

# PROJET SPACE INVADER : MODE DIFFICILE

Elèves de NSI c'est la guerre. Le projet Space Invader traditionnel ne vous convenais pas. Vous saviez déjà quoi coder et où. C'est votre jour de chance : vous allez pouvoir me prouver votre dignité face au python POO.

## Enoncé

Vous allez devoir coder par vous-même un jeu *Space Invader*. Le joueur incarnera un vaisseau qui se déplacera dans la fenêtre. Celui-ci peut se déplacer, tirer, se défendre et caetera. Des hordes d'ennemis apparaitront en face. Un score sera affiché et augmentera en fonction de la progression du joueur / des ennemis tués.

### Fonctionnalités obligatoires :

- Le joueur : un ou plusieurs vaisseaux , qui ont une vitesse et une Fire rate
- Des Ennemis qui ont des PVs et un score (Il faut au moins 2 types d'ennemis différents)
- Des Bullets qui ont une vitesse / direction
- Le joueur a un système de vies
- Des hitboxes et collisions pertinentes.
- Un système d'apparition des ennemis (sous forme de vagues ou de niveaux, libre à vous)

### Fonctionnalités Facultatives (en faire 2 minimum, ça c'est obligatoire 😊)

- Des Sprites animés
- Un menu principal
- Système de power-ups / bonus
- Scoreboard / LeaderBoard
- Système de sauvegarde (persistance entre les parties)
- Multijoueur
- Système de sons
- Des défenses destructibles qui protègent le vaisseau
- Autres ?

## Le rendu

- Un rapport joli, mis en forme en PDF, expliquant :
  - o Les fonctionnalités réussites / ratées / pas finies
  - o Vos difficultés / ce que vous avez appris en python
  - o Des solutions que vous avez mis en place pour résoudre certaines contraintes
- Un dossier contenant l'intégralité de votre projet

Les modalités de rendu seront précisées plus tard. Essayez de finir le projet avant le **lundi 18 décembre**.

## Conseils

N'hésitez pas à vous familiariser avec le concept d'héritage des classes en POO (voir avec le prof / google). Cela permet d'avoir des classes qui utilisent du code en commun. Exemple : le joueur, des balles et les ennemis sont des entités. Tous peuvent-être dessinés à l'écran.

Code	/18
Organisation du projet (quel fichier gère quoi), partitionnement en modules	/2
Un code lisible et compréhensible (Documentation partielle /Commentaires)	/2
Fonctionnalités obligatoires	/8
Fonctionnalités facultatives	/4
Code maintenable et flexible : possibilité d'ajouter de futurs ennemis ou niveaux etc ...	/2
Bonus	/ ?
Game Play	/3
Jouabilité (Maniabilités, Lags)	/2
Effet WOW	/1
Rapport	/4
Total	/25