

Projet trading

Créer un bot qui peut trader en autonomie

I. Contexte et objectif.....	1
II. Stack technique.....	1
III. Résultat.....	2
IV. Conclusion.....	3

I. Contexte et objectif

Contexte: En avril 2025, après avoir consacré mes projets précédents à l'univers du jeu vidéo, j'ai souhaité explorer un domaine radicalement différent. Le trading ne m'attirait pas pour ses aspects financiers car je n'ai aucune prétention à être rentable sur les marchés, ni la compétence nécessaire pour cela. En revanche, ce domaine m'a offert un excellent terrain d'expérimentation pour connecter et faire interagir des technologies variées. Ce projet m'a permis de sortir de ma zone de confort en m'attaquant à un système complexe, tout en me concentrant sur ce que j'aime : l'architecture logicielle, la communication entre langages, et l'intégration de composants techniques avancés.

Objectif: concevoir un bot capable de trader automatiquement la paire EUR/USD et de tester différentes stratégies de trading, des plus simples aux plus complexes. Un point central du projet est la possibilité de backtester ces stratégies sur des données historiques étendues, couvrant la période de 2002 à aujourd'hui, afin d'analyser leurs performances dans des contextes de marché variés.

Le système inclura également une gestion des positions après l'entrée en trade, permettant d'explorer des logiques de suivi, de prise de profit ou de limitation des pertes.

II. Stack technique

Pour ce projet, j'ai choisi de combiner plusieurs technologies afin d'optimiser les performances et la modularité. J'utilise Python comme interface principale pour communiquer avec les marchés via l'API de MetaTrader 5 (MT5). Les calculs de backtests ainsi que les décisions de trading sont réalisés en C++, ce qui garantit une exécution rapide. Python joue ici un rôle d'intermédiaire entre le moteur de calcul en C++ et la plateforme de trading MT5.

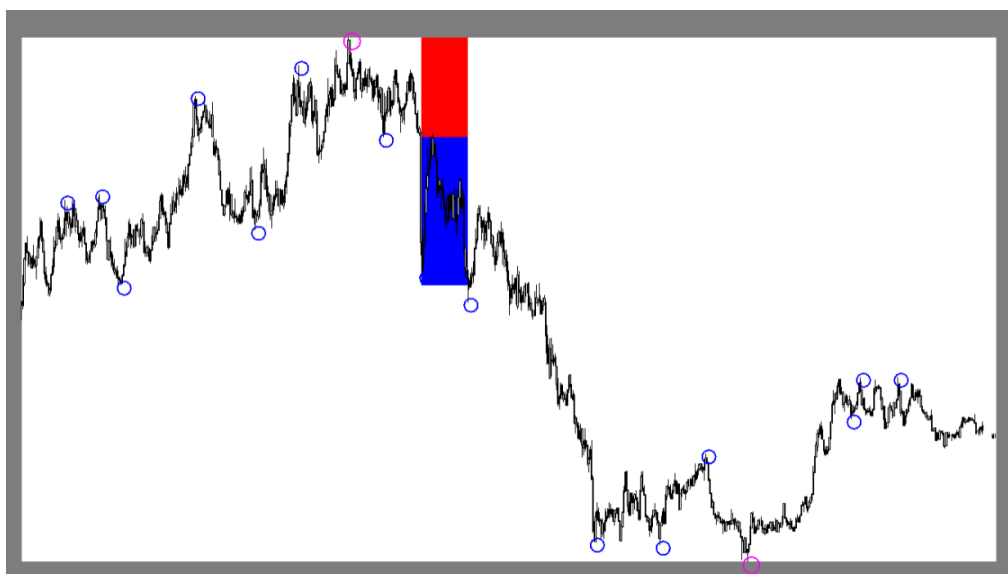
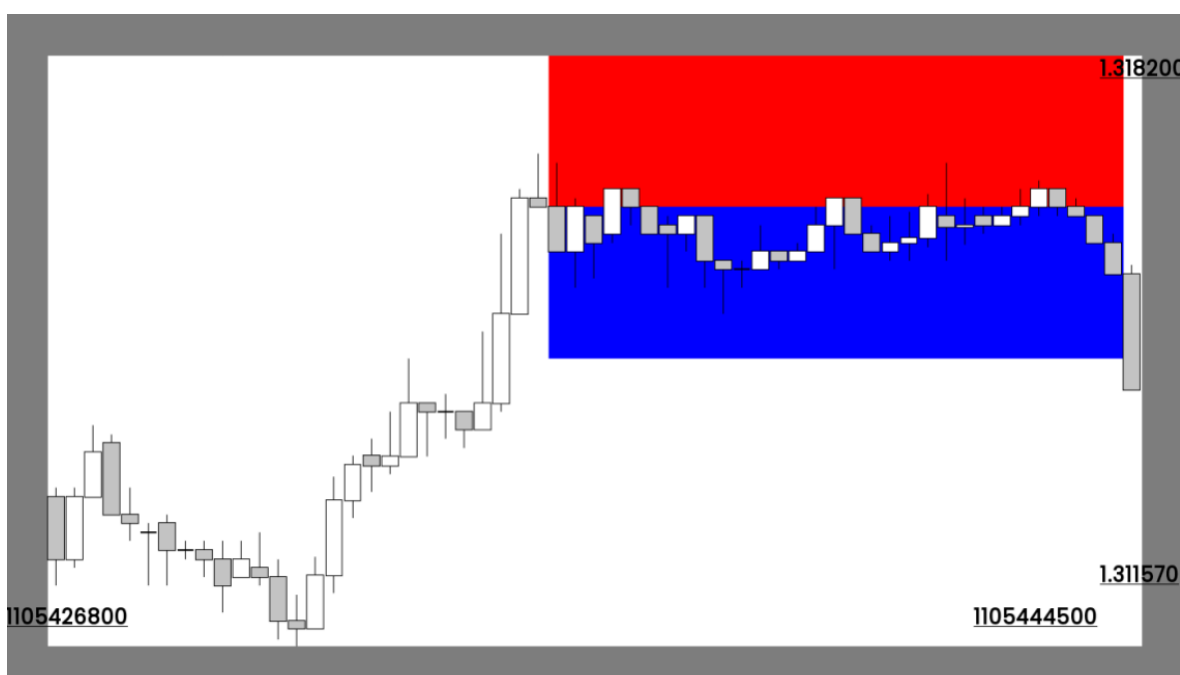
J'intègre également Dlib, une bibliothèque C++, pour concevoir et entraîner des réseaux de neurones. Enfin, j'utilise SFML pour visualiser graphiquement les résultats des backtests, notamment l'évolution des bougies et les points d'entrée et de sortie des trades.

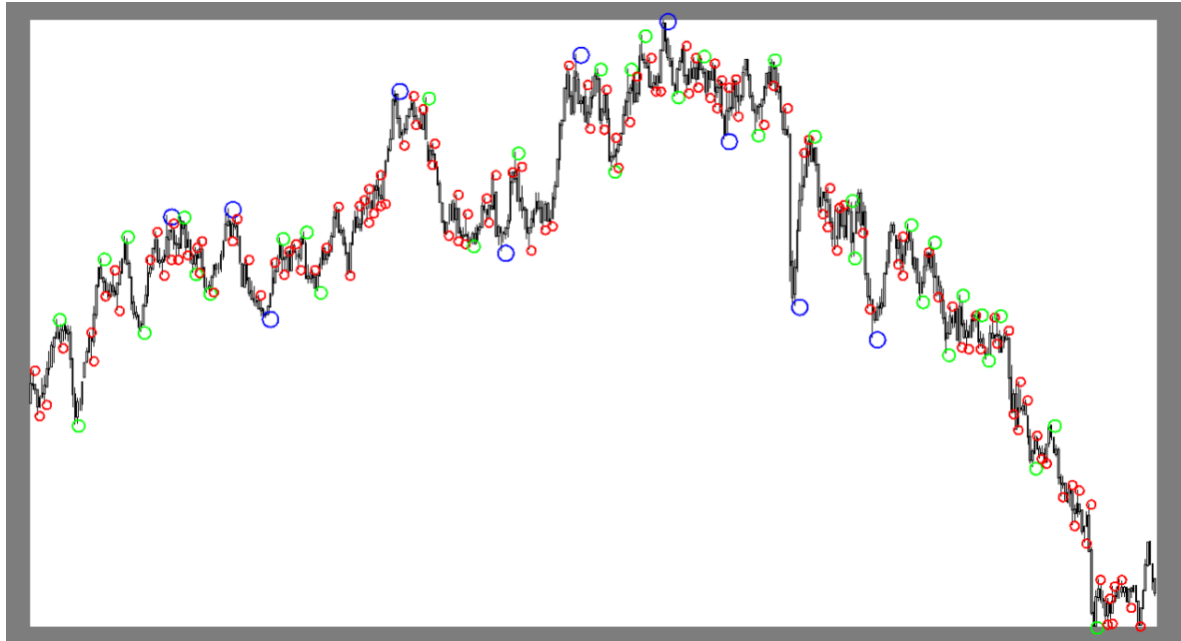
III. Résultat

Le système que j'ai développé me permet aujourd'hui de définir et exécuter des stratégies de trading personnalisées basées sur une combinaison d'indicateurs techniques, de motifs récurrents (patterns) ou d'événements spécifiques, exclusivement sur la paire EUR/USD. Les stratégies peuvent s'appuyer sur plusieurs timeframes simultanément, offrant ainsi une analyse multi-échelle.

Un des aspects clés de l'architecture est la modularité de l'objet Strategy, identique pour le backtest et le trading en temps réel. Cela me permet de passer de l'historique au réel sans dupliquer la logique métier, en gardant un code cohérent et facilement testable.

Voici quelques screens de trades issus de mes backtest.





IV. Conclusion

Ce projet est toujours en cours de développement. Mon objectif initial, qui était de sortir du domaine du jeu vidéo pour explorer une nouvelle approche technique, est pleinement atteint. Aujourd'hui, je continue à améliorer et affiner la solution, notamment en travaillant sur l'optimisation du code C++ pour accélérer les backtests et mieux exploiter les ressources de la machine.

Ce travail m'a permis de progresser sur un point essentiel : l'interconnexion de différentes technologies. Je me sens désormais plus à l'aise pour concevoir des architectures multi-langages, capables de dialoguer efficacement entre elles.