Hello Loic :D

Voici le petit doc pour expliquer l'optimisation et aussi 2 ou 3 autres trucs

Table des matières

[Optimisation : 2](#_Toc134979505)

[Pathfinding : 4](#_Toc134979506)

[Multithreading : 4](#_Toc134979507)

[Epilogue de cette aventure : 6](#_Toc134979508)

# Optimisation :

J’ai retiré certaines boucles du style (image) quand le personnage est déjà connu pour éviter des boucles inutiles

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

Au lieu de faire CastleNightGame.Instance… à chaque fois, je crée une variable et je stock les infos à l’init. Par exemple j’ai stocké des textures ou bien des ReaderWriterLockSlim communs



Pareil avec la récupération de la texture qui se fait à chaque Draw

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Retirer les boucles dans la liste des personnages pour savoir si un perso est sur une tile

A la place, la tile c’est déjà si quelqu’un est à la même position

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

La liste des tilesToBeDestroyed ne se vide jamais, donc certaines tiles peuvent être dans la liste plusieurs fois, ce qui peut faire ralentir le jeu au bout d’un moment

Une image contenant Police, texte, Graphique, logo

Description générée automatiquement

Ajout d’une fonction pour récupérer les tiles ou les positions

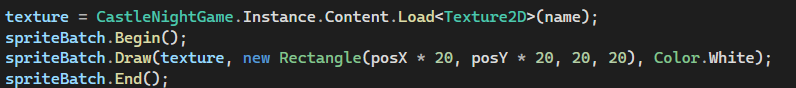
Une image contenant texte, capture d’écran, Police

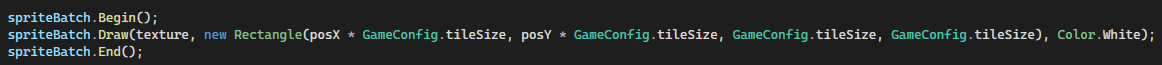
Description générée automatiquement

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Changer certaines valeurs brutes en variable utilisable à plusieurs endroits, comme la taille des tuiles





Le spriteBatch.Begin() et spriteBatch.End() n’est fait que dans le draw principal (celui de la map) et est envoyé en paramètre aux autres Draw

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Le timer du timeBeforeWeaponDrop est set à 10 au lieu d’utiliser la variable dans Gameconfig



Le timeBeforeDesctruction, qui est utilisé avec les tiles qui se cassent, est set avec le weaponDropTimer et dans la partie qui gère les armes

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

# Pathfinding :

J’ai utilisé le A\* pour l’exercice

Les personnages vont checker si une arme est sur la carte. Si oui, ils vont regarder l’arme la plus proche et s’y rendre.

Si en chemin, la tile où ils devaient se rendre n’est plus disponible, ils vont checker un nouveau chemin

Les tiles considérées comme non disponibles sont : les tiles avec un perso dessus, les tiles « trou » ou bien les tiles qui vont être détruites SEULEMENT si elles sont détruites dans moins de 0.5 sec (peu importe le multiplicateur de temps)

S’il n’y a pas d’armes sur la carte, les personnages vont prendre une tile random et essayer de s’y rendre

Une option a été désactiver mais si tu changes dans Gameconfig la variable « needRecalcule» ligne 28, à chaque drop d’arme les perso vont rechecker leur chemin si une nouvelle arme est plus proche (mais ça ralenti un peu le jeu)

# Multithreading :

Pour le multithreading, chaque personnage a son propre thread. Il y a aussi un thread qui check si tous les autres (ceux des perso) ont fini.

Chaque tile est lock et dès qu’un personnage veut se déplacer, il lock la case pour s’y déplacer sans danger

Les pv des perso sont aussi locks pour qu’ils ne se déplacent pas et se fasse attaquer quand il se déplacent

# Epilogue de cette aventure :

Je pense que l’opti n’est pas parfaite mais y a des trucs cool :D

Je suis plutôt content du pathfinding et du multithreading aussi (le multithread m’a fait peur un peu)

Merci !

Luca SCHIFANO