|  |
| --- |
|  |
| Guardian of Time |
| Manuel de fonctionnement |

|  |
| --- |
| [Catoryu] Rui Manuel Mota Carneiro  [Leco] Leandro Saraiva Maia  [FrankamaNaxos] Maxim Golay  [RedAxeLight] Alexandre Baseia  [Johnson] Théo Esseiva  [BeatNation] Leo Zmoos  11/07/2018 |

Table des matières

[Blocs 2](#_Toc519150749)

[1. Définition 2](#_Toc519150750)

[2. Liste des paramètres 2](#_Toc519150751)

[3. Condition des blocs 2](#_Toc519150752)

[Entités 3](#_Toc519150753)

[1. Définition 3](#_Toc519150754)

[2. Liste des paramètres 3](#_Toc519150755)

[Evénements 4](#_Toc519150756)

[1. Définition 4](#_Toc519150757)

[2. Liste des conditions 4](#_Toc519150758)

[Acteurs 5](#_Toc519150759)

[1. Définition 5](#_Toc519150760)

[2. Id 5](#_Toc519150761)

[3. Initialisation alliés/hostiles 5](#_Toc519150762)

[4. Table de tâches 5](#_Toc519150763)

# Blocs

## Définition

Un bloc est un carré qui est placé selon une grille et ne peut donc pas être placé librement

## Liste des paramètres

Un bloc a les paramètres suivants :

* isSolid
  + Le joueur entre en contact avec le bloc
* isLiquid
  + Réservoir à liquide pouvant aller de 0% à 100% de remplissage
* isDestructible
  + A des points de vie et peut être détruit
  + Un script est déclenché
* isVisible
  + Visible dans la salle
* isGravityAffected
  + Affecté par la gravité
* isFireProof
  + Résiste au feu
* isFireStoppable
  + Eteint les acteurs en feu entrant en collision avec

## Condition des blocs

Les blocs détectent les conditions suivants (chaque condition remplie déclenche un script si il y en a défini) :

* onTouch
  + Le bloc est touché
  + Prend une valeur en paramètre
    - 0 = N’importe où
    - 1 = En haut
    - 2 = En bas
    - 3 = A gauche
    - 4 = A droite
* onTtlReach
  + On peut donner une durée à un bloc pour qu’un script se déclenche lorsque la durée atteint zéro (par exemple avec une durée de 1.5 secondes le script s’exécutera dans 1.5 secondes)
  + La durée du bloc est achevée

# Entités

## Définition

Une entité est un objet dans la salle pouvant avoir la forme d’un rectangle ou d’un cercle. C’est l’équivalent d’un bloc mais qui peut se mettre dans différentes positions (pas coincé dans une grille) et dans différentes formes.

## Liste des paramètres

Une entité a les paramètres suivants :

* isSolid
  + Le joueur entre en contact avec l’entité
* isDestructible
  + A des points de vie et peut être détruit
  + Un script est déclenché
* isVisible
  + Visible dans la salle
* isGravityAffected
  + Affecté par la gravité
* isFireProof
  + Résiste au feu
* isFireStoppable
  + Eteint les acteurs en feu entrant en collision avec

# Evénements

## Définition

Un événement est un rectangle invisible, qui déclenche un script lorsqu’une condition est remplie.

## Liste des conditions

Les événements peuvent être déclenché par les conditions suivantes :

* onTouch
  + Le joueur est en contact avec l’événement
* onTtlReach
  + On peut donner une durée à un événements pour qu’il se déclenche lorsque la durée atteint zéro (par exemple avec une durée de 1.5 secondes l’événement s’exécutera dans 1.5 secondes)
  + La durée de l’événement est achevée
* onEnter
  + Le joueur vient d’entrer dans l’événement
* onLeave
  + Le joueur vient de sortir de l’événement

# Acteurs

## Définition

Un acteur est un objet qui peut bouger spontanément pour accomplir des tâches qui lui sont attribuées.

## Id

Chaque acteur a un id unique et un id de faction. L’id unique sert à désigner l’acteur individuellement alors que l’id de faction permet de désigner un groupe d’acteur ayant le même id de faction.

Les acteurs ont tous deux tables, une table « alliés » et une table « hostiles » qui contiennent des id unique.

## Initialisation alliés/hostiles

Lors de l’initialisation d’un acteur, l’acteur initialisé va regarder l’id de faction de chaque autre acteur dans la salle et va utiliser les relations entre son id de faction et l’id de faction des autres acteurs pour classer leurs id uniques dans 3 catégories :

* Alliés
* Hostiles
* Passif

A noter que les acteurs « passif » ne sont stockés dans aucune table.

## Table de tâches

Tous les acteurs (sauf le joueur) ont une table qui contient la liste des tâches à effectuer sous forme d’id de tâches. L’acteur va toujours essayer d’effectuer la tâche qui se trouve en premier dans son tableau.

Par exemple, on pourrait avoir un acteur hostile au joueur qui a dans sa table de tâches deux choses : {5, 1}. Pour l’exemple, l’id 5 sera la tâche « Arrêter d’être en feu » et l’id 1 sera la tâche « Attaquer un acteur hostile ». D’abord l’acteur va chercher à arrêter d’être en feu en trouvant un point d’eau ou en attendant que le feu s’éteigne. Une fois le feu éteint, l’acteur va donc chercher à attaquer l’acteur qui se trouve en premier dans sa table « hostiles » en trouvant sa position et en allant a porté d’attaque.