Forum: Moteur de jeu GameBlender et alternatives

Topic: Projets simples jeux basiques navires de guerre (essaie pour apprendre)

Subject: Re: Projets simples jeux basiques navires de guerre (essaie pour apprendre)

PostÃ© par: DaWaaaaghBabal

Contribution le : 17/10/2017 17:15:43

Citation :
Si c'est moins que shp donc c'est "tch" mais si c'est égal à shp sachant que c'est la seule valeur (rien au-dessus) alors c'est "shp"
Syntax error : "ça" undefined.

Si *quoi* est moins que shp, *qui* est tch ?
Par ailleurs : if shp - xcs : est équivalent à if shp != xcs :. Du coup, que choisir entre if shp - xcs: et if shp - xcs == tch: ? Ben j'en ai aucune idée, c'est toi qui sais ce que c'est censé faire. Je ne sais même pas ce que ça veut dire, shp ou xcs !

Ce qui m'amène à une règle essentielle en programmation : en pratique, on passe seulement 10% environ du temps à écrire du code, et les 90% restants à en *lire*. Pour comprendre comment il marche, comment on l'utilise, ou pourquoi il ne marche pas. Du coup, si tu as le choix entre un code facile à écrire et un code facile à lire, choisir toujours, toujours, toujours le code lisible et compréhensible. On ne doit pas avoir besoin de commentaires ou d'explications pour comprendre comment le code fonctionne (les commentaires disent comment on l'utilise et *pourquoi* on fait tel ou tel choix).

Donc, les noms de variables comme "tch", "xcs" ou "shp" sont à jeter. Utilise des noms explicites genre "shipHealth" à la place de "shp". Pour les deux autres je n'ai aucune idée de ce que ça pourrait vouloir dire

Citation :
Est-ce que celui que j'ai fais, à grand échelle (jeu vidéo) mènerait à prendre plus de place et de ressources?
Est-ce que tu as vraiment l'impression d'en être arrivé au point où les performances sont un problème ? Genre, est-ce que tu as un prototype fonctionnel dont les tests montrent qu'il rame ? Si non, tu te poses les mauvaises questions.

En général, les choix de structure sont plus importants que les choix d'écriture. Un mauvais algorithme écrit au petits oignons avec toutes les optimisations ésotériques sera toujours moins performant qu'un bon algorithme écrit avec les pieds. Et en pratique, tu n'as pas spécialement besoin de tout optimiser. Si ça se trouve, le jeu est suffisamment simple pour que la charge des calculs de gameplay soit négligeable même sur une machine médiocre.

Donc : assure-toi d'avoir un code qui marche et fait ce que tu veux. Ensuite, assure-toi que le code est propre, compréhensible, bien structuré, facile à débugger, maintenir, étendre, modifier. Ensuite seulement, optimise pour les performances.