

# NEXUS LAB

CREATIVE AND GENERATIVE  
CODING ART

## CAHIER DES CHARGES

NexusLab : Plateforme collaborative  
de creative coding et d'art génératif

Nom de l'entreprise : NexusLab studio

Nom du projet : NexusLab

Responsable du projet : Patrick TABURET

Adresse : 20 rue Alphonse Daudet,  
33700 MERIGNAC

Telephone : +33.6.11.01.23.70

Email : taburet.patrick@gmail.com

# TABLE DES MATIÈRES

<b>1. CONTEXTE ET OBJECTIFS</b>	<b>2</b>		
1.1 Description	2		
1.2 Objectifs	2		
1.3 Enjeux	3		
<b>2. ANALYSE PRÉLIMINAIRE</b>	<b>3</b>		
2.1 Veille technologique	3		
2.2 Étude des projets existants	4		
2.3 Analyse des besoins	4		
<b>3. BESOINS FONCTIONNELS</b>	<b>5</b>		
3.1 Fonctionnalités principales	5		
3.2 Gestion des utilisateurs	5		
3.3 Interactivité communautaire	6		
<b>4. BESOINS TECHNIQUES</b>	<b>6</b>		
4.1 Architecture du système	6		
4.2 Gestion des données	7		
4.3 Sécurité	7		
4.4 Serveurs et hébergement	8		
<b>5. CONTRAINTES</b>	<b>8</b>		
5.1 Contraintes de temps	8		
5.2 Contraintes budgétaires	8		
5.3 Contraintes techniques	8		
5.4 Contraintes liées aux utilisateurs	9		
		<b>6. METHODOLOGIES DE DEVELOPPEMENT</b>	<b>9</b>
		6.1 Approche choisie	9
		6.2 Suivi et gestion de projet	10
		<b>7. ERGONOMIE ET DESIGN</b>	<b>10</b>
		7.1 Processus de conception	10
		7.2 Design UI/UX	10
		7.3 Accès aux fonctionnalités	11
		7.4 Arborescence et navigation	11
		<b>8. MAINTENANCE ET EVOLUTIVITE</b>	<b>11</b>
		8.1 Suivi des performances	11
		8.2 Mises à jour	12
		8.3 Évolutivité	12
		<b>9. LIVRABLES</b>	<b>12</b>
		9.1 Application web	12
		9.2 Application mobile	12
		9.3 Documentation technique	13
		9.4 Documentation utilisateur	13
		<b>10. ANNEXES</b>	<b>13</b>

# 1. CONTEXTE ET OBJECTIFS

## 1.1 Description

Le projet consiste en la création d'une plateforme collaborative dédiée à l'art génératif et au creative coding en général, accessible à la fois via une application web et une application mobile. Cette plateforme permettra aux utilisateurs - qu'ils soient artistes, codeurs, ou amateurs de nouvelles technologies - de créer, partager et collaborer sur diverses formes d'œuvres interactives.

Le **creative coding** est une discipline artistique impliquant l'utilisation du code dans son processus de création, que ce soit pour des visuels, du son, des performances interactives, ou d'autres formes d'expression artistique. Cela englobe une large gamme de pratiques artistiques utilisant la programmation, dont l'Art Génératif, le Live Coding et le Data Art.

L'**Art Génératif** est une forme d'expression artistique qui utilise des algorithmes et des systèmes informatiques pour créer des visuels uniques de manière autonome ou semi-autonome.

Le **Data Art** utilise des ensembles de données comme matériau artistique, transformant des informations (statistiques, données environnementales, réseaux sociaux, etc.) en œuvres visuelles ou interactives. Il permet de rendre des données abstraites plus accessibles et esthétiques tout en mettant en reliefs certains patterns, tendances ou réalités à travers ces données.

Le **Live Coding** quant à lui, consiste à écrire et modifier du code en temps réel. Cette technique est généralement utilisée pour créer des médias audiovisuels, et peut être pratiqué seul ou à plusieurs.

Ces trois pratiques constitueront les principales formes d'expression créative proposées sur la plateforme, mais nous restons ouverts aux évolutions et aux propositions de la communauté. La plateforme sera donc un espace numérique où les utilisateurs pourront partager et manipuler des œuvres interactives, explorer des créations d'autres artistes et co-crée des projets en temps réel, tout en bénéficiant d'une communauté ouverte et créative.

## 1.2 Objectifs

Les objectifs à **court terme** du projet sont de mettre en place un espace fonctionnel permettant aux utilisateurs de créer et de partager facilement leurs œuvres. Cela comprend des fonctionnalités de base telles que l'authentification, la création et la visualisation d'œuvres numériques, ainsi que la possibilité d'explorer et d'interagir avec le travail des autres utilisateurs.

À **long terme**, l'objectif est d'établir cette plateforme comme un pôle de référence pour les artistes et les programmeurs intéressés par le creative coding. Il s'agit non seulement de favoriser la collaboration entre les créateurs à travers le monde, mais aussi de promouvoir ces pratiques auprès d'un plus large public.

L'aspect communautaire est donc fondamental, favorisant l'échange et le partage entre les utilisateurs, qu'ils soient visiteurs, artistes ou experts en programmation. Avec le temps et l'implication de ses membres, la plateforme pourra fournir des ressources pédagogiques, des tutoriels, et organiser des événements collaboratifs (comme des hackathons ou des expositions virtuelles), renforçant ainsi la notoriété du creative coding en tant que moyen d'expression artistique.

## 1.3 Enjeux

L'un des principaux enjeux de ce projet réside dans la **création** autour de ces **pratiques artistiques**, en offrant un espace où les utilisateurs peuvent facilement créer et partager leurs travaux, recevoir des retours d'autres utilisateurs, et collaborer sur des œuvres communes. La plateforme contribuera à développer la création et à renforcer les interactions au sein de cette communauté artistique qui se verra bienveillante et inclusive. Les utilisateurs pourront y échanger librement sur leurs créations, s'entraider et apprendre les uns des autres dans un environnement respectueux.

L'importance de ce projet tient également dans son potentiel à **démocratiser l'art génératif** et plus généralement à populariser les pratiques artistiques en lien avec les nouvelles technologies. Actuellement, ces disciplines sont souvent perçues comme réservées à des experts en programmation. En proposant des outils accessibles et en encourageant l'échange de savoirs, la plateforme vise à rendre ces pratiques plus accessibles à un public non-initié, tout en soutenant l'innovation et l'expérimentation artistique.

## 2. ANALYSE PRÉLIMINAIRE

### 2.1 Veille technologique

Dans le domaine du creative coding, plusieurs technologies et plateformes émergent comme des acteurs clés.

#### Langages et frameworks :

- **Processing** : Largement utilisé pour l'art génératif et le creative coding.
- **p5.js** : Version JavaScript de Processing, permettant la création d'œuvres interactives sur le web.
- **Three.js** : Bibliothèque JavaScript pour la création de graphiques 3D.
- **Python** : apprécié pour sa simplicité et sa grande communauté. Des bibliothèques comme Pygame ou Pyglet permettent de créer facilement des visuels.
- **C++** : Utilisé via des frameworks comme openFrameworks, offrant d'excellentes performances pour des projets complexes. Il est prisé des artistes et développeurs expérimentés.
- **GLSL (OpenGL Shading Language)** : Langage de programmation spécialisé utilisé pour écrire des shaders dans WebGL. Il offre un contrôle puissant et flexible sur le rendu graphique, permettant des effets visuels complexes et des performances élevées directement dans le navigateur.

#### Plateformes de partage :

- **GitHub** : Bien que non spécifique à l'art, très utilisé pour partager du code créatif.
- **CodePen** : Dédié plus généralement au partage de code frontend, dispose d'une large communauté, supporte de nombreuses technologies web.

#### Technologies émergentes :

- **Machine Learning** : Intégration croissante de l'IA dans l'art génératif.

## 2.2 Étude des projets existants

Plusieurs plateformes existantes offrent des fonctionnalités similaires à celles envisagées pour notre projet :

Plateformes	Forces	Faiblesses
<b>OpenProcessing</b>	Grande communauté, facilité de partage des sketches Processing.	Limité principalement à Processing, peu de fonctionnalités collaboratives.
<b>Glitch</b>	Collaboration en temps réel, large éventail de technologies supportées.	Non spécifique à l'art génératif, peut être complexe pour les débutants.
<b>ArtBlocks</b>	Plateforme spécialisée dans l'art génératif NFT.	Focalisé sur la blockchain, moins accessible pour les artistes traditionnels.
<b>Alba</b>	Plateforme spécialisée dans l'art génératif NFT.	Focalisé sur la blockchain, moins accessible pour les artistes traditionnels.
<b>fx(hash)</b>	Plateforme spécialisée dans l'art génératif NFT.	Focalisé sur la blockchain, moins accessible pour les artistes traditionnels.
<b>Bohét</b>	Plateforme spécialisée dans la vente de tirages d'art génératif sous forme d'impressions physiques.	Plateforme jeune encore en développement. Actuellement dépendante de Shopify pour le backend e-commerce.

## 2.3 Analyse des besoins

Basé sur l'étude du marché et des projets existants, les besoins suivants ont été identifiés :

- **Création et partage :**

Interface intuitive et accessible à tous les niveaux de compétence pour créer et éditer des œuvres d'art génératives. Outils de partage simples à utiliser.

Support de multiples langages de programmation et frameworks pour encourager la diversité créative.

- **Collaboration et communauté :**

Fonctionnalités de création d'œuvres ou de projets collaboratifs.

Système de commentaires, de like et de feedback sur les œuvres pour encourager l'interaction.

Profils utilisateurs personnalisables permettant aux artistes de présenter leur travail et leur

démarche.

Système de suivi et de découverte d'artistes pour faciliter les connexions au sein de la communauté.

- **Apprentissage, et partage de ressources :**

Promotion active de la philosophie de l'open source et du partage de connaissances.

Mise à disposition d'un espace dédié au partage de techniques, d'astuces ou de ressources éducatives.

- **Accessibilité et inclusivité :**

Interface responsive pour une utilisation optimale sur différents supports (desktop, tablette, smartphone).

Application mobile dédiée pour faciliter l'accès et l'interaction hors du foyer ou en déplacement.

Design inclusif prenant en compte les besoins des utilisateurs de tous niveaux et capacités.

- **Aspect non mercantile et éthique :**

Focalisation sur la création, le partage et l'apprentissage plutôt que sur la monétisation des œuvres.

Absence de fonctionnalités de vente directe ou de marketplace pour préserver l'intégrité artistique.

Promotion d'une éthique de partage et de collaboration, en contraste avec les plateformes orientées profit.

Accent mis sur la protection des données utilisateurs.

Cette analyse met en évidence le potentiel d'innovation de notre plateforme, qui se distinguera par son approche centrée sur la communauté, l'apprentissage et le partage. En combinant les fonctionnalités pertinentes des solutions existantes tout en répondant aux besoins spécifiques de la communauté du creative coding, nous visons à créer un espace unique où la créativité et la collaboration priment sur les considérations commerciales.

Notre plateforme doit se positionner comme un acteur éthique et innovant dans l'écosystème de l'art numérique, favorisant ainsi l'émergence d'une communauté engagée et dynamique.

## 3. BESOINS FONCTIONNELS

### 3.1 Fonctionnalités principales

Partage et intégration de scènes d'arts génératifs et de visualisation de data dotés d'interfaces manipulables par les utilisateurs.

Création et partage d'œuvres d'art générées à partir de ces scènes.

Accès aux scènes et aux créations pour tous les utilisateurs, possibilité de retravailler ces créations à partir des derniers paramètres utilisés au moment de la sauvegarde de l'œuvre.

Système de collaboration entre utilisateurs : espace de co-crédation en temps réel (collective drawing / live coding)

### 3.2 Gestion des utilisateurs

Inscription et authentification sécurisée (via email ou réseaux sociaux).

Gestion des profils utilisateurs (édition des informations, galerie personnelle).

Gestion des droits et permissions (rôles Admin, Artist, User).

Attribution des accès en fonction des rôles utilisateur

ROLES ACCESS	VISITOR (no role)	USER	ARTIST	ADMIN
Visit Gallery / Manipulate scenes Read community space	✗	✗	✗	✗
Save scenes results Like & comment on gallery Write in community space User section		✗	✗	✗
Artist Dashboard (manage my scenes) Artist community section			✗	✗
Admin dashboard (manage users and gallery) Moderate community				✗

### 3.3 Interactivité communautaire

Système de feedback et de like sur les œuvres.

Possibilité de suivre d'autres créateurs et d'être informé de leurs dernières créations par le biais de notifications.

Accès à un espace communautaire de discussion de type forum, avec un espace dédié aux visiteurs, aux membres et aux rôles Artist.

## 4. BESOINS TECHNIQUES

### 4.1 Architecture du système

Notre plateforme sera construite sur une architecture moderne et robuste, conçue pour offrir une expérience utilisateur fluide tout en assurant la sécurité et la scalabilité nécessaire à une application potentiellement gourmande en ressources.

#### • Backend :

Symfony sera utilisé comme framework principal pour le backend, avec donc PHP comme langage pour la logique métier. Son architecture suivant le modèle MVC offre une structure solide et flexible pour l'application.

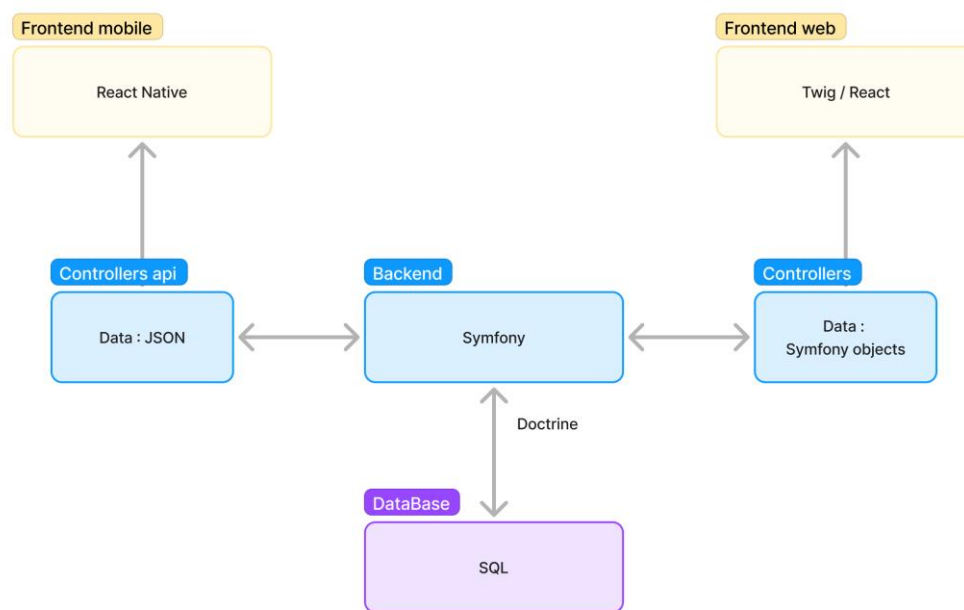
Une API sera implémentée au niveau des controllers Symfony pour faciliter la communication entre le backend et les différents clients (web et mobile).

Une **base de données relationnelle** sera utilisée pour stocker efficacement les informations des utilisateurs, les métadonnées des œuvres d'art, et les relations entre les différentes entités du système.

- **Frontend web :**

Le moteur de template Twig sera intégré pour le rendu côté serveur, et React sera utilisé comme bibliothèque pour le développement de certains composants. Permettant ainsi la création d'une interface utilisateur dynamique et responsive, cette solution hybride combine les avantages du rendu serveur et du rendu client.

- **Application mobile :** React Native pour le développement cross-platform, permettant le partage de code avec les composants React de l'application web.



## 4.2 Gestion des données

- **Types de données à gérer :** Œuvres numériques (métadonnées, paramètres), fichiers multimédias, code source des scènes, données utilisateurs.

- **Performances et scalabilité :** Doit pouvoir supporter un grand nombre d'œuvres génératives avec des ressources partagées. Des index appropriés et des requêtes optimisées seront mis en place pour assurer des temps de réponse rapides.

## 4.3 Sécurité

- **Authentification :**

Mise en place d'une authentification sécurisée utilisant le système fournis par Symfony pour l'application web, et un système de token JWT et de refresh token pour l'application mobile.

- **Protection contre les attaques :**

Implémentation de mesures de sécurité contre les attaques courantes telles que XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery), et les injections SQL. Notamment via le security bundle



de Symfony.

A terme utilisation de HTTPS pour toutes les communications.

- **Cryptage des données :**

Les données sensibles telles que les mots de passe et les informations personnelles seront cryptées avant d'être stockées dans la base de données. Utilisation de techniques de hachage sécurisées et modernes pour les mots de passe.

#### 4.4 Serveurs et hébergement

- **Infrastructure cloud :**

Utilisation d'une infrastructure cloud (AWS) pour l'hébergement de l'application, offrant flexibilité et scalabilité. Mise en place d'un système de load balancing pour répartir la charge et assurer une haute disponibilité.

- **Sauvegarde et récupération de données :**

Mise en place de sauvegardes régulières et automatisées de la base de données et des fichiers utilisateurs. Élaboration d'un plan de reprise après sinistre pour assurer la continuité du service en cas de problème majeur.

### 5. CONTRAINTES

#### 5.1 Contraintes de temps

Le projet sera développé en deux temps, une première phase de conception et de développement pour assurer rapidement la livraison d'un prototype fonctionnel de l'application (Avant le 31 octobre 2024).

Et dans un second temps la phase de test et de déploiement si le prototype est validé par le client.

#### 5.2 Contraintes budgétaires

Le projet étant basé sur une approche **open source**, les contraintes budgétaires seront principalement liées aux coûts d'infrastructure et de main-d'œuvre :

**Hébergement :** Utilisation de solutions cloud économiques et évolutives.

**Licences :** Priorité aux outils et frameworks open source pour minimiser les coûts de licence.

**Développement :** Optimisation des ressources en favorisant la contribution de la communauté open source. L'objectif est de maintenir les coûts au minimum tout en assurant la qualité et la pérennité du projet.

#### 5.3 Contraintes techniques

- **Compatibilité :**

**Navigateurs :** Assurer la compatibilité avec les dernières versions des principaux navigateurs (Chrome, Firefox, Safari, Edge).

**Systèmes d'exploitation :** Garantir le fonctionnement sur Windows, macOS, et Linux pour la version desktop.

Mobile : Développer une version native compatible iOS et Android, en tenant compte des spécificités de chaque plateforme.

- **Technologies spécifiques :**

Intégration de p5.js et d'autres bibliothèques ou langages spécifiques à l'art génératif dans Symfony et dans l'environnement mobile natif (Webviews).

Intégration d'un éditeur de code en ligne pour le live coding.

Possibilité d'interagir simultanément sur la même interface que d'autres utilisateurs (WebSockets).

Optimisation des performances pour assurer une expérience fluide.

## 5.4 Contraintes liées aux utilisateurs

- **Facilité d'utilisation :**

Conception d'une interface intuitive, ludique et conviviale, adaptée aux créateurs non-techniques.

Mise en place d'une documentation pour faciliter l'apprentissage et l'utilisation de la plateforme.

Définir un protocole simple et clair pour faciliter le partage de code source et son intégration dans l'application en tant que nouvelle scène créative.

- **Accessibilité :**

Conformité aux normes WCAG (Web Content Accessibility Guidelines) pour rendre la plateforme accessible aux personnes en situation de handicap.

Implémentation de fonctionnalités telles que : Navigation au clavier, compatibilité avec les lecteurs d'écran, contraste de couleurs suffisant, textes alternatifs pour les images, sous-titres pour les contenus audio et vidéo.

- **Contraintes légales et éthiques :**

Respect des réglementations sur la protection des données personnelles (RGPD en Europe).

Mise en place de politiques claires concernant les droits d'auteur et la propriété intellectuelle des œuvres créées sur la plateforme.

Élaboration d'un code de conduite pour la communauté afin de promouvoir un environnement respectueux et inclusif.

Gestion de la modération des espaces de discussion.

Ces contraintes guideront le développement et la mise en œuvre de la plateforme, assurant qu'elle soit non seulement fonctionnelle et performante, mais aussi accessible, éthique et respectueuse des besoins variés de sa communauté d'utilisateurs.

## 6. METHODOLOGIES DE DEVELOPPEMENT

### 6.1 Approche choisie

Pour mener à bien ce projet nous avons opté pour une approche **Agile** basée sur la méthode **Kanban**. Cette méthodologie nous permettra de maintenir une flexibilité optimale tout en assurant une visualisation claire du flux de travail.

- **Adaptabilité** : Kanban offre une grande flexibilité pour s'adapter rapidement aux changements et aux retours des utilisateurs.
- **Collaboration en équipe** : La visualisation du travail encourage une meilleure communication et collaboration au sein de l'équipe.
- **Cycles courts** : Kanban permet une livraison continue des fonctionnalités, assurant des retours rapides et une amélioration itérative du produit.

## 6.2 Suivi et gestion de projet

Dans le cadre de la gestion de projet, voici les outils utilisés pour assurer un suivi optimal des tâches et une collaboration fluide au sein de l'équipe.

**Trello**, pour la gestion du tableau Kanban, la visualisation du flux de travail et le suivi des tâches.

**GitHub**, pour le contrôle de version et la gestion du code source. .

## 7. ERGONOMIE ET DESIGN

### 7.1 Processus de conception

Le processus de conception de l'interface a suivi une démarche itérative, incluant les phases de moodboarding, wireframing et maquettage. Ces étapes ont permis de définir l'identité visuelle, l'architecture des pages et l'organisation des fonctionnalités.

Le moodboard a été utilisé pour explorer des inspirations esthétiques, alignant les choix visuels avec l'univers créatif de l'art génératif et des nouvelles technologies, en vue de l'élaboration de la charte graphique.

Les wireframes ont permis d'établir la structure fonctionnelle des différentes pages et parcours utilisateurs, en s'assurant que la navigation soit simple et intuitive. Enfin, des maquettes haute-fidélité ont été développées pour illustrer le design final avant l'implémentation.

En complément de cette démarche, un logo spécifique a été créé pour l'application, incarnant l'identité de la plateforme NexusLab. Ce logo, représentant la lettre « N », évoque à la fois l'esprit technologique et créatif de la plateforme. Sa forme s'inspire des motifs organiques que l'on retrouve fréquemment dans l'art génératif, en particulier ceux issus des simulations de systèmes autonomes et de réseaux. Ce design reflète l'interconnexion et la dynamique collaborative qui sont au cœur de l'expérience utilisateur et de la création numérique sur NexusLab.

**Vous retrouverez le résultat de ces différentes étapes de conception en annexe (p. 14, 15, 16)**

### 7.2 - Design UI/UX

L'interface utilisateur doit allier efficacité, esthétique et simplicité. Le design adopte une approche minimaliste, mettant en avant les œuvres d'art génératives tout en garantissant une navigation fluide et agréable. L'objectif est de minimiser la complexité visuelle pour favoriser une immersion complète dans le processus créatif. Le style graphique s'inspire de l'univers de la science-fiction et du rétro gaming, apportant une touche ludique à l'expérience utilisateur.

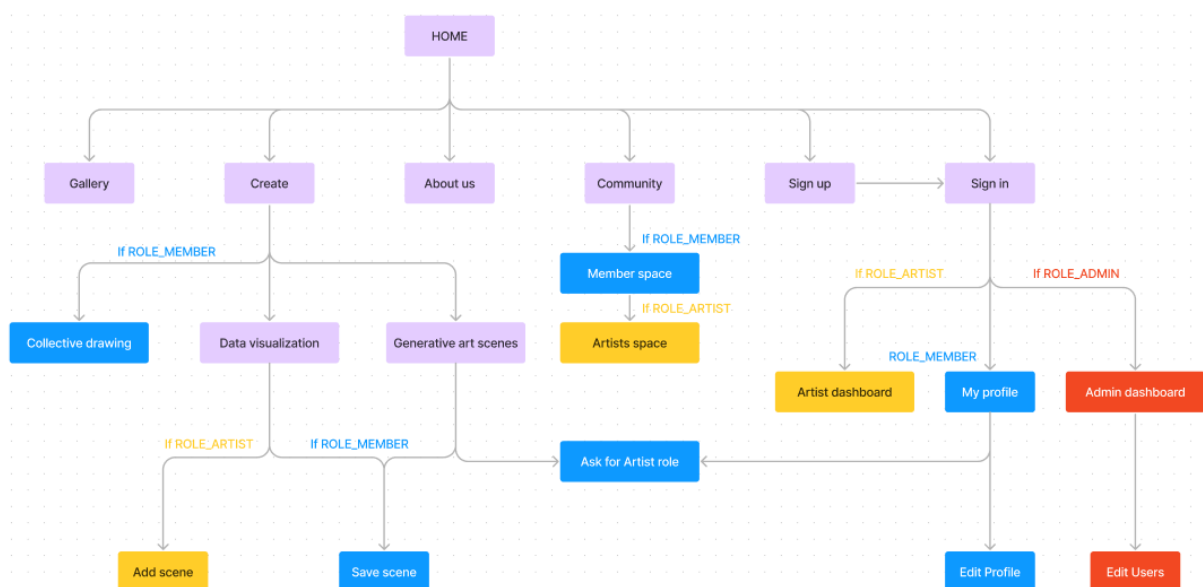
Le responsive design est un élément clé du projet, garantissant une adaptation optimale de l'interface sur tous types d'appareils (ordinateurs, tablettes et smartphones). L'expérience utilisateur est ainsi cohérente et agréable, quel que soit le support utilisé.

### 7.3 - Accès aux fonctionnalités

Les principales fonctionnalités de la plateforme seront facilement accessibles à partir de n'importe quel écran. Des raccourcis doivent permettre d'atteindre rapidement les outils essentiels tels que la création et le partage des œuvres, l'accès aux projets collaboratifs ou l'accès à la galerie. Une architecture intuitive a été mise en place afin de rendre ces actions disponibles en un minimum d'étapes, optimisant ainsi le flux de travail des utilisateurs.

### 7.4 Arborescence et navigation

L'arborescence du site a été pensée pour offrir une navigation fluide et intuitive, en facilitant l'accès aux principales fonctionnalités.



## 8. MAINTENANCE ET EVOLUTIVITE

La maintenance et l'évolutivité de la plateforme sont des aspects essentiels pour garantir la pérennité et la stabilité du projet. Elles incluent le suivi des performances, la mise en œuvre d'améliorations continues, la correction de bugs, ainsi que l'ajout de nouvelles fonctionnalités pour répondre aux besoins des utilisateurs.

### 8.1 Suivi des performances :

Afin de garantir des performances optimales, des outils de monitoring seront mis en place pour suivre en temps réel le comportement de l'application.

Ces outils permettront de :

- Surveiller les temps de réponse du backend et du frontend.

- Suivre l'utilisation des ressources (processeur, mémoire, bande passante).
- Détecter les problèmes de performance (latence, ralentissements) avant qu'ils n'affectent les utilisateurs.

Les données collectées grâce à ces outils permettront d'optimiser l'infrastructure et d'assurer la fluidité des interactions sur la plateforme, notamment lors de l'exécution des scripts chargés de générer les scènes visuelles.

## 8.2 Mises à jour :

Le processus de mises à jour inclura des améliorations régulières à la fois fonctionnelles et techniques, ainsi que des correctifs pour assurer la stabilité de l'application.

Les mises à jour seront planifiées afin de minimiser les interruptions de service. Une maintenance préventive pourra également être effectuée de façon régulière pour s'assurer que les systèmes de sécurité et les sauvegardes fonctionnent correctement.

## 8.3 Évolutivité

L'application a été conçue pour être évolutive et capable de s'adapter aux futurs besoins des utilisateurs et des avancées technologiques. Le plan d'évolution prévoit :

- L'ajout de nouvelles fonctionnalités basées sur les retours des utilisateurs et sur les tendances dans le domaine du creative coding.
- L'intégration de nouvelles formes d'art ou d'outils créatifs (par exemple, l'ajout d'outils pour la création en réalité virtuelle ou augmentée).
- L'extension des outils de live coding avec des éditeurs de code plus complets et des visualisations plus avancées, permettant de travailler avec de nouvelles technologies (ex: WebGL, machine learning).
- Un support offline pour permettre la création en mode déconnecté pour l'application mobile.

# 9. LIVRABLES

## 9.1 Application web

Déploiement en production d'une application web complète et fonctionnelle.

Conception responsive, compatible avec ordinateurs, tablettes et smartphones.

Respect des normes d'accessibilité (WCAG) pour garantir une expérience utilisateur inclusive.

## 9.2 Application mobile

Application native ou cross-platform déployée sur l'App Store (iOS) et Google Play (Android).

Intégration des fonctionnalités principales de la version web (création, partage, collaboration).

Notifications push pour les interactions en temps réel (commentaires, likes, news).

### 9.3 Documentation technique

Instructions pour l'installation et le déploiement des applications (environnements de développement, test, production).

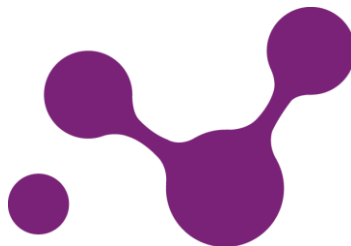
Documentation des processus d'intégration continue (CI/CD) et gestion des environnements.

### 9.4 Documentation utilisateur

FAQ couvrant les questions fréquentes sur l'utilisation des outils et la création de scènes.

Documentation claire et complète sur l'ajout de scènes par les utilisateurs ayant obtenu le rôle Artist, en vue de faciliter l'implémentation de ces dernières dans l'application.

Guide de support utilisateur pour la soumission de bugs et demandes d'assistance.

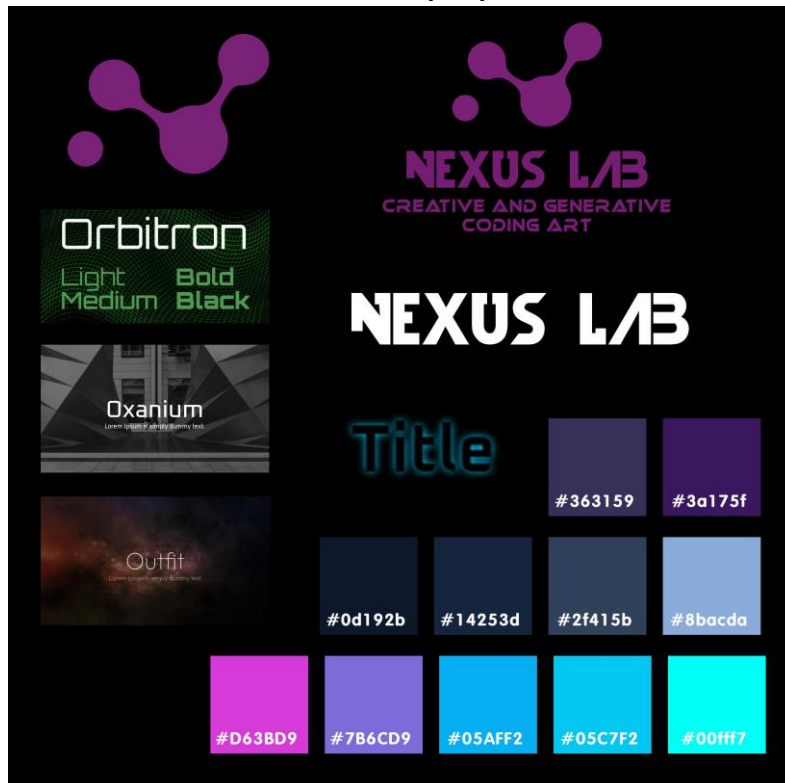


## 10. ANNEXES

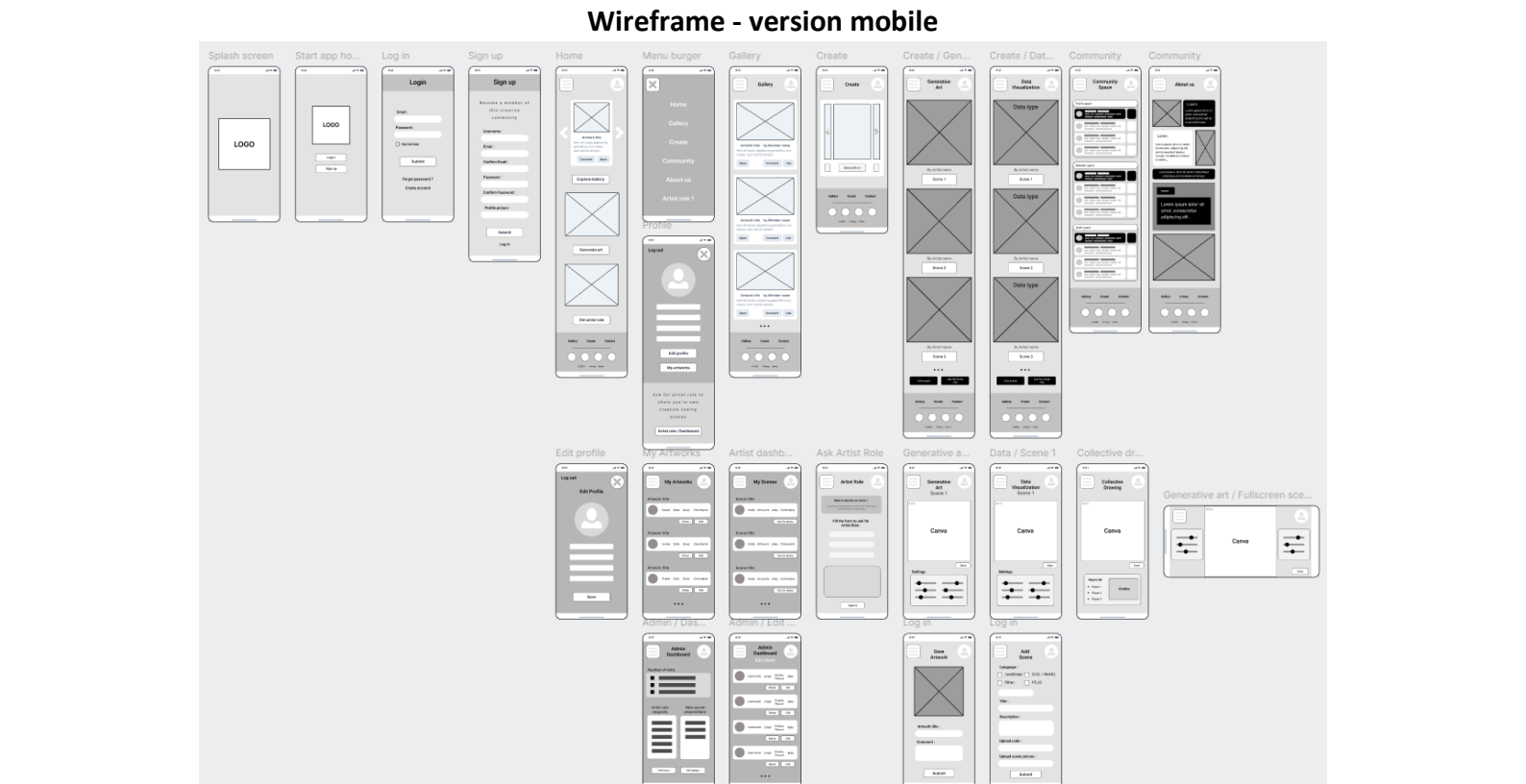
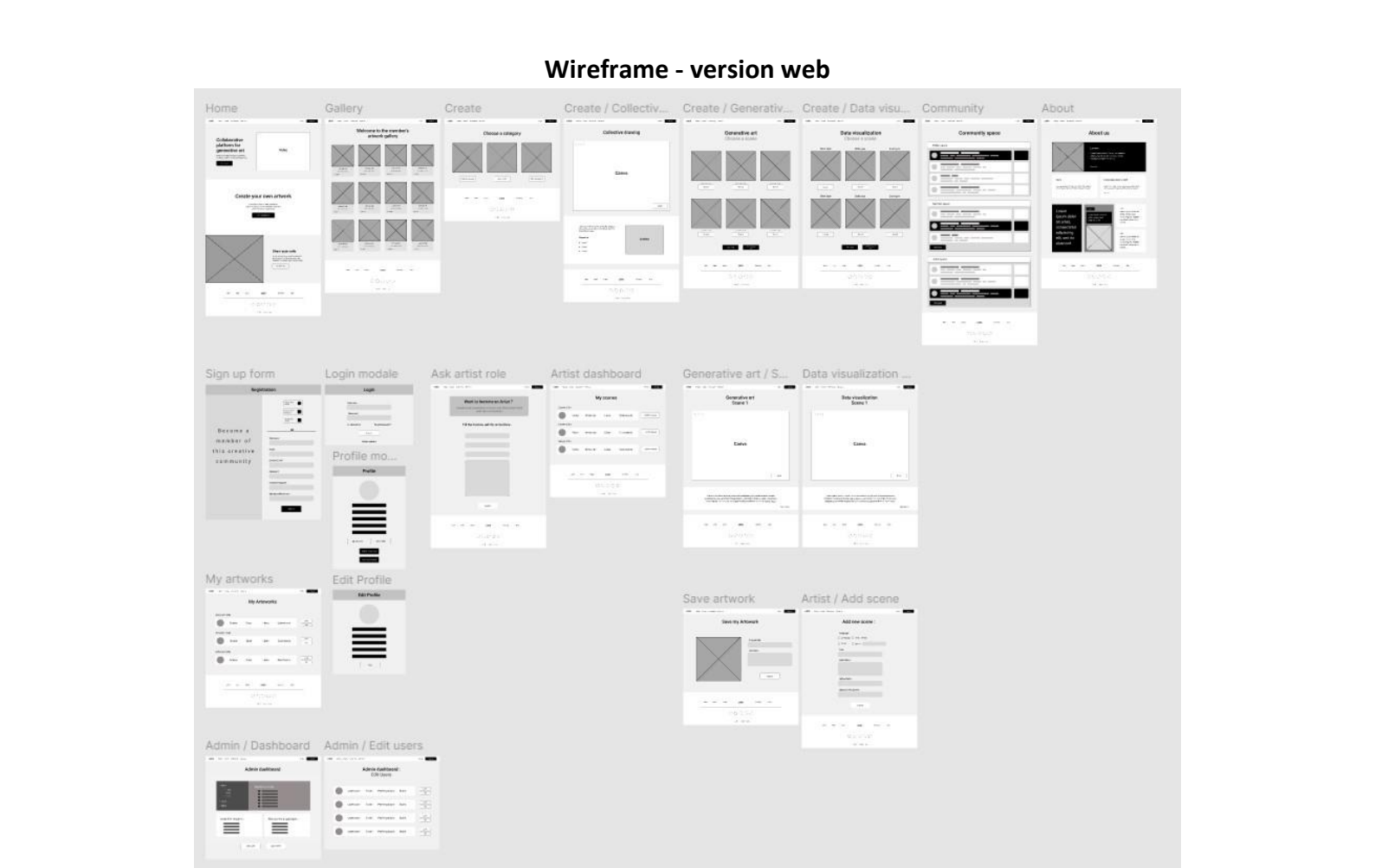
### Moodboard



### Charte Graphique

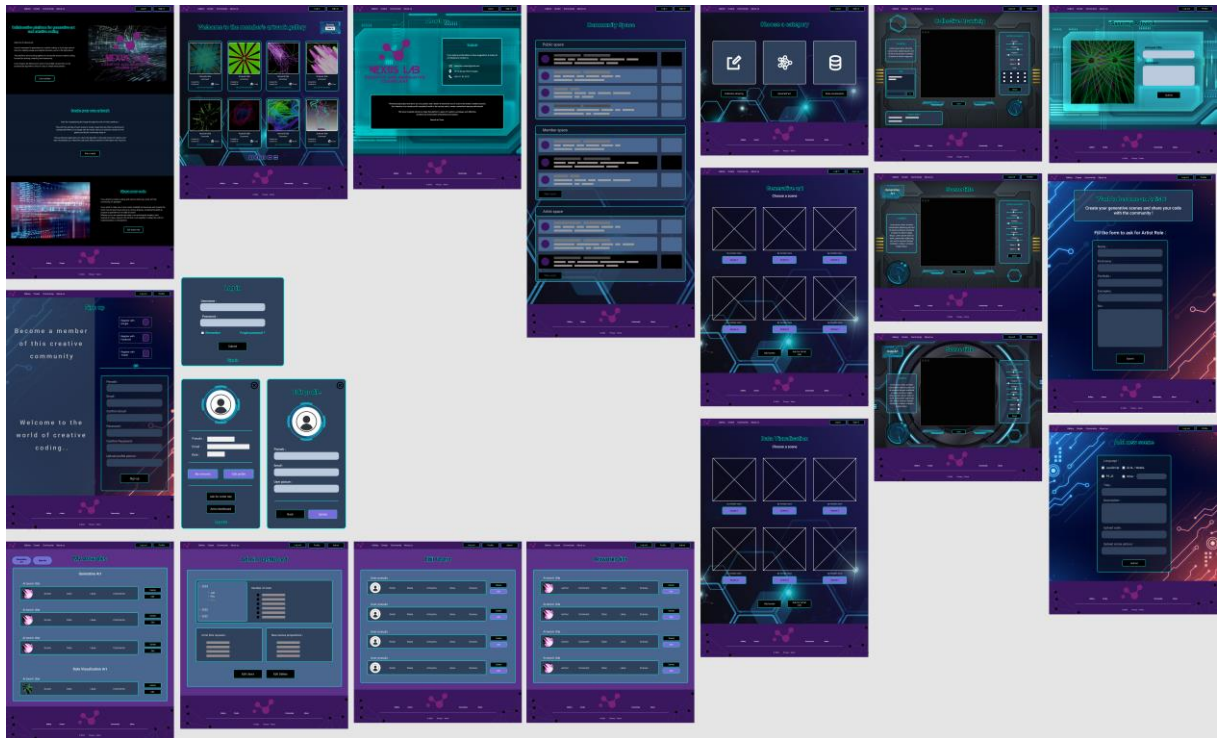








## Maquette - version web



## Maquette - version mobile

