|  |
| --- |
| is::Level Editor |
| Guide utilisateur |
|  |

Table des matières

[***Introduction*** 2](#_Toc37104991)

[***I.*** ***C’est quoi is::Level Editor?*** 2](#_Toc37104992)

[***II.*** ***Comment is::Level Editor gère-t-il les objets*** 2](#_Toc37104993)

[***III.*** ***Comment utiliser is::Level Editor*** 2](#_Toc37104994)

[***1.*** ***Comment créer un Nouveau Niveau ?*** 3](#_Toc37104995)

[***2.*** ***Comment charger un Niveau ?*** 3](#_Toc37104996)

[***3.*** ***C’est quoi la sauvegarde auto ?*** 3](#_Toc37104997)

[***4.*** ***A quoi sert l’option Charger Texture ?*** 3](#_Toc37104998)

[***5.*** ***Comment utiliser l’éditeur de niveau ?*** 3](#_Toc37104999)

[***6.*** ***Exemple de niveau créé*** 5](#_Toc37105000)

Introduction

Le but de ce document est de vous expliquer comment l’éditeur de niveau is::Level Editor fonctionne, d’apprendre à créer des niveaux et à les intégrés au moteur de jeu.

1. C’est quoi is::Level Editor?

is::Level Editor est un éditeur de niveau du moteur de jeu is::Engine, il classe les objets du niveau en **5 types**:

* **Blocs** : ils servent de masque de bloc, ascenseur (plateforme déplaçable) bref tout ce qui est lié aux plateformes solides
* **Bonus** : sert à ajouter la santé, les items du joueurs, vie bonus, etc.
* **Ennemis** : Représente les ennemis d’un niveau
* **Le joueur** : Permet de positionner le joueur dans le niveau
* **Gameplay Obj** : Sert à ajouter les objets qui gèrent le mécanisme d’un niveau (objet goal, interrupteur, contrôleur d’objets)
* **Tiles** : Sert à ajouter des objets décors qui seront plus tard transformés en tile map par le moteur

1. Comment is::Level Editor gère-t-il les objets

* Chaque objet de l’éditeur est associé à un **numéro unique** qui permet de **l’identifier**.
* Chaque **type d’objet** représente **un intervalle** dans lequel sont **regroupés les objets**.

Les intervalles de chaque type d’objet :

* Les Objets de type Bloc commence de 0 à 99
* Les Objets de type Bonus commence de 100 à 199
* Les Objets de type Ennemi commence de 200 à 299
* Les Objets de type Player commence de 300 à 399
* Les Objets de type Game Play commence de 400 à 499
* Les Objets de type Tuiles commence de 500 à ∞
* Chaque type d’objet est lié à **une texture**. Les textures permettent de définir **le nombre d’objet** qu’un **type d’objet** aura en fonction **du nombre de sous image** qui la compose.

Voici un exemple pour mieux comprendre :

 Cette image est la texture du **type d’objet Bonus** elle est composée de **3 sous images** donc ça fait **trois (3) objets** pour le **type d’objet Bonus** qu’on peut utiliser pour créer un niveau. Les textures sont composées d’image de taille **32x32**. L’éditeur de niveau s’occupe de la découper pour vous lors de son utilisation. ***(Vous pouvez charger vos propre texture d’image mais à voir plus tard dans le document)***

Donc ici le numéro du premier sous image sera 100, le deuxième 101 et ainsi de suite. Notez que ces grâces à ces numéros que le moteur de jeu arrive à interpréter les données du niveau et à déterminer comment les niveaux seront construits avec les objets.

1. Comment utiliser is::Level Editor?

Cette partie vous montre comment utiliser les différentes options de l’éditeur de niveau.

***(S’il vous plaît voir la page suivante)***

|  |  |  |
| --- | --- | --- |
| Figure | Figure | Figure |

1. Comment créer un Nouveau Niveau ?

* Cliquer sur **Nouveau Niveau** (voir **Figure 1**)
* En suite cliquer sur **la console** (voir **Figure 2**) pour renseigner **la taille du niveau** (la taille s’exprime en case, 1 case = 32 pixels) et le **nom du niveau** (s’il vous plait éviter les espaces)

1. Comment charger un Niveau ?

* Cliquer sur **Charger Niveau** (voir **Figure 1**)
* Choisissez **le fichier sauvegarde (.SAV)** grâce à la boite de dialogue et cliquer sur **Ouvrir**

1. C’est quoi la sauvegarde auto ?

La sauvegarde auto vous permet de sauvegarder automatiquement les données de votre niveau en cour d’édition sans votre intervention. Par défaut si vous ne sélectionnez pas l’emplacement du fichier de sauvegarde de votre niveau, il sera stocké dans le répertoire d’exécution de l’éditeur de niveau.

* ***Comment l’activer ou le désactiver ?***
* Cliquer sur **Activer sauvegarde auto pour le désactiver et faite le même geste pour l’activer** (voir **Figure 1**)

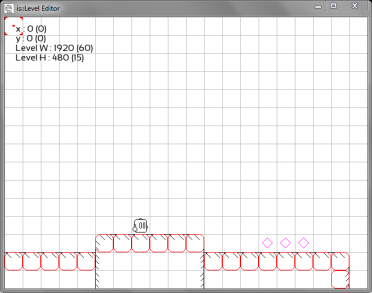
1. A quoi sert l’option Charger Texture ?

Cette option (voir **Figure** **1**) vous permet d’utiliser vos propres textures à la place de ceux qui sont pré enregistrées.

Une fois qu’elle est activée une proposition vous sera faite lors de l’édition d’un nouveau niveau (voir **Figure 3**) afin de charger vos propres images pour créer les objets de l’éditeur de niveau.

1. Comment utiliser l’éditeur de niveau ?

Cette partie vous montre comment utiliser les outils de l’éditeur (voir **Figure 4**) pour créer votre niveau.



**Figure 4**

|  |  |
| --- | --- |
| 1. ***Comment choisir un objet à ajouter au niveau ?***  * Appuyer sur le **bouton droit** de la sourie pour afficher **l’option de choix** (voir **Figure 5**) * Ensuite choisissez un **type objet** (dans les 6 premières lignes) * Appuyer sur le **bouton gauche** de la sourie pour **insérer l’objet** * Pour **changer l’objet d’un type** utiliser la **roue vertical** de la sourie ou les **touches « Haut » et « Bas »**  1. ***Comment effacer un objet du niveau ?***  * Appuyer sur le **bouton droit** de la sourie pour afficher **l’option de choix** (voir **Figure 5**) * Cliquer sur **Effacer Objet** pour afficher **le curseur de suppression** * Cliquer sur un objet pour le supprimer   Vous pouvez aussi utiliser **la touche de suppression** pour afficher **le curseur de suppression**   1. ***Comment défiler les pages du niveau ?***  * Mettez **le curseur** de la sourie au **bord de la fenêtre** (voir **Figure 4**)  1. ***Comment sauvegarder le niveau ?***  * Appuyer sur le **bouton droit** de la sourie pour afficher **l’option de choix** (voir **Figure 5**) * Cliquer sur **Sauvegarder Niveau** * renseigner le **nom du fichier**, en suite choisissez **l’emplacement de sauvegarde** et cliquer sur **Enregistrer**  1. ***Comment exporter le niveau vers le moteur is::Engine ?***  * Appuyer sur le **bouton droit** de la sourie pour afficher **l’option de choix** (voir **Figure 5**) * Cliquer sur **Exporter Niveau** * renseigner le **nom du fichier**, en suite choisissez **l’emplacement de sauvegarde** et cliquer sur **Enregistrer** * ***Comment intégrer le niveau au moteur de jeu ?*** * Mettez le fichier exporté dans le dossier **levels** du moteur (Chemin : **Project/app\_src/levels**) * Ouvrer le fichier **Level.h** qui se trouve dans **levels** * Ajouter cette ligne pour inclure le niveau au projet :   **#include "../levels/nom\_du\_niveau\_genere.h"**   * Ajouter dans **enum LevelId** l’index du niveau **LEVEL\_n** (ici **n** représente le énième niveau) * Ajouter dans la fonction **getLevelMap()** cette ligne :   **case LEVEL\_n**  **: return NOM\_DU\_NIVEAU\_GENERE; break;**   1. ***Comment recharger un niveau ?*** (Attention cette opération supprimera vos données en cours)  * Appuyer sur le **bouton droit** de la sourie pour afficher **l’option de choix** (voir **Figure 5**) * Cliquer sur **Recharger Niveau**  1. ***Comment cacher les grilles de l’éditeur de niveau***  * **Appuyer sur la touche F3 pour cacher ou afficher les grilles**  1. ***Les raccourcis clavier***   Utiliser ces raccourcis clavier pour aller plus vite lorsque vous éditez un niveau :   * **Touche B** pour choisir un objet de type Bloc * **Touche I** pour choisir un objet de type Bonus * **Touche E** pour choisir un objet de type Ennemi * **Touche P** pour choisir le joueur * **Touche G** pour choisir un objet de type Gameplay * **Touche T** pour choisir un objet de type Tuile * **Touche Suppression** pour afficher le curseur de suppression * **Ctrl +Touche S** pour sauvegarder le niveau * **Ctrl +Touche R** pour recharger le niveau * **Ctrl +Touche H** pour exporter le niveau | Figure |

1. ***Exemple de niveau créé***

Vous pouvez trouver des niveaux exemples (qui ont été utilisés pour la démo du moteur de jeu) dans le répertoire **level\_backup** du projet.