

Projet : Flipper

Web3 - HETIC

Sommaire

1. Introduction.....	3
1.1. Le flipper : d'un jeu mécanique à une icône culturelle.....	3
1.2 Présentation du projet.....	5
2. Compétences techniques.....	7
Liens avec le monde professionnel.....	7
3. Description du Projet.....	8
3.1 Architecture globale.....	8
3.2 Détails des composants.....	8
3.3 Exemple de flux de données.....	9
4. Fonctionnalités Attendues.....	9
4.1 Fonctionnalités de base (obligatoires).....	9
4.2 Fonctionnalités avancées (bonus).....	10
5. Grille de Notation.....	10
6. Livrables.....	11
7. Annexe - IOT présent dans le flipper.....	12
Solénoïdes - IOT 1.....	12
Contrôleurs - IOT 2.....	13

Projet : Flipper

Projet Flipper : HETIC Web 3

Durée : à définir

Nombre d'étudiants par groupe : 3 à 5

Technologies principales :

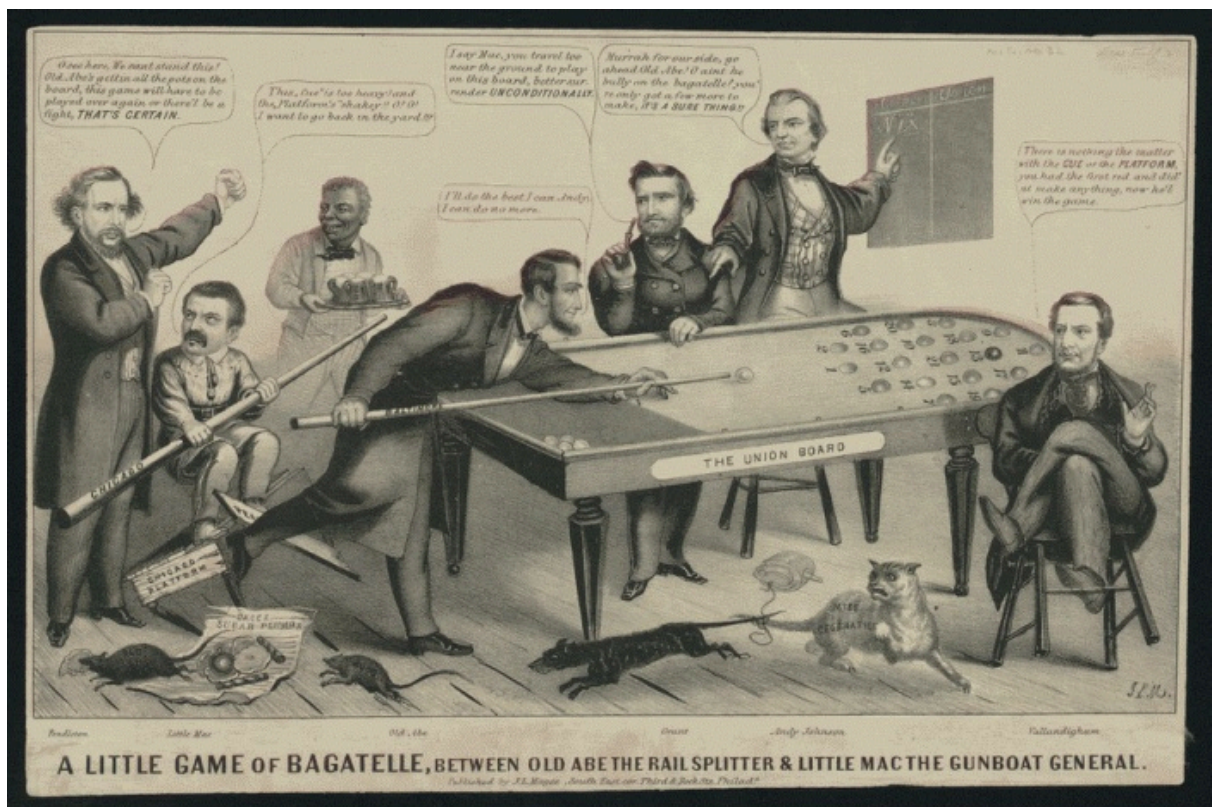
- **Frontend** : Three.js, HTML5 Canvas, WebSockets (Socket.io)
 - **Backend** : Node.js (pour la communication temps réel)
 - **Physique** : Cannon.js ou Ammo.js
 - **IoT** : Arduino/ESP32 (pour simuler les contrôleurs et contrôles haptiques)
 - **IA** : TensorFlow.js ou Python (scikit-learn)
-

1. Introduction

1.1. Le flipper : d'un jeu mécanique à une icône culturelle

Le flipper, ou *pinball*, est bien plus qu'un simple jeu d'arcade : c'est un **objet culturel** qui a traversé les époques, mêlant **mécanique, électronique et art**.

- **Origines (XIX^e siècle)** : Le flipper trouve ses racines dans des jeux comme le *Bagliatelle* (France, XVIII^e siècle), une table inclinée avec des piquets. En **1871**, *Montague Redgrave* brevète le premier jeu de flipper mécanique aux États-Unis, inspiré du croquet. Ces premiers modèles n'avaient **pas de flippers** : le joueur devait incliner la table pour guider la bille.



- **L'ère électrique (1930-1940) :** L'ajout de l'électricité dans les années 1930 révolutionne le flipper :
 - **1931 :** *Ballyhoo* (Bally) introduit les **bumpers** et les **trous** pour marquer des points.
 - **1933 :** *Contact* (Bally) utilise des **relais électromécaniques** pour compter les scores.
 - **1947 :** **Invention des flippers** par *Harry Mabs* (jeu *Humpty Dumpty*). Les joueurs peuvent désormais **agir directement** sur la bille !
- **L'âge d'or (1970-1990) :** Les flippers deviennent des **œuvres d'art interactives** :
 - **1976 :** *Bally* sort *Captain Fantastic*, le premier flipper à **microprocesseur**.
 - **1979 :** *Williams* lance *Gorgar*, le premier flipper à **voix numérique**.
 - **1992 :** *The Addams Family* (Bally) devient le **flipper le plus vendu de l'histoire** (20 270 unités).

- **Design** : Les tables intègrent des **thèmes pop culture** (Star Wars, Indiana Jones) et des **animations complexes**.
- **Déclin et renaissance (2000-à aujourd'hui)** :
 - **Années 2000** : La popularité des jeux vidéo fait chuter les ventes de flippers physiques.
 - **2010s** : Retour en grâce grâce aux **éditions limitées** (Stern Pinball) et aux **simulateurs virtuels** (Visual Pinball, *Pinball FX*).
 - **Aujourd'hui** : Le flipper est un **objet de collection** (prix moyen : 5 000–10 000 €) et un **support d'innovation** (flippers connectés, réalité augmentée).
 - Depuis la fin du Covid, de nombreuses salles d'arcade ré-ouvrent et le flipper y est présent.



1.2 Présentation du projet

Ce projet consiste à **recréer un flipper virtuel complet**, en s'inspirant des mécaniques des machines physiques, mais en utilisant des **technologies web modernes**.

Objectif : Concevoir une **table de flipper interactive** composée de **trois applications distinctes** communiquant en temps réel, avec :

- Un **playfield** (zone de jeu 3D en Three.js).
- Un **backglass** (écran supérieur affichant le score et les animations).
- Un **DMD** (*Dot Matrix Display*, écran rétro pour les messages).
- Une **simulation de contrôleurs** (flippers, boutons) via IoT.
- Une **IA** pour analyser les performances ou jouer contre le joueur.



Pourquoi un flipper ?

Le flipper est un **cas d'étude idéal** pour aborder :

- La **physique en temps réel** (collisions, gravité).
- L'**architecture distribuée** (3 applications + serveur).
- L'**interaction homme-machine** (contrôleurs, feedback visuel et haptique).
- L'**IA et l'analyse de données** (scoring, comportement du joueur).

Ce projet permet aux étudiants de **consolider et appliquer** des compétences clés en développement web, tout en abordant des **concepts avancés** dans un cadre ludique.

2. Compétences techniques

Domaine	Compétences acquises	Technologies associées
Frontend 3D	Maîtrise de Three.js, gestion des scènes, animations, textures.	Three.js, WebGL
Physique interactive	Simulation de collisions, gravité, forces (moteurs physiques).	Cannon.js, Ammo.js
Communication temps réel	Échanges entre applications via WebSockets.	Socket.io, Node.js
Architecture distribuée	Développement d'applications indépendantes mais interconnectées.	Modularité, API REST/WebSockets
IoT (optionnel)	Simulation de périphériques physiques (flippers, boutons).	Arduino, ESP32, MQTT
IA	Analyse de données (scores, trajectoires), prise de décision (bot joueur).	TensorFlow.js, Python (scikit-learn)
UX/UI	Conception d'interfaces rétro-modernes (backglass, DMD).	HTML5 Canvas, CSS, Figma

Liens avec le monde professionnel

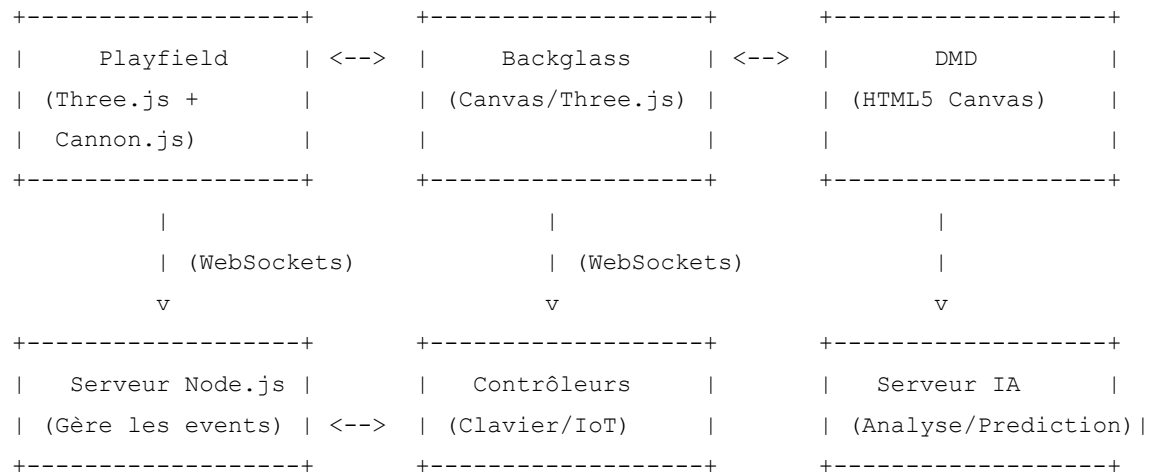
- **Fullstack Web** :
 - Intégration frontend/backend, gestion des states en temps réel.
- **Jeu vidéo** :
 - Moteurs physiques, game design, interactions utilisateur.

- **IoT/Embarqué :**
 - Communication entre devices (simulation des contrôleurs).
- **Data Science :**
 - Analyse des scores, prédiction des trajectoires (IA).

3. Description du Projet

3.1 Architecture globale

Le flipper virtuel est composé de **trois applications web indépendantes** communiquant via un **serveur central** (Node.js + WebSockets) :



3.2 Détails des composants

Composant	Technologies	Fonctionnalités
Playfield	Three.js, Cannon.js	- Affichage 3D de la table. - Gestion de la physique (bille, flippers, bumpers). - Détection des collisions (événements).
Backglass	HTML5 Canvas, Three.js	- Affichage du score. - Animations thématiques. - Feedback visuel (ex : "Tilt!").
DMD	HTML5 Canvas, CSS	- Simulation d'un écran Dot Matrix (style rétro). - Messages ("Game Over", "High Score").

Contrôleurs	Arduino/Clavier	- Simulation des flippers (gauche/droite). - Bouton Start, Nudge (secousse).
Serveur	Node.js, Socket.io	- Centralise les événements (collisions, score). - Synchronise les 3 applications.
IA	TensorFlow.js ou Python	- Analyse des trajectoires. - Proposition de défis. - Bot joueur (optionnel).

3.3 Exemple de flux de données

1. Le joueur appuie sur le **flipper gauche** (via clavier ou bouton IoT).
2. Le **Playfield** active l'animation du flipper et applique une force à la bille.
3. La bille entre en collision avec un **bumper** → événement "**bumper_hit: 100pts**" envoyé au serveur.
4. Le **serveur** relaie l'information au **Backglass** (mise à jour du score) et au **DMD** (affichage "100 POINTS!").
5. L'**IA** analyse la trajectoire et propose un défi (ex : "Atteint le trou central !").

4. Fonctionnalités Attendues

4.1 Fonctionnalités de base (obligatoires)

Fonctionnalité	Description	Points
Playfield 3D	Table de flipper avec bille, flippers, bumpers, trous.	20
Physique réaliste	Collisions, gravité, rebonds (Cannon.js/Ammo.js).	15
Contrôleurs	Flippers gauche/droite + bouton Start (clavier ou IoT).	10
Backglass	Affichage du score, animations basiques.	10
DMD	Écran Dot Matrix affichant score, messages ("Game Over").	10

Communication en temps réel	Synchronisation des 3 apps via WebSockets.	15
Gestion des événements	Détection des collisions (bumpers, trous) et mise à jour du score.	10
Documentation	README, diagrammes, explication du code.	10

4.2 Fonctionnalités avancées (bonus)

Fonctionnalité	Description	Points
Thème original	Design cohérent (ex : "Flipper Steampunk", "Flipper Spatial").	+5
Effets sonores	Musiques et sons (collisions, bumpers).	+5
Multijoueur	2 joueurs en alternance (score séparé).	+10
Éditeur de table	Outil pour créer/modifier des obstacles.	+10
IA adversaire	Bot qui joue contre le joueur.	+10
Analyse des scores	IA qui propose des conseils ("Tu rates souvent le bumper droit !").	+5
Sauvegarde des scores	Base de données (Firebase ou localStorage) pour les high scores.	+5
Effets visuels avancés	Particules, lumières dynamiques (ex : néons).	+5

5. Grille de Notation

Critère	Points	Détails
Fonctionnalités de base	80	Voir tableau 4.1.

Code & Architecture	10	Propreté, modularité, commentaires.
Créativité	5	Originalité du thème, design.
IoT/Contrôleurs	5	Simulation réaliste (Arduino ou clavier).
Bonus	Jusqu'à +20	Voir tableau 4.2.
Documentation	10	Clarté, exhaustivité (README, schémas).
Présentation orale	10	Démonstration fluide, explications techniques.

Barème global (sur 100 + bonus) :

- **≥ 90** : Excellent (toutes fonctionnalités + bonus, code pro).
- **80-89** : Très bon (tout est fonctionnel, quelques bonus).
- **70-79** : Bon (manques mineurs, documentation correcte).
- **60-69** : Passable (fonctionnel mais incomplet ou bugs).
- **< 60** : Insuffisant (manques majeurs, problèmes de synchronisation).

6. Livrables

Chaque groupe devra rendre :

1. **Code source** (3 repositories Git ou 1 mono-repo avec 3 dossiers).
2. **Documentation technique** :
 - README.md (installation, lancement, dépendances).
 - Schéma d'architecture (ex : diagramme de flux).
 - Explication des choix techniques (pourquoi Three.js ? Pourquoi WebSockets ?).
3. **Vidéo de démonstration** (2-3 min) montrant :
 - Le gameplay.
 - Les fonctionnalités avancées (IA, multijoueur, etc.).
4. **Présentation orale** (10-15 min) :
 - Démonstration en direct sur le flipper mis à disposition pour le projet
 - Explication des défis rencontrés et solutions apportées.

7. Annexe - IOT présent dans le flipper

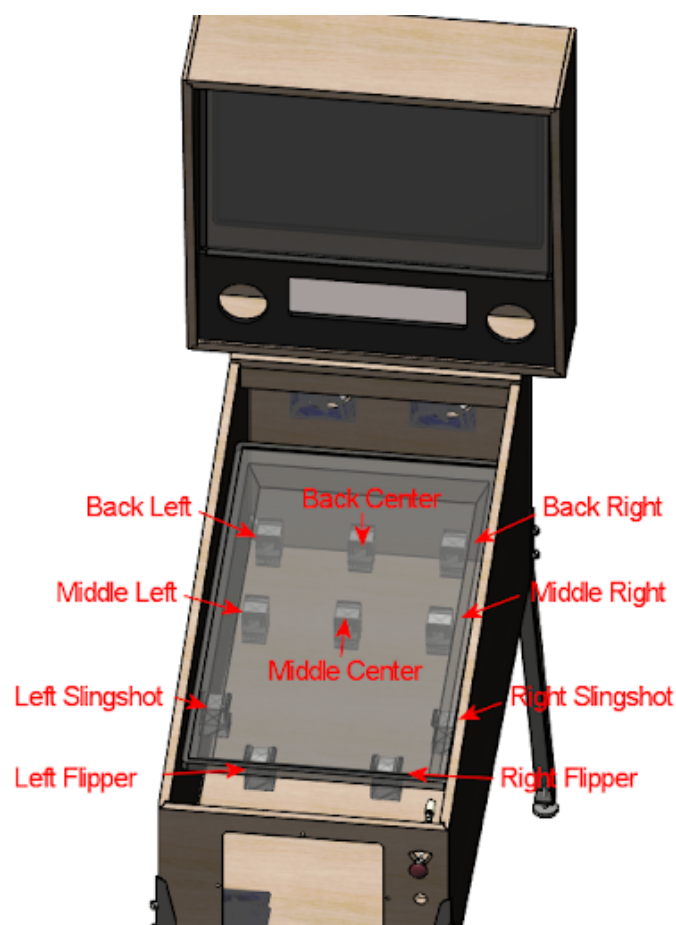
Pour les soutenances sera mis à disposition un flipper virtuel taille réel composée de 3 machines hébergeant chacune une des applications ainsi qu'un ensemble d'objets connectés permettant de simuler les comportements d'un flipper.

Chaque groupe devra implémenter le code permettant de piloter ces IOTs. Le schéma physique de câblage fera l'objet d'un document complémentaire.

Solénoïdes - IOT 1

Les solénoïdes ont pour objectif de “claquer” pour reproduire le comportement réel d'un flipper lorsqu'une bille touche une certaine zone ou que les batteurs sont actionnés.

Le flipper sera équipé de 10 solénoïdes répartis de la manière suivante :



L'ensemble de ces solénoïdes seront pilotés par 2 IOTs pilotant des relais.

Contrôleurs - IOT 2

Un second dispositif IOT connecté au PC du playfield sera mis en place pour simuler des entrées clavier pour les différents boutons :

- Flipper Gauche : X
- Flipper Droit : C
- Start : D
- Pièce entrée : F