

**LOG8715-TP3**

Quoc-Hao Tran (1972967)

Jeffrey Lavallée (1847777)

# Discussion

## Implémentation extrapolation

Pour l’extrapolation, on calcul simplement la latence entre le client et le serveur à chaque frame. Cette latence est utilisée pour calculer le nombre de frame que le client est en retard comparé à serveur. Finalement, on simule, à chaque frame, à partir des messages de réplication du serveur, le nombre de frame en retard pour être en synchronisation avec le serveur au niveau de la simulation. Cette solution n’est pas très maintenable car elle implique de toujours tout resimuler plutôt que de resimuler seulement lorsqu’il y a un mistmatch avec le serveur. Donc cette solution sera difficile à mettre à l’échelle.

## Implémentation input prédiction

Pour l’input prédiction, on se sert de la latence calculé et on la multiplie par deux pour retrouver la frame historique de la position du joueur, s’il y a un *mismatch* avec ce que le serveur nous envoi, on replace tous les entités à leur position historique calculé selon la latence, puis on resimule tout jusqu’au présent, incluant les inputs pour se synchroniser avec le serveur. Cela permet au joueur de voir ses inputs immédiatement et de se faire corriger par le serveur au besoin. Cette solution est maintenable puisqu’elle permet une bonne mise à l’échelle puisque on corrige le client seulement au besoin.

## Conséquences

Utiliser l’input prédiction sans l’extrapolation cause les entités de passé à travers le joueurs coté client lorsque la latence est élevée puisque le reste des entités sont dans le passé comparé à lui. C’est pratiquement injouable.

À l’inverse, si on utilise l’extrapolation sans l’input prédiction, les inputs du joueurs ont un delay qui augmente avec la latence avant de voir l’effet. Mais au moins, la simulation est synchronisée avec le serveur de sorte qu’il n’y aura pratiquement jamais de réconciliation. C’est jouable mais de plus en plus frustrant plus que la latence est élevée.