[Note du traducteur : ceci est une traduction en français libre et non officielle du FFF #262]

Dans le FFF-241, nous avons discuté de la façon dont le jeu fournit de l'information au joueur de plusieurs façons plutôt confuses : flèches clignotantes et cercles, messages de chat dans le coin inférieur gauche de l'écran, objectifs dans le coin supérieur gauche, bulles orange sur le dessus du joueur, et ainsi de suite. Ces problèmes sont exacerbés sur les moniteurs à haute résolution, où l'information est encore plus dispersée.

Nous avons essayé plusieurs façons d'unifier cette information, mais une grande partie de celle-ci devait se trouver dans l'espace monde, ou devait avoir un lien entre l'espace écran et l'espace monde. La solution habituelle est d'avoir l'interface graphique "pointer" vers une entité dans le monde du jeu, mais nous voulions quelque chose de plus intéressant.

Bonjour, je m'appelle : Compilatron (par Abregado)

L'idée d'un compagnon robot dans un jeu vidéo n'est pas nouvelle, mais la solution s'adapte très bien à Factorio. Quand je suis arrivé au bureau et que j'ai mentionné le concept, Albert m'a immédiatement demandé "*As-tu déjà remarqué le forum FactorioBot*" et m'a montré quelques rendus de concept pour le robot. L'idée d'Albert avait déjà été discutée à plusieurs reprises comme une solution à ce problème, n'attendant que le bon moment. Eh bien, Compilatron c'est maintenant le moment !

Le dernier rendu d'Albert de Compilatron, notre personnage compagnon.
[NdT: cliquez sur l'image pour voir l'animation]

Le personnage Compilatron nous permet de lier le texte et les données de position, ce qui crée une liaison beaucoup plus étroite entre le sujet de l'information et l'information elle-même. Auparavant, les bulles orange et les flèches clignotantes pouvaient être utilisées de cette façon, car elles pouvaient indiquer des entités dans le monde. Compilatron peut le faire sans interrompre l'immersion et le déroulement du jeu.

Nous ne voulons pas en révéler trop, mais vous pouvez voir que c'est déjà mieux que les bulles orange.

Compilatron nous donne aussi un coup de main avec les informations sur l'espace écran. Si nous parvenons à styliser certains éléments de l'interface utilisateur (comme la nouvelle et pour l'instant inédite fenêtre d'objectif) pour qu'ils correspondent au style de Compilatron, je crois que nous pouvons créer une association encore plus forte entre le texte dans l'espace écran et celui dans l'espace monde.

Une autre façon où le Compilatron nous aide à garder l'immersion intacte est en tant qu'agent de traçage. Actuellement, les événements liés à l'intrigue de la campagne "se produisent" lorsque les objectifs sont atteints ou lorsque le monologue interne des personnages décide de la prochaine étape. Maintenant nous pouvons avoir Compilatron comme une force motrice de la campagne, il donne les objectifs au joueur, interagit avec les entités pendant les scènes, et dans l'ensemble rend les choses moins gênantes que maintenant.

Ce sont deux des principaux avantages d'avoir Compilatron, il y a aussi des choses plus intéressantes que nous pouvons faire avec elle à l'avenir. Les spécificités de la façon dont Compilatron va travailler

et interagir avec le joueur sont encore en développement, donc attendez-vous à certains changements à ce que vous voyez ici.

Jusqu'à présent, j'ai l'impression que la solution est en cours d'élaboration, mais non sans quelques problèmes, dont le plus important était de travailler avec l'unité d'IA [NdT : intelligence artificielle] actuelle…

Compilatron - Ajouts techniques (par Wheybags)

J'ai travaillé sur l'aspect technique de Compilatron, qui devrait intéresser les moddeurs parmi vous. Compilatron est entièrement codé en Lua, ce n'est pas du tout une entité C++ personnalisée, ce qui signifie que nous avons dû ajouter quelques éléments à notre API de script pour que cela soit possible.

Tout d'abord, le Compilatron est une Unité, ce qui signifie qu'il est contrôlé de base de la même manière que les biters. Comme les Unités n'ont jamais été utilisées dans le jeu de base pour autre chose que des morsures, il y a un certain nombre de comportements inhabituels introduits dans le code de contrôle des Unités, qui n'ont pas vraiment de sens en dehors du contexte des biters. Pour résoudre ce problème sans casser le jeu de base, j'ai ajouté des balises prototypes aux unités qui désactivent ces comportements spéciaux par défaut, et ne les activent que sur les biters (par exemple, essayer de revenir de temps en temps à un spawner [NdT : un nid], ce qui peut gâcher l'exécution de vos commandes actuelles). Ces problèmes sont corrigés de façon ponctuelle chaque fois que nous les rencontrons, donc si vous êtes un moddeur qui a souffert de ce genre de choses, veuillez nous faire savoir ce que vous aimeriez changer.

Le changement le plus significatif que j'ai fait est probablement d'ouvrir le chercheur de chemin au script Lua. Cela signifie que vous pouvez émettre des requêtes de recherche de chemins asynchrones, qui ne sont pas associées à une commande réelle, utile par exemple pour vérifier si vous pouvez aller d'un endroit à un autre. Plus tard, nous inclurons également la possibilité d'émettre un ordre de déplacement et de regarder dans les données de la recherches de chemins calculées, ce qui vous permettra de modifier les chemins avant de les exécuter, ou même d'écrire votre propre recherche de chemin totalement personnalisée en Lua.

En plus de ce qui est mentionné ci-dessus, il y a eu toute une série de changements et d'ajouts divers, notamment :

- Événement de script à la fin de la commande IA, pour savoir quand une Unité a terminé sa commande en cours.
- Permettre aux unités de se frayer un chemin à travers des bâtiments amis (c'est-à-dire de s'y rendre, de les détruire, puis de continuer par-dessus l'endroit où elles se tenaient auparavant). Seulement sur demande explicite, bien sûr.
- Permettre aux unités de passer et d'ouvrir des portes amies.
- Ajout d'un tas de balises pour les commandes *go_to_location*, exposant des requêtes de recherches internes de chemins telles que des requêtes de recherches de chemins de priorité basse, et donnant la priorité aux chemins droits.
- Les unités éviteront désormais de passer dans toutes les entités amies, contrairement à l'ancien comportement consistant à éviter uniquement les entités de même force.
- Ajout d'une nouvelle commande IA *stop*, et amélioration significative de la commande *wander* [Ndt : errance].

- Ajout d'une nouvelle entité *HighlightBox*, qui dessine une boîte de sélection autour d'une entité spécifiée ou à une position spécifique.
- Ajout d'une entité bulle de parole.
- Permet d'accéder aux scripts Lua par un chemin du style *__MOD-NAME__/script.lua*, de sorte que vous pouvez référencer les mêmes scripts Lua de plusieurs scénarios dans un seul mod.

Klonan a déjà créé un mod pour aider à tester et déboguer toutes les nouvelles commandes de modding, les scripts et les fonctionnalités de l'API :

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

Nous sommes très intéressés par les avantages que ces changements apporteront "gratuitement" aux moddeurs, donc si vous êtes un moddeur intéressé avec quelques opinions/requêtes à ce sujet, faites-le-nous savoir, nous avons commencé un fil de discussion spécifique sur le sujet ici.

Comme toujours, faites-nous savoir ce que vous en pensez sur notre forum.