

Projet C & Web

Présentation générale

CIR1 Brest/Caen/Nantes/Rennes - 2022

michael.aron@isen-ouest.yncrea.fr



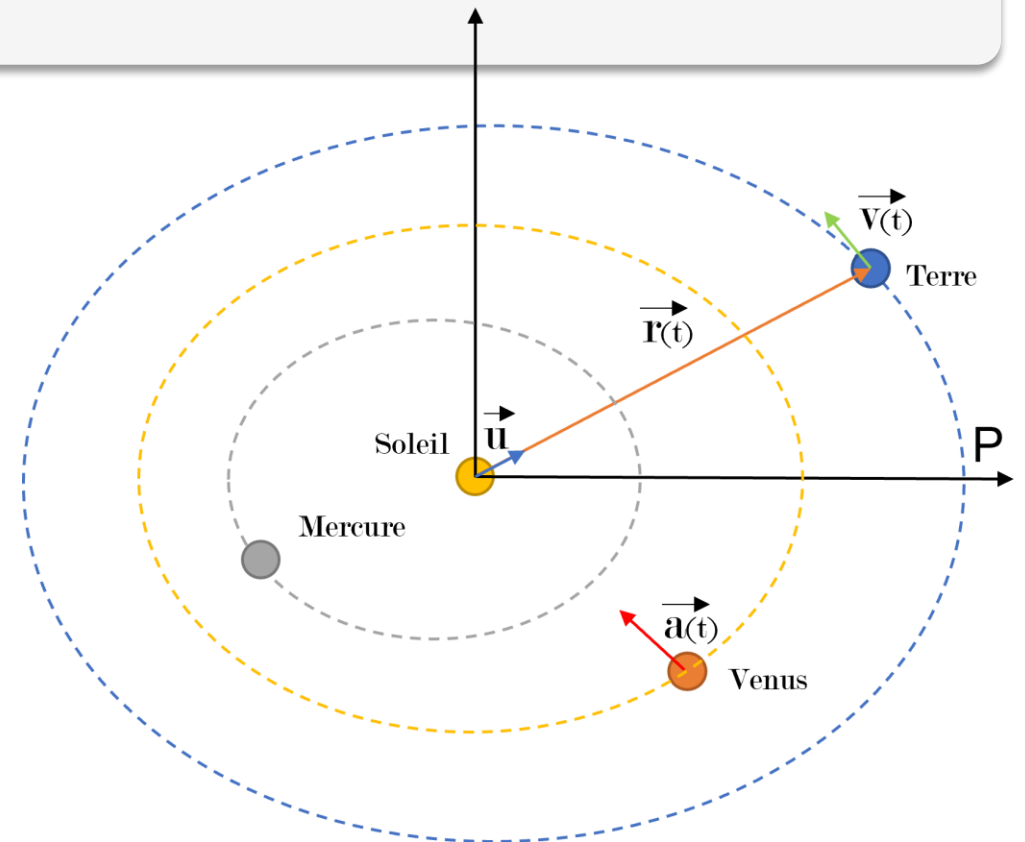
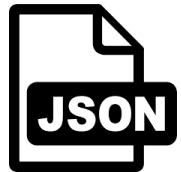
Objectif

Générateur de trajectoires de planètes

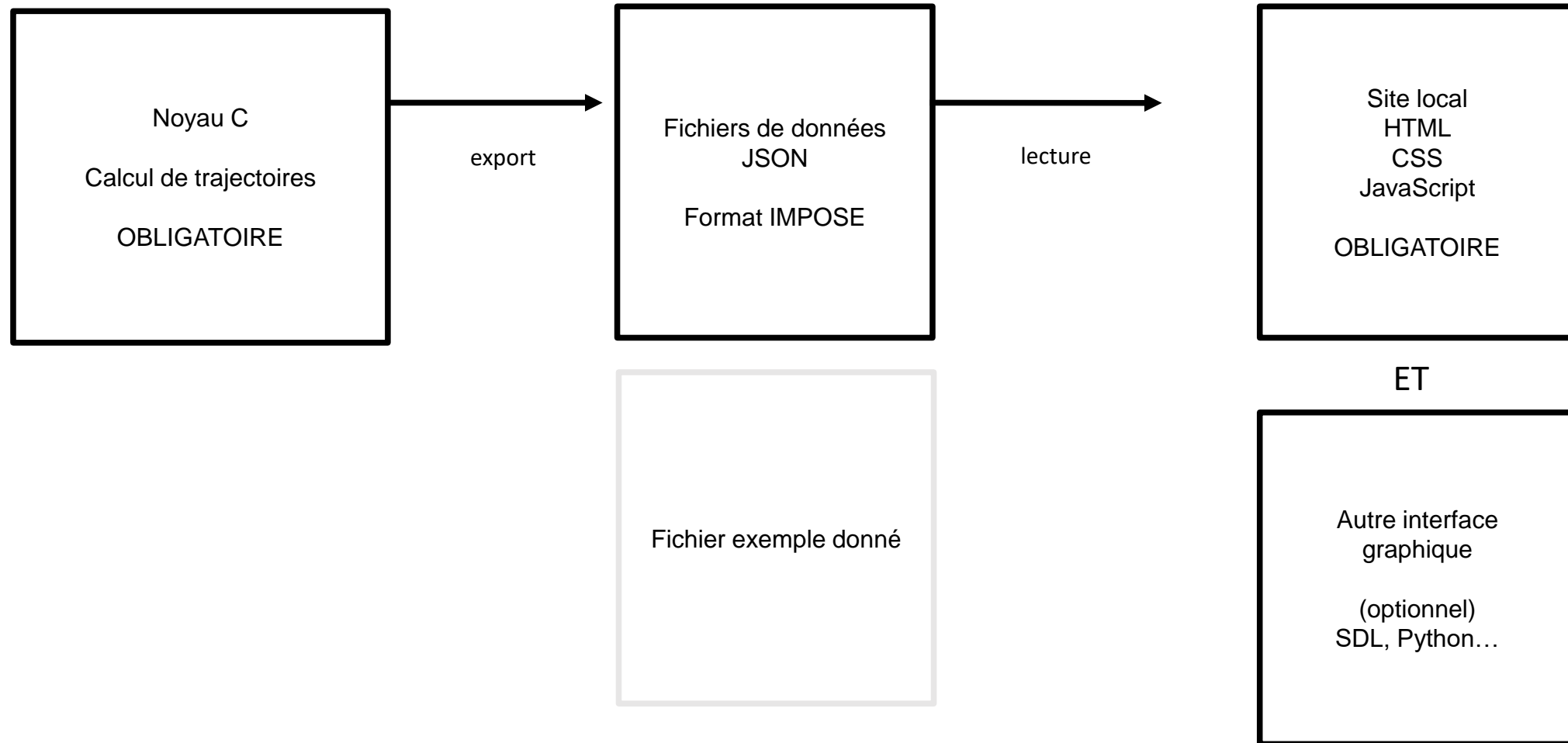
Calcul de trajectoires en C

Export et Lecture des données au format JSON

Affichage des données en HTML/CSS Javascript



Architecture globale du système



■ Fichier de données JSON :

- 1 fichier exemple est fourni (planète mercure)
- Un fichier JSON peut s'ouvrir et être édité avec un simple éditeur de texte
- 1 entrée par trajectoire
- 1 entrée = `nom_de_planete_methode` puis une liste de données : position, vitesse, temps.

■ Noyau de calcul en C :

- **Organiser les données dans des structures**
- **Organiser les calculs et les traitements dans des fonctions courtes**
- **Exporter les données dans des fichiers JSON avec le format imposé**
- Organiser le programme pour le rendre accessible à l'utilisateur

Architecture du système

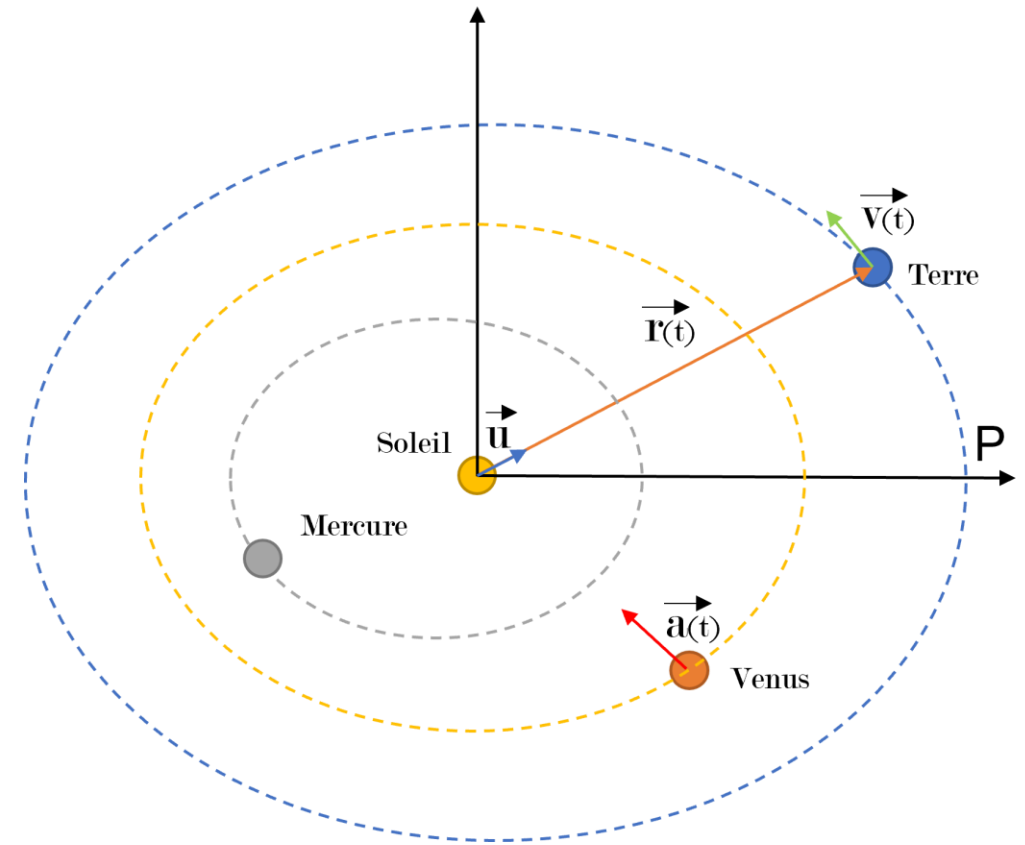
Noyau C – structures de données et traitements

■ Structures de données

- **Position** : vecteur x, y, z
 - z peut rester à 0 mais reste imposé dans le format d'échange
- **Vitesse** : vecteur x, y, z
 - z peut rester à 0 mais reste imposé dans le format d'échange
- **Temps** : entier
 - 0 en début de trajectoire puis incrémentation à chaque pas
- **Point** : triplet
 - position, vitesse, temps
- **Trajectoire**
 - Nom + une suite de points

■ Se référer au fichier d'exemple

```
{"earth-euler" : [  
  [[1.470000e+11, 0.000000e+00, 0.000e+00], [0.000000e+00, 3.028629e+04, 0.000e+00], 0],  
  [[1.470000e+11, 2.616735e+08, 0.000e+00], [-5.306168e+01, 3.028629e+04, 0.000e+00], 1],  
  ...  
]
```



- Le projet doit se composer d'une version minimale et d'extensions
- **Projet minimal**
 - Le noyau C permettant de générer une trajectoire de planète avec la méthode de résolution la plus simple : la méthode d'Euler
 - Le noyau C doit exporter cette trajectoire dans un fichier au format imposé en JSON
 - Affichage de la trajectoire dans une interface graphique en HTML/JS
 - Le projet minimal ne se suffit pas à lui seul.
- **Extensions possibles**
 - Extension mathématique : améliorer le calcul de la trajectoire
 - Avec la méthode d'Euler asymétrique
 - Avec la méthode de Runge-Kutta (comparer les méthodes)
 - Recherche sur les constantes des planètes
 - Ajout de fonctionnalités à la page HTML

■ Autres extensions possibles

- Ajouter plusieurs planètes au système (recommandé)
- Modéliser toutes les interactions du système (La Terre et Mars s'attirent entre-elles) ;
- Modéliser des trajectoires plus réalistes : ajout d'une troisième dimension, amélioration de l'orientation des ellipses ;
- Rechercher et lire des données de trajectoires de corps célestes trouvées sur des sites officiels (ESA, NASA) puis les importer pour les retranscrire dans le format d'échange du projet ;
- Ajout de la troisième dimension (les trajectoires ne sont pas toutes sur le même plan).

■ Challenge (attention, difficile)

- Réussir à stabiliser la Lune autour de la Terre (challenge);
- Modéliser la ceinture d'astéroïdes ;
- Prévoir une éclipse lunaire, un alignement de planète;
- Modéliser la trajectoire d'une comète, par exemple la comète de Halley (expliquer la forme de la trajectoire).

Extension graphique (obligatoire)

Page HTML de base :

- Présentation du projet en quelques mots
- Intégration d'un canvas pour visualiser vos orbites (orbites complètes / animations)
- Autres fonctionnalités (boutons de contrôle, autre canvas pour autre visualisation, changement de référentiel...)

Javascript

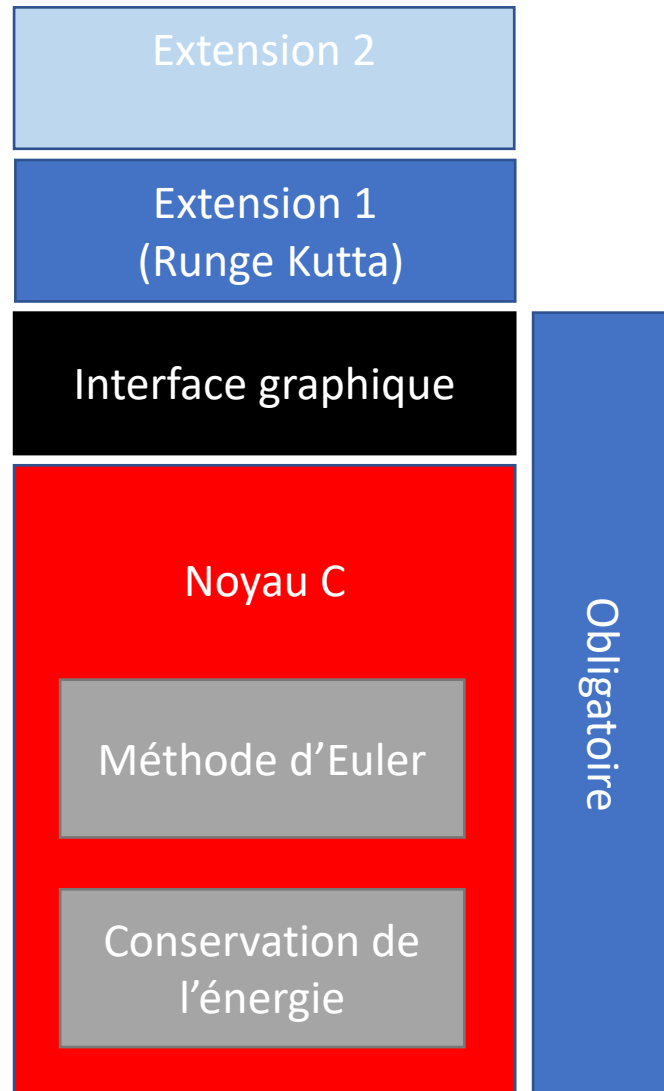
- Utilisation de la librairie p5.js : <https://p5js.org/>
- Les références de la librairie sont présentes à la page <https://p5js.org/reference/>

■ Documentez-vous !

- Le sujet est concret, il est en rapport avec la physique (l'astronomie) .
- Cherchez par vous-même les informations sur internet :
 - Quelle est la distance de la terre au soleil ?
 - Quelle est la masse de la planète Terre ?
 - A vous de vous poser les bonnes questions et d'y répondre.
 - Culture générale : renseignez-vous sur les personnages clefs (Kepler, Copernic, Galilée, Newton, Euler ...)
 - Des questions générales sur la physique et son histoire pourront être posées.

- **Le projet doit fonctionner**
 - Ne pas être trop ambitieux
 - Bien s'organiser (en groupe)
 - Être capable de justifier les choix initiaux
 - Savoir expliquer les limites du projet.
- **Le code doit être lisible**
 - Séparation en fichiers (.h / .c)
 - Plusieurs fonctions courtes plutôt qu'une seule fonction incompréhensible
 - Code bien commenté et présenté
 - Pas de copie de code ! (internet, collègues)

Critères d'évaluation



- **Code**
 - Code bien commenté et organisé.
 - **Fichier JSON (projet fonctionnel)**
 - Doit respecter le format imposé
 - **Présentation**
 - Présentation de 3-4 minutes
 - Bilan / cahier des charges
 - **Questions**
 - Questions individuelles
 - Analyse des résultats
- La note peut varier dans le binôme !**

■ Phase 1 : étude préliminaire

- Présentation du projet
- Lecture et compréhension du sujet
- Réflexions sur la planification et la répartition des tâches au sein du groupe (détail des fonctionnalités principales de votre application)
- Rédaction du 1^{er} livrable

Rendu sur moodle

■ Phase 2 : implémentation, tests et rendu final

- Implémentation des fonctionnalités principales
- **Tests et validation**
- Implémentation des fonctionnalités additionnelles
- **Tests et validation**
- Préparation du livrable final et de la recette (présentation)

Rendu sur moodle

■ Cahier des charges / planification

- Lecture et compréhension du cahier des charges
- Identification des tâches
- Gestion des dépendances
- Répartition des tâches dans le temps et par personne
- Identification des points difficiles et/ou bloquants dans l'implémentation

	J1	J1	J2	J2	J3	J3	J7
Eleve1		A ?	B ?						
Eleve2		A ?	B ?						

- Identification des tâches
 - Tâche A : structure planète, point, vecteur et trajectoire
 - Tâche B : écrire les résultats au format json
- Gestion des dépendances
 - B dépend de A
- Répartition des tâches dans le temps
 - Traiter A avant B, temps nécessaire...
- Répartition des tâches par personnes
 - Elève 1 fait A, Elève 2 fait B (attente ...)
 - Elève 1 fait A puis B (seul)
 - Elève 1 et 2 travaillent en binôme sur A puis B (en échangeant)

Une fois la 1^{ère} planification terminée

- **Cahier des charges / planification**
 - Prévoir les ajouts au projet
 - Décrire et décomposer en sous-tâches : spécifiques, mesurables, atteignables, réalistes, temporelles
 - Ajouter ces tâches au planning
 - Ne pas oublier les tâches suivantes :
 - Intégration (cf slide suivante)
 - Tests et validation
 - Préparation du rendu
 - Préparation de la recette (présentation sur BBB)
 - Relecture du code
 - Création et dépôt de l'archive (éviter de s'y prendre à la dernière minute...)

Phase 2 : implémentation

- Implémenter le code
- Travail à 2 : relire et commenter le code de l'autre
- Communiquer et échanger le plus possible avec son binôme : attention aux incompréhensions qui peuvent engendrer des catastrophes (phase "d'intégration")
- Sauvegarder régulièrement son code pour éviter la réflexion "ce matin, ça marchait, mais là, plus rien ne fonctionne" !



Organisation du projet (30h)

Jeudi 23/06 au Mercredi 29/06 2021

Jeudi	Vendredi	WE	WE
Présentation	Étude Rendu 12h		
Étude	Implémentation		

- **horaire : 9h-12h et 13h30-16h30**
- **lieu : salle 309**

Lundi	Mardi	Mercredi
Implémentation	Implémentation / Relecture	Finalisation du code Envoi sur Moodle à 12h
Implémentation	Implémentation / Relecture	Présentation

- Phase 1 : Etude préliminaire

→ vendredi 24/06, 12h00 :

- Au format pdf, un document regroupant votre planning prévisionnel détaillé, vos différents choix d'implémentation (structures de données, utilisation des menus/scripts...).

- Phase 2 : Rendu final

→ mercredi 29/06, 12h00 sur Moodle :

- Rendu du code source : code organisé et commenté + 1 fichier `readme.md` (Markdown) contenant les procédures de compilation et d'exécution de votre programme
- Rendu du support de présentation (pdf ou ppt)

1 seul rendu par groupe

Attention

Archives sur moodle : format zip, nommage : `projetCIR1_groupeX.zip`

X représente votre numéro de groupe. Tout retard sera sanctionné (l'heure du réseau fait foi). Les fichiers au mauvais format ou avec mauvais nommage seront pénalisés.

- **Modalités**
 - 6 minutes de présentation (2x3min) + 4 minutes de question
 - Doit utiliser des supports (pdf ou ppt)
 - Les 2 membres du groupe **doivent** prendre la parole
- **Contenu attendu**
 - Rappel du contexte (besoin, organisation, etc.)
 - Vos grands choix de conception/structures de données, avec les arguments clés
 - Ce qui fonctionne et ne fonctionne pas
 - Captures d'écran
 - Bilan technique (principales difficultés rencontrées, etc.). Pas de considérations personnelles et subjectives.
- **Conseils :**
 - Pas de diapo vide
 - Évitez de mettre du code
 - Privilégiez les schémas, en ne gardant que l'information pertinente
 - Tout ce qui est présent dans une diapo doit être utile (et présenté). Sinon, épurez !

- **Barème indicatif :**
 - Phase 1 : étude préliminaire : 25%
 - Rendu de code final : 50%
 - Présentation : 25%

- **Remarques :**
 - Malus possible sur des membres du groupe si l'investissement est jugé trop faible
 - Possibilité d'être interrogé durant le projet de façon individuelle
 - Possibilité d'interroger un membre du groupe sur une question de physique ou de math (en lien avec le projet)
 - Plagiat sévèrement sanctionné pour TOUS les membres du/des groupe(s)
 - Le code sera évalué uniquement d'un point de vue fonctionnel. Toutefois, des mauvaises pratiques significatives seront sanctionnées par des malus.

- Le ou les encadrants ne font pas le code à votre place. **Ils vous aident** éventuellement à **corriger vos bugs** et à trouver vos erreurs.
- **Votre code est unique.** Copier/coller le code source d'un autre groupe sans le citer vous expose, ainsi que vos camarades, à une sanction dans la note du projet.
- **Sauvegardez régulièrement** votre travail. Dès que vous avez une version opérationnelle, mettez là de côté au cas où...
- **Ne misez pas tout sur le mercredi** pour terminer le travail (dernier jour réservé aux finitions et commentaires).
- **Validez étape par étape**, découpez votre code en fonctions, testez le plus souvent possible. Réfléchissez à des solutions partielles pour pouvoir avancer.
- Ce projet pour être mené à bien nécessite **un investissement personnel et de groupe**, et de nombreux tests.

- Sur moodle, tous les supports, liens, documents sont présents :

<https://web.isen-ouest.fr/moodle/course/view.php?id=779>

FAQ (Foire Aux Questions) à utiliser pour poser vos questions ou le public chat de la salle de projet BBB (salle CIR1R)

Clé pour se connecter à la salle : CIRCARE1

?

Des questions ?