

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



Friday Facts N°322

Le nouveau système de particules

Posté par Klonan et Rseding

le 22/11/2019

Programmes des publications (par Klonan)

Cette semaine, nous avons publié la version 0.17.79, et l'avons marquée comme stable. En interne, nous l'avons appelé 'Stable 3', et la caractéristique principale en est les nouvelles infobulles que nous avons présentées dans le [FFF-318](#).

Il y a une contrainte que nous nous sommes imposée lorsque nous avons commencé ce calendrier de sorties plus rapide : nous voulons éviter de casser les mods. C'est assez facile en principe, ne pas commencer à renommer des choses, ne pas supprimer des fonctionnalités de l'API, etc. Cependant, au fur et à mesure que nous évoluons, il y a certaines fonctionnalités et améliorations que nous ne pouvons pas faire concrètement d'une manière qui ne casse pas les mods, comme la nouvelle interface graphique du joueur ([FFF-289](#)), et la correction des couleurs ([FFF-320](#)).

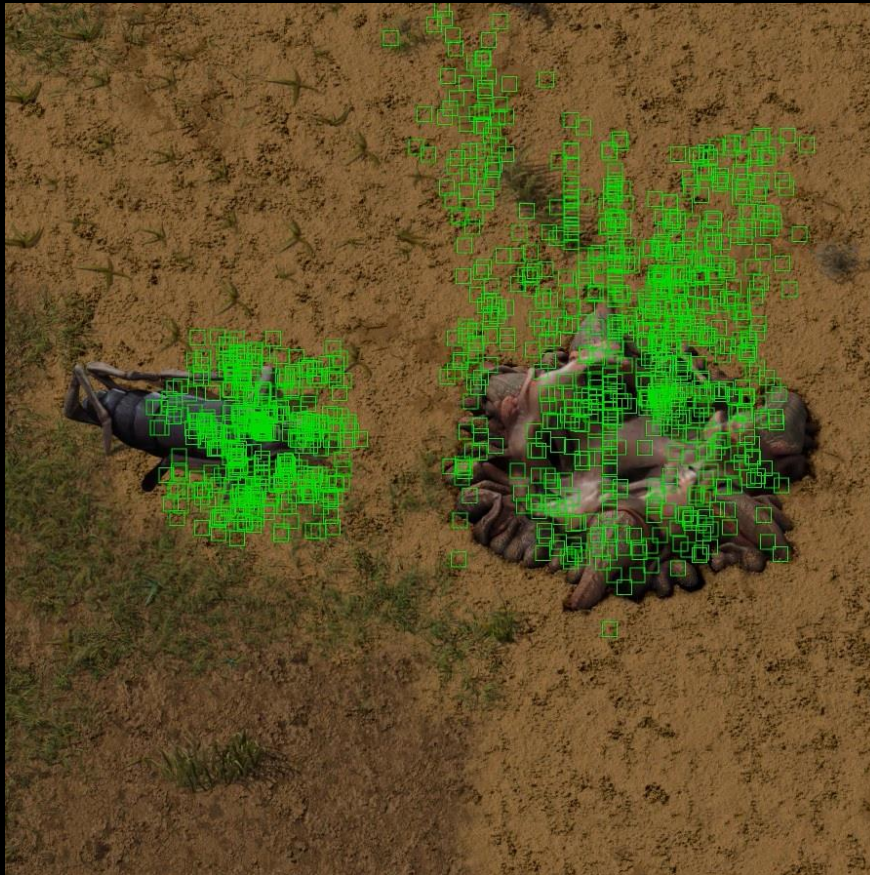
C'est pour cette raison que nous allons accumuler certains de ces changements destructeurs de mod, et les publier tous à la fois. Puisque cela cassera certainement les mods, nous remplacerons le numéro en version majeure, donc ce sera la 0.18.0.

Nous avons déjà commencé à fusionner en interne ces fonctionnalités "0.18" dans notre branche principale, donc nous ne ferons plus de versions 0.17 (à moins que quelque chose d'absolument catastrophique soit découvert).

Le problème avec les particules (par Klonan)

Les particules sont dans le jeu depuis aussi longtemps que tout le monde se souvient, et tout bien considéré, elles sont assez simples. Il s'agit d'une petite entité principalement décorative, que nous utilisons pour ajouter un peu de gratifications visuelles à la mort et à l'agonie des insectes et des machines.

Par exemple, les biters sont pleins de sang, donc quand ils meurent, ils font tout un éclaboussement. Essayons de compter combien de particules nous produisons dans une explosion typique.



Dans ce clip, le sprite des particules de sang a été remplacé par une visualisation de débogage, pour faciliter le comptage.

[NdT : Cliquez sur l'image pour voir la vidéo]

Il y en a donc pas mal. Le biter a produit 427 particules, et le nid 749. Eh bien, ils ne persistent pas très longtemps, et ils ne sont que décoratifs, donc tout va bien, n'est-ce pas ?

Un mot clé que j'aimerais souligner dans la description précédente, c'est que c'est une *entité*. Lorsque kovarex et slpwnd ont commencé à fabriquer Factorio, ils ont créé un système robuste pour gérer les objets du jeu et leurs interactions - le système des entités - et tout ce qui a une représentation physique dans le monde du jeu était construit sur ce système. Au fur et à mesure que le jeu a pris de l'ampleur, il est devenu évident que le système des entités est trop lourd pour certaines choses, et nous pouvons obtenir de meilleures performances en créant des systèmes plus spécialisés. Cela a conduit à supprimer les éléments sur les tapis du système des entités dans la version 0.12, et à faire la même chose pour la fumée et les décorations de terrain dans les versions ultérieures. Malgré le fait que la plupart des autres jeux ou moteurs de jeu aient des systèmes de particules très efficaces dès les premiers stades de développement, les particules de Factorio sont encore accrochées au système des entités dans la version 0.17.

Cela signifie que les particules sont enregistrées sur la surface de jeu de la même manière générique que tout le reste. Cela signifie également qu'elles sont itérées lors des recherches d'entités. Quels types d'actions du moteur effectue une recherche par entité ?

- Effets déclencheurs dans une zone donnée - tels que grenades, dégâts causés par le jet du lance-flammes, bombes atomiques, capsules de poison, etc.
- Recherche de trajectoires - Pour vérifier si une trajectoire peut traverser une tuile donnée.
- Contrôles de collision en mouvement - tels que personnages, unités, projectiles.
- Et bien d'autres....

Comme vous pouvez l'imaginer, le fait d'avoir des milliers d'entités supplémentaires pour l'itération à chaque tick, peut commencer à ralentir la simulation. Le pire cas actuellement, c'est de défendre vos murs avec des lance-flammes. Dans l'image ci-dessous, toutes les entités sont mises en évidence avec une visualisation de débogage.



Si vous avez perdu le compte, cette scène contient 15 689 entités

[NdT : une comparaison entre cette image et une version sans les cases vertes est disponible dans la version originale, en anglais]

Les lance-flammes contre les biters sont à peu près l'exemple parfait du problème :

- Les biters doivent vérifier à chaque tick quand ils se déplacent s'ils entrent en collision avec quelque chose.
- Lorsque le jet du lance-flammes retombe, il fait des dégâts par éclaboussures, ce qui est un effet déclencheur de zone.
- Le jet du lance-flammes crée également un incendie au sol, ce qui fait un contrôle de collision.
- Tous les 10 ticks, le feu au sol fait des dégâts, avec un effet déclencheur de zone.
- Les lance-flammes tuent les biters très efficacement, de sorte que beaucoup de particules jaillissent dans un laps de temps très court.

Cette combinaison entraîne par la suite des ralentissements importants lorsque de grands groupes viennent frapper à la porte. De plus, depuis que nous avons amélioré la recherche de trajectoires, le problème est encore pire dans les dernières versions de la 0.17.

Alors, comment on résout ça ?

Optimisation des particules – technique (par Rseding)

Quand Posila m'a parlé pour la première fois de retravailler le fonctionnement des particules dans le jeu, j'ai eu beaucoup d'idées. Elles n'ont pas toutes fonctionné à la fin, mais elles avaient l'air bien :

1. Ce ne seraient pas des entités.
2. Elles fonctionneraient comme la fumée (stockées dans des blocs de mémoire contigus).
3. Elles n'auraient pas besoin d'être mises à jour à chaque tick si elles n'ont pas d'effet sur l'état du jeu.

En fin de compte, j'ai trouvé que cela ne valait pas la peine d'ajouter de la complexité pour faire fonctionner le numéro 3.

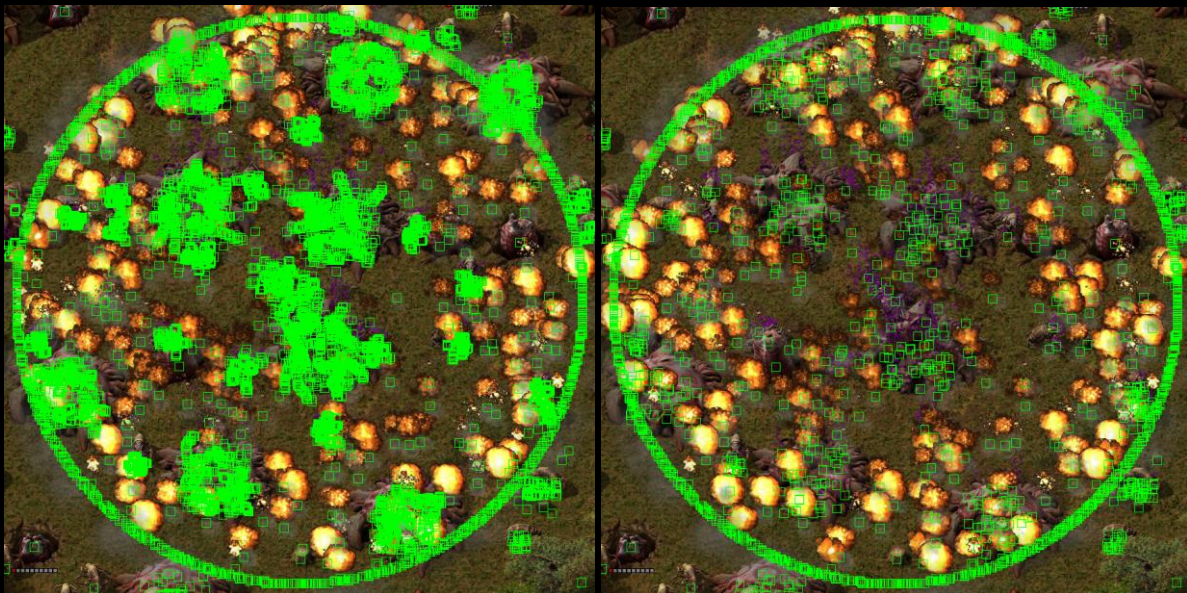
Le procédé

Les particules ne devraient pas être des entités. Quelque chose étant une entité a beaucoup de mémoire et de performances que les particules n'ont tout simplement pas besoin. Malheureusement, lorsque j'ai mis en place l'artillerie, j'ai fait du marqueur de ciblage manuel une particule. Les raisons ne sont plus importantes, mais cela signifie que j'ai dû ajouter des migrations pour gérer cela, puis des migrations pour supprimer toutes les particules de l'entité.

J'ai retiré toutes les données/logiques supplémentaires des particules dont elles n'avaient plus besoin - réduisant la taille de chaque particule en mémoire de 224 à 64 octets. Le fait de ne plus en faire des entités réduit également la quantité d'informations qui doivent être enregistrées dans le fichier de sauvegarde. Cependant, dans la plupart des cas, les particules n'existent pas assez longtemps pour finir dans le fichier de sauvegarde, donc cela n'a pas vraiment d'importance.

J'avais besoin d'un endroit pour stocker/travailler les durées de vie des particules au fur et à mesure qu'elles sont créées, qu'elles existent pendant une courte période de temps et qu'elles disparaissent. La plupart des choses dans le jeu finissent par être stockées sur un bloc donné (une zone 32x32 du monde). Dans le cas des particules, je ne me soucie pas du tout des autres choses sur un bloc donné, donc je ne voulais pas les coller là. Au lieu de cela, j'ai fait une chose séparée qui fonctionne très étroitement aux blocs et l'ai appelée "bloc de particules". Les principales différences sont les suivantes : ils existent seulement lorsqu'il y a des particules à mettre à jour et les seules données qu'ils contiennent sont des particules. Cela signifie que lorsque le jeu a besoin de passer en revue chaque particule pour les mettre à jour, tout ce qu'il a à faire est de passer en revue tous les "blocs de particules" qui existent et de lancer la mise à jour. De plus, comme j'ai le contrôle total de ces nouveaux blocs de particules, je peux les recycler en mémoire au besoin pour éviter de passer plus de temps à allouer et à dé-allouer de la mémoire à mesure que les particules vont et viennent.

Le résultat final est une amélioration des performances, une utilisation réduite de la mémoire et une simplification de la logique de fonctionnement des particules. Un simple test non scientifique que nous avons effectué était d'atomiser la même base de bits dans l'ancien système et dans le nouveau système, et d'enregistrer la durée maximale de mise à jour.



L'anneau circulaire d'entités autour du bord sont les projectiles de l'onde de bombe atomique.

Ancien système :

- Temps maximum de mise à jour de l'entité = 7 ms
- Nombre d'entités (à l'exclusion des projectiles) = 7 769

Nouveau système :

- Temps maximum de mise à jour de l'entité = 2,4 ms
- Temps maximum de mise à jour des particules = 1,7 ms
- Nombre d'entités (à l'exclusion des projectiles) = 786

Ce n'est pas très scientifique ou très contrôlé, mais juste pour donner une idée de l'ampleur de l'amélioration. Une réduction du nombre d'entités créées de 10 fois et une réduction d'environ 40% du temps maximum de mise à jour. Avec le nouveau système de particules optimisé, l'équipe des effets spéciaux peut désormais devenir un peu plus folle dans le futur avec les effets de particules....

Comme toujours, faites-nous savoir ce que vous en pensez sur notre [forum](#).

[Discuter sur nos forums](#)

[Discuter sur Reddit](#)

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