

Module *PDFFile*

Module créé par Kitai, version 1.2 du 09/03/2013, développé sous le logiciel Adventure Game Studio version 3.2.2 bêta FR

Description : Ce module permet de compiler sous AGS un fichier texte balisé en fichier PDF.

1. Installation

Après avoir téléchargé et décompressé le fichier *PDFFile.zip*, ouvrez l'éditeur AGS et faites un clic droit sur l'onglet *Scripts* puis cliquez sur "*Import new script...*" (ou "*Importer script...*" en version française) puis choisissez le fichier *PDFFile.scm* que vous venez d'extraire de l'archive. Ça y est, le module est installé !

2. Utilisation

Pour utiliser ce module, vous devez avoir un fichier texte contenant l'intégralité du texte à écrire dans le fichier PDF de sortie. Ensuite, il vous suffira d'entrer ces commandes à l'endroit désiré dans le script :

```
PDFFile monPdf;  
pdf.Compile("fichier_source.txt", "fichier_cible.pdf");
```

Éventuellement, vous pouvez passer une troisième chaîne de caractères comme paramètre correspondant au nom du fichier dans lequel vous voudriez voir un rapport d'erreurs créé à la compilation :

```
PDFFile monPdf;  
pdf.Compile("fichier_source.txt", "fichier_cible.pdf", "rapport.txt");
```

3. Liste des balises

Attention, pour retourner à la ligne, vous devez laisser une ligne vierge dans le fichier (comme sur Wikipedia, par exemple). Comme cela :

```
Première ligne  
  
/centerDeuxième ligne
```

La séquence suivante n'aura **pas** l'effet désiré (une seule ligne, pas de centrage) :

```
Première ligne  
/centerDeuxième ligne
```

3.1 Balises en début de ligne

<code>/left</code>	Définit l'alignement à gauche pour la suite du texte
<code>/right</code>	Définit l'alignement à droite pour la suite du texte
<code>/center</code>	Définit l'alignement au centre pour la suite du texte
<code>/justified</code>	Définit un alignement justifié pour la suite du texte
<code>/x-marginN/</code>	Redéfinit la largeur des marges horizontales à N pouces
<code>/newline</code>	Insère une ligne vide
<code>/newpage</code>	Passé à une nouvelle page

3.2 Balises dans la ligne

<code>/bo</code>	Active/Désactive le gras
<code>/it</code>	Active/Désactive l'italique
<code>/sizeN/</code>	Redéfinit la taille de la police à N
<code>/fontPOLICE/</code>	Redéfinit la police utilisée pour POLICE
<code>//</code>	Affiche le caractère /

Les valeurs possibles pour POLICE sont les suivantes : *Courier*, *Helvetica*, *Times*, *Symbol* et *ZapfDingbats*.

Les polices *Symbol* et *ZapfDingbats* n'ont pas de version gras ni italique.

4. Aspect de la page

Avant de lancer la commande *Compile*, vous pouvez définir certains points de l'aspect de la page à l'aide des propriétés de votre instance *PDFFile*.
Par défaut, votre page sera au format A4, avec des marges d'environ 2cm. Elle utilisera initialement la police **Courier** d'une taille de **12 points**.

Voici un exemple de redéfinition de ces aspects avant compilation :

```
PDFFile monPdf;
// Format paysage A4
monPdf.Largeur = 835; // 835/72 = 11 pouces, soit environ 29cm
monPdf.Hauteur = 594; // 594/72 = 8 pouces, soit environ 21cm
monPdf.Marges_horizontales = 72; // 72/72 = 1 pouce, soit 2.54cm
monPdf.Marges_verticales = 72; // 144/72 = 2 pouces, soit 5.08cm
monPdf.Police = ePDFTimes; // La police utilisée en début de compilation sera Times
monPdf.TaillePolice = 10; // La taille de police utilisée en début de compilation sera
de 10 points
monPdf.Compile("fichier_source.txt", "fichier_cible.pdf", "rapport.txt"); // On compile
if (monPdf.Error) Display("Il y a eu des erreurs à la compilation. Consultez le
fichier 'rapport.txt' pour plus de détails."); // On informe s'il y a des erreurs
```