



EpiAirConsole

**Cahier des charges**

Projet Professionnalisant

# Description du document :

Titre	Cahier des charges
Objet	Cahier des charges du projet EpiAirConsole
Auteurs	Théo SERRANO, Jan KONIOR
Promotion	2026
Dernière mise à jour	26 Octobre 2025
Version	1.0

# Tableau des révisions :

Date	Version	Auteur(s)	Section(s)	Commentaires
26/10/2025	1.0	L'équipe	Le document	Création du document

Description du projet	3
Liste des livrables	3
Tâches	4
Front Web	4
Front mobile	6
Back End	8
Jeux	10
CI/CD	11

## Description du projet :

EpiAirConsole est un projet qui a pour but de reproduire au mieux les fonctionnalités du site [AirConsole](#). Celui-ci consiste en la création d'une application web permettant de jouer à des jeux faits sur Godot directement sur son navigateur. La particularité principale du projet cependant, est la fonctionnalité d'utiliser des téléphones comme manettes pour un jeu en cours. EpiAirConsole a pour but de simuler un projet avec des attentes professionnelles, et sera donc accompagné d'un cadre professionnel (Outil de gestion de projets, tests automatisés, etc...).

## Liste des livrables :

### Front Web :

Le front tel qu'il sera vu par l'ordinateur. Celui-ci affiche le jeu en question, et représente le rôle de "console de jeux" pour le projet.

### Front mobile :

Le front sur lequel doivent arriver les appareils mobiles pour se connecter à la "console". Ceux-ci se connectent au jeu en cours pour servir de "manettes".

### Back-End :

Le back-end contient les jeux et informations des jeux en question, comme leur nom, leur image de vitrine, etc... Il contient également les données relatives aux comptes utilisateurs.

### Jeux :

Les jeux sont des jeux développés sous Godot engine qui sont ensuite exportés pour fonctionner sur le front web en WebGL.

### CI/CD :

La gestion du projet sera faite via Github Project et Github Action.

### Documentation :

Une documentation sera mise à jour au cours du développement du projet et sera disponible sur le Github.

# Tâches :

## Front Web

### Techno utilisé:

- HTML5 + React.js + CSS + balise Canvas/WebGL (Export d'un mini jeu Godot).

### 1. Espace Utilisateur

#### Description :

Gestion des comptes utilisateurs

#### Definition of done :

- Page de connexion / Création de compte
- Profil avec statistiques (Nombre de parties jouées / High score)

Temps de travail : 1j

### 2. Landing page

#### Description :

Page de présentation du site

#### Definition of done :

- Présentation des fonctionnalités.
- Mini vitrine des jeux disponibles en carrousel.

Temps de travail : 1j

### 3. Page de jeu

Description :

List des différents jeux proposés sur la plateforme

Definition of done :

- Différents jeux jouables avec une image de couverture
- Lors de la sélection d'un jeu, un code apparaît pour la connexion au téléphone
- Une fois la connexion établie, seul l'UX du jeu se lance sur le web

Temps de travail : 2j

Front mobile :

Technos utilisées:

- HTML5 + React.js + Socket.IO client.

## 1. Page d'accueil

Description :

Page de présentation du site

Definition of done :

- La page où arrivent les téléphones supportés
- Présentent le site et les fonctionnalités
- Mène à l'appairage

Temps de travail : 0.5j

## 2. Appairage du téléphone

Description :

Le téléphone se connecte à un jeu proche

Definition of done :

- Un pseudonyme est demandé au joueur
- Le téléphone cherche les PCs connectés sur le même réseau WiFi
- Un code de vérification est demandé
- Quand le code du PC correspondant est rentré, la manette est connectée

Temps de travail : 2j

### 3. Page de manette

#### Description :

Interface de manette pour jouer au jeu en cours

#### Definition of done :

- Une interface avec des touches distinctes
- Appuyer sur ces touches influe sur les actions du joueur correspondant au pseudonyme entré
- Si le joueur quitte la page pendant trop longtemps, il est déconnecté

Temps de travail : 2j

## Back End:

### Technos utilisées:

- Node.js + Express + WebSocket ([Socket.IO](https://socket.io/)).
- DB: MongoDB.

#### 1. Compte utilisateur

Description :

Gestion des données utilisateur

Definition of done :

- Sauvegarde des comptes
- Sécurité des données
- Sauvegarde des informations (score/partie/ext..)

Temps de travail :

#### 2. Sauvegarde des jeux

Description :

Stocker les jeux et informations de jeux

Definition of done :

- Avoir les jeux en DB
- Avoir les informations de jeux en DB (Image vitrine/nom)
- Faire la sauvegarde des parties en cours non fini (Si réalisable dans la DL)

Temps de travail :



### 3. Connexion

Description :

Connexion et synchronisation entre les appareils

Definition of done :

- Connexion entre appareil après l'entrée du code sur téléphone.
- Synchronisation entre les mouvements du téléphone et les players en jeu.
- Synchronisation des éléments (attaque, objets, ext...)

Temps de travail :

Jeux :

Techno utilisée:

- Godot engine.

## 1. Création de jeux

Description :

Création de jeux sous Godot

Definition of done :

- Possibilité de créer et exporter des jeux peu gourmands.
- Différents mini-jeux (~5 - 10min de game loop).
- 1 mini-jeu au moins.
- Cette tâche sera scale en fonction de l'avancement du projet, et potentiellement divisée en plusieurs tâches

Temps de travail : 4j

CI/CD :

## 1. Github Project

Description :

Organisation de l'équipe

Definition of done :

- Définition des rôles aux membres de l'équipe.
- Création d'issues pour un suivi des tâches.
- Ajout de Milestone sur les jours importants (FU/Défense).
- Création de branches pour chaque issue.
- Date de début et de fin pour chaque issue.
- Des tags pour la répartition des tâches (front/back/mobile/jeu).

Temps de travail :

## 2. Github Action

Description :

Webhook discord

Definition of done :

- Ajout d'un webhook discord pour le suivi des commits en temps réel.

Temps de travail :