IN'TECH INFO
ECOLE SUPERIEURE D'INGENIERIE INFORMATIQUE

ITI.Map2D

Ce projet Windows Forms met en œuvre une navigation graphique sur une carte (classe Map) à 2 dimensions constituée de carrées (Box).

Nous l'avons réalisé parce que ce n'est pas complètement trivial de gérer un affichage de ce type qui gère correctement le zoom. Le principe est que la Map est en centimètres (unité arbitrairement choisies) et que les Box se dessinent toujours dans le même repère (de 0,0 à leur largeur et hauteur), sans se préoccuper de leur position exacte ni du facteur de zoom : tout cela est géré par la Map, le ViewPort et le Contrôle graphique.

Cela montre l'utilisation du Graphics (qui permet de dessiner à l'écran) avec les transformations (qui sont des matrices 3x3) que tous les frameworks graphiques utilisent pour représenter les transformations du plan.

Allez donc voir sur Code Project : http://www.codeproject.com/Articles/8281/Matrix-Transformation-of-Images-using-NET-GDIplus, il est très bien fait et le programme associé permet de manipuler et de bien comprendre le principe des matrices de transformation.

L'implémentation actuelle et la conception de ITI.Map2D sont basiques : c'est une base de départ dont l'objectif est d'être la plus simple possible et sur laquelle beaucoup d'aspects supplémentaires peuvent être développées.

Ci-dessous quelques idées (qui ne sont que des pistes) :

- Gérer le click et la souris sur le Control.
  En cliquant sur la Map graphique, il faut calculer le point correspondant dans la Map. Une méthode virtuelle OnClick(int x, int y) sur la Box peut alors être appelée où x et y sont dans le référentiel de la Map (donc en centimètres).

- Faire en sorte que les Box soient d'un type paramétré (Map<T> where T : Box).
  Cela permettra de réutiliser la logique de la Map avec une classe Box spécifique (une autre solution est que Map travaille avec une interface IBox et non avec une classe Box de base). Dans les deux cas, la Map doit être capable de créer les Box spécialisées (ou les implémentations de IBox) : une petite Func<int,int,IBox> fournie au constructeur de Map lui permettra de créer la box à une position (les deux int en paramètres).

- Gérer le déplacement de la carte (agrippé par la souris comme dans les viewers de pdf). Cela nécessite de redéfinir (au moins) les méthodes `OnMouseDown` et `OnMouseUp`.

- Faire un Control de mini-map qui visualise le viewport sous la forme d'un simple rectangle sur la carte globale).

- Implémenter une carte dynamique (les Box sont créées on-demand, l'espace est alors potentiellement très grand).
  On ne peut plus utiliser un tableau bi-dimensionnel dans ce cas pour tenir l'ensemble des Box, il faut une autre structure de données.
  Tous les calculs par rapport au zoom sont à changer : il n'y a plus de zoom « qui visualise tout » : il faut nécessairement définir un zoom maximal (par exemple 10x10 box maximum).

- Découpler la méthode Box.Draw de la Box, c'est-à-dire : faire en sorte que la méthode Draw soit portée par un autre objet (typiquement un « BoxRenderer »). Cela permet de n'avoir dans la Map et les Box que le « modèle », les « objets métiers » (des animaux, des meubles, des murs, ou la description de la surface par exemple), et de sortir complètement (dans une autre dll), la façon dont on les affiche à l'écran.

- Prendre en compte le Clipping du Graphics (voir http://msdn.microsoft.com/en-us/library/155t36zz%28v=vs.110%29.aspx) qui permet d'optimiser l'affichage (ne dessiner que ce qui est réellement nécessaire) en exploitant la région qui est disponible sur le Graphics (Graphics.Clip).

- Etc.