
ARCHITECTURE DU PROJET

Groupe 2

AL BAKALI Roudaina
EL MOUDNI Zakaria
KOC Damian
ROBALINO ROBLES Jeremy
YAZBECK John



05 MAI 2025
HELB IODA B2 2024-2025
Cours : Gestion de projets 1

Architecture du Projet

Table des matières

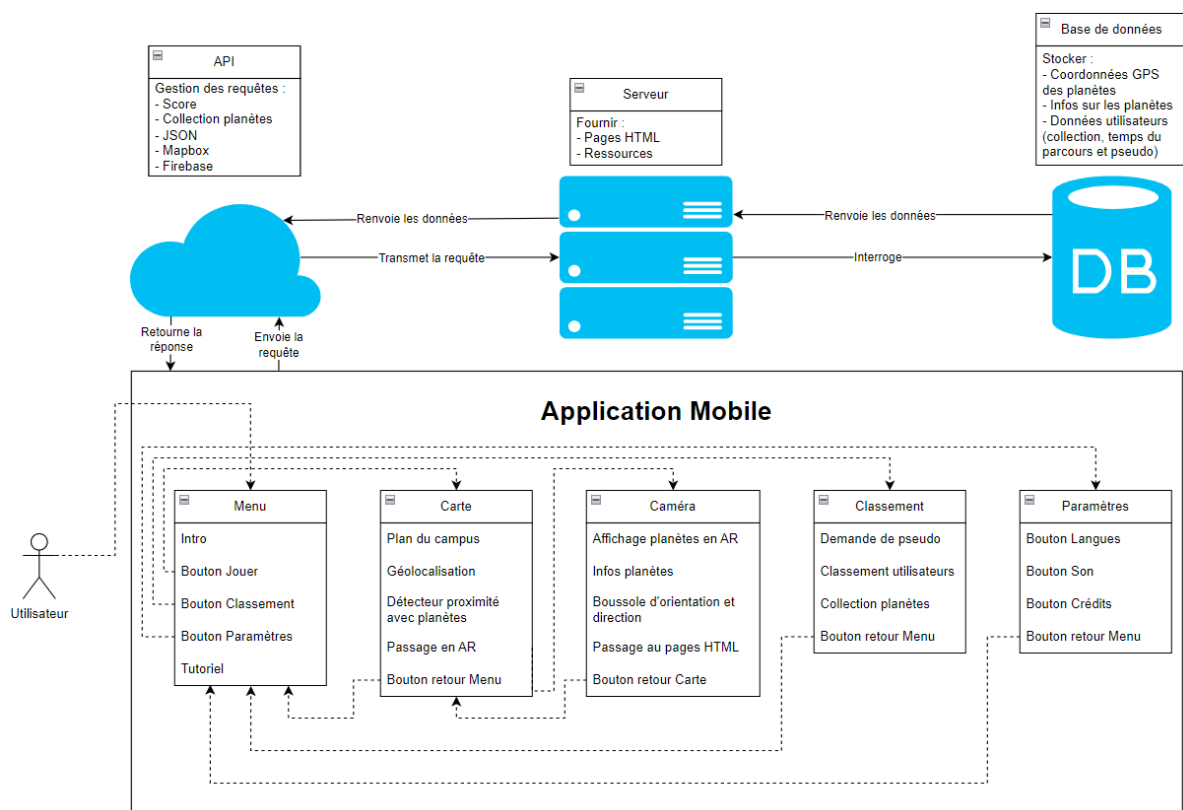
1	Introduction.....	2
2	Schéma HLD.....	2
3	Informations contextuelles sur la solution.....	3
3.1	Récupération de planètes à proximité (Scène carte)	3
3.2	Affichage d'une planète en AR (Scène caméra)	3
3.3	Consultation des infos planètes (Pages WEB)	3
3.4	Classement des utilisateurs (Scène classement).....	3
4	Technologies.....	4
5	Framework et libraires	4
6	Design applicatif	5
6.1	Structure de navigation principale	7
6.2	Fonctions principales accessibles depuis le menu	7
6.3	Fonctionnalités immersives.....	7
6.4	Contenu externe.....	8
7	Processus de déploiement	8
8	Besoin matériel.....	8

Architecture du Projet

1 Introduction

Ce document présente l'architecture et les choix techniques réalisés dans le cadre du développement d'une application mobile en réalité augmentée. L'objectif est d'offrir une expérience immersive d'observation de planètes, avec un système de géolocalisation, une visualisation AR et un classement des utilisateurs.

2 Schéma HLD



L'architecture de l'application repose sur plusieurs modules interconnectés :

- Carte : localisation de l'utilisateur, affichage des planètes via Mapbox.
- Caméra (AR) : instanciation de modèles 3D de planètes selon la position GPS.
- Pages Web : consultation de détails via WebView.
- Scène classement : affichage d'un score basé sur la collecte et le temps.

Le système communique via :

- Fichier JSON local pour les données de planètes,
- Appels API pour les classements,
- Navigation entre scènes pour changer de contextes utilisateur.

3 Informations contextuelles sur la solution

Le projet a été conçu dans un cadre éducatif à la HELB IODA avec une demande de clients, pour démontrer l'intégration entre géolocalisation, AR et gamification. Il repose sur une approche modulaire pour faciliter l'ajout de contenus (planètes) et l'évolution de la solution (ex. multijoueur ou base de données distante).

Flux de données :

3.1 Récupération de planètes à proximité (Scène carte)

1. L'application obtient la position GPS de l'utilisateur.
2. Elle charge le fichier planets.json.
3. Elle filtre les planètes proches dans un rayon proche.
4. Les planètes filtrées sont affichées.
5. L'utilisateur sélectionne une planète sur la carte. → Passage point 2.2.

3.2 Affichage d'une planète en AR (Scène caméra)

1. Unity charge le prefab de la planète et ses coordonnées GPS.
2. Le prefab est instancié dans la scène AR à l'emplacement physique détecté.
3. Affichage d'un panel d'infos sur la planète, récupérés de la base de données.
4. L'utilisateur clic sur un boutons pour plus d'informations. → Passage point 2.3.

3.3 Consultation des infos planètes (Pages WEB)

1. L'application ouvre une WebView avec une page HTML.
2. La page est soit locale, soit servie par le serveur web.
3. L'utilisateur lit les infos supplémentaires sur la planète.

3.4 Classement des utilisateurs (Scène classement)

1. L'utilisateur collecte une planète. → Événement déclenché au point 2.2.
2. L'application envoie une requête API au serveur (POST /score).
3. Le serveur met à jour la base de données.
4. À l'ouverture du classement, l'application appelle l'API (GET /classement) pour afficher les scores selon la collection des planètes et du meilleur temps.

4 Technologies

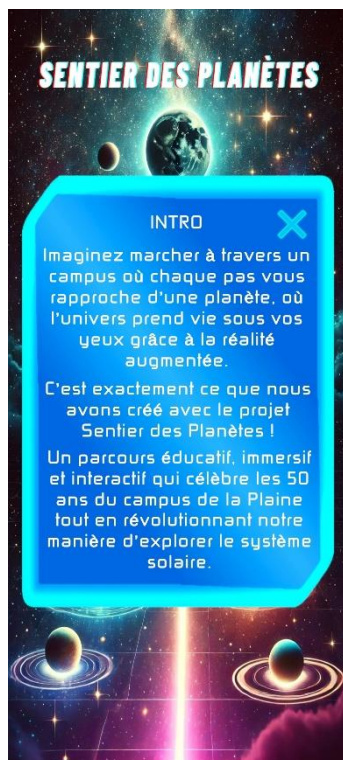
Technologies	Domaine / Usage
Unity 2022.3.58f1 et C#	Développement mobile
AR Foundation avec ARCore (Android)	Réalité augmentée
Mapbox SDK pour Unity	Carte interactive
HTML5, CSS	Pages web
JSON local et Firebase (Classement)	Base de données
Unity Version Control	Contrôle de version
Visual Studio et Visual Studio Code	Éditeurs

5 Framework et libraires

Framework / Librairie	Usage
Unity Engine	Moteur principal du projet
AR Foundation	Abstraction pour l'AR, compatible ARCore
Mapbox SDK for Unity	Affichage de carte interactive et des marqueurs
UnityWebRequest	Gestion des appels réseau pour le classement
TextMeshPro	Rendu textuel avancé
WebView	Affichage des pages HTML

6 Design applicatif

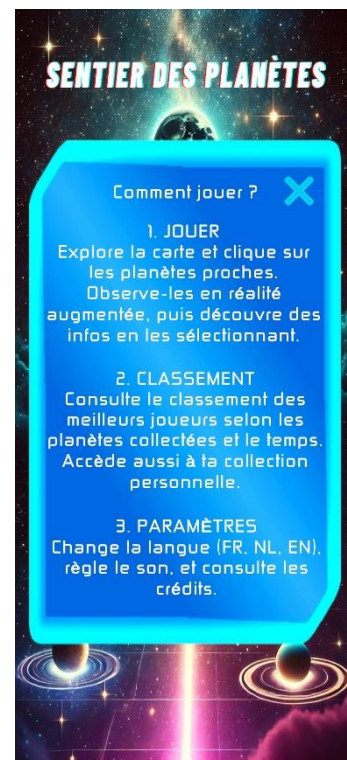
Introduction



Menu



Tutoriel



Demande Pseudo



Classement



Paramètres



Changement de langues



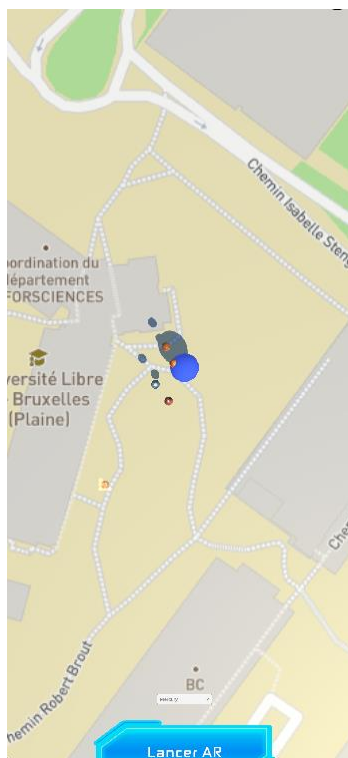
Gestion du son



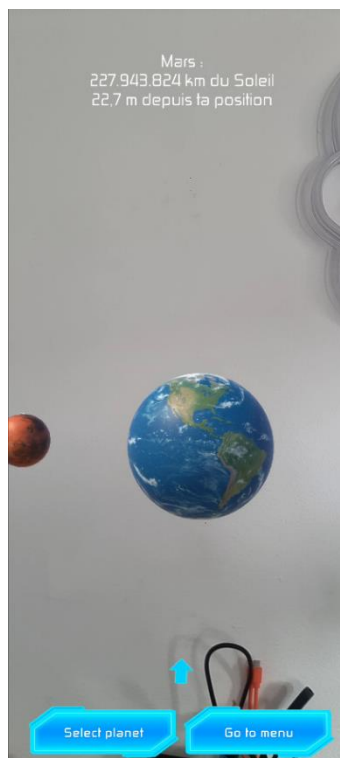
Affichage des crédits



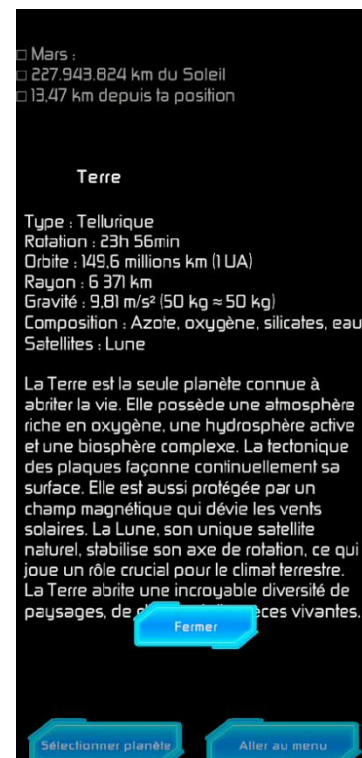
Carte interactive

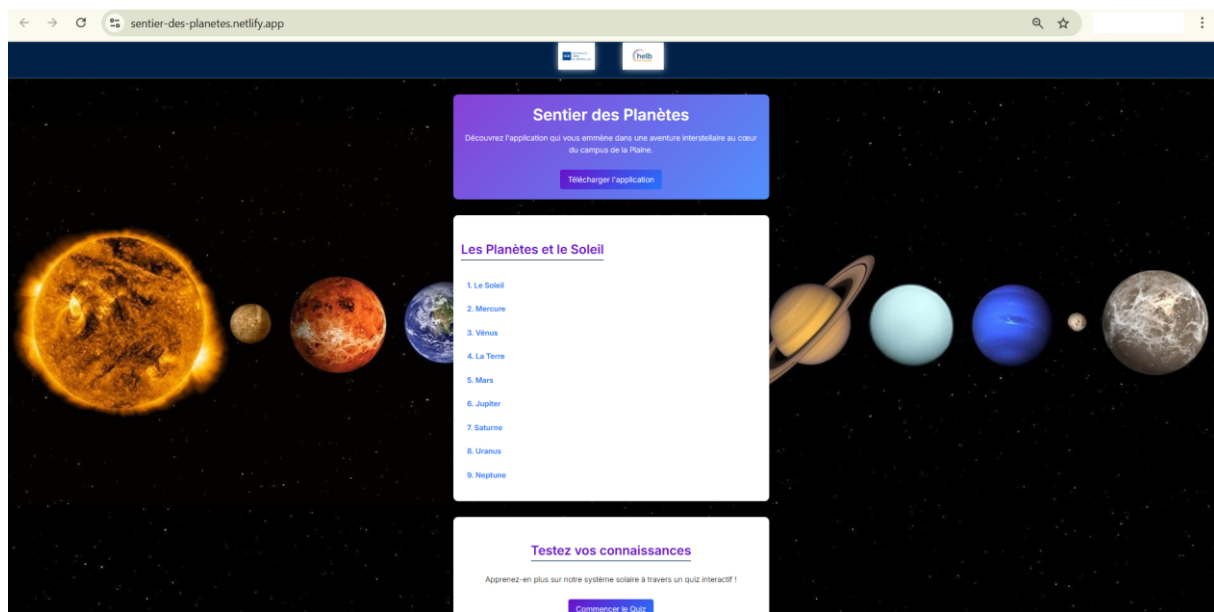


Caméra + AR



Informations sur planètes





L'application est organisée en plusieurs modules présentés sous forme de scènes Unity. Chaque scène ou menu remplit une fonction précise dans l'expérience utilisateur.

6.1 Structure de navigation principale

- **Introduction** : écran d'accueil présentant brièvement l'application.
- **Menu principal** : point central d'accès vers les autres fonctionnalités.
- **Tutoriel** : guide interactif pour apprendre à utiliser l'application.

6.2 Fonctions principales accessibles depuis le menu

- **Demande de pseudo** : l'utilisateur saisit un nom pour être identifié dans le classement.
- **Classement** : affichage du score des joueurs en fonction de la collecte de planètes et du temps.
- **Paramètres** :
 - **Changement de langue** : sélection de la langue de l'interface (FR/NL/EN).
 - **Gestion du son** : activation/désactivation des effets sonores.
 - **Affichage des crédits** : liste des développeurs.

6.3 Fonctionnalités immersives

- **Carte interactive** : vue Mapbox montrant la position GPS de l'utilisateur et les planètes à proximité.

- **Caméra + AR** : module de réalité augmentée permettant d'afficher les planètes dans l'environnement réel.
- **Infos planètes** : accès à des informations enrichies sur les planètes via un panneau intégré.

6.4 Contenu externe

- **Page Web** : affichée via WebView, cette page fournit du contenu supplémentaire sur les planètes, en HTML/CSS, hébergé en ligne.

7 Processus de déploiement

1. Ouvrir le projet via Unity Hub en version 2022.3.58f1.
2. Synchroniser les sources via Unity Version Control.
3. Faire une configuration pour ajouter une fonctionnalité.
4. Builder l'application pour Android.
5. Tester sur un smartphone réel (AR & GPS requis).

8 Besoin matériel

Matériel	Usage
Smartphone	Exécution de l'application en AR
Caméra arrière + AR	Affichage des objets en AR
GPS	Localisation utilisateur
Capteurs	Orientation et précision AR
Ordinateur	Développement sous Unity