1. **Gérer le Scoring**

Le Game Designer en charge du scoring a eu l’idée géniale d’ajouter du gameplay sur la bombe. Il vous demande de prototyper son gameplay. Ce qu’il veut c’est qu’un kill de NPC rapporte 10 points. Son expérience lui permet d’anticiper les problèmes, et pour éviter de frustrer les joueurs, il vous demande de partager (en parts égales) les points de kill entre tous les joueurs qui feront des dégâts dans un certain laps de temps. Au vu de la vitesse de déflagration, vous vous mettez d’accord sur 450 ms.

1. Spawnez un ennemi dans la map, en utilisant le code existant. Complétez-le pour que l’ennemi soit répliqué. Ce NPC peut rester immobile pour votre prototype.

Votre lead (qui sait qu’il aura à gérer les dégâts directs entre joueurs), vous demande de créer un manager dédié à la gestion des dégâts. Ce manager devra centraliser les demandes et gérer le scoring.

1. Proposez une architecture pour ce manager. Faites les schémas que vous jugerez utile pour expliquer votre système.

DamageManager

1. Faites le développement du manager et n’oubliez pas de commenter le code.
2. Validez la feature scoring en plaçant les deux joueurs à côté du même NPC et en activant la bombe quasi simultanément. Capturez une vidéo (par exemple avec FRAPS). Servez-vous en pour valider le fonctionnement.
3. Pour une meilleure compréhension, vous pouvez afficher le scoring ou le sortir en log.
4. Activez NEWT, et testez avec plusieurs latences (allez-simple) : 80ms, 150ms, 250ms, 1000ms. Que constatez-vous dans chaque cas? Comment l’expliquez-vous? Est-ce problématique et pourquoi?

Il n'y a pas de problème quand la latence est inférieure à 450 ms. Les attaquants ont bien été enregistrés dans le répartisseur de points. Les points sont partagés 450 ms après que l'hôte est reçu l'information de la première mort.

En revanche, dans le cas des 1000 ms de latence, si l'hôte de la session tue en premier, la deuxième personne ne pourra transmettre l'information au minimum 550 ms après la répartition des points (1000-450). Au contraire, si une autre personne tue, cela marchera si l'hôte tue la même personne entre 550 ms et 1450 ms après la première mort. Si l'hôte tue ces avant 550 ms, la trame de la personne non-hôte arrivera trop tard de quelques millisecondes car le partage aura déjà été fait. Si l'hôte tue après 1450 ms, il aurait déjà dû recevoir l'info de mort à 1000 ms, et c'est l'hôte qui ne recevra aucun point car c'est trop tard.

Dans un sens, on peut dire que l'hôte est avantagé car il a une plus grande marge de temps pour récupérer des points entre la première mort et l'instant où les points sont partagés. A l'inverse, ceux qui ont une grande latence ont beaucoup moins de chance de récupérer des points.

1. Est-ce que votre manager est fonctionnel en cas de paquets perdus ? Expliquez pourquoi ? S’il vous manque des informations allez au paragraphe 1b.

Si la trame perdue est l'information que la cible est déjà morte mais qu'il est possible que

Le Game designer passe tester votre prototype et croit remarquer qu’avec 150ms de latence (aller simple) le NPC ne meurt pas instantanément chez les joueurs.

* 1. Est-ce le cas ? Et pourquoi ?
  2. Si oui, comment pouvez-vous corriger cette impression ?