

# Critères pour l'évaluation du rendu

## Conception du projet

- 1.1 Architecture du projet : L'architecture du projet est-elle bien conçue et modulaire ?
- 1.2 Utilisation des structures de données : Les structures de données utilisées sont-elles adaptées aux besoins du projet ?
- 1.3 Gestion de la complexité : Le projet gère-t-il efficacement la complexité grâce à une conception bien pensée ?

## Qualité du Code

- 2.4 Clarté et lisibilité : Le code est-il clair, bien indenté et facile à comprendre ?
- 2.5 Conventions de codage : Le code suit-il les conventions de codage C++ standards ?
- 2.6 Utilisation des fonctionnalités avancées : Le code utilise-t-il efficacement les fonctionnalités avancées du langage C++ (par exemple, les templates, la programmation orientée objet, les lambdas, etc.) ?
- 2.7 Compilation du code : Le code compile ? sans erreurs ? sans warning ? avec plusieurs compilateurs ?

## Performance

- 3.8 Execution : le code s'exécute sans erreur
- 3.9 Optimisation : Le code est-il optimisé pour les performances, en évitant les opérations coûteuses et en utilisant efficacement la mémoire ?
- 3.10 Temps d'exécution : Le projet s'exécute-t-il rapidement et de manière efficace ?
- 3.11 Utilisation des ressources : Le projet utilise-t-il les ressources matérielles de manière efficace ?

## Fonctionnalités Implémentées

- 4.12 Complétude des fonctionnalités : Les fonctionnalités spécifiées ont-elles été implémentées correctement et de manière exhaustive ?
- 4.13 Robustesse : Le projet gère-t-il correctement les erreurs et les cas limites ?

## Documentation

- 5.14 Commentaires dans le code : Le code est-il bien documenté avec des commentaires clairs expliquant la logique derrière chaque section de code ?
- 5.15 Documentation utilisateur : Une documentation utilisateur est-elle fournie, expliquant comment utiliser le projet et ses fonctionnalités ?

## Gestion de Version

- 6.16 Utilisation d'un système de contrôle de version : Le projet est-il géré à l'aide d'un système de contrôle de version tel que Git ?

## Tests

- 7.17 Couverture des tests : Le projet est-il accompagné d'une suite de tests exhaustive qui couvre toutes les fonctionnalités principales ?
- 7.18 Automatisation des tests : Les tests sont-ils automatisés pour faciliter la validation continue ?