# Projet IHM (L3) : Mini-jeu web protégé par session (PHP)

*Premier projet court pour évaluer le niveau, travailler la démarche IHM, et introduire des réflexes sécurité + exploitation.*

## 1. Objectif et contexte

Vous réalisez une petite application web en PHP qui donne accès à un mini-jeu (Snake, Tetris-lite, Flappy-like, etc.) uniquement après authentification par un mot de passe unique. L'objectif principal est de pratiquer une démarche IHM complète : recueil du besoin, maquettage, conception des interactions, puis mise en œuvre (programmation événementielle côté interface).

Même si le projet reste volontairement simple (pas de base de données, pas de comptes), vous devez démontrer en soutenance des réflexes professionnels : gestion correcte de session/cookies, protection minimale des pages, traces utiles (logs) et déploiement reproductible.

## 2. Périmètre attendu (MVP)

Le MVP attendu correspond à un site qui se lance facilement, protège correctement l'accès au jeu, et permet d'envoyer le score de fin de partie par email.

Fonctionnellement, le MVP inclut :

- Une page de connexion (mot de passe unique) avec messages d'erreur clairs.
- Une page "jeu" inaccessible sans session valide.
- Un mini-jeu jouable au clavier (événementiel : touches, pause, fin de partie).
- Un écran ou panneau "fin de partie" avec le score.
- L'envoi du score par email via PHP à la fin de la session.
- Un journal (log fichier) des événements importants : connexions, accès refusés, score envoyé.

Vous pouvez choisir le jeu (ou en proposer un autre), mais il doit rester raisonnable : règles simples, jouable en quelques minutes, et démontrable (vous allez devoir le mettre en place, donc choisissez un jeu qui vous parle et/ou que vous avez envie de faire).

## 3. Contraintes techniques

Technologies : PHP (pages), HTML/CSS, JavaScript côté navigateur pour le jeu.

Pas de base de données. Les informations temporaires (ex. score courant) peuvent être conservées en mémoire (JS) et/ou en session PHP.

Authentification : un mot de passe unique codé en dur côté serveur (dans un fichier de configuration), jamais dans le JavaScript.

Gestion d'accès : la page du jeu doit vérifier côté serveur qu'une session est ouverte, sinon redirection vers la connexion.

## 4. Exigences sécurité et exploitation (à montrer en démo)

Sécurité (niveau attendu pour ce premier projet) :

- Protection des pages : impossible d'ouvrir directement la page du jeu sans être connecté (contrôle côté serveur).
- Session : démarrage propre, destruction à la déconnexion, et régénération d'identifiant de session après connexion.
- Cookies : paramétrage des attributs recommandés (HttpOnly, SameSite, Secure si HTTPS).
- Gestion des erreurs : messages compréhensibles pour l'utilisateur, sans révéler d'informations techniques.
- Protection "anti-bricolage" : validation serveur du score reçu (type numérique, bornes raisonnables).

Exploitation (basique):

- Déploiement reproductible : fournir une procédure simple (ex. serveur PHP intégré, ou conteneur Docker).
- Logs : fichier de log lisible (format date + type + message).

## 5. Travail demandé - étapes et jalons

Les étapes ci-dessous servent de guide. Vous avancez dans l'ordre et vous gardez des traces (captures, choix, décisions).

### Étape A - Recueil du besoin et cahier des charges (court)

Produire une page (max 2 pages) qui décrit : le but du site, le public, les fonctionnalités du MVP, les règles du jeu choisi (ou proposées), et les critères de réussite pour la démo.

### Étape B - Spécification IHM et maquettage

Réaliser des maquettes simples (papier, Figma ou autre) pour au minimum : page de connexion, page du jeu, écran de fin de partie. Décrire les interactions importantes : navigation, feedback (erreur, réussite), et gestion des états (chargement, fin de partie).

### Étape C - Implémentation MVP (accès + jeu + fin de partie)

Développer la structure du site : connexion, session, page protégée, puis intégrer le jeu (JavaScript). Le jeu doit être jouable, gérer les événements clavier, et afficher clairement le score.

### Étape D - Sécurité minimale et robustesse

Ajouter les protections attendues : régénération de session après login, déconnexion propre, validation serveur du score, et messages d'erreur propres. Préparer un petit scénario de démonstration : accès direct refusé, puis accès autorisé après connexion.

### Étape E - Exploitation : logs, health, déploiement

Mettre en place un fichier de logs et écrire un README de lancement. Ajouter une page /health (ou une info équivalente) et vérifier que la démo peut être lancée rapidement sur une autre machine.

## 6. Livrables attendus

Vous rendez :

- Un dépôt (zip ou git) contenant le code source.
- Un court rapport (5-8 pages) : besoin, maquettes, choix UX, architecture simple, éléments sécurité, éléments exploitation.
- Un README "lancer la démo" (pas à pas).
- Un script de démo (1 page) décrivant ce que vous montrez en soutenance.

## 7. Évaluation (critères)

La note tient compte de la qualité de l'IHM, du fonctionnement du MVP, et de votre capacité à expliquer vos choix.

- IHM / UX : parcours clair, feedback, cohérence visuelle, gestion des états (jeu, pause, fin).
- Programmation événementielle : contrôles clavier, boucle de jeu, gestion des événements et du score.
- Sécurité : page protégée, session correcte, validation serveur, démonstration d'un accès refusé.
- Exploitation : déploiement reproductible, logs utiles, page /health, README compréhensible.
- Qualité générale : clarté du rapport et de la soutenance, code lisible, organisation du projet.

## Annexe - Structure minimale conseillée

Exemple de structure (à adapter) : index.php (connexion), game.php (page protégée), logout.php, score.php (réception et envoi email), assets/ (css, js), config.php (mot de passe, email), logs/app.log.