Projet de Fin d’Année

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| **Version** | **Deadline** |
| **Alpha** | 30/05/2025 – 17h00 |
| **Beta** | 06/06/2025 – 09h00 |
| **Gold (Fin du projet)** | 13/06/2025 – 17h00 |
|  |  |

**Nombre d’étudiants par groupe : 2 ou** 3

**Rendu pour la Gold :**

* tous les fichiers sources du jeu dans un dossier “**game/**”
* votre bibliothèque de mathématiques dans un dossier “**libmaths/**”
* votre bibliothèque graphique OpenGL dans un dossier “**libGL/**”
* vos assets (textures, mesh 3D, sons, etc…) dans un dossier “**assets/**”
* toutes les libs externes que vous utilisez dans un dossier “**external/**”
* vos documentations (diagrammes, screenshots…) dans un dossier “**documents**/”
* Un ou plusieurs **CMakeLists.txt** qui permettent de tout compiler sans problème ni warnings.

**Rendu attendu pour la présentation :**

* Dans le dossier documents :
  + 5 (cinq) screenshots qui montrent les moments importants du jeu
  + 1 (une) vidéo de gameplay de 30 secondes minimum qui montre l’essentiel de votre jeu.

## **Git :**

* **N’oubliez pas de tagger vos milestones ALPHA | BETA | GOLD**
* **N’oubliez pas votre .gitignore (vérifiez avant de push !!)**

# Sujet

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Le but de ce PFA est de vous faire travailler toutes les compétences que vous avez acquises cette année en C++, 3D et détection de collision.

Cette année, le PFA est un FPS en 3D réalisé avec OpenGL. Vous devez réaliser un prototype du jeu de puzzle sorti en 2018, [The Spectrum Retreat](https://en.wikipedia.org/wiki/The_Spectrum_Retreat) :

<https://www.youtube.com/watch?v=7O6KJJt1QyY>

Vous devez concevoir un niveau de jeu similaire à un “authentication challenge” de Spectrum Retreat. Le but de ce projet n’est pas de faire du level design : allez à l’essentiel et gardez ça simple. Un corridor et une salle d’énigme suffiront. Assurez-vous seulement que votre niveau démontre et utilise bien toutes les fonctionnalités qu’on vous demande. Un niveau où il suffit de marcher en ligne droite ne sera pas accepté.

# Règles

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Voici les contraintes à impérativement respecter pour ce projet :

## Organisation :

* Le projet dure trois semaines.
* Le groupe aura un **Team Lead** qui devra prendre les décisions techniques et suivre l’avancement du projet.
* Il y aura 3 milestones : **ALPHA**, **BETA** et **GOLD**.
* Chaque Lundi, l’équipe effectue un point d’avancement avec le ou la responsable.

## Technologies :

* Le projet doit être développé sur Windows en C++ avec l’API OpenGL 3.3 ou plus. (<https://glad.dav1d.de/> Profile : Core)
* **Vous devez réutiliser le projet OpenGL moderne pour la bibliothèque graphique.**
  + L’objectif n’est pas de repartir de zéro. Reprenez l'existant pour aller plus vite.
* Pour les mathématiques, vous **devez** utiliser votre propre lib de maths.
* Pour le fenêtrage, vous **devez** utiliser [**GLFW**](https://www.glfw.org/).
* Pour concevoir des interfaces utilisateur, vous **devez** utiliser [**ImGUI**](https://github.com/ocornut/imgui).
* Pour le chargement d’images, vous pouvez utiliser [**stb\_image**](https://github.com/nothings/stb/blob/master/stb_image.h).
* Pour charger des modèles 3D autres que des fichiers OBJ, vous pouvez utiliser [**Assimp**](https://github.com/assimp/assimp).
* Pour jouer du son, vous pouvez utiliser [**irrKlang**](https://www.ambiera.com/irrklang/).

## 

## Gameplay :

Vous développez un jeu et non une démo technique. Il faut donc que le gameplay soit précis et agréable.

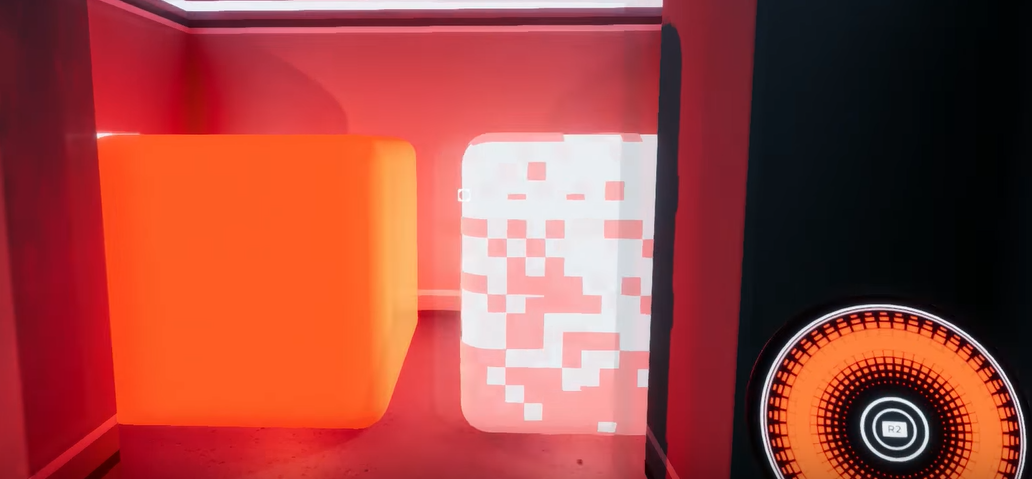
On souhaite retrouver au moins les éléments de jeu suivants :

* Un “téléphone” visible dans le coin de l'écran. (Ce téléphone peut être un simple cube) Le téléphone servira à déterminer quelle porte/mur le joueur peut traverser. Une porte/mur de la même couleur que votre téléphone peut etre traverse, une porte/mur de couleur différente ne doit pas l’etre. (On peut faire disparaître les portes lorsqu'elles sont de la même couleur que votre téléphone pour bien montrer que l’on peut les traverser).

*A picture containing electronics, computer, projector

Description automatically generated*

* Des objets interactifs (cube) de couleurs. Lorsque vous cliquez sur l’un de ces objets, la couleur de votre téléphone est échangée avec la couleur de l’objet visé. (documentez-vous sur l’object picking ou les raycasts)



* Évidemment, on ne doit pas pouvoir passer à travers les murs ou le plancher.
* De plus, une musique d'ambiance doit jouer en boucle.
* Finalement, on veut une zone qui démontre bien que l’on a réussi le niveau. (ex: un texte qui s’affiche à l'écran, une zone ou l’on se fait téléporter au niveau suivant)

A picture containing indoor, black, piano, opened

Description automatically generated

Les contrôles utiliseront une caméra en vue à la première personne classique (FPS), les touches ZQSD (ou WSAD en QWERTY) permettront de se déplacer et la souris permettra de tourner la tête dans toutes les directions. On doit être capable de sauter avec la touche Espace.

Pour l’illumination : un shader de Blinn-Phong suffira.

Vous devez démontrer au moins deux types de lumière différents.

Vous **devez** programmer **l’atténuation** de vos lumières afin qu’elles n’illuminent pas le niveau entier.

(Programmer un **Light Volume**, une box de collision qui active les lumières en dedans et désactive les lumières en dehors, serait une alternative, mais aurez-vous le temps ?)

On souhaite diversifier le gameplay en ajoutant les éléments suivants au jeu :

* Des objets interactifs qui se déplace selon un trajet prédéterminé (vous pouvez vous référez au niveau 4\_03 a titre d'exemple)
* On veut aussi ajouter des effets sonores différent en fonction des actions réalisées par le joueur (quand il utilise une cible ou quand il échange la couleur de son téléphone)

## Bonus:

On veut maintenant se rapprocher le plus possible “feeling” du jeu original :

* Des cibles interactives de couleur que lorsque vous cliquez dessus, si votre téléphone est de la même couleur que la cible, vous attire automatiquement vers elles. (Référez-vous au niveau du 3e étage.)
* Rendre les portes transparentes. Par exemple: 50% opaque lorsque le téléphone est de couleur different et 10% opaque lorque les portes ont la même couleur que le téléphone.
* Ajouter des portes/planchers à votre niveau (comme dans le niveau 2\_05)
* Ajouter des murs avec des trous à travers desquelles on peut interagir avec des objets placés de l’autre côté.
* Ajouter un menu accessible par la touche ESC qui possède les bouton suivant:
  + Recommencer
  + Quitter

## Super Bonus :

Si vous arrivez ici avec du temps restant, on vous propose les défis suivants :

* Ajouter sur les murs des plaques interactives qui permettent de changer la direction de la gravité. (Inspirez-vous des niveaux du 4e étage.)
* Ajouter des couloirs non linéaires qui vous permettent de vous déplacer rapidement pour atteindre des endroits que la physique suggérerait que vous n'atteignez pas par le trajet effectué. (Par exemple ce couloir: <https://youtu.be/MmjFzJwGDhM?t=197>, qui permet de monter au 2e en tournant simplement un coin gauche et un coin droit) (Attention à ce que vos téléportation pour réussir ce bonus soit bien invisible.)

## Performances :

Le jeu doit bien sûr ne pas souffrir de problèmes de performances (60 FPS constant).

# 

# Planning

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Documentation : Mardi 2 au Jeudi 4 Avril (2 Jours)

Vous devez réfléchir à toutes les fonctionnalités nécessaires pour ce projet, et construire des diagrammes UML à partir de cette analyse. Cela vous servira à anticiper les difficultés que vous allez rencontrer et vous répartir le travail.

Un document devra recenser :

* **les membres de l’équipe** ainsi que leurs **rôles attribués.**
  + Il n’y a pas de répartition des rôles obligatoire, mais voici une proposition :
    - Rôle **Core Gameplay** : s’occupe des déplacements du joueur, des interactions avec les objets (et les cibles) ainsi que du son.
    - Rôle **Rendu 3D** : s’occupe des shaders, lighting, effets graphiques. S’assure que le moteur graphique affiche correctement le jeu.
    - Rôle **Physique** : s’occupe de la solidité des murs et de la détection des collisions.
* **une analyse de votre moteur**
  + Identifier les problématiques actuelles : ce qui manque, ce qui ne marche pas…
  + **Lister les actions** nécessaires pour **régler** les problèmes du projet.
* **une analyse de votre projet**
  + L’architecture de votre niveau, les fonctionnalités que vous devez implémenter.
  + Utilisez une application d’UML comme Lucidchart ou Star UML 2 pour créer des diagrammes présentant votre architecture “idéale”
  + Lister les problématiques à résoudre découpées en tâches avec une estimation de temps accompagnée d’un “indice de confiance” (certain / incertain / très incertain).
* Les outils comme [Trello](https://trello.com/) ou [HackNPlan](https://hacknplan.com/) que vous utilisez pour faire le suivi de vos avancées. Il revient au Team Lead de votre équipe de s’assurer que tout le monde est à jour sur ses tâches et n’est pas bloqué plus que de raison.

Ce document devra être fourni dans le dossier “documents” sur le dépôt Git en **pdf.**

## 

## MILESTONE Alpha : Mercredi 12 Avril

Sont attendus pour l’alpha :

* l’essentiel d’un **character controller FPS** :
  + Se déplacer avec les touches **WASD**
  + Sauter avec la barre d’espace
  + Diriger la caméra avec la souris.
* Le jeu devrait comporter un terrain simple (comme un sol et des murs).
  + Ce n’est pas grave si les mesh ou les textures sont placeholder.
* La mécanique d’échange de couleur entre les objets interactif et le téléphone devrait être finie ou presque.
* Les murs devraient être solides. On ne doit pas pouvoir passer à travers.

## MILESTONE Beta (Code Freeze) : Mercredi 19 Avril

Sont attendus pour la beta :

* Le contrôleur doit être terminé (complètement debugué)
* L’architecture du niveau peut ne pas être finale. Par contre tous les bonus que vous voulez implémenter doivent être présents.
* On doit pouvoir traverser les portes qui ont la même couleur que votre téléphone.
* On doit pouvoir entendre la BGM (background music) qui tourne en boucle.
* Idéalement, à ce point toutes les fonctionnalités principales sont finies et il ne reste plus qu’à polish avant le rendu final.

## 

## MILESTONE Gold Master : 23 Avril

Votre projet devra contenir un maximum de features et avoir le moins de bugs possible.

Il est attendu un niveau complet qui démontre toutes les fonctionnalités requises par le sujet ainsi que tous les bonus que vous avez ajouté.

Toute tentative d’optimisation sera valorisée : frustum culling, occlusion culling…

La propreté de votre code et le respect de la norme de code de l'école seront aussi valorisés.

**Bon courage !!!**

# Liens utiles

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SON**

[**https://learnopengl.com/In-Practice/2D-Game/Audio**](https://learnopengl.com/In-Practice/2D-Game/Audio)

[**https://www.ambiera.com/irrklang/tutorial-helloworld.html**](https://www.ambiera.com/irrklang/tutorial-helloworld.html)

**TEXTE**

[**https://learnopengl.com/In-Practice/Text-Rendering**](https://learnopengl.com/In-Practice/Text-Rendering)

[**https://www.freetype.org/freetype2/docs/tutorial/index.html**](https://www.freetype.org/freetype2/docs/tutorial/index.html)

**MODELS**

[**https://learnopengl.com/Model-Loading/Assimp**](https://learnopengl.com/Model-Loading/Assimp)

**ASSETS**

Libre à vous de choisir vos assets à condition d’obtenir l’autorisation (ne piratez pas)