

Pattern du projet :

Pour commencer ce projet, je suis partie du tp3 et ai gardé pratiquement le même pattern. J'ai gardé les classes « MyControlEngine », « MyGameEngine », et « MyGraphicEngine » et créé une classe pour chaque élément de jeu : Vaisseaux, Missile, Astéroïdes, Damier, Variables de jeu telles que le score , etc. et les ai stockés dans des vecteurs , transmis à chacune des classes en ayant besoin.

Héritage :

Les classes Missiles et Asteroides héritent d'une même classe « ObjetsAandM », car elles ont les mêmes attributs et méthodes.

Les différents types de vaisseaux « lowV », « mediumV » et « speedV » héritent de la classe « Vaisseaux ».

Les différents types d'astéroïdes « lowAsteroïde », « strongAsteroïde » et « fastAsteroïde » héritent de la classe « Asteroides ».

Pour finir, les classes MyControlEngine, MyGameEngine, et MyGraphicEngine héritent d'une classe « myEngine ».

Création des classes :

L'implémentation a commencé par la création du damier. Une case d'abord, qui changeait de couleur lorsqu'on cliquait dessus, puis plusieurs colonnes. A suivi la création d'une classe « Vaisseaux » qui permettait de poser un vaisseau par case – uniquement - lors du click du joueur.

Après cela, j'ai implémenté une classe « Missiles ». Les missiles sont créés dans la méthode « draw() » de la classe Vaisseaux. A chaque ticks(), ils avancent d'une certaine distance en fonction du type de vaisseau qui les lance. Si c'est un « LowV » le missile avancera de $0.025 * 8$ à chaque appel de ticks(). Si c'est un « MediumV » il avancera de $0.025 * 10$. Si c'est un « SpeedV » il avancera de $0.025 * 12$ par ticks.

- Ayant eu de nombreux problèmes avec la création de missile à un certain temps T en utilisant la variable « GetTickCount » (Mon vaisseau envoyait 4 missiles d'un coup, puis 3, puis 2), j'ai décidé d'utiliser une variable « nbTicks » qui s'incrémente à chaque appel de la fonction tick(). J'ai ensuite remarqué grâce à la variable « GetTickCount » qu'un missile mettait entre 2300 et 2700 ticks avant d'arriver au bout du tableau de 10 colonnes. J'ai donc décidé de prendre la valeur 2300, et de la diviser par la fréquence de missiles à envoyer, variable en fonction du vaisseau choisit, et d'envoyer un nouveau missile à chaque fois que « $nbTicks \% (2300 / \text{frequence}) == 0$ »
- Le vaisseau représenté en vert sur la fenêtre d'exécution est le « LowV ». Il a une fréquence de missile de 5 (voir ci-dessus pour l'utilisation des fréquences), une puissance de 1, et une vitesse de 8.
Le vaisseau en bleu sur la fenêtre d'exécution est le « MediumV ». Il a une fréquence de missile de 5, une puissance de 2, et une vitesse de 10. Le vaisseau en rouge est le « SpeedV », il a une fréquence de 5, une puissance de 4, et une vitesse de 12.
- Plus la puissance est augmentée, plus le missile enlève un grand nombre de vies à un astéroïde.

Pour l'IHM, j'ai créé une classe « Variable de jeu » qui contient tous les éléments de l'interface.

Elle gère tous les messages d'information tels que :

- La nécessité d'acheter des vaisseaux si le joueur n'en a plus.
- L'impossibilité de poser des vaisseaux sur le damier si l'utilisateur n'en a sélectionné aucun .
- La réussite ou l'échec d'une partie, et le nombre de pièces gagnées.

Cette classe prend en charge le calcul et l'affichage des variables de score, d'achat des vaisseaux, de pièces disponibles.

C'est dans cette classe que se gère le lancement de la vague d'astéroïdes et la fin d'une partie, grâce au bouton « Lancer la partie ».

Difficultés : Il y'a 4 niveaux différents (cf. voir dans le readme) que vous pouvez choisir sur le menu déroulant à gauche du bouton de lancement.

Pour utiliser le programme :

Pour lancer le programme, ouvrez la solution « Projet_C.sln » puis appuyez sur « Ctrl F5 ».

Une fenêtre graphique va s'ouvrir, et vous n'aurez plus qu'à suivre les indications affichées.

1. Pour acheter des vaisseaux, appuyez sur le bouton « + » du vaisseau que vous avez choisi.
2. Choisissez une difficulté
3. Posez vos vaisseaux sur le damier
4. Lancez la vague

<p>CONSEIL : Il est préférable de jouer au minimum 2 fois à un niveau, afin d'avoir assez de vaisseaux pour gagner la difficulté supérieure.</p>
