

FACTORI:

Friday Facts #314 - 0.17 stable

Klonan, Kovarex

2019-09-27

Bonjour, Techniquement, ce post est le π FFF, mais malheureusement on ne peut penser à rien de spécial à faire ... peut être quelqu'un peut faire un gâteau avec de la logique... qui calcule π ?

0.17 stable

On a rendu la 0.17.69 stable ce Mardi passé. Il parait que tout va bien, pas d'avalanche de crash, seulement quelques mails du support technique à propos de la mise à jour des pilotes vidéo.



[cliquer ici pour voir](#)

Au-delà de la stabilité, essentiellement, aucun travail de développement n'a eu lieu cette semaine ; quasiment tout le monde est parti en vacances (j'écris ça dans l'aéroport en attendant d'embarquer direction Londres pour un mariage). On espère que le début de la semaine prochaine, avec le retour au travail, on se remettra à bosser sur la prochaine mise à jour avec le plein d'énergie.

En vrai, je pense être un peu optimiste quand je dis ça, mais je pense qu'on embarque dans une aventure excitante avec l'équipe. Avant, on faisait toujours un cycle ayant toute l'équipe dans le développement pour quelques mois. Puis, on toute l'équipe se met à la correction de bugs pour quelques mois. Cette approche binaire est ce qui donne une phase "stable" et "expérimentale". Ce qui ne veut pas dire que la correction de bugs s'arrête une fois que l'on est stable (au contraire), mais cela a été la stratégie générale.

Ce qu'on planifie, si la logistique le permet, est d'avoir de petite mises à jour contenant seulement quelques nouveautés. Ceci afin d'avoir un cycle mixte de manières différentes : Par exemple, pendant qu'un dev. travaille sur une nouveauté, l'autre sera en correction de bugs des nouveautés déjà mises en place. Ceci n'est réalisable que si la fréquence des mises à jour est assez élevée.

Cette nouvelle structure a été longuement pensée et on considère que c'est le moment de l'essayer. Avec l'interface utilisateur graphique, on doit vraiment produire des itérations et recevoir un retour utilisateur rapidement. La manière de travail traditionnelle ne permet de recevoir une retour que plusieurs mois après qu'une nouveauté fut développée. On doit donc dédier plus de temps à celle-ci en nous réorientant au code et comment il a été écrit. Si le retour été plus rapide, on ferait des modifications d'une manière plus efficace.

De plus, Je pense que c'est psychologiquement plus efficace de travailler sur un mix de correction de bugs et développement. C'est juste une théorie, mais je me base sur mes observations.

Travail de développement, nouveauté, ... c'est généralement un processus créatif à long terme. Les nouveaux systèmes doivent être créés à partir de la pensée. Les idées d'implémentation doivent être évaluées et déterminées et impliquent donc beaucoup de traitement de fond. Le développement des nouveautés a toujours plus d'espace pour s'étendre et il est difficile de dire "C'est fini". Il est aussi assez subjectif. Donc, parfois il est difficile de savoir si c'est "du bon travail".

La correction de bugs quant à elle est assez rapide et plutôt est défi à relever. C'est comme si on jouait au Cluedo et qu'on complétait l'histoire. Traquer un problème dans un moteur de jeu occupe la partie logique du cerveau en essayant d'assembler les pièces et remonter à la source. Généralement, le bug a une "condition de victoire" assez claire et que vous pouvez fermer le jeu et reposer votre esprit. Le résultat est une correction de bugs qui est strictement objective. Donc, on peut raisonnablement dire que c'est "du bon travail"

Donc, ces deux parties du travail sont, en quelques sortes, très distinctes et séparées : Le Développement est un processus créatif à long terme ; La correction de bugs est un processus logique à court terme. De tout ça, mon raisonnement est que ce concentrer sur un seul pour une longue période de temps mène à une fatigue mentale ; Et que un équilibre dans le travail nous gardera plus motivé et productif. Pour résumé, Le développement permet au circuit de correction de bugs de se reposer et la correction de bug permet au batterie de créativité du développement de se recharger.

Il y a aussi des raisons pragmatiques à cause desquelles je pense que les mises à jour rapides vont faire avancer les choses de manière plus fluide :

- Les corrections de bugs après la stabilisation sont rapidement fait.
- Les bugs arriveront de manière progressive au lieu d'arriver comme une grosse vague déferlante.
- Il y aura moins de blocages ; quand une nouveauté n'est pas fini, on la remet à la mise à jour suivante.
- Le retour sera plus précis et plus facile à évaluer.

Au début de l'année, j'ai lu le livre *"Rolling Rocks Downhill"* par *Clarke Ching*. C'est un livre qui parle de la gestion de projets de programmation. Il est très plaisant à lire et m'a donné de l'inspiration pour essayer d'optimiser nos efforts de développement. À l'époque, on venait de finir la 0.17. Donc, il n'y avait pas de quoi changer les choses. Maintenant que la 0.17 est stable, et avec ça, le cycle traditionnel complété. Il y a une opportunité pour la nouvelle approche. Je pense que je vais relire le livre la semaine prochaine...

World of Warcraft a de bonnes idées

Comme vous le savez probablement, j'ai passé un temps considérable à jouer à WoW classique ces dernières semaines. Je trouve ça très intéressant que beaucoup de gens (moi inclus) trouve cette version bien meilleure que les autres. Il y a plusieurs raisons à cela. Cela mérite une petite analyse car il vaut toujours mieux apprendre des erreurs des autres. En disant ça, je veux dire que je ne veux pas passer les 10 prochaines années à développer une meilleure version de Factorio pour découvrir que les anciennes étaient beaucoup mieux.

Le sujet qui lie tout ça est "rendre les choses plus aisées n'est pas forcément toujours mieux". Il y a beaucoup d'activités dans la vie et des jeux que l'on voudrait enlever ou rendre plus aisés. Mais, on ne voit pas vraiment les bienfaits de cela. Quelque chose telle que la conception d'un ascenseurs pour gagner du temps et de l'effort et ce sentir mal car on bouge moins souvent.

Une des choses dans WoW bien illustrée par cette [vidéo](#), quand les gens ont à ce regrouper pour aller dans un donjon, ils devaient le faire manuellement. Il devait spammer des demandes d'ajout sur les canaux, demander aux guildes ou trouver des amis pour jouer avec. Dans le WoW moderne, Il y a un "trouveur de donjon" qui permet aux joueurs de sélectionner un rôle et automatiquement les téléporter au donjon avec des partenaires appropriés. C'est magnifique, n'est ce pas ? En vérité, ça enlève tellement de choses que beaucoup de gens pensent que c'est la pire invention dans le jeu.

- Quand il est difficile de trouver un groupe, la valeur d'un groupe déjà en place est plus grande. Ce qui veut dire que les gens sont plus enclins à essayer de surmonter les problèmes en combattant au lieu de simplement quitter et relancer le donjon avec d'autres gens. Ceci rend les choses est bien plus intéressantes car cela crée plus de défi où une situation initiale non-optimale ou bien un noob dans un groupe force les gens à apprendre les uns des autres, ou juste à essayer de surmonter les problèmes de manières créatives.
- Quand vous n'êtes pas téléporté au donjon mais devait y aller par vos propres moyens, ça garde l'immersion dans le jeu. Le monde est alors exploré et non pas juste une machine arcade où on appuie sur un bouton pour jouer.
- Et comme dans le point précédent, aller à un endroit requiert plus d'efforts, donc la valeur de faire le donjon est plus importante. Et généralement, avoir plus de donjon effectués en même temps ne donne pas un sentiment d'accomplissement car faire un donjon devient mécanique.

Un autre grand exemple est la mise à l'échelle de niveau. L'idée est encore une fois intéressante à rendre plus pratique. Avec la mise à l'échelle des niveaux, les monstres s'adaptent parfaitement à votre niveau dans la zone où vous vous trouvez, vous pouvez donc effectuer des quêtes à tout endroit sans devoir éviter les zones de niveaux trop élevés ou trop bas par rapport à votre niveau. Mais, je considère ceci comme la peste de n'importe quel jeu. Au lieu de zones, leurs niveaux et votre niveau signifiant quelque chose, et le progrès étant clairement visible à tous les niveaux, tout disparaît soudainement. Le monde n'est pas un endroit à explorer, mais plutôt un lieu semblable à une arcade qui se résout autour de vous. Votre niveau ne veut pas dire grand chose, étant donné qu'il est trop bas, ce n'est plus un problème. Les moments où vous devez fuir des zones où vous êtes trop faible pour revenir plus tard sont l'un des plus importants des jeux de rôle, et du coup, ils ont disparu. Les progrès que vous voyez dans les jeux sont comme si vous regardiez depuis la fenêtre d'un train. Si le train bougeait et que le paysage bougeait de la même façon, vous ne sentiriez pas que vous bougiez du tout.

Il y a beaucoup plus de leçons apprises en jouant. Non seulement les erreurs commises dans la version moderne, mais aussi, évidemment, la qualité du jeu est réduite. Comment pourraient-elles être liées à Factorio ? Je crois que sur le plan plus profond, il pourrait y avoir beaucoup de parallèles.

- Obtenir la première monture au niveau 40 pour 100 or ? Il faut beaucoup marcher avant de l'obtenir et cela peut parfois sembler ennuyeux. Mais quand vous l'obtenez, cela signifie BEAUCOUP. Semblable à la façon dont c'est une avancée majeure d'obtenir des robots de construction/robots logistiques/une armure de puissance. S'y rendre n'est pas facile, et il y a certainement beaucoup de moments où vous/nous avons envie de les avoir plus tôt car ça pourrait être plus pratique. Mais cela diminuerait à quel point la mise à niveau est importante et précieuse.
- Nous pensions à un outil de construction de tapis convoyeurs, similaire à l'outil de construction de rail. Il ne ferait que relier les extrémités et même trouver des liaisons souterraines, etc. pour se rendre au point de destination. Ce serait super aisé. Mais tout à coup, résoudre le problème des tapis convoyeurs ne signifierait plus grand chose, et avoir l'usine de tapis convoyeurs serait un accomplissement bien plus modeste.

En tout, merci à vous tous pour cette année grandiose. Merci à tous nos amis sur le forum et toute la commu-

nauté qui nous ont aidé dans cette guerres des bugs de la 0.17 et comme toujours, dîtes nous ce que vous en penser sur notre [forum](#).