Projet Graphique et base GIT

Fred,

**JE RAJOUTE CETTE LIGNE POUR VOIR SI LE CHANGEMENT EST REPERE PAR GIT**

## Je fais une nouvelle modif après avoir créer une nouvelle branche DéveloppementfichierWordPPO

Merci de m’avoir aider à créer le projet Graphique et la base Git pour l’OTS#1 de Rabigh2.

Dans la foulée, nous allons créer le projet et traduire les graphiques pour l’OTS#2.

Par rapport au document ‘’Instructor Graphic Centum VP integration’’

La création du CENTUM VP project ne pose pas de problème si elle est créée sur le disque dur local. (de préférence pas sur le serveur ; de toute façon le tout ira ensuite sur Git, accessible à tous).Nous avons créer le répertoir R410 sur mon D :

Par contre la connexion avec le viewer pose problème : le viewer e connecte à IndissPlus mais pas au simulateur que l’on ouvre. Le viewer crash.

C’est lié au fichier Dev Bundle

Ne pas prendre le fichier CentumVP Emulation\_setup.exe dans [\\int.rsi-france.com\data\Setups\IndissPlus\Finales\2.2.1.1000](file:///\\int.rsi-france.com\data\Setups\IndissPlus\Finales\2.2.1.1000) (Latest)\Graphic Development\Emulation Libraries mais le fichier que tu as mis à disposition dans ‘’To Pascal’’ : CentumVP\_DevBundle\_setup\_15102015\_161616.exe

Malgré tout, lorsque l’on fait run viewer il y a 2 messages d’erreurs qui nécessitent d’enlever 2 paragraphes dans le fichiers xaml

Note ce problème est gênant uniquement dans l’étape de traduction des fichiers graphiques. HRS ne le rencontrera pas dans l’implémenattion des couches FOD

Avant la traduction, nous avons importé créé un suos-répertoire OriginalGraphicViews dans le répertoire R410 dans lequel nous avosn mis tous les fichiers graphiques originaux.

Pour traduire les fichiers une fois que le viewer est ouvert c’est facile : on prend le répertoire ou sont logés tous les répertoires de fichiers à traduire (Note : pour ne pas surcharger les machines et Git on peut supprimer de ces répertoires tous les fichiers avec une extension autre que xaml)

Les fichiers ont été traduits avec le viewer 2.3

Pour la première création il faut cocher la case ‘’generate Xaml layers’’. Il faudra la décocher si un jour on veut traduire de nouveau fichiers.

Nous avons installé le programme NSIS qui sert à faire des setups (on livrera au client un seul setup qui contiendra le viewer, le devbundle, les graphiques… si j’ai bien compris)

Si HRS n’arrive pas à télécharger le bon visual studio. On peut leur mettre à dispo par hightail l’ensemble du répertoire [\\srv-data\IndissPlus Dev Tools\Visual Studio 2013](file:///\\srv-data\IndissPlus%20Dev%20Tools\Visual%20Studio%202013)

HRS doit installer Github for windows à partir de <https://desktop.github.com/>

HRS doit nous fournir leurs identifiants pour pouvoir avoir accès à la base git remote et faire un clone

Si HRS a Visual Studio Professionnal ils peuvent l’utiliser sinon ‘’Visual Studio Community’’

Privilégier Visual Studio 2013 au lieu de 2015. 2013 est la version utilisée à RSI

Nous avons téléchargé et installé SMARTGIT

Fred, Ludovic et Cyril sont les administrateurs de Git pour RSI. Il faut leur demander de créer la base en remote avec accès privé. Nous l’avons appelé Rabigh2RSI410.

Nous avons ensuite cloner cette base sur mon D :. Un répertoire Rabigh2RSI410 s’est créé sur mon D : avec un sous répertoire .git

Dans ce répertoire D : Rabigh2RSI410 et donc au même niveau que le sous-répertoire .git, nous avons copié tours les fichiers et répertoires qui étaient dans le répertoire R410 que nous avions cééer à partir de Blend.

Fred rajoute un fichier exemple pour le csv du layer generator. HRS peut d’ores et déjà travailler à la construction de ce fichier.

Le fichier crossreferenceTag du répertoire configuration files n’a rien à voir avec le layer générator. C’est le fichier de correspondance des TAG de notre modèle et des Tag du graphiques

La version IndissPlus2.3 devrait sortir en début de semaine.

1 – JLE : faire le tri des variables qu’il est nécessaire d’afficher et dont il est donc nécessaire de reconfiguer les tags. Note : si on doit cacher les variables que l’on n’émule pas sur la station instructeur alors il vaut mieux émuler toutes les variables

2 – Attendre la sortie officielle de la 2.3

3 – informer HRS de la mise à dispo sous github

4 – Donner les spec à respecter pour l’implémentation du FOD / BL / FAILURE

5 – Demander à HRS de préparer les 2 fichiers Csv : modelTag/GraphicTag et FOD FAILUR BL/Graphic Layer

6 – HRS génère les lignes FOD et place correctement les objest FOD sur les graphiques

1- Générer le projet Graphique sous Blend avec viewer 2.3

2- Dans le projet, créer un nouveau répertoire avec les original graphic files ne contenant que des répertoires avec 1 seul fichier xaml

3-Penser à exécuter le devbundle

4- Run viewer

5- Faire la traduction : à cette étape on aura 2 messages d’erreur à voir avec Fred pour supprimer ce qu’il faut dans le bon fichier xaml.

6- Faire créer par Fred un nouveau répertoire Git Rabigh2\_R420

7- Clôner ce repertoire en local

8- Copier le projet graphique dans le répertoireRabigh2\_R420. Y inclure le viewer, le devbundle, l’exemple de fichier csv

9- Faire commit and push

10- Le projet est disponible pour HKD

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Pascal Pouchot-Camoz**

Project Manager

*Office:*    +33 (0) 4 76 52 52 78

*Mobile:*  +33 (0) 6 74 67 48 91



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_