

---

**MR ROLIN****2025**

# [HARDWARE] Le bottleneck graphique

---

## Objectifs

À la fin de ce cours, l'apprenant doit être capable de :

- Définir ce qu'est un bottleneck dans un pipeline graphique.
- Identifier les principaux types de goulets d'étranglement : CPU, GPU, mémoire et bande passante.
- Diagnostiquer un bottleneck à partir de symptômes et d'outils d'analyse.
- Proposer des actions concrètes d'optimisation adaptées au problème détecté.
- Comprendre l'impact du bottleneck sur l'expérience utilisateur et la fluidité du rendu.

## Introduction

Dans les systèmes graphiques, le rendu d'une image repose sur la collaboration de plusieurs composants : processeur, carte graphique, mémoire vive, mémoire vidéo et bus de communication. Si l'un de ces éléments fonctionne plus lentement que les autres ou atteint une limite technique, il devient le point bloquant du pipeline. On parle alors de bottleneck, ou goulot d'étranglement. Ce phénomène est essentiel à comprendre pour tous les métiers liés à l'infographie, au développement de jeux, au traitement d'images ou à la modélisation 3D, car il impacte directement la fluidité d'affichage et les performances globales.

## Définition du bottleneck graphique

Un bottleneck survient lorsqu'un composant matériel ou une étape du pipeline ne peut pas suivre le rythme des autres. Le flux de données s'accumule ou ralentit, provoquant une chute notable de performances. Ce n'est pas nécessairement le composant le moins performant qui crée le problème, mais celui qui atteint en premier sa limite.

## Emplacement des bottlenecks dans le pipeline

Le pipeline graphique implique plusieurs étapes : traitement CPU, envoi des commandes au GPU, exécution des shaders, gestion des textures, puis affichage sur l'écran. Chacune de ces phases peut être saturée, en fonction du type de charge : calculs de logique, traitement visuel, transferts mémoire ou accès aux ressources. Comprendre ce cheminement permet de cerner plus rapidement la source des ralentissements.

## Le bottleneck CPU

Le CPU peut devenir un point critique lorsqu'il doit gérer trop d'instructions en trop peu de temps. Cela arrive souvent dans les jeux ou applications avec beaucoup d'objets, de collisions, d'IA ou de scripts. Même si la carte graphique est puissante, elle peut rester en attente si le processeur n'alimente pas suffisamment vite la file de commandes. Les draw calls, par exemple, sont une cause fréquente de surcharge CPU.

## Le bottleneck GPU

Le GPU devient limitant lorsqu'il exécute des calculs graphiques lourds. Les shaders complexes, les résolutions élevées, l'anti-aliasing avancé ou les effets de post-traitement peuvent saturer les unités de calcul graphiques. Dans ce cas, le CPU est souvent sous-utilisé tandis que la carte graphique fonctionne en permanence à pleine capacité.

## Le bottleneck mémoire et bande passante

Les transferts de données entre RAM, VRAM et bus PCIe peuvent également devenir saturés. Une VRAM trop petite entraîne des chargements fréquents ou du swapping, ce qui provoque des saccades sévères. La bande passante joue un rôle critique lorsque de grandes textures ou des modèles lourds sont envoyés en continu vers le GPU.

## Symptômes courants

Les signes d'un bottleneck se traduisent en pratique par des chutes d'image par seconde, une latence perceptible, des micro-saccades ou des écarts importants entre l'utilisation CPU et GPU. Par exemple, un CPU à 100 % associé à un GPU peu sollicité montre un bottleneck CPU, tandis qu'un GPU saturé indique une limitation graphique.

## Méthodes de diagnostic

Les outils de profilage intégrés aux moteurs graphiques ou aux pilotes permettent d'examiner les performances en détail. L'analyse des draw calls, la mesure de l'utilisation des unités de calcul, et la modification de paramètres graphiques aident à localiser la source du ralentissement. Une bonne pratique consiste à modifier la résolution ou la qualité graphique : si le FPS change fortement, le GPU est probablement le goulot ; sinon, le problème est ailleurs.

## Stratégies d'optimisation

Une fois le bottleneck identifié, l'optimisation doit être ciblée. Côté CPU, on peut réduire le nombre d'objets actifs, simplifier la logique ou regrouper les draw calls. Côté GPU, alléger les shaders, réduire les textures ou ajuster les effets complexes peut être suffisant. Pour la mémoire, il est possible d'employer des mipmaps, des formats plus légers ou des stratégies de streaming.

## Cas pratiques

Un jeu qui reste fluide en basse résolution mais tourne difficilement en haute suggère un bottleneck GPU. À l'inverse, un GPU qui dépasse rarement 50 % d'utilisation alors que les performances sont faibles indique que le CPU limite le flux. Enfin, des messages d'erreur liés à la mémoire vidéo ou des saccades régulières témoignent d'un problème de VRAM ou de bande passante.

## Conclusion

Le bottleneck graphique est un concept essentiel pour comprendre et optimiser les performances d'un système de rendu. Identifier correctement le composant limitant permet d'éviter des optimisations inutiles et de concentrer les efforts sur le facteur réellement bloquant. Une bonne méthodologie de diagnostic, associée à des outils de profilage adaptés, permet d'améliorer sensiblement la fluidité et la qualité d'affichage. Pour approfondir, il est recommandé de pratiquer l'analyse sur des projets variés : scènes 3D, jeux simples, rendus complexes ou applications interactives, afin de développer un regard critique sur les performances.