

Mission

Batman travaille dans la Batcave sur le programme de la Batmobile. Il cherche à programmer les instructions pour vaincre ses ennemis de façon automatique.

Règles

Le système scanne les ennemis alentours et renvoie les informations sur les ennemis :

- x:3 signifie que l'ennemi se trouve en position 3 sur l'axe x
- pv:35 signifie que l'ennemi a 35 points de vie

Pour l'instant, dans cette version bêta de la Batmobile, celle-ci ne se déplace que dans une seule direction et démarre toujours à la position x:0.

La batmobile peut faire feu et retire 10 PV à chaque fois. Pour vaincre un ennemi qui a 22 PV, il faudra faire feu 3 fois.

La batmobile ne se déplace que dans un sens, il faudra donc vaincre les ennemis selon l'ordre croissant de leur position x.

Tu dois retourner les instructions que doit réaliser la batmobile pour vaincre tous les ennemis :

- un « **D** » signifie un **D**éplacement de 1 sur l'axe x
- un « **F** » signifie faire **F**eu 1 fois

Par exemple, s'il y a un ennemi « x:3 pv:35 », tu dois retourner : « DDDFFFF »

\$ennemis = ['x:14 pv:18', 'x:26 pv:33', 'x:9 pv:13', 'x:16 pv:33', 'x:29 pv:20'];

DDDDDDDDDDFFDDDDDDFFDDFFFFDDDDDDDDDDDDFFFFDDDDFF

Je me Déplace de 9 pour viser l'ennemi à la position 9
Je fais Feu 2 fois pour vaincre l'ennemi qui a 13 PV
Je me Déplace de 5 pour viser l'ennemi à la position 14
Je fais Feu 2 fois pour vaincre l'ennemi qui a 18 PV
Je me Déplace de 2 pour viser l'ennemi à la position 16
Je fais Feu 4 fois pour vaincre l'ennemi qui a 33 PV
Je me Déplace de 10 pour viser l'ennemi à la position 26
Je fais Feu 4 fois pour vaincre l'ennemi qui a 33 PV
Je me Déplace de 3 pour viser l'ennemi à la position 29
Je fais Feu 2 fois pour vaincre l'ennemi qui a 20 PV

On commence par parcourir les ennemis pour extraire les données importantes.

On utilise la fonction **sscanf**, qui permet d'extraire les données selon leur format. Là on définit, qu'il y a « x: » puis un nombre (%d), puis » pv: », puis un autre nombre (%d). Et on utilise une double attribution pour renseigner à la fois \$position et \$pv.

On crée donc un nouveau tableau, indexé par la position.

La fonction **ksort**, qui fonctionne par référence, me permet de trier selon les clés (**keys**) du tableau.

La batmobile démarre en position 0.

Je parcours à nouveau chaque ennemi.

Le nombre de déplacement vers la **D**roite correspond à la différence entre la position de l'ennemi et la position courante de la batmobile. Puis elle avance.

Le nombre de coup de **F** correspond à l'arrondi supérieur des points de vie de l'ennemi divisés par 10.

J'utilise la fonction `str_repeat` pour répéter mes caractères D ou F.