

TP 16

Utilisation des fichiers ini

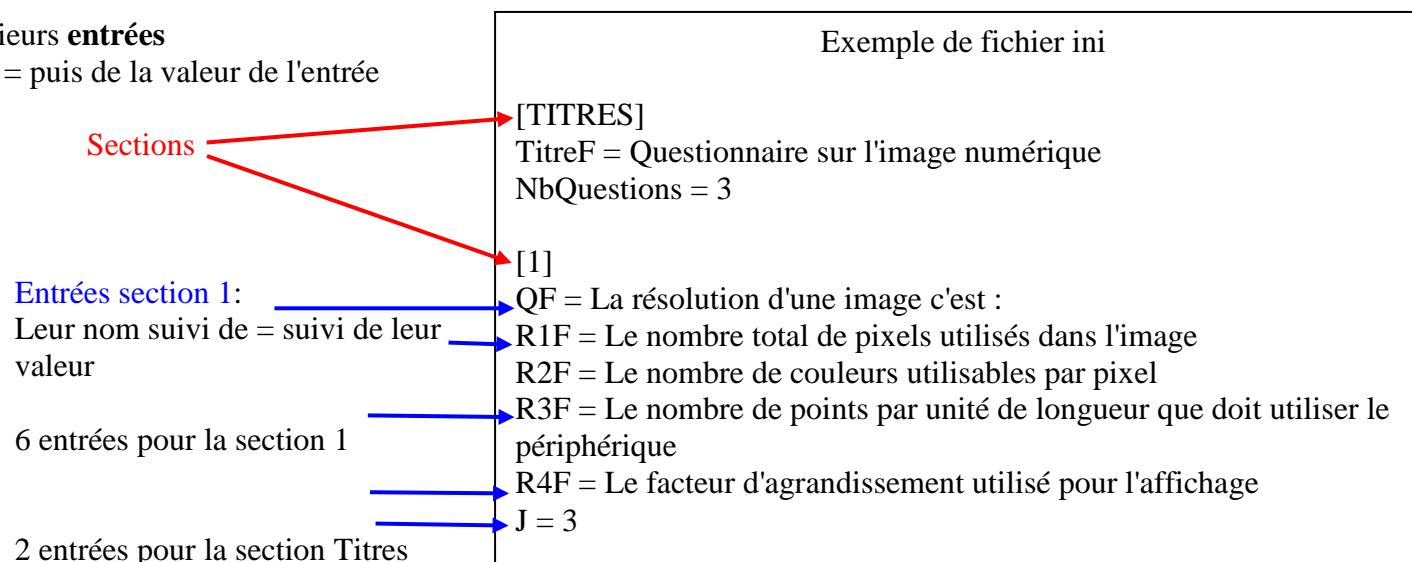
1°) Structure des fichiers ini

Un fichier ini est un document texte auquel on a attribué une extension .ini. On peut donc le créer tout simplement avec le bloc-note de Windows
 Pour que Médiateur puisse l'utiliser, il faut qu'il respecte une syntaxe particulière :

Les sections : leur nom est écrit entre crochets

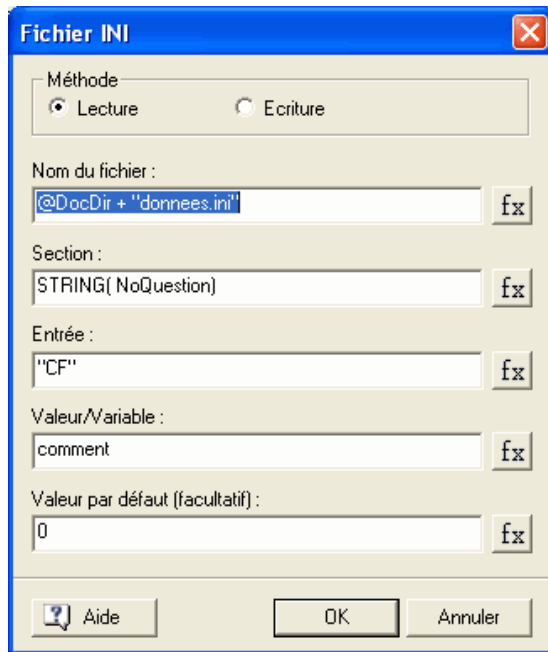
A l'intérieur de chaque section figurent une ou plusieurs **entrées**

Une entrée est composée d'un nom suivi d'un signe = puis de la valeur de l'entrée



2°) L'action fichier ini

Dans l'onglet externe des actions



On peut utiliser l'action fichier ini en Entrée de données (Méthode lecture cochée comme ci-contre) ou en mode écriture de données (Méthode Ecriture cochée)

Dans l'exemple ci-contre :

on lit à partir du **fichier donnees.ini** placé dans le même répertoire (@docdir) que le fichier médiateur (.md8)

On lit dans une section dont la valeur est contenue dans la variable NoQuestion. Comme on ne peut utiliser que du texte et que la variable NoQuestion est une variable numérique on convertit son contenu en chaîne de caractères (fonction STRING)

On lit l'entrée nommée CF et son contenu est stocké dans la variable nommée comment

La variable doit en principe être déclarée avant et la valeur de l'entrée sera stockée en tant que texte même si c'est un nombre.

2°) Allons-y

Rapatriez dans votre répertoire de travail le dossier TP16_fichier_ini0

Ouvrez le fichier TP16_fichier_ini0


Sur la page 1 se trouvent un certain nombre d'objets texte, c'est uniquement pour vous montrer une utilisation de la fonction RDN (voir les ressources de la page et la ligne de temps associée à l'événement page prête)

Dans le répertoire de travail se trouve un fichier texte nommé donnees.ini : c'est lui que nous allons utiliser comme source d'informations.

Le Titre

Sur cette page se trouve un objet Titre qui est vide. Nous allons aller chercher son contenu dans le fichier donnees.ini :

Commencez par créer une variable (Menu Document, Commande Variables, bouton Créer) : nommez la **titre**, portée **locale**, type **texte**.

Contenu du fichier données.ini	Actions à associer à l'événement Page Prête
<pre>[TITRES] TitreF = Questionnaire sur l'image numérique NbQuestions = 3 [1] QF = La résolution d'une image c'est : R1F = Le nombre total de pixels utilisés dans l'image</pre>	 <div style="display: flex; justify-content: space-around;"> <div data-bbox="750 406 1153 885"> <p>Fichier INI</p> <p>Méthode: Lecture</p> <p>Nom du fichier: @docdir + "donnees.ini"</p> <p>Section: "TITRES"</p> <p>Entrée: "TitreF"</p> <p>Valeur/Variable: Titre</p> </div> <div data-bbox="1176 478 1400 774"> <p>Dans le fichier donnees.ini, on lit l'entrée TitreF de la section TITRES et on stocke son contenu dans la variable Titre</p> </div> <div data-bbox="1422 438 1915 869"> <p>Définir propriété</p> <p>Objet: Titre</p> <p>Propriété: Text</p> <p>Valeur sous forme d'expression: titre</p> </div> <div data-bbox="1915 518 2161 662"> <p>La propriété Text de l'objet Titre reçoit le contenu de la variable titre</p> </div> </div>

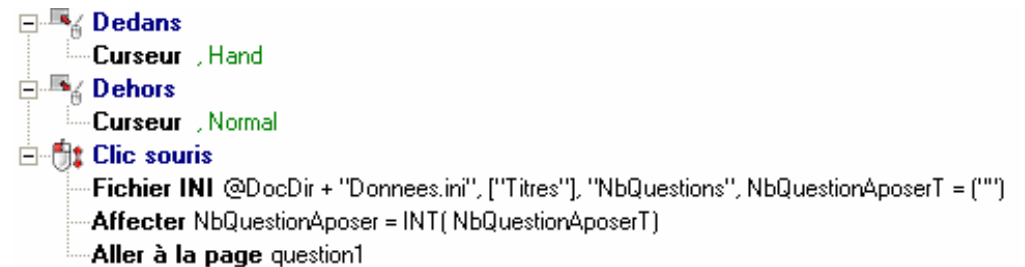
Le bouton Démarrer :

Ce bouton va servir à passer à la première page de questions. On va en profiter pour stocker le nombre de questions que l'on souhaite poser :

Commencez par créer **une variable NbQuestionAposer de type Entier et de portée globale**

Ainsi qu'une **variable NbQuestionAposerT de type Texte et de portée locale** (pour le stockage provisoire de l'information)

Rédigez événement et action du bouton Démarrer comme sur la comme ci-contre (vue script) ou ci-dessous (vue icônes).



3°) Les pages Question

Vous allez programmer la page Question1 qui pourra ensuite servir de modèle (uniquement par copier-coller) pour les autres pages question Le nombre total de questions est défini dans le fichier Donnees.ini et a été mémorisé dans la variable NbQuestionAposer (bouton démarrer page précédente).

Sur cette page figurent 5 objets Entrée vides pour l'instant (Question, Reponse01, Reponse02, Reponse03, Reponse04) 4 boutons cachés associés aux réponses (Button01 à Button04), un bouton OK, caché, deux objets texte exact et faux, cachés, un objet entrée Commentaire, caché, un objet entrée Ascore, caché et un bouton suite, caché.

Créez les variables suivantes :

NoQuestion (numéro de la question) type Entier, portée globale. **Question** (texte de la question) : type texte, portée globale

Questions_tirees (stockage des numéros de question déjà utilisés) type texte, portée globale.

Reponse1 (texte de la première proposition) : type texte, portée globale.

Reponse2 : idem

Reponse3 : idem

Reponse4 : idem

ReponseCorrecteT (stockage du numéro de la bonne réponse) type Texte, portée globale

Bonnerep (conversion de la valeur précédente en nombre) : type entier, portée globale. **Selection** (réponse selectionnee par l'utilisateur: : type entier, portée globale.

Score (comme son nom l'indique) : type entier, portée globale

Comment (Commentaire): type texte, portée globale

I. Commençons par les événements page prête

1 Une action Affecter va donner une valeur aléatoire à la variable NoQuestion la fonction RND tire un nombre aléatoire compris entre 1 et le nombre de questions à poser.

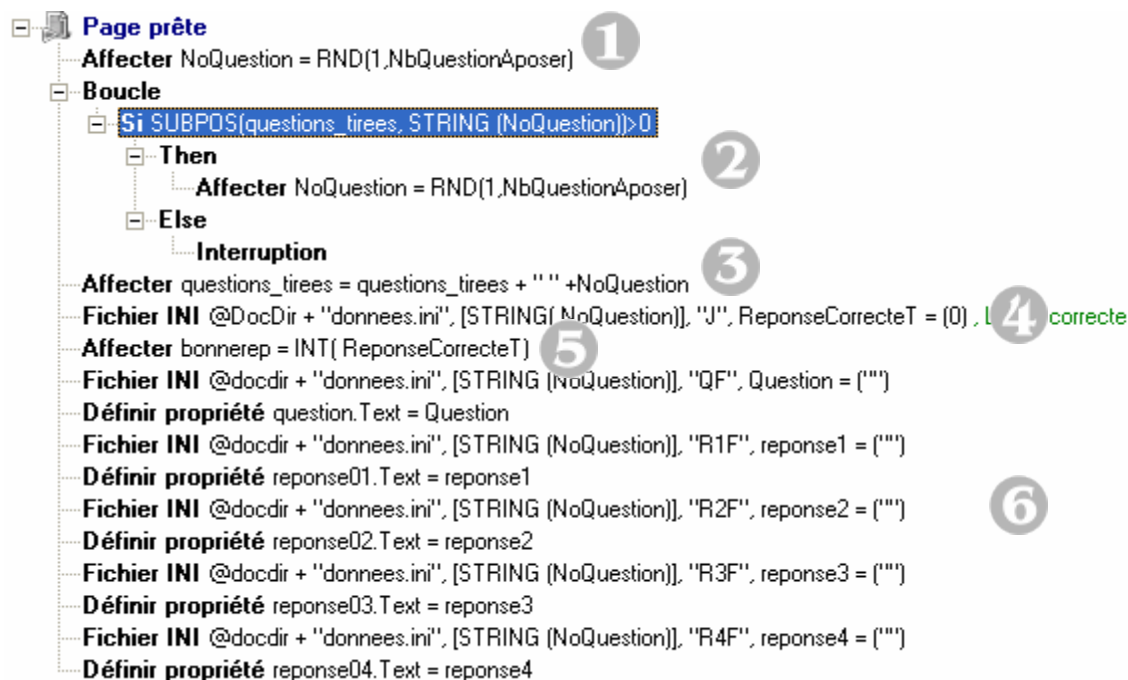
2 Si c'est la première question , pas de problème, mais si on en est à la deuxième, le tirage du numéro étant aléatoire, on peut très bien retomber sur le même numéro.

On va donc insérer une boucle dans laquelle on teste si le numéro de la question se trouve dans la variable stockant les numéros : si le numéro tiré se trouve dans la liste des numéros tirés alors SUBPOS rend un nombre entier position du numéro dans la liste sinon il rend zéro.

Donc si SBPOS ()>0 on retire un numéro sinon on interrompt la boucle.

3 On rajoute le numéro de question à la liste des questions tirées.























4 On va lire dans le fichier donnee.ini, le nombre associé à la bonne réponse à la question tirée : section STRING(NoQuestion), entrée J, variable de stockage : ReponseCorrecteT.



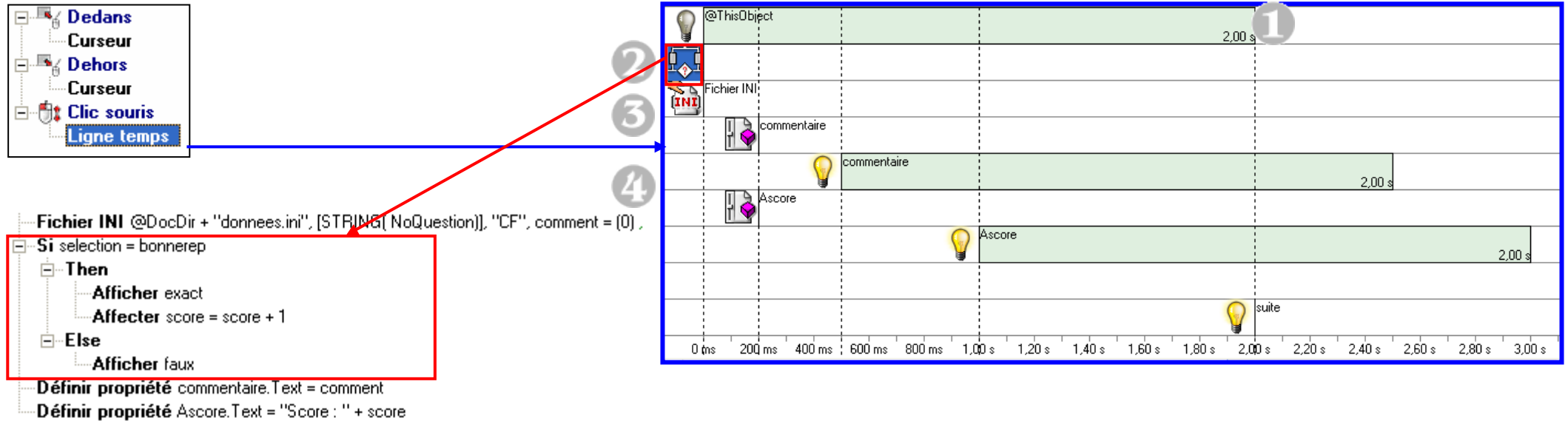
- 5 On affecte à la variable ReponseCorrecte la conversion en nombre de ReponseCorrecteT
- 6 Pour chacun des objets entrée posés sur la page, on va lire dans Donnees.ini la valeur correspondante et on la stocke dans la variable prévue.
 Pour le titre : le nom du fichier est toujours @docdir + "donnees.ini", la section STRING(NoQuestion), l'entrée QF et la variable de stockage Question.
 Pour la première réponse proposée : même fichier, même section, l'entrée est R1F et la variable de stockage Reponse1
 Idem pour les autres réponses : entrées R2F pour variable Reponse2...
 Après chaque stockage dans la variable appropriée, on Définit la propriété Text de l'objet en lui donnant la valeur de la variable correspondante
 Variable Question pour l'objet Question, variable Reponse1 pour l'objet Reponse01...

OUF

II. Evénements et actions sur chaque objet Reponse

Entrée Reponse01		Entrée Reponse02
<ul style="list-style-type: none">  Dedans  Curseur ,Hand  Dehors  Curseur ,Normal  Clic souris  Affecter selection = 1  Afficher Button01  Cacher Button02  Cacher Button03  Cacher Button04  Afficher OK 	<p>Changement de curseur en entrée et en sortie</p> <p>Lorsqu'on clique sur la réponse en question, on stocke le numéro de la réponse dans la variable selection (1 pour reponse 01, 2 pou reponse02...) on affiche le bouton associé pour indiquer la réponse sélectionnée (au cas où l'on change d'avis, on cache les autres boutons) et on affiche le bouton OK .</p>	<ul style="list-style-type: none">  Dedans  Curseur ,Hand  Dehors  Curseur ,Normal  Clic souris  Affecter selection = 2  Afficher Button02  Cacher Button01  Cacher Button03  Cacher Button04  Afficher OK

III. Le bouton OK



- 1 On cache le bouton OK (ici avec effet)
- 2 Si le numéro de la sélection correspond à la valeur de la bonne réponse (selection=bonnerep)
 Alors on affiche l'objet Exact
 Sinon on affiche l'objet Faux
- 3 On va lire dans le fichier ini le commentaire correspondant à la question tirée (entrée CF) et on le stocke dans la variable comment.
- 4 L'objet entrée Commentaire reçoit le contenu de la variable comment dans sa propriété texte. On le fait ensuite afficher
 L'objet entrée Ascore reçoit dans sa propriété texte "Sore : " + score. On le fait ensuite afficher
 On fait ensuite afficher le bouton suite (passage à la question suivante : @next) sauf s'il s'agit de la dernière question.

Voilà, il ne reste plus qu'à recopier cette page pour avoir autant de pages que de questions rédigées dans le fichierdonnees.ini

4°) Et si j'avais besoin d'un fichier multilingue ?

Pour ne pas que ce soit fastidieux et pour gagner du temps, le fichier est tout prêt, mais si vous souhaitez le faire vous-même vous pouvez partir du fichier précédent.

Rapatriez le dossier TP16_bilingue dans votre répertoire de travail : le fichier .ini associé a été nommé donnees_multilingues.ini, mais, pour plus de simplicité, vous pouvez continuer à travailler avec donnees.ini en le complétant.

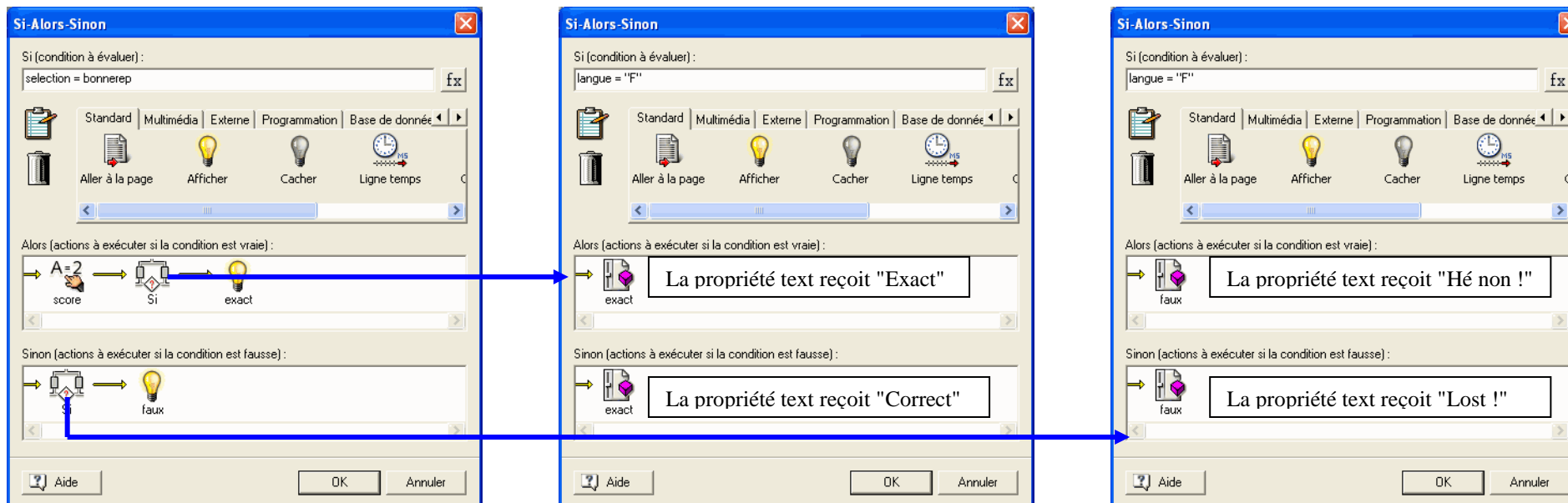
Pour chaque section, les entrées CF, QF, RIF etc. on été dupliquées, nommées CA, QA, R1A etc. et traduites en anglais. Les entrées correspondant à des valeurs numériques restent inchangées.

Nous avons besoin d'une nouvelle variable : langue, de type texte, de portée globale qui contiendra F si la langues choisie est le français, A si c'est l'anglais.

Sur la première page, on fait choisir la langue désirée et on fait afficher le bouton "commencer" correspondant à la langue choisie.

Sur les pages questions : à chaque appel au fichier ini, il faut modifier l'entrée : au lieu de lire l'entrée "QF" ou "QA" , on va lire l'entrée "Q" + langue. "R1F" devient "R1" + langue et ainsi de suite.

Le seul petit problème concerne les objets texte Exact et Faux. Soit on crée de nouveaux boutons avec le texte en anglais, soit (c'est le choix effectué dans l'exemple) on remplace les objets texte par des objets Entrée afin de pouvoir modifier le texte affiché en utilisant l'action définir propriété. Et on modifie l'affichage exécuté lors du clic que le bouton OK :



On peut envisager de placer des boutons de changement de langue sur chaque page, de rajouter une troisième langue, les modifications sont très simples à faire, une fois la traduction effectuée !!