

Fichiers INI

Mediator est capable de lire, ou d'écrire, des fichiers textes, afin de charger, ou de stocker, des informations externes à l'application utilisée. Cela peut correspondre à un score, un parcours déjà fait, ou carrément des données qui seront utilisées, ou générées, par d'autres applications, Cela se fait par l'intermédiaire de fichiers dits « INI ».

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 Médiator 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**

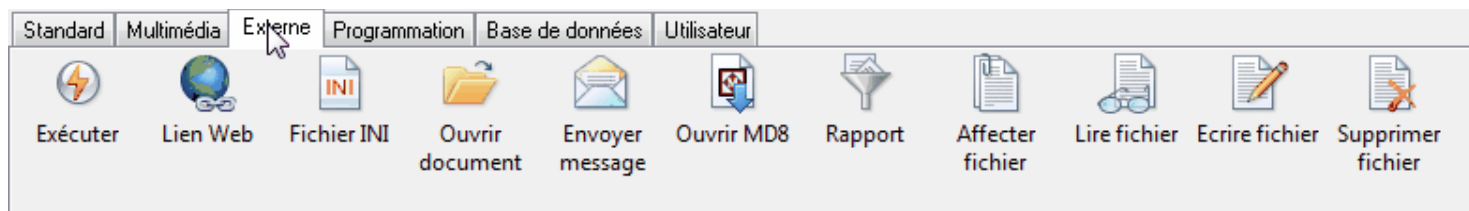
Une entrée est composée d'un nom suivi d'un signe = puis de la valeur de l'entrée.

Analyse du fichier ini suivant :

	<p>Sections déclarées : TITRES, 1 et 2</p> <p>Entrées disponibles dans la section TITRE : TitreF et NbQuestions</p> <p>Entrées disponibles dans la section 1 : QF, R1F, R2F, R3F, R4F, J et CF</p> <p>Valeur de l'entrée R1F de la section 1 : Le nombre total de pixels utilisés dans l'image</p> <p>Valeur de l'entrée R1F de la section 2 : 8 bits</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

L'action fichier ini

Disponible dans l'onglet **Externe** des actions



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

Fichiers INI

Fichier INI

Méthode
 Lecture
 Ecriture

Nom du fichier :
 @DocDir + "donnees.ini" fx

Section :
 STRING(NoQuestion) fx

Entrée :
 "CF" fx

Valeur/Variable :
 comment fx

Valeur par défaut (facultatif) :
 0 fx

Aide OK Annuler

Dans l'exemple ci-contre :

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

La variable **comment** est affectée de la **valeur** de l'**entrée** « CF » contenue dans la **section** définie par la valeur de la variable NoQuestion. Si NoQuestion vaut 1, on aura alors comment égal à « La résolution exprimée en points par pouce (ppp ou dpi) est le nombre de points par pouce que le périphérique doit créer en acquisition ou restituer en impression ou affichage. ». La fonction intégrée STRING() permet de convertir la valeur de NoQuestion, d'entier en texte.

Fichier INI

Méthode
 Lecture
 Ecriture

Nom du fichier :
 @DocDir + "reponse_" + login + ".ini" fx

Section :
 STRING(NoQuestion) fx

Entrée :
 "Reponse" fx

Valeur/Variable :
 choix fx

Valeur par défaut (facultatif) :
 "" fx

Aide OK Annuler

Dans l'exemple ci-contre :

on écrit dans un fichier placé dans le même répertoire (@DocDir) que le fichier mediator (.md8), portant comme nom « reponse_ » suivi de la valeur de login, variable définie précédemment dans l'application mediator (login = SUBSTR(nom,1,5) + SUBSTR(prenom,1,2) + @Time)

La valeur de la variable choix sera définie comme valeur de l'entrée Reponse contenue dans la section définie par la valeur de la variable NoQuestion.

Attention, si choix est une variable de type numérique, la valeur de l'entrée Reponse sera quand même de type texte !

Attention : 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.

Allons-y

Rapatriez dans votre répertoire de travail le dossier TP9_fichier_ini0

Ouvrez le fichier TP9_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.


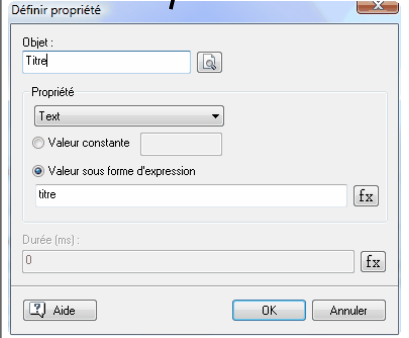
Fichiers INI

La page Page1

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**.

Fichier donnees.ini	Actions à associer à l'événement Page prête
<p>[TITRES] TitreF = Questionnaire sur l'image numérique NbQuestions = 3</p> <p>[1] QF = La résolution d'une image c'est : R1F = Le nombre total de pixels utilisés dans l'image</p>	 <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>
	 <p>La propriété Text de l'objet Titre reçoit le contenu de la variable titre.</p>

Le bouton **Commencer** :



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 événements et actions du bouton Commencer comme sur les captures d'écran ci-dessous :

Fichiers INI

	<pre> Dedans Définir pointeur , Hand Dehors Définir pointeur , Normal Clic souris Fichier INI @DocDir + "Donnees.ini", ["Titres"], "NbQuestions", NbQuestionAposer T = ("") Affecter NbQuestionAposer = INT(NbQuestionAposerT) Aller à la page question1 </pre>
vue icônes	vue script

La page Question1 (et les suivantes)

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 :

Nom de la variable	Type	Portée	Commentaires
NoQuestion	entier	globale	numéro de la question
Question	texte	globale	texte de la question
Questions_tiree	texte	globale	stockage des numéros de question déjà utilisés
Reponse1	texte	globale	texte de la première proposition
Reponse2	texte	globale	texte de la deuxième proposition
Reponse3	texte	globale	texte de la troisième proposition
Reponse4	texte	globale	texte de la quatrième proposition
ReponseCorrecteT	texte	globale	stockage du numéro de la bonne réponse
Bonnerep	entier	globale	conversion de la valeur précédente en nombre
Selection	entier	globale	réponse selectionnee par l'utilisateur
Score	entier	globale	comme son nom l'indique
Comment	texte	globale	commentaires

Fichiers INI

Commençons par les événements Page Prête

Voici la liste, longue, des actions à insérer à la suite de l'évènement Page Prête de notre page :

```

Page prête
  Affecter NoQuestion = RND(1,NbQuestionAposer)
  Boucle
    Si SUBPOS(questions_tirees, STRING (NoQuestion))>0
      Then
        Affecter NoQuestion = RND(1,NbQuestionAposer)
      Else
        Interruption
  Affecter questions_tirees = questions_tirees + "" +NoQuestion
  Fichier INI @DocDir + "donnees.ini", [STRING( NoQuestion)], "J", ReponseCorrecteT = (0) , L Rép correcte
  Affecter bonnerep = INT( ReponseCorrecteT)
  Fichier INI @docdir + "donnees.ini", [STRING (NoQuestion)], "QF", Question = ("")
  Définir propriété question.Text = Question
  Fichier INI @docdir + "donnees.ini", [STRING (NoQuestion)], "R1F", reponse1 = ("")
  Définir propriété reponse01.Text = reponse1
  Fichier INI @docdir + "donnees.ini", [STRING (NoQuestion)], "R2F", reponse2 = ("")
  Définir propriété reponse02.Text = reponse2
  Fichier INI @docdir + "donnees.ini", [STRING (NoQuestion)], "R3F", reponse3 = ("")
  Définir propriété reponse03.Text = reponse3
  Fichier INI @docdir + "donnees.ini", [STRING (NoQuestion)], "R4F", reponse4 = ("")
  Définir propriété reponse04.Text = reponse4
  
```

1. Une action Affecter va donner une valeur aléatoire à la variable NoQuestion la fonction RND() renvoie un nombre aléatoire compris entre 1 et le nombre de questions à poser. 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.
2. 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() retourne un nombre entier position du numéro dans la liste sinon il renvoie 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 :
7. 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.
8. Pour la première réponse proposée : même fichier, même section, l'entrée est R1F et la variable de stockage Reponse1
9. Idem pour les autres réponses : entrées R2F pour variable Reponse2...

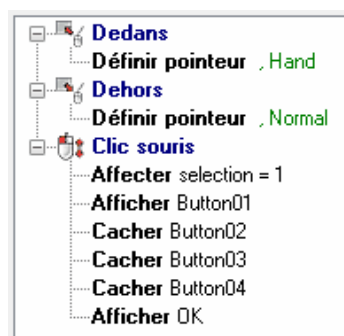
Fichiers INI

10. 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 !!!

Evénements et actions sur chaque objet Reponse

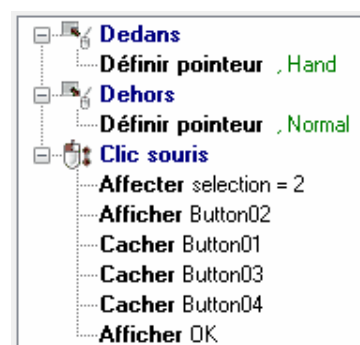
Entrée Reponse01



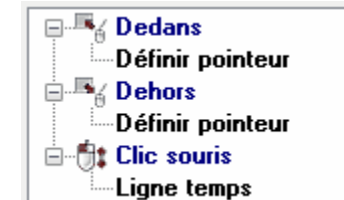
Changement de curseur en entrée et en sortie

Lorsqu'on clique sur la réponse en question, on stocke le numéro de la réponse dans la variable selection (1 pour reponse01, 2 pour 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 .

Entrée Reponse02



Le bouton OK

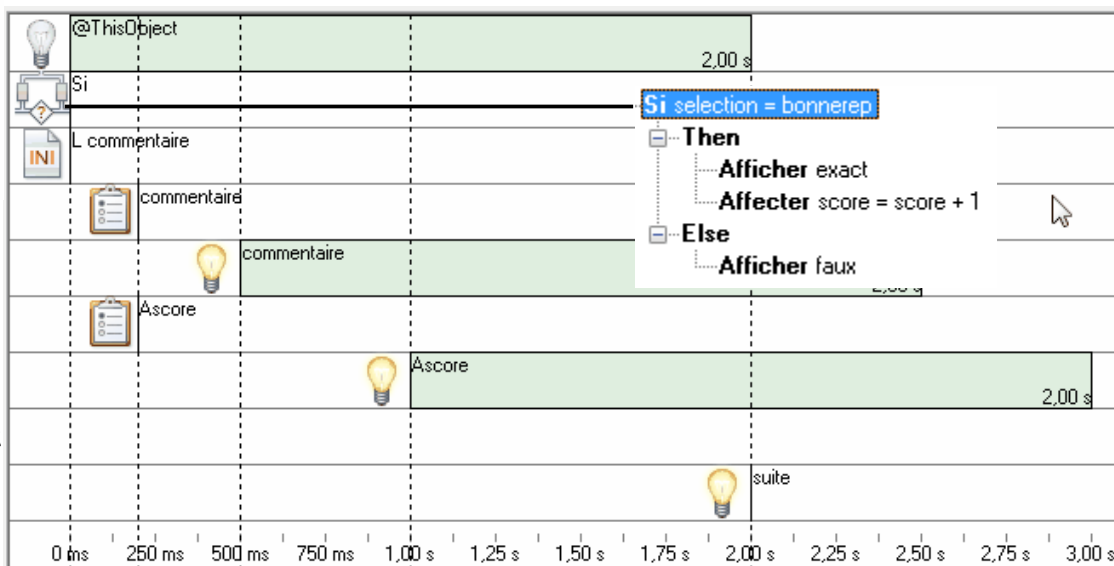


On cache le bouton OK (ici avec effet)

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



On va lire dans le fichier ini le commentaire correspondant à la question tirée (entrée CF) et on le stocke dans la variable comment.

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 "Score : " + score.

Fichiers INI

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 fichier donnees.ini .

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 TP15_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 langue 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 effectué