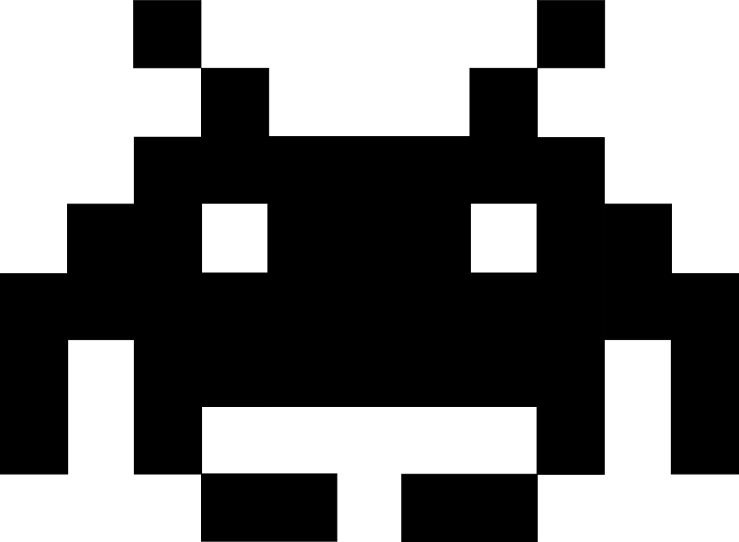
|  |
| --- |
| ETML |
| Rapport de projet |
| Projet Space invaders |

|  |
| --- |
| Evin Paramanathan  30/10/2023 |



Rapport de projet

Table des matières

[UX 2](#_Toc149557050)

[a. Introduction 2](#_Toc149557051)

[b. Analyse 2](#_Toc149557052)

[1. Conception centrée utilisateur 2](#_Toc149557053)

[2. Choix palette graphique 3](#_Toc149557054)

[3. Eco-conception 3](#_Toc149557055)

[4. Accessibilité 3](#_Toc149557056)

[c. Conception 4](#_Toc149557057)

[d. Évaluation 5](#_Toc149557058)

[DB 5](#_Toc149557059)

[a. Importer les données et le schéma de la base de données 5](#_Toc149557060)

[b. Gestions des utilisateurs 5](#_Toc149557061)

[c. Requêtes de sélection 6](#_Toc149557062)

[d. Création des index 10](#_Toc149557063)

[1. Pourquoi certain Index existe déjà 10](#_Toc149557064)

[2. Avantage inconvénients des Index 10](#_Toc149557065)

[e. Backup / Restore 10](#_Toc149557066)

[POO 10](#_Toc149557067)

[a. Introduction 10](#_Toc149557068)

[b. Analyse fonctionnelle 11](#_Toc149557069)

[c. Analyse technique 11](#_Toc149557070)

[d. Test Unitaire 11](#_Toc149557071)

[e. Chat GPT 11](#_Toc149557072)

[f. Conclusion 11](#_Toc149557073)

# UX

## Introduction

Le projet Consiste à crée un jeu très célèbre, le SpaceInvaders. Dans ce cadre-là on va parler de la partie UX. Dans ce projet Il y a plusieurs choses à faire, il y les maquettes graphique (haute et basse fidélité). Cela va servir à avoir une idée du jeu qu’on va développer. Ensuite il y a aussi les personnas qui servent à prendre du recul et se mettre à la place d’un potentiel utilisateur de notre jeu. Cela permet D’être plus objectif quand on développe notre application

## Analyse

### Conception centrée utilisateur

Création de personas

Il y a deux personnas Martin Alex et Sarah Dupuis.

Martin Alex est un étudiant en littérature de 22 ans qui aime bien jouer au jeu vidéo que ce soit retro ou moderne.

Il est travailleur, ambitieux mais surtout très mauvais perdant.

Une image contenant texte, Visage humain, homme, capture d’écran

Description générée automatiquement

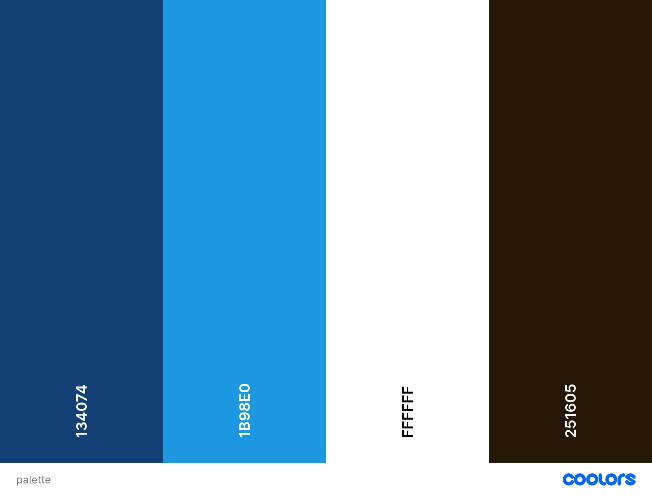
Ensuite il y a Sarah Dupuis, elle a 32 ans et elle est développeuse dans une boite à Lausanne. Elle est mariée à Marc et elle a deux enfant Lucas et Emma.

Une image contenant texte, Visage humain, homme, capture d’écran

Description générée automatiquement

Les deux personnas sont très diffèrent l’un est jeune Célibataire et fait encore des études et à part quand il joue au jeux vidéo il n’a pas tellement de lien avec le monde numérique alors que la deuxième et un peu plus âgé, est mariée, à des enfants et a déjà fini ses études. Et travaille dans le monde informatique donc Ils tous le deux une vision différente des jeux vidéo

### Choix palette graphique



### Eco-conception

Pour l’éco-conception, le jeu peut aussi être en noir et blanc afin de réduire la consommation d’énergie pour l’affichage du jeu. Dès qu’on appuie sur mode sombre tout le jeu passe en noir et en blanc.

Une image contenant texte, Appareils électroniques, capture d’écran, Ingénierie électronique

Description générée automatiquement

### Accessibilité

Sur la maquette haute-fidélité on peut constater que dans paramètre on peut changer les couleurs du jeu pour les personnes qui sont daltonien. Cela va permettre aux personnes qui ne voient pas bien les couleurs d’avoir une meilleure expérience

Une image contenant texte, capture d’écran, Appareils électroniques, multimédia

Description générée automatiquement

## Conception

1. Définition des écrans
2. Choix effectués

Maquette : Pour la maquette j’ai fait exprès de mettre le titre de la page Vaisseau en bleu et la page ennemies en rouge parce que souvent les ennemies sont représentées par le rouge car ça représente le danger. Et voir ça rend le titre un peu plus agressif

Une image contenant capture d’écran, texte, dessin humoristique

Description générée automatiquement

Dans le jeu il y a deux modes de jeu le mode solo et le mode multijoueur. Dans le mode solo c’est un space invaders classique le canon est en bas de l’écran et doit avoir toutes les ennemies qui viennent d’en haut. Le mode multijoueur est totalement différent il se joue à plusieurs donc les ennemies ne viennent pas seulement d’en haut mais peut aussi les encercler et le but est d’avoir le plus de point que sont amis. Donc dans ce mode les joueurs doivent s’entraider afin de rester le plus longtemps en vie mais doivent aussi essayer de marquer plus de points que leur ami.

Une image contenant capture d’écran, texte, télévision, Bleu Majorelle

Description générée automatiquement

1. ChatGPT

Dans cette partie du projet ChatGPT à principalement été utilisé afin de créer les personnas. Pour créer leur Biographie et à partir de cette Bio développer leur motivation, leur attitude, leurs Objectifs et leurs frustrations.

## Évaluation

1. Test

# DB

## Importer les données et le schéma de la base de données

**Docker :** docker exec -i db mysql -h 6524cdd35220a9f403be8eda44b3f03922c816bb09b36bb9a0bc20768e9568ec-P 3306 -uroot -proot < db\_space\_invaders.sql

On sélectionne la base de donner, ensuite on met l’identifiant root avec son mot de passe root et pour finir on met ce signe (<) pour dire que on importe la base de données qui dans ce cas la db\_space\_invaders.sql

## Gestions des utilisateurs

#### Crée un Utilisateur pour chaque Partie :

**Administrateur :** CREATE USER 'admin'@'localhost' IDENTIFIED BY 'admin' ;

on utilise CREATE USER pour dire qu’on créer un nouveau utilisateur ensuite on lui donne un nom et attribut un serveur, l’IDENTIFIED BY permet de donner un mot de passe a cet utilisateur.

**Joueur :** CREATE USER 'joueur1'@'localhost' IDENTIFIED BY 'joueur1' ;

La on crée le joueur1 et on lui donne le mot de passe joueur1

**Gestionnaire :** CREATE USER 'gestion'@'localhost' IDENTIFIED BY 'gestion'

Et la on crée un utilisateur qui s’appelle gestion avec son mot de passe qui est gestion.

#### Créer des rôles :

**Administrateur :** CREATE ROLE 'r\_administrateur';

Cette requête crée un rôle grâce au CREATE ROLE qu’on appelle r\_administrateur

**Joueur** **:** CREATE ROLE 'r\_joueur';

Cette requête crée un rôle qu’on appelle r\_joueur

**Gestionnaire de la boutique :** CREATE ROLE r\_gestionnaire';

Cette requête crée un rôle qu’on appelle r\_gesionnaire

#### Donner les rôles à chaque utilisateur :

**Admin :** GRANT 'r\_administrateur' TO 'admin'@'localhost' ;

Grâce a cette requête on donne le rôle r\_administrateur à l’utilisateur admin

**Joueur :** GRANT 'r\_player' TO 'joueur1'@'localhost' ;

On donne le rôle de r\_player à l’utilisateur joueur1

**Gestionnaire de la boutique :** GRANT ‘r\_manager' TO 'gestion'@'localhost';

Et là on donne le rôle r\_manager à l’utilisateur gestion

#### Attribution des rôles à tout l’utilisateur

**Admin :** Grant , select ON \*.\* To ‘r\_administrateur’ ;

**Joueur :** Grant \*, select ON \*.\* To ‘r\_player’

**Gestionnaire :** Grant UPDATE,, select ON \*.\* To ‘r\_manager’

## Requêtes de sélection

**Requêtes n°1 :**

SELECT \* FROM `t\_joueur` ORDER BY jouNombrePoints DESC LIMIT 5 ;

Une image contenant texte, Police, nombre, capture d’écran

Description générée automatiquement

Avec cette requête on sélection toutes les colonnes de la table t\_joueur et on affiche les 5 joueurs qui ont le plus de point dans l’ordres décroissant

**Requêtes n°2**

SELECT Max(armPrix) AS PrixMaximum, MIN(armPrix) AS prixMinimum, AVG(armPrix) AS PrixMoyen FROM `t\_arme`;

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

On sélection armPrix, on prend son maximum et on le renomme PrixMaximum, on prend aussi la moyenne d’armPrix et on renomme par PrixMoyen et pour finir on prend le minimum de armPrix et on le renomme Prixmoyen. Tout ça dans la table t\_arme.

**Requêtes n°3**

SELECT fkJoueur AS idJoueur, COUNT(fkJoueur) AS NombreCommandes FROM t\_commande GROUP BY fkJoueur ORDER BY COUNT(fkJoueur) DESC;

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Cette requête compte le nombre de commandes pour chaque joueur dans la table "t\_commande," puis renvoie ces informations triées par ordre décroissant du nombre de commandes.

**Requêtes n°4**

SELECT fkJoueur AS idJoueur, COUNT(fkJoueur) AS NombreCommandes FROM t\_commande GROUP BY fkJoueur HAVING COUNT(fkjoueur)>2 ORDER BY COUNT(fkJoueur) DESC;

**Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement**

Cette requête extrait des données de la table "t\_commande". Elle groupe les enregistrements par la colonne "fkJoueur" (les joueurs), compte combien de commandes chaque joueur a passé, et ne renvoie que les groupes (joueurs) qui ont passé plus de 2 commandes. Les résultats sont triés par ordre décroissant.

**Requêtes n°5**

SELECT t\_joueur.jouPseudo AS Pseudo, t\_arme.armNom AS NomArme FROM t\_commande JOIN t\_detail\_commande ON t\_commande.idCommande = t\_detail\_commande.fkCommande JOIN t\_arme ON t\_detail\_commande.fkArme = t\_arme.idArme JOIN t\_joueur on t\_commande.idCommande = t\_joueur.idJoueur;

**Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement**

Cette requête SQL effectue une jointure entre quatre tables : t\_commande, t\_detail\_commande,t\_arme, et t\_joueur. Elle sélectionne le pseudonyme d'un joueur ("Pseudo") depuis la table "t\_joueur" et le nom de l'arme ("NomArme") depuis la table "t\_arme" en utilisant des liaisons entre ces tables via les clés étrangères "fkCommande" et "fkArme" dans la table "t\_detail\_commande."

**Requêtes n°6**

SELECT t\_joueur.jouPseudo AS Pseudo, t\_arme.armNom AS NomArme FROM t\_commande JOIN t\_detail\_commande ON t\_commande.idCommande = t\_detail\_commande.fkCommande JOIN t\_arme ON t\_detail\_commande.fkArme = t\_arme.idArme JOIN t\_joueur on t\_commande.idCommande = t\_joueur.idJoueur;

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Cette requête SQL récupère le pseudo des joueurs et le nom des armes associées aux commandes en reliant les tables t\_commande, t\_detail\_commande, t\_arme, et t\_joueur à travers les relations existantes.

**Requêtes n°7**

SELECT t\_joueur.jouPseudo AS joueur, t\_commande.comNumeroCommande AS t\_commande FROM `t\_joueur` LEFT JOIN t\_commande ON t\_commande.fkJoueur = t\_joueur.idJoueur WHERE t\_joueur.jouPseudo IS NOT NULL;

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Sélectionne les pseudonymes des joueurs ("joueur") et les numéros de commande correspondants ("t\_commande") dans la table "t\_joueur" et la table "t\_commande." Elle inclut tous les joueurs ayant un pseudonyme défini, même s'ils n'ont pas passé de commande.

**Requêtes n°8**

SELECT t\_joueur.jouPseudo AS JOUEUR, t\_commande.comNumeroCommande AS t\_commande FROM `t\_joueur` RIGHT JOIN t\_commande ON t\_joueur.idJoueur = t\_commande.fkJoueur;

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Avec cette on sélectionne la table t\_joueur.jouPseudo et on le renomme par JOUEUR, ensuite on sélectionne la table t\_commande.comNumeroCommande et on le renomme par t\_commande. Tout ça de la table t\_joueur ensuite on fait une jointure droite

**Requêtes n°9**

SELECT DISTINCT j.jouPseudo AS Pseudo, SUM(dc.detQuantiteCommande) AS NombresTotalArmes FROM t\_commande AS c JOIN t\_detail\_commande AS dc ON c.idCommande = dc.fkCommande JOIN t\_arme AS a ON dc.fkArme = a.idArme RIGHT JOIN t\_joueur AS j ON c.fkJoueur = j.idJoueur GROUP BY jouPseudo HAVING jouPseudo IS NOT NULL;

**Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement**

Cette requête sélectionne le pseudonyme des joueurs et la somme des quantités d'armes associées à leurs commandes. Les données proviennent de plusieurs tables et sont agrégées grâce à des jointures et des groupements. Les résultats sont filtrés pour exclure les joueurs sans pseudonyme.

**Requêtes n°10**

SELECT DISTINCT j.jouPseudo AS Pseudo, COUNT(DISTINCT a.armNom) AS NomArme FROM t\_commande AS c JOIN t\_detail\_commande AS dc ON c.idCommande = dc.fkCommande JOIN t\_arme AS a ON dc.fkArme = a.idArme JOIN t\_joueur AS j ON c.fkJoueur = j.idJoueur GROUP BY jouPseudo HAVING COUNT(DISTINCT a.armNom) > 3;

**Une image contenant texte, capture d’écran, Police

Description générée automatiquement**

## Création des index

CREATE INDEX nom\_de\_l\_index ON nom\_de\_la\_table

On Crée un index grâce au CREATE INDEX, ensuite on lui donne un nom et grâce aux ON on sélectionne la table qu’on veut indexer

### Pourquoi certain Index existe déjà

Mysql crée automatiquement certaine Index comme pour les clé primaire, clé étrangère et pour les champs unique.

### Avantage inconvénients des Index

Avantage : Avoir des index permet d’effectuer des recherches plus rapides.

Inconvénients : Ils ralentissent certaines requêtes comme l’insertion, modification et la suppression.

## Backup / Restore

**Backup :** Mysqldump -u root -proot –databases db\_space\_invaders > BackupInvaders.sql

Mysqldump est utilisée pour sauvegarder une base de données MySQL nommée "db\_space\_invaders" avec l'utilisateur "root" et le mot de passe "root". La sauvegarde est stockée dans un fichier nommé "BackupInvaders.sql".

**Restore** **:** Mysql -u root -proot db\_space\_invaders < backup.db\_space\_invaders.sql

Ça nous permet d'importer une sauvegarde fichier SQL dans une base de données MySQL nommée "db\_space\_invaders" en utilisant l'utilisateur "root" avec le mot de passe "root

# POO

## Introduction

Dans le Projet SpicyInvaders nous avons aussi une partie C# qui est de donné vie à ce projet grâce à l’orienté objet.

## Analyse fonctionnelle

## Analyse technique

1. Diagramme de classe
2. Explications

## Test Unitaire

## Chat GPT

## Conclusion