
| Objectif | Permettre au joueur de s'inscrire |
| Description | Cet écran proposera au joueur de renseigner son pseudo, son mot de passe, de choisir une question et une réponse secrète. Lors du clique sur 'S'inscrire', retour à l'écran de connexion, les champs 'Mot de passe' et 'Login seront déjà renseignés |
| Contraintes | L'envoi du mot de passe et de la réponse secrète au serveur sera crypté |
| Niveau de priorité | moyen |

Écran d'accueil

Cet écran sera affiché après la connexion d'un joueur. Cet écran sera la salle d'attente des parties. Le joueur pourra voir les parties en cours, en créer une nouvelle ou rejoindre une partie en cours de création. Cet écran comportera les éléments suivants :

* Un menu (Quitter, Profil, Option, Crédits) ;
* La liste des parties en cours avec Nom de la partie, Nom du MJ, Côte du MJ, type de partie (privée ou non), langue de la partie ;
* Bouton 'Rejoindre', disponible après avoir présélectionné une partie ;
* Bouton 'Créer une partie'.

|  |  |
| --- | --- |
| Objectif | Afficher les éléments nécessaires aux joueurs pour jouer |
| Description | Cet écran devra proposer aux joueurs différentes options: accéder aux éléments du menu, accéder à la liste des parties en cours, rejoindre une partie en création, ou créer une nouvelle partie |
| Contraintes | / |
| Niveau de priorité | Haute |

Écran de création d'une partie

Lorsque le joueur cliquera sur ‘Créer une partie’, cet écran sera affiché. Il permettra au joueur de créer une nouvelle partie de jeu. Cet écran comportera les éléments suivants :

* Un menu (Quitter, Profil, Option, Crédits) ;
* Champ ‘Nom de la partie’ ;
* Champ ‘Nombre de joueur maximum’ ;
* Champ ‘Mot de passe’ (optionnel) ;
* Bouton ‘Valider’ (actif seulement si une partie a été présélectionnée).

|  |  |
| --- | --- |
| Objectif | Permettre de créer une partie |
| Description | Cet écran devra proposer au joueur de créer une nouvelle partie. Le joueur qui créé la partie sera désigné Maitre du Jeu. Si le champ ‘Mot de passe’ est renseigné, la partie sera privée. Le créateur de la partie pourra choisir le nombre de joueur maximum (jusqu’à 4, lui exclut). |
| Contraintes | Créer la partie (l’insérer dans la base de donnée), et l’afficher sur l’écran d’accueil |
| Niveau de priorité | Haute |

ECRAN REJOINDRE UNE PARTIE

Lorsqu’un joueur aura sélectionné une partie dans l’écran d’accueil et cliquer sur le bouton ‘Rejoindre la partie’, il arrivera sur cet écran. Cet écran permet au joueur de choisir le personnage qu’il incarnera durant la partie. Une liste des joueurs déjà présents dans la partie sera également affichée. Lorsque le joueur présélectionnera un personnage, un pop-up d’information indiquera les caractéristiques du personnage. Les personnages déjà sélectionné par les autres joueurs seront grisé afin de ne pas avoir de doublons. Un espace de chat permettra de chatter avec les joueurs de la partie. Cet écran comportera ainsi les éléments suivants :

* Un menu (Quitter, Profil, Option, Crédits) ;
* Une liste des joueurs déjà présents dans la partie ;
* La liste des personnages à jouer (grisé si déjà sélectionné par un autre joueur) ;
* Un pop-up d’information concernant le personnage présélectionné ;
* Un bouton ‘Choisir ce personnage’ (actif seulement si un personnage est présélectionné) ;
* Un chat.

|  |  |
| --- | --- |
| Objectif | Permettre de sélectionner un personnage et rejoindre la partie |
| Description | Cet écran permet au joueur de choisir le personnage qu’il incarnera durant la partie. Une liste des joueurs déjà présents dans la partie sera également affichée. Lorsque le joueur présélectionnera un personnage, un pop-up d’information indiquera les caractéristiques du personnage. Les personnages déjà sélectionné par les autres joueurs seront grisé afin de ne pas avoir de doublons. Un espace de chat permettra de chatter avec les joueurs de la partie. |
| Contraintes | Griser les personnages déjà sélectionnés. Gérer le choix simultané d’un même personnage par deux joueurs. Gérer le chat |
| Niveau de priorité | Haute |

Ecran de jeu pour le joueur

Lorsque les joueurs entrent dans la partie, un écran et un menu s’affichent. Le menu est composé de « options » et de « quitter ». Concernant l’écran, Les joueurs sont assis autour d’une table et le MJ est toujours en face de tout le monde. Le dialogue est gérer par des bulles de dialogue au-dessus de chaque personne et d’un résumé dans le chat. Les différentes actions sont produites par le MJ et affichés sur la table (Combat, apparition d’un PNJ, jet de dés…). Les différentes actions sont encore une fois résumées dans le chat. Les monstres et les joueurs ont un résumé des principales informations qui s’affiche à leur survol. Le décor change en fonction de l’histoire. Cet écran comportera les éléments suivants :

* Un chat ;
* Un résumé ;
* Une table de jeu ;
* Un décor ;
* Des joueurs (MJ et normaux) ;
* Un menu (Options, Quitter).

|  |  |
| --- | --- |
| Objectif | Afficher l’écran de jeu des joueurs |
| Description | L’écran principal du jeu vu par le joueur |
| Contraintes | Gérer le chat, modifier l’écran en fonction des actions du MJ et des joueurs |
| Niveau de priorité | Haute |

ECRAN DE JEU POUR LE MJ

L’écran de jeu est le même pour les joueurs et le MJ, la seule différence viens dans le menu. Des options viennent ainsi s’ajouter : Lancer un combat, Afficher un PNJ, Montrer un objet, Changer le décor et la musique. Les éléments qui changent sont :

* Un menu supplémentaire (Lancer un combat, Afficher un PNJ, Montrer un objet, Changer le décor et la musique)

|  |  |
| --- | --- |
| Objectif | Afficher l’écran de jeu du MJ |
| Description | L’écran principal du jeu vu par le MJ, avec le menu supplémentaire |
| Contraintes | Gérer le chat, modifier l’écran en fonction des actions du MJ et des joueurs |
| Niveau de priorité | Haute |

ACTIONS SUPPLEMENTAIRES DU MJ

Les différentes actions supplémentaires du MJ seront sélectionnées grâce à une liste déroulante où sont fixés les différents éléments stockés dans la base de données. Il suffit alors de cliquer sur la flèche pour ajouter un élément (par exemple un monstre). Selon l’option choisie, il est possible de sélectionner un ou plusieurs éléments. Lors de la sélection de monstres, et donc du lancement d’un combat, une musique associée est automatiquement lancée. Lors du changement de décor, la musique d’ambiance associée est également automatiquement sélectionnée.

LA PARTIE NIVEAU SERVEUR

Partie creation du serveur

Il faudra d’abord créer un serveur en initialisant une socket de connexion, sur un port et une IP donnés. Le serveur écoutera ensuite les connexions.

|  |  |
| --- | --- |
| Objectif | Lancement du serveur |
| Contraintes | Créer un serveur suivant un fichier de configuration (IP, Port …) |
| Niveau de priorité | Haute |

Partie connexion à la base de données

Le serveur devra se connecter localement à la base de données contenant les données du jeu. Il devra ensuite faire une copie mémoire de la base pour éviter des temps de traitements supplémentaires.

|  |  |
| --- | --- |
| Objectif | Charger les éléments du jeu en mémoire |
| Contraintes | / |
| Niveau de priorité | Haute |

Partie gestion des connexions avec les clients

Le serveur, écoutant sur son port, devra accepter ou non (suivant des conditions) les clients. Une fois la connexion acceptée, le serveur créera un nouveau processus pour le client.

|  |  |
| --- | --- |
| Objectif | Gestion des connexions |
| Contraintes | Gérer les droits de connexion, gestion dynamique des clients |
| Niveau de priorité | Haute |

Partie gestion des threads clients

Les processus client devront gérer les échanges de données entre les différents joueurs de la partie.

|  |  |
| --- | --- |
| Objectif | Gestion des échanges de données |
| Contraintes | / |
| Niveau de priorité | Haute |

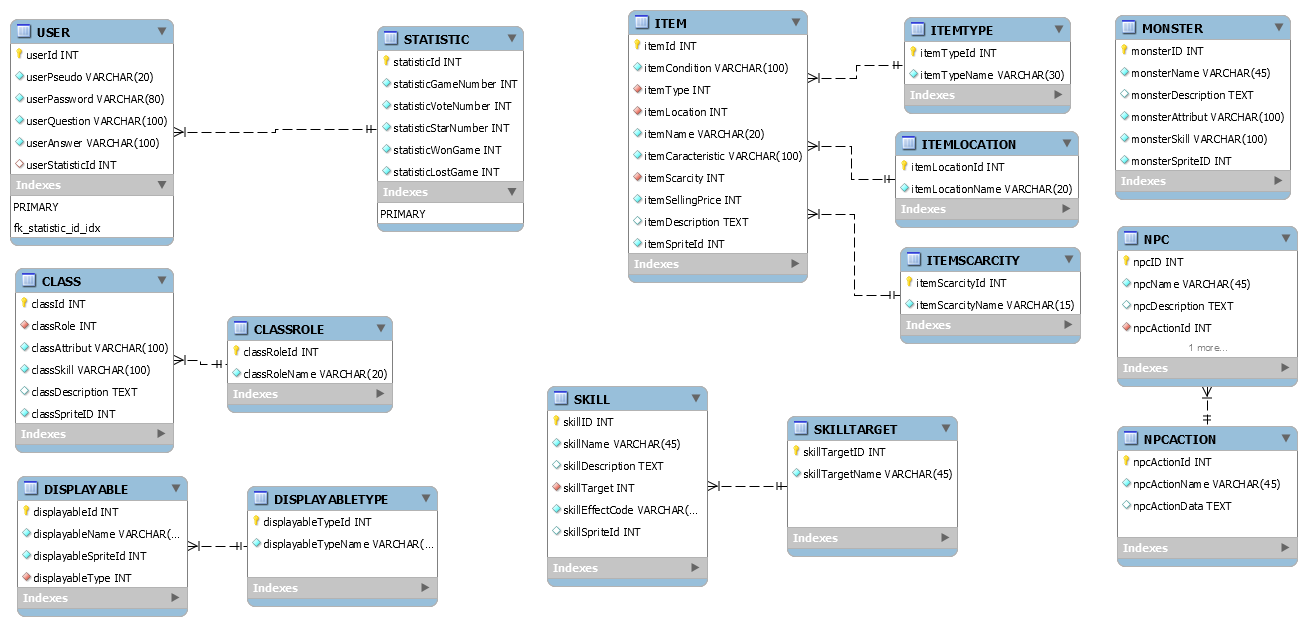
Base de données

BASE DE DONNÉES

La base de données servira à stocker les données statiques :

* Table utilisateur
* Table Item
* Table classe
* Table compétence
* Table monstre
* Table PNJ
* Table décors
* Table Dé
* Table ObjetAffichable
* Table Table
* Table BulleDialogue
* Table Interface

|  |  |
| --- | --- |
| Objectif | Stocker les données statiques |
| Description | Base de données SQL |
| Contraintes | Gestion des clés étrangères |
| Niveau de priorité | Haute |



Choix techniques

* + 1. Choix des langages et des bibliotheques

Pour programmer le serveur, nous utiliserons le C et le C++ :

* Le C pour partie connexion avec la base de données
  + La bibliothèque utilisée sera « mysqlclient »
* Le C++ pour la partie gestion du réseau, gestion des threads
  + Pour la gestion du réseau, nous utiliserons SFML NETWORK
  + Pour la gestion des thread, nous utiliserons <thread> du C++

Pour programmer le client :

* C++ combiné à la SFML :
  + SFML GRAPHICS
  + SFML AUDIO
  + SFML NETWORK
  + SFML MAIN
  + SFML SYSTEM
    1. Environnement de travail

Nous travaillerons sous l'IDE CLion de JetBrains, en utilisant à côté GitHub pour la mise en commun du travail. Lors du développement, nous utiliseront la bibliotheque Google test (gtest) pour faire du test unitaire. Le code sera compilé avec l'outil MinGW (g++) et non pas le compilateur de Visual Studio.

Pour réalisé le modèle SQL, nous avons utilisé MySQL Workbench de Oracle. Le SGBD sera donc MySQL.