

1 INFORMATIONS GENERALES

Élève :	Nom:	Prénom:
Lieu de travail :	ETML / Sébeillon 12 / 1004 Lausanne	
Client UX	Nom:	Prénom:
	□:	
Client DB	Nom:	Prénom:
	□:	
Client OO	Nom:	Prénom:
	□:	
Dates de réalisation :	1 ^{er} trimestre	
Temps total :	~60 périodes	

2 PROCÉDURE

- Tous les apprentis réalisent le projet sur la base d'un cahier des charges.
- Le cahier des charges est présenté, commenté et discuté en classe.
- Les apprentis sont entièrement responsables de la sécurité et sauvegarde de leurs données.
- En cas de problèmes graves, les apprentis avertissent le client au plus vite.
- Les apprentis ont la possibilité d'obtenir de l'aide externe, mais ils doivent le mentionner.
- Les informations utiles à l'évaluation de ce projet sont disponibles au chapitre 8.

3 TITRE

SpicyInvader

4 SUJET

Faire un réplica du célèbre jeu SpaceInvader.

5 MATÉRIEL ET LOGICIEL À DISPOSITION

- Un PC ETML
- Accès à Internet
- <https://etml.icescrum.com>
- <https://figma.com>

6 PRÉREQUIS

Modules de programmation de base

7 CAHIER DES CHARGES

7.1 Gestion de projet

1. Utilisation de IceScrum
 - Stories à faire valider au CDP avant l'implémentation
 - Chaque story contient
 - les tests d'acceptance (avec maquette)
 - les tâches techniques associées
 - Un seul sprint pour tout le projet

7.2 Qualité

1. Réaliser un programme informatique de qualité
 - Organisé (namespace, classes, commit log,...)
 - Compacté (pas de copié/collé,...)
 - Optimisé (utilisation de structures adaptées)
 - Testé
 - Commenté
2. Prouver que vous êtes digne de confiance lorsqu'on vous confie un projet
 - Travail avec IceScrum selon les directives (à jour, ...)
 - Pro-activité
 - **Poser des questions** au client
 - Faire des démonstrations
 - Utiliser un système de versioning de code (GIT)

7.3 Fonctionnalités requises (du point de vue client)

7.3.1 Réplica du célèbre jeu SpicyInvader

Squelette exemple : <https://ici.section-inf.ch/spicy-draft>

- a. POO
 - i. Un vaisseau avec possibilité de tir et déplacement
 - ii. 10 ennemis qui descendent sur l'axe vertical
- b. DB
 - i. Afficher les 5 joueurs qui ont les meilleurs scores. Les joueurs doivent être classés dans l'ordre décroissant
- c. UX
 - i. Menu

7.3.2 Spécificités UX

Au niveau de l'UX, l'objectif du projet est de proposer un prototype (menu et options de jeu) cliquable comprenant les options suivantes (qui ne seront pas implémentées) :

- Jeu solo ou multijoueur
- Choix de palettes graphiques additionnelles
- Choix de design différents pour les ennemis
- Page de "highscore"
- Choix de Gameplay (jouabilité, accessibilité, éco-conception)

A la fin du projet, le groupe devra livrer à l'enseignant :

- Une documentation comprenant :
 - Une introduction avec explication du contexte
 - Un chapitre d'analyse :
 - Conception centrée utilisateur
 - Création de deux profils de joueurs sous forme de "Personas"
 - Choix de la palette graphique
 - Éco-conception
 - Accessibilité
 - Un chapitre de conception
 - Définition des écrans
 - Choix effectués
 - Un chapitre d'évaluation
 - Tests (A/B tests, test d'utilisabilité)

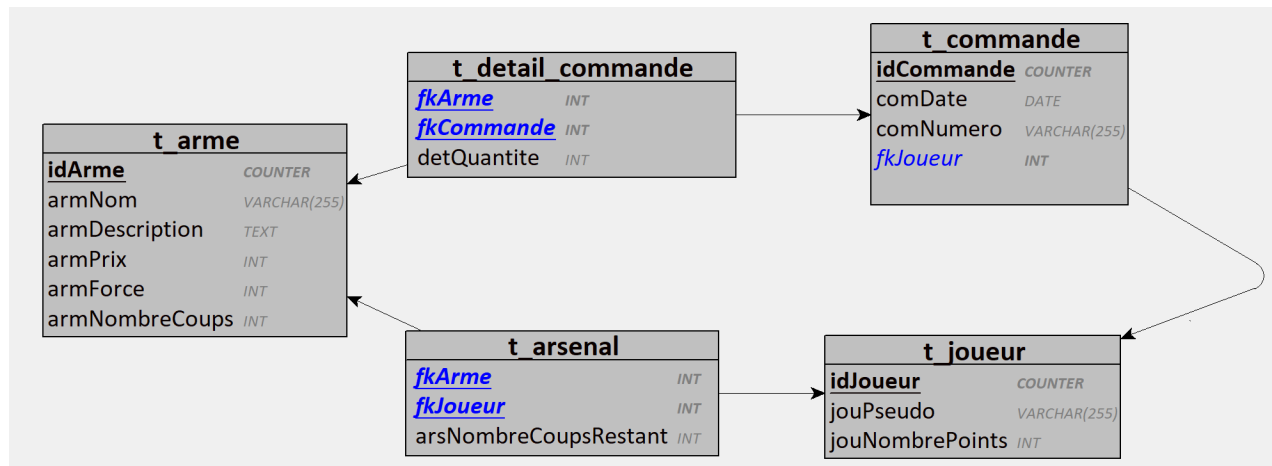
7.3.3 Spécificités POO

1. Programmation orientée objet et MVC
2. Tests unitaires
3. Documentation
 - Automatique du code
 - Schémas et description des éléments principaux
 - Interactions entre les classes
 - Paramètres
 - Cas particuliers

7.3.4 Spécificités DB : Répondre aux questions concernant la DB **db_space_invaders**

Attention : Il n'est pas demandé d'implémenter la base de données **db_space_invaders** dans votre application C#. Vous devez uniquement répondre aux questions de cette section.

Voici le **MLD** (Modèle Logique de Données) de la base de données **db_space_invaders** :



Quelques informations complémentaires :

- Pour acheter des armes, le joueur utilise le nombre de points **jouNombrePoints** qu'il a obtenu en détruisant ses vaisseaux ennemis
- Le champ **armForce** indique le dégât que peut faire une arme offensive (0 pour une arme défensive)
- Le champ **armNombreCoups** indique combien de « cartouches » on dispose avec cette arme (offensive ou défensive)

Remarque : Si vous avez d'autres questions sur ces différentes tables, vous devez aller les poser à votre enseignant !

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

Vous avez à disposition un fichier **db_space_invaders.sql**

Ce fichier est un dump MySQL de la base de données ayant pour nom **db_space_invaders**.

Travail demandé :

Charger le dump de la DB dans MySQL.

Expliquer en détail la commande utilisée.

B. Gestions des utilisateurs

Différents utilisateurs vont devoir interagir avec la base de données.

Pour ce faire, ces utilisateurs ont des missions différentes :

1. Administrateur du jeu :

- Peut créer, lire, mettre à jour et supprimer (CRUD) n'importe quelle table.
- Gérer les utilisateurs et leurs privilèges.

2. Joueur :

- Lire les informations des armes (pour voir quelles armes il peut acheter).
- Créer une commande.
- Lire toutes les commandes.

3. Gestionnaire de la boutique :

- Lire les informations sur tous les joueurs (pour savoir qui a passé une commande).
- Mettre à jour, lire et supprimer des armes (ajout de nouvelles armes, modification des prix, etc.).
- Lire toutes les commandes.

Pour chaque « catégorie d'utilisateurs », on pourrait être amené à créer plusieurs voire même un grand nombre d'utilisateurs.

Travail demandé :

Vous devez expliquer, comment mettre en place ces différents « types d'utilisateurs » afin de rendre la gestion des utilisateurs la plus simple possible.

Pour cela, vous devez définir un ensemble de requêtes SQL.

Vous devez créer au moins un utilisateur par « type d'utilisateurs ».

C. Requêtes de sélection

Voici une liste de requêtes SQL que vous devez réaliser et expliquer en détail.

Requête n°1 :

La première requête que l'on vous demande de réaliser est de sélectionner les 5 joueurs qui ont le meilleur score c'est-à-dire qui ont le nombre de points le plus élevé. Les joueurs doivent être classés dans l'ordre décroissant

Requête n°2 :

Trouver le prix maximum, minimum et moyen des armes.

Les colonnes doivent avoir pour nom « PrixMaximum », « PrixMinimum » et « PrixMoyen »

Requête n°3 :

Trouver le nombre total de commandes par joueur et trier du plus grand nombre au plus petit.

La 1ère colonne aura pour nom "IdJoueur", la 2ème colonne aura pour nom "NombreCommandes"

Requête n°4 :

Trouver les joueurs qui ont passé plus de 2 commandes.

La 1ère colonne aura pour nom "IdJoueur", la 2ème colonne aura pour nom "NombreCommandes"

Requête n°5 :

Trouver le pseudo du joueur et le nom de l'arme pour chaque commande.

Requête n°6 :

Trouver le total dépensé par chaque joueur en ordonnant par le montant le plus élevé en premier, et limiter aux 10 premiers joueurs.

La 1ère colonne doit avoir pour nom "IdJoueur" et la 2ème colonne "TotalDepense"

Requête n°7 :

Récupérez tous les joueurs et leurs commandes, même s'ils n'ont pas passé de commande.

Dans cet exemple, même si un joueur n'a jamais passé de commande, il sera quand même listé, avec des valeurs `NULL` pour les champs de la table `t_commande`.

Requête n°8 :

Récupérer toutes les commandes et afficher le pseudo du joueur s'il existe, sinon afficher `NULL` pour le pseudo.

Requête n°9 :

Trouver le nombre total d'armes achetées par chaque joueur (même si ce joueur n'a acheté aucune Arme).

Requête n°10 :

Trouver les joueurs qui ont acheté plus de 3 types d'armes différentes

D. Création des index

En étudiant le dump MySQL **db_space_invaders.sql** vous constaterez que vous ne trouvez pas le mot clé **INDEX**.

1. Pourtant certains index existent déjà. Pourquoi ?
2. Quels sont les avantages et les inconvénients des index ?
3. Sur quel champ (de quelle table), cela pourrait être pertinent d'ajouter un index ? Justifier votre réponse.

E. Backup / Restore

Nous souhaitons réaliser une sauvegarde (Backup) de la base de données **db_space_invaders**.

Ensuite, nous souhaitons nous assurer que cette sauvegarde est correcte en la rechargeant dans MySQL (opération de restauration).

Donner la commande permettant de faire :

- Un backup de la base de données **db_space_invaders**
- Un restore de la base de données **db_space_invaders**

En expliquant en détail chaque commande utilisée.

7.4 Livrables

1. Rapport des tâches PDF généré avec le plugin IceScrum
2. 1 release sur Github contenant
 - a. Une archive ZIP avec les éléments UX
 - b. Une archive ZIP avec les éléments DB
 - c. Une archive ZIP avec les éléments POO
 - d. [Code source inclus automatiquement]

8 Évaluation

1. Auto-évaluation challengée par le client basé sur des éléments observables inscrits dans la partie « retrospective » de IceScrum.
2. Le recours à des outils en ligne d'intelligence artificielle (ex. : Chat GPT) doit être mentionné et ne peut servir que d'inspiration à la réalisation. En cas d'abus, l'évaluation du projet en tiendra compte.