

Note du traducteur : ceci est une traduction en français libre et ouverte du FFF #259 effectuée par SNova.

## Friday Facts # 259 – Scan-codes, Prototype IDs, Vers HR.

Publié par Posila, Rseding, Albert le 2018-09-07, tous les [posts](#)

### Scan-codes vs codes-clés (posila)

Lors de la migration d'Allegro vers SDL, HanziQ et Jiri ont remplacé le traitement des touches par l'utilisation de codes-clés et de scan-codes. Avant de commencer à sauter avec joie, vous vous demanderez probablement : qu'est-ce que c'est et pourquoi devrais-je m'en soucier ?

Eh bien, vous devriez demander. Vous ne vous en souciez probablement pas, sauf si vous habitez en dehors des États-Unis et / ou si vous utilisez une disposition de clavier non américaine. En bref, le code-clé est un identificateur de clé dépendant du symbole que la clé va afficher lorsqu'il est pressé. Le code-scan est un identificateur de clé basé sur l'emplacement physique d'une clé. Ainsi, par exemple, les joueurs disposant d'un clavier français (AZERTY) doivent sauter aux options de contrôle après avoir lancé le jeu pour la première fois et reconfigurer le mouvement de WASD à ZQSD afin de pouvoir déplacer leur personnage sans se blesser la main.



En 0.17, les contrôles seront mappés aux clés correctes par défaut, indépendamment de votre disposition, et resteront mappés sur ces clés physiques même si, pour une raison quelconque, vous modifiez la disposition du clavier pendant que le jeu est en cours d'exécution. L'inconvénient est que la plupart des dispositions non américaines n'ont pas abouti à des contrôles complètement dysfonctionnels, alors les gens ont continué à jouer avec et s'y sont habitués. Ils devront donc s'habituer à la disposition avec laquelle le jeu a été conçu à l'origine, ou configurer manuellement les contrôles en fonction de leur utilisation habituelle.

Mais attendez, il y a un piège ...

Il y a quelques semaines, nous avons annoncé de nouveaux outils de construction, liés par défaut à des raccourcis universels (Ctrl + C pour « copier », Ctrl + X pour « couper » et Ctrl + Z pour « annuler »). Bilka a souligné que le clavier allemand a remplacé le Y par Z (comme le tchèque, mais que les développeurs ne l'utilisent souvent pas) et « annuler » par défaut Ctrl + Y à la place. Pour corriger ce type de raccourcis, nous déterminons les codes de numérisation par défaut appropriés au démarrage, afin que l'annulation soit toujours Ctrl +

Z, quelle que soit votre présentation, mais l'action reste liée à ces touches si vous modifiez la disposition du clavier lors de l'exécution. Ce qui est un compromis raisonnable.

Nous pouvons également le faire pour d'autres contrôles (il est naturel que M soit toujours la clé par défaut pour ouvrir la carte et T pour ouvrir l'écran technologique), mais il y a un autre problème. Il est tout à fait raisonnable que le joueur marche vers le nord, et Ctrl + clique sur certaines entités. Rappelez-vous le clavier AZERTY ? Le joueur garde Z enfoncé pour marcher vers le nord et appuie sur Ctrl pour démarrer le clic de contrôle. Eh bien, je l'ai testé et cela ne déclenche pas, mais empêche toujours le joueur de marcher. Donc, ce n'est pas complètement destructeur, plutôt ennuyeux. Je ne sais pas comment ou si nous allons résoudre ce problème, peut-être que les personnes qui ont créé ces types de collisions devront les résoudre en modifiant manuellement les options des commandes.

## Prototype IDs Stable (Rseding)

Le nouvel éditeur de carte est terminé, l'une des dernières tâches achevées étant l'importation de surfaces à partir d'autres fichiers de sauvegarde. L'importation de surfaces d'une sauvegarde dans une autre s'est avérée assez complexe, mais pas pour les raisons que vous pourriez penser. La copie d'une surface sur une nouvelle surface est relativement facile, le principal problème provient de la manière dont nous stockons les données cartographiques dans un fichier de sauvegarde.

Factorio crée en interne des listes de chaque entité, élément, décoration, etc. au démarrage et génère un mappage du nom du texte sur un ID entier. Ceci est appelé en interne un mappage d'ID et offre plusieurs avantages à la fois pour l'exécution et pour l'enregistrement / le chargement du jeu. Le principal avantage est le suivant : nous (et les modders) n'avons pas besoin de suivre et d'attribuer manuellement des identifiants à tout ce qui est ajouté au jeu. Le moteur le fait automatiquement. Quiconque est familier avec Minecraft moddé avant la version 1.7 peut attester combien la gestion manuelle des ID peut être pénible et avec quelle facilité il peut y avoir un dysfonctionnement.

Étant donné que des modifications internes ou l'ajout / la suppression / la modification de mods permettent d'ajouter et de supprimer des éléments, le mappage d'ID change souvent. Parce que cela peut changer, nous devons l'inclure dans le fichier de sauvegarde pour savoir ce que c'était quand la sauvegarde a été créée. Lorsque Factorio enregistre la carte, l'une des premières choses écrites dans le fichier de sauvegarde est cette correspondance.

La façon dont cela fonctionnait auparavant était la suivante : au moment du chargement - le jeu tenterait de restaurer le mappage d'ID correspondant à la sauvegarde de son chargement. Tout ce qu'il ne trouve pas signifie que la chose n'existe plus, et que tout ce qui est nouveau est ajouté. Cela a conduit à des sauvegardes ayant différents mappages d'identifiants pour le même ensemble de mods en fonction des étapes prises pour créer ce fichier de sauvegarde. Cela a fonctionné pour la plupart mais il y a eu quelques petits problèmes :

- Chaque fois que quelque chose a été supprimé, elle a été signalée en définissant l'identifiant à cet emplacement dans le mappage à 0.
- Les identifiants supprimés ne pouvaient pas être réutilisés jusqu'à ce que l'enregistrement soit complètement chargé, enregistré et chargé à nouveau (entraînant des lacunes dans les IDs).
- Différents fichiers de sauvegarde pourraient finir par utiliser différents identifiants, même si vous utilisez le même jeu de mods.

### ID mapping (Old system)

Before load	After load
0x0001 = "iron-plate"	0x0001 = "copper-cable"
0x0002 = "copper-plate"	0x0002 = "iron-stick"
0x0003 = "mod-item-1"	0x0003 = nil (removed)
0x0004 = "mod-item-2"	0x0004 = nil
0x0005 = "stone-brick"	0x0005 = "iron-plate"
0x0006 = "copper-cable"	0x0006 = "stone-brick"
0x0007 = "iron-stick"	0x0007 = "copper-plate"
	0x0008 = "new-mod-item"

Avec le nouvel éditeur de carte qui veut prendre n'importe quel fichier de sauvegarde et l'importer dans votre jeu en cours d'exécution, le fait que le mappage d'ID puisse être différent était un problème. Pour ajouter à la complexité, nous avons également un système de migration en place qui vous permet de dire au jeu "Je veux changer toutes les entités / éléments / ... nommés \*A\* en \*B\*". Ce système de migration fonctionne aussi pour enlever la source, car autrement cela ne serait pas utile.

Après plusieurs faux départs (thème commun de ces retouches complexes), je suis arrivé aux simples exigences suivantes :

- Les informations d'identification seront attribuées une fois après le chargement des mods - après cela, ils ne sont jamais autorisés à changer pendant la durée de vie du programme
- Lorsqu'une carte est chargée au lieu de restaurer les mappages d'ID qu'elle utilisait, elle créera une migration de l'ancien identifiant vers le nouvel identifiant (s'il existe toujours).
- Le suivi de ce qui a été supprimé se fait via un système différent au lieu de traiter un identifiant « 0 » en tant qu'identifiant supprimé.

## ID mapping (New system)

Before load

```
0x0001 = "iron-plate"  
0x0002 = "copper-plate"  
0x0003 = "mod-item-1"  
0x0004 = "mod-item-2"  
0x0005 = "stone-brick"  
0x0006 = "copper-cable"  
0x0007 = "iron-stick"
```

After load

```
0x0001 = "copper-cable"  
0x0002 = "copper-plate"  
0x0003 = "iron-plate"  
0x0004 = "iron-stick"  
0x0005 = "new-mod-item"  
0x0006 = "stone-brick"  
Removed:  
0x0003 = "mod-item-1"  
0x0004 = "mod-item-2"
```

Au cours de ce réaménagement, j'ai découvert que nous faisons beaucoup de travail supplémentaire (certains causant même des bogues) pour restaurer l'ID du fichier de sauvegarde et que je pouvais tout supprimer maintenant que les ID migraient simplement aux valeurs correctes chaque fois qu'un fichier de sauvegarde est chargé. J'ai même par inadvertance corrigé un bogue que quelqu'un avait signalé en cas de plantage lors du chargement d'un fichier de sauvegarde spécifique à l'aide d'un scénario en raison de cette révision.

Dans l'ensemble, le système est maintenant plus simple et n'est plus "bizarre". Le nouvel éditeur de carte peut importer n'importe quel fichier de sauvegarde que le jeu est capable de charger et tout fonctionne.

### HR vers (Albert)

Comme vous le savez peut-être déjà d'un [article](#) précédent, nous travaillons avec Ernestas sur la version haute résolution des ennemis. Ceci est une excellente excuse pour affiner le concept. Cette semaine, nous voulons montrer le nouveau look des vers.



L'intention est de conserver ce que nous avons dans la version précédente, mais de souligner sa personnalité et son comportement. Cette nouvelle version tente d'être plus agressive, puissante et dégoûtante, tout en agissant de manière plus nerveuse et agitée. Les "doigts" sur sa tête servent à creuser des tunnels, ce qui n'a pas vraiment été expliqué auparavant et qui aidera également à exprimer le caractère du ver.

Maintenant, nous travaillons sur les animations, essayant de mettre l'accent sur ces concepts, le ver est essentiellement une créature de pure haine et d'instinct de tueur. Avec les effets sonores, j'espère que ça va être un plaisir de les tuer.

Comme toujours, dites nous ce que vous en pensez sur le [forum](#).