

Rapport

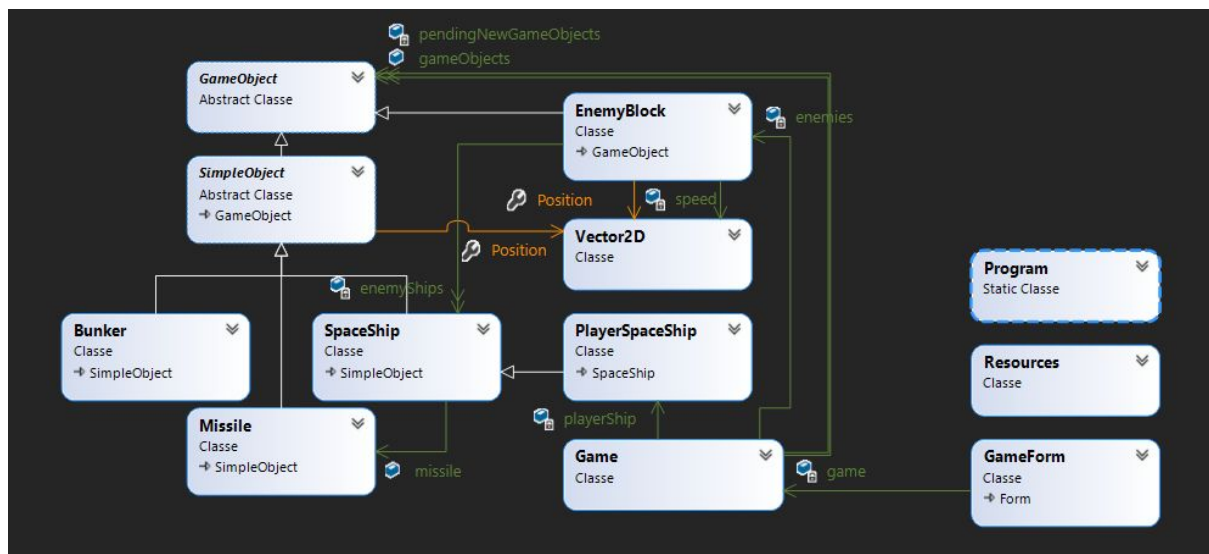
Projet - Space Invaders

Description du problème

Création d'une copie fonctionnelle du jeu Space Invaders en C#, en suivant la piste verte proposée.

Celle-ci doit inclure les entités "missile", "bunker", "canon" et "invader" du modèle, la représentation de ceux-ci à l'écran ainsi que les interactions entre les différentes entités, en plus d'un affichage sommaire.

Structure du programme



Le programme est structuré tel qu'illustré ci-dessus.

Une fois définie, la fenêtre instancie la classe **Game** pour créer un objet **game** représentant l'instance de jeu. Celle-ci instancie, dans son champ "ensemble d'objets", un vaisseau de joueur ainsi qu'un bloc d'ennemis contenant des vaisseaux ennemis, ainsi que des bunkers, qu'elle affiche et met fait se mettre à jour tour à tour.

Les sprites des entités sont tirés des ressources fournies.

Les vaisseaux génèrent, lorsqu'ils tirent, une instance de la classe **missile** qui avance verticalement dans le sens qui lui est attribué et vérifie si le missile entre en collision avec une entité d'un camp différent de son lanceur, collision gérée par l'entité le cas échéant.

Chaque entité a un nombre de vies limité, décrémenté lorsque celle-ci prend des dégâts, sauf dans le cas du bunker, qui ne fait que se désagréger.

Le vaisseau du joueur est contrôlé à partir du clavier, à l'aide des flèches de gauche et de droite ainsi que de la barre espace. Le joueur peut de plus mettre le jeu en pause (i.e. empêcher la mise à jour du jeu) en appuyant sur la touche "P".

Le bloc d'ennemis se déplace et déplace les vaisseaux qu'il contient horizontalement à un rythme constant jusqu'à rencontrer le bord de la fenêtre de jeu, moment où le déplacement se fait verticalement, en plus d'un alignement au bord de la fenêtre de jeu et d'un changement de sens accompagné d'une accélération du déplacement horizontale.

Le bloc intègre de plus une probabilité que chaque vaisseau tire individuellement toutes les secondes.

Lorsqu'une entité n'a plus de vie, celle-ci est détruite.

Si tous les vaisseaux ennemis sont détruits avant que le bloc d'ennemis n'atteigne la position du joueur, celui-ci gagne. Dans le cas contraire, il perd. Il peut alors relancer une nouvelle partie à l'aide de la barre espace.

Problèmes rencontrés

- Premier problème : erreur de signe, impliquant des déplacements du canon dans le mauvais sens. Aisément corrigé.
- Second problème : Détection pixel par pixel défectueuse. Problème dû à une mauvaise comparaison, résolu en comparant les valeurs ARGB des pixels au lieu des objets.
- Dernier problème : réinitialisation de la partie compromise. Résolu en créant une procédure "InitialiseGame()" (ré)initialisant le jeu dans son état initial.