

**Département de génie logiciel et des T.I.**

Rapport de Laboratoire

|  |  |
| --- | --- |
| **Numéro du laboratoire** | Laboratoire 3 |
| **Nom du laboratoire** | Primitives, remplissage et composition d’image |
| **Étudiant(s)** |  |
| **Code(s) permanent(s)** |  |
| **Numéro d’équipe** |  |
| **Cours** | GTI 411 |
| **Session** |  |
| **Groupe** |  |
| **Chargé(e) de laboratoire** |  |
| **Date** |  |

Table des matières

[Introduction 1](#_Toc191643398)

[Outils et concepts 1](#_Toc191643399)

[Implémentation 1](#_Toc191643400)

[Résultats et discussion 1](#_Toc191643401)

[Conclusion 2](#_Toc191643402)

[Références 2](#_Toc191643403)

[Annexe : Manuel d’utilisateur 2](#_Toc191643404)

# Introduction

* Introduisez ***LE CONTEXTE*** du laboratoire, quelles applications pouvons-nous avoir avec les fonctionnalités à développer dans ce laboratoire?
* Résumez ensuite les problématiques du laboratoire sans faire un copier-coller
* Présentez les sections à venir en un paragraphe.

La manipulation de primitives géométriques telles que les lignes, cercles, rectangles et ellipses est un élément fondamental en imagerie numérique et en traitement d'images. Ces techniques sont utilisées dans des domaines variés, tels que la détection automatisée de véhicules, la reconnaissance faciale (exemple FaceID), ou encore dans les jeux vidéo où l'interaction avec des objets 2D et 3D repose sur la représentation de formes géométriques. En outre, la capacité à remplir ces formes (comme les techniques de *boundary fill* et *flood fill*) est essentielle pour des applications comme l'édition d'images ou la création de graphiques interactifs. Par conséquent, ce laboratoire a pour objectif d'apprendre à implémenter ces manipulations fondamentales en utilisant des méthodes algorithmique et des outils de programmation, en se concentrant sur les bases sans recourir à des librairies spécialisées comme OpenCV.

Les problématiques abordées dans ce laboratoire sont multiples. Il s'agit, dans un premier temps, de comprendre et d'implémenter les algorithmes nécessaires pour dessiner des primitives géométriques, en prenant en charge leur affichage et leur personnalisation en termes de couleur et de forme. Ensuite, il sera nécessaire de développer des méthodes de remplissage automatique pour ces primitives, en intégrant les techniques de *boundary fill* et *flood fill*, qui sont au cœur de nombreuses applications de traitement d'images. Enfin, la troisième partie du laboratoire se concentre sur l'extraction et la séparation d'une silhouette humaine d'un fond en temps réel, une tâche couramment rencontrée dans les applications de visioconférence ou de réalité augmentée.

Le laboratoire est structuré en trois parties. La première se focalise sur l'implémentation des primitives géométriques (lignes, cercles, rectangles et ellipses) ainsi que leur gestion à travers une interface graphique simple. La deuxième partie, quant à elle, vous entraînera à implémenter les algorithmes de remplissage *boundary fill* et *flood fill*. La troisième et dernière partie introduira la manipulation d'images en temps réel, avec l'implémentation d'un système permettant de retirer le fond d'une vidéo en direct, une fonctionnalité utilisée par des applications comme Zoom ou Teams.

# Outils et concepts

* Pourquoi les primitives sont-elles importantes en computer graphics? (Rasterization etc.)
* Donner quelques (2 ou 3) exemples dans lesquelles la manipulation des primitives (lignes, rectangles etc.) est utile (jeux vidéos, traitement d'image etc.)

# Implémentation

* Présenter très succinctement (quelques lignes, pas de pseudo-code cette fois pour cette partie) les algorithmes de tracé de primitives utilisés et de Fill
* Présenter votre approche **(partie à bien détailler)** pour la partie 3  
  Pas de copier/coller du code : diagramme ou formules, pseudo-code autorisé mais très haut niveau (pas 3 pages de pseudo-code)
* Penser à lister les librairies utilisées et leur rôle

# Résultats et discussion

* Afficher les résultats sous forme de captures d’écrans et commentez les au besoin (si des parties ne fonctionne pas etc.)
* Expliquer les éventuelles difficultés rencontrées, comment vous avez fait pour les résoudre ou comment vous feriez si vous n’avez pas eu le temps de finir
* Proposer d’éventuelles améliorations / fonctionnalités supplémentaires au cahier des charges notamment pour la partie 3

# Conclusion

* Résumer en un paragraphe ce que vous avez appris

# Références

* Penser à bien citer vos sources (algorithme, figure etc.) sous le format APA7 (voir le site de la bibliothèque de l’ETS : <https://bibliotheque.etsmtl.ca/Services/Bibliographies-et-citations/Citer-avec-le-style-APA-ETS>)

# Annexe : Manuel d’utilisateur

* Comment installer les dépendances nécessaires
* Comment exécuter votre code