

# INFORMATIQUE

---

## R-TYPE

François COLAS ([francoiscolas@gmail.com](mailto:francoiscolas@gmail.com))  
Claire BELLARD ([claire.belliard@gmail.com](mailto:claire.belliard@gmail.com))

### CONSIGNES

- Pour le rendu, vous devrez mettre tous les fichiers de votre projet dans un dossier nommé **r-type**. Le dossier devra contenir :
  - Un fichier par niveau : **r-type-base.py**, **r-type-ext1.py**, etc..
  - Un dossier nommé **assets** contenant les ressources utilisées par votre projet.
  - Tout autres fichiers nécessaires au fonctionnement de votre projet.
- Le dossier devra être archivé dans un fichier nommé **r-type.zip**.
- Si l'archive ne contient pas de dossier nommé **r-type**, la note sera 0.
- Si les fichiers **.py** ne s'appellent pas comme demandé, la note sera 0.
- Les variables globales sont interdites.
- Les constantes globales suivantes sont autorisées :

```
GAME_FPS      = 30
GAME_WIDTH    = 800
GAME_HEIGHT   = 400

OBJ_PLAYER     = -2
OBJ_BACKGROUND = -1
OBJ_FIRE       = +1
OBJ_MONSTER    = +2
OBJ_BLAST      = +3

PLAYER_SPEED   = 10
BACKGROUND_SPEED = 4
MONSTER_COUNT  = 3
MONSTER_SPEED  = 7
MONSTER_SCORE  = 1
FIRE_SPEED     = 9
```

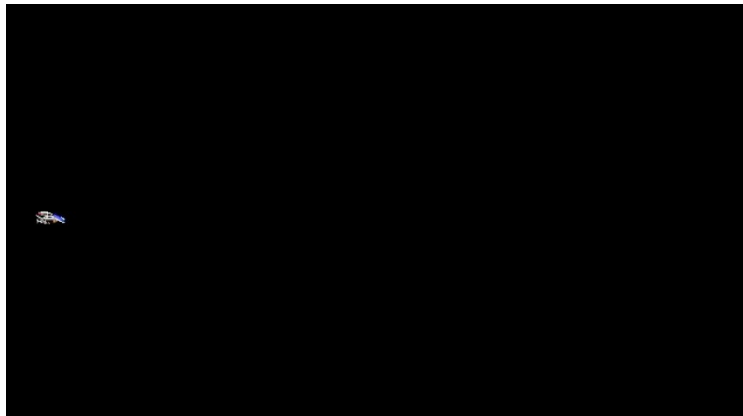
- Dans la notation, sont pris en compte : Les fonctionnalités, le découpage du code en fonctions, la propreté du code, le choix du nom des variables et fonctions.

## LA BASE /10

**Nom du fichier à rendre :** r-type-base.py

Pour valider cette étape, le programme devra fonctionner de la manière suivante :

- › Afficher l'avatar du joueur (image **assets/player\_1.bmp**). Au lancement, le joueur devra être centré verticalement et positionné à **32px** du côté gauche de la fenêtre.
- › Le joueur devra pouvoir déplacer son avatar à l'aide des flèches du clavier. Bien-sûr le joueur ne devra pas être capable de sortir de l'écran. Il sera déplacé de **10px** à chaque détection d'une touche appuyé.
- › La barre d'espace devra permettre au joueur de faire feu (image **assets/fire.bmp**). Les tirs devront être espacés d'un minimum de **35px**. Ils seront déplacés de **9px** à chaque tour de la boucle principale.
- › ▲ Les tirs doivent être supprimés de la mémoire lorsqu'ils sortent de l'écran.
- › Appuyer sur la touche **q** permet de quitter le jeu.
- › Démonstration dans **demo/base.webm**.



## EXTENSION 1 – LES ÉTOILES /2

**Nom du fichier à rendre :** r-type-ext1.py

Pour valider cette étape, le programme devra fonctionner de la manière suivante :

- › Le reste du programme doit fonctionner comme précédemment.
- › Afficher un fond d'écran (image **assets/level\_1.bmp**) qui se déplacera de **4px** par tour de boucle.
- › Pour cela, vous devrez composer une image de la taille de votre fenêtre. Le code suivant montre comment procéder :

```
# Créer une surface de la taille de notre fenêtre.  
bg = pygame.Surface((GAME_WIDTH, GAME_HEIGHT))  
  
# Charger l'image de fond d'écran.  
img = load_bmp("assets/level_1.bmp")  
  
# Imprimer l'image à la position (0, 0).  
bg.blit(img, (0, 0))
```

- › Démonstration dans **demo/ext1.webm**.



## EXTENSION 2 – LES ENNEMIS /2

**Nom du fichier à rendre :** r-type-ext2.py

Pour valider cette étape, le programme devra fonctionner de la manière suivante :

- › Le reste du programme doit fonctionner comme précédemment.
- › Générer des ennemis (image **assets/monster\_1.bmp**).
- › Le nombre d'ennemis présents dans le jeu sera défini par la constante **MONSTER\_COUNT** qui vaudra **3**.
- › Les ennemis entrent dans le jeu par la droite de la fenêtre.
- › Un ennemi est détruit lorsqu'il entre en contact avec un tir.
- › Lorsqu'un ennemi est détruit, il faut afficher l'animation **assets/blast.bmp**, supprimer le tir, supprimer l'ennemi et en générer un nouveau.
- › Démonstration dans **demo/ext2.webm**.



## EXTENSION 3 – ENCORE DES ENNEMIS /2

**Nom du fichier à rendre :** r-type-ext3.py

Pour valider cette étape, le programme devra fonctionner de la manière suivante :

- › Le reste du programme doit fonctionner comme précédemment.
- › Lorsque le joueur entre en contact avec un ennemi alors les deux doivent-être détruits, l'animation **assets/blast.bmp** s'affiche et le programme se termine.
- › Lorsqu'un ennemi arrive à passer du côté gauche de la fenêtre alors le programme se termine. La partie est perdue.
- › Démonstration dans **demo/ext3.webm**.



## EXTENSION 4 – SCORE /2

**Nom du fichier à rendre :** r-type-ext4.py

Pour valider cette étape, le programme devra fonctionner de la manière suivante :

- › Le reste du programme doit fonctionner comme précédemment.
- › Afficher le score du joueur en haut à gauche à **16 pixels** des bords.
- › À chaque fois que le joueur détruit un ennemi, le score augmente de 1.
- › Pour afficher le score, vous devrez utiliser **assets/Beware.ttf**. Le code suivant permet d'afficher du texte :

```
# Charger la police « Beware » en taille 32.  
font = pygame.font.Font('assets/Beware.ttf', 32)  
  
# Génère le texte à afficher puis l'affiche à la position (0, 0).  
# (100, 100, 100) est la représentation de la couleur grise.  
text = font.render("mon texte", True, (100, 100, 100))  
window.blit(text, (0, 0))
```

- › Démonstration dans **demo/ext4.webm**.

