

avril 2025

Rapport BTClicker

Jeu incrémental de minage de
Bitcoin

Préparé par : Sacha Simon,
Dorian Roux, Pierre Zhou,
Arthur Garnier

Pourquoi BTClicker ?

Après une année consacrée à la création de sites e-commerce et à la refonte de sites web variés, nous avons souhaité sortir des sentiers battus pour explorer une autre manière de concevoir et de développer des projets. Ce changement de cap s'est traduit par le choix de développer un mini jeu incrémental, afin de diversifier notre approche du développement tout en continuant à mettre en pratique nos compétences techniques. Le thème choisi pour ce projet est celui des cryptomonnaies, et plus spécifiquement du Bitcoin, un sujet d'actualité qui nous semblait pertinent et inspirant. L'univers du minage de Bitcoin s'est naturellement imposé comme base de gameplay, offrant un terrain propice à la gamification et à la progression incrémentale.

Objectifs initiaux

L'objectif principal de cette application est de plonger l'utilisateur dans un monde immersif où il incarne un mineur de Bitcoin. À travers une console (shell) simulée dans le navigateur, l'utilisateur peut interagir en ligne de commande pour miner du bitcoin virtuel, acheter des améliorations, et progresser dans le jeu.

Voici les points clés du gameplay :

- Connexion et enregistrement via la console.
- Accès à un ensemble de commandes disponibles via /rules, permettant d'effectuer des actions comme miner, acheter des améliorations, consulter son solde ou son classement.
- Intégration d'éléments cosmétiques et de valeurs fictives à l'affichage pour renforcer l'immersion et le côté ludique (Facultatif).
- Mise en place d'un classement des meilleurs joueurs, basé sur le nombre total de bitcoins minés.

L'application vise donc à mêler ludification, design immersif et interactivité, tout en s'appuyant sur des mécanismes classiques des jeux de type "idle" ou "incrémentaux".

Périmètre du projet

La plus grande difficulté anticipée réside dans l'équilibrage des mécaniques de jeu :

- Définir des valeurs d'améliorations cohérentes (multiplicateurs, vitesses, coûts, etc.)
- Maintenir un rythme de progression stimulant pour éviter que le joueur ne se lasse.

Pour garantir un développement fluide, nous avons décidé de prioriser la création du shell dans le navigateur ainsi que la mise en place de ses fonctionnalités de base (authentification, commande de minage, affichage du solde, etc.). Cela nous permettra ensuite de répartir plus efficacement les tâches backend.

Les éléments secondaires (comme les cosmétiques ou l'ajout de la dernière amélioration ultime) seront intégrés selon notre avancement, sans compromettre le bon fonctionnement général du jeu.

Le cœur du projet repose donc sur :

- L'interaction via console
- La progression incrémentale par commandes
- L'équilibrage des améliorations

Diagrammes de classe

Voici un aperçu du MCD du site (progression est un JSONB que vous trouverez à la racine du projet)

Users	
id	AUTO_INCREMENT
username	VARCHAR (20)
password	VARCHAR (255)
progression	TEXT

Architecture Logicielle

Base de données : PostgreSQL

Frontend : React, HTML, CSS, TailWind, Next.js

Backend : Express.js, Node

DevOps : Git

Organisation de l'équipe

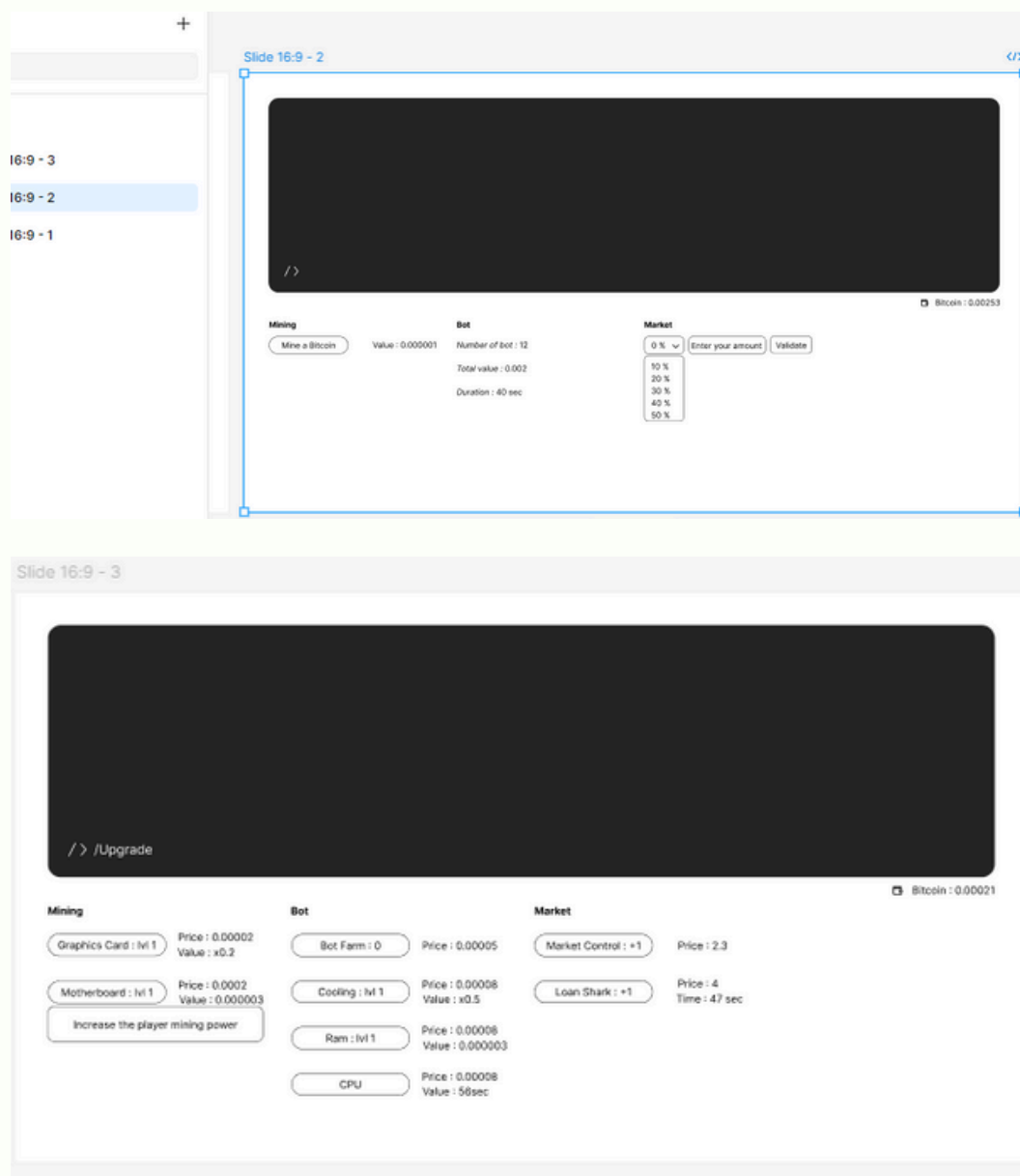
Sacha Simon : Backend/BDD/Diagramme/Rapport

Dorian Roux : Frontend/Figma/Rapport/ppt

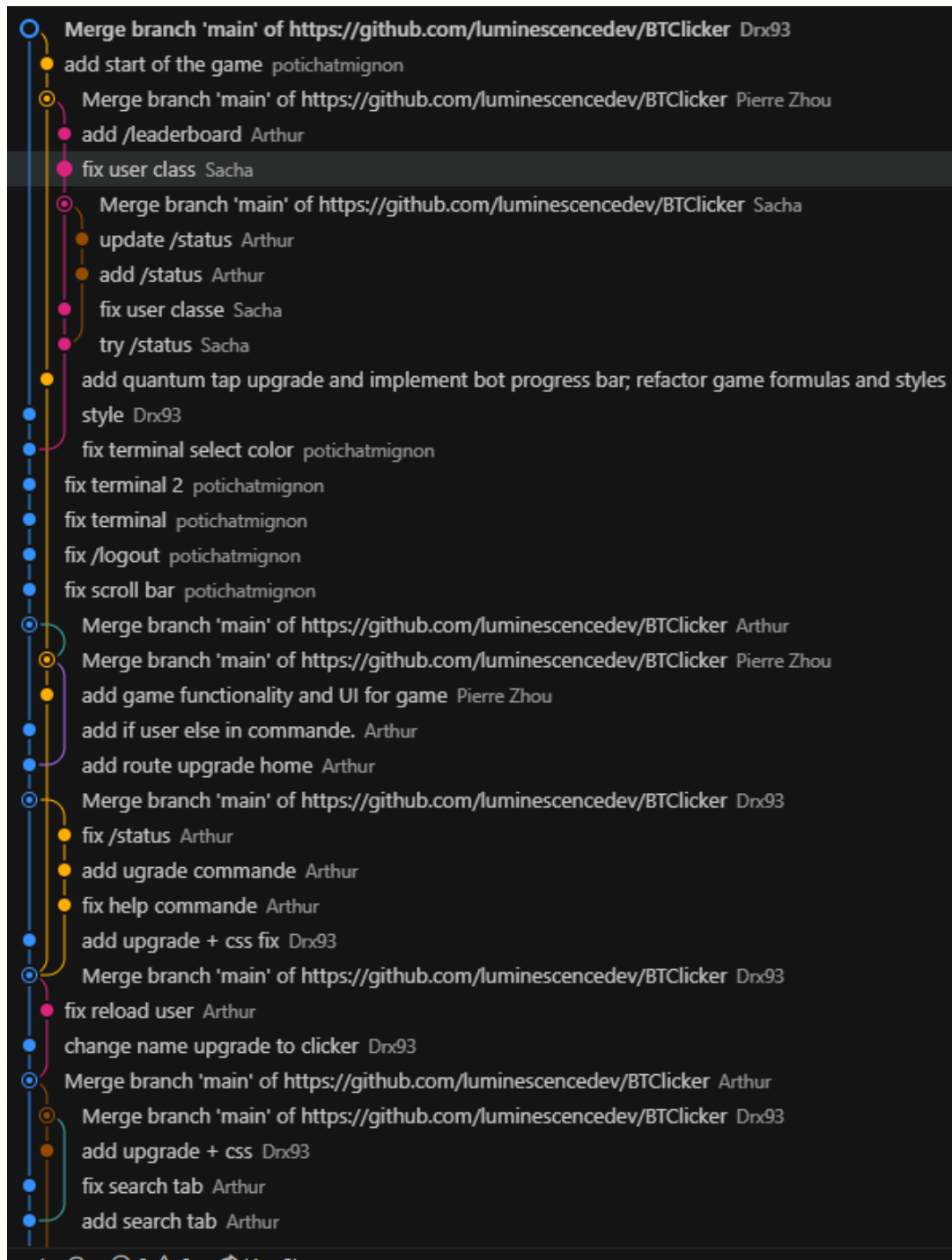
Pierre Zhou : Backend/Frontend

Arthur Garnier : Backend/Frontend

Maquette/Wireframe



Gestion de version



Choix Techniques

ExpressJS

Framework léger et minimaliste pour Node.js.

Facile à prendre en main et très flexible pour créer des API REST.

PostgreSQL

C'est une base de données relationnelle, donc parfaite pour ça. + c'est un outils que l'on gère bien

Tailwind CSS

Utilisé rarement car la majorité était fait en css natif, cependant assez pratique lors de l'utilisation ça nous permet d'aller plus vite.

Next.js

Next.js était la solution pour le frontend parce qu'il est basé sur React, ça nous a permis de faire un site rapide, bien structuré

Design de la solution

Le design de l'application a été pensé pour être minimaliste et épuré, en cohérence avec l'univers des systèmes d'exploitation Unix/Linux. L'objectif est de recréer l'ambiance d'un terminal de commande, afin de renforcer l'immersion et de proposer une expérience utilisateur originale et cohérente avec le thème du minage de Bitcoin.

L'ensemble de l'interaction avec le jeu se fait via une console intégrée dans le navigateur. Les différentes actions disponibles pour le joueur (minage, améliorations, consultation du solde, classement, etc.) sont accessibles exclusivement par saisie de commandes textuelles. À chaque commande reconnue, un ou plusieurs composants visuels s'affichent dynamiquement dans la console, sans jamais quitter la page principale. Cela permet de maintenir une interface centrée sur l'expérience du terminal, tout en conservant une modularité dans le rendu des informations.

Visuellement, l'interface repose sur une palette de couleurs sobres et contrastées (fond sombre, texte clair, touches de vert ou orange pour les retours console), ainsi qu'une typographie monospace pour coller au style des shells traditionnels. Ce choix esthétique contribue à renforcer le côté immersif et à rappeler l'univers technique du minage.

Intégration / Déploiement

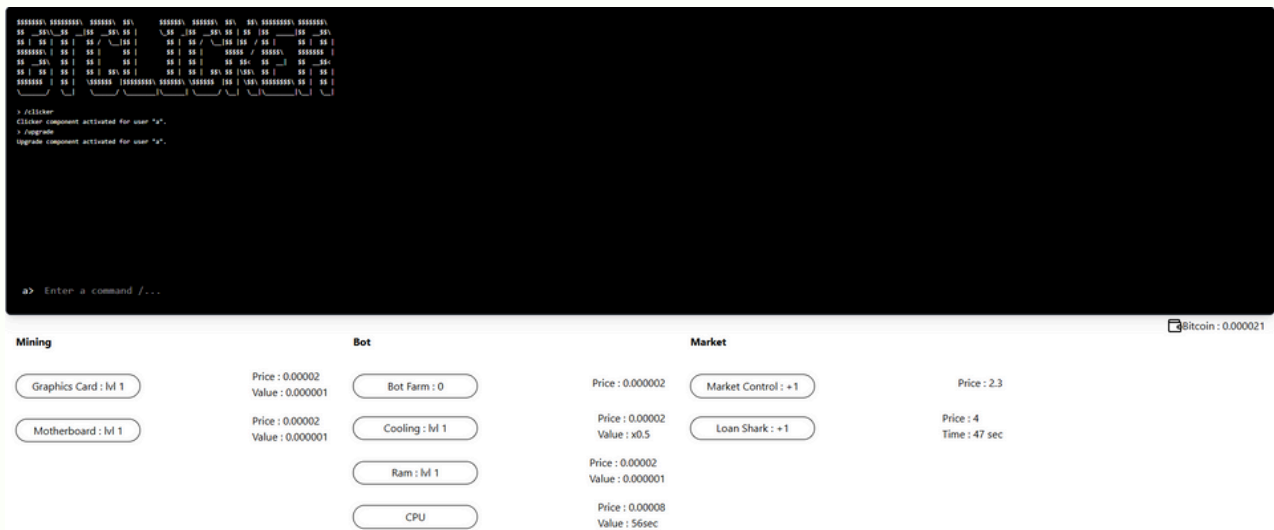
Nous utilisons directement le terminale dans visual studio code pour lancer l'API avec la commande `npx nodemon index.js` ainsi que le frontend avec `npm run dev` dans un second terminal.

```
PS C:\Users\sacha\Desktop\HACKATON\BTClicker> cd api
>> npm i
>> npx nodemon index.js
```

```
PS C:\Users\sacha\Desktop\HACKATON\BTClicker> cd frontend
>> npm i
>> npm run dev
```

Captures d'écran et démonstration (Frontend)





Difficultés rencontrées

Pour les difficultés rencontrées, nous avons eu des problèmes concernant l'intégration du jeu dans le fichier global, car le code du jeu (clicker,upgrade) avait été fait en javascript et qu'il fallait convertir le tout en React.

À côté nous avons le game design pour ce qui est question d'équilibrage de l'avancé dans le jeu.

Leçons apprises et améliorations futures :

Ce projet a été l'occasion pour nous de changer complètement de cadre de développement. Après une année principalement axée sur des projets e-commerce pour la plupart, nous nous sommes dit qu'essayer de sortir des standards en faisant un jeu vidéo, avec un thème original et une interaction en ligne de commande. Au-delà de la technique, nous avons appris la difficulté de bien équilibrer un jeu, qu'il ne soit ni trop, ni trop court, que le joueur reste motivé pour avancé, qu'il ne stagne pas trop...

- Concernant les améliorations à venir, Mettre en place un système de feedback plus interactif, comme des messages dynamiques, des sons ou des mini animations dans la console, pour renforcer la sensation de progression et donner plus de gratification à chaque action.
- Déployer un mode "prestige" ou "reset avec bonus" : une fois un certain palier atteint, le joueur peut tout recommencer avec un multiplicateur ou des commandes exclusives.

Conclusion

À travers ce projet, nous avons pu mettre en place une application web originale, bien différente de ce que nous avons l'habitude de développer. En choisissant de créer un mini jeu incrémental autour du thème du minage de Bitcoin, nous avons exploré de nouvelles logiques de conception, centrées sur la progression, l'interaction ludique et l'expérience utilisateur immersive.

Nous avons pu continuer à approfondir des technologies vue récemment comme next, react et express même si nous avons eu du mal à s'adapter au départ à leur caractéristiques.

Annexes

Voici le lien du github : <https://github.com/luminescencedev/BTClicker.git>

Toutes les informations pour installer les stacks et lancer le projet se trouvent dans le Read.me du repository sur GitHub