

ETML

Rapport de Projet SpicyInvaders

CID2B

Mateen Salem Khaiy
06/10/2023

Table des matières

UX – User Expérience	3
Introduction.....	3
Analyse	3
Conception centrée utilisateur.....	3
Choix de la palette graphique	5
Eco-conception	5
Accessibilité	6
Conception	6
Définition des écrans.....	6
Choix effectués	6
Evaluation.....	8
Tests.....	8
DB – Data base	9
A. Importer les données et le schéma de base de données.....	9
B. Gestion des utilisateurs.....	10
Administrateur :	10
Joueur :	11
Gestionnaire de la boutique :	12
C. Requêtes de sélection	13
D. Création des index.....	17
E. Backup / Restore	17
Backup	17
Restore	18
F. Liaison database & C#	19
POO – Programmation orientée objet	19
A. Introduction.....	19
B. Analyse fonctionnelle	19
Démarrer une partie.....	19
Bouger Joueur	19
Tirer missile	20
Apparaître alien.....	20
Afficher menu.....	20
Afficher score	20
Afficher les hp joueur	21

GameWin.....	21
GameLoose.....	21
C. Analyse technique	21
Diagramme de classes	21
Explication DocFX	22
D. Tests Unitaire.....	22
E. Utilisation ChatGPT	24
F. Conclusion	25

UX – User Expérience

Introduction

Dans la partie UX de ce rapport, ce trouve une explication du projet UX qui vise à crée un prototype interactif qui comprends des menus et des sous menus cliquables. Le projet UX doit avoir les points suivants :

1. Possibilité de choisir entre le mode jeux solo et multijoueur
2. Pouvoir choisir différentes palettes de couleurs
3. Avoir le choix entre plusieurs designs d'ennemis
4. Avoir une page de "highscore"
5. Pouvoir régler le Gameplay pour adapter la jouabilité, l'accessibilité et l'éco-conception

L'objectif de ce projet est de crée une interface fluide et intuitive pour crée une expérience de jeux agréable pour les joueurs, tout en respectant les contraintes du projet.

Dans ce rapport se trouve, un chapitre d'analyse qui analysera : la conception centrée utilisateur, le choix de la palette de couleur du projet, l'éco-conception ainsi que l'accessibilité.

Un second chapitre sur la conception, qui comportera la définition des écrans, ainsi que les choix pris durant la conception du projet.

Et finalement le dernier chapitre de ce rapport conclura par une évaluation du prototype crée.

Analyse

Conception centrée utilisateur

La conception centrée utilisateur (CCU) représente une grande partie du projet, car tout le projet ne pourrait pas être valide s'il ne convient pas au préférences, besoins et attentes des utilisateurs.

Pour ce faire, deux personas a été créé pour guidée l'avancement du projet vers une finalité qui conviendrait aux futurs utilisateurs.

Également, une palette graphique a été créé. Et finalement, nous avons ajouté plusieurs options pour que notre projet soit le plus accessible envers les utilisateurs.

Création de deux profils de joueurs

Les personas sont créés pour représenter les utilisateurs typiques que notre projet est sensé cibler. Pour mieux comprendre les besoins, les préférences, caractéristiques des utilisateurs et mieux identifier les possibles futurs problèmes.

1. Le premier persona, qui se trouve ci-dessous, a été créé pour prendre en compte que l'expérience de jeux ne soit pas excessivement technique pour ne pas frustrer les utilisateurs, l'expérience de jeux devait également être nostalgique.

Bobby Kotick

Occupation
Menuisier

Location
Angleterre, Londres

Age
50

Family
Marié, 3 enfants

Tech

Internet

Social Media

Youtube

Biographie

Bobby Kotick est un menuisier de 50 ans résident à Londres. Marié depuis 25 ans et père de trois enfants. Dans sa jeunesse, Bobby était un passionné de jeux vidéo. Il se souvient avec nostalgie des heures passées à jouer à des jeux d'arcade et à des consoles de jeu vintage. Bien qu'il n'ait pas beaucoup de temps pour les jeux vidéo aujourd'hui, il garde précieusement sa collection de jeux rétro et se replonge parfois dans cet univers pour revivre ces souvenirs d'enfance.

Besoin

- **Accessibilité:** Étant donné qu'il n'utilise pas beaucoup la technologie, Bobby a besoin d'un jeu simple et convivial.
- **Réminiscence:** Bobby pourrait ressentir le besoin de retrouver les jeux auxquels il jouait dans sa jeunesse.
- **Expérience solo:** Bobby pourrait préférer des jeux solo plutôt que des jeux multijoueurs complexes. Cela lui permettrait de jouer à son propre rythme, sans la pression de compétition avec d'autres joueurs en ligne.

Frustration

- **Difficulté excessive:** Bobby a besoin de défis adaptés à son niveau.
- **Complexité excessive:** Bobby préfère les jeux simples et intuitifs.
- **Achats in-app obligatoires:** Si le jeu contient des achats in-app excessifs ou rend difficile la progression sans dépenser de l'argent réel, cela pourrait frustrer Bobby.



2. Le deuxième persona, qui se trouve ci-dessous, a été créé dans le but de rendre l'interface de jeux attrayante pour un utilisateur passionné d'informatique et de jeux vidéo.

Kazuki Nakamura

Occupation
Étudiant

Location
Suisse, Lausanne

Age
17

Family
Célibataire

Tech

Internet

Social Media

Youtube

Biographie

Kazuki Nakamura, 17 ans, est un étudiant en Suisse d'origine japonaise. Il a une passion pour la technologie et les jeux vidéo, jouant régulièrement en ligne. Kazuki est un élève brillant, notamment en mathématiques et en sciences. Il envisage de poursuivre des études en ingénierie informatique ou dans le domaine du développement de jeux. Sa vie est un mélange de deux cultures, ce qui contribue à sa personnalité diversifiée et ouverte d'esprit.

Besoin

- **Expérience multijoueur en ligne:** Kazuki cherche à interagir avec d'autres joueurs, que ce soit pour jouer avec ses amis ou pour s'engager dans des compétitions en ligne.
- **Système de collection:** Kazuki aime collecter des objets en jeu, ce qui lui offre un sentiment d'accomplissement et un moyen d'ordonner la durée de vie des jeux.
- **Téléchargement facile d'accès:** Kazuki aimerait que les jeux vidéo qu'il apprécie soient facilement téléchargeables, de préférence depuis des plateformes populaires telles que Steam.

Frustration

- **Problèmes techniques:** Les bugs, les plantages et les temps de chargement longs agacent les joueurs.
- **Achats in-app excessifs:** Les incitations à effectuer des achats in-app excessifs agacent de la frustration.
- **Niveaux de difficulté déséquilibrés:** Des niveaux de difficulté inadaptés peuvent être source de frustration pour les joueurs.



Choix de la palette graphique

Dans le projet, la palette graphique est primordiale dans l'identité visuelle du jeu. De plus le thème du jeu étant l'espace les quatre fonds ci-dessous ont été choisis, pour correspondre à l'ambiance général du jeu.



Par rapport à la palette de couleur, le choix à été fait d'avoir une palette de couleur avec comme couleur principale le bleu pour mettre en valeur l'immersion de l'utilisateur d'une ambiance spatiale.

Voici ci-dessous la palette de couleur choisis pour le projet :



Eco-conception

Dans un projet comme celui-ci, l'éco-conception à plusieurs bénéfices comme la réduction de l'impact environnemental, l'économie des coûts, améliorer l'image de marque, etc...

De ce fait, pour rendre ce projet le plus éco-responsable, plusieurs choix ont été fait pour respecter le critère de la réduction de consommation d'énergie. Pour ce faire en premier lieu, la grande majorité de l'interface du projet est de couleur sombre qui nécessite moins d'énergie pour être affichée, ce qui économise la durée de vie de la batterie et de l'écran de l'utilisateur. De plus, les images utilisées sont de type PNG, ce qui économise également de la durée de vie des appareils, car les images de type PNG utilisent moins d'espace de stockage, ont un temps de chargement réduit, etc.

Accessibilité

En ce qui concerne l'accessibilité dans ce projet, plusieurs choix ont été fait pour rendre ce projet plus accessible, dont la plupart se trouvant dans le menu option qui se compte au nombre de 4 :

1. La première option, permet aux utilisateurs mal entendant de pouvoir avoir des sous-titres afin que les utilisateurs puissent avoir une retransmission écrite des dialogues vu à l'écran.
2. La deuxième option, est le fait que les utilisateurs puissent activer, ou non, le fait d'afficher les textes du jeu avec la police « OpenDyslexic » pour que les utilisateurs atteint de dyslexie puissent lire les textes afficher dans le jeu avec plus de facilité.
3. La troisième option, permet aux utilisateurs d'augmenter, ou de baisser, le contraste pour mieux identifier les couleurs, et de mieux identifier les éléments à l'écran.
4. Finalement la quatrième option, est une option qui permet aux utilisateurs atteint de daltonisme de choisir un filtre qui modifie les schémas de couleurs, afin de mieux d'améliorer leur visibilité.

Conception

Définition des écrans

Les maquettes créées ont toutes une résolution de 1980 x 1080, car dans le monde de l'informatique cette résolution est une norme, de ce fait il est garanti que les maquettes soient compatibles avec un grand nombre d'appareils.

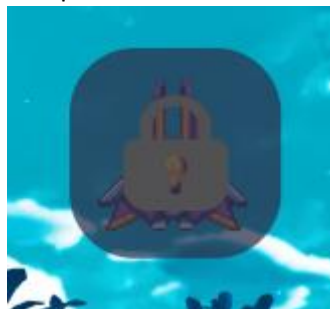
Choix effectués

Voici ci-dessous les choix les plus importants pris durant la conception de ce projet.

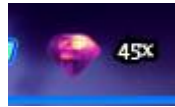
- **Barre de navigation**, la barre de navigation est visible dans la grande majorité des maquettes pour que les utilisateurs puissent se déplacer plus librement à travers les maquettes.



- **Menu principale**, il a été choisis que les boutons *Play*, *Crédit*, *Score*, *Option*, *LeaderBoard* soient afficher uniquement sur le menu principal et pas sur la barre de navigation, car les utilisateurs cliquent moins souvent sur ces boutons.
- **Skin menu**, les skins des vaisseaux qui ne sont pas débloqué sont grisé et ont une icône de cadenas pour faire comprendre facilement aux utilisateurs que ces vaisseaux en question ne sont pas encore débloqués.



- **Monnaie payante du jeu**, une monnaie payante a été créée dans l'idée de pouvoir rémunérer les développeurs du jeu, et de pouvoir continuer à améliorer le jeu grâce aux rémunérations de cette monnaie.



- **Difficulté du niveau**, Pour représenter la difficulté d'un niveau, il a été pensé d'utiliser des étoiles (une étoile pour les plus simples niveaux, trois étoiles pour les plus complexes) afin de simplifier le choix que les utilisateurs auront à faire. (Un joueur qui voudra jouer à un niveau facile aura plus de facilité à le trouver)

Niveau	Nom	Difficulté
#1	Bison	★
#2	Attila	★
#3	Barracuda	★★
#4	Bonite	★★
#5	Condor	★
#6	Nimrod	★★★
#7	Diapason	★★
#8	Artémis	★★★

- **Les slides d'information**, les slides d'information permettent de partager simplement des informations importantes avec les utilisateurs, ce qui crée une communication transparente envers les joueurs.



Evaluation

Tests

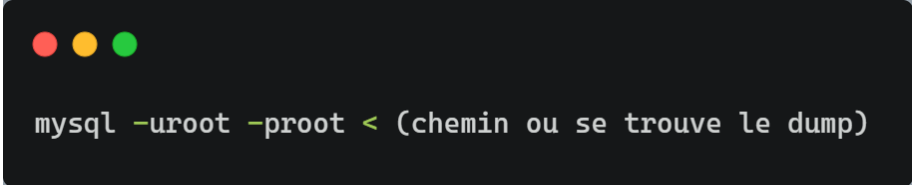
Nom du menu	Description	Condition pré-requise	Etape de test	Donnée de test	Résultat attendu	Résultat actuelle
Menu principal	Menu principal du jeu avec les sous menu principaux	Application installée	Clique sur le sous menu Crédit	Cliquer sur un sous menu	Renvoie sur le sous menu cliqué	Renvoie sur le sous menu cliqué
Menu skin	Menu qui permet de sélectionner un skin de vaisseau	Application installée	Cliquer sur la flèche de droite à coter du nom du vaisseau	Cliquer sur la flèche de droite	Sélectionne le vaisseau de droite	Sélectionne le vaisseau de droite
Menu inventaire	Menu qui permet de voir tous les items en possession de l'utilisateur	Application installée	Cliquer sur un des items posséder	Cliquer sur un des items posséder	Affiche la description de l'item	Aucun résultat
Menu shop	Menu qui contient des items à la ventes	Application installée	Cliquer sur un item afficher dans la boutique	Cliquer sur un item afficher dans la boutique	Affiche un message qui remercie le joueur pour avoir acheté un item	Aucun résultat
Sous menu option audio	Sous menu option audio qui permet de changer les options relatives à l'audio	Application installée Avoir cliquer sur le sous menu option qui se trouve dans le menu principal	Régler la barre de son du volume principal	Régler la barre de son du volume principal	Modifie le son de l'application selon le pourcentage de la barre de son	Aucun résultat
Menue préparation jouer	Menu qui permet de choisir les skins du vaisseau et des aliens	Application installée Avoir cliquer sur le bouton play du menu principal	Cliquer sur le bouton "next"	Cliquer sur un bouton	Renvoie sur le menu choix du niveau	Renvoie sur le menu choix du niveau

DB – Data base

A. Importer les données et le schéma de base de données

Pour pouvoir charger un fichier dump dans une base de données, il faut utiliser les commandes suivantes :

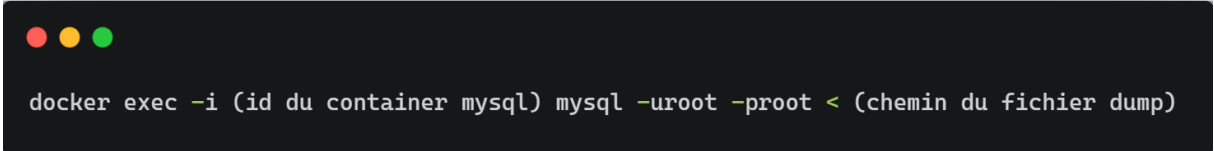
1. En premier lieu il faut utiliser la commande **"mysql -u (suivis du nom d'utilisateur) -p (suivis du mot de passe)"** cette dernière commande permet de se connecter au serveur MySQL avec le nom d'utilisateur choisi, après s'être connecté il est possible d'interagir avec un serveur de base de données.
 - a. La première partie de la commande, **"mysql"** permet d'indiquer que nous voulons utiliser le client MySQL dans notre invite de commande.
 - b. Deuxièmement, **"-u (suivis du nom d'utilisateur)"** permet de spécifier en tant que quel utilisateur nous voulons-nous connecter, il est important de noter que le nom d'utilisateur de base créé par MySQL est le super utilisateur **"root"**.
 - c. En dernier lieu, la commande **"-p"**, permet d'indiquer à MySQL qu'il faut nous demander un mot de passe après la saisie de la commande.
2. Ensuite, la commande à insérer après s'être connecté dans MySQL est la suivante :



```
mysql -uroot -proot < (chemin ou se trouve le dump)
```

Cette commande permet de charger un fichier dump dans le serveur MySQL.

- a. Comme dit précédemment, **"mysql -uroot -proot"** permet de se connecter au client MySQL en tant qu'utilisateur **"root"**.
 - b. Ensuite, **"< (chemin où se trouve le dump)"** permet de rediriger le contenu du fichier qui se trouve après le signe **"<"** dans la base de données.
3. Si le serveur MySQL se situe dans un conteneur docker, la commande à utiliser est la suivante :



```
docker exec -i (id du container mysql) mysql -uroot -proot < (chemin du fichier dump)
```

- a. **"docker exec"** permet d'exécuter la commande qui suit, à l'intérieur d'un conteneur docker.
 - b. La commande : **"-i (id du container mysql)"** permet de spécifier quelle container docker nous voulons utiliser.

B. Gestion des utilisateurs

Administrateur :

1. La commande suivante permet de créer un nouvel utilisateur dans MySQL

```
CREATE USER 'Administrateur_de_jeu'@'localhost' IDENTIFIED BY 'adminjeu';
```

- a. "CREATE USER" permet d'indiquer à mysql que nous voulons créer un nouvel utilisateur.
- b. " 'administrateur_de_jeu' " spécifie que l'utilisateur que nous voulons créer va se nommer administrateur_de_jeu.
- c. " 'localhost' " permet de spécifier à partir de quelle endroit l'utilisateur aura le droit se connecter.
- d. " IDENTIFIED BY 'adminjeu' " cette dernière partie de la commande informe le mot de passe associé à l'utilisateur que nous allons créer, pour se faire le mot de passe est placé entre des guillemets pour informer a mysql qu'il s'agit d'une suite de caractère.

2. La commande suivante permet de créer un rôle nommé "r_Administrateur_de_jeu".

```
CREATE ROLE 'r_Administrateur_de_jeu';
```

- a. La première partie, "CREATE ROLE" permet d'indiquer que l'on veut créer un nouveau rôle.
- b. Ensuite, " 'r_Administrateur_de_jeu' " est le nom du rôle que nous voulons créer.

3. La commande suivante permet d'accorder des privilèges à un utilisateur ou à un rôle :

```
GRANT SELECT, CREATE, UPDATE, DROP ON db_space_invaders.* TO 'r_Administrateur_de_jeu' WITH GRANT OPTION;
```

- a. " 'GRANT' " cette première partie de la commande informe MySQL que nous voulons accorder des privilèges à un rôle.
- b. " 'SELECT, CREATE, UPDATE, DROP' " cette partie de la commande spécifie tous les privilèges que l'on veut accorder au rôle, dans l'ordre,
SELECT : permet d'exécuter des requêtes SELECT sur la base de données pour y lire ses données.
CREATE : autorise à l'utilisateur de pouvoir créer des tables.
UPDATE : autorise à l'utilisateur de mettre à jours les données des tables dans la base de données
DROP : Autorise la suppression d'objet dans la base de données.
- c. " 'ON db_space_invaders.* ' " permet de spécifier sur quelle base de données et sur quelles tables l'utilisateur pourra jouir de ses privilèges, dans cet exemple les utilisateurs

qui auront le rôle pourrons utiliser leurs privilèges sur la base de données "db_space_invaders", le signe "*" signifie que les privilèges précédemment citer s'applique sur toutes les tables.

- d. **"TO 'r_Administrateur_de_jeu'"** signifie que l'on donne les privilèges au rôle "r_Administrateur_de_jeu".
- e. **"WITH GRANT OPTION"** cette dernière partie de la commande indique que le rôle "r_administrateur_de_jeu" peut déléguer les privilèges SELECT, CREATE, UPDATE et DROP à d'autre utilisateurs ou rôles.

4. La commande suivante permet d'accorder un rôle à un utilisateur :

```
GRANT 'r_Administrateur_de_jeu' TO 'Administrateur_de_jeu'@'localhost';
```

- a. **"GRANT"** indique que nous allons accorder un rôle à un utilisateur
- b. **"'r_Administrateur_de_jeu' "** ici nous spécifions quel rôle nous allons accorder par la suite
- c. **" TO 'Administrateur_de_jeu'@'localhost' "** cette dernière partie de commande indique à quel utilisateur nous allons accorder le rôle.

Joueur :

1. La commande suivante crée un utilisateur nommer Joueur

```
CREATE USER 'Joueur'@'localhost' IDENTIFIED BY 'joueur';
```

- a. **"CREATE USER"** indique à MySQL que nous voulons crée un nouvel utilisateur.

2. La commande suivante crée un rôle :

```
CREATE ROLE 'r_Joueur';
```

3. La commande suivante donne le privilège "SELECT" sur la table "t_arme" de la base de données "db_space_invaders" au rôle "r_joueur" :

```
GRANT SELECT ON db_space_invaders.t_arme TO 'r_Joueur';
```

4. La commande suivante donne le privilège "SELECT" ainsi que "CREATE" sur la table "t_commande" de la base de données "db_space_invaders" au rôle "r_Joueur" :

```
GRANT SELECT, CREATE ON db_space_invaders.t_commande TO 'r_Joueur';
```

5. La commande suivante donne le rôle "r_Joueur" à l'utilisateur "Joueur" :

```
GRANT 'r_Joueur' TO 'Joueur'@'localhost';
```

Gestionnaire de la boutique :

1. La commande suivante crée l'utilisateur "Gestionnaire_de_la_boutique" dont le mot de passe est "gestboutique" et qui peut se connecter depuis l'hôte "localhost"

```
CREATE USER 'Gestionnaire_de_la_boutique'@'localhost' IDENTIFIED BY 'gestboutique';
```

2. La commande suivante crée le rôle "r_Gestionnaire_de_la_boutique"

```
CREATE ROLE 'r_Gestionnaire_de_la_boutique';
```

3. La commande suivante donne le privilège au rôle "r_Gestionnaire_de_la_boutique" sur la table "t_Joueur" de la base de données "db_space_invaders"

```
GRANT SELECT ON db_space_invaders.t_joueur TO 'r_Gestionnaire_de_la_boutique';
```

4. La commande suivante donne les privilèges "SELECT, UPDATE, INSERT, DELETE" sur la table "t_arme" sur la base de données "db_space_invaders" au rôle "r_Gestionnaire_de_la_boutique"

```
GRANT SELECT, UPDATE, INSERT, DELETE ON db_space_invaders.t_arme TO 'r_Gestionnaire_de_la_boutique';
```

- a. Le privilège "INSERT" permet de rajouter des données dans une table.

5. La commande suivante donne le privilège "SELECT" au rôle "r_Gestionnaire_de_la_boutique" sur la table "t_commande" dans la base de données "db_space_invaders" :

```
GRANT SELECT ON db_space_invaders.t_commande TO 'r_Gestionnaire_de_la_boutique';
```

6. La commande suivante donne le rôle "r_Gestionnaire_de_la_boutique" à l'utilisateur "Gestionnaire_de_la_boutique" :

```
GRANT 'r_Gestionnaire_de_la_boutique' TO 'Gestionnaire_de_la_boutique'@'localhost';
```

C. Requêtes de sélection

Requête n°1 : La requête suivante sélectionne toutes les données de la table "t_joueur" et ensuite trie les résultats par ordre décroissant en fonction de la colonne "jouNombrePoints" et limite les résultats à 5, ce qui affiche les 5 meilleurs joueurs en fonction de leurs points :

```
SELECT *  
FROM t_joueur  
ORDER BY jouNombrePoints DESC LIMIT 5;
```

- a. "SELECT" permet d'indiquer à MySQL les tables que nous voulons sélectionner, dans ce cas nous sélectionnons toutes les tables car "*" signifie toutes les colonnes d'une table
- b. "FROM t_joueur" permet de spécifier le ou les tables à partir desquelles nous avons sélectionner les colonnes précédentes
- c. "ORDER BY jouNombrePoints" permet d'afficher les résultats de la requête en fonction de la colonne "jouNombrePoints"
- d. "DESC" permet d'afficher les résultats dans l'ordre décroissant
- e. "LIMIT 5" permet de limiter les résultats afficher, dans ce cas il limite les résultats afficher à 5.

Requête n°2 : La requête suivante affiche le prix maximal, le prix minimal et le prix moyen des armes de la tables "t_arme"

```
SELECT MAX(armPrix) AS PrixMaximum, MIN(armPrix) AS PrixMinimum,
AVG(armPrix) AS PrixMoyen
FROM t_arme;
```

- "MAX (armPrix) AS PrixMaximum" permet d'afficher la valeur maximale de la colonne "armPrix" et de renommer le résultat par "PrixMaximum"
- "MIN (armPrix) AS PrixMinimum" permet d'afficher la valeur minimale de la colonne "armPrix" et de renommer le résultat par "PrixMinimum"
- "AVG (armPrix) AS PrixMoyen" permet d'afficher la valeur moyenne de la colonne "armPrix" et de renommer le résultat par "PrixMoyen"

Requête n°3 : La requête suivante sélectionne les id des joueurs, affiche le nombre de commandes effectuées en fonction de chaque utilisateur et les trie par ordre décroissant.

```
SELECT fkJoueur AS IdJoueur, COUNT(fkJoueur) AS NombreCommandes
FROM t_commande
GROUP BY fkJoueur ORDER BY NombreCommandes DESC;
```

- "COUNT (fkJoueur) AS NombreCommandes" permet de compter le nombre de fois que chaque identifiant que "fkJoueur" apparaît dans la table "t_commande". De plus dans cette commande, "AS NombreCommandes" permet de renommer la colonne ou résultat apparaîtra, dans cette exemple la colonne s'appellera "NombreCommandes"
- "GROUP BY fkjoueur" groupe les résultats en fonction de la colonne "fkJoueur"

Requête n°4 : La requête suivante sélectionne les id des joueurs affiche le nombre de commandes effectuées en fonction de chaque utilisateur, seuls les joueurs ayant exécuter plus de 2 commandes seront afficher, de plus le nombre de commandes sera afficher dans l'ordre décroissant.

```
SELECT fkJoueur AS IdJoueur, COUNT(fkJoueur) AS NombreCommandes
FROM t_commande
GROUP BY fkJoueur HAVING COUNT(fkJoueur) > 2
ORDER BY NombreCommandes DESC;
```

- "HAVING" est utilisé pour filtrer les groupes, dans cette exemple il est utiliser pour spécifier que seuls les "fkJoueur" qui apparaissent plus de 2 fois seront afficher.

Requête n°5: La requête suivante permet d'afficher l'entièreté des colonnes "jouPseudo", "armNom" ainsi que "comNumeroCommande"

```
SELECT t_joueur.jouPseudo, t_arme.armNom, t_commande.comNumeroCommande
FROM t_joueur
JOIN t_arsenal ON t_joueur.idJoueur = t_arsenal.fkJoueur
JOIN t_arme ON t_arsenal.fkArme = t_arme.idArme
JOIN t_commande ON t_joueur.idJoueur = t_commande.fkJoueur;
```

- "JOIN t_arsenal ON t_joueur.idJoueur = t_arsenal.fkJoueur" permet d'interconnecter deux table entre elle, dans cette exemple les deux tables connectées sont : "t_joueur" et "t_arsenal"

Requête n°6: La commande suivante permet d'afficher la dépense totale de chaque joueur, ainsi que de regrouper les résultats par joueur, les trie par ordre décroissant et limite les résultats à 10.

```
SELECT t_joueur.idJoueur AS Idjoueur, SUM(t_arme.armPrix *
t_detail_commande.detQuantiteCommande) AS TotalDepense
FROM t_commande
JOIN t_joueur ON t_commande.fkJoueur = t_joueur.idJoueur
JOIN t_detail_commande ON t_commande.idCommande = t_detail_commande.fkCommande
JOIN t_arme ON t_detail_commande.fkArme = t_arme.idArme
GROUP BY Idjoueur
ORDER BY TotalDepense DESC
LIMIT 10;
```

Requête n°7: la commande ci-bas affiche tous les pseudonymes des joueurs associés aux numéros de commande, même les joueurs qui n'ont pas passé de commande seront afficher en revanche s'ils n'ont pas passé de commandes, la colonne affichera null.

```
SELECT t_joueur.jouPseudo, t_commande.comNumeroCommande
FROM t_joueur
LEFT JOIN t_commande ON t_commande.fkJoueur = t_joueur.idJoueur;
```

- "LEFT JOIN t_commande ON t_commande.fkJoueur = t_joueur.idJoueur" Comme la commande JOIN mais au lieu de lier et d'afficher toutes les informations des deux colonnes des deux tables, toutes les informations de la table qui se situe à gauche du "LEFT JOIN" seront afficher même si il n'a pas de correspondance avec la deuxième table.

Requête n°8: la commande suivante permet d'afficher toutes les commandes ainsi que les joueurs qui ont passé ces commandes, les commandes seront toutes affichées même si aucune id de joueur y correspond

```
SELECT t_commande.comNumeroCommande, t_joueur.jouPseudo
FROM t_joueur
RIGHT JOIN t_commande ON t_commande.fkJoueur = t_joueur.idJoueur;
```

- “RIGHT JOIN t_commande ON t_commande.fkJoueur = t_joueur.idJoueur” Comme la commande JOIN LEFT, mais à la place d'afficher toutes les informations de la colonne qui est spécifiée à gauche du Join, ici c'est à droite du Join que toutes les informations seront affichées

Requête n°9: La requête ci-dessous, affichera

```
SELECT SUM(t_detail_commande.detQuantiteCommande) AS nombres_armes_achetes,
t_joueur.jouPseudo
FROM t_detail_commande
JOIN t_commande ON t_detail_commande.fkCommande = t_commande.idCommande
JOIN t_arme ON t_arme.idArme = t_detail_commande.fkArme
RIGHT JOIN t_joueur ON t_commande.fkJoueur = t_joueur.idJoueur
GROUP BY jouPseudo;
```

- L'expression “SUM (t_detail_commande.detQuantiteCommande)” permet de calculer la somme totale des colonnes qui se trouvent dans les parenthèses qui suivent le “SUM”

Requête n°10: la requête suivante affichera une liste de pseudo de joueur qui a acheté au moins une arme.

```
SELECT t_joueur.jouPseudo
FROM t_detail_commande
LEFT JOIN t_commande ON t_detail_commande.fkCommande = t_commande.idCommande
RIGHT JOIN t_joueur ON t_commande.fkJoueur = t_joueur.idJoueur
GROUP BY t_joueur.idJoueur
HAVING COUNT(t_detail_commande.fkArme);
```

D. Création des index

1. Certains index existent déjà car, MySQL crée automatiquement des index pour les colonnes suivantes : clé primaire, clé étrangère et colonne uniques, pour améliorer les performances des requêtes et garantir l'incorruptibilité des données

2.

Avantage	Inconvénients
<ul style="list-style-type: none">- <u>Performances des requêtes</u> (les index accélèrent les recherches, ainsi que les filtrages, jointure et le tri des données)- <u>Intégrité des données</u> (les index permettent d'être sûr que les données stocker dans les colonnes uniques sont uniques)- <u>Facilité d'utilisation</u> (si MySQL crée des index automatiquement pas besoin d'en créer)	<ul style="list-style-type: none">- <u>Surutilisation espace de stockage</u> (si un nombre excessif d'index automatique est créé cela peut devenir problématique)- <u>Maintenance complexe</u> (gérer un nombre important d'index leur d'une maintenance peut être plus complexe)- <u>Index inutiles</u> (parfois des index peuvent être créés sur des colonnes qui ne sont pas fréquemment utilisées)

3. Un index pertinent à ajouter serait sur la colonne de la colonne "t_joueur", car le pseudonyme des joueurs est une information importante et de ce fait il est très probable que de nombreuses requêtes comporteront l'information des pseudonymes des joueurs (comme le prouve les commandes de sélections précédentes, la colonne "jouPseudo" apparaît très souvent).

E. Backup / Restore

Backup

Voici la commande pour réaliser un backup de la base de données db_space_invaders :

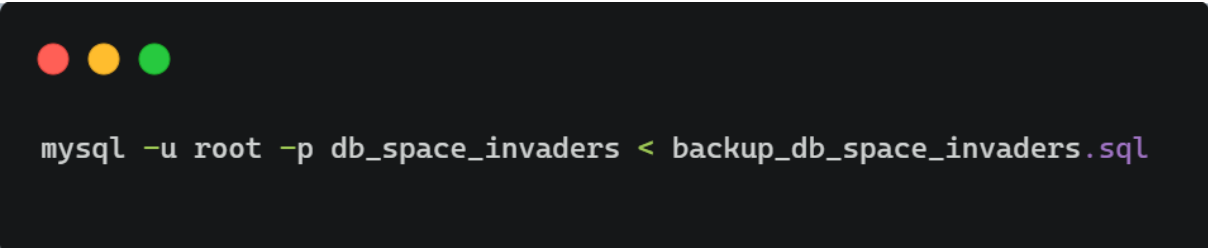
```
mysql -u root -p db_space_invaders > backup_db_space_invaders.sql
```

Au préalable de cette commande, il faut créer un fichier qui contiendra le backup (dans cet exemple : backup_db_space_invaders.sql)

Également "root" est le nom d'utilisateur de l'utilisateur qui est utilisé pour accéder à la base de données "db_space_invaders", il faut donc le remplacer en conséquence.

Restore

Voici la commande pour réaliser un restore de la base de données db_space_invaders :



```
mysql -u root -p db_space_invaders < backup_db_space_invaders.sql
```

Il est important de préciser qu'il faut que la base de données "db_space_invaders" soit préalablement créée pour que le restore fonctionne.

Et tout comme la commande de Backup précédente, dans cet exemple "root" est le nom d'utilisateur de l'utilisateur qui va se connecter à la base de données.

F. Liaison database & C#

Pour pouvoir Lier un code c# avec une base de données voici les points à réaliser :

1. En premier lieu il faut ajouter le package NuGet "MySQL.Data".
2. Après avoir télécharger l'extension MySQL, il faut ajouter le using suivant "using MySQL.Data.MySqlClient" pour pouvoir utiliser les fonctionnalités de MySQL dans le code.
3. Pour finir, il faut déclarer une variable qui stocker les informations nécessaires à la connections avec la base de données, les informations en questions : L'host du serveur, le Port du serveur, le nom de la base de données, le nom de l'utilisateur, mot de passe.

Voici à quoi ressemble mon string de connexion :

```
public static string dbConnexion = "Server=localhost;Port=6033;Database=db_space_invaders;User=root;Password=root;";
```

POO – Programmation orientée objet

A. Introduction

Ce rapport retrace la création d'un programme c# en console qui imite le célèbre jeux "Space Invaders", ce projet utilise : la programmation orientée objet, l'utilisation de classes ainsi que de l'héritage, et pour finir, une structure de moteur de jeu qui sépare les informations et l'affichage que l'utilisateur vois à travers son écran.

De plus, pour structurer les informations partager, cette introduction suivra de 5 chapitres, voici lesquelles : Analyse fonctionnelle, analyse technique, test unitaires, utilisation ChatGPT, conclusion.

B. Analyse fonctionnelle

Démarrer une partie

En tant que joueur Je veux démarrer une partie	
Tests d'acceptance:	
Lancer le programme	Sur le bureau de mon pc Quand je double clique sur l'icône du programme La fenêtre du jeu apparait pour afficher le playground (voir maquette)

Bouger Joueur

En tant que joueur J'aimerais pouvoir bouger mon joueur	
Tests d'acceptance:	
Bouger à droite	Dans une partie quand j'appuis sur la flèche de droite mon joueur bouge d'une case à droite
Bouger à gauche	Dans une partie Quand j'appuie sur la flèche de gauche Mon joueur bouge d'une case à gauche
Limitation terrain	Dans une partie Quand mon joueur est à l'extrémité gauche ou droite du playground et qu'il se décale encore un fois mon joueur ne bouge pas et reste à l'extrémité

Tirer missile

En tant que joueur Je veux pouvoir tirer des missiles Pour tuer des ennemis	
Tests d'acceptance:	
Tirer missile	Dans une partie Quand j'appuie sur la touche espace mon joueur tire un missile devant lui
Blessar ennemi	Dans une partie Quand un missile touche un ennemi "en forme" le missile disparaît, l'ennemi ne change pas d'aspect
Blessar ennemi	Dans une partie Quand un missile touche un ennemi blessé (hp "moyen") le missile disparaît, l'ennemi change de couleur
Achever ennemi	Dans une partie Quand un missile touche un ennemi blessé (hp très bas) le missile disparaît, l'ennemi disparaît

Apparaître alien

En tant que joueur je veux pouvoir voir des aliens	
Tests d'acceptance:	
Apparition aliens	Quand une partie est lancé deux rangés d'aliens apparaissent
Mouvement aliens	Quand les aliens sont apparus ils bougent en suivant le schémas (voir maquette)

Afficher menu

En tant que joueur Je veux pouvoir voir un menu et interagir avec	
Tests d'acceptance:	
Afficher menu	Quand je suis sur le menu principale Un titre s'affiche, ainsi qu'un sous menu "joueur", un sous menu "option" , et un sous menu "score"
Se déplacer dans menu	Dans le menu principal Quand j'appuie sur flèche du bas, ou du haut le sous menu du bas ou du haut est sélectionné en fonction de la touche que j'ai utilisé, et se met en surbrillance
Sélectionner un sous menu	Dans le menu principal Quand j'ai sélectionné un sous menu, et que j'appuie sur la touche enter Je me déplace dans le sous menu sélectionné

Afficher score

En tant que joueur Je veux pouvoir voir mon score actuel en bas à gauche de l'écran quand je joue	
Tests d'acceptance:	
Score	Dans une partie Quand l'utilisateur joue En bas à gauche de l'écran le score du joueur s'affiche en temps réel

Afficher les hp joueur

En tant que joueur Je veux pouvoir voir mon nombre de HP en jeux en bas à droite de mon écran

Tests d'acceptance:

Afficher HP joueur Dans une partie Quand l'utilisateur joue En bas à ses HP sont afficher en temps réel

GameWin

En tant que joueur Quand je tue tous les aliens je veux qu'un menu "GameWin" s'affiche

Tests d'acceptance:

Afficher menu "GameWin" Dans une partie Après avoir tué tous les ennemis Un menu "GameWin" s'affiche

GameLoose

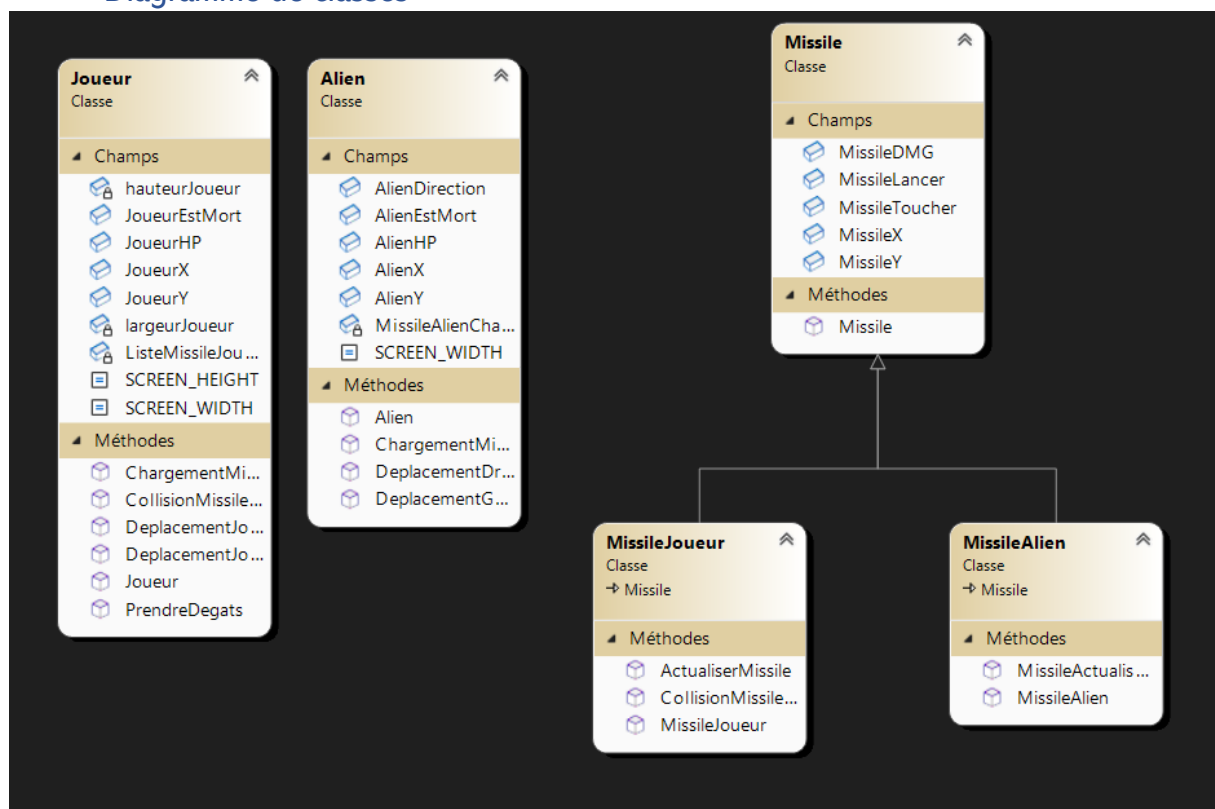
En tant que joueur Quand les Hp du joueur tombe à 0, je veux qu'un menu "GameLoose" s'affiche conformément à la mquette.

Tests d'acceptance:

Afficher menu GameLoose Dans une partie Quand les Hp du joueur tombe à 0 Un menu "GameLoose" s'affiche

C. Analyse technique

Diagramme de classes



Explication DocFX

- L'explication du diagramme est disponible dans le dossier "Document" sous le nom de "DocFX.pdf".

D. Tests Unitaire

Ci-dessous se trouve des tests unitaires pour 4 méthodes du projet c#.

Les méthodes choisis sont :

- 1) La méthode de déplacement droite du joueur
- 2) Le constructeur de la classe Joueur
- 3) La méthode de déplacement droite de l'alien
- 4) Le constructeur de la classe Alien

Voici les tests unitaires ci-dessous :

```
[TestMethod()]
public void DeplacementJoueurDroiteTest()
{
    // Arrange
    Joueur joueur = new Joueur(0, 0);

    // Act
    joueur.DeplacementJoueurDroite();

    // Assert
    Assert.AreEqual(4, joueur.JoueurX);
}

[TestMethod()]
public void DeplacementJoueurDroiteTest_NeDépassePasLaLargeurDeEcran()
{
    // Arrange
    int largeurEcran = 150;
    Joueur joueur = new Joueur(largeurEcran - 7, 0);

    // Act
    joueur.DeplacementJoueurDroite();

    // Assert
    Assert.AreEqual(largeurEcran - 7, joueur.JoueurX);
}
```

Ces deux tests ci-dessus permettent de tester en premier lieu, le fait que la méthode déplace bien le joueur de 4 cases, secondement il test que si le joueur se trouve à la limite de la console, il ne pourra pas se déplacer en dehors des limites.

```
[TestMethod()]
public void JoueurTest()
{
    // Arrange
    int X = 10;
    int Y = 5;

    // Act
    Joueur joueur = new Joueur(X, Y);

    // Assert
    Assert.AreEqual(X, joueur.JoueurX);
    Assert.AreEqual(Y, joueur.JoueurY);
    Assert.IsFalse(joueur.JoueurEstMort);
    Assert.AreEqual(20, joueur.JoueurHP);
}
```

Ce test ci-dessus vérifie que le constructeur de la classe Joueur initialise correctement les données de l'objet joueur.

```
[TestMethod()]
public void DeplacementDroiteAlienTest()
{
    // Arrange
    Alien alien = new Alien(0, 0, 0);

    // Act
    alien.DeplacementDroiteAlien();

    // Assert
    Assert.AreEqual(2, alien.AlienX);
}

[TestMethod()]
public void DeplacementDroiteAlienTest_AlienDoitChangerDirection()
{
    // Arrange
    int largeurEcran = 150;
    Alien alien = new Alien(largeurEcran - 8, 0, 0);

    // Act
    alien.DeplacementDroiteAlien();

    // Assert
    Assert.AreEqual(largeurEcran - 6, alien.AlienX);
    Assert.IsFalse(alien.AlienDirection);
}
```

Les testes ci-dessus permettent de déterminer, en premier lieu que la méthode de déplacement à droite des aliens les déplace en effet à droite (de 2 cases), secondement que si un alien atteint la limite droite de la console, il change de sens.


```
[TestMethod()]
public void AlienTest()
{
    // Arrange
    int X = 10;
    int Y = 5;
    int HP = 20;

    // Act
    Alien thomas = new Alien(X, Y, HP);

    // Assert
    Assert.AreEqual(X, thomas.AlienX);
    Assert.AreEqual(Y, thomas.AlienY);
    Assert.AreEqual(HP, thomas.AlienHP);
    Assert.IsFalse(thomas.AlienEstMort);
}
```

Le test ci-dessus, permet d'évaluer si le constructeur de la classes Alien, initialise correctement les données de l'objet thomas (qui est un Alien).

Pour finir, voici ci-dessous la preuve que les tests unitaires précédent sont fonctionnels

The screenshot shows the 'Explorateur de tests' (Test Explorer) window in Visual Studio. At the top, it indicates 'Série de tests achevée : 1 tests (1 réussi(s), 0 non réussi(s), 0 ignoré(s)) exécutés en 894 ms'. Below this, a table lists the tests and their durations:

Test	Durée	Caractéris...	Message d'err
ModelTests (6)	39 ms		
Model.Tests (6)	39 ms		
AlienTests (3)	23 ms		
AlienTest	8 ms		
DeplacementDroiteAlienTest	7 ms		
DeplacementDroiteAlienTestAlienChangeDirection	8 ms		
JoueurTests (3)	16 ms		
DeplacementJoueurDroiteTest	3 ms		
DeplacementJoueurDroiteTestNeDépassePasLargeurEcran	7 ms		
JoueurTest	6 ms		

On the right side, a summary box shows 'Résumé du groupe' for ModelTests, indicating 'Tests dans le groupe: 6' and 'Durée totale: 39 ms'. Below this, the 'Résultats' section shows '6 Réussite' with a green checkmark icon.

E. Utilisation ChatGPT

Durant ce projet j'ai utilisé ChatGPT modérément pour m'aider dans la conception du projet,

J'ai quelque fois utilisé ChatGPT pour m'aider à comprendre des erreurs quand je passais un temps excessif à régler ces dernières.

Également, j'ai utilisé ChatGPT quand je ne connaissais pas la syntaxe que certaines commandes, ou si je ne savais tout simplement pas comment réaliser une tâche spécifique, comme par exemple, dans la ligne de code ci-contre :

```
ListeAlien.RemoveAll(alien => alien.AlienHP <= 0);
```

Mais dans tout mes cas d'utilisation de ChatGPT j'ai toujours fait attention à comprendre le code qu'il me proposait, pour toujours comprendre l'entièreté de mon projet.

F. Conclusion

Pour clôturer ce rapport, je dirais que ce projet ma permit d'avoir une meilleure compréhension de ce qu'était la programmation orienté objet et d'appliquer les concepts vus préalablement en classes, ce qui a été une expérience enrichissante dans la carde de la programmation.

Ce projet de recreation du célèbre jeu Space Invaders a été une expérience d'apprentissage agréable, enrichissant et ludique. Il m'a permis de solidifier mes compétences en POO, d'utiliser de nouveau concepts de la programmation, tel que l'héritage ainsi que les tests unitaires. Ce projet a définitivement renforcé mon intérêt pour la programmation.