TECH

CAHIER DES CHARGES

SHOOT'EM UP

INTRODUCTION

Nom du projet : Shoot'em up

Projet N° : 05

Date : 02/12 - 20/12

Classe : GTECH-2

MODALITÉS

En groupe de 3

CONTEXTE ET DESCRIPTION DU PROJET

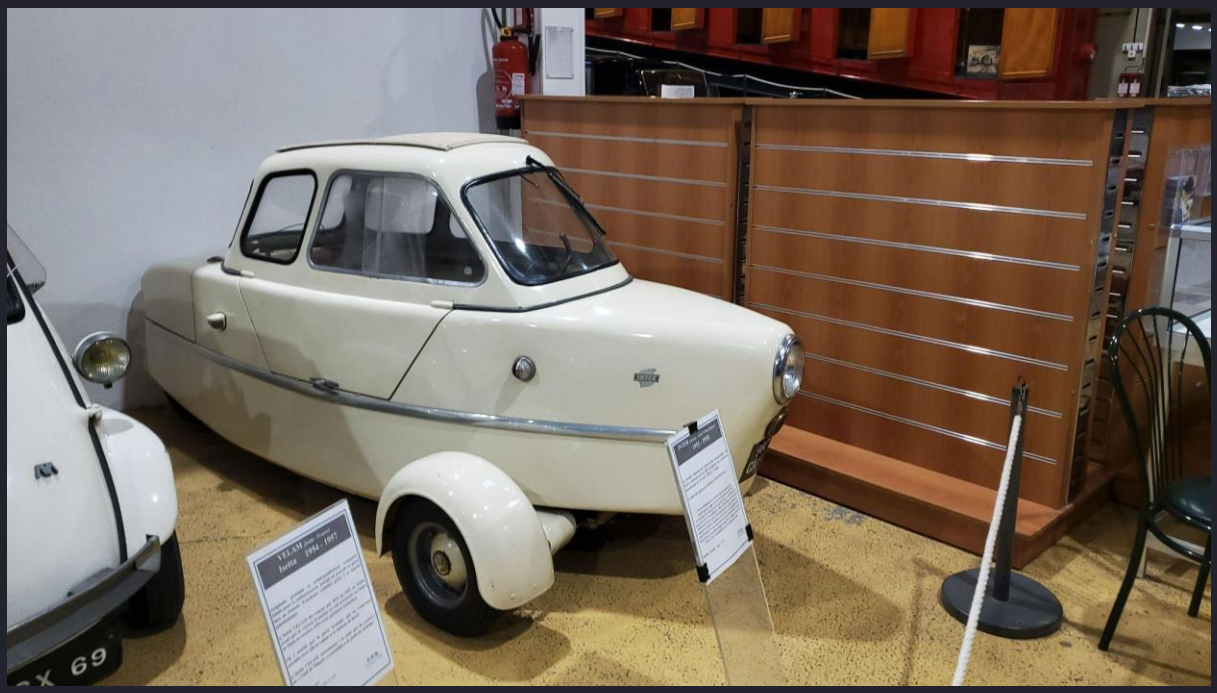
Brève description du contexte dans lequel le projet s'inscrit et détails du projet :

Ce projet consiste à développer un **shoot'em up 2D horizontal** en **C++**, en utilisant la bibliothèque de rendu du **GC-Engine**, le moteur de jeu conçu par les étudiants du Gaming Campus.

Les thèmes choisies sont en relation avec le projet du musée Malartre et s'intégreront dans le jeu qui devrait sortir au 15 avril 2025 en fonction des équipes de 3ème année qui se positionneront .

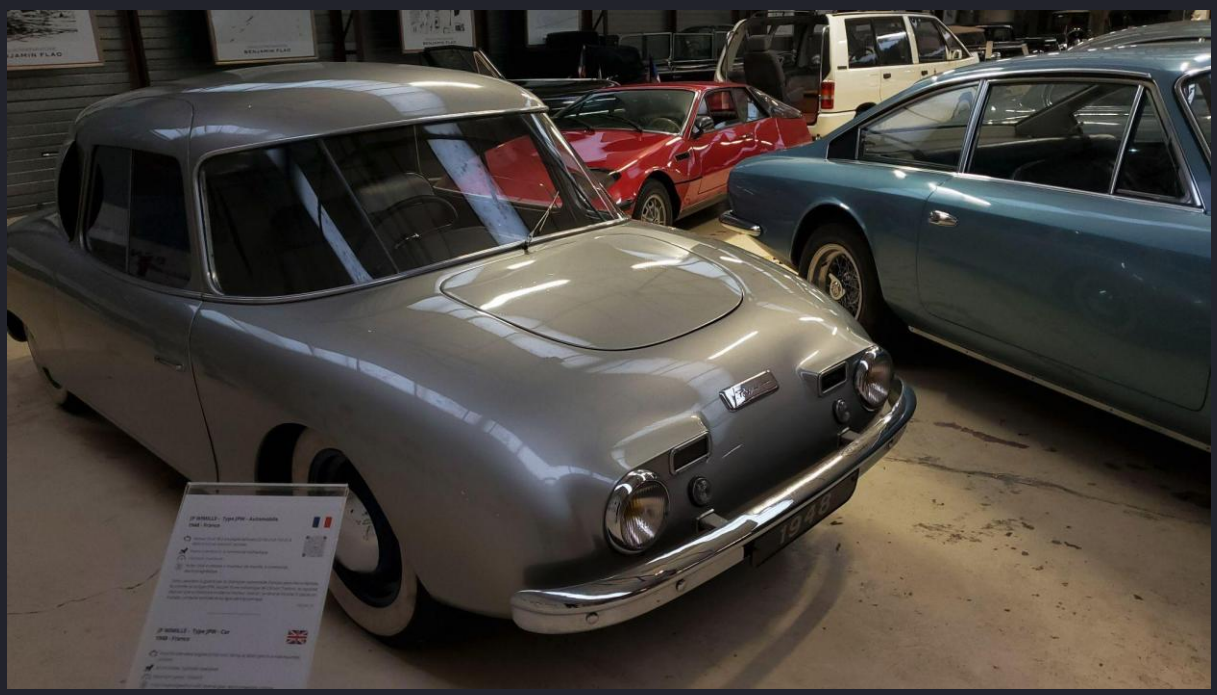
1er thème : le véhicule "aquatique", celui ci ne pourra se déplacer qu'en bord de Saône mais pas dans la Saône (souhait du musée pour faire découvrir)

Le véhicule doit être reproduit en 2d ou /et en 3D :



2ème thème : le véhicule “futuriste”, celui ci pourra se déplacer dans un environnement futuriste

Le véhicule doit être reproduit en 2d ou /et en 3D :



Le joueur contrôlera les deux voitures , traversant un environnement hostile, peuplé d'obstacles et d'ennemis divers.

L'objectif est de créer un jeu complet, comportant au minimum un niveau entièrement jouable, se terminant par un combat contre un boss, ainsi qu'un écran de démarrage, un écran de game-over, et un écran de fin après la victoire sur le boss.

Pour enrichir l'expérience de jeu, le projet inclura des éléments visuels tels qu'un défilement en parallaxe pour simuler la profondeur, ainsi que des feedbacks sonores et une musique d'accompagnement, grâce à une **bibliothèque audio**. Le projet comprendra également un éditeur de niveau permettant de concevoir et de personnaliser le niveau en plaçant des ennemis et des obstacles à des positions spécifiques.

Specs

Vous devez implémenter dans votre application :

1. Gameplay

- **Déplacement du vaisseau :**
    - Le joueur peut déplacer son vaisseau dans toutes les directions (haut, bas, gauche, droite) dans une zone de jeu rectangulaire limitée.
- **Système de tir :**
    - **Projectiles standards :** Le joueur peut tirer des projectiles à mouvement rectiligne qui se déplacent horizontalement à travers l'écran.
    - **Projectiles à tête chercheuse :** Le joueur peut également tirer un nombre limité des projectiles à tête chercheuse qui se dirigent automatiquement vers l'ennemi le plus proche.
- **Types d'ennemis :**
    - Le jeu comprend trois types d'ennemis, chacun ayant un pattern de mouvement et de tir spécifique.
    - Chaque type d'ennemi tirera des projectiles différents, nécessitant des stratégies de contournement ou de destruction spécifiques.
- **Obstacles destructibles :**
    - Des obstacles destructibles seront disposés dans le niveau, forçant le joueur à adapter ses mouvements et ses tirs.
- **Mécanique de boss :**

- Le jeu se termine par un combat contre un boss qui dispose de plusieurs patterns de mouvements et d'attaques complexes.

2. Outils

- **Éditeur de niveau textuel :**
  - Implémenter un éditeur de niveau textuel permettant de concevoir des niveaux en plaçant des ennemis et des obstacles à des positions spécifiques.
  - Chaque ennemi et obstacle est associé à un caractère, permettant de les positionner facilement dans un fichier texte.

3. Habillage

- **Défilement en parallaxe :**
  - Chaque couche de l'arrière-plan se déplace à une vitesse différente pour créer un effet de parallaxe.
- **Feedbacks sonores et musique :**
  - Le jeu doit intégrer des effets sonores pour les tirs, les explosions, et autres feedbacks.
  - Une musique d'ambiance doit jouer en arrière-plan durant la partie.
  - Les sons et la musique peuvent être obtenus à partir de banques d'assets, générés par une IA, ou créés manuellement si vous avez le temps et l'envie.
- **Interface utilisateur :**
  - **Écran d'Accueil :** Présente un écran d'accueil demandant au joueur d'appuyer sur un bouton pour commencer.
  - **Écran de Game-Over :** Affiche un écran de game-over en cas de mort du joueur et propose de recommencer une partie.
  - **Écran de Fin de Jeu :** Affiche les crédits après la défaite du boss final.
  - **Affichage en Jeu :** Le jeu doit afficher en permanence les points de vie du joueur et son score, qui augmente à chaque ennemi détruit.

Pour les plus avancés uniquement (ou pour aller plus loin)

- **Plusieurs niveaux :** Implémenter plusieurs niveaux avec des ennemis différents et un boss unique pour chaque niveau.
- **Menu options :** Ajouter un menu permettant de régler le volume de la musique et des effets sonores.
- **Éditeur de niveau graphique :** Créer un éditeur de niveau avec interface graphique permettant de placer directement les ennemis et de tester des séquences du niveau en temps réel.

OBJECTIFS PÉDAGOGIQUES ET PROFESSIONNELS DU PROJET

A l'issue de ce projet, les étudiants seront capables de :

Savoirs

- **Connaître** les fonctionnalités et les composants essentiels d'un jeu vidéo, tels que les mécanismes de mouvement, la gestion des collisions et le contrôle de la caméra.
- **Comprendre** les principes du défilement en parallaxe pour créer une illusion de profondeur.
- **Maîtriser** les bases de la programmation en C++ et l'utilisation d'une bibliothèque graphique.

Savoir-être

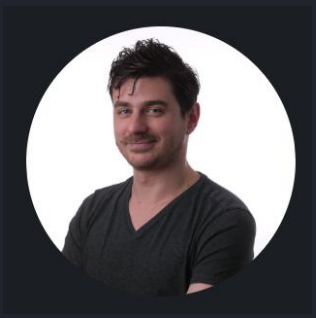
- **Faire preuve de rigueur** dans la structuration et l'organisation du code pour assurer sa clarté et sa maintenabilité.
- **Collaborer efficacement** avec les membres de l'équipe pour intégrer les différents aspects du jeu, tels que les mécaniques de jeu, l'interface utilisateur et les outils d'aide au développement.
- **Faire preuve de créativité** et d'adaptabilité pour résoudre les problèmes rencontrés durant le développement, comme les bugs ou les contraintes techniques..

- **Gérer le stress et la pression** en respectant les délais du projet tout en assurant la qualité du produit final.

Savoir-faire

- **Implémenter** des mécaniques de jeu, telles que les déplacements, les systèmes de tir, et les comportements des ennemis..
- **Créer** et **Utiliser** un éditeur de niveau textuel pour créer et personnaliser des niveaux en plaçant des ennemis et des obstacles à des positions spécifiques.
- **Intégrer** des éléments visuels et sonores, tels que le défilement en parallaxe, les effets sonores et la musique d'ambiance, pour enrichir l'immersion du joueur.
- **Optimiser** le gameplay en ajustant les paramètres du jeu pour offrir une expérience fluide et engageante.

INTERVENANT



Nom : Peter Vystavel
Titre : Développeur C/C++
LinkedIn : https://www.linkedin.com/notifications/?filter=all

BIOGRAPHIE

Développeur informatique depuis le début des années 2000, j'ai commencé en autodidacte en C/C++ et ai créé plusieurs petits jeux ainsi qu'un moteur 2D avant d'obtenir un Master en informatique en 2016. J'ai ensuite travaillé comme développeur gameplay chez **Eugène Systems** (*Wargame*, *Steel Division*), puis comme développeur moteur chez **Kylotonn Games** (*WRC*, *V-Rally 4*).

En 2017, j'ai lancé mon premier jeu, **Space Note**, suivi en 2024 de **Space Note 2** sur Android et iOS. Aujourd'hui, j'ai cofondé une entreprise avec quatre associés, et nous travaillons actuellement sur notre troisième projet, un jeu d'action/aventure pour PC et consoles.

ROADMAP

DESCRIPTION

Veuillez trouver ci-dessous une description des livrables attendus ainsi que les dates d'échéance associées. Il est essentiel de respecter les échéances suivantes pour assurer une progression harmonieuse et structurée du projet. Chaque livrable représente une étape importante dans le processus de réalisation et permet d'évaluer l'avancement du travail. Les dates limites fixées doivent être rigoureusement respectées afin de garantir une évaluation équitable et de permettre un feedback constructif en temps opportun.

Jalon	Livrables attendus	Date limite	Moyens / formats
1	**Travail préparatoire**		
2			
3			
4	**Projet final**		

5	**Bilan du projet**		

EVALUATION

L'évaluation est conçue pour être holistique, prenant en compte non seulement le produit final, mais aussi le processus, les compétences acquises et les attitudes démontrées tout au long du projet. Les compétences sont classées en trois catégories principales : "savoirs" (connaissances théoriques), "savoir-faire" (compétences pratiques), et "savoir-être" (compétences interpersonnelles et attitudes).

SYSTÈME DE NOTATION

Savoirs (Connaissances)

- **Compréhension théorique** : Évaluée soit en amont du projet pendant la semaine théorie soit lors de la soutenance et restitution du projet. Cela permet de mesurer la compréhension des concepts fondamentaux et des connaissances liées au projet des étudiants.

Savoir-faire (Compétences)

- **Compétences techniques et application** : Évaluées à travers la soumission finale du projet. Cela inclut la qualité, la fonctionnalité, et la précision technique du travail produit.
- **Gestion de projet** : Évaluée en fonction de l'organisation, du respect des échéances, et de l'utilisation efficace des ressources. Cela peut être évalué à travers la documentation du projet et les journaux de processus.

Savoir-être (Attitudes/Compétences interpersonnelles)

- **Travail d'équipe et collaboration** : Évalués à travers des évaluations par les pairs et les membres du groupe. Les critères incluent la communication, la coopération, et la contribution aux tâches du groupe.
- **Autonomie et initiative** : Évaluées en fonction des contributions individuelles, de la capacité à travailler de manière autonome, et de la résolution proactive des problèmes.
- **Soutenance** : Évaluées lors de la présentation finale du projet. Les critères incluent la clarté, la cohérence, et la capacité à articuler et défendre les résultats du projet.

GRILLE D’ÉVALUATION

La grille d’évaluation du projet devra impérativement reprendre les savoirs/savoir-être/savoir-faire qui ont été cités à la partie Objectifs pédagogiques. Vous êtes libre de déterminer combien de points à accorder à chaque connaissance/compétence, mais la note finale doit être sur 20.

EXAMEN INDIVIDUEL DE FIN DE PÉRIODE
Notez ici les éléments relatifs au partiel : acquis à valider / modalités envisagées

.....
.....
.....
.....