

PRJ1401

Compte rendu sur le fonctionnement de l'outil

Questions Moodle

Table des matières

Introduction :	2
Présentation de la fonction questchoixmutiplrepmult :	3
Présentation de la fonction questchoixmutiplrepuni:	4
Présentation de la fonction Questionappariement	5
Présentation de la fonction Questvf	5
Conclusion	6

Introduction :

Durant notre dernière séance, nous avons commencé à trouver des s

Durant les deux premières séances de notre projet, nous avons pris le temps de comprendre les outils mis à notre disposition. Notamment les macros d'Excel que nous avons commenté. Il y a 8 macros présentes, cependant plusieurs fonctions se distinguent parmi celles-ci : la fonction « Questchoixmutiplrepmult », « Questchoixmutiplrepuni », « Questvf » et « Questionappariement ».

Dans ces fonctions, certaines sous-fonctions sont présentes :

- La fonction transformestring permet de modifier des caractères par d'autre, par exemple pour pouvoir afficher « { », il faut un « \ » derrière. Ceci est fait pour d'autre caractères également,
- La fonction convertImageToBase64 permet, comme son nom l'indique de convertir une image en base 64. Pour cela, un objet de données binaire est créé :

```
Set inputStream = CreateObject("ADODB.Stream")|  
inputStream.Open
```

Ensuite, deux autres objets sont créés : un fichier XML ainsi qu'un élément « tmp » où sa valeur est définie. Puis l'image se convertit en base 64 :

```
convertImageToBase64 = "data:image/png;base64," & Replace(elem.Text, vbCrLf, "")
```

Nous voyons aussi souvent dans les fonctions des « vbCrLf », ceci signifie un retour à la ligne.

Présentation de la fonction questchoixmutiplrepmult :

La fonction questchoixmutiplrepmult est utilisé pour la feuille de calcul Questionnaire Choix Multiples.

Dans cette macro on commence par calculer le nombre de question en de plaçant sur la dernière ligne ou il y a du texte.

```
| DernLigne = Range("D" & Rows.Count).End(xlUp).Row
```

Ensuite on détermine le chemin d'accès du document pour a posteriori exporter au même endroit les documents générés.

```
chemin = Workbooks(ActiveWorkbook.Name).Path & "\"
```

Ensuite, on crée l'objet pour exporter le fichier en précisant l'encodage utf-8.

```
Dim objStream As Object  
Set objStream = CreateObject("ADODB.Stream")  
objStream.Open  
objStream.Position = 0  
objStream.Charset = "UTF-8"
```

Par la suite on entre dans une boucle for qui va se répéter pour chaque question présente dans la feuille Excel.

Puis on vérifie qu'il y a bien des éléments dans les colonnes du nom de la question et de la question (colonne A et B). Ainsi si les 2 sont vides on demande à l'utilisateur de les compléter. Mais s'il n'y a que la colonne de la question qui est vide cela ne pose pas de problème et on prend alors le nom de la question que l'on sauvegarde dans l'objStream.

```
If (IsEmpty(ActiveSheet.Cells(I, 2))) Then  
If (IsEmpty(ActiveSheet.Cells(I, 1))) Then  
    MsgBox "Il ne peux y avoir de ligne vide après la question " & nques  
    Exit Sub  
End If  
cat = "/" & transformstring(Worksheets(nomfeuille).Range("A" & I))  
indicecat = 1  
End If  
  
If (indicecat = 1) Then  
objStream.WriteText "SCATEGORY: " & cat & vbCrLf  
objStream.WriteText "" & vbCrLf  
indicecat = 0  
I = I + 1  
End If
```

Ensuite on s'intéresse au réponse et feedback entré par l'utilisateur. On commence par calculer le nombre de choix puis on calcule la note de chaque question en additionnant toutes les colonnes ou l'utilisateur a mis une bonne réponse. Puis on vérifie qu'il y a au moins une bonne

réponse par question. S'il n'y en a pas au moins une, on demande à l'utilisateur de rentrer une réponse. On détermine ensuite le coefficient de chaque question.

```
note = 0
nbrchx = Worksheets(nomfeuil).Range("A" & I).MergeArea.Rows.Count
note = Application.WorksheetFunction.Sum(Range(Cells(I, 5).Address & ":" & Cells(I + nbrchx - 1, 5).Address))

If note < 1 Then

    MsgBox "Il des questions sans réponse et ce n'est pas possible"

    Exit Sub

End If

coef = Round(100 / note, 5)
coef = Replace(coef, ",", ".")
```

Par la suite on vérifie si l'utilisateur à entrer une image dans la colonne C. S'il y a bien une image en la convertie en base 64 grâce à la fonction `convertImageToBase64`.

```
If Worksheets(nomfeuil).Range("C" & I) <> "" Then
codimage = "<br/><center><img src="" & convertImageToBase64(Worksheets(nomfeuil).Range("C" & I)) & ""/></center>"
Else: codimage = ""
End If
```

Ensuite on met dans des variables le nom de la question, la question et le feedback générale. Tout en remplaçant les caractères qui posent un problème grâce à la fonction `transformestring` défini juste avant cette macro.

Pour continuer on parcourt la colonne des réponses une à une. Ensuite on regarde si la case est différente de 1 et donc si elle est fausse car l'utilisateur met un 1 si la réponse est vraie. Alors on regarde si la case des points négatifs est cochée, ainsi on remplit le tableau des réponses en conséquence. Ensuite on vérifie si la réponse est positive et on la met dans le tableau si c'est le cas.

On regarde ensuite s'il y a des feedbacks par question et on les rentre dans le tableau correspondant.

Pour finir on écrit dans l'`objStream`, le nom de la question, la question, les réponses, les feedbacks par réponse. Et on finit par sauvegarder l'`objStream` dans le même dossier que l'outil Excel.

Présentation de la fonction `questchoixmutiplrepuni`:

La fonction `questchoixmutiplrepuni` est très similaire à la fonction `questchoixmutiplrepmult` tant dans la structure que dans le fonctionnement.

En effet le seul la seule chose qui diffère est la partie sur le fonctionnement des réponses en effet après avoir calculé la variable `note` (ligne 71) on vérifie (ligne 96) s'il n'y a pas plusieurs réponses. S'il y a plusieurs réponses, on demande à l'utilisateur de changé et de n'en mettre qu'une seule.

De plus lorsque l'on remplit le tableau des réponses (à partir de la ligne 105) on regarde juste si la réponse est vraie ou fausse, il n'y a pas de point négatif à gérer.

Présentation de la fonction Questionappariement

Comme pour les fonctions précédentes, lors de l'exécution de cette fonction, on définit la variable « DernLigne » pour voir combien de question comporte le quizz.

De plus, nous définissons l'endroit où le fichier s'enregistre ainsi que le nom du fichier. Le programme reconnait aussi combien de question va comporter le quizz :

```
For I = 2 To DernLigne                                'Faire ça

If (IsEmpty(ActiveSheet.Cells(I, 2))) Then
If (IsEmpty(ActiveSheet.Cells(I, 1))) Then
    'message à l'utilisateur
    MsgBox "Il ne peut y avoir de ligne vide après la question "
    'sortie de la procédure

Exit Sub                                            'Fin de l

End If
```

La suite du programme est ensuite plus classique : c'est l'écriture du fichier texte qui servira au site MOODLE.

Pour vérifier ensuite que le programme a bien été achevé, une boîte de message nous affiche si l'export est réussi :

```
MsgBox "export réussi"
```

Présentation de la fonction Questvf

Le schéma des fonctions se répète une quatrième fois pour le début : définition des variables puis définir l'endroit où le fichier s'enregistre. Le but principal de cette fonction est de faire un questionnaire Vrai-Faux. Pour cela, nous pouvons insérer des images ainsi que noter le nom des questions. Ce qui diffère des autres fonctions est qu'il n'y a que deux choix de réponses possibles.

Conclusion

Lors de ces deux séances, nous avons donc appris un nouveau langage qui est le VBA. Celui-ci nous a permis de comprendre les fonctions utilisées pour créer des questions pour le site MOODLE.

Notre prochain objectif pour les semaines à venir est de chercher et appliquer, si possible, des moyens pour appliquer l'extraction des questions sur Excel macOS de cette manière, l'utilisateur va pouvoir utiliser l'outil de son choix.