

# Liste des composants disponibles

---

- Les composants en *italique* sont des composants virtuels pure
- "Activé" dans ce document signifie soit que les boutons de jeu associé au component sont tous pressé ou que le component a été notifié d'une interaction (voir InteractiveComponent dans la documentation)

## Debug

- DebugComponent
  - Affiche le boundingRect de l'entité
- DebugHitboxComponent
  - Affiche les SquareHitboxComponent attaché à l'entité, peut être ajouté automatiquement avec le DebugComponent
- DebugTextComponent
  - Affiche la position et taille de l'entité, peut afficher un texte personnalisé

## Gameplay

- CameraSequenceComponent
  - Centre la caméra sur son entité pendant un certain temps quand activé.
- DoorComponent
  - Notifie **GameScene** de la nouvelle map qu'il faut charger quand activé.
- GameButtonComponent
  - Crée une zone nommée pouvant être activée (entité interagissant avec, joueur pressant un bouton dans la zone du game bouton). Contient une méthode statique permettant d'obtenir si un bouton est pressé ou non.
- HurtReactorComponent
  - Tue une entité entrant en contact avec sa hitbox (changement d'animation et désactivation du PlayerInputComponent) Peut éventuellement recharger la map actuelle.
- PhysicsComponent
  - Ajoute de la physique a une entité, gère la gestion des collisions avec les hitbox de "mur".
- PlayerInputComponentComponent
  - Gère le PlayerState (running, standing, jumping, ...) du joueur en fonction des touches pressées.
- SoundButtonReactorComponent
  - Joue un son quand activé / désactivé
- Spark
  - Chasse certaines entités quand elles rentrent dans son rayon d'action. Possède un HurtComponent pour tuer l'entité en cas de contact.
- StoryActivationComponent
  - Fonctionne comme un GameButtonComponent et immobilise les entités qui touchent son hitbox.
- StoryMagnetComponent
  - S'immobilise lors du contact avec certaines hitboxes, joue un son et charge une autre map.

## Graphics

- *GraphicsComponent*
  - Classe de base devant être hérité pour tout les composants souhaitant dessiner.
- *GenericRenderComponent*
  - Permet de dessiner n'importe quoi (utile pour ce qui n'est pas réutilisé). Prend une fonction à dessiner en argument.
- *AnimationComponent*
  - Affiche différentes animations.
- *GUILItemComponent*
  - Définit la position du parent à celle de la caméra, afin d'afficher des éléments dans l'interface.
- *ImageComponent*
  - Affiche une image statique (peut être tournée, inversée, ...)
- *ParallaxComponent*
  - Définit la position du parent en fonction de celle de la caméra et d'un facteur de parallax.
- *ParticleSpawnerComponent*
  - Crée des particules. Prend 3 fonctions en arguments afin de créer, afficher et mettre à jour les particules.
- *TextComponentComponent*
  - Affiche du texte dans un cadre quand activé.
- *TransitionComponent*
  - Affiche une animation personnalisée et appelle un callback après la durée spécifiée.

## Hitboxes

- *HitboxComponent*
  - Représente une zone nommée. Le nom de la zone est le nom du component. Contient une méthode permettant de récupérer toutes les hitboxes possédant un certain nom. Les deux hitboxes disponibles sont le rectangle et le cercle.
- *HitboxReactorComponent*
  - Possède une méthode appelée quand son entité intersecte avec une autre hitbox spécifique.
- *CircleHitboxComponent*
  - Zone de collision ronde.
- *SquareHitboxComponent*
  - Zone de collision carrée.
- *InteractiveComponent*
  - Zone de collision spéciale réagissant au clavier.
- *InteractiveReactorComponent*
  - Zone du jeu où il est possible au joueur d'interagir, typiquement avec la touche INTERACT. Doit être couplé avec un *PlayerInputComponent* pour fonctionner.

## Magnets

Voir: Les différents types d'aimant dans le rapport

- *MagnetGravityComponent*
  - Crée une *CircleHitbox* et stocke les propriétés d'un aimant gravitationnel
- *MagnetGravityReactorComponent*

- Change les forces du PhysicsComponent de l'entité quand elle entre dans le champ magnétique d'un aimant gravitationnel
- MagnetJumperComponent
  - Crée une SquareHitbox et stocke les propriétés d'un aimant jumper
- MagnetJumperReactorComponent
  - Change les forces du PhysicsComponent de l'entité quand elle entre dans le champ magnétique d'un aimant jumper
- MagnetZipperComponent
  - Crée une SquareHitbox et stocke les propriétés d'un aimant zipper
- MagnetZipperReactorComponent
  - Change les forces du PhysicsComponent de l'entité quand elle entre dans le champ magnétique d'un aimant zipper

## Misc

- *Component*
  - La classe mère de tout les autres composants. Contient principalement une fonction d'update à redéfinir.