Journal de bord : Space Invaders

Jour 1

J'ai crée un compte GitHub ce qu'il me permet de contrôler les versions du code. Cela m'a pris du temps à apprendre les différentes commandes Git.

Voici mon repo GitHub : www.github.com/watsum08/Les-envahisseurs-de-l-espace

J'ai aussi du chercher et installer un IDE pour C# compatible sur Linux car Visual Studio Community ne l'est pas.

Jour 2

J'ai téléchargé tous les fichiers du projet.

J'ai crée la classe Vector2 qui permettra d'instancier des vecteurs de positions pour les objets.

J'ai crée la classe SpaceShip qui hérite de la classe GameObject(déjà existante) permettra d'instancier le vaisseau du joueur. J'ai crée ses attributs ainsi que ses méthodes demandés.

J'ai instancié «SpaceShip playerShip» et l'ai ajouté dans la liste des objets du jeu.

J'ai testé le jeu et le vaisseau playerShip s'affiche correctement.

Jour 3

J'ai cherché le code nécessaire pour implémenter les touches du clavier. Ce qui permettra de déplacer le vaisseau du joueur en appuyant sur les flèches.

J'ai écris un code dans la méthode Update() du vaisseau joueur qui évitera le vaisseau de sortir du cadre de jeu.

J'ai crée une nouvelle classe, Missile qui hérite de GameObject, qui permettra au joueur de tirer.

J'ai crée une méthode Shoot() dans la classe SpaceShip qui permettra de tirer, donc d'instancier un Missile. Si le joueur appuie sur la touche Espace, cette méthode est appelée.

Jour 4

Refactoring de la classe SpaceShip et Missile… J'ai crée une nouvelle classe SimpleObject qui hérite de GameObject. SpaceShip et Missile héritent maintenant de la classe SimpleObject.

En gros, j'ai du presque tout modifié pour que mon projet corresponde à la nouvelle architecture de jeu demandée.

Jour 5

J'ai cherché comment afficher du texte au jeu, et j'ai trouvé la fonction drawString() de la classe Graphics.

J'ai crée une nouvelle Enum ayant deux état : Pause et Play.

J'ai implémenté un bool dans la méthode Update() de chaque objet qui vérifiera l'état de jeu. Si l'état est Play, l'objet se déplace, s'il est en Play, rien ne se déplace.

Jour 6

J'ai crée une nouvelle classe Bunker qui permettra au joueur de se couvrir. J'en ai instancié 3 juste au dessus du joueur.

J'ai fait des recherches sur la collision et trouvé les conditions qui vérifie si deux rectangles s'intersectent.

Jour 7

J'ai implémenté la détection de collision entre le rectangle du missile et le rectangle du bunker.

J'ai trouvé les méthodes getPixel() et setPixel() de la classe Bitmap. Avec ces méthodes je peux vérifier si le Pixel à une certaine position de l'image (bunker) est blanc ou noir, selon sa transparence Alpha. *255 = noir, 0 = blanc*

Après des heures d'essais j'ai réussie à changer la couleur du pixel du bunker qui est touché par le missile. Et à chaque pixel que le missile touche, il diminue son nombre de vies par un.

Lorsqu'ils rentrent en collision, les missiles se détruisent et détruisent aussi petit à petit les bunkers. Tous marche bien.

J'ai aussi affiché le nombre de vies du joueur en bas a gauche du cadre de jeu.