

PyGame - Maps

Cet article décrit comment créer les Maps Pour pouvoir les afficher et les explorer dans un jeu Python.

Le programme utilisé pour cette démonstration est celui qui a été développé pour les portes ouvertes du CEPES. Le code est assez compliqué pour un niveau débutant mais si tu peux toujours y jeter un œil. Il existe un tutoriel en ligne simplifié Que tu pourras trouver dans la section "liens".

L'objectif de cet exercice est de créer des cartes afin de les explorer avec ton personnage et de découvrir différentes mécaniques à mettre en place lors de la création d'un jeu.

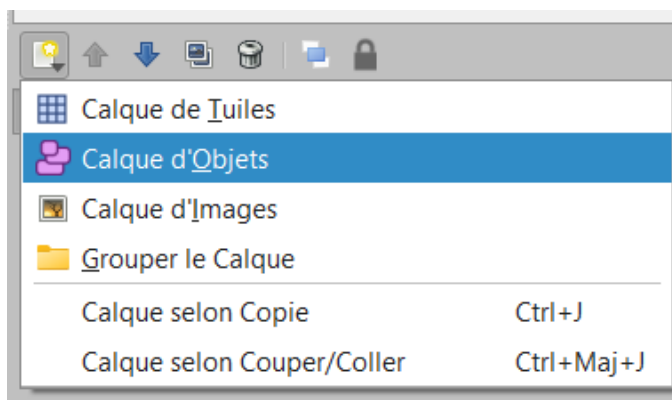
Ici, tu vas apprendre:

- une façon de gérer les collisions
- comment faire apparaître ton personnage à un endroit spécifique de ta carte
- comment voyager d'une map à l'autre

Les collisions, les points de *spawn* et les portails sont créés à l'aide d'**objets**. Les objets ne seront pas visibles au jouer dans le jeu, mais pourront être utilisés par le code afin de créer des interaction spécifiques.

Créer des objets

Les objets doivent être placés sur un calque (GB *layer*) spécifique de type "Calque d'objets" (GB *object layer*). Tu veux donc d'abord créer un nouveau calque d'objets dans tiled:



Quand un calque d'objet est sélectionné, la barre d'outils spécifique s'active:



Les points de *spawn*

Les points de spawn sont les **points** où le personnage va apparaître sur la carte. Au départ, chaque carte doit avoir **au moins un point** de spawn.

Ces points peuvent être créés à l'aide de l'outil "Insérer un point (i)":



Une fois l'outil sélectionné, clique à l'endroit sur ta carte où tu veux que ton personnage apparaisse:



Il faut ensuite **éditer ses propriétés**:

Propriétés	
Propriété	Valeur
▼ Objet	
ID	21
Modèle	
Nom	main
Classe	Spawn
Visible	<input checked="" type="checkbox"/>
X	400,00
Y	430,67
Rotation	0,00

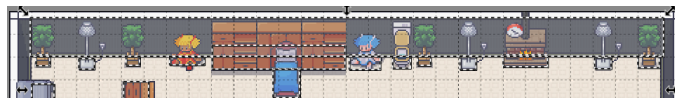
La **classe** (GB *type*) DOIT être **spawn** et son **nom** DOIT être **main**.

Ceci n'est qu'une convention (on aurait pu choisir d'autres noms). Mais **il faut la respecter** car c'est exactement ça que le code cherche pour afficher ton personnage sur la carte.

Gérer les collisions


Dans cet exercice, tu vas apprendre à gérer des collisions de façon simplifiée. On va utiliser des **objets** dans tiled qui seront gérés dans le code du jeu.

Le plus simple et le plus efficace dans pygame pour détecter les collisions est d'utiliser des **rectangles**. Dans cet exercice, tu ne vas donc créer des collisions qu'à l'aide d'objets de type rectangle.



Le code du jeu gère les collisions sur base des pieds du personnage. Il faudra éviter pour certains objets de créer une collision sur l'entièreté de celui-ci pour donner l'impression que le personnage peut passer derrière.

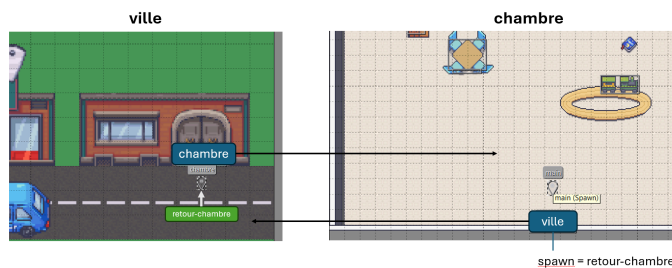
Une fois tes rectangles créés, tu peux tous les sélectionner à l'aide de la souris et changer la propriété "classe" pour lui donner la valeur "**collision**".

Propriétés	
Propriété	Valeur
▼ Objet	
ID	19
Modèle	
Nom	
Classe	collision
Visible	<input type="checkbox"/>

Explorer d'autres cartes


Dans le cadre de cet exercice, il a été choisi de ne montrer qu'une seule carte à la fois sur l'écran. Ceci étant dit, le jeu supporte plusieurs cartes.

Pour pouvoir voyager de cartes en cartes, on a mis au point ce que l'on a appelé un portail. Un portail **permet de provoquer le chargement d'une autre carte**. Ce sera par exemple le cas quand ton personnage se balade en forêt et passe par la porte d'une maison. On affichera alors la carte de l'intérieur de cette maison.



Pour rester simples et efficaces, les portails doivent être créés sous forme d'objets rectangles (comme pour les collisions).

Pour que le code puisse gérer le portail correctement, il faut éditer ses propriétés:

Propriétés	
Propriété	Valeur
▼ Objet	
ID	24
Modèle	
Nom	intro-prof
Classe	portal
Visible	<input checked="" type="checkbox"/>

La classe doit être **portal** et **nom** doit correspondre exactement au nom de la carte que tu veux ouvrir (ex: intro-sara).

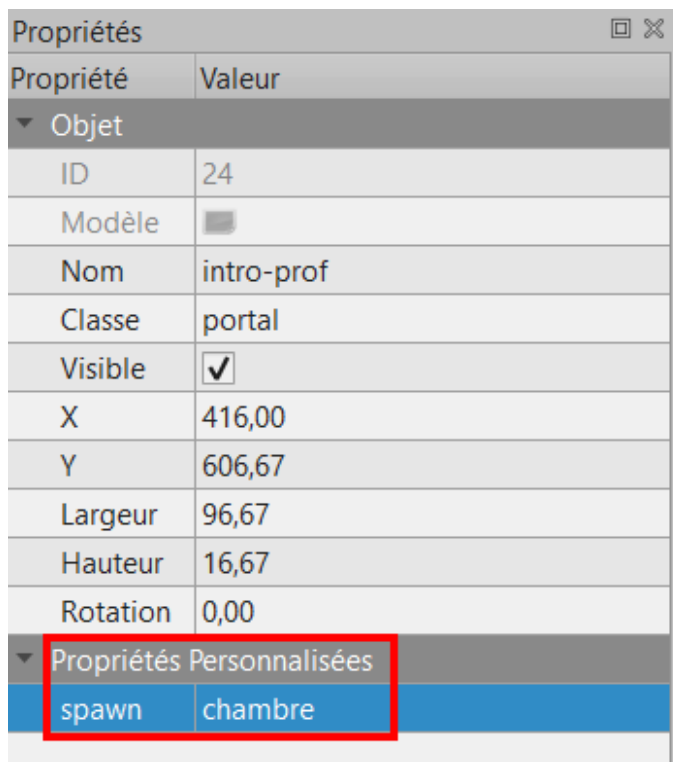
Généralement, les portails seront créés dans une porte.


Il ne faut pas que les portails et les points de spawn soient trop près l'un de l'autre sinon ton personnage risque d'apparaître dans le portail et de retourner directement dans la carte d'où il vient.

Enfin, dernière subtilité: Il est possible qu'une carte ait plusieurs points de spawn. Ce sera généralement le cas de ta carte principale à partir de laquelle plusieurs endroits peuvent être explorés. Dans ce cas, tes points de spawn devront porter des noms différents. Et le portail "de retour" (par exemple, quand tu passes par la porte pour sortir de la maison) Devra indiquer explicite du point où ton personnage doit réapparaître. Ceci renforcera l'impression

de réalisme. Sinon ton personnage risque de réapparaître très loin de l'endroit d'où il vient. Ce qui pourrait paraître bizarre...

Pour spécifier le nom du point de spawn où ton personnage doit apparaître en sortant de la carte, tu dois utiliser des **propriétés personnalisées** dans tiled. Ajoute une propriété qui s'appelle ***spawn*** et donne-lui comme valeur le nom du point où tu veux que ton personnage réapparaisse:



Propriété	Valeur
▼ Objet	
ID	24
Modèle	
Nom	intro-prof
Classe	portal
Visible	<input checked="" type="checkbox"/>
X	416,00
Y	606,67
Largeur	96,67
Hauteur	16,67
Rotation	0,00
▼ Propriétés Personnalisées	
spawn	chambre