

P1RV – 1^{er} Projet de l'option RV 2025-2026

Jean-Marie NORMAND
Loën BOBAN
Olivier AUGEREAU

Présentation

- Projets Binôme : 32h **par étudiant**
- Envoi d'un rapport d'avancement **CHAQUE SEMAINE** le vendredi 18h dernier délai :
 - À l'encadrant et copie à moi (pas en double si je suis le seul encadrant ☺)
 - Contenu :
 - Avancement
 - Programme de la semaine à venir
 - Difficultés rencontrées
 - Planning mis à jour
- 06/10 : Présentation des sujets
- 08/10 soir : Retour de la répartition
- Contacts : loen.boban@ec-nantes.fr, olivier.augereau@ec-nantes.fr & jean-marie.normand@ec-nantes.fr

Rendus - Notation

- Rendus et échéances :
 - Cahier des Charges : 17/10 (après échange avec votre « client » !)
 - Rapport final : 09/01/2026 matin
 - Soutenance : 14/01 après-midi
- La note peut être individualisée
- Composantes
 - Travail : 50% (encadrant) – incluant le Cahier de charges
 - Soutenance : 20% (moyenne jury)
 - Rapport : 20% (encadrant)
 - Reporting : 10% (JMN)
- Absences ou retards
 - -1 par soutenance manquée (retard => soutenance manquée)
 - Rapport final : -1 par jour de retard

Sujet 1 : Editeur « Minecraft-like »

- Objectif :
 - Développer en C++/OpenGL un éditeur vous permettant de créer un monde à la MineCraft
 - Vous pouvez créer différents types de cubes et les combiner pour créer un monde
 - Pour gérer les textures vous aurez sûrement besoin de bibliothèque additionnelle comme p. ex. SOIL (<https://github.com/kbranigan/Simple-OpenGL-Image-Library/tree/master>)



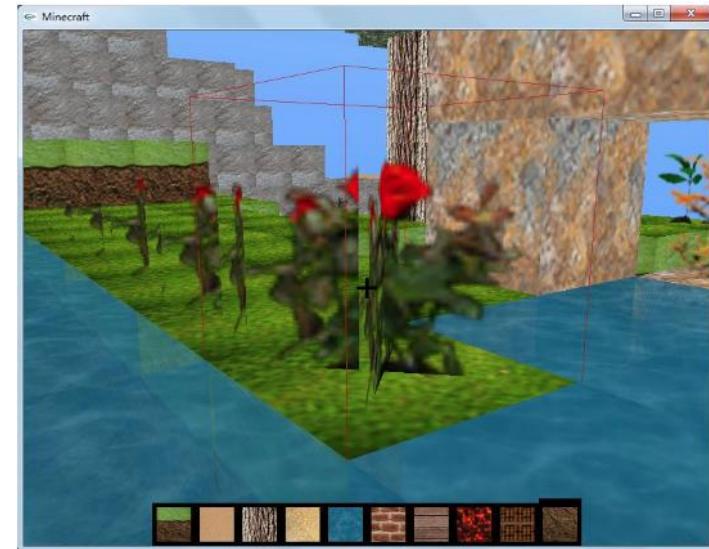
Sujet 2 : Jeu en OpenGL

- **Objectif :**

- Implémenter un jeu (objectifs et design de votre choix en accord avec l'encadrant) en C++/OpenGL
- Exemple: MineCraft-like, FPS-like, 3D Snake etc.

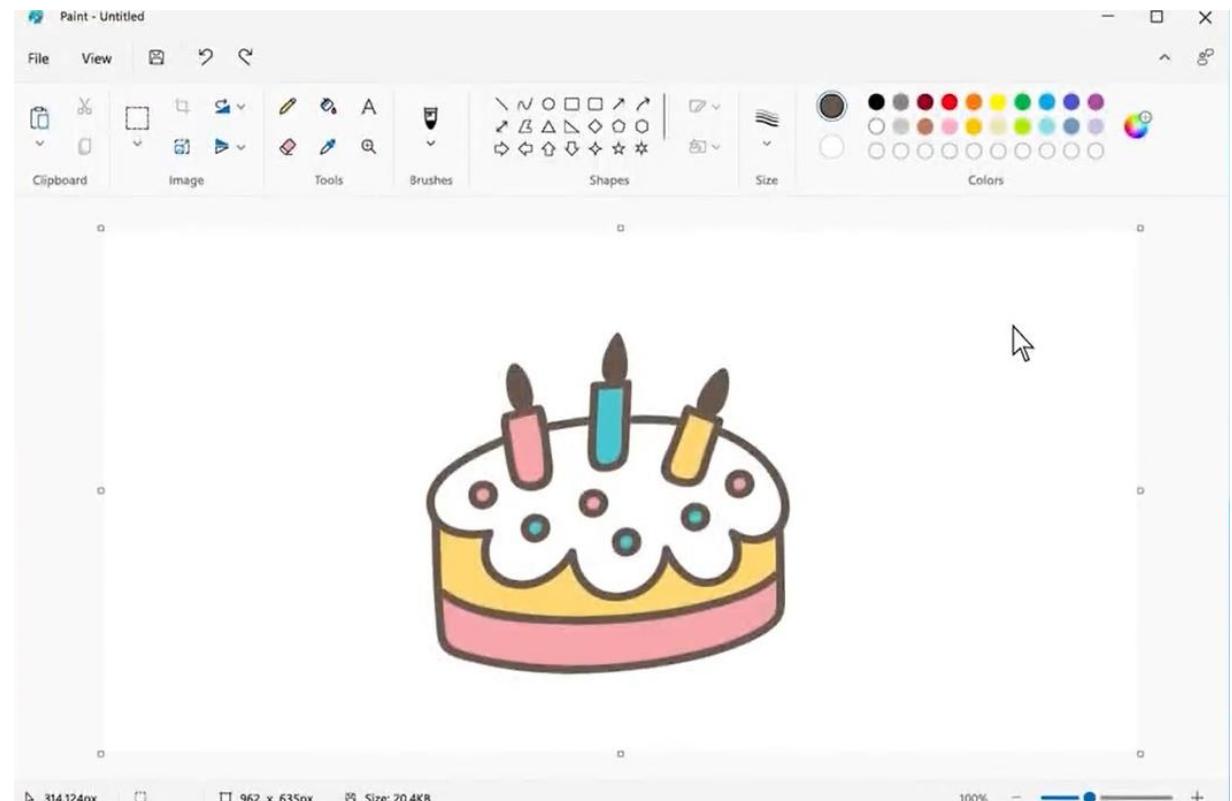
- **Principe :**

- Développer votre jeu en C++/OpenGL



Sujet 3 : Application « Paint-like »

- **Objectif :**
 - Implémenter un « Paint » simple en OpenGL
- **Principe :**
 - Pouvoir dessiner en 2D en OpenGL
 - Ajouter un certain nombre de brosses et fonctionnalités d'un éditeur d'images basiques



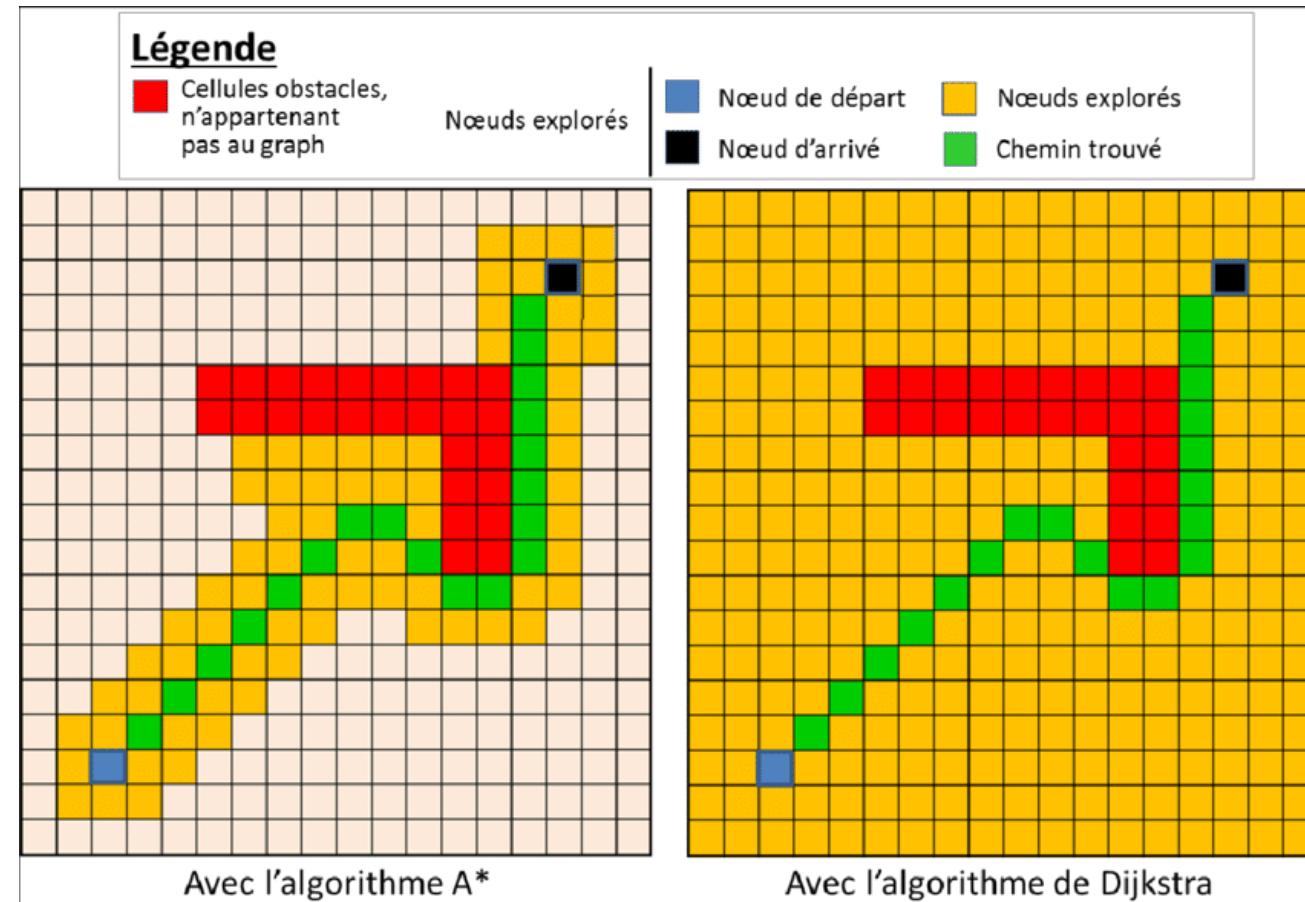
Sujet 4 : L'IA avant le Deep Learning

- **Objectif :**

Implémenter des algorithmes d'IA avant que la mode soit au DL.

- **Principe :**

- Choisir un algorithme après concertation avec votre encadrant
- Implémenter cet algorithme (p. ex. Montecarlo, A*, algo génétique, etc.)



Sujet 5 : Simulateur de jardinage

- **Objectif** : Développer avec Unity3D un simulateur simple de jardinage permettant de faire des pousser des plantes
- **Système** :
 - Une graine doit être plantée dans un pot (ou parcelle) et être arrosée
 - La plante passe par plusieurs stades avant d'être totalement développée
 - La plante peut être arrachée pour libérer l'emplacement



Sujet 6 : Jeu d'aventure

- **Objectif** : Développer avec Unity3D un petit jeu d'aventure simple
- **Système** :
 - Un PNJ a besoin d'aide ! Il donnera des instructions au fur et à mesure pour guider le joueur
 - Le joueur doit résoudre des puzzles
 - Ex : trouver des objets, lancer des objets, placer correctement des objets, rayons lasers à orienter, ...



Sujet 7 : Système de recette

- **Objectif** : Développer avec Unity3D un simulateur simple de cuisine basé sur des recettes à suivre
- **Système** :
 - Une recette est affichée et doit être suivie pour produire un résultat conforme
 - Les étapes à suivre peuvent être variées ex : mélanger des ingrédients, mixer, cuire ou laisser reposer (temps à respecter), décorer,...

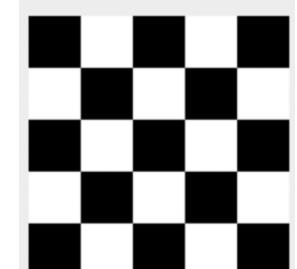
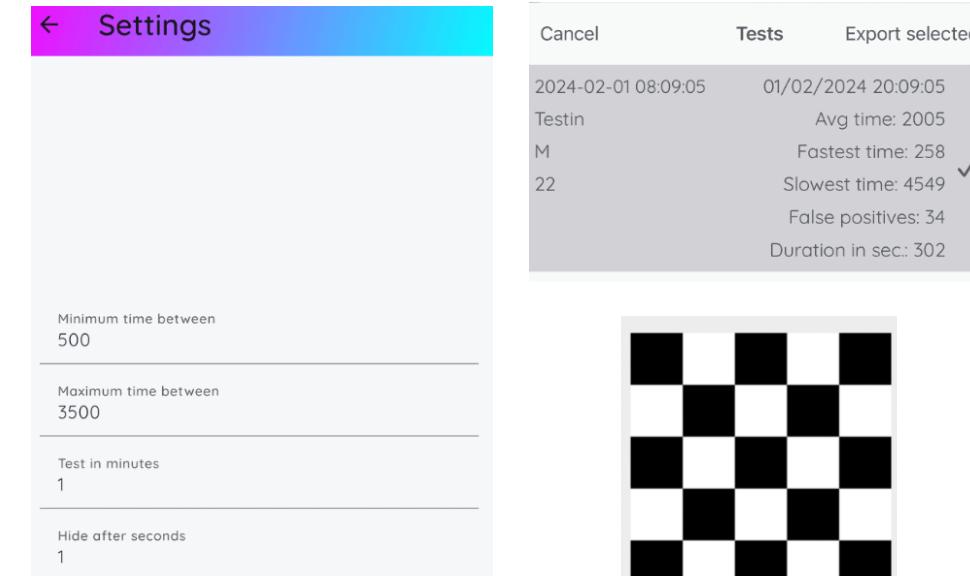
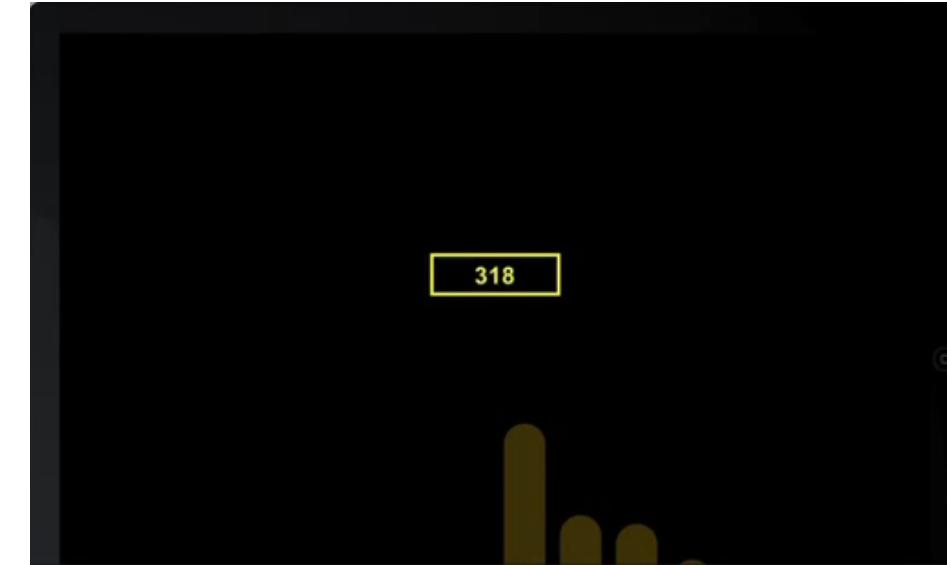


Sujet 8 : Test de vigilance psychomotrice (PVT)

Un PVT est un test d'attention qui mesure la vitesse de réponse à un stimuli visuel, demandant aux participants de réagir à des signaux présentés à des intervalles aléatoires.

Le score de PVT est utilisé pour étudier la fatigue, notamment par la NASA.

- L'objectif est de développer une interface graphique permettant de **paramétriser** et d'utiliser des PVT.
- Les données enregistrées doivent être **exportées** en fichier CSV.
- Implémenter une version dynamique



Sujet 9 : Générateur d'histoires C++ / SLM

Créer un programme pour **générer des histoires courtes** et simples pour enfants ou afin d'apprendre une langue étrangère

- Créer un programme permettant de générer un texte (prompt) qu'une IA utilisera ensuite pour générer une histoire. L'utilisateur peu choisir la langue, la longueur de l'histoire (nombre de phrases, nombre de pages), l'âge du lecteur, un thème, un genre, etc.
- Intégrer (soit en local, soit via API) un Small Language Model (SLM) ou un « petit » LLM type Mistral 7B ou Minstral 3B, pour générer l'histoire avec `llama.cpp` <https://github.com/ggml-org/llama.cpp>
- (optionnel) Intégrer Stable-Diffusion pour générer une illustration par page. Ajouter des paramètres pour contrôler le style des illustrations
<https://github.com/leejet/stable-diffusion.cpp>
- Créer une fonction d'export