

# Chips Vs Virus



# Documentation technique

## Structure de la source du projet :

Dans le dossier du projet on trouve deux dossier un pour la documentation le deuxième src qui contient les sources du notre projet dans ce dernier, on a 3 dossiers un pour les images utilisées dans le projet notamment les images des tourelles, virus et arrières-plans un autre dossier pour les sons et la police utilisée dans le projet.

Ainsi que les fichiers principaux :

1. Le fichier « utils.c » qui contient toutes les fonctions primaires du projet (insertion – suppression – intialisation – update des infos des virus/chips ...) avec le fichier utils.h qui contient la définition de ces fonctions ainsi que les typedef et les includes nécessaires.
2. Le fichier « ascii.c » dans ce fichier on a toutes les fonctions responsables du mode ascii notamment l’affichage du plateau en terminal la boucle main du jeu en ce mode avec le fichier ascii.h qui contient la définition de ces dernières.
3. Le fichier « graphique.c » contenant tous ce qui est en relation avec le graphique dessin des points, vagues, plateau, tourelles et boutons ainsi que la boucle principale de ce mode dans une autre fonction avec le fichier .h aussi.
4. Le fichier « main.c » qui contient la main du projet dans ce fichier on détecte seulement le mode du jeu choisi par l’utilisateur et selon son choix on fait appel soit au fonction responsable du graphique ou bien celle du terminal.

En ce qui concerne la compilation de ces fichiers un fichier **makefile** a été crée contenant les commandes nécessaires pour compiler chaque fichier séparément de cette forme « clang -c graphique.c » en faisant ça on produit le fichier « graphique.o » après on produit le programme exécutable du jeu qui s’appellera « app » en compilant tous les fichiers déjà cités avec la commande « clang -Wall -std=c17 main.o graphique.o utils.o ascii.o -o app ».

Ensuite, pour exécuter le programme on exécute le fichier « app » avec la commande « . /app nomFichier.txt modeJeu (-g ou -a) ».

### Explication des structures ajoutées :

Les deux structures qu'on a rajoutée était la structure « ChipsGraphique » et la structure « Boutton »

Les deux structures avaient pour but de stocker les coordonnées de l'emplacement des tourelles et des boutons.

Cela nous a facilité la détection des clics vu qu'on avait déjà stocké toutes les informations dont on a besoin.

### Fonctions principales :

Les fonctions les plus importantes présentes dans le programme sont présentes dans le fichier « utils.c » vont donc être décrites brièvement Dans leur fonctionnement

- Les fonctions d'allocations des tourelles et virus « Virus \*alloueVirus (int tour, int ligne, char type), Chips \*alloueTourelle (char choix) » c'est les 2 fonctions responsables d'allocations de nos types et d'affectation des champs selon les paramètres.
- Les fonctions d'insertions des tourelles et virus « void insereVirus (Game \*jeu, Virus \*virus, Virus \*\*tabVirusLigne), void insereChips (Game \*jeu, Chips \*chips) » c'est les 2 fonctions responsables d'insertion de nos types selon le chaînage défini, ainsi que les fonctions de suppression, de libération de liste de la mémoire et de suppression des types morts.

Ensuite, on s'intéresse aux fonctions responsables de l'update des infos de nos types selon plusieurs conditions qui se réunissent dont on cite :

- Les fonctions « Virus \*virusMemeCase (Game \*jeu, int lig, int col), Chips \*chipsMemeCase (Game \*jeu, int lig, int col) » sont responsables de nous renvoyer le virus ou chips qui se trouve dans une même case qu'on passe au paramètre cela nous facilite la détection des collisions.
- Les fonctions « void updateInfoVirus (Game \*jeu), void updateInfoChips (Game \*jeu, char mode) » s'occupent de l'update de la vie des virus et des chips selon plusieurs conditions que détectent d'autres fonctions comme la collision des deux types, les types des virus et des chips en prenant en considération les cas exceptionnels ou les tourelles ont des pouvoirs spécifiques.

- La fonction « void moveVirus (Game \*jeu) » c'est la fonction principale responsable des mouvements des virus (par convention on considère que la position des virus s'augmente selon leur vitesse par 1) et ça prend en considération le fait qu'un virus va diminuer sa vitesse au cas où la vitesse de celui qui suit est plus grande que la sienne.

Les autres fonctions sont des fonctions principales aussi mais ils s'occupent plus de l'affichage et de la saisie de l'utilisateur par exemple la fonction « void choixTourelle (Game \*jeu) » et la fonction « void affichePlateauAscii (Game \*jeu) ».

Pour la version graphique du jeu, toutes les fonctions responsables du bon fonctionnement du jeu étaient déjà prêtes, on s'occupait juste de l'affichage du plateau et du menu et de plus la détection des clics d'utilisateur et faire des actions selon ça, par exemple :

- « Void dessinTourelle (int money, ChipsGraphique \*tabImage, MLV\_Font \*police), void dessinPoints () » une fonction qui dessine les tourelles selon leur type qui est conservé dans un tableau de structure des tourelles et l'autre elle s'occupe du dessin du plateau.
- « Void detecteClick (Game \*jeu, Bouton \*btn, ChipsGraphique \*chipsImage, int largeur, int longueur, char \*nomFic, MLV\_Font \*police); » Cette fonction gère tous les clics de l'utilisateur et fait des actions selon ce dernier.

## Déroulement du travail:

Durant le développement de ce projet, on a beaucoup appris sur les listes chaînées et comment ça fonctionne dans un jeu. Grâce à nos efforts et l'aide des professeurs on a pu réussir à réaliser notre projet, on a dû rencontré quelques difficultés au début mais après avoir réfléchi beaucoup dessus et essayer plusieurs fois on a réussi.