

DURON
GAETAN

DOSSIER-PROJET ISN

The lone spaceship



Participants au projet: _Duron Gaetan
_Burillard Stanislas

A/ Choix du sujet :

Intéressés par le game design, nous avons choisi de programmer un shoot them up (jeu de vaisseau à défilement). Space invader est à l'origine de ce genre.

Nous avons adopté ce gameplay (mécanique de jeu) car il permet de rapidement aboutir à un projet fonctionnel. Ainsi nous avons pu nous assurer d'avoir un programme à rendre, même si nous avons rencontré des difficultés. En effet, l'avantage du "shoot them up" est d'être très vite opérationnel avec des fonctionnalités de base (vaisseau du joueur + un type d'ennemi), on a ensuite la possibilité d'améliorer le programme en fonction du temps que l'on a, au fur et à mesure (ajout d'ennemis, nouvelles armes, complexification des trajectoires, level design...).

Le but de notre projet est simplement de créer un jeu amusant. Cependant l'enjeu majeur est de réussir à trouver un équilibre entre le temps disponible pour réaliser le projet, la complexité du jeu et les connaissances supplémentaires (programmation orientée objet) que nous devons acquérir pour finir à temps et que le jeu soit plaisant.

B/ Cahier des charges

1) Le jeu

But : Finir le niveau en tuant le boss, finir le niveau rapidement pour augmenter le nombre de point en fin de partie.

Comment jouer au jeu : - Un joueur

- Gestion du vaisseau à la souris
- Touche espace pour tirer

Principe du jeu : Le vaisseau du joueur est en bas de l'écran. Les ennemis viennent du haut de l'écran et vont vers le bas de l'écran.

Les ennemis tirent vers le bas de l'écran.

Le joueur doit éviter les tirs ennemis et ne pas entrer en collision avec les ennemis. Le joueur doit tirer sur les ennemis pour les tuer et avancer dans le niveau.

Si le joueur est touché, il perd une vie sur ses trois vies.

Le score du joueur dépend de sa vitesse et sa précision.

2) Les graphismes du jeu

Graphismes simple en pixel-art

sprites:

- vaisseau du joueur x3	- logo du temps
- vaisseaux ennemi x6	- logo du nombre d'ennemis tués
- explosions x6	

L'image des balles tirées et des étoiles défilants en background sont des rectangles de couleur.

3) Musiques et effets sonores

Une musique en rapport avec le theme du jeu et cohérente avec le style graphique. Ils faut donc une musique chiptune et dynamique.

Pour les effets sonores, il faudra des bruits de laser et d'explosion.

C/ Environnement de travail

Pour la programmation:

- langage utilisé, Python
- module pour l'interface graphique, Pygame
- IDE utilisé pour l'écriture du code, IDLE et Spyder

Pour les images:

- Photoshop CS5

Pour apprendre Pygame:

- Chaine youtube rootofthenull
- site de pygame
- documentation de pygame

Pour organiser le code:

- Apprendre la programmation objet

D/ Répartition des taches

Ce qui devait être fait pour commencer à programmer le jeu:

<u>Gaetan Duron</u>	Stanislas Burillard
Design du jeu	Arrière plan
Gestion de l'organisation du code	Mouvement des ennemis
Vaisseau du joueur(déplacement, tirs, sprites)	Affichage des textes
(Ajouter les collisions...), etc	(Ajouter le temps...), etc

Ce qui a finalement été fait:

<u>Gaetan Duron</u>	Stanislas Burillard
Design du jeu	Affichage des textes
Gestion de l'organisation du code (OOP)	
Vaisseau du joueur (déplacements, tirs, sprites, collisions, vies et changement de sprites en fonction du nombre de vies, sons, explosion)	
ennemis (déplacements, tirs, sprite, collisions, vies, différences entres les ennemis, génération aléatoire des ennemis, ordre d'apparition, sons, explosions)	
arrière plan (génération aléatoire d'étoile)	
musique (recherche et impémentation)	
écran de mort - game over	

écran de succès (calcul du score)	
design du niveau (niveau de difficulté progressif, invention des différents comportements d'ennemis par exemple les trajectoires sinusoidales, création de hasard)	
Gestion de la fenêtre (frame par seconde, coordination des appels de fonctions, actions en fonction des commandes entrées par le joueur, affichage de tous les sprites, tests de logique, propriétés de la fenetre)	
Score et temps (compter le score, compter le temps qui passe et le convertir en seconde et minutes)	

Nous avons collaboré via Github et par mail pendant le projet.

E/ Réalisation

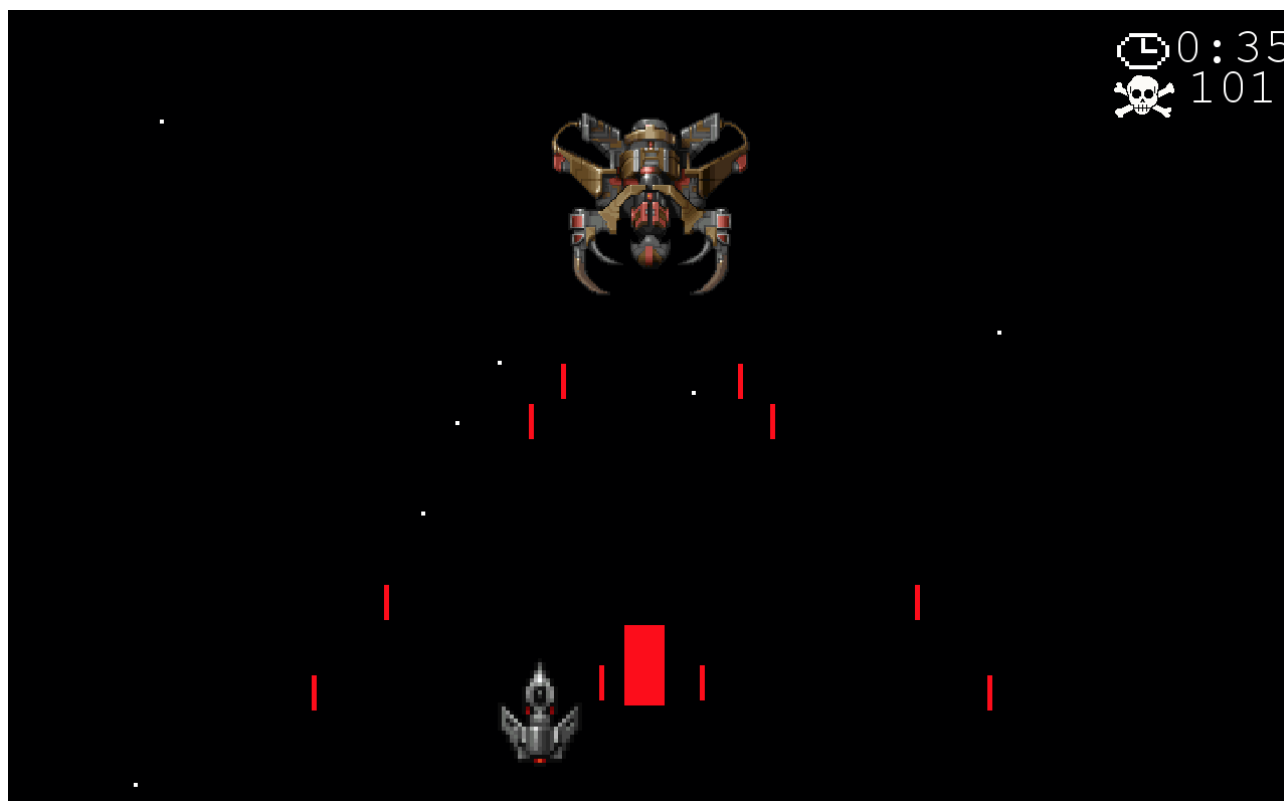
Les difficultés que j'ai rencontré:

- Organisation du code. J'étais vite perdu dans mon code après 200 lignes. Puis j'ai finalement opté pour la programmation objet qui paraissait plus claire et plus simple à utiliser sur un code plus long.
- J'ai mis un peu de temps à appréhender la programmation objet. Mais ensuite la programmation est devenue de plus en plus agréable.
- J'ai longtemps eu des problèmes avec pygame qui ne fonctionnait pas chez moi. Ce qui m'a beaucoup ralenti.
- A cause de l'installation tardive de pygame et mon apprentissage tardif de la programmation orientée objet, je n'ai malheureusement pas eu le temps d'aider mon binôme à comprendre le code. J'ai

seulement pu lui transmettre les ressources qui m'ont permis d'apprendre.

F/ Intégration et validation

Capture d'écran de la version finale du jeu :



Le jeu est jouable et amusant

G/ Bilan et Perspectives

Le jeu respecte le cahier des charges, de nombreux ajouts ont même pu être réalisés, des variétés d'ennemis et différentes attaques.

Mais d'autres ajouts sont encore possibles:

- Menus de départ
- Bouton replay
- Liste des scores - Leaderboard
- Différentes attaques pour le joueur
- Des animations

Bien que l'expérience de travail d'équipe fut difficile, créer un jeu m'a beaucoup plu. Cela m'a donné envie de continuer à programmer cet été, même si mon futur cursus n'est pas en lien avec l'informatique (cinéma d'animation). Ce projet a été très positif pour moi, il m'a également donné envie de me pencher sur le javascript pour une approche plus orientée web.