

3 Fonctionnalités Phase II (Semaine du 30/03/2020)

Il s'agit de développer une **application WPF** permettant de gérer des objets cartographiques affichés sur une carte. Ces objets sont des points d'intérêt modélisés par la classe POI, des trajets modélisés par la classe Polyline et des surfaces modélisées par la classe Polygon

Développez votre application en fonction de vos connaissances. Le document ci-dessous ne donne pas la liste des choses à faire dans l'ordre où elles doivent l'être, mais donne plutôt une liste des caractéristiques des classes et de l'application finale. Lisez donc bien la totalité de l'énoncé avant de commencer votre nouvelle application. Travaillez étape par étape et testez au fur et à mesure.

3.1 Une nouvelle classe

Il s'agit de créer une **classe MyPersonalMapData** complète respectant le canevas déjà utilisé dans le précédent dossier et permettant de créer et de gérer l'ensemble des objets cartographiques manipulés par l'application graphique.

Les **caractéristiques** de l'objet MyPersonalMapData sont :

- **Nom, Prénom**
- **Email**
- **ObservableCollection** générique d'objets **implémentant une nouvelle interface ICartoObj**. Seuls les POI, Polyline et Polygon implémentent cette interface.

La classe MyPersonalMapData étant directement liée aux objets qu'elle contient, elle est ajoutée au projet « Class Library (.Net Framework) » **MyCartographyObjects**. *co m'pourrait logique*

Cette classe propose

- Une propriété de type « **get** » qui **retourne la collection**,
- Des méthodes qui permettent de **réinitialiser** la collection, la **charger** ou **l'enregistrer** au format **binaire**. Le nom du fichier est donné par la concaténation du prénom et du nom. Il n'est pas demandé de gérer les mots de passe ni les doublons possibles dans les chaînes « prénom + nom ».

Celle-ci doit être testée grâce à un projet console de test avant de pouvoir être utilisée dans l'interface graphique.

3.2 Application WPF

Il s'agit de créer une application WPF « PersonalMap Manager » permettant de créer et gérer des objets cartographiques liés à un utilisateur et de les afficher sur une carte

Pour cela, il faut créer un nouveau projet C# WPF (.Net Framework) qui utilise la librairie de classe créée dans la phase I. Ce nouveau projet est ajouté à la solution de cette Phase I.

La première fenêtre est une fenêtre de connexion dans laquelle l'utilisateur encode son prénom, son nom et son adresse email. Si un fichier à son nom existe, il est ouvert et les données sont chargées. Sinon, les données encodées peuvent servir à un futur enregistrement.

La fenêtre principale est agencée de façon optimale en utilisant au mieux la notion de Layout. Nous attendons une utilisation pertinente de ceux-ci afin d'avoir une interface qui reste utilisable et ergonomique si la taille de la fenêtre est modifiée. Elle doit contenir au minimum un GridSplitter et un StackPanel. Dans celle-ci, on y trouve :

1. Un menu contenant au moins les éléments suivants :
 - a. File → Open
 - b. File → Save
 - c. File → POI → Import
 - d. File → POI → Export
 - e. File → Trajet → Import
 - f. File → Trajet → Export
 - g. File → Exit
 - h. Tools → Option
 - i. Tools → About box : Créer une « About box » comme il en existe dans de nombreux programmes (Auteur, copyright ☺, date).
2. Plusieurs zones destinées à :
 - a. Encoder / modifier les caractéristiques (donc les propriétés) des objets cartographiques (Label, Textbox, TextBlock, ComboBox, ...),
 - b. Visualiser dans une ListBox, les objets cartographiques chargés / ajoutés. Dans la ListBox, on affiche uniquement la description du POI, du trajet ou de la surface,
 - c. Visualiser ces mêmes objets sur une carte BingMap voir §3.5.
3. Une Toolbar est ajoutée juste en dessous du menu. Elle contient des boutons permettant d'activer les différentes actions de création, modification, suppression, sur les objets POI, Trajets et Surface. Choisissez une organisation des boutons dans la Toolbar pour que votre utilisateur comprenne facilement comment il doit s'en servir.
4. Finalement, il doit être possible de :
 - a. Créer un « point d'intérêt » par encodage de données ou par clic sur la carte,
 - b. Créer un « trajet » par clic sur la carte. Une liste de Coordonnee peut contenir les POI récupérés par clic,
 - c. Créer une « surface » par clic (BONUS),
 - d. Sélectionner dans la ListBox ou par Clic puis
 - i. visualiser les détails,
 - ii. supprimer,
 - iii. modifier (couleur, description, ...).

La fenêtre « Option » est une boîte de dialogue secondaire non modale (non bloquante) permettant de :

1. Initialiser le « dossier de travail »,
2. Choisir et modifier la couleur utilisée pour afficher la ListBox contenant les objets visualisés sur la carte (couleur du texte et couleur du fond).

Elle fonctionne de la même manière que la fenêtre qui permet de gérer la résolution de l'écran sous Windows, elle contient donc 3 boutons :

- OK : valide le choix et ferme la fenêtre,
- Cancel : ferme la fenêtre sans valider le dernier choix (qui n'a pas encore été appliqué),
- Appliquer : valide le choix, ne ferme pas la fenêtre.

La mise à jour de l'interface utilisateur principale se fera par la création d'un **événement personnalisé** envoyé par cette fenêtre à la fenêtre principale au moment opportun.

3.3 Informations complémentaires

- Le « dossier de travail » est un répertoire choisi par l'utilisateur de l'application dans lequel les fichiers de données sont enregistrés. Il est initialisé dans la fenêtre secondaire « Options ». A ce stade du projet, il n'est pas demandé de sauvegarder le dossier de travail mais seulement de le conserver en mémoire pour le cas où l'utilisateur demanderait à ouvrir ou enregistrer un fichier.
- La sauvegarde (File→Save) et le chargement (File→Open) doivent permettre de sauvegarder l'objet MyPersonalMapData courant dans un fichier binaire (Save) ou de charger un fichier binaire contenant des données précédemment sauvegardées (Open).
- Un POI ou un trajet au format texte peut être importé ou exporté de ou vers un fichier portant l'extension « .csv ». Il est possible d'en créer à partir d'un fichier Excel (choisir CSV avec le ; comme séparateur). Si tout le monde a le même format, il doit être possible d'importer les POI ou les trajets d'un autre étudiant. Les fichiers d'exemple (voir §0) se trouvent sur la plateforme Moodle du cours.
- Pensez à la modularité du code et mettez en évidence vos compétences de conception. Permettez de rendre ce code réutilisable si cela vous semble légitime. Aussi, veillez à ne pas réinventer la roue, servez-vous un maximum des fonctionnalités déjà existantes.

3.4 Exemples de fichier à importer / exporter

3.4.1 Point of Interest

Soit le fichier « HEPL Seraing POI.csv »

	A	B	C	D
1	50,610597	5,509206	HEPL Seraing	
2				
3				

La colonne A contient la latitude, la colonne B contient la longitude et la colonne C, la description.

3.4.2 Trajet

Soit le fichier « HEPL Seraing Liege Trajet.csv »

	A	B	C	D
1	50,610597	5,509206	HEPL Seraing	
2	50,610013	5,514195		
3	50,605601	5,543163		
4	50,611906	5,584131		
5	50,619873	5,581654	HEPL Liege	
6				

Plusieurs lignes du fichier constituent l'ensemble d'un trajet. Chaque point de passage peut être un **POI** (cas du premier et du dernier point) ou une **Coordonnee**.

3.5 Carte Bing

Il s'agit ici d'utiliser un composant existant permettant de visualiser une carte et de déposer dessus des composants cartographiques tels que des points d'intérêt, des lignes ou des surfaces. La documentation permettant d'utiliser le composant se trouve ici :

[https://docs.microsoft.com/en-us/previous-versions/bing/wpf-control/hh750210\(v=msdn.10\)](https://docs.microsoft.com/en-us/previous-versions/bing/wpf-control/hh750210(v=msdn.10)) → lire les différentes sections.

1. Créer un compte développeur pour pouvoir utiliser / afficher les cartes dans une application que vous développez. <https://www.bingmapsportal.com/Account>
Créer une nouvelle clé (donner le nom de votre application, choisir le type « Basic » et Application type = Dev/Test)
Récupérer la clé de votre application,
2. Télécharger le SDK (<https://www.microsoft.com/en-us/download/details.aspx?id=27165>) puis ajouter une référence vers l'assembly Microsoft.Maps.MapControl.WPF pour pouvoir ajouter les composants dans votre projet WPF. Voir rubrique « Getting Started » [https://docs.microsoft.com/en-us/previous-versions/bing/wpf-control/hh745791\(v=msdn.10\)](https://docs.microsoft.com/en-us/previous-versions/bing/wpf-control/hh745791(v=msdn.10)) puis « Creating an Application ». [https://docs.microsoft.com/en-us/previous-versions/bing/wpf-control/hh830433\(v=msdn.10\)](https://docs.microsoft.com/en-us/previous-versions/bing/wpf-control/hh830433(v=msdn.10)),

3. Ajouter le composant graphique à votre projet WPF, voir exemple [https://docs.microsoft.com/en-us/previous-versions/bing/wpf-control/hh709042\(v=msdn.10\)](https://docs.microsoft.com/en-us/previous-versions/bing/wpf-control/hh709042(v=msdn.10)) en insérant la clé créée au §1 ci-dessus,
4. Consulter les exemples de code des § qui suivent pour changer l'échelle de la carte, la centrer sur des coordonnées GPS données (<https://docs.microsoft.com/en-us/previous-versions/bing/wpf-control/hh830433%28v%3dmsdn.10%29>),
5. L'objet Pushpin sera utilisé pour visualiser un POI,
6. L'objet MapPolyline sera utilisé pour visualiser une Polyline,
7. L'objet MapPolygon sera utilisé pour visualiser un Polygon.
8. L'objet Location contient des coordonnées GPS.
9. Les événements sur la carte seront utilisés en fonction des cas pour sélectionner un objet existant sur celle-ci pour créer des objets cartographiques.

Volontairement, nous avons voulu créer un premier ensemble de classes (MyCartographyObject) complètement indépendant des classes proposées par la nouvelle librairie ajoutée. Nous voulons que la carte Bing ne soit utilisée que pour l'affichage et l'interaction avec la carte via des clics sur base de la logique de commande que vous mettrez en place. Il s'agit donc de maintenir en mémoire la liste d'objets de la première librairie ainsi que la liste des objets utiles à la visualisation sur la carte Bing.

3.6 Evaluation

L'évaluation portera sur :

1. La qualité de la solution (découpage en projets)
2. Respect des règles de nommage des composants graphiques, des variables membres, propriétés et méthodes
3. Respect de l'énoncé et des fonctionnalités demandées
4. La gestion de la liste des objets en mémoire et la mise à jour automatique des données lors de modifications
5. L'optimisation de l'utilisation des technologies

4 Fonctionnalités Phase III (juin 2020)

En cours de rédaction