

Sketch Helper

Elisa LI

Nathan NYABANGANG JOKÉ
TIMBA

William SARDON ARRAZ

1. Présentation générale du projet

1.1. Contexte

De nos jours, la création d'un logo est un élément essentiel pour toute entreprise, association ou organisme indépendant, car elle permet de construire une identité visuelle forte et reconnaissable, sur laquelle s'appuyer pour communiquer. Cependant, concevoir un logo de qualité n'est pas accessible à tous, notamment aux personnes débutantes en dessin ou ne disposant pas de compétences en design graphique.

C'est dans ce contexte que nous souhaitons proposer une application intelligente de création de logo, capable d'accompagner les utilisateurs dans la conception d'une identité visuelle professionnelle, simplement et efficacement.

Ce projet s'inscrit dans le cadre du cours d'Interaction Humain-IA (HAII), dont l'objectif principal est de concevoir des systèmes d'assistance permettant d'améliorer la performance globale de l'équipe humain + IA, plutôt que celle de l'IA seule. Dans cette perspective, l'application développée ne vise pas à automatiser entièrement la création graphique, mais à proposer une assistance partielle et contrôlée, intégrée dans un outil de dessin interactif.

1.2. Objectif du projet

1.2.1. Objectif technique

L'objectif principal de ce projet est de concevoir un prototype fonctionnel d'une application graphique d'aide à la création de dessins et de logos, intégrant des fonctionnalités classiques de dessin ainsi qu'un assistant de suggestions automatiques inspiré des outils d'intelligence artificielle.

1.2.2. Objectif expérimental et conceptuel

Au-delà de l'objectif technique de l'application de dessin, ce projet poursuit un objectif expérimental et conceptuel : évaluer dans quelle mesure une assistance intelligente peut améliorer la performance de l'utilisateur sans introduire de surconfiance ou de dépendance excessive de l'IA.

Conformément aux principes vus en cours, la performance n'est pas évaluée du point de vue de l'IA seule, mais du point de vue de l'équipe humain + IA. L'objectif est donc de comparer des situations où l'utilisateur travaille seul à des situations où il est assisté, afin de mesurer un éventuel gain en efficacité, notamment en termes de temps de réalisation et de nombre d'actions nécessaires.

Enfin, le projet vise à mettre en pratique les notions d'Appropriate Reliance, en observant si les utilisateurs sont capables d'utiliser l'assistant lorsque celui-ci est utile, tout en conservant la capacité de s'en écarter lorsqu'ils estiment que les suggestions ne correspondent pas à leurs intentions.

2. État de l'art et positionnement du projet

Il est pertinent de situer le projet par rapport aux outils existants. De nombreux logiciels professionnels tels qu'Adobe Illustrator, Figma ou Inkscape proposent déjà des fonctionnalités avancées de dessin vectoriel. Plus récemment, certains outils intègrent des mécanismes

d'assistance basés sur l'intelligence artificielle, capables de suggérer des formes, des couleurs ou des compositions.

Cependant, ces logiciels sont souvent complexes et peu accessibles pour des utilisateurs débutants. Leur richesse fonctionnelle peut constituer un frein à l'apprentissage et à la créativité. Le projet présenté ici se positionne volontairement à l'opposé de cette logique : il s'agit d'un prototype pédagogique, centré sur la simplicité, la compréhension des mécanismes internes et l'étude de l'impact de l'assistance sur l'utilisateur.

Ainsi, l'objectif n'est pas de concurrencer des solutions professionnelles, mais d'explorer, à petite échelle, comment une assistance automatisée peut influencer le processus créatif et la performance, dans un environnement contrôlé et volontairement limité.

3. Analyse des besoins et des utilisateurs

3.1. Public cible

L'analyse des besoins des utilisateurs a été menée en tenant compte de la diversité des profils et des niveaux d'expertise. Les utilisateurs visés par l'application sont des personnes débutantes ou intermédiaires en création graphique, pour lesquelles une assistance peut représenter un gain significatif, mais aussi un risque potentiel de dépendance.

En effet, l'expertise de l'utilisateur constitue une variable confondante majeure dans l'évaluation des systèmes d'assistance. Un utilisateur expérimenté peut percevoir l'IA comme intrusive ou inutile, tandis qu'un utilisateur novice peut avoir tendance à lui accorder une confiance excessive. Ces différences justifient la mise en place de mécanismes permettant à chacun d'adapter son niveau d'assistance, notamment via l'activation ou la désactivation des suggestions automatiques de l'IA.

3.2. Besoins fonctionnels

Les besoins identifiés sont les suivants :

- Pouvoir dessiner facilement des formes simples (ellipse, rectangle, trait etc.)
- Modifier et manipuler les éléments dessinés
- Être aidé lors de la création sans pour autant perdre le contrôle
- Pouvoir sauvegarder et reprendre son travail
- Comprendre rapidement le fonctionnement de l'application

4. Principes généraux de l'application

L'application est structurée autour de plusieurs composants :

- Un script principal (main.py) responsable du lancement
- Une fenêtre d'accueil
- Une fenêtre de dessin principale
- Des widgets spécialisés (toolbar, assistant, panneau IA)

4.1. Menu principal

Au lancement de l'application, une fenêtre s'ouvre :

- permettant de créer un nouveau dessin.
- présentant les fonctionnalités principales de l'application
- expliquant le rôle de l'assistant IA
- disant ce que le système fait et ne fait pas (capacités/ limites et quelques exemples)

4.2. Fenêtre de création

Depuis le menu principal, on peut accéder à la fenêtre de création de logo. Elle constitue le cœur de l'application et contient :

4.2.1. Une barre d'outils (Toolbar)

Les actions proposées sont :

- Dessin libre (stylo)
- Gomme
- Trait
- Rectangle
- Ellipse
- Triangle
- Remplissage de couleur
- Annuler / Rétablir
- Copier / Coller / Couper / Dupliquer
- Sélection d'objets
- Activation / désactivation des suggestions automatiques de l'IA
- Affichage / masquage de l'assistant
- Bouton de d'assistance IA

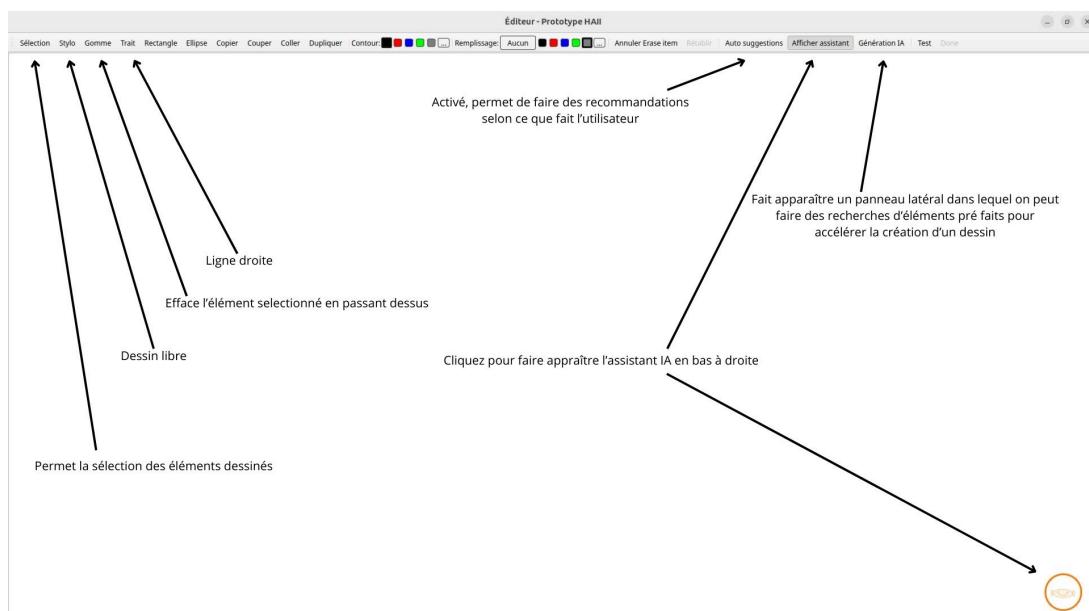


Fig. 1. – Différents outils de l'application

4.2.2. Une zone de dessin

- Implémentée à l'aide du framework QGraphics de PySide6
- Chaque élément dessiné est un objet graphique (QGraphicsItem)
- Support du clic, sélection et transformation des objets

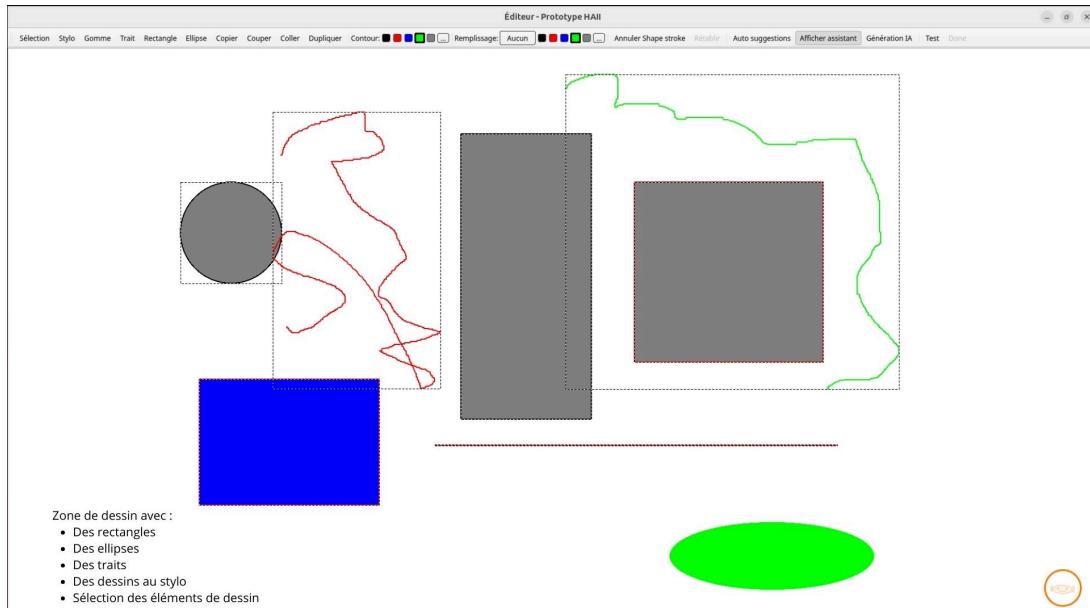


Fig. 2. – Zone de dessin de l'application

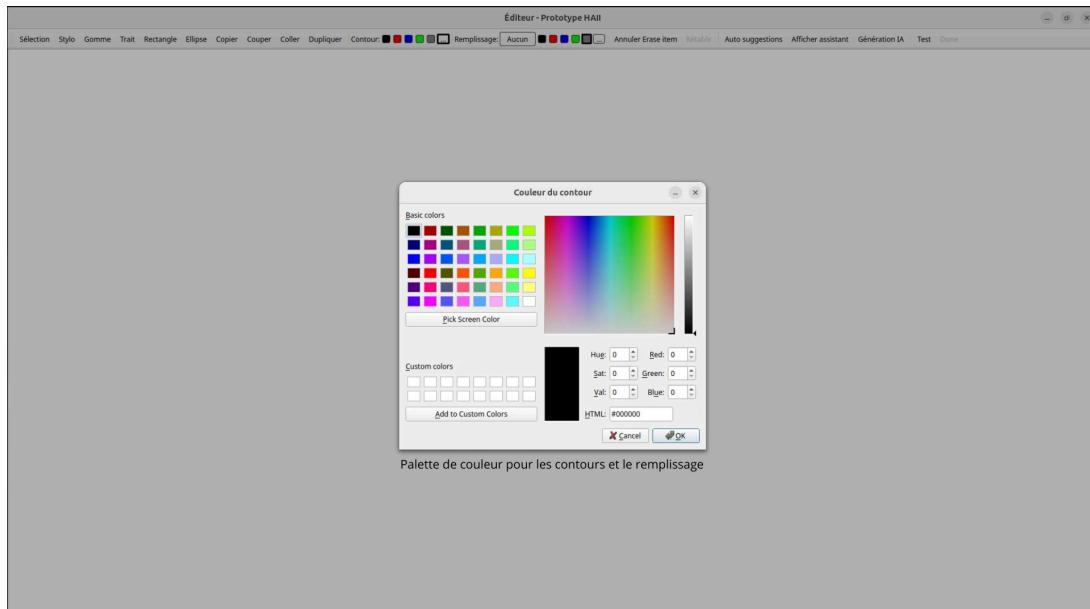


Fig. 3. – Palette de couleur pour le remplissage et le contour des éléments graphiques

4.2.3. Assistant IA

Il y a **3 types** d'assistance IA :

- Un panneau latéral de génération IA en appuyant sur « Génération IA »(recherche d'éléments pré faits pour faciliter la réalisation de dessin dans le cadre du prototype)
- Un bouton en bas à droite sur lequel appuyé pour invoquer l'IA (visible en activant « Afficher assistant »)
- Les suggestions automatiques qu'on peut activer directement depuis la toolbar en cliquant sur « Auto suggestion » et qui suggère des éléments pertinents pour le dessin de l'utilisateur

4.2.3.1. Panneau latéral de génération IA

- S'ouvre via un bouton de la Toolbar (« génération IA »)
- Contient un menu déroulant avec 3 catégories :
 - Tour
 - Porte

- Mur
- Chaque catégorie propose des suggestions graphiques simples (rectangles, ellipses, couleurs différentes)
- Ajout par double clic

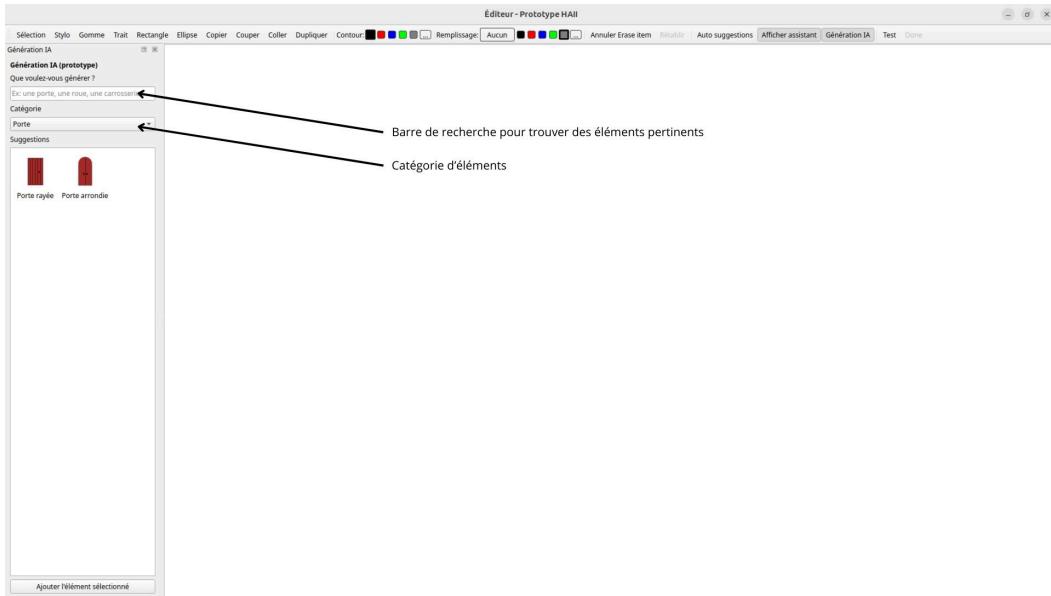


Fig. 4. – Panneau latéral de génération d'éléments IA avec barre de recherche et catégorie

4.2.3.2. Bouton assistant IA

- Widget dédié (en bas à droite)
- L'utilisateur peut appuyer dessus pour qu'il fasse une recommandation sur l'objet courant

4.2.3.3. Suggestion automatique

- Lorsqu'elle est activée (via la toolbar « auto suggestion ») :
 - propose automatiquement des éléments prédéfinis
 - affiche une boîte de dialogue permettant à l'utilisateur d'accepter, d'ignorer ou de refuser la suggestion de l'IA

Lorsque l'IA est active et recommande une suggestion, celle-ci affiche explicitement son niveau d'incertitude (70%, 65%). Chaque suggestion inclut 3 raisons claires (« Pourquoi ? ») avec une justification basée sur l'analyse de la scène. L'utilisateur comprend alors que l'IA n'est pas infaillible. De plus, l'IA affiche une prévisualisation fantôme de suggestion avant validation, donc il n'y a pas de modification permanente sans accord de l'utilisateur.

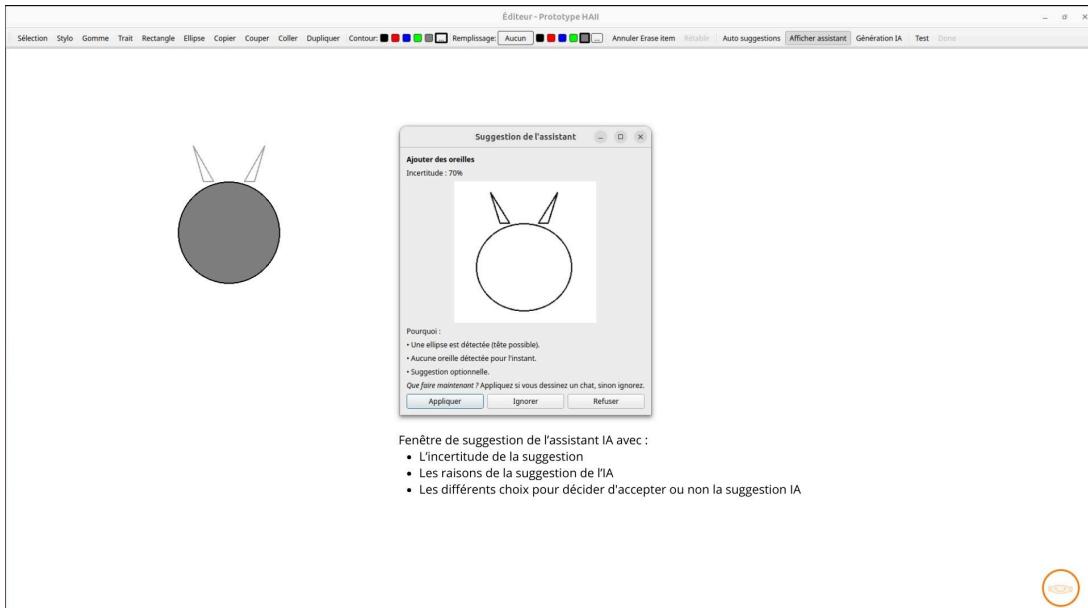


Fig. 5. – Pop up de suggestion de l'IA avec raison de suggestion et incertitude

La barre d'outils regroupe l'ensemble des actions manuelles de dessin, ce qui permet de maintenir une séparation claire entre les capacités de l'utilisateur et celles de l'assistant. Cette séparation est volontaire et répond aux recommandations HAI visant à éviter le "mode confusion", dans lequel l'utilisateur ne sait plus si une action résulte de son propre choix ou d'une automatisation par l'IA.

L'utilisateur conserve ainsi un contrôle total sur les opérations de base (dessin, sélection, modification), tandis que l'assistant intervient sous forme de propositions (désactivable à tout moment) ou bouton activable depuis la toolbar, sans jamais se substituer aux outils classiques.

4.3. Mode test de performance

- L'utilisateur doit reproduire un dessin cible affiché
- Le chronomètre démarre au début
- Le temps s'arrête lorsque le dessin est jugé suffisamment similaire
- Deux modes :
 - avec assistance IA
 - sans assistance IA

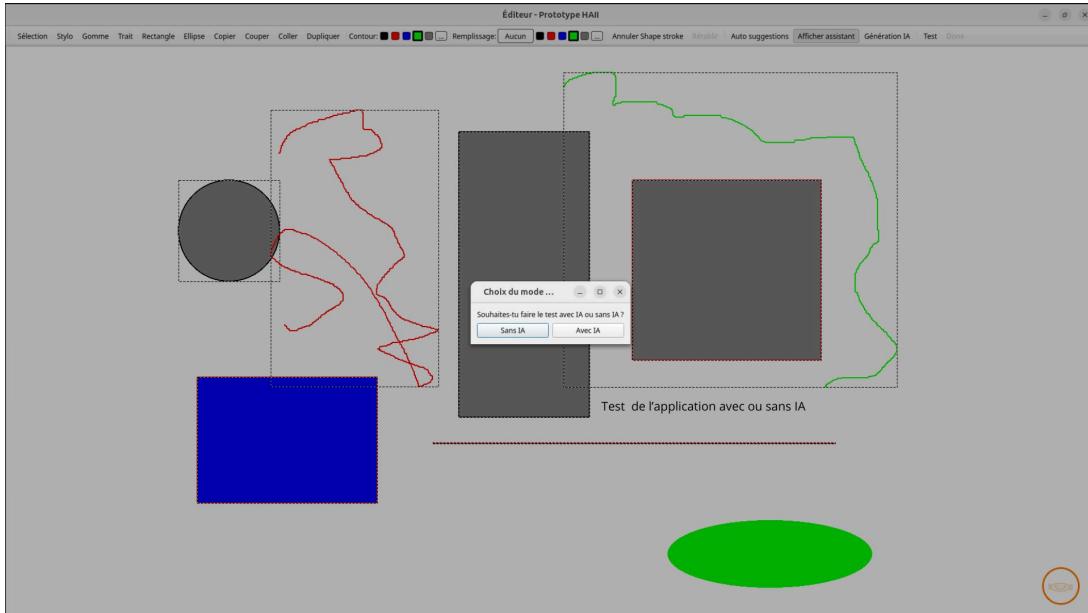


Fig. 6. – Mode test : Permet d'entrer en mode avec ou sans IA pour commencer à log l'expérience

5. Choix d'implémentation et justification

5.1. Choix technologiques

- **Python** : Langage de programmation accessible et polyvalent
- **Pyside6** : Bibliothèque robuste et bien documentée (déjà utilisé dans l'UE IHM en M1)
- **QGraphics** : Adapté pour notre application de dessin interactif

5.2. Justification de l'IA simulée

Notre objectif est d'étudier l'interaction humain-IA et de prouver que la performance IA + Humain est meilleure que IA tout seul ou Humain tout seul (Team Uplift). L'IA de notre système nous permet:

- Un contrôle total sur les suggestions (les accepter, les ignorer ou les refuser)
- De gagner du temps en proposant des éléments pertinents selon ce qu'on fait sur la zone de dessin

6. Protocole expérimental

6.1. Objectif du protocole

Évaluer l'impact de l'assistance IA sur la performance et l'efficacité des utilisateurs lors des tâches de reproduction graphique.

6.2. Déroulement de l'expérience

Chaque participant :

1. Réalise une série de dessins sans IA
2. Réalise les mêmes dessins avec IA
3. Les temps et actions sont enregistrés automatiquement à chaque session

6.3. Variables étudiées

- Présence de l'IA ou non
- Temps
- Nombres d'actions fait pour reproduire le dessin
- Taux d'acceptation des suggestions fait par l'IA

Le choix d'un protocole comparatif avec et sans IA permet d'estimer un éventuel uplift de performance. L'objectif n'est pas de démontrer que l'IA est toujours bénéfique, mais d'identifier dans quelles conditions elle apporte un gain mesurable, et pour quels types d'utilisateurs.

7. Données collectées

7.1. Nature des données

- Temps de réalisation du logo demandé
- Nombre d'actions utilisateur
- Mode utilisé (avec/sans IA)
- Traçabilité de toutes les invocations (manual/auto)
- Enregistrement des réponses (accept/ignore/refuse)

7.2. Format et stockage

- Format CSV pour logger les actions de l'utilisateur
- Données exploitables dans un notebook

8. Hypothèses de recherche

9. Résultats

10. Aspects éthiques et limites de l'assistance IA

L'intégration d'un assistant IA soulève également des questions éthiques, même dans un cadre expérimental. L'une des principales interrogations concerne la dépendance potentielle de l'utilisateur à l'assistance automatisée. Si l'IA devient trop performante ou trop présente, l'utilisateur risque de perdre une partie de ses compétences ou de sa créativité en faisant trop confiance à l'IA.

Dans le cadre de ce projet, cette problématique est volontairement limitée par le caractère simple et non adaptatif de l'IA. Toutefois, elle est discutée dans l'analyse des résultats, notamment lorsque certains utilisateurs préfèrent s'appuyer fortement sur les suggestions proposées.

11. Conclusion

12. Évolution future

Bien que le projet soit conçu comme un prototype, son architecture a été pensée pour être évolutive. L'utilisation de modules distincts pour l'interface, la logique de dessin et l'assistant permet d'envisager des extensions futures sans remettre en cause l'ensemble du code.

Par exemple, il serait possible d'ajouter de nouvelles catégories de suggestions, d'intégrer un véritable modèle d'intelligence artificielle ou encore de proposer un mode collaboratif.