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

|  |
| --- |
| Evin Paramanathan  06/10/2023 |

Rapport de projet

Table des matières

[UX 2](#_Toc147480466)

[a. Introduction 2](#_Toc147480467)

[b. Analyse 2](#_Toc147480468)

[1. Conception centrée utilisateur 2](#_Toc147480469)

[2. Choix palette graphique 2](#_Toc147480470)

[3. Eco-conception 2](#_Toc147480471)

[4. Accessibilité 2](#_Toc147480472)

[c. Conception 2](#_Toc147480473)

[d. Évaluation 2](#_Toc147480474)

[DB 2](#_Toc147480475)

[a. Importer les données et le schéma de la base de données 2](#_Toc147480476)

[b. Gestions des utilisateurs 2](#_Toc147480477)

[c. Requêtes de sélection 3](#_Toc147480478)

[d. Création des index 6](#_Toc147480479)

[e. Backup / Restore 6](#_Toc147480480)

[POO 6](#_Toc147480481)

[a. Introduction 6](#_Toc147480482)

[b. Analyse fonctionnelle 6](#_Toc147480483)

[c. Analyse technique 6](#_Toc147480484)

[d. Test Unitaire 6](#_Toc147480485)

[e. Chat GPT 6](#_Toc147480486)

[f. Conclusion 6](#_Toc147480487)

# UX

## Introduction

Le projet Consiste à crée un jeu très célèbre, le SpaceInvaders. Dans ce cadre la 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),

## Analyse

### Conception centrée utilisateur

* 1. Création de personas

### Choix palette graphique

### Eco-conception

### Accessibilité

## Conception

1. Définition des écrans
2. Choix effectués
3. ChatGPT

## Évaluation

1. Test

# DB

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

Docker exe -i

## 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' ;

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

#### Créer des rôles :

Administrateur : CREATE ROLE 'Administrateur';

Joueur : CREATE ROLE 'Joueur';

Gestionnaire de la boutique : CREATE ROLE 'Gestionnaire';

#### Donner les rôles à chaque utilisateur :

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

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

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

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

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

Joueur :

Gestionnaire :

## 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

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

On sélection fkjoueur et on change son nom par Id Joueur

**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**

**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**

**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

**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

**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

**Requêtes n°9**

**Requêtes n°10**

## Création des index

## Backup / Restore

# 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