

# **Compétition multi-joueurs de construction de mon(s)tres**

Christian Toinard

## **1. Introduction**

L'idée est de poursuivre le projet de programmation précédant en ajoutant des fonctionnalités de jeu en réseau. Pour cela vous allez tous repartir du projet de l'équipe Team JBZZ afin de permettre à différents joueurs d'entrer en compétition en exécutant différentes évolutions des populations de monstres, en visualisant les populations distantes et en récupérant les résultats de populations adverses pour constituer de nouvelles populations et ainsi tenter d'augmenter le score c'est à dire la qualité d'un monstre.

## **2. Règles du jeu.**

L'objectif est de réaliser un vrai jeu en réseau. Cependant, nous considérons les mêmes règles de jeu pour tout le monde.

a) Chaque joueur doit partager sans restriction avec tous les autres joueurs ses résultats ainsi que son paramétrage pour générer les populations et choisir le meilleur monstre.

b) Un joueur A peut décider d'utiliser la population et éventuellement le paramétrage d'un joueur B. Les deux populations A et B sont alors fusionnées pour obtenir une nouvelle population. Lorsque A décide d'utiliser le paramétrage de B, il abandonne son paramétrage au profit de celui de B.

c) Il est toujours possible d'arrêter ou continuer les évolutions de la population que cela soit à partir de la population initiale, de l'évolution courante ou de la fusion avec une population distante. Le joueur doit pouvoir modifier son paramétrage des coûts des monstres ainsi que les poids des différents éléments de notation d'un monstre.

d) Un joueur doit voir en temps réel la progression des populations distantes, notamment la synthèse avec la note du meilleur monstre d'une population distante. S'il le souhaite, le joueur doit aussi pouvoir visualiser en temps réel graphiquement le rendu des différents meilleurs monstres distants.

e) Un joueur peut quitter à tout moment une partie et obtient alors son classement vis à vis des joueurs distants. Le classement correspond simplement à l'ordre des notes des différents meilleurs monstres. A partir du moment où un joueur quitte le jeu sa population n'est plus utilisable par les autres joueurs. Cela donne une place disponible pour un nouveau joueur.

## **3. Objectifs d'un point de vue réseau**

L'idée est de privilégier un mode pair à pair entre les différents joueurs. Ainsi, si vous estimez devoir mettre en place un serveur, celui ci ne doit servir qu'à découvrir les extrémités réseau des différents joueurs, les transmissions s'effectuant ensuite directement sans emprunter le serveur.

Le mode pair à pair doit supporter différentes fonctionnalités :

- a) Un joueur doit pouvoir entrer (sous réserve de places disponibles) dans une partie. Lorsqu'il entre, les autres joueurs sont informés de son arrivée et voient aussi sa population initiale et son paramétrage.
- b) Chaque nouvelle itération dans la population d'un joueur doit être transmise efficacement à l'ensemble des participants. On s'intéresse notamment à ne transmettre pour chaque monstre que des informations du type liste et description des composants, liste des connexions entre composants, score du monstre.
- c) Le rendu graphique s'il est possible ne doit être réalisé que localement. Concrètement pour des raisons d'efficacité, un participant ne peut pas transmettre aux autres l'image graphique de son meilleur monstre. Cette image graphique sera recalculée localement par les participants distants au moyen des informations du monstre.
- d) Le départ d'un participant termine proprement c'est à dire que celui-ci informe les participants distants. Les pairs distants suppriment alors les données de ce participant notamment sa population, ce qui libère une place pour un nouveau joueur.

#### **4. Objectifs de passage à l'échelle**

Votre jeu doit pouvoir passer le plus possible à l'échelle. Cet aspect est secondaire mais il offre des bonus supplémentaires en terme d'évaluation puisque le jeu sera plus ou moins "massivement" multi-joueurs.

Le passage à l'échelle se caractérise sur différents aspects :

- a) Le nombre de places qu'autorise votre jeu. Celui-ci doit supporter au moins deux joueurs. Le nombre de place peut être limité pour des raisons d'interface utilisateur, de performance système ou d'efficacité réseau.
- b) Vous devrez donner une mesure des performances de votre jeu qui se base sur des expérimentations réelles. Ainsi, il n'est pas utile d'annoncer un nombre illimité de joueurs si vous n'avez testé votre jeu qu'entre quelques joueurs.
- c) Vos mesures de performances peuvent notamment porter sur :
  - la capacité de l'interface utilisateur à supporter un grand nombre de joueurs
  - l'architecture système qui autorise une dégradation lente des performances en fonction du nombre de joueurs
  - les protocoles réseau que vous avez définis pour minimiser la surcharge réseau et machine
  - la fluidité du jeu en fonction du nombre de joueurs.

Sur tous ces points, il est inutile de dire ma solution est efficace si vous n'avez pas fait de mesures portant sur différentes expérimentations ou des évaluations quantifiées.

## **5. Objectifs de sécurité**

Votre jeu doit éventuellement proposer des propriétés et des mécanismes de sécurité. Cet aspect est aussi secondaire.

Les objectifs de sécurité portent notamment sur :

- l'identification des participants. Ainsi, si vous utilisez des identifiants pour chaque participant, ceux ci doivent être le plus possible confidentiels et intègres. La confidentialité peut par exemple reposer sur une connexion de type SSL qui chiffre les données transmises entre les différents joueurs. L'intégrité peut consister en un mécanisme de signature de l'identifiant afin d'éviter que quelqu'un ne forge un identifiant à la place d'un autre joueur.
- la difficulté de rejouer un identifiant pour usurper une identité. Pour cela, vous pouvez par exemple ajouter une durée de vie à l'identifiant afin que celui-ci soit régénéré régulièrement. Couplé à un mécanisme de signature on peut alors limiter les attaques par rejeu.
- une utilisation de http afin que les transmissions passent les firewalls. Pour cela vous pouvez par exemple utiliser un serveur sur une DMZ qui accepte les connexions http entrantes et retransmet les données sur les différents tuyaux http établis. Dans ce cas, vous réfléchissez notamment aux mécanismes d'intégrité et de confidentialité nécessaires.

Dans tous les cas, vous devrez tenter d'identifier les limites de votre solution en terme de sécurité. Ainsi, vous pourrez décrire les failles résiduelles ou les scénarios d'attaque non couverts voir dérouler des preuves de concept pour ces attaques.

## **6. Organisation de l'équipe**

Vous allez structurer une équipe d'au maximum 6 étudiants.

Un étudiant sera choisi comme coordinateur pour l'équipe afin d'organiser le travail au sein de l'équipe, mettre en place des processus de décision et de coopération entre les membres de l'équipe. Le coordinateur sera chargé des échanges avec les enseignants qui participent à l'encadrement du projet (Armen Petrossian et Christian Toinard).

Vous êtes entièrement libre de vous organiser au sein de l'équipe. Cependant, vous devrez, avant de vous engager dans une solution ou une organisation, l'exposer précisément à vos encadrants. La présence aux séances de suivi est obligatoire pour toute l'équipe et doit servir à la coordination et aux échanges.

Une évaluation par les pairs est demandée. Pour cela les différents membres devront donner un avis sur le coordinateur. Celui ci bénéficiera de la possibilité d'attribuer des bonus et des malus à somme nulle entre les membres via une discussion avec les encadrants.

Les encadrants font une évaluation personnalisée des étudiants tenant compte de la participation aux séances de suivi, de l'implication des élèves et de l'avis du coordinateur.

Dans tous les cas, l'avis du coordinateur est anonyme puisque la notation est individualisée par les enseignants globalement sans que les membres puissent connaître l'évaluation par le coordinateur. Les avis des élèves d'une équipe serviront éventuellement à ajuster la note du coordinateur.

## **7. Travail attendu**

Votre travail doit permettre de présenter très vite une première version du jeu que vous présentez à vos encadrants. Ainsi, plus vous fournissez rapidement une première version qui permet de partager les populations en "temps réel" plus l'on considère que le premier échelon est réussi.

Sur la base de cette première version jouable vous proposerez différentes améliorations en terme d'interface, réseau, système et de sécurité. Pour chacune de ces améliorations vous présenterez les limites.

La livraison finale consiste en un rapport sommaire qui décrit les services offerts, les mesures ou évaluations faites et les limites. Une soutenance sera axée sur la démonstration et résumera en deux diapositives les résultats obtenus et les limites.

Votre travail sera évalué sur la base de votre capacité à présenter et justifier votre solution durant les séances, le rapport et la démonstration finale.

Une attention particulière sera portée à la justification de vos protocoles sur le plan de l'efficacité et de la robustesse. De même une justification vous est demandée en terme de robustesse et d'efficacité de votre approche système.

Nous vous conseillons d'éviter les solutions complexes sources en général d'erreurs et très souvent d'inefficacité. Par exemple, vous devez justifier le format des messages vis à vis d'objectifs fonctionnels et non fonctionnels (performance, robustesse, ...) et le respect du modèle OSI pour vos différents services et protocoles. Vous devrez aussi justifier que l'architecture système ne consomme pas trop de ressources et évite au maximum les interblocages ou les attentes actives. Des mesures de performance sont les bienvenues ainsi que des évaluations de la complexité sur les plans réseau, système et logiciel.