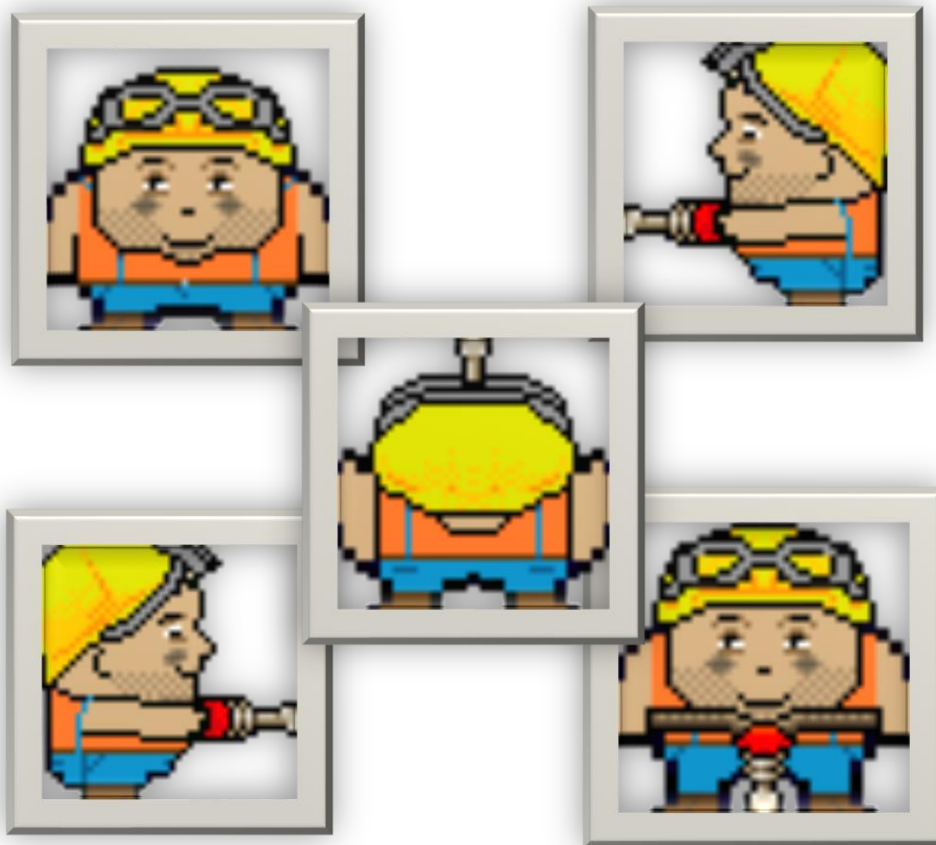


---

*Documentation technique*

# Mr. Driller



---

# Sommaire



- I. Présentation du jeu*
- II. Fonctionnement de l'application*
- III. Composition du jeu*
- IV. Explication du fonctionnement des fonctions*



---

# *Présentation du jeu*

Objectif : L'objectif du MR.Driller est d'arriver à parcourir les 10 niveaux du jeu avec seulement 3 vies.

But du jeu : Le but du jeu étant de détruire des blocs et de descendre dans les profondeurs de bétons en gardant suffisamment d'air et en évitant les plots piégés pour ne pas perdre de vie, et survivre aux différents niveaux de plus en plus compliqués.



---

# Fonctionnement de l'application

Le jeu étant codé en python, compatible sur chaque OS et lancé en interface graphique grâce au module pyGame, il demande d'avoir installer python sur sa machine pour pouvoir le lancer ;

Pour installer python :

<https://www.python.org/downloads/>



Téléchargez le pack correspondant à votre machine puis suivez les instructions pour l'installation.

Pour ajouter le module pyGame, lancez un terminal et exécutez la commande suivante :

```
python -m pip install pygame
```

Si pip affiche :

```
Requirement already satisfied (use --  
upgrade to upgrade): pygame in  
c:\python34\lib\site-packages
```

Cela veut dire que pyGame est déjà installé

---

# Composition du jeu

Notre MR.Driller se décompose en 5 fichiers :

Les différentes librairies utilisées  
sont inclus dans les constantes

Library.py

Les constantes sont incluses dans le  
fichier function pour plus de visibilité

Constante.py

Le fichier function inclus toutes les  
Fonctionnalités et les pages de jeu  
Dans le main

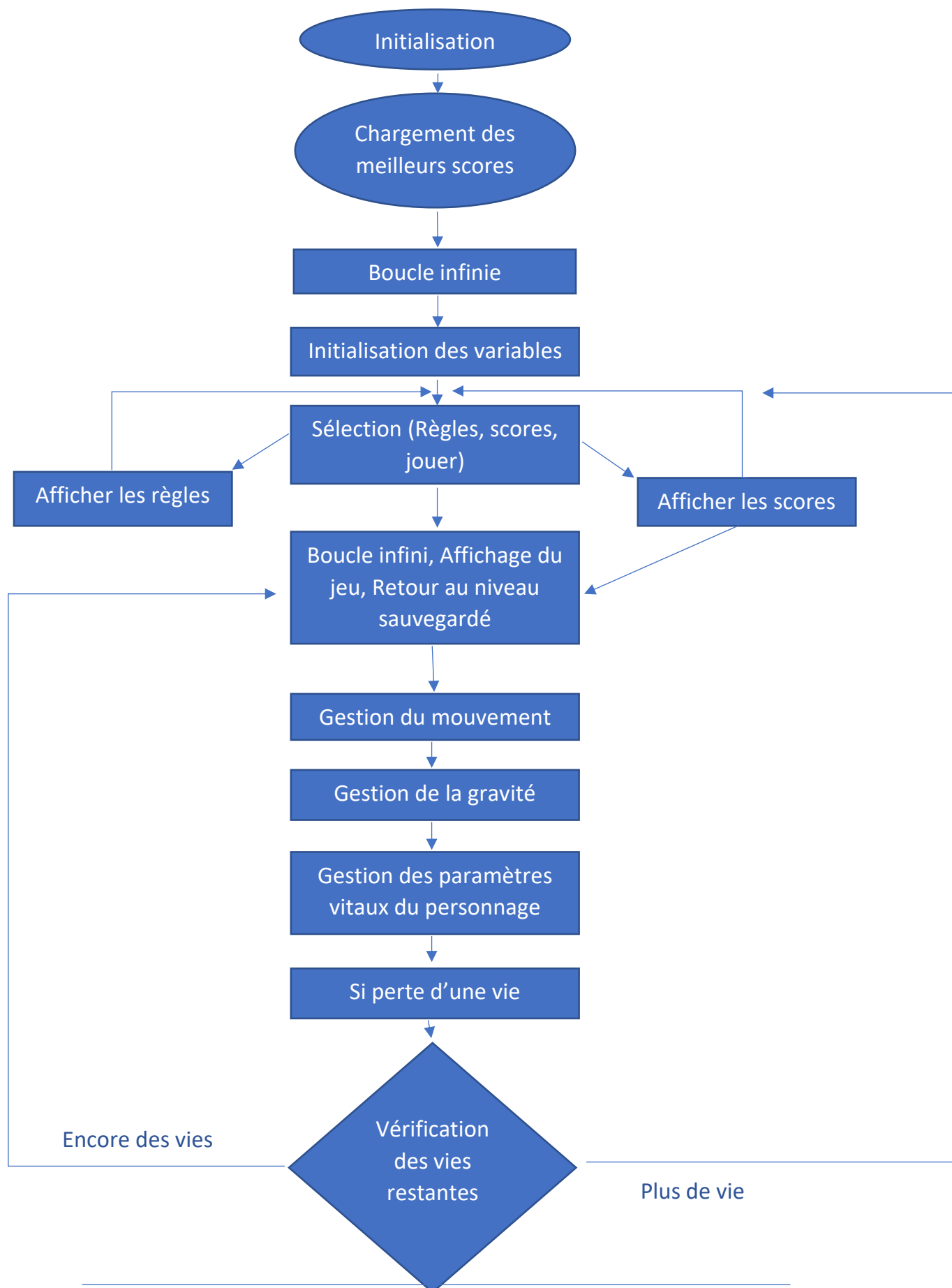
Function.py

Le fichier main regroupe tous les  
fichiers

Main.py

Le fichier score est inclus dans le  
Main afin d'afficher les meilleurs  
Scores obtenus et récupérer les  
Scores de jeu

Score.json



---

# Fonctionnement des fonctions

Fonctions principales :

```
def display(board, posX, posY, position, current_height, vie, air, score, best_scores, level):
```

La fonction display permet d'afficher la page de jeu, qui regroupe plusieurs fonctions, les fonctions : blocs

```
display_bloc(board, tmp, ordo, level)
```

, les informations

```
display_info(posY, current_height, vie, air, best_scores, score)
```

, les personnages

```
display_perso(position, posX, posY, current_height, level)
```

 .

```
def move(board, posX, posY, position, current_height, score, level, air):
```

La fonction move permet de gérer les déplacements du personnage, tout en vérifiant le niveau auquel est le personnage, la hauteur du déroulement de jeu pour l'altitude.

```
def gravity(board, posX, posY, current_height, level):
```

La fonction gravity permet de gérer le déroulement du jeu, et la gravité des blocs.

```
def propagation(board, posX, posY, level=0):
```

La fonction propagation permet de détecter autour d'un bloc tous les blocs qui vont se coller les uns aux autres.

```
def chooseBloc(board, x, y, typeBloc, level):
```

La fonction chooseBloc permet de choisir le bon type de bloc à mettre (1 coin seulement, deux coins, à droite, à gauche selon les 16 type de bloc par couleur).

```
def display_fusions_bloc(board, x, y, id_bloc, level):
```

La fonction display\_fusions\_bloc permet de détecter autour d'un bloc s'il y a un autre bloc de la même couleur, il le fusionne.

---

*« Jouez pour le plaisir, et  
perdez noblement. »*

*Bon amusement*

*. L'équipe de développement .*

