

P-OO_SpaceInvaders



Auteur : Thomas Nardou

Date : 28.08.2023 – 03.11.2023

Périodes : 60

Chef de projet : Mathieu Meylan, Aurélie Curchod, Xavier Carrel

Table des matières

Chapitre 1 : Partie programmation	1
1.1 : Introduction.....	1
1.2 : Analyse fonctionnelle	1
Movement of player	1
shoot ammo	1
Movement of ennemy.....	2
Change the program structure	2
Finish game.....	2
backGroundMusic	3
dataBase connection	4
backGroundMusic	4
limit ammo	4
Score display.....	5
Make Menu	5
change language.....	5
visit highscores	6
1.3 : Analyse technique	6
Diagramme de classe :.....	6
Documentation.....	7
1.4 : Test Unitaires.....	8
1.5 : Chat GPT	9
1.6 : Conclusion	9
Chapitre 2 : Partie UX/UI	10
2.1 : Introduction.....	10
2.2 : Analyse	10
3.2.1 : Conception centré utilisateur.....	10
3.2.2 : Palette Graphique	11
3.2.3 : Éco-Conception :	12
3.2.4 : Accessibilité :	12
2.3 : Conception :	12
3.3.1 : définition des écrans	12
3.3.2 : Choix effectués :	13
2.4 : Tests.....	14
2.5 : Conclusion	15
Chapitre 3 : Partie Base de données	16

3.1 : Importer la base de données.....	16
3.2 : Requêtes.....	16
3.2.1 : Requête N°1	16
3.2.2 : Requête N°2	17
3.2.3 : Requête N°3	17
3.2.4 : Requête N°4	17
3.2.5 : Requête N°5	18
3.2.6 : Requête N°6	18
3.2.7 : Requête N°7	19
3.2.8 : Requête N°8	19
3.2.9 : Requête N°9	20
3.2.10 : Requête N°10	20
3.3 : Création des types d'utilisateur et utilisateur	20
3.4 : Assignment des permissions.....	22
3.4.1 : Rôle "r_admin"	22
3.4.2 : Rôle "r_player"	22
3.4.3 : Rôle "r_shopKeeper"	23
3.5 : exporter / importer la base de données	23
3.6 : Création des index.....	24
3.7 : Intégration de la base de données dans le programme C#.....	24
3.8 : Conclusion	24
Chapitre 4 : Conclusion Final	25

Chapitre 1 : Partie programmation

1.1 : Introduction

Dans le cadre du Projet P-OO il a été demandé de réaliser une réplique du célèbre jeu "Space Invaders". Le jeu doit contenir au minimum :

- Un vaisseau avec possibilité de tir et déplacement
- 10 ennemis qui descendent sur l'axe vertical

1.2 : Analyse fonctionnelle

Movement of player

As a Player I want to go left and right	
Tests d'acceptance:	
go left	In the main program When i press "a" the player shifts two pixels to the left
go right	In the main program When i press "d" the player shifts two pixels to the right
border left	In the main program When i press "a" and the player is on de border left the player don't move
border right	In the main program When i press "d" and the player is on de border right the player don't move
player show	in the main program when the game start the player appears at the bottom

shoot ammo

As a player I want to shoot ammo In order to kill ennemy	
Tests d'acceptance:	
shoot ammo	In the main program When i press the spacebar a bullet appears
ammo movement	In the main program with a bullet in the game at each frame the ammo shifts two pixels towards the top
ammo remove	In the main program when a bullet arrives two pixels before the top of the screen the ammo is deleted
enemy kill	in the main program When a ammo touch an ennemy the enemy is deleted and the bullet too

Movement of ennemy

As a Player I want enemies to be able to move	
Tests d'acceptance:	
Enemy show	In the main program when the game start 10 Enemies appear on the top (see maquette 1)
go right	In the main program when the game start the Enemies go to right
go down	In the main program when one enemy is on the border left or right the Enemies go down two pixels
go left	in the main program when the enemies go down they change their direction

Change the program structure

As a programmer I want to change the program structure to separate the different elements of the program	
Tests d'acceptance:	
change structure	In the structure of the program when i see the structure the program is separate in 4 project (model, display, console, storage)

Finish game

As a Player I want to finish game	
Tests d'acceptance:	
Lose game	In the main program if the enemies are on the same line as the player the console is cleared, the lose logo appears (see LoseMaquette)
Win game	In the main program if the player kill all anemies the console is cleared, the win logo appears (see WinMaquette)
speed increases	In the main program when the player win the speed increases by 1

backGroundMusic

As a player I want to have backGroundMusic

Tests d'acceptance:

main menu music	In the main program When the program start the main lobby music start
game music	in the main program when the game start the game music start
Win music	In the main program when the win page appears the win music start
loose music	In the main program when the loose page appears the loose music start
change game background music	in the game when the number of enemies halved the game background music change

dataBase connection

As a devloppe I want to do the dataBase connection	
Tests d'acceptance:	
open connection	In the main progam when the highscore page is loaded the database connection is open
Select scores	In the main progam when the highscore page is loaded a select query is made to get all games data: * the player name * player Score The data is sorted on time ascending, only the first 5 players are selected
close connection	In the main progam When the highscore page is unloaded the database connection is closed
Insert player score	In the main program when the player lose the game her score is save in the database

backGroundMusic

As a player I want to have backGroundMusic	
Tests d'acceptance :	
main menu music	In the main program When the program start the main lobby music start
game music	in the main program when the game start the game music start
Win music	In the main program when the win page appears the win music start
Loose music	In the main program when the loose page appears the loose music start
Change game background music	in the game when the number of enemies halved the game background music change

limit ammo

As a player I want to limit a number of ammo In order to have more difcult and not spam	
Tests d'acceptance :	
Display number ammo	In the program When game starts he display a number ammo under the score see maquette_2
lose ammo	In the game when the player shoot the number of ammo goes down by 1
run out of ammo	In the game with only 1 ammo left when the player shoots and misses the ennemies the player loses game
reset ammo	in the game when the player lost the number of ammo is reset
give ammo	In the game when the player kill one enemie the player receive 2 ammo

Score display

As a player I want to see my score	
Tests d'acceptance:	
add Point	In the game when player kill an enemie the score add 10 points
display score	In the game when the game starts the score of the previous game appears (first game the score is equals 0) see maquette_1.png
reset score	in the game when the player lose the game her score is reset
keep score	in the game when the player win the game he keep her score

Make Menu

As a player I want a menu In order to navigate in game	
Tests d'acceptance:	
ask the default language	In the desktop when i start the program the program asks for the default language
ask player name	In the game when i chose the default language the program asks the player name
display menu	In the game when the player chose her name he display a menu (see menu_maquette.png)
display option page	in the main menu when i click on {2} the option page appears (see option_maquette.png)
display highScore page	in the main menu when i click on {3} the option page appears (see highScore_maquette.png)
display lockerroom page	in the main menu when i click on {4} nothing happens
exit game	in the main menu when i click on {5} the game closes
start game	in the main menu when i click on {1} the game start

change language

As a player I want to change language	
Tests d'acceptance:	
put the game in English	in the option page when i click on {2} the language of the game is English
put the game in French	in the option page when i click on {1} the language of the game is French

go back main menu	in the option page when I press a key other than 1 or 2 the main menu appears
-------------------	---

visit highscores

As a player I want to visit highscores In order to know the bests scores

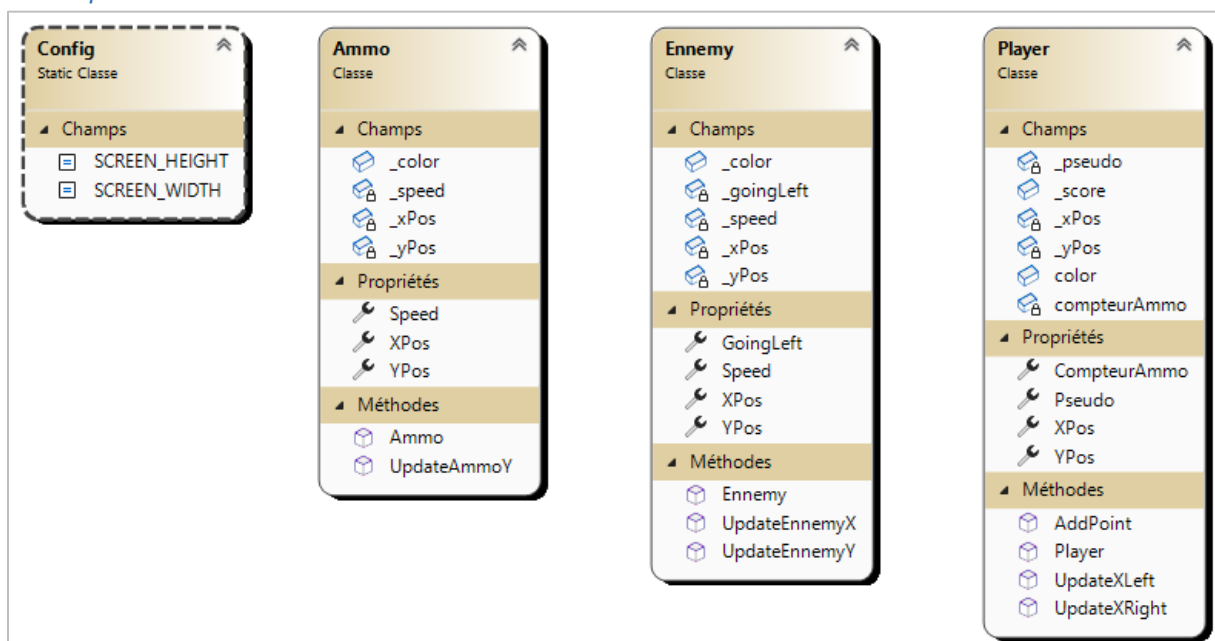
Tests d'acceptance:

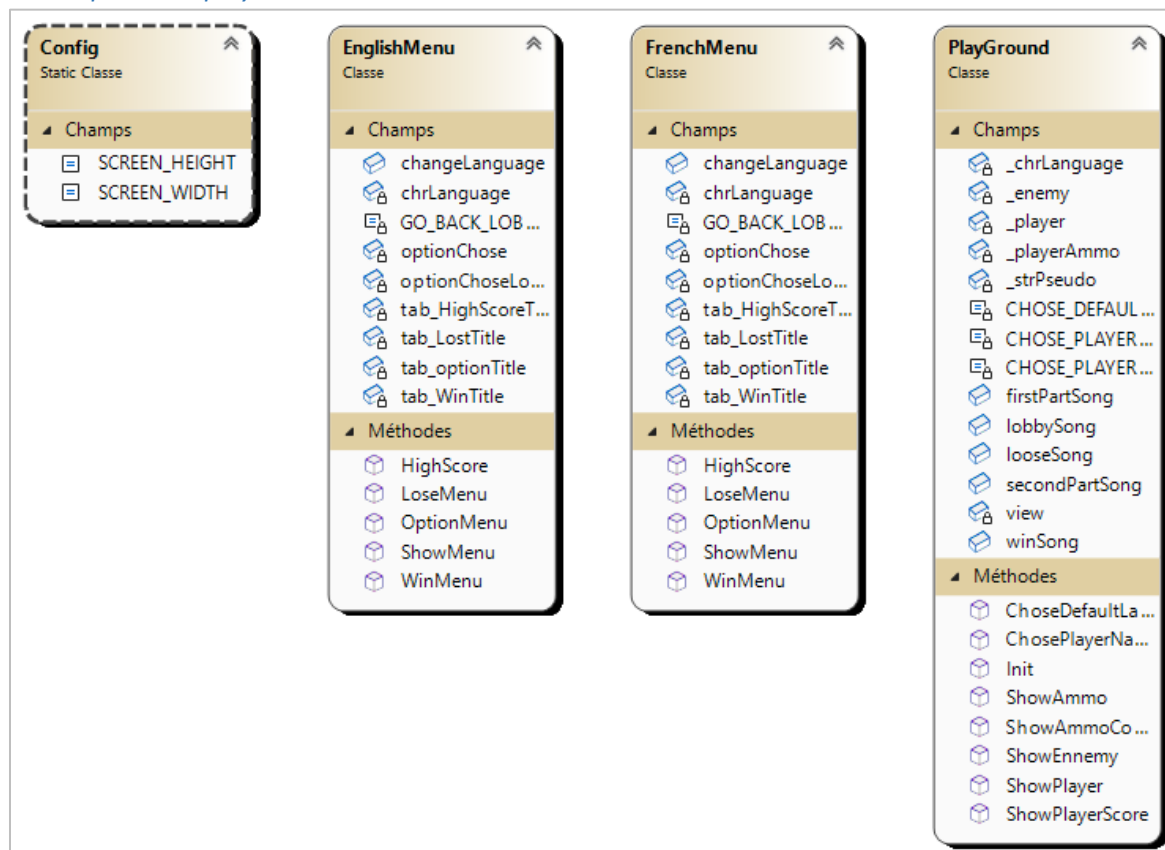
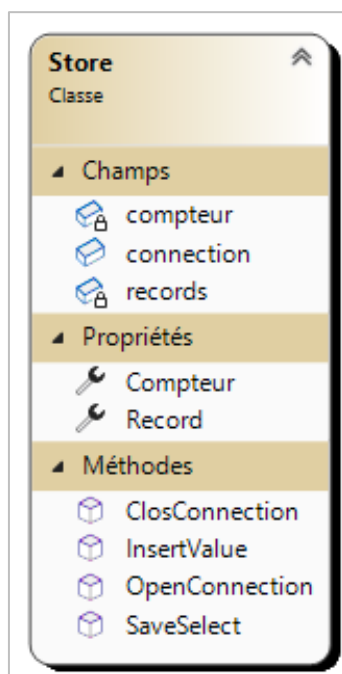
load highscore	in the highscore page when the page is loaded the first five players appears (see maquette)
go back main menu	in the highscore page when I press a key the main menu appears

1.3 : Analyse technique

Diagramme de classe :

Namespace : Model



Namespace : Display*Namespace : Storage**Documentation*

La documentation est disponible dans le répertoire «Doc/AnalyseTechnique ». Chaque classe est séparé dans le dossier de leur projet respectif.

1.4 : Test Unitaires

<u>Nom Test</u>	<u>Classe</u>	<u>Méthode testée</u>	<u>Description</u>	<u>Condition réussite</u>
UpdateXLeftTest	Player	UpdateXLeft	Tester le déplacement du joueur vers la gauche	La nouvelle valeur de X est égale à l'ancienne moins 2
UpdateXRightTest	Player	UpdateXRight	Tester le déplacement du joueur vers la droite	La nouvelle valeur de X est égale à l'ancienne plus 2
AddPointTest	Player	AddPoint	Tester l'attribution de points au joueur.	Le Score actuel du joueur est égal à l'ancien score plus 10
UpdateEnemyXTestLeft	Enemy	UpdateEnemyX	Tester le déplacement de l'ennemie vers la gauche	La nouvelle valeur de X est égale à l'ancienne moins 2
UpdateEnemyXTestRight	Enemy	UpdateEnemyX	Tester le déplacement de l'ennemie vers la droite	La nouvelle valeur de X est égale à l'ancienne plus 2
UpdateEnemyYTest	Enemy	UpdateEnemyY	Tester le déplacement de l'ennemie vers le bas	La nouvelle valeur de Y est égale à l'ancienne Plus 2
UpdateAmmoYTest	Ammo	UpdateAmmoY	Tester le déplacement de la munition vers le haut	La nouvelle valeur de Y est égale à l'ancienne moins 2

Résultat :

Voici les résultats de mes tests :

The screenshot shows the 'Explorateur de tests' (Test Explorer) window in Visual Studio. At the top, there are icons for running tests (play, stop, refresh, etc.) and a summary bar indicating 'Série de tests achevée : 7 tests (7 réussis(s), 0 non réussis(s), 0 ignoré(s)) exécutés en 53 ms' with 0 warnings and 0 errors. Below this is a table of test results:

Test	Durée	Caractéris...	Mess...
ModelTests (7)	3 ms		
Model.Tests (7)	3 ms		
AmmoTests (1)	3 ms		
UpdateAmmoYTest	3 ms		
EnemyTests (3)	< 1 ms		
UpdateEnemyXTestLeft	< 1 ms		
UpdateEnemyXTestRight	< 1 ms		
UpdateEnemyYTest	< 1 ms		
PlayerTests (3)	< 1 ms		
AddPointTest	< 1 ms		
UpdateXLeftTest	< 1 ms		
UpdateXRightTest	< 1 ms		

On the right side, the 'Récapitulatif des détails du test' (Test Details) pane shows the details for the selected test, 'UpdateAmmoYTest'. It indicates the source as 'AmmoTests.cs li' and the duration as 'Durée: 3 ms'.

1.5: Chat GPT

L'utilisation de l'IA ChatGPT a été nécessaire afin de reformuler des phrases dans le rapport

1.6: Conclusion

Le projet m'a permis d'approfondir mes connaissances en C# notamment en Orienté Objet. Certaines choses peuvent encore être améliorées comme l'emplacement de certaine variable. Il existe d'autres aspects qui pourraient être améliorés, notamment sur l'utilisation d'IceScrum car au début du projet, nous n'avions pas accès au logiciel, et il a été difficile de s'adapter en grande partie en raison d'un manque de motivation. Si le projet était à refaire je ne pense pas que je changerais grand-chose à ma façon de travailler si ce n'est peut-être mieux calculer mon temps pour IceScrum.

Chapitre 2 : Partie UX/UI

2.1 : Introduction

Dans le cadre du projet UX il nous a été demandé de réaliser une maquette pour le menu du célèbre jeu d'arcade « Space Invaders ». On nous a chargés de concevoir des maquettes pour chaque page du menu, tant pour le mode console que pour le mode graphique.






2.2 : Analyse

3.2.1 : Conception centré utilisateur

Pendant ce projet, j'ai choisi de créer deux persona afin de mieux comprendre les types d'utilisateurs potentiels de mon application, en prenant en compte les besoins et les préférences des utilisateur potentiel. Afin de réaliser cette partie j'ai dû utiliser l'IA chatGPT afin de me donner :

- Les informations personnelles de mes persona
- Leurs biographies
- Leurs envies
- Leurs frustrations

Lors de la réalisation de mes maquettes j'ai essayé de respecter les envies de mes persona le plus possible afin d'attirer un publique plus large. Voici mes persona :

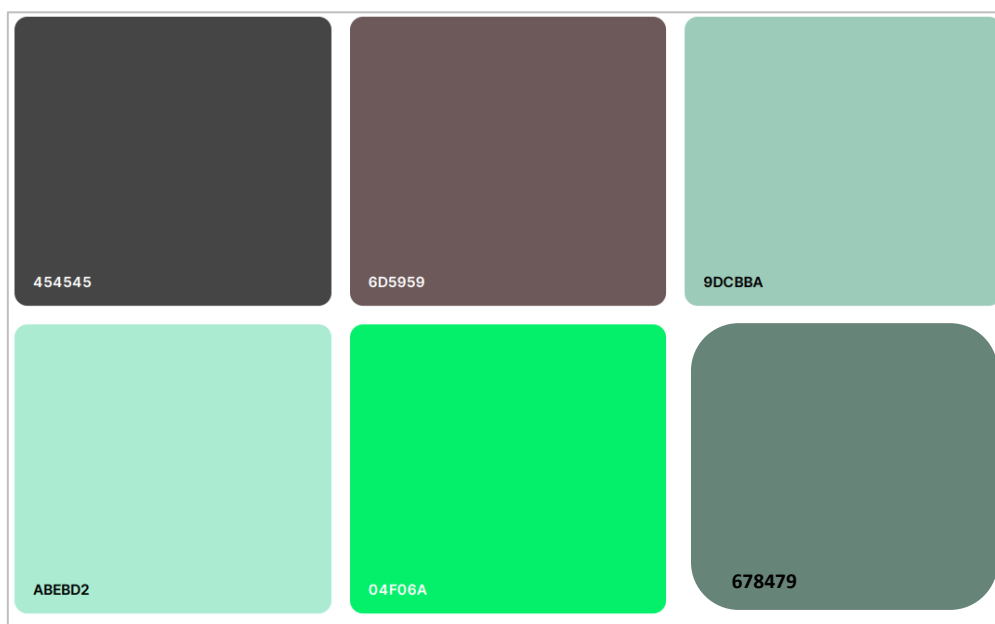
	<p><i>“Je trouve que les jeux sont absolument géniaux, ils me transportent vers des mondes extraordinaires où l'aventure et la créativité n'ont aucune limite.”</i></p>	<h3>Envies</h3> <ol style="list-style-type: none">1. Max Astral rêve de moderniser le style "Space Invaders" en capturant la nostalgie des jeux rétro tout en y ajoutant une touche innovante pour attirer les joueurs d'aujourd'hui.2. Max aimerait ajouter un bouton "HighScore" au menu pour afficher les meilleurs scores et ainsi ajouter de la compétition entre joueur.3. Max aimerait composer une bande-son originale pour le menu principal, créant une atmosphère captivante qui plonge immédiatement les joueurs dans l'univers du jeu.	<h3>Frustration</h3> <ol style="list-style-type: none">1. Max Astral est quelqu'un qui a un dégoût profond pour les microtransactions et les jeux vidéo qui favorisent les achats intégrés excessifs.2. Max pourrait être frustré si les joueurs ont du mal à naviguer dans le menu en raison d'une conception complexe ou de commandes peu intuitives.3. Les longues heures de travail et les échéances serrées peuvent entraîner un stress considérable, ce qui peut être une source de frustration pour Max.
<h2>Max Astral</h2> <div><div><p>Age/Genre 32/Homme</p></div><div><p>Localisation Los Angeles, Californie</p></div><div><p>Profession Programmeur de jeux vidéo indépendant</p></div><div><p>État civil Célibataire</p></div></div>	<h3>Bio</h3> <p>Max Astral est un passionné de jeux vidéo depuis son enfance. Né et élevé à Los Angeles, il a grandi en jouant à des jeux classiques tels que Space Invaders, Pac-Man et Galaga. C'est cette fascination pour les jeux rétro qui l'a finalement conduit à poursuivre une carrière de programmeur de jeux vidéo indépendant. Il a obtenu son diplôme en informatique à l'Université de Californie du Sud et a depuis travaillé sur plusieurs projets de jeux indépendants.</p>	<h3>Motivations</h3> <div><div>Créativité</div><div>Innovation</div><div>Passion pour les jeux</div><div>Récompense personnelle</div><div>Communauté</div></div>	<h3>Technologie</h3> <div><div>Internet</div><div>Réseau sociaux</div><div>Banque en ligne</div><div>Messengers</div></div> <div><div>★★★★☆</div><div>★★★★☆</div><div>★★★★☆</div><div>★★★★☆</div></div>

	<p><i>"Je suis convaincu que les jeux sont absolument géniaux, offrant une expérience immersive et divertissante comme aucune autre forme de divertissement."</i></p>	<h3>Envies</h3> <ol style="list-style-type: none"> 1. Créer un menu pour un jeu Space Invaders qui capture l'essence rétro du jeu tout en introduisant des éléments modernes. 2. Proposer aux joueurs la possibilité de personnaliser leur propre avatar ou vaisseau spatial dans le jeu Space Invaders, en leur permettant de choisir parmi une gamme de skins uniques 3. Ajouter un système de classement en ligne où les joueurs peuvent comparer leurs scores avec d'autres amateurs de Space Invaders du monde entier, encourageant ainsi la compétition amicale. 	<h3>Frustration</h3> <ol style="list-style-type: none"> 1. Elle est déçue par le fait que de nombreux jeux modernes manquent de la simplicité et du charme des jeux rétro qu'elle adore. 2. Sarah est parfois frustrée par le manque de temps pour jouer en raison de ses responsabilités professionnelles. 3. Si Sarah s'est fixé l'objectif ambitieux de battre le record du monde, elle peut ressentir une pression énorme pour réussir, ce qui peut parfois être décourageant et stressant, surtout si elle fait face à des obstacles. 								
<h2>Sarah Johnson</h2> <div> <div>  <p>Age/Genre 29/Femme</p> </div> <div>  <p>Localisation Toronto, Canada</p> </div> </div> <div> <div>  <p>Profession Ingénieure en informatique</p> </div> <div>  <p>État civil Célibataire</p> </div> </div>	<h3>Bio</h3> <p>Sarah Johnson est née à San Francisco, en Californie, et a grandi avec un intérêt précoce pour la technologie et les jeux vidéo. Elle a obtenu son diplôme en informatique de l'Université de Californie à Berkeley, où elle a participé activement à des clubs de jeux vidéo et de programmation. Après avoir obtenu son diplôme, elle a déménagé à Seattle pour travailler en tant qu'ingénieure en informatique dans une entreprise de jeux vidéo, où elle a contribué au développement de plusieurs jeux à succès.</p>	<h3>Motivations</h3> <div> <div>Créativité</div> <div>Innovation</div> <div>Passion pour les jeux</div> <div>Récompense personnelle</div> <div>Communauté</div> </div>	<h3>Technologie</h3> <table> <tr> <td>Internet</td> <td>★★★★★</td> </tr> <tr> <td>Réseau sociaux</td> <td>★★★★★</td> </tr> <tr> <td>Banque en ligne</td> <td>★★★★★</td> </tr> <tr> <td>Messengers</td> <td>★★★★★</td> </tr> </table>	Internet	★★★★★	Réseau sociaux	★★★★★	Banque en ligne	★★★★★	Messengers	★★★★★
Internet	★★★★★										
Réseau sociaux	★★★★★										
Banque en ligne	★★★★★										
Messengers	★★★★★										

Vous trouverez aussi les persona en pièce jointe ou en appuyant sur [ce lien](#)

3.2.2 : Palette Graphique

Lors de la réalisation de mes maquettes j'ai dû choisir une palette de couleur afin d'avoir des pages plus jolies pour réaliser cette partie je suis allé sur le site coolors.co qui est un site qui génère des palettes de couleur et après avoir générer un certain nombre de palette je suis arrivé à ce résultat :



J'ai décidé aussi de rajouter encore une nuance de vert (#678479) car je trouvais qu'il manquait des couleurs sombres donc j'ai décidé de la rajouter pour ajouter plus de contraste dans maquette tout en gardant une cohérence entre les couleurs. Si je me suis arrêté sur ces couleurs c'est pour ces deux raisons :

1. Le vert et ses nuances peuvent faire rappeler les aliens et aussi le Space Invaders original où le joueur est vert
2. Le noir et le brun peuvent faire rappeler l'espace c'est donc pour cela que j'ai choisi

3.2.3 : Éco-Conception :

Lors de conception des maquettes j'ai porté attention sur l'éco conception de mes maquettes en me basant sur les 115 bonnes pratiques afin pouvoir consommer le moins d'énergie possible notamment en choisissant un thème principal plutôt sombre qui sont les couleurs les moins énergivores de plus, j'ai décidé d'opter pour seulement deux polices d'écriture afin de limiter l'utilisation de ressources. Lors de la réalisation de l'application j'ai essayé de limiter le plus possible les transactions faites à la base de données hébergée sur un serveur

3.2.4 : Accessibilité :

Pour cette partie je porté attention aux personnes atteintes de daltonisme lors du choix de la palette de couleur. Afin de garantir que ma palette de couleurs est accessible, j'ai consulté des personnes atteintes de daltonisme pour recueillir leurs impressions sur les couleurs qu'elles perçoivent. De plus, j'ai également utilisé un simulateur de daltonisme comme color-blindness.com pour visualiser comment les personnes atteintes de différents types de daltonisme percevraient les couleurs de ma palette.

J'ai aussi porté attention aux personnes ayant des troubles de vue pour ce faire j'ai consulté des personnes atteintes de problèmes de vue en leur demandant s'ils pouvaient clairement lire ce qui était écrit.

2.3 : Conception :

3.3.1 : définition des écrans

Lors de la réalisation de mes maquettes et de mon application j'ai décidé de prendre ces dimensions :

- Console : 759 x 638, si j'ai opté pour ces dimensions parce que les options en mode console sont assez restreintes. Par conséquent, j'ai choisi de réduire la taille de la page afin de réduire la sensation de vide qui peut survenir lorsque la page est trop grande par rapport à son contenu.
- Graphique : pleine écran, si j'ai opté pour le plein écran c'est parce nous avons plus de choix pour le design du menu et par conséquent nous pouvons agrandir la résolution la deuxième raison est que généralement les jeux se jouent en pleine écran pour avoir une meilleure immersion

3.3.2 : Choix effectués :

Voici l'une de mes maquettes que je vais utiliser pour illustrer mes choix :



Sur ce menu j'ai décidé de représenter l'espace avec différentes planètes dont deux qui ont déjà été envahies par des « invaders ». J'ai choisi d'utiliser cette typographie car c'est celle qui m'a le plus pensé au futur/espace. Pour les boutons j'ai décidé de mettre le texte en vert sur un fond brun pour bien pouvoir faire ressortir le vert. Pour tout ce qui concerne les boutons « Quitter » / « Retour » j'ai décidé de mettre le texte toujours en vert mais sur fond noir car cela permet de dire implicitement que ces boutons n'ont pas la même fonction que les autres.

2.4 : Tests

Nom :	Description :	Étape de test :	Données de test :	Sortie :	Condition :
Mode de jeu	Savoir si l'utilisateur peut accéder à la page du mode de jeu	Appuyer sur « jouer » dans la page d'accueil	-	La page de sélection de mode de jeu apparaît	-
Commencer une partie en Local	Savoir si l'utilisateur va pouvoir commencer une partie en Local	Appuyer sur « Local » dans la page du mode de jeu	-	La partie se lance	Avoir déjà choisi l'option « jouer » dans la page d'accueil
Commencer une partie en multijoueur	Savoir si l'utilisateur va pour commencer une partie en ligne	Appuyer sur « en ligne » dans la page du mode de jeu	-	La partie se lance	Avoir déjà choisi l'option « jouer » dans la page d'accueil
Option	Savoir si l'utilisateur peut accéder à la page des option	Appuyer sur « option » dans la page d'accueil	-	La page des options apparaît	-
Modifier language	Savoir si l'utilisateur peut changer de langue	Appuyez sur l'une des deux flèches	-	La langue est changée	Être dans la menu « option »
Record	Savoir si l'utilisateur peut accéder à la page des record	Appuyer sur « record » dans la page d'accueil	-	La page des records apparaît	-
Vestiaire	Savoir si l'utilisateur peut accéder au vestiaire	Appuyer sur « Vestiaires » dans la page d'accueil	-	La page du vestiaire apparaît	-
Changé de skin	Savoir si l'utilisateur peut changer de skin des ennemies	Appuyer sur « équiper » pour équiper le skin voulu	-	Le bouton change de couleur et le texte est modifié	Être dans la page du vestiaires
Retourner à la page d'accueil	Savoir si depuis tous les pages on peut retourner à la page d'accueil	Appuyer sur « retour » peu importe la page page d'accueil	-	L'utilisateur est redirigé vers la page d'accueil	-

2.5 : Conclusion

Ce projet m'a permis d'exploiter ce que j'ai appris lors des cours théoriques lors d'un cas concret. J'ai eu pas mal de difficulté à faire mes personnas car ChatGPT me donnait pas forcément des informations en lien direct avec le jeu et quand l'IA en donnait je ne les trouvais pas forcément pertinentes. Si le projet était à refaire je ne pense pas que je changerai grand-chose à ma façon de travailler si ce n'est peut-être mieux calculer mon temps pour iceScrum et essayer d'être mieux autonome.

Chapitre 3 : Partie Base de données

Pour cette partie on nous a demandé d'exécuter des requêtes SQL dont une qui sert à importer un dump d'une base de données existante depuis un invite de commande et les autres qui servent à aller chercher des informations qui se trouvent dans la base de données.

3.1 : Importer la base de données

Pour cette partie voici la commande à utiliser pour pouvoir importer la base de données



```
docker exec -i cc94b65c0f303c76950339ad0386b23a592afb747eb69ebe8b6d505180319e3f mysql -uroot -proot < C:\users\pk88yte\Downloads\db_space_invaders.sql
```

Sur cette commande on peut voir un "**docker exec -i cc94b...**" cette partie signifie que l'on va exécuter la commande dans un conteneur docker qui possède l'id "**cc94b...**" la seconde partie va être la commande que l'on va exécuter dans le conteneur dans notre cas la commande va lancer MYSQL avec l'username (-u) "**root**" et le mot de passe (-p) "**root**" et la dernière partie signifie que le conteneur va exécuter le script SQL qui se trouve à l'emplacement indiqué.

3.2 : Requetes

3.2.1 : Requête N°1

Voici la requête que qu'il faut utiliser pour répondre à la demande du cahier des charges :



```
SELECT `jouPseudo`, jouNombrePoints FROM t_joueur ORDER BY `jouNombrePoints` DESC LIMIT 5;
```

Pour cette requête il est nécessaire de sélectionner le pseudo des jours et les leurs nombres de points comme demander dans la requête dans la table qui leur correspond grâce au "**FROM**". Le "**ORDER BY**" va permettre d'ordonner les résultats (par défaut : par ordre croissant) le "**DESC**" quant à lui va permettre d'ordonner par ordre décroissant et pour finir le "**LIMIT 5**" va permettre de sélection seulement les 5 premier résultats.

4.2.2 : Requête N°2

Voici la requête :

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

Sur cette requête il est indispensable de sélectionner le prix le plus haut, le plus bas et Moyen des armes grâce aux fonctions **MAX()**, **MIN()**, **AVG()** il est essentiel renommer les colonnes par les noms imposés grâce à l'alias (**AS**) depuis la table **t_arme**.

3.2.3 : Requête N°3

Voici la requête utilisée :

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

Sur cette commande il est nécessaire de sélectionner le nombre de commande que chaque joueur a fait grâce à la fonction **COUNT()** et de sélectionner l'id des joueurs les colonnes. L'étape d'après consiste à renommer les colonnes avec des alias (**AS**) ensuite il est indispensable de regrouper le tout par joueur (**GROUP BY**). L'étape suivante consiste à ordonner le tout par le nombre de commande du plus grand au plus petit grâce à la clause "**DESC**".

3.2.4 : Requête N°4

Voici la requête :

```
SELECT COUNT(idCommande) AS "NombreCommande", fkJoueur AS "IdJoueur" FROM t_commande GROUP BY fkJoueur
HAVING COUNT(`idCommande`) > 2 ;
```

Sur cette requête les début et pareil que pour la requête N°3 mais en plus de regrouper le tout par joueur il faut regrouper par les joueurs dont le nombre de commande est plus grand que deux s'il est possible de réaliser cette partie c'est grâce à la clause "**HAVING**" qui va mettre une condition sur chaque groupement

3.2.5 : Requête N°5

Pour cette question voici la requête à utiliser :

```
SELECT t_joueur.jouPseudo, t3.armNom FROM t_joueur
JOIN t_commande AS t2 ON t_joueur.idJoueur = t2.fkJoueur
JOIN t_detail_commande AS t1 ON t2.fkJoueur = t1.fkCommande
JOIN t_arme AS t3 ON t1.fkArme = t3.idArme;
```

Pour cette requête il est demandé de sélectionner le pseudo des joueurs et le nom des armes de leurs commande ensuite il est nécessaire de faire des jointures entre les différentes tables pour pouvoir utiliser leurs données, il a été décidé de faire un inner join (**JOIN**) car il est demandé seulement les joueurs qui ont passé une commande.

3.2.6 : Requête N°6

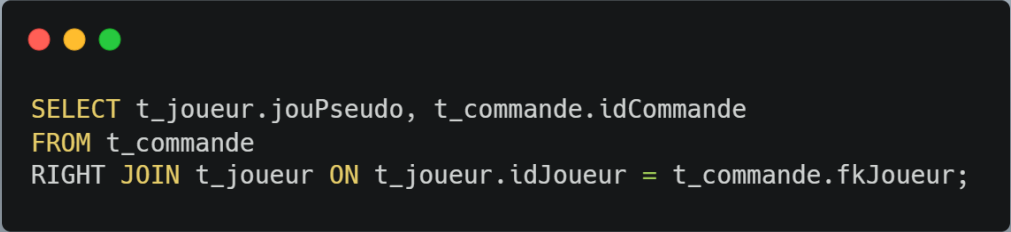
Voici ma requête :

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

Cette requête sélectionne le total dépensé en calculant la somme du prix de l'arme * quantité d'arme dans la commande ensuite elle nomme la colonne "**TotalDepense**". L'étape d'après la requête sélectionne les joueurs de la table commande et renomme la colonne en "**idJoueur**". La suite consiste à joindre les tables pour pouvoir utiliser leurs données ensuite il est nécessaire de faire un **GROUP BY** pour connaître le total par joueur ensuite il faut ordonner les résultats par ordre décroissant en fonction du total dépensé et pour finir la requête prend que les 10 premiers résultats

3.2.7 : Requête N°7

Voici la requête trouvée :

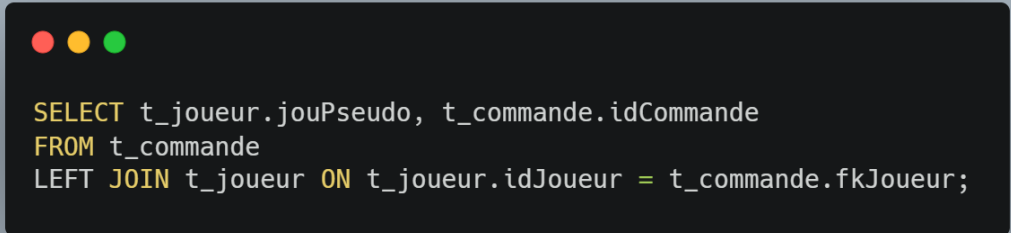


```
SELECT t_joueur.jouPseudo, t_commande.idCommande
FROM t_commande
RIGHT JOIN t_joueur ON t_joueur.idJoueur = t_commande.fkJoueur;
```

Cette requête sélectionne le pseudo des joueurs ainsi que l'id de leurs commandes ensuite elle joint avec un **"RIGHT JOIN"** car il est aussi demandé de sélectionner les joueurs qui n'ont pas passé de commande

3.2.8 : Requête N°8

Voici la requête :



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

La requête est la même que dans la requête N°7 mais au lieu de faire un **"RIGHT JOIN"** elle fait un **"LEFT JOIN"** car il est demandé d'afficher les commandes même si elles existent ou pas

3.2.9 : Requête N°9

Voici ma requête :

```
SELECT SUM(t_detail_commande.detQuantiteCommande), t_joueur.idJoueur
FROM t_joueur
LEFT JOIN t_commande ON t_commande.fkJoueur = t_joueur.idJoueur
LEFT JOIN t_detail_commande ON t_detail_commande.fkCommande = t_commande.idCommande
GROUP BY t_joueur.idJoueur;
```

Cette requête sélectionne le nombre total de commande qu'un joueur a passé ensuite elle joint les tables entre elles avec un "LEFT JOIN" car il est aussi demandé de sélectionner les joueurs qui n'ont pas passé de commande ensuite la requête groupe le tout par joueur.

3.2.10 : Requête N°10

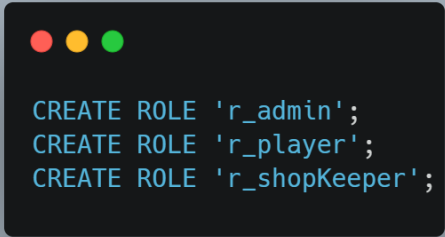
Voici la requête :

```
SELECT idJoueur
FROM t_joueur
JOIN t_commande ON fkJoueur = idJoueur
JOIN t_detail_commande ON fkCommande = idCommande
GROUP BY idJoueur
HAVING COUNT(DISTINCT fkArme) > 3;
```

Cette requête sélectionne l'id des joueurs ensuite elle joint les tables entre elles avec un "JOIN" ("INNER JOIN") car nous voulons seulement les joueur qui ont passé commande ensuite elle trie le tout par joueur mais seulement si le nombre d'arme différentes (COUNT(DISTINCT fkArme)) et plus grand que trois

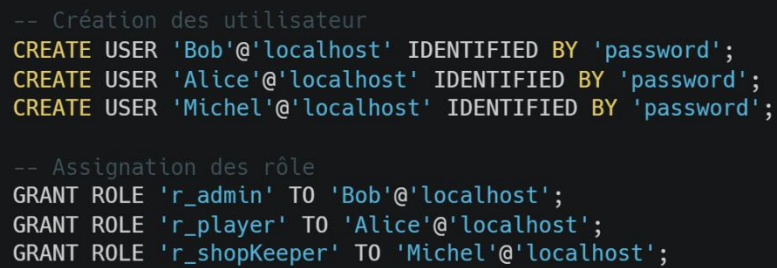
3.3 : Création des types d'utilisateur et utilisateur

Une partie du projet consiste à donner des permissions a des types d'utilisateurs mais pour ce faire nous devons d'abord créer ces types d'utilisateur et dans ce but il est nécessaire d'utilisé ces trois requêtes :



```
CREATE ROLE 'r_admin';  
CREATE ROLE 'r_player';  
CREATE ROLE 'r_shopKeeper';
```

Il a été nécessaire de créer un utilisateur par type voici les requêtes qu'il faut utiliser pour les créer et assigner les utilisateurs à leurs rôles :



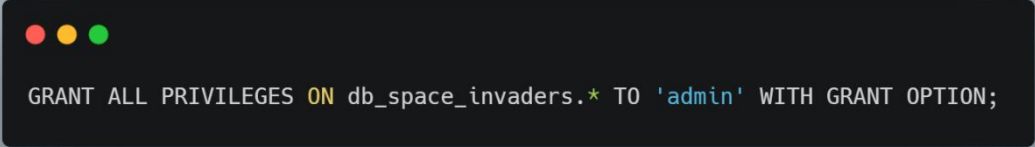
```
-- Création des utilisateur  
CREATE USER 'Bob'@'localhost' IDENTIFIED BY 'password';  
CREATE USER 'Alice'@'localhost' IDENTIFIED BY 'password';  
CREATE USER 'Michel'@'localhost' IDENTIFIED BY 'password';  
  
-- Attribution des rôle  
GRANT ROLE 'r_admin' TO 'Bob'@'localhost';  
GRANT ROLE 'r_player' TO 'Alice'@'localhost';  
GRANT ROLE 'r_shopKeeper' TO 'Michel'@'localhost';
```

Dans la première partie, la requête crée les utilisateurs dans l'host "*localhost*" et leurs donnent le mot de passe "*password*" grâce au "**IDENTIFIED BY**". Ensuite elle assigne les utilisateurs au rôle.

3.4 : Assignment des permissions

3.4.1 : Rôle "r_admin"

Voici la requête qu'il faut utiliser pour donner les permissions d'écriture, lecture, suppression, mis à jour, gestion des permissions pour ce rôle :

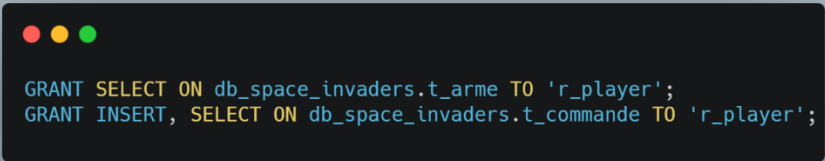


```
GRANT ALL PRIVILEGES ON db_space_invaders.* TO 'admin' WITH GRANT OPTION;
```

Sur cette requête donne toutes les permissions au rôle (select, insert, ...) grâce au "**ALL PRIVILEGES**" sur la base de données **db_space_invaders** avec la permission de pouvoir gérer les utilisateurs

3.4.2 : Rôle "r_player"

Voici la requête qui donner les permissions de Lire les informations des armes, créer une commande et lire toutes les commandes :



```
GRANT SELECT ON db_space_invaders.t_arme TO 'r_player';  
GRANT INSERT, SELECT ON db_space_invaders.t_commande TO 'r_player';
```

La première requête donne uniquement la permission de sélectionner les données de la table **t_arme** pour pouvoir sélectionner les informations d'une arme. Ensuite la requête donne les permissions d'insérer et de sélectionner uniquement sur la table **t_commande** comme demandé afin de créer des command et de voir les commandes en cours.

3.4.3 : Rôle "r_shopKeeper"

Voici la requête utilisée pour donner les permissions de lire les informations sur tous les joueurs, mettre à jour, lire et supprimer des armes et lire toutes les commandes :

```
GRANT SELECT ON db_space_invaders.t_commande TO 'r_shopKeeper';  
GRANT SELECT ON db_space_invaders.t_joueur TO 'r_shopKeeper';  
GRANT ALTER, SELECT, DELETE ON db_space_invaders.t_arme TO 'r_shopKeeper';
```

En premier lieu la requête donne les permissions de lire les informations sur les tables **t_commande** (pour pouvoir lire toutes les commandes) et sur la table **t_joueur** (pour pouvoir lire toutes les informations d'un joueur), ensuite je lui donne les permissions de modifier, lire et supprimer sur la table **t_armes** pour pouvoir gérer la vente des armes.

3.5 : exporter / importer la base de données

Pour exporter un dump de la base de données voici la commande nécessaire afin d'exporter une base de données dans l'invite de commande windows :

```
docker exec -i {id container docker} mysqldump -uroot -proot --databases db_space_invades >  
db_space_invaders.sql
```

Dans cette commande ouvre un terminale docker grâce au "**docker exec -i {id container docker}**" ensuite elle exécute la commande "**mysqldump**" qui va exporter ma base de données avec l'utilisateur root. Le "**--databases**" va permettre d'insérer la création et l'utilisation de la base de données dans le script car par défaut ils n'y sont pas ensuite la requête indique la base de données à exporter. L'étape d'après consiste à donner le chemin d'accès où le fichier va se créer le signe ">" va permettre d'importer VERS l'endroit indiqué.

Pour restaurer le dump de la base de données il est nécessaire d'exécuter cette commande toujours dans le terminal Windows :

```
docker exec -i cc94b65c0f303c76950339ad0386b23a592afb747eb69ebe8b6d505180319e3f mysql -uroot -proot <  
C:\users\pk88yte\Downloads\db_space_invaders.sql
```

Cette commande ouvre un terminale docker grâce au "**docker exec -i cc94...**" ensuite elle exécute la commande "**mysql**" qui va restaurer la base de données avec l'utilisateur root ensuite il est primordial d'indiquer le chemin d'accès du fichier .sql le signe "<" est important car il veut dire que cela va importer le fichier.

3.6 : Création des index

1. MySql génère automatiquement des index pour les clés primaires et étrangères afin de réaliser des jointures. Il génère aussi des index pour les champs uniques afin d'augmenter la rapidité des requêtes de type "**SELECT**".
2. Un index va permettre d'optimiser les requêtes de type "**SELECT**" pour cela il va créer ce que l'on appelle un "*B-Tree*" qui est un arbre auto-équilibré dans lequel chaque nœud contient les données sous une forme triée. L'inconvénient c'est que ça prend beaucoup de stockage de plus les requêtes du type "**INSERT**", "**ALTER**" et "**DELETE**" sont ralenties car en plus de mettre à jour la table elles doivent mettre à jour l'index.
3. Il serait judicieux de créer un index composite pour les champs "**armNom**", "**armDescription**" et "**armPrix**" dans la table **t_arme**, étant donné que ces valeurs ne sont pas susceptibles de changer fréquemment, et il est peu probable qu'il y ait fréquemment de nouvelles armes ajoutées de plus se sont des données qui apparaîtront souvent ensemble notamment dans la boutique.

3.7 : Intégration de la base de données dans le programme C#

Pour pouvoir se connecter il a été nécessaire d'utiliser une chaîne de connexion qui est une expression qui contient les valeurs qui sont nécessaires à la connexion à la base de données voici les informations dont on a besoin :

- Le serveur : (SERVER) il s'agit de l'adresse Ip du serveur dans notre cas il s'agit de localhost (127.0.0.1)
- La base de données : (DATABASE) il s'agit du nom de la base de données dans le cadre du projet il s'agit de "**db_space_invaders**"
- L'utilisateur : (UID) il s'agit du nom d'un utilisateur de la base de données dans notre cas il s'agit de "**root**"
- Le mot de passe : (PASSWORD) il s'agit du mot de passe de l'utilisateur dans notre cas il s'agit de "**root**"
- Le port : (PORT) celui n'est pas obligatoire à mettre car MYSQL possède déjà son propre port qui est le 3306 mais comme dans le cadre du projet on utilise docker qui donne un port à utiliser il est obligatoire de mettre le port (**6033**) dans la chaîne de connexion

3.8 : Conclusion

Ce projet m'a permis de bien comprendre et appliquer ce qu'on avait appris dans les cours théoriques de plus il m'a permis d'apprendre comment connecter une application C# à une base de données. Si ce projet était à refaire je ne pense pas que je changerais grand-chose si ce n'est mon autonomie je posais souvent des questions pour me rassurer.

Chapitre 4 : Conclusion Final

Le projet dans sa globalité a été particulièrement amusant à réaliser, et j'ai pu faire preuve d'une grande productivité. Si jamais l'opportunité se présentait de refaire ce projet dans son entièreté, j'accorderais davantage d'attention à la gestion du temps passé avec IceScrum, tout en améliorant ma capacité à gérer mon autonomie de manière plus efficace.