

# Liste des entités Disponibles sur Tiled

---

Les entités suivantes peuvent être placées sur une map à l'aide de *Tiled*.

## Note sur certaines propriétés communes

### Propriétés "buttons"

- Plusieurs boutons peuvent être spécifiés à l'aide d'un espace.
- Si plusieurs boutons sont spécifiés, l'action sera effectuée quand tous les boutons sont pressés
- Il est possible d'inverser l'état requis d'un bouton en ajoutant "!" devant son nom

### Propriétés "chemins"

- Pour les images il faut ajouter 😞 afin d'utiliser le système de ressource QT. Les chemins peuvent être absolus afin de rapidement tester une texture.
- Pour les musiques il faut spécifier la musique en utilisant /assets/sounds/<music>
- Pour les sons, une liste est disponible dans sounds.cpp (actuellement: leverON, leverOFF, magnetON, magnetOFF)

## Rectangles

---

La plupart des entités supportent le positionnement et le changement de taille du rectangle. Certaines entités supportent également la rotation.

## graphic

Permet d'afficher des images, animations avec ou sans parallax. Supporte la rotation

### Components attachés

Animation, Image, Parallax

### Propriétés

Nom	Description
parallax (float)	Compris entre -1.0 et 1.0 Définit la profondeur (1 = background, -1 = foreground, 0 = pas de parallax)
texture (string)	Chemin de la texture à utiliser
animationName (string)	Le nom d'une "liste d'animation". Définit dans AnimationFactory. (Ex: lever, door, insideDoor, ...)
animationToPlay (string)	Le nom de l'animation à jouer dans la liste définie par animationName. (Ex: active, idle,)

Nom	Description
buttons (string)	Si défini, animationToPlay ne marchera pas. Définit quel animation doit être affichée (idle, start, active, end) en fonction de l'état des boutons. Contrôle également si la texture doit être affiché ou non

## sound

Permet de jouer des sons en réaction à des boutons

### Components attachés

SoundButtonReactor

### Propriétés

Nom	Description
buttons (string)	Permet de déterminer quand le son doit être joué
start (string)	Son à jouer quand les boutons sont pressés.
active (string)	Non implémenté
end (string)	Son à jouer quand les boutons ne sont plus pressés
idle (string)	Non implémenté

## button

Zones pouvant être activé / désactivé au contact d'une entité (ou éventuellement en pressant une touche du clavier). L'état des boutons sont utilisé par de nombreux composants pour effectuer diverses actions.

### Components attachés

GameButton

### Propriétés

Nom	Description
name (string)	Le nom du bouton, est utilisé par d'autres objets pour obtenir son état
invertOnOff (bool)	Inverse l'état de sorti du bouton si activé
pressedDurationInTick (int)	La duration en tick que le bouton doit resté pressé avant qu'il se désactive par lui même. Si définit à 0 le bouton ne peut pas être pressé
stayPressed (bool)	Overrides pressedDurationInTick. Si sélectionné, il n'est plus possible de désactivé le bouton après qu'il ait été pressé

Nom	Description
requiredKey (string)	La touche à presser pour interagir avec le bouton. (LEFT RIGHT JUMP INTERACT ZIP) Laisser vide pour pouvoir presser le bouton à son contact
isTogglable (bool)	Overrides stayPressed et pressedDurationInTick. Autorise l'utilisateur à désactiver le bouton quand il est repressé. Le bouton ne sera plus désactivé automatiquement si sélectionné. (Ne doit pas être utilisé sans requiredInput)
requiredButtonsToPress (string)	Une liste de bouton qui doivent être pressé pour autoriser ce bouton à être pressé
requiredButtonsToRelease (string)	Une liste de bouton qui doivent être pressé pour autoriser ce bouton à être désactivé
pressableBy (string)	Le nom du HitboxComponent autorisé à presser le bouton. Empêche le bouton de fonctionner si utilisé avec requiredKey. Laisser vide pour qu'il soit pressable par le joueur

## door

Charge une nouvelle map lors d'une interaction. Peut également charger les crédits et être activée / désactivée

### Components attachés

Door, Animation, Parallax

### Propriétés

Nom	Description
animation (string)	Nom de l'animation à créer pour la porte (généralement door ou insideDoor)
mirrorX (bool)	Inverse l'orientation de la porte si coché
targetMap (string)	Chemin de la map à charger. Ne pas oublier 🐼 pour utiliser les ressources QT
targetSpawn (string)	Nom du spawn ou le player doit être placé, définir à "default" en cas de doute
isFinalDoor (bool)	Si <i>true</i> , ignore les paramètres précédents pour à la place charger les crédits

## spark

Une entité poursuivant le joueur, le tuant au contact.

### Components attachés

Animation, Spark, CircleHitbox

### Propriétés

Nom	Description
speed (float)	Facteur de modification, sur une échelle de 0 à 5, de la vitesse à laquelle la spark se dirigera vers le joueur
radius (float)	Rayon d'activation de la spark
hitboxName (string)	Nom d'une hitbox supplémentaire à affecter (faisant la taille de la spark). Peut être utilisé pour les rails électriques

## hurt

Zone pouvant tuer certaines entités au contact.

### Components attachés

SquareHitbox

### Propriétés

- Aucune

## invisible-wall

Mur de collision invisible.

### Components attachés

SquareHitbox

### Propriétés

- Aucune

## camera

Centre la caméra sur son entité pendant un certain temps quand activé.

### Components attachés

CameraSequence

### Propriétés

Nom	Description
buttons (string)	Liste des boutons devant être pressé pour trigger la séquence
speed (float)	Vitesse de déplacement de la caméra
cancelAfter (int)	Nombre de tick avant de remettre la caméra sur le joueur
lockPlayerInput (bool)	Désactive le PlayerInputComponent de l'entité si sélectionné

Nom	Description
playOnce(bool)	Permet de jouer la séquence seulement une fois

## text

Affiche du texte dans un rectangle

Components attachés

Text

Propriétés

Nom	Description
buttons (string)	Liste des boutons devant être pressé pour afficher le texte
text (string)	Texte du bouton
fontSize (int)	Taille de la font du texte

## story-activation

Fonctionne comme un GameButtonComponent mais immobilise les entités qui touchent son hitbox.

Components attachés

StoryAnimation

Propriétés

Nom	Description
name (string)	Nom du bouton associé au StoryAnimation

## story-magnet

S'immobilise lors du contact avec certaines hitboxes, joue un son et charge une autre map.

Components attachés

StoryMagnet, Animation

Propriétés

Nom	Description
buttons (string)	Liste des boutons devant être pressé pour activer la physique du magnet
targetMap (string)	Map devant être chargée quand le bouton entre contact avec certaine entités

# Tiles

Certaines entités utilisent le système de tile, permettant d'avoir un retour visuel plus agréable.

## Magnet zipper

Attire certaines entités de façon linéaire

### Components attachés

MagnetZipper

Nom	Description
direction (string)	Orientation de l'aimant (left/right/up/down)
speed (float)	Vitesse de déplacement de l'entité une fois dans le champ
h (float)	Nombre de tiles de l'épaisseur du champ magnétique l'aimant (généralement 1) (= hauteur quand l'aimant pointe à droite/gauche)
w (float)	Nombre de tiles de la largeur du champ magnétique (= largeur quand l'aimant pointe à droite(gauche)
buttons (string)	Boutons requis pour activer l'aimant. Laisser vide pour toujours activé
noAnimations (bool)	Empêche le zipper de générer des animations si coché

## Magnet jumper

Repousse certaine entités tel un "jumper" Note: Peut être tourné afin de changer la direction

### Components attachés

MagnetJumper, ParticleSpawner, Animation

Nom	Description
force (float)	Vitesse à donner à l'entité quand elle touche l'aimant
buttons (string)	Boutons requis pour activer l'aimant. Laisser vide pour toujours activé

## Magnet gravity

Attire / repousse certaines entités en fonction de la distance à son centre de gravité

### Components attachés

MagnetGravity, Animation, GenericRender,

Nom	Description
force (float)	Vitesse à donner à l'entité quand elle touche l'aimant
radius(float)	Rayon d'effet de l'aimant
buttons (string)	Boutons requis pour activer l'aimant. Laisser vide pour toujours activé

## box

Une boite affecté par les champs magnétiques. Peut être déplacé par le joueur

### Propriétés

- Aucune
- 

### Components attachés

Image, Physics, MagnetZipperReactor, MagnetJumperReactor, MagnetGravityReactor (x2), MagnetGravity, SquareHitbox (2x)

## Player spawn

L'entité contrôlé par le joueur, possède un aimant qui peut être activé/désactivé. Permet également de créer un "Story Player"

### Components attachés

Animation, MagnetGravity, SquareHitbox (x2), PlayerInput, Physics, HurtReactor, MagnetZipperReactor, MagnetJumperReactor, MagnetGravityReactor, ParticleSpawner

### Propriétés

Nom	Description
animationName (string)	Le nom correspond à l'animation qui sera utilisé par le player généré (player ou playerStory recommandé)
spawnName (string)	Le nom correspond à ce spawn. Utiliser "default" si un seul spawn présent
h (Float)	La hauteur du joueur à spawn
w (Float)	La largeur du joueur à spawn

Note : entrer la valeur **playerStory** dans **animationName** provoquera la création d'une entité player sans les composants liés au aimants.

## Map

La map possède également des propriétés générale au niveau.

## Propriétés

### Components attachés

Nom	Description
zoom (float > 0)	Spécifie le zoom de la camera pour la map actuelle
musicPath (string)	Chemin de la musique à jouer pour ce niveau. DOIT COMMENCER PAR "/" (n'utilise pas les ressources, relatif au chemin de l'executable)

## Entités non créable sur Tiled

Certaines entités sont créés dans le code.

### PlayerCredits

L'entité affichée dans l'écran des crédits, ne peut pas être contrôlé.

### Components attachés

Animation

### PlayerStory

L'entité contrôlé par le joueur au début du jeu, ne possède pas d'aimant. Peut être créé à l'aide du PlayerSpawn

### Components attachés

Animation, SquareHitbox (x2), PlayerInput, Physics, HurtReactor