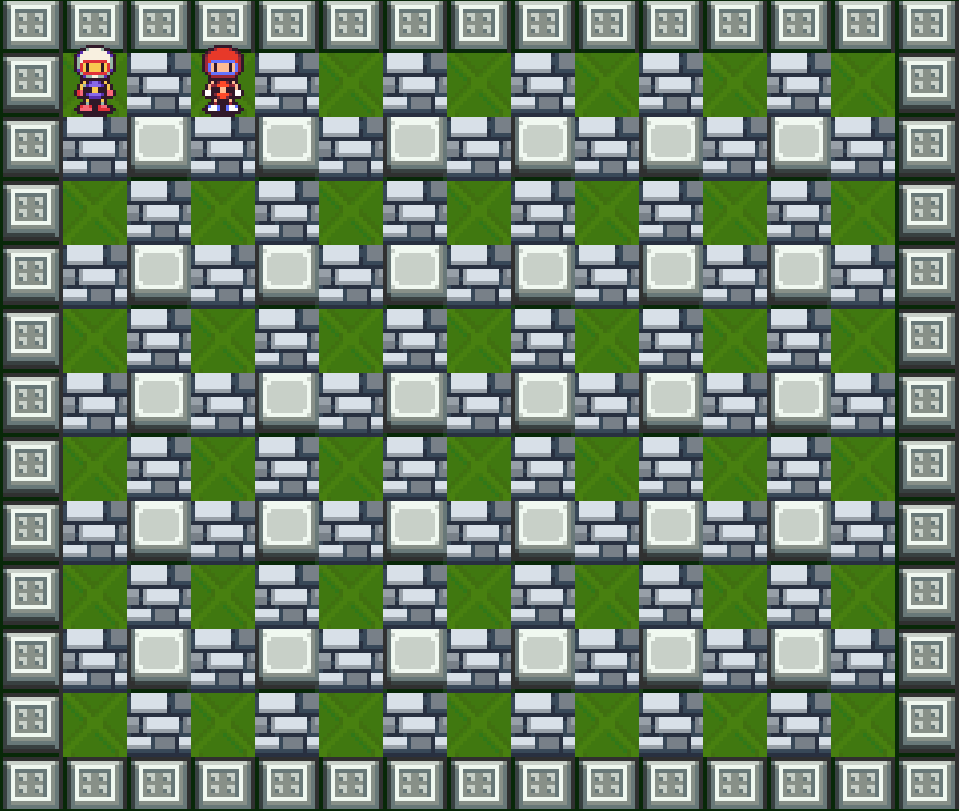
Delrieu Nils TS4

Bomberman-Like

Introduction :

L'idée nous est venu de créer un jeu avec l'aspect et le fonctionnement d'un Bomber man, mais à un autre niveau et avec un gameplay quelque peu différent.

 *Carte d'un vrai bomberman* *Carte de notre « bomberman »*

Nous avons gardé la carte et le système de bloc destructible, en adaptant cela à notre manière pour réaliser un jeu différent.

En effet, c'est en détruisant les blocs, qui vont réapparaître 7 secondes plus tard, qu'on a une chance de tuer les ennemis, car ces derniers meurent s’ils se trouvent à un endroit ou un bloc réapparaît.

Pour cela il peut pousser l'autre joueur.

Les parties ont été clairement définies :

* Toutes les collisions et également l'environnement (Ma partie)
* Gère les déplacements
* Gère l'interaction avec l’attaque

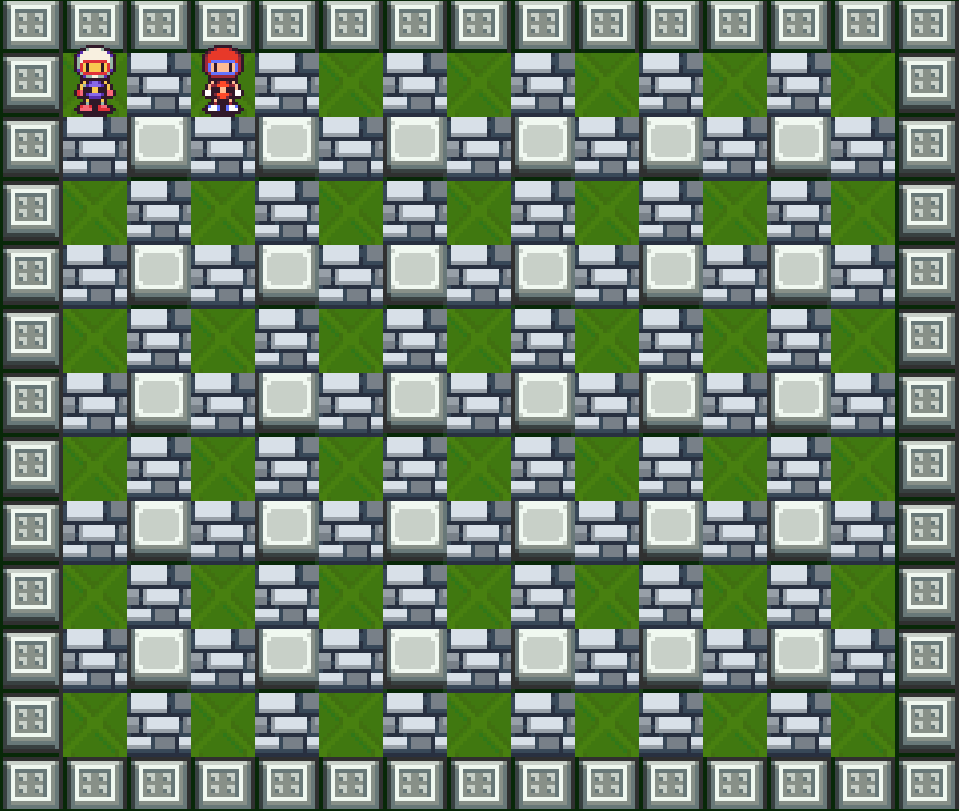
Structure générale du code :

Le code est composé d'une grande partie initialisation, où les différentes classes sont présentées :

Cela permet de « gérer » les différents objets qui vont être réutilise par la suite :

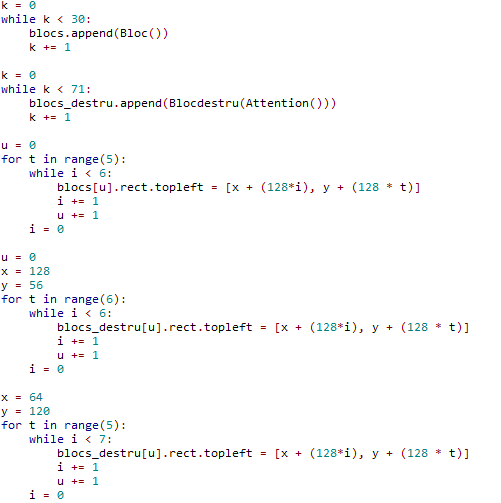


*Exemple de nos classes bloc et blocdestru, permettant de regrouper les définitions de*  *fonctions dès le départ et permettre de réutiliser les fonctions au moment voulu.*

Nous avons des classes pour tous les objets que nous souhaitons utiliser, des blocs jusqu'aux personnages.

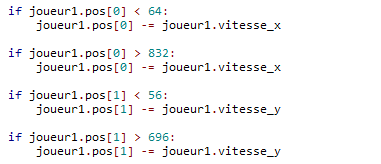
*Les objets présents sur l'écran sont gérés depuis les classes*

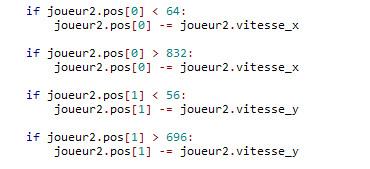
Ensuite, nous avons la partie où nous initialisons la position des blocs, sans les faire apparaître à l'écran :



*Créations des blocs destructibles et non destructibles de la carte*

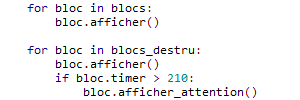
Puis viens la partie de « logique » du jeu, ou a lieu toutes les interactions :





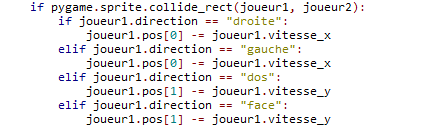
*Exemples avec les Limites de déplacement*

*du joueur 1 et du joueur 2 sur l'écran de jeu*

Enfin viens la partie où nous faisons apparaître les objets sur notre écran :

*Apparition des différents blocs avec la fonction afficher définit dans les classes*

Présentation partie personnelle :

Le but de ma partie était de gérer les différentes collisions, que ce soit les personnages avec les objets de l'environnement ou encore les personnages entre eux.

Dans cette partie par exemple, à lieu les collisions entre les deux personnages :

S’il y a une collision entre les images des joueurs 1 et 2, le joueur 1 restera à sa position initiale au lieu de se déplacer vers la case voulue.

Nous réalisons cela en annulant le déplacement lors de la collision en inversant la vitesse.

Le problème toujours présent est quand les deux joueurs avancent en même temps, car les deux joueurs se retrouvent ainsi bloquée a des coordonnées qui ne correspondent pas au mouvement habituel, mais cela n’est pas un problème important car lors d’une attaque , le joueur qui se fait pousser sur un bloc meurt.

ANNEXE ( CODE ) :

**import** pygame

# variable qui définie si le jeu doit continure de tourner ou non

done **=** **False**

# couleurs en rgb

BLACK **=** **(**0**,** 0**,** 0**)**

WHITE **=** **(**255**,** 255**,** 255**)**

GREEN **=** **(**0**,** 255**,** 0**)**

RED **=** **(**255**,** 0**,** 0**)**

BG\_SPRITES **=** **(**33**,** 57**,** 148**)**

# variables de position et compteurs

x **=** 128

y **=** 120

i **=** 0

j **=** 0

m **=** 0

# création de Clock de Display et d'un fond

clock **=** pygame**.**time**.**Clock**()**

screen **=** pygame**.**display**.**set\_mode**((**960**,** 816**))**

background **=** pygame**.**image**.**load**(**"fond.png"**).**convert**()**

**class** **Bloc(**pygame**.**sprite**.**Sprite**):**

# classe des blocs indestuctibles

**def** \_\_init\_\_**(**self**):**

# fonction d'initialisation

# utilisation de pygame.sprite

super**().**\_\_init\_\_**()**

# création image

self**.**image **=** pygame**.**image**.**load**(**"bloc.png"**).**convert**()**

# création Rect

self**.**rect **=** self**.**image**.**get\_rect**(**topleft**=[**0**,** 0**])**

**def** afficher**(**self**):**

#fonction d'affichage

# on affiche l'image en fonction du Rect

screen**.**blit**(**self**.**image**,** self**.**rect**)**

**class** **Blocdestru(**pygame**.**sprite**.**Sprite**):**

# classe des blocs destructibles

**def** \_\_init\_\_**(**self**,** attention**):**

# fonction d'initialisation

# utilisation de pygame.sprite

super**().**\_\_init\_\_**()**

# création des images

self**.**image\_solide **=** pygame**.**image**.**load**(**"bloc\_destructible.png"**).**convert**()**

self**.**image\_casse **=** pygame**.**image**.**load**(**"bloc\_casse.png"**).**convert**()**

# création rect

self**.**rect **=** self**.**image\_solide**.**get\_rect**(**topleft**=[**0**,** 0**])**

# création d'un timer pour la réaparition du bloc

self**.**timer **=** 0

# création d'un variable représentant l'état du bloc

self**.**etat **=** "solide"

# récupération de l'objet Attention

self**.**attention **=** attention

**def** afficher**(**self**):**

# fonction d'affichage du bloc

# si le bloc est solide

**if** self**.**etat **==** "solide"**:**

# on affiche l'image de bloc solide en fonction du Rect

screen**.**blit**(**self**.**image\_solide**,** self**.**rect**)**

# si il est cassé

**elif** self**.**etat **==** "casse"**:**

# on affiche l'image de bloc cassé en fonction du Rect

screen**.**blit**(**self**.**image\_casse**,** self**.**rect**)**

**def** afficher\_attention**(**self**):**

# fonction d'affichage de l'objet attention

# le Rect de attention est mit au même endroit que celui du bloc

self**.**attention**.**rect**.**center **=** self**.**rect**.**center

# et on l'affiche

self**.**attention**.**afficher**()**

**class** **Attention(**pygame**.**sprite**.**Sprite**):**

#classe du panneau attention

**def** \_\_init\_\_**(**self**):**

# fonction d'initialisation

# utilisation de pygame.sprite

super**().**\_\_init\_\_**()**

# création d'une image

self**.**image **=** pygame**.**image**.**load**(**"attention.png"**).**convert**()**

# dont on transforme une certaine couleur en transparent

self**.**image**.**set\_colorkey**(**BG\_SPRITES**)**

# création d'un Rect

self**.**rect **=** self**.**image**.**get\_rect**(**topleft**=[**0**,** 0**])**

**def** afficher**(**self**):**

# fonction d'affichage du panneau

# on affiche l'image en fonction du Rect

screen**.**blit**(**self**.**image**,** self**.**rect**)**

**class** **SpriteSheet(**object**):**

**def** \_\_init\_\_**(**self**,** file\_name**,** bg**):**

# fonction d'initialisation

# on récupère la couleur du fond dont on veux se débarrasser

self**.**bg **=** bg

# création d'une image

self**.**sprite\_sheet **=** pygame**.**image**.**load**(**file\_name**).**convert**()**

# on enlève la couleur non voulue

self**.**sprite\_sheet**.**set\_colorkey**(**self**.**bg**)**

# création d'un compteur

self**.**i **=** 0

# création d'une liste vide qui contiendra les images d'une animation

self**.**list\_images **=** **[]**

**def** get\_all\_images**(**self**,** largeur**,** hauteur**,** marge**,** nbr**):**

# fonction qui récupère toutes les images d'une animation en fonction de plusieurs paramètres

# on le fait autant de fois qu'il y a d'images

**while** self**.**i **<** nbr**:**

# on récupère une image qu'on ajoute à la liste

self**.**list\_images**.**append**(**self**.**sprite\_sheet**.**subsurface**(**pygame**.**Rect**(**largeur **\*** self**.**i **+** marge **\*** **(**self**.**i**),** 0**,** largeur**,** hauteur**)))**

# on incrémente le compteur

self**.**i **+=** 1

# quand on a finis on le remet à 1

self**.**i **=** 1

# et on renvoie la liste d'images

**return** self**.**list\_images

**class** **Bomberman(**pygame**.**sprite**.**Sprite**):**

**def** \_\_init\_\_**(**self**,** touches**=[**pygame**.**K\_LEFT**,** pygame**.**K\_RIGHT**,** pygame**.**K\_UP**,** pygame**.**K\_DOWN**,** pygame**.**K\_KP1**],** sprites**=**"1"**,** slash**=None):**

# fonction d'initialisation

# utilisation de pygame.sprite

super**().**\_\_init\_\_**()**

# création de compteurs

self**.**i **=**0

self**.**j **=** 0

# variables de la dernière direction du personnage

self**.**last\_direction **=** "face"

# on récupères les touches enfoncées dans une variable

self**.**touches **=** touches

# on récupère le chemin du dossier des sprites

self**.**sprites **=** sprites **+** "/"

# création des images

self**.**droite **=** pygame**.**image**.**load**(**self**.**sprites**+**"droite.png"**).**convert**()**

self**.**dos **=** pygame**.**image**.**load**(**self**.**sprites**+**"dos.png"**).**convert**()**

self**.**face **=** pygame**.**image**.**load**(**self**.**sprites**+**"face.png"**).**convert**()**

self**.**gauche **=** pygame**.**image**.**load**(**self**.**sprites**+**"gauche.png"**).**convert**()**

# création d'un Rect

self**.**rect **=** pygame**.**Rect**(**0**,** 0**,** 64**,** 64**)**

# on enlève la couleur non voulue des images

self**.**droite**.**set\_colorkey**(**BG\_SPRITES**)**

self**.**dos**.**set\_colorkey**(**BG\_SPRITES**)**

self**.**face**.**set\_colorkey**(**BG\_SPRITES**)**

self**.**gauche**.**set\_colorkey**(**BG\_SPRITES**)**

# variable de position du personnage

self**.**pos **=** **[**64**,** 56**]**

# variables de vitesse du personnage

self**.**vitesse\_x **=** 0

self**.**vitesse\_y **=** 0

# variable de direction du personnage

self**.**direction **=** "face"

# booléens pour savoir si le personnage est en train de bouger, d'attaquer, de se faire pousser ou est mort

self**.**moving **=** **False**

self**.**attacking **=** **False**

self**.**dead **=** **False**

self**.**knockback **=** **False**

# récupération de l'objet Slash associé au joueur

self**.**slash **=** slash

# booléen pour ne faire reculer l'autre personnage qu'un fois

self**.**knockback\_f\_montant **=** **False**

# création de variables et de listes pour l'animation du personnages

self**.**frame **=** 0

self**.**v\_frame **=** 0

self**.**animation\_b **=** **[]**

self**.**animation\_h **=** **[]**

self**.**animation\_d **=** **[]**

self**.**animation\_g **=** **[]**

# on récupère les animations

sprite\_sheet **=** SpriteSheet**(**self**.**sprites**+**"anim\_face.png"**,** BG\_SPRITES**)**

self**.**animation\_b **=** sprite\_sheet**.**get\_all\_images**(**42**,** 72**,** 15**,** 8**)**

sprite\_sheet **=** SpriteSheet**(**self**.**sprites**+**"anim\_dos.png"**,** BG\_SPRITES**)**

self**.**animation\_h **=** sprite\_sheet**.**get\_all\_images**(**42**,** 72**,** 15**,** 8**)**

sprite\_sheet **=** SpriteSheet**(**self**.**sprites**+**"anim\_droite.png"**,** BG\_SPRITES**)**

self**.**animation\_d **=** sprite\_sheet**.**get\_all\_images**(**48**,** 72**,** 15**,** 8**)**

sprite\_sheet **=** SpriteSheet**(**self**.**sprites**+**"anim\_gauche.png"**,** BG\_SPRITES**)**

self**.**animation\_g **=** sprite\_sheet**.**get\_all\_images**(**48**,** 72**,** 15**,** 8**)**

**def** afficher**(**self**):**

# fonction d'affichage du personnage

# si le joueur est de face et qu'il bouge verticallement

**if** self**.**direction **==** "face" **and** self**.**vitesse\_y **!=** 0**:**

# si on a va dépasser la dernière image de l'animation

**if** self**.**frame **>=** len**(**self**.**animation\_b**):**

# on retourne à la première

self**.**frame **=** 0

# on affiche l'image au centre du rect

screen**.**blit**(**self**.**animation\_b**[**self**.**frame**],** **[**self**.**pos**[**0**]+**11**,** self**.**pos**[**1**]-**8**])**

# on incrémnte le compteur

self**.**v\_frame **+=** 1

# si on a afficher 3 fois de suite la même image

**if** self**.**v\_frame **>** 3**:**

#on passe à la suivante

self**.**frame **+=** 1

#et on remet le compteur à 0

self**.**v\_frame **=** 0

# si le joueur est de face et ne bouge pas verticallement

**elif** self**.**direction **==** "face" **and** self**.**vitesse\_y **==** 0**:**

# on affiche l'image fixe au centre du rect

screen**.**blit**(**self**.**face**,** **[**self**.**pos**[**0**]+**11**,** self**.**pos**[**1**]-**8**])**

# même chose pour les autres directions

**elif** self**.**direction **==** "dos" **and** self**.**vitesse\_y **!=** 0**:**

**if** self**.**frame **>=** len**(**self**.**animation\_h**):**

self**.**frame **=** 0

screen**.**blit**(**self**.**animation\_h**[**self**.**frame**],** **[**self**.**pos**[**0**]+**11**,** self**.**pos**[**1**]-**8**])**

self**.**v\_frame **+=** 1

**if** self**.**v\_frame **>** 3**:**

self**.**frame **+=** 1

self**.**v\_frame **=** 0

**elif** self**.**direction **==** "dos"**and** self**.**vitesse\_y **==** 0**:**

screen**.**blit**(**self**.**dos**,** **[**self**.**pos**[**0**]+**11**,** self**.**pos**[**1**]-**8**])**

**elif** self**.**direction **==** "gauche" **and** self**.**vitesse\_x **!=** 0**:**

**if** self**.**frame **>=** len**(**self**.**animation\_g**):**

self**.**frame **=** 0

screen**.**blit**(**self**.**animation\_g**[**self**.**frame**],** **[**self**.**pos**[**0**]+**11**,** self**.**pos**[**1**]-**8**])**

self**.**v\_frame **+=** 1

**if** self**.**v\_frame **>** 3**:**

self**.**frame **+=** 1

self**.**v\_frame **=** 0

**elif** self**.**direction **==** "gauche" **and** self**.**vitesse\_x **==** 0**:**

screen**.**blit**(**self**.**gauche**,** **[**self**.**pos**[**0**]+**11**,** self**.**pos**[**1**]-**8**])**

**elif** self**.**direction **==** "droite" **and** self**.**vitesse\_x **!=**0**:**

**if** self**.**frame **>=** len**(**self**.**animation\_d**):**

self**.**frame **=** 0

screen**.**blit**(**self**.**animation\_d**[**self**.**frame**],** **[**self**.**pos**[**0**]+**11**,** self**.**pos**[**1**]-**8**])**

self**.**v\_frame **+=** 1

**if** self**.**v\_frame **>** 3**:**

self**.**frame **+=** 1

self**.**v\_frame **=** 0

**elif** self**.**direction **==** "droite" **and** self**.**vitesse\_x **==** 0**:**

screen**.**blit**(**self**.**droite**,** **[**self**.**pos**[**0**]+**11**,** self**.**pos**[**1**]-**8**])**

**def** move**(**self**):**

# fonction pour faire bouger le personnage

# si le personnage est dirigé vers la droite et que le compteur n'est pas terminé

**if** self**.**direction **==** "droite" **and** self**.**j **<** 17**:**

# on change la vitesse du personnage pendant uen frame

self**.**vitesse\_x **=** 4

# et on incrémente le comteur

self**.**j **+=** 1

# si on a l'a fait sur 17 frames

**if** self**.**j **>=** 17**:**

# on remet le compteur à 0

self**.**j **=** 0

# et on arrête le personnage

self**.**moving **=** **False**

self**.**vitesse\_x **=** 0

# même chose pour les autres directions

**if** self**.**direction **==** "gauche" **and** self**.**j **<** 17**:**

self**.**vitesse\_x **=** **-**4

self**.**j **+=** 1

**if** self**.**j **>=** 17**:**

self**.**j **=** 0

self**.**moving **=** **False**

self**.**vitesse\_x **=** 0

**if** self**.**direction **==** "face" **and** self**.**j **<** 17**:**

self**.**vitesse\_y **=** 4

self**.**j **+=** 1

**if** self**.**j **>=** 17**:**

self**.**j **=** 0

self**.**moving **=** **False**

self**.**vitesse\_y **=** 0

**if** self**.**direction **==** "dos" **and** self**.**j **<** 17**:**

self**.**vitesse\_y **=** **-**4

self**.**j **+=** 1

**if** self**.**j **>=** 17**:**

self**.**j **=** 0

self**.**moving **=** **False**

self**.**vitesse\_y **=** 0

# enfin on déplace le joueur selon sa vitesse

self**.**pos**[**0**]** **+=** self**.**vitesse\_x

self**.**pos**[**1**]** **+=** self**.**vitesse\_y

**def** attack**(**self**):**

# fonction pour faire bouger le rect du slash

# si le personnage regarde vers la droite

**if** self**.**direction **==** "droite"**:**

# on change le Rect du slash pour qu'il soit une case à droite du personnage

self**.**slash**.**rect**.**topleft **=** **(**self**.**pos**[**0**]+**64**,** self**.**pos**[**1**])**

# même chose pour les autres directions

**if** self**.**direction **==** "gauche"**:**

self**.**slash**.**rect**.**topleft **=** **(**self**.**pos**[**0**]-**64**,** self**.**pos**[**1**])**

**if** self**.**direction **==** "dos"**:**

self**.**slash**.**rect**.**topleft **=** **(**self**.**pos**[**0**],** self**.**pos**[**1**]-**64**)**

**if** self**.**direction **==** "face"**:**

self**.**slash**.**rect**.**topleft **=** **(**self**.**pos**[**0**],** self**.**pos**[**1**]+**64**)**

**def** update**(**self**,** pressed\_keys**=None):**

# fonction qui met a jour les touches enfoncées et agit en conséquence

# si une touche appuyée est celle 0 (gauche)

**if** pressed\_keys**[**self**.**touches**[**0**]]:**

# si le personnage regarde vers la gauche et qu'il la regardait aussi avant

**if** self**.**direction **==** "gauche" **and** self**.**direction **==** self**.**last\_direction**:**

# on l'autorise a bouger

self**.**moving **=** **True**

# sinon on le fait regarder la gauche

**else:**

self**.**direction **=** "gauche"

# même chose pour les autres touches de mouvements

**if** pressed\_keys**[**self**.**touches**[**1**]]:**

**if** self**.**direction **==** "droite" **and** self**.**direction **==** self**.**last\_direction**:**

self**.**moving **=** **True**

**else:**

self**.**direction **=** "droite"

**if** pressed\_keys**[**self**.**touches**[**2**]]:**

**if** self**.**direction **==** "dos" **and** self**.**direction **==** self**.**last\_direction**:**

self**.**moving **=** **True**

**else:**

self**.**direction **=** "dos"

**if** pressed\_keys**[**self**.**touches**[**3**]]:**

**if** self**.**direction **==** "face" **and** self**.**direction **==** self**.**last\_direction**:**

self**.**moving **=** **True**

**else:**

self**.**direction **=** "face"

# si la touche 4 (attaque) est enfoncée

**if** pressed\_keys**[**self**.**touches**[**4**]]:**

# on autorise le pseronnage à attaquer

self**.**attacking **=** **True**

# si le personnage est orienté vers la gauche mais qu'on a appuie pas sur la touche 0 (gauche)

**if** self**.**direction **==** "gauche" **and** **not** pressed\_keys**[**self**.**touches**[**0**]]:**

# on change la direction du personnage vers la gauche

self**.**last\_direction **=** "gauche"

# même chose pour les autres directions

**if** self**.**direction **==** "droite" **and** **not** pressed\_keys**[**self**.**touches**[**1**]]:**

self**.**last\_direction **=** "droite"

**if** self**.**direction **==** "dos" **and** **not** pressed\_keys**[**self**.**touches**[**2**]]:**

self**.**last\_direction **=** "dos"

**if** self**.**direction **==** "face" **and** **not** pressed\_keys**[**self**.**touches**[**3**]]:**

self**.**last\_direction **=** "face"

**def** afficher\_slash**(**self**):**

# fonction pour afficher les slash du personnage

# si le compteur n'est pas terminé

**if** self**.**i **<** 9**:**

# on affiche le slash pendant une frame

self**.**slash**.**afficher**()**

# et on incrémente le compteur

self**.**i **+=** 1

# si il l'est

**else:**

# on remet le compteur à 0

self**.**i **=** 0

# on met le rect du slash hors de l'écran

self**.**slash**.**rect**.**topleft **=** **(-**128**,** **-**128**)**

# et on arrête d'attaquer

self**.**attacking **=** **False**

**class** **Slash(**pygame**.**sprite**.**Sprite**):**

# classe du slash (attaque des personnages)

**def** \_\_init\_\_**(**self**):**

# fonction d'initialisation

# utilisation de pygame.sprite

super**().**\_\_init\_\_**()**

# on crée une liste vide pour les images de l'animation

self**.**animation **=** **[]**

# on ajoute dedans chaque images de celle ci

sprite\_sheet **=** SpriteSheet**(**"slash.png"**,** **[**64**,** 128**,** 0**])**

self**.**animation **=** sprite\_sheet**.**get\_all\_images**(**64**,** 64**,** 0**,** 3**)**

# on crée un Rect

self**.**rect **=** pygame**.**Rect**(**128**,** 128**,** 64**,** 64**)**

# on crée compteurs pour afficher l'animation

self**.**frame **=** 0

self**.**v\_frame **=** 0

# variables de cooldown du slash

self**.**in\_cd **=** **False**

self**.**cd **=** 0

**def** afficher**(**self**):**

# fonction pour afficher l'animation du slash

# si on va dépasser la dernière image

**if** self**.**frame **>=** len**(**self**.**animation**):**

# on revient à la première

self**.**frame **=** 0

# on affiche l'image actuelle pendant une frame

screen**.**blit**(**self**.**animation**[**self**.**frame**],** self**.**rect**)**

# et on incrémente un compteur

self**.**v\_frame **+=** 1

# on passe a l'image suivante seulement quand on a affiché l'image 2 fois d'affilé

**if** self**.**v\_frame **>** 2**:**

self**.**frame **+=** 1

self**.**v\_frame **=** 0

# on crée un slash et on change sa position

slash1 **=** Slash**()**

slash1**.**rect**.**topleft **=** **(-**128**,** **-**128**)**

# on crée un premier joueur auquel on associe le slash

joueur1 **=** Bomberman**(**slash**=**slash1**)**

# même chose pour le joueur 2

slash2 **=** Slash**()**

slash2**.**rect**.**topleft **=** **(**1088**,** **-**128**)**

joueur2 **=** Bomberman**([**pygame**.**K\_a**,** pygame**.**K\_d**,** pygame**.**K\_w**,** pygame**.**K\_s**,** pygame**.**K\_t**],** "2"**,** slash2**)**

# on change aussi sa position

joueur2**.**pos **=** **[**832**,**56**]**

# on crée des liste vides pour les blocs

blocs **=** **[]**

blocs\_destru **=** **[]**

# on crée 30 bloc indestructibles

k **=** 0

**while** k **<** 30**:**

blocs**.**append**(**Bloc**())**

k **+=** 1

# on crée 71 blocs destructibles associés à des attentions

k **=** 0

**while** k **<** 71**:**

blocs\_destru**.**append**(**Blocdestru**(**Attention**()))**

k **+=** 1

# on place les blocs indestructibles

u **=** 0

**for** t **in** range**(**5**):**

**while** i **<** 6**:**

blocs**[**u**].**rect**.**topleft **=** **[**x **+** **(**128**\***i**),** y **+** **(**128 **\*** t**)]**

i **+=** 1

u **+=** 1

i **=** 0

# et on place les blocs destructibles

u **=** 0

x **=** 128

y **=** 56

**for** t **in** range**(**6**):**

**while** i **<** 6**:**

blocs\_destru**[**u**].**rect**.**topleft **=** **[**x **+** **(**128**\***i**),** y **+** **(**128 **\*** t**)]**

i **+=** 1

u **+=** 1

i **=** 0

x **=** 64

y **=** 120

**for** t **in** range**(**5**):**

**while** i **<** 7**:**

blocs\_destru**[**u**].**rect**.**topleft **=** **[**x **+** **(**128**\***i**),** y **+** **(**128 **\*** t**)]**

i **+=** 1

u **+=** 1

i **=** 0

u **=** 0

# on crée deux groupes de sprites pour les différents blocs

list\_bloc **=** pygame**.**sprite**.**Group**()**

list\_blocdestru **=** pygame**.**sprite**.**Group**()**

# et on les rajoute dedans

**for** bloc **in** blocs**:**

list\_bloc**.**add**(**bloc**)**

**for** bloc **in** blocs\_destru**:**

list\_blocdestru**.**add**(**bloc**)**

# tant que le jeu n'est pas fini

**while** **not** done**:**

# --- Main event loop

# pour chaque évènement pygame

**for** event **in** pygame**.**event**.**get**():**

# si l'on veut quitter

**if** event**.**type **==** pygame**.**QUIT**:**

# on change la valeur de la variable qui entraine l'arrêt du jeu

done **=** **True**

#on met les rect des 2 joueurs a leurs positions

joueur1**.**rect**.**topleft **=** **[**joueur1**.**pos**[**0**],** joueur1**.**pos**[**1**]]**

joueur2**.**rect**.**topleft **=** **[**joueur2**.**pos**[**0**],** joueur2**.**pos**[**1**]]**

# on remet la vitesse des deux joueurs à 0

joueur1**.**vitesse\_x **=** 0

joueur1**.**vitesse\_y **=** 0

joueur2**.**vitesse\_x **=** 0

joueur2**.**vitesse\_y **=** 0

# --- Game logic

# on récupère les touches enfoncées

touches **=** pygame**.**key**.**get\_pressed**()**

# si le joueur 1 n'est pas mort

**if** **not** joueur1**.**dead**:**

# si il ne bouge pas et qu'il n'attaque pas

**if** **not** joueur1**.**moving **and** **not** joueur1**.**attacking**:**

# on met a jour le joueur en fonction des touches

joueur1**.**update**(**touches**)**

# si le joueur peut bouger

**if** joueur1**.**moving**:**

# on le fait bouger

joueur1**.**move**()**

# si le joueur attaque qu'il ne bouge pas et que sont slash n'est pas en cooldown

**if** joueur1**.**attacking **and** **not** joueur1**.**moving **and** **not** joueur1**.**slash**.**in\_cd**:**

# on le fait attaquer

joueur1**.**attack**()**

# et on met son slash en cooldown

joueur1**.**slash**.**in\_cd **=** **True**

# si le slash du joueur est e ncooldown

**if** joueur1**.**slash**.**in\_cd**:**

# si il l'est depuis moins de 30 frames

**if** joueur1**.**slash**.**cd **<** 30**:**

# on le garde en cooldown pour une autre frame

joueur1**.**slash**.**cd **+=** 1

# sinon

**else:**

# on remet le compteur de frame à 0

joueur1**.**slash**.**cd **=** 0

# et le slash n'est plus en cooldown

joueur1**.**slash**.**in\_cd **=** **False**

# on récupère la liste de blocs destructible qui est entré en collision avec le slash du joueur pendant cette frame

collision **=** pygame**.**sprite**.**spritecollide**(**joueur1**.**slash**,** list\_blocdestru**,** **False)**

# si il est rentré en collision un

**if** len**(**collision**)** **>** 0**:**

# on change son état à cassé

collision**[**0**].**etat **=** "casse"

# si le joueur est entré en collision avec un bloc indestructible

**if** pygame**.**sprite**.**spritecollide**(**joueur1**,** list\_bloc**,** **False):**

# on le fait ressortir du bloc

**if** joueur1**.**direction **==** "droite"**:**

joueur1**.**pos**[**0**]** **-=** joueur1**.**vitesse\_x

# même chose pour les autres directions

**elif** joueur1**.**direction **==** "gauche"**:**

joueur1**.**pos**[**0**]** **-=** joueur1**.**vitesse\_x

**elif** joueur1**.**direction **==** "dos"**:**

joueur1**.**pos**[**1**]** **-=** joueur1**.**vitesse\_y

**elif** joueur1**.**direction **==** "face"**:**

joueur1**.**pos**[**1**]** **-=** joueur1**.**vitesse\_y

# on récupère la liste de blocs destructible qui est entré en collision avec lejoueur pendant cette frame

collision **=** pygame**.**sprite**.**spritecollide**(**joueur1**,** list\_blocdestru**,** **False)**

# si il est rentré dans un bloc

**if** len**(**collision**)** **>** 0**:**

# et si ce bloc est solide

**if** collision**[**0**].**etat **==** "solide"**:**

# on le fait ressortir du bloc (2 \* la vitesse car sinon le joueur clip dans le bloc et reste bloqué)

**if** joueur1**.**direction **==** "droite"**:**

joueur1**.**pos**[**0**]** **-=** 2**\***joueur1**.**vitesse\_x

# même chose pour les autres directions

**elif** joueur1**.**direction **==** "gauche"**:**

joueur1**.**pos**[**0**]** **-=** 2**\***joueur1**.**vitesse\_x

**elif** joueur1**.**direction **==** "dos"**:**

joueur1**.**pos**[**1**]** **-=** 2**\***joueur1**.**vitesse\_y

**elif** joueur1**.**direction **==** "face"**:**

joueur1**.**pos**[**1**]** **-=** 2**\***joueur1**.**vitesse\_y

# si le joueur est renté dans le mur de gauche

**if** joueur1**.**pos**[**0**]** **<** 64**:**

# on le fait ressortir

joueur1**.**pos**[**0**]** **-=** joueur1**.**vitesse\_x

**if** joueur1**.**pos**[**0**]** **>** 832**:**

joueur1**.**pos**[**0**]** **-=** joueur1**.**vitesse\_x

**if** joueur1**.**pos**[**1**]** **<** 56**:**

joueur1**.**pos**[**1**]** **-=** joueur1**.**vitesse\_y

**if** joueur1**.**pos**[**1**]** **>** 696**:**

joueur1**.**pos**[**1**]** **-=** joueur1**.**vitesse\_y

# si le joueur s'est fait pousser

**if** joueur1**.**knockback**:**

# si le joueur 2 regardais vers la droite lorsqu'il a touché le joueur 1

**if** joueur2**.**direction **==** "droite"**:**

# on le fait reculer vers la droite

joueur1**.**pos**[**0**]** **+=** 64

**if** joueur2**.**direction **==** "gauche"**:**

joueur1**.**pos**[**0**]** **-=** 64

**if** joueur2**.**direction **==** "dos"**:**

joueur1**.**pos**[**1**]** **-=** 64

**if** joueur2**.**direction **==** "face"**:**

joueur1**.**pos**[**1**]** **+=** 64

# puis on arrête de le pousser

joueur1**.**knockback **=** **False**

# sil le joueur 1 tape le joueur 2 et que c'est lors de la première frame qu'il l'a touché

**if** pygame**.**sprite**.**collide\_rect**(**joueur1**.**slash**,** joueur2**)** **and** **not** joueur1**.**knockback\_f\_montant**:**

# on pousse le joueur 2

joueur2**.**knockback **=** **True**

# et on empêche que le slash repousse le joueur 2 sur toute la longueur de son animaton

joueur1**.**knockback\_f\_montant **=** **True**

# même chose pour le joueur 2

**if** **not** joueur2**.**dead**:**

**if** **not** joueur2**.**moving **and** **not** joueur2**.**attacking**:**

joueur2**.**update**(**touches**)**

**if** joueur2**.**moving**:**

joueur2**.**move**()**

**if** joueur2**.**attacking **and** **not** joueur2**.**moving**:**

joueur2**.**attack**()**

**if** joueur2**.**slash**.**in\_cd**:**

**if** joueur2**.**slash**.**cd **<** 30**:**

joueur2**.**slash**.**cd **+=** 1

**else:**

joueur2**.**slash**.**cd **=** 0

joueur2**.**slash**.**in\_cd **=** **False**

**if** pygame**.**sprite**.**spritecollide**(**joueur2**,** list\_bloc**,** **False):**

**if** joueur2**.**direction **==** "droite"**:**

joueur2**.**pos**[**0**]** **-=** joueur2**.**vitesse\_x

**elif** joueur2**.**direction **==** "gauche"**:**

joueur2**.**pos**[**0**]** **-=** joueur2**.**vitesse\_x

**elif** joueur2**.**direction **==** "dos"**:**

joueur2**.**pos**[**1**]** **-=** joueur2**.**vitesse\_y

**elif** joueur2**.**direction **==** "face"**:**

joueur2**.**pos**[**1**]** **-=** joueur2**.**vitesse\_y

collision **=** pygame**.**sprite**.**spritecollide**(**joueur2**.**slash**,** list\_blocdestru**,** **False)**

**if** len**(**collision**)** **>** 0**:**

collision**[**0**].**etat **=** "casse"

collision **=** pygame**.**sprite**.**spritecollide**(**joueur2**,** list\_blocdestru**,** **False)**

**if** len**(**collision**)** **>** 0**:**

**if** collision**[**0**].**etat **==** "solide"**:**

**if** joueur2**.**direction **==** "droite"**:**

joueur2**.**pos**[**0**]** **-=** 2**\***joueur2**.**vitesse\_x

**elif** joueur2**.**direction **==** "gauche"**:**

joueur2**.**pos**[**0**]** **-=** 2**\***joueur2**.**vitesse\_x

**elif** joueur2**.**direction **==** "dos"**:**

joueur2**.**pos**[**1**]** **-=** 2**\***joueur2**.**vitesse\_y

**elif** joueur2**.**direction **==** "face"**:**

joueur2**.**pos**[**1**]** **-=** 2**\***joueur2**.**vitesse\_y

**if** joueur2**.**pos**[**0**]** **<** 64**:**

joueur2**.**pos**[**0**]** **-=** joueur2**.**vitesse\_x

**if** joueur2**.**pos**[**0**]** **>** 832**:**

joueur2**.**pos**[**0**]** **-=** joueur2**.**vitesse\_x

**if** joueur2**.**pos**[**1**]** **<** 56**:**

joueur2**.**pos**[**1**]** **-=** joueur2**.**vitesse\_y

**if** joueur2**.**pos**[**1**]** **>** 696**:**

joueur2**.**pos**[**1**]** **-=** joueur2**.**vitesse\_y

**if** joueur2**.**knockback **==** **True:**

**if** joueur1**.**direction **==** "droite"**:**

joueur2**.**pos**[**0**]** **+=** 64

**if** joueur1**.**direction **==** "gauche"**:**

joueur2**.**pos**[**0**]** **-=** 64

**if** joueur1**.**direction **==** "dos"**:**

joueur2**.**pos**[**1**]** **-=** 64

**if** joueur1**.**direction **==** "face"**:**

joueur2**.**pos**[**1**]** **+=** 64

joueur2**.**knockback **=** **False**

**if** pygame**.**sprite**.**collide\_rect**(**joueur2**.**slash**,** joueur1**)** **and** **not** joueur2**.**knockback\_f\_montant**:**

joueur1**.**knockback **=** **True**

joueur2**.**knockback\_f\_montant **=** **True**

# si les deux joueur se rentrent dedans

**if** pygame**.**sprite**.**collide\_rect**(**joueur1**,** joueur2**):**

# si le joueur 1 regarde à droite

**if** joueur1**.**direction **==** "droite"**:**

# on le fait ressortir du joueur 2

joueur1**.**pos**[**0**]** **-=** joueur1**.**vitesse\_x

# même chose pour les autres directions

**elif** joueur1**.**direction **==** "gauche"**:**

joueur1**.**pos**[**0**]** **-=** joueur1**.**vitesse\_x

**elif** joueur1**.**direction **==** "dos"**:**

joueur1**.**pos**[**1**]** **-=** joueur1**.**vitesse\_y

**elif** joueur1**.**direction **==** "face"**:**

joueur1**.**pos**[**1**]** **-=** joueur1**.**vitesse\_y

# même chose pour le joueur 2

**if** joueur2**.**direction **==** "droite"**:**

joueur2**.**pos**[**0**]** **-=** joueur2**.**vitesse\_x

**elif** joueur2**.**direction **==** "gauche"**:**

joueur2**.**pos**[**0**]** **-=** joueur2**.**vitesse\_x

**elif** joueur2**.**direction **==** "dos"**:**

joueur2**.**pos**[**1**]** **-=** joueur2**.**vitesse\_y

**elif** joueur2**.**direction **==** "face"**:**

joueur2**.**pos**[**1**]** **-=** joueur2**.**vitesse\_y

# pour chaque bloc destructible

**for** bloc **in** blocs\_destru**:**

# si le bloc est cassé

**if** bloc**.**etat **==** "casse"**:**

# on incrémente le timer

bloc**.**timer **+=** 1

# si le bloc est cassé depuis 240 frames

**if** bloc**.**timer **>** 240**:**

# on remet le time à 0

bloc**.**timer **=** 0

# et on remet le bloc à l'état solide

bloc**.**etat **=** "solide"

# si le bloc est solide et que le joueur 1 est dans le bloc

**if** bloc**.**etat **==** "solide" **and** pygame**.**sprite**.**collide\_rect**(**joueur1**,** bloc**)** **and** **not** joueur1**.**moving**:**

# le joueur 1 meurt

joueur1**.**dead **=** **True**

# mêm chose pour le joueur 2

**if** bloc**.**etat **==** "solide" **and** pygame**.**sprite**.**collide\_rect**(**joueur2**,** bloc**)** **and** **not** joueur2**.**moving**:**

joueur2**.**dead **=** **True**

# si on a touché le joueur 2

**if** joueur1**.**knockback\_f\_montant **==** **True:**

# on empêche de le faire reculer pendant toute l'animation en gardant le variable à True pendant 9 frames

**if** i **<** 9**:**

i **+=** 1

# si on l'a fait on peut repousser le joueur 2

**else:**

i **=** 0

joueur1**.**knockback\_f\_montant **=** **False**

# même chose pour le joueur 2

**if** joueur2**.**knockback\_f\_montant **==** **True:**

**if** u **<** 9**:**

u **+=** 1

**else:**

u **=** 0

joueur2**.**knockback\_f\_montant **=** **False**

# --- Drawing code

# on remplit l'écran de noir

screen**.**fill**(**BLACK**)**

# on affiche le fond

screen**.**blit**(**background**,** **(**0**,** 0**))**

# on affiche chaque bloc indestructible

**for** bloc **in** blocs**:**

bloc**.**afficher**()**

# on affiche chaque bloc destructible

**for** bloc **in** blocs\_destru**:**

bloc**.**afficher**()**

# selon leurs positionset si ils sont mort ou non, on affiche un joueur devant l'autre

**if** joueur1**.**pos**[**1**]** **<** joueur2**.**pos**[**1**]:**

**if** **not** joueur1**.**dead**:**

joueur1**.**afficher**()**

**if** **not** joueur2**.**dead**:**

joueur2**.**afficher**()**

**elif** joueur2**.**pos**[**1**]** **<** joueur1**.**pos**[**1**]:**

**if** **not** joueur2**.**dead**:**

joueur2**.**afficher**()**

**if** **not** joueur1**.**dead**:**

joueur1**.**afficher**()**

**else:**

**if** **not** joueur2**.**dead**:**

joueur2**.**afficher**()**

**if** **not** joueur1**.**dead**:**

joueur1**.**afficher**()**

# si ils sont mort ou non on affiche les slashs des joueurs

**if** **not** joueur1**.**dead**:**

joueur1**.**afficher\_slash**()**

**if** **not** joueur1**.**dead**:**

joueur2**.**afficher\_slash**()**

# on affiche chaque panneau attention (en dernier pour que ce soit par dessus tout les reste)

**for** bloc **in** blocs\_destru**:**

**if** bloc**.**timer **>** 210**:**

bloc**.**afficher\_attention**()**

# on affiche tous nos sprites

pygame**.**display**.**flip**()**

# on fait en sorte qu'on est 30 frames par seconde

clock**.**tick**(**30**)**

**print(**joueur1**.**dead**)**

# si l'on sort de la boucle, on ferme le jeu

pygame**.**quit**()**