**Rapport Rendu n°1 :**

**Rapport sur les Prompts Clés Liés à la Visualisation du Graphe en C#**

**Introduction**

**Ce rapport met en avant les prompts les plus pertinents concernant la visualisation du graphe en C#. Il détaille les problématiques rencontrées et les solutions mises en place pour améliorer l'affichage et l'interaction utilisateur sous Windows Forms.**

**1. Implémentation de la Visualisation du Graphe**

**Prompt Clé :**

*"Passons directement à l'implémentation de la visualisation du graphe. Attention, j'aurai sûrement besoin d'explications très détaillées et d'être guidé."*

**Problématique et Solution :**

* Création d'une interface graphique Windows Forms.
* Utilisation de System.Drawing pour représenter les nœuds sous forme de cercles et les liens sous forme de lignes.
* Ajout d'un événement Paint pour rafraîchir l'affichage à chaque mise à jour du graphe.

**2. Amélioration de la Disposition des Nœuds**

**Prompt Clé :**

*"Comment optimiser un maximum la répartition des nœuds et comment avoir un affichage agréable à regarder ? "*

**Problématique et Solution :**

* Vérification de la lecture des données et correction des erreurs d'instanciation des liens.
* Mise en place de OrganiserEnCercle() pour répartir les nœuds de manière uniforme autour d'un point central.
* Amélioration de l'affichage des liaisons incorrectes en mettant à jour les références des liens après le repositionnement des nœuds.

**3. Ajout d'Interactions avec des Boutons**

**Prompt Clé :**

*"Nous avons maintenant un code fonctionnel auquel j'aimerais ajouter des boutons sur la visualisation Windows Forms sur lesquels je pourrai cliquer ou survoler et qui me diraient si le graphe est connexe ou s'il contient des circuits."*

**Problématique et Solution :**

* Ajout de boutons interactifs (Button) pour tester la connectivité et la présence de cycles.
* Affichage des résultats dans un Label dynamique pour informer l'utilisateur.
* Liaison des boutons avec EstConnexe() et ContientCycle() via des EventHandler.

**4. Débogage sous Windows Forms**

**Prompt Clé :**

*"Je suis sur Windows Forms et je n'ai pas accès à la console, comment voir les logs ?"*

**Problématique et Solution :**

* **Proposition de quatre méthodes pour afficher les logs :** 
  1. MessageBox.Show() pour afficher des pop-ups informatives.
  2. Ajout d’un Label sur l’interface pour afficher les dernières informations en temps réel.
  3. Écriture des logs dans un fichier debug\_log.txt pour une analyse après exécution.
  4. Utilisation de Debug.WriteLine() pour afficher les logs dans l’onglet Sortie de Visual Studio.

**Conclusion et Améliorations Futures**

**La mise en place d'une visualisation efficace du graphe a permis d'améliorer la compréhension et l'interactivité du projet. Les améliorations possibles incluent :**

* L’ajout d’une interface plus dynamique permettant de déplacer les nœuds.
* Une meilleure gestion des superpositions pour éviter le chevauchement des liens.
* L’intégration d’une bibliothèque de visualisation plus avancée comme SkiaSharp ou GraphSharp.

**Rapport sur les Tests Uniatires**

**Introduction**

L'objectif des tests unitaires est de s'assurer du bon fonctionnement des différents méthodes de la classe Graphe. Ces méthodes permettent respectivement d'ajouter des nœuds et des liens entre eux dans un graphe.

Les Test sur les méthodes :

**Difficulté rencontrée**

* Une erreur Une référence d'objet est requise a été détectée, indiquant que graphe n'était pas correctement instancié.
* Correction en s'assurant que l'objet Graphe est bien initialisé avant chaque test.

Une fois que cette difficulté a été surmonté, les tests ont très bien fonctionné et nous ont permis de bel et bien vérifier que l’ajout des nœuds et des liens se faisaient effectivement bien.

**Conclusion**

Les tests réalisés permettent de valider le bon fonctionnement de différentes méthodes de la classe Graphe. Malgré quelques erreurs initiales liées à l’instanciation de graphe, les tests ont été corrigés et permettent d’assurer une meilleure fiabilité du code.