Bastien Adam

Software Engineer
Fullstack Developer

07 67 56 47 46
bastien.adam.buom01@gmail.com
Strasbourg - Véhiculé
bastien-adam.fr/projets
bastien-adam-buom01

Spécialiste des applications web réactives. Maîtrise approfondie du développement d'applications serveur en C# et C++. Développeur expérimenté en programmation et tests de bout en bout, consolidés chez Euro-Information Développements.

{ } Compétences

- **Langages informatiques**
  - HTML5 / CSS3
  - JavaScript ES7+ / TypeScript
  - C / C++ / C#
  - Bash
- **DevOps**
  - Kubernetes
  - Docker / Docker Compose
  - HTTP / DNS / SSL
  - Apache HTTPD / Nginx
  - Linux / SSH
- **Web**
  - **Frontend**
    - React.js
    - MUI
  - **Backend**
    - Node.js
    - Express.js
    - Feathers.js
  - **Fullstack**
    - Next.js
    - REST / WebSockets
- **Base de données**
  - MongoDB
  - SQL (MariaDB / PostgreSQL)
- **Outils**
  - Copilot / Code assisté IA
  - GDB / Valgrind
  - Electron
- **Gestion du travail**
  - Git
  - Agile / Kanban
  - CI / CD
- **Langues**
  - **Français**
    - Natif
  - **Anglais**
    - Fluent
  - **Allemand**
    - Scolaire

Expériences professionnelles

Avril 2023 - Avril 2024 — **Ingénieur logiciel** chez **Euro-Information**, section Middleware monétique — *CDI*
- Restructuration et réduction de la dette technique du noyau des middlewares
- Réécriture de la librairie de message bancaire, avec les philosophies RAII et DRY
- Développement de tests unitaires et de bout en bout

Octobre 2022 - Février 2023 — **Consultant DevOps** chez **Radio I|O** — *CDD*
- Architecturation et conteneurisation optimisée d'un service de streaming sans interruption
- Support et intégration frontend

2019/2020 — Développeur sur une borne interactive pour le **Musée de la 2 CV** — *Stage hors cycle*
- Développement — Développement du logiciel configurable de quiz
- SysOp — Installation du système, calibration de l'écran tactile, intégration du logiciel

2014+ — **Fondateur / Développeur** d'un réseau social LGBT+ — **Libre LGBT** — *Projet*
- Imagination et développement du réseau social de A à Z (PWA, TWA, SEO, Accessibilité)
- 40 visiteurs uniques quotidiens, 1er sur Google pour «Réseau social LGBT» en 2023

Formations

Avril 2024 - Décembre 2024 — **Développeur Web Fullstack** formé chez **OpenClassrooms** bac+2 — *Formation diplômante*
- Certifications de mes compétences préacquises par une formation accélérée
- CSS3 / HTML5 / JS / NodeJS (Express / MongoDB) / React / REST / SEO / Intégration Figma

2019+ — **Architecte du numérique** formé à l'**École 42 Paris** bac+5 — *École*
- Maîtrise du code et développement logiciel via projets pratiques (C, C++, Shell, Web, ...)
- Leadership et résolution autonome de problèmes complexes en équipe (GDB, CyberSec, ...)

2017/2019 — **CPGE: TSI** - Lycée Heinrich Nessel bac+2 — *École*

2012+ — **OpenClassrooms** - Formations Web en accès libre — *Formations*

2009 — **Scratch** - Découverte de la programmation — *Éducation*

Centres d'intérêts

- **Technologie**
  - Conception et impression 3D
  - Contributeur dans le gaming OpenSource
- **Jeux vidéo en équipe**
  - Team Fortress 2 (Compétitif)
- **Loisirs Créatifs**
  - Modélisation / Texturing 3D
  - Piano
- **Social**
  - Convention et débats sur la technologie

Projets

- **Euro-Information Developpements**
  - Simulateur de partenaire
  - Testeur de middleware inter-entreprise simulé
  - Middleware de routage inter-entreprises
  - Bibliothèque de messages bancaires ↔ JSON
  - Middleware simple sur SSL
  - Testeur de middleware réseau
  - Framework de middleware réseau
- **Radio I|O**
  - Application web
  - Streaming WebRadio
- **École 42 Paris**
  - Transcendence
  - WebServ
  - ft_Containers
  - ft_Push_Swap
  - ft_Philosophers
  - ft_MiniShell
  - ft_Services
  - ft_LibASM
  - Cub3D
  - ft_Server
  - ft_Printf
  - ft_Get_Next_Line
  - FlipScreen (GameJam)
  - LibFT
  - Présentation web de l'École 42
- **Communauté GrangerHub**
  - Launcher: Jeux vidéos Tremulous & variantes
  - Jeu vidéo: Fork de Tremulous
  - Site web: Présentation de mon fork de Tremulous
- **Musée de la 2CV**
  - Borne interactive personnalisable
- **Libre LGBT**
  - Application web Libre LGBT v4
  - Application web Libre LGBT v3
  - Projet complet
- **CPGE Technologie et Sciences Industrielles**
  - Localisation: Marche piétonne connecté
  - Robot «fil d'Ariane»: Rex
- **Bac STI2D Systèmes d'Information et Numérique**
  - Robot éducatif: Coccino
- ...

EID: Simulateur de partenaire 2024

Il s'agissait de faire un simulateur imitant les API partenaires pour pouvoir réaliser nos tests.

Software Engineer (Main developper)

Réalisations

Implémentation des simulations dont :
- les API clients avec au moins une réponse valide par type de requête
- les simulations des erreurs HTTP
- les simulations des connexions zombies
- les simulations de réponses mal formatées

Soft tasks :
- Conduite de projet
- Déterminer le module interne adéquat à utiliser pour faire notre serveur web
- Se documenter sur les API manquantes de la documentation nécessaire au projet
- Contacter les équipes adéquates pour mettre en place l'application

Résultats

Un projet prêt et efficient pour l'arrivée de nos partenaires.

Environnement

C# JSON HTTPS Windows Visual Studio

EID: Middleware de routage inter-entreprises 2024

Ce projet s'appuie sur trois des projets ci-dessous. Il s'agissait de faire un middleware qui route et convertit les messages bancaires internes en JSON pour ensuite appeler les API HTTP partenaires correspondantes, similaires à des requêtes REST.

Software Engineer (Main developper)

Réalisations

- Routeur en arboraissance
- Multiples conditions (Equal, End with, Regex, Greater than, Lower than, Begin with, ...)
- Connecteurs logiques (ET, OU, OU Exclusif, Inversion binaire)
- Création d'une configuration par partenaire
- Tests automatiques de bout en bout
- Tests automatiques unitaires
- Documentations Markdown

Environnement

C# JSON HTTPS Windows Visual Studio

EID: Bibliothèque de messages bancaires ↔ JSON 2023

Refactoring d'une librairie C# de manipulation et conversion de message bancaire au format interne et au format JSON.

Software Engineer (Main developper)

Réalisations

Refactoring intégral en :
- Programmation orientée objet
  - Suivant la philosophie DRY
  - Suivant la philosophie RAII
- Une API claire inspirée de la philosophie KISS
- Architecturé en arborescence

La réparation de bug majeurs :
- La librairie fonctionne maintenant avec plusieurs appels consécutifs
- La librairie se nettoie correctement après chaque appel :
  - Elle réinitialise les données propres au traitement de l'appel
  - Sans pour autant décharger les données communes

L'implémentation d'options et de configuration :
- Extensible sans recompilation
- Configurable pour les différents cas d'usage
- Support de formats de messages custom

L'implémentation propre de la conversion en JSON :
- Ajout d'accesseur sur chaque objet de l'arbre
- Configurabilité pour couvrir les potentiels cas d'usages de nos partenaires
- Synthèse des données selon la configuration

Environnement

C# JSON Windows Visual Studio

EID: Testeur de middleware réseau 2023

Il s'agissait de faire un testeur pour le framework ci-dessous. From scratch.

Software Engineer (Main developper)

Réalisations

Implémentation :
- Framework interne du testeur pour réutilisation ultérieure
- Serveurs de simulation des réponses
- Logique de simulation client
- Gestion dynamique du service Windows du middleware
- Gestion dynamique des certificats SSL
- Débugueur semi-automatique
- Interface du testeur :
  - Statuts en temps réel de chaque test (en exécution, OK, KO)
  - Couleurs
  - Hiérarchie organisée en sections

Tests réalisés :
- Tests comportementaux :
  - Détection de la présence du certificat SSL
  - Usage du certificat SSL le plus récent
  - Logging
- Connectivité :
  - Comportement lors de messages mal formatés
  - Comportement lors d'une mauvaise connexion
  - Comportement avec un client zombie
  - Multiplexing
  - Connexion SSL

Résultats

Une admiration de l'équipe et de la hiérarchie.

Environnement

C# Trames binaires SSL Windows Visual Studio

EID: Framework de middleware réseau 2023

Il s'agissait de terminer et de corriger une ébauche de framework commun pour nos futurs middlewares réseau.

Software Engineer (Main developper)

Réalisations

Le projet m'a intégralement été confié à partir du stade de prototype.

On notera tout particulièrement :
- La gestion multiplexée des connexions clients
- L'usage de l'asynchrone fournit par le C#
- La gestion multiplexée des requêtes clients au sein d'une même connexion
- La gestion des états clients
- Le décodage des en-têtes binaires
- L'ajout du chiffrement TLS
- Modernisation de l'API en gardant une rétrocompatibilité
- La réalisation des tests (voir le projet ci-dessus)

Soft tasks :
- Rédiger des documentations avec l'incontournable language Markdown

Résultats

Moins de travail de maintenance sur le long terme, réduction de la dette technique.

Environnement

C# Asynchrone Windows Visual Studio

Radio I|O: Streaming WebRadio 2022-2023

Cette partie du projet de la startup consistait à trouver comment streamer efficacement des flux audios à large échelle tout en gardant un code maintenable et compatible avec une majorité d'appareils.
Il fallait gérer différents cas de figure, notamment les comportements en cas de plantage.

Consultant

Réalisations

- Choisir la méthode de streaming :
  - Réaliser des tests de performances
  - Faire des recherches de compatibilité et de faisabilité
  - Choisir le logiciel de streaming :
    - Affiner le cahier des charges
    - Vérifier la présence des fonctionnalités souhaitées
    - Tester les différents comportements et les configurer à souhait
    - Mettre en concurrence les différents logiciels avec leur configuration obtenue
- Organiser nos containers Docker :
  - Créer un container Docker de base
  - Créer un script de démarrage du container :
    - Démarrer plusieurs instances de streaming
    - Optimiser les temps de redémarrage
    - Mettre en place la récupération des données précédentes lorsque applicable
    - Prendre en entrée différentes configurations de streaming
    - Gérer les différents plantages
  - Créer un script d'orchestration des différents containers :
    - Répartir la charge sur les différents containers
    - Répartir la charge sur les différents serveurs
    - Prendre en entrée différentes configurations de streaming
    - Gérer les différents plantages

Environnement

MPEG-DASH Shell Docker Docker Compose Unix

Réseau social autour du jeu Pong, revisité en 3D.

Game Developper

Réalisations

Tout le jeu :
- La logique du jeu dont :
  - La synchronisation Client-Serveur indépendante du ping
  - Mise en pause et interruptions
  - Mode spectateur
  - Physique de la balle :
    - Rebondissement selon la zone de contacte de la raquette
    - Consistance peu importe le ping
    - Animation en sortie de zone de jeu
  - Scores
  - Échauffement
  - Matchmatching selon différents critères
- Carte 3D animées configurables dont :
  - Taille de la zone de jeu
  - Vitesse de la balle
  - Taille des raquettes
  - Contrôles autorisés
  - Position des scores et des avatars
  - Positions de la caméra
  - Matériaux de la zone de jeu
  - Effets 3D
  - Arrière-plan 3D
  - Implantation des cartes :
    - Forest: Une carte pour débutant
    - Classic: Une carte fidèle au jeu original
    - Synthwave: Une carte pour un niveau avancé
- Différents contrôles : Curseur, Molette, Clavier, Gyroscope, ...
- Transitions de la caméra
- L'interface du jeu dont :
  - Statuts : Score, Avatar de l'adversaire, Warmup, ...
  - Paramètres 3D et paramètres du jeu
  - Mise en pause
  - Abandon du jeu
- Des interfaces de l'application web relatives au jeu dont :
  - La liste des parties réactive
  - La page de matchmatching
  - Statut du jeu dans le menu principal de l'application

L'application web :
- Responsivité de l'application
- Layouts réutilisables
- Peaufinage de l'interface générale de l'application

Environnement

Vue.JS Three.JS Quasar Vite.JS Socket.IO REST Axios Nest.JS PostgreSQL JavaScript Shell Nginx Docker Compose Ubuntu Server

Voir le projet →

42: WebServ — 2022

Un serveur web HTTP/1.1 multiplexé en C++ 98 à la Nginx, réalisé sous forme de middleware, from scratch.
Avec lecture / écriture de fichiers, et CGI. Hautement configurable, code facilement extensible.
Benchmarké et validé par un testeur tier.

Software Developper

Réalisations

Le cœur du projet :
- L'API d'architecture en middleware
- Une grande partie de la logique principale définie par la configuration
- Les middlewares de parsing des requêtes HTTP
- Les middlewares de réponses HTTP
- La gestion d'asynchronicité de la lecture et de l'écriture avec epoll
- La gestion des sockets et de la concurrence
- Le traitement en parallèle des requêtes
- La fonctionnalité keep-alive du HTTP 1.1

Quelques middlewares applicatifs :
- Les middlewares de services statiques (lecture) dont :
  - La lecture d'un dossier
  - La lecture d'un fichier, avec détection du mime-type
  - Le service d'upload

Environnement

C++ Ubuntu

Voir le projet →

42: ft_Containers — 2021

Les conteneurs C++98 from scratch, en language C++98, avec leurs itérateurs respectifs.

C++ Developper

Réalisations

L'intégralité du projet comprenant les containers :
- map
- set
- vector
- stack

Leurs itérateurs, itérateurs constants, itérateurs inversés, itérateurs inversés constants, suivant une architecture :
- De simple array (vector)
- De red-black-tree (map)

Tous les tests associés, en s'assurant de la conformité avec le standard C++98 des containers originaux. Ceci inclut les leaks, les performances, et les fonctions non membres.

Quelques annexes comme std::pair

Environnement

C++ Ubuntu

Voir le projet →

42: Cub3D 2020

Jeu vidéo avec rendu en raycasting, en language C. Inspiré du concept du premier FPS : Wolfenstein. From scratch.

Software Developper

Réalisations

Le projet de base :
- Rendu en raycasting
- Rendu en raytracing basique
- Différentes textures
- Optimisations diverses
  - Clipping
  - Profiling avec gprof
  - Optimisation du code C à partir du code Assembly
- Gestion cohérente des contrôles clavier et souris
- Physique élémentaire

Additionnellement :
- Déplacement de la vue vertical possible
- Saut / Déplacement accroupi
- Portes avec ouvertures animées
- Canal alpha
- Sprites animées et collectibles
- Passages secrets

Environnement

C MiniLibX

Voir le projet →

GrangerHub: Jeu vidéo: Fork de Tremulous 2020

Amélioration d'un jeu Open-Source à travers mon fork.
On y retrouve notamment une refonte de l'interface en HD, un nouveau système de police d'écriture dont une par clan, et le support du gamepad.
Toutes ces modifications sont amenées à être ajoutée sur le jeu StellarPrey de la communauté GrangerHub.

Voir le projet →

Musée 2CV: Borne interactive personnalisable

2019 - 2020

Borne de quiz et de présentation vidéo configurable.
Elle a été réalisée sur du matériel de récupération et un RaspberryPi.
Celle-ci a été exploitée au Musée de la 2CV à Troisfontaines.

Main Developer

Réalisations

Toutes les parties software du projet :

- Développement de l'interface sur Electron :
  - Interface d'accueil configurable
  - Lecteur vidéo configurable
  - Quiz multiples configurables
  - Leaderboard
- Installation et configuration du système :
  - Calibration manuelle de l'écran tactile
  - Verrouillage sur l'application (absence d'interface graphique système)
  - Gestion des plantages avec redémarrage automatique

Environnement

Electron | React | Material-UI (MUI) | Shell | JSON | ArchLinuxArm | RaspberryPi

Réalisation complète et administration d'un réseau social réactif. Associatif.
Il a été fait pour être rapide, et parallèlement à mes études en CPGE TSI.
Cette section concerne uniquement le développement de la version 3.

Fondateur / Développeur

Réalisations

L'intégralité du projet.

On a du côté technique :
- Choix des solutions techniques
- Développement et mise en ligne, dont la programmation :
  - D'un système de souscription aux données réactives
  - Traçage du statut en ligne
  - Chaîne de traitement d'image pour les photos de profil avec ImageMagick
  - Création initiale d'un système de notification Push et courriel.
- Mise en place d'une application Android (Trusted Web Activity)
- Protections XSS
- Accessibilité aux malvoyants

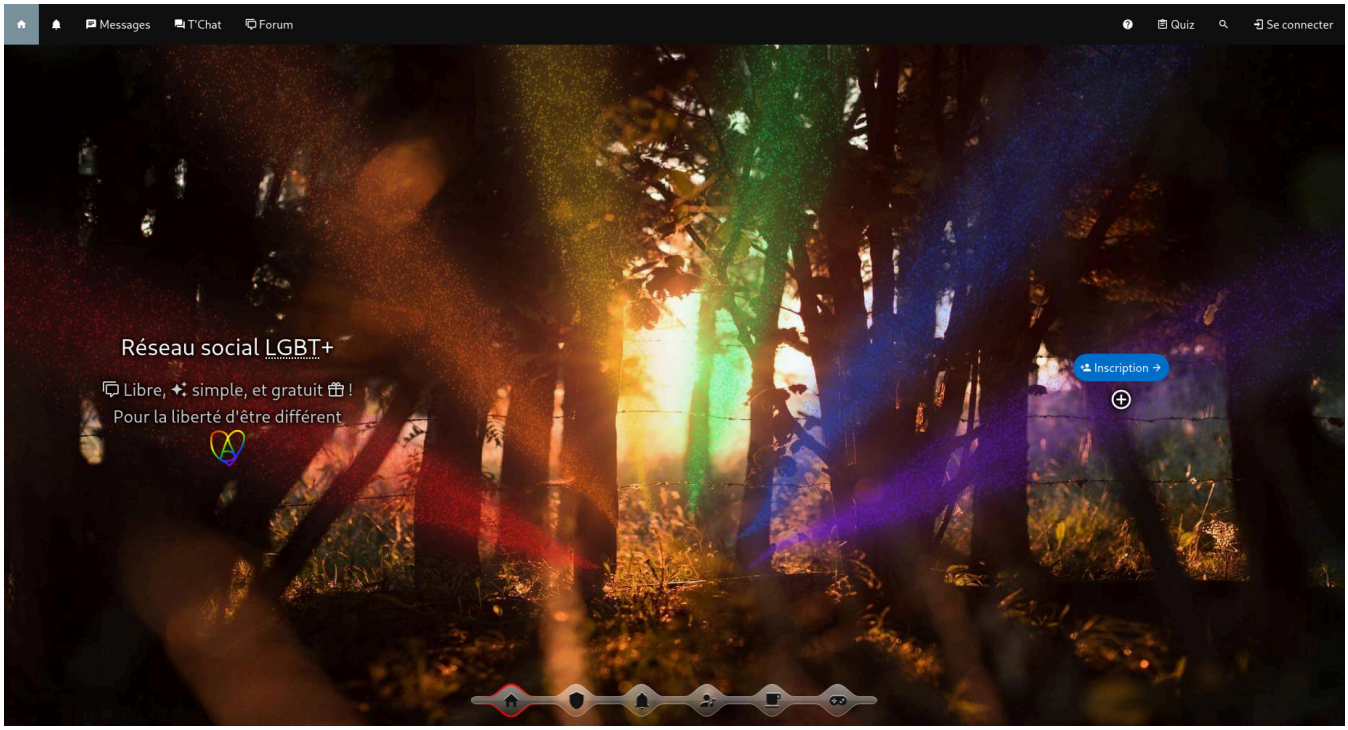
On a également :
- Identité graphique
- Animation du logo
- Animations SVG de présentation
- Référencement SEO et marketing passif

Résultats

- 1er sur le mot clé "réseau social LGBT" sur Google en 2023
- 40 visiteurs uniques quotidiens en moyenne
- De nombreuses lettres de remerciment, en partie de personnes malvoyantes
- Des liens sociaux retrouvés

Environnement

Preact Blaze.CSS Feathers.JS Node.JS Socket.IO MongoDB JSON SVG Nginx Docker Compose Ubuntu Server ImageMagick



CPGE TSI: Localisation: Marche piétonne connecté — 2018 - 2019

Projet des concours de la CPGE TSI.
L'idée est de fournir des panneaux directionnels dynamiques selon la destination souhaitée par l'utilisateur.
La destination sera alors dynamiquement inscrite sur une flèche pré-orienté, par la présence de l'avatar de celui-ci, permettant l'usage concourent à plusieurs utilisateurs.

CPGE TSI: Robot «fil d'Ariane»: Rex — 2019

Un robot de concours programmé en sept jours pour sortir d'un labyrinthe.
J'ai été le seul candidat proposant un algorithme auto-alignant le robot sur les cases.
Cela m'a donné un avantage considérable lorsque le labyrinthe était connu quelques minutes à l'avance, me permettant de programmer mon robot avec des instructions de direction simples.
J'ai aussi été le seul à utiliser la LED RGB du robot pour indiquer ses statut (en attente de lancement, calibration, en course, terminé).

Un algorithme de base suivant la cloison de droite a aussi été développé. J'ai utilisé le capteur de distance ultrason fourni pour regarder, à chaque ouverture se situant la droite de la course du robot, s'il y a des sous-ouvertures, plutôt que de systématiquement rentrer dans celle-ci. Cela m'a fait gagner un temps considérable en compétition.

Voir le projet →

Bac STI2D SIN: Robot éducatif: Coccino — 2017

Projet de mon baccalauréat STI2D.
Le projet a été amené à son terme et est fonctionnel.
Toutes les images relatives au projet présentent dans la présentation ont été réalisées par mes soins, sur le logiciel Inkscape.

Voir le projet →

Libre LGBT: Projet complet — 2014+

Libre LGBT est un réseau social LGBT+. Totalement gratuit, il comporte un forum, un t'chat et une carte interactive.
Il fait partie de mes plus gros projets, mais a une particularité majeure : il y a eu beaucoup de versions différentes développées dans le passé.

En commençant par PHP, finissant avec Next.JS, et passant par Meteor.JS, c'est le projet sur lequel j'ai le plus appris par la pratique dans le domaine du web, au sens large.

Voir le projet →