[Note du traducteur : ceci est une traduction en français libre et non officielle du FFF paru sur le forum]

Friday Facts N°326
Émetteur de particules et cache de données

Posté par Allaizn, Rseding et Klonan

le 20/12/2019

Encore plus d'optimisations des particules (par Allaizn)

Les récentes optimisations du système de particules par Rseding (FFF-322) ont rendu les particules beaucoup plus légères qu'auparavant, mais elles ont tout de même transformé les particules en des créatures plutôt complexes. Voici un résumé rapide des actions qu'une particule peut faire lors de sa mise à jour :

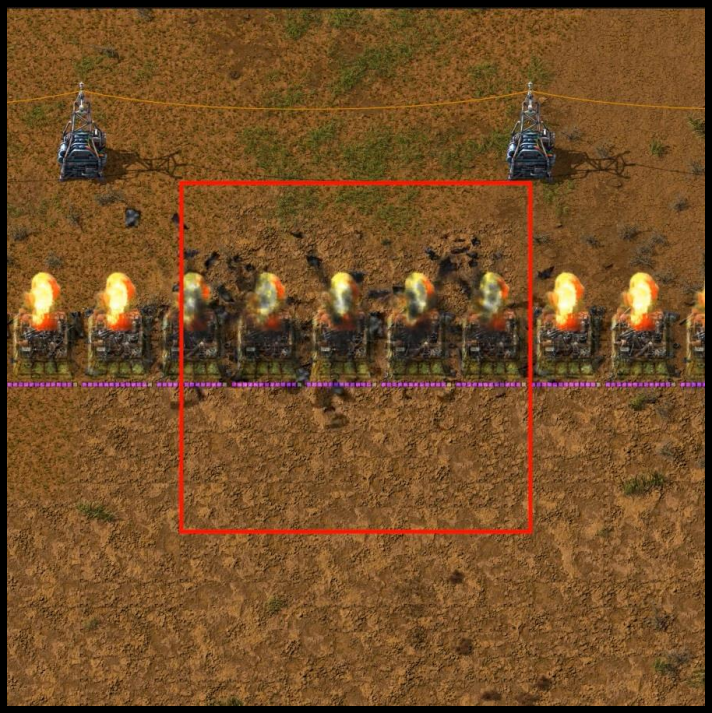
- Déplacer leur propre position.
- Déplacer leur animation sur une autre scène.
- Se poser dans l'eau et activer un déclenchement.
- Activer un autre déclenchement avec une certaine fréquence.
- Se retirer du monde du jeu une fois leur vie terminée.

Ce qui rend ce complexe est que les déclenchements sont des systèmes à usage général qui peuvent faire toutes sortes de choses, y compris créer et détruire des entités, du feu, de la fumée et d'autres particules ainsi que jouer des sons ou encore activer de façon récursive encore plus de déclenchements. En d'autres termes : activer un déclenchement est une situation où "tout peut arriver" et est donc totalement imprévisible, ce qui rend les optimisations extrêmement difficiles.

L'émetteur de particules

Le jeu de base, et la plupart des mods, n'utilisent pas de déclenchements particulièrement fous lors de la création de particules - le but est généralement de faire apparaître un tas de petites textures animées et de les faire voler à l'écran (ce qui est un peu ironiquement ce qu'on appelle habituellement une "particule"). Une idée pour optimiser davantage les particules était donc de créer une sorte de particule "simple", qui ne pourrait pas activer toutes sortes de déclenchements, pour permettre de les manipuler en masse, ce qui est généralement plus rapide que de les manipuler individuellement. Cette manipulation en masse serait faite par un truc appelé "`particle emitter`", dont tout le travail est de créer, mettre à jour, dessiner et finalement détruire les particules simples qu'il gère, l'idée étant qu'un biter agonisant n'aurait pas à engendrer des centaines de particules, mais seulement un seul/peu d'émetteur(s).

Mais ce n'est pas tout : les particules simples ne sont pas capables de changer un autre état de jeu, et ne sont donc mises à jour que pour maintenir leurs propres valeurs internes - principalement leur position et leur vitesse. Un petit exercice de physique plus tard, l'idée est née de ne pas mettre du tout les particules à jour - vous pouvez calculer leur position actuelle à partir de leur position de départ après tout ! Encore mieux : si les particules ne sont pas affichées, alors il n'y a en fait pas de nécessité de les créer, donc il n'y a aucune raison de le faire tant que l'émetteur n'est pas à distance de dessin - des millions de biters mourant dans de gigantesques fontaines de sang en dehors de l'écran seront donc sans importance pour votre image et votre temps de mise à jour !



Une visualisation de l'émetteur en action : la case rouge représente l'écran actuel.
Les particules gérées par les émetteurs en dehors de la zone de l'écran n'existent tout simplement pas du tout.
[NdT : cliquez sur l'image pour voir l'animation]

Le fait que les particules elles-mêmes ne puissent pas affecter l'état de jeu a un autre avantage : dans un jeu multijoueur, chaque joueur n'a qu'à générer les particules qu'il voit lui-même, au lieu de celles qui sont visibles par les autres. Cela suggère également de ne pas utiliser la fonction de mise à jour de l'émetteur, mais plutôt celle de dessin, ce qui apporte encore plus d'avantages car la fonction de dessin est appelée pendant la phase de préparation du rendu, qui fonctionne sur autant de processus que vous lui permettez d'avoir.

Cependant, tout cela ne fonctionne pas correctement comme par magie, et il y a des cas limites qui doivent être traités. Par exemple : que se passe-t-il si un émetteur est créé hors écran puis se retrouve à distance de vue ? Que se passe-t-il si vous sauvegardez et rechargez ? Que se passe-t-il si vous sauvegardez et rechargez avec un ensemble de mods qui n'a plus les particules définies ? Il serait très étrange de voir votre silo à fusée exploser en d'innombrables morceaux, de voir comment ils volent et s'écrasent au sol - puis de sauvegarder et recharger et de tout revoir parce que l'effet de particule a redémarré.

La gestion de ce genre de problèmes a pris un certain temps et n'a heureusement augmenté que marginalement la complexité interne du système, ce qui m'a permis de me concentrer sur le développement de ses fonctionnalités. Actuellement, les choses suivantes sont supportées et présentes sur un émetteur :

- Manipulation de particules simples avec des positions de départ et des vitesses individuelles aléatoires.
- Manipulation de flux de particules simples via des traînées normales et instantanées comme indiqué dans le FFF-325.
- Manipulation de particules simples avec une traînée de fumée derrière elles (le FFF-325 en a quelques exemples, mais l'effet existait déjà auparavant).
- Manipulation de particules simples ayant un impact sur le sol en étant potentiellement remplacées par un jet d'eau lorsqu'elles frappent l'eau.

Les émetteurs de particules ont deux restrictions principales :

- Ils ne traitent qu'un seul type de particules (et les fumées et éclaboussures d'eau qui y sont techniquement associées). Pour faire éclater une machine d'assemblage en morceaux métalliques et en projection d'huile, il faudra donc deux émetteurs.
- Les particules gérées par un émetteur ne peuvent pas s'envoler trop loin de l'émetteur (qui lui-même ne bougera jamais), car il faut déterminer à quelle distance en dehors de la distance de dessin il faut chercher des émetteurs susceptibles de vouloir afficher leurs particules.

Une animation de particules, pour démonstration, montrant tous les effets à la fois - tous sont gérés par un seul émetteur.

[NdT : cliquez sur l'image pour voir l'animation]

Durée de démarrage - Cache de données (par Rseding)

La durée de démarrage du jeu (la durée avant atteindre le menu principal) est aussi importante pour nous que la durée de compilation (voir FFF-206). Avec la fréquence à laquelle nous compilons et lançons le jeu pour tester les choses, chaque petit bout de temps supplémentaire passé à attendre que le jeu se charge est du temps perdu.

Le processus de démarrage de Factorio comprend deux parties principales :

1. Passer en revue chaque mod activé et collecter les données du prototype qu'il définit/génère (l'"étape des données").
2. Charger et traiter les sprites dont le jeu a besoin pour s'exécuter.

C'est un spectacle familier pour ceux qui jouent avec beaucoup de mods.

[NdT : cliquez sur l'image pour voir l'animation. Vraiment ?]

Dans le passé, nous avons fait un dispositif expérimental qui mettait en cache le chargement et le traitement des sprites, de sorte que nous n'avions jamais besoin d'attendre lorsque que rien n'a changé autour d'eux. Cependant, le jeu avait toujours le processus de toute l'"étape des données" à chaque fois que le jeu démarrait.

Durant le développement normal, ce n'était pas vraiment un problème - cela se passait en une fraction de seconde dans la plupart des cas. Cependant, au fur et à mesure que le jeu a grandi, la quantité de choses qui sont traitées pendant l'étape des données a également augmenté. De plus, pour chaque mod activé qui a quelque chose à faire à cette étape, le temps doublait environ. Récemment, j'ai commencé à me demander ce qu'il faudrait pour faire le même genre de système de cache que celui que nous avons pour le chargement des sprites lors de l'étape des données.

Comme les résultats sont généralement les mêmes entre les redémarrages, cela signifierait qu'il n'a pas besoin de faire la plus grande partie du travail – et cela devrait être plus rapide. Après avoir travaillé dessus pendant environ une journée, j'avais un prototype fonctionnel ; mais il n'était en fait pas plus rapide, avec le jeu de base seulement. Ne voulant pas encore abandonner, j'ai passé un peu de temps avec le profileur et j'ai réussi à trouver quelques zones que je pouvais optimiser et ainsi réduire de moitié le temps que la logique de mise en cache nécessite. Donc, ça a finalement eu un certain bénéfice pour le jeu de base (bien qu'assez petit).

Ce à quoi je ne m'attendais pas, c'est à quel point cela allait améliorer le cas moddé. Ce qui prenait 25 secondes dans mes tests n'en prenait plus que 4 avec le nouveau réglage de cache activé. Le gain de temps est d'autant plus important que le nombre de mods activés augmente. Le réglage est toujours désactivé par défaut parce qu'il est très expérimental, mais s'il s'avère suffisamment stable, nous pourrions l'activer par défaut.

Mods de Noël (par Klonan)

C'est le dernier FFF avant Noël, donc j'ai pensé que nous pourrions aussi célébrez certains des mods qui visent à créer un esprit de fête dans le jeu.

Alien biomes

Le terrain enneigé de Alien Biomes est tout simplement magnifique. Dans les réglages de génération de la carte, vous pouvez activer l'option "Climats froids", de sorte que votre monde entier n'est qu'un doux paradis hivernal.

Holiday artillery

Il ne faut pas oublier non plus de partager l'amour avec nos amis mordeurs, ce mod vous permettra de livrer des cadeaux partout, et il embellit la tourelle ou le wagon d'artillerie / livraison de cadeaux avec une jolie peinture rouge et verte.

Christmas tree

Et bien sûr, aucune usine hivernale n'est complète sans un beau sapin de Noël.

[NdT : cliquez pour voir la vidéo]

Nous vous souhaitons un joyeux Noël, et comme toujours, faites-nous savoir ce que vous en pensez sur notre forum.

Discuter sur nos forums

Discuter sur Reddit

[NdT : Toute l'équipe du Discord francophone se joint à moi pour également vous souhaiter un joyeux Noël !]

[NdT : Traduit avec l'aide de www.DeepL.com/Translator (version gratuite)]