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.

5 NPC sont créés dans la fonction Game::\_OnUI, au clic du bouton "create session".

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.

Le DamageManager, qui permet est créé par l'hôte de la session dans Game::\_OnUI. Seul l'hôte l'instancie et gère la répartition des points.

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.

Les points ont bien été répartie (avec un délai de 2000ms au lieu de 450ms pour que la marge de manœuvre soit plus facile à saisir).

1. Pour une meilleure compréhension, vous pouvez afficher le scoring ou le sortir en log.

Les points sont affichés dans le log, dans un message d'erreur entouré de retour à la ligne pour mieux les reconnaitre.

1. 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 aucun problème avec la latence en ce qui concerne les bombes. Une bombe est répliquée dans le service hôte et ce n'est que ces bombes-là qui tuent les characters et définissent les points à répartir. Si les deux clients ou l'hôte lui-même appuie sur la touche espace au même instant, le décompte avant l'explosion de la bombe sera adapté pour exploser au même timestamp. Il faudrait avoir une latence de plus de 4000ms (décompte initial de la bombe) avant de voir la bombe exploser en retard et en dessus de 4449ms (décompte + temps avant répartition des points), le client n'aura pas les points qu'il aurait du avoir. Cela augmente à 4898ms si le client envoie la bombe 449ms avant un autre player avec moins de latence.

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.

Il y a trois cas où cela pose problème:

* Si la trame de réplication de bombe a été perdu lors de la réplication vers l'hôte. Il n'y aura pas de désynchronisation de l'état de la scène après l'explosion car seules les bombes chez l'hôte font des dégâts, mais les personnes ne verront pas la même chose et personne ne perdra de point de vie.
* Si la trame de réplication des dégats n'est pas envoyé vers les clients, alors le character en question sera mort partout sauf chez celui qui n'a pas reçu la trame. Ce character ne pourra plus mourir car il est déjà mort chez l'hôte et les bombes du client ne lui feront rien.
* Si la trame de répartition des points est perdu en allant chez le client, alors le tableau des scores ne sera pas le même chez toutes les personnes. L'Hôte aura cependant toujours le bon tableau.

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 ?

Oui, la bombe explose partout au même timestamp mais l'information de dégâts et de mort est reçu 150ms plus tard chez les clients. Donc, le NPC meurt 150ms après que la déflagration de la bombe le touche.

* 1. Si oui, comment pouvez-vous corriger cette impression ?

On pourrait quand même tuer le NPC au moment où la bombe le touche. Mais dans ce cas, il faut le ranimer s'il est toujours vivant chez l'hôte, et ne pas lancer l'animation de mort quand on reçoit la bonne trame cette fois.