



Projet : Chips VS Virus

Objectif du projet:

Ce projet a pour objectif la réalisation du jeu de type tower defense, nommé Chips VS Virus.

Ce jeu consiste à générer des défenses (placées au début de la partie par le joueur) que nous appellerons “tourelles” (ici, ce seront nos chers amis les pokémons), afin de détruire les virus qui arrivent pour protéger la puce (chip). Soit les tourelles réussissent à tuer tous les virus, dans ce cas les chips gagnent, sinon les virus gagnent si l'un d'eux réussit à atteindre la chip en détruisant ou non toutes les tourelles. Le jeu s'arrête lorsque l'un des deux gagnent. Il est possible de générer différentes vagues de virus en fonction de la difficulté recherchée.

Démarche logique/mathématique

- - - - X

Pour commencer à réaliser ce projet, il a fallu le diviser en plusieurs petites tâches afin de ne pas se perdre dans tout ce qu'il y avait à faire. On pourrait aussi diviser notre travail en deux parties distinctes: la partie ascii et la partie graphique. En effet, il était important de réaliser une version ascii fonctionnelle afin d'avoir une base correcte sur laquelle s'appuyer pour la partie graphique.

Démarche de programmation

- - - - X

Afin de réaliser ce projet avec succès, nous avons commencé par réfléchir aux premières fonctions dont nous allions avoir besoin. Ainsi, nous avons commencé par recopier les structures Chips, Virus et Game fournies par le sujet puis avons commencé à faire apparaître un plateau. Ensuite, nous avons décidé de commencer par réaliser toutes les fonctions liées aux virus et leurs

caractéristiques (vie, vitesse...) ainsi que leur positionnement dans le plateau au fur et à mesure des tours. Par la suite, nous avons réalisé le chaînage des virus ce qui fut une des parties les plus difficiles et que nous avons eu besoin de reprendre légèrement par la suite puisqu'il était l'objet d'une erreur de segmentation. C'est aussi à cet endroit là qu'il a fallu faire attention à la gestion de la mémoire, nous avons donc fait des mallocs pour limiter la perte, notamment dans `insertion_virus()`. Une fois cette partie terminée, nous avons réutilisé ces structures afin de créer les fonctions correspondant aux tourelles puisqu'elles étaient pour la plupart similaires. Enfin, la dernière chose à faire pour que la version ascii fonctionne était de rajouter aux tourelles et aux virus la possibilité de faire des dégâts à l'adversaire et de la tuer. Ainsi, il a fallu trouver un moyen de supprimer les virus tout en conservant le bon ordre de chaînage pour les virus. Il ne restait donc plus qu'à faire une fonction correspondant aux conditions de victoire pour que le joueur gagne ou perde à la fin de la partie.

Difficultés rencontrées:

- A un moment du projet, Lucien pouvait lancer le jeu mais Joanna avait une erreur de segmentation inexplicquée. Nous avons finalement réussi à trouver la source de l'erreur et la corriger en utilisant Valgrind.
- L'ordinateur de Lucien a crash lorsqu'il a voulu allouer de la mémoire :(