1 INFORMATIONS GENERALES

Élève :	Nom:	Prénom:
Lieu de travail :	ETML / Avenue de Valmont 28b, 1010 Lausanne	
Client UX	Nom:	Prénom:
	□:	
Client DB	Nom:	Prénom:
	□:	
Client OO	Nom:	Prénom:
	□:	
Dates de réalisation :	1er trimestre	
Temps total :	80 périodes (32 P_OO + 24 P_DB + 24 P_UX)	

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

Shoot me up !

4 SUJET

Concevoir un jeu 2D modulaire de tir à la troisième personne et en réaliser une partie.

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
- Modules de bases de données de base
- ICT-320 en cours
- ICT-322 en cours
- ICT-106 en cours

7 CAHIER DES CHARGES

7.1 Gestion de projet

1. La planification est à faire selon les instructions spécifiques de votre chef de projet.
2. Un journal de travail devra être rendu. L'outil que vous utilisez est libre, mais les caractéristiques suivantes doivent être respectées :
    - La structure et la présentation sont claires et soignées.
    - Les sources, les fichiers, les répertoires, les commits, et autres sources d'informations concernées par le journal sont référencés.
    - L'état et les durées des tâches mentionnées sont précisés.
    - Toutes les activités planifiées, les aides extérieures, ainsi que les imprévus et les heures supplémentaires y sont mentionnés.
    - Les succès et les échecs sont mentionnés.
    - Le travail journalier et son appréciation critique, ainsi que les réflexions y figurent.

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 (tests unitaires)
    - Commenté
    - Complet (code, script DB, maquettes PDF, éxécutable, …)
2. Prouver que vous êtes digne de confiance lorsqu'on vous confie un projet
    - Journal de travail 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 d'un « shoot'em up 2D », comme Space Invader

a. Maquettes
   i. Menu principal
   ii. Ecran de jeu (niveau)
   iii. Éditeur de niveau (voir détails ci-dessous)
   iv. High scores

b. Contraintes de réalisation
   i. Un concept de niveaux décrivant
      1. Le numéro du niveau (Level 1, Level 2, …)
      2. Le joueur
         a. Déplacements
         b. Nombre de vies
         c. Capacités de tir : direction, rafale, cooldown, décompte munitions, recharge, …
         d. Un sprite
      3. Les ennemis du niveau avec (pour chaque type)
         a. Nombre de vies
         b. Minutage d'apparition
         c. Tir (oui / non)
         d. Un sprite
      4. Les obstacles avec (pour chaque type)
         a. Une taille
         b. Une position X,Y
         c. Un sprite
         d. Le comportement en cas de dégâts (tir, collision)
   ii. Structure et données des niveaux décrits et stockés dans une base de données relationnelle

c. Fonctionnalités
   i. Au moins 2 niveaux implémentés avec
      1. Joueur
      2. Ennemis
      3. Obstacles
   ii. Gestion des highscores (en base de données)

7.3.2 Spécificités UX

La documentation contenue dans livraison finale du projet (Github) comprend :

- Un chapitre d'analyse de l'UX:
  - Conception centrée utilisateur
    - Création de deux profils de joueurs sous forme de "Personas"
  - Choix de la palette graphique
  - Eco-conception
  - Accessibilité
- Un chapitre de conception
  - Définition de tous les écrans - maquettes base-fidélité / low-fidelity wireframes
  - Ecran "Éditeur de niveau" - maquette haute-fidélité / high-fidelity wireframe
  - Choix effectués
- Un chapitre d'évaluation
  - Tests (A/B tests, test d'utilisabilité)

7.3.3 Spécificités POO

La livraison finale du projet (Github) comprend :

1. Le code
   - Programmation orientée objet
   - Tests unitaires
2. Documentation (chapitres du rapport)
   - Une analyse fonctionnelle centrée utilisateur sous forme de User Stories
   - Automatique du code (manuel de référence)
   - Schémas des classes
   - Au moins un détail d'implémentation spécifique

7.3.4 Spécificités DB

1. MCD
2. MLD
3. Script SQL de création

7.4 Livrables

Il n'y a qu'un livrable: une release Github, à laquelle est attaché le rapport, au format PDF, contenant :

a. Introduction
b. Planification
c. Analyse fonctionnelle
d. Maquettes
e. Schémas (DB, POO, diagrammes de classe)
f. Manuel de référence des classes
g. Rapport de tests
h. Journal de travail
i. Chapitre explicatif de l'usage fait de l'IA dans ce projet

8 Évaluation

1. Auto-évaluation challengée par le client basé sur des éléments observables.
2. Le recours à des outils en ligne d'intelligence artificielle (ex. : Chat GPT) est autorisé mais ne peut servir que d'inspiration à la réalisation. Chaque développeur doit être à tout moment en mesure d'expliquer le code de manière précise et convaincante.
   En cas d'abus, l'évaluation du projet en tiendra compte.