

Contenu de la Page

- Sélection d'un Territoire sur une carte
- 1 : Récupération des Territoires dans un fichier xml
- 2 : Sélection d'une zone rectangulaire
- 3 : Affichage du nom du territoire sélectionné
- 4 : Sélection par quadrillage
- 5 : Sélection par couleur
- Modification d'une image
- 6 : Ajout de cercle sur la carte
- Sélection sur une carte avec des milliers de couleurs
- 7 : Identification du Territoire
- 8 : Subsidiare : Les prémisses d'un jeu

Nous allons dans ce TP, adapter des programmes à des interfaces non standards.

Couramment dans une IHM, nous avons des boutons, des zones de texte, des cases à cocher...

Que des contrôles prévus, par notre langage.

Mais il peut arriver que l'on ait besoin d'une interface très spécifique. Les cartes géographiques sont un terrain de jeu fabuleux.

Sélection d'un Territoire sur une carte

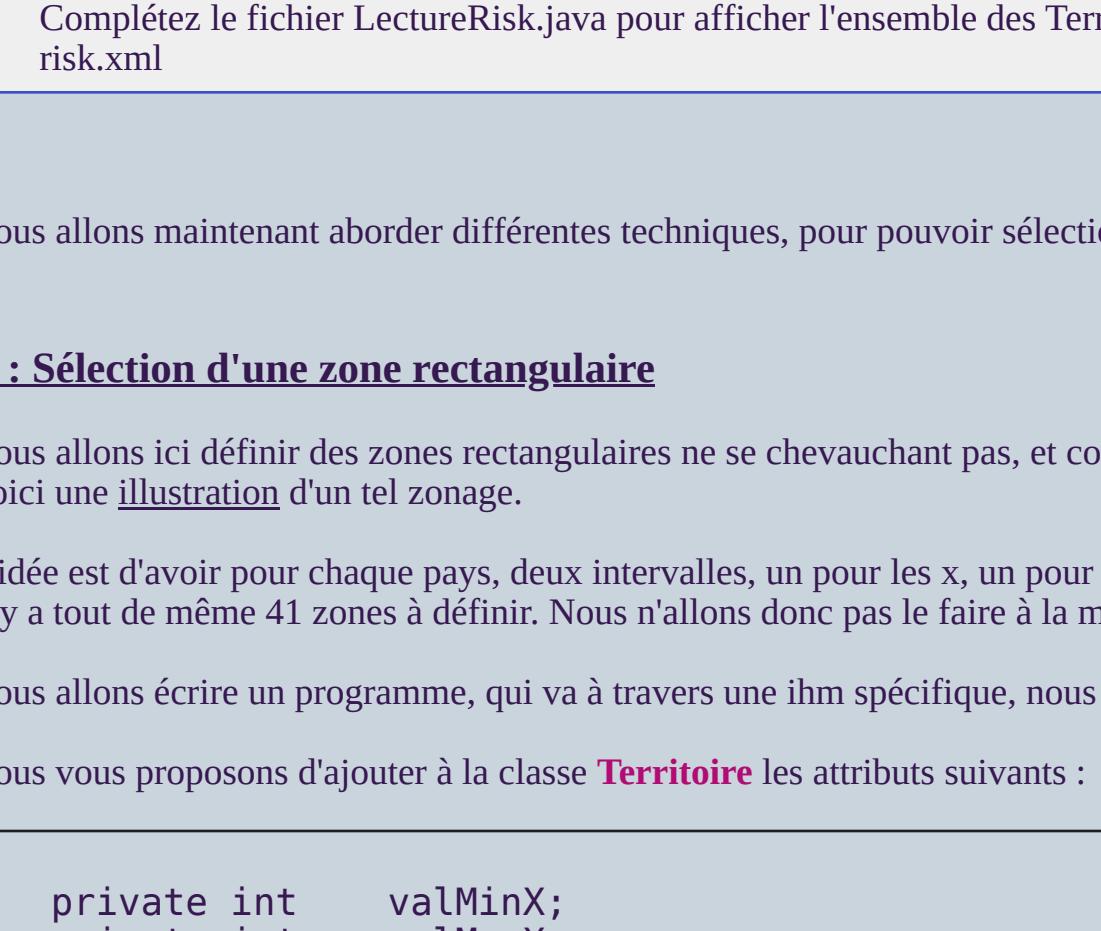
1 : Récupération des Territoire dans un fichier xml

Un Territoire est une zone ne répondant à aucune forme géométrique régulière.

Des Territoires s'imbriquent les uns dans les autres.

Donc lorsque l'on clique sur un Territoire, cela va être très difficile de l'identifier de façon précise à partir des coordonnées de la souris.

Nous vous proposons de partir de la carte utilisée dans le jeu Risk.



carte_risk.gif

Vous trouverez dans le fichier `risk.xml`, la liste de tous les territoires du risk.

Voici le positionnement sur une carte de l'ensemble de ces Territoires, dans l'ordre du fichier XML.

Nous vous invitons à consulter l'icône complémentaire XML en Java.

Pour l'utilisation de Java, nous vous conseillons d'utiliser la version 2 JDOM2: `jdom-2.0.6.jar`

Tous faudra mettre dans votre répertoire de paquage (`$HOME/TP/algo/paquetage_class`) et ajouter ce .jar à votre classpath `/home/TP/java/paquetage_class/jdom-2.0.6.jar`

À adapter en fonction de votre arborescence.

Travail à Faire

- Dans un répertoire **exercice 1**, placez les ressources suivantes :

`LectureRisk.java`

`Territoire.java`

`risk.xml`

Complétez le fichier `LectureRisk.java` pour afficher l'ensemble des Territoires contenus dans le fichier `risk.xml`

Nous allons maintenant aborder différentes techniques, pour pouvoir sélectionner un Territoire.

2 : Sélection d'une zone rectangulaire

Nous allons ici définir des zones rectangulaires ne se chevauchant pas, et couvrant au maximum un Territoire donné.

Voici une illustration d'un tel zonage.

L'idée est d'avoir pour chaque pays, deux intervalles, un pour les x, un pour les y.

Il y a tout de même 41 zones à définir. Nous n'allons donc pas le faire à la main à travers un logiciel de dessin.

Nous allons écrire un programme, qui va à travers une ihm spécifique, nous aider à définir la zone rectangulaire de chaque Territoire.

Nous vous proposons d'ajouter la classe `Territoire` les attributs suivants :

```
private int valMinX;
private int valMaxX;
private int valMinY;
private int valMaxY;
```

Ainsi que les méthodes suivantes :

```
public void affecterIntervalles ( int valMinX, int valMaxX, int valMinY, int valMaxY )
{
    /* A compléter */
}

// Cette méthode détermine si le pixel de coordonnées x, y appartient ou non à la zone du Territoire courant.
public boolean contient ( int x, int y )
{
    /* A compléter */
}

// Retourne la valeur x du centre du Territoire
public int getCentreX()
{
    /* A compléter */
}

// Retourne la valeur y du centre du Territoire
public int getCentreY()
{
    /* A compléter */
}
```

Nous vous proposons maintenant de récupérer les classes suivantes :

`DefinirZone.java`

`FrameDefinirZone.java`

`PanelDefinirZone.java`

`VerifSerialisation.java`

Récupérez également l'image :

`carte_zone_risk.gif`

Travail à Faire

- Dans un répertoire **exercice 2**,

Recopier le contenu du répertoire exercice1,

Compléter le code de la classe `Territoire` Compléter le code de `DefinirZone` et `PanelDefinirZone`.

Conseil : Mettez au point votre programme en récupérant les données que de 5 Territoires

- Placer le fichier `VerifSerialisation` dans le répertoire exercice2

Modifier son code.

Il faudra également modifier le code de la méthode `toString` de la classe `Territoire`, pour afficher comme dans la trace d'exécution ci-dessous, les coordonnées [x,y] des deux coins (c1,c2) du rectangle délimitant le Territoire.

Vous avez effectué une sauvegarde à l'aide d'une serialisation, vous allez maintenant vérifier que cette serialisation a correctement fonctionnée.

Voici le début du résultat attendu :

American du Nord Alstek	c1 [27;113]	c2 [69;126]
American du Nord Territoire du Nord-Ouest	c1 [114;195]	c2 [67;126]
American du Nord Groenland	c1 [197;399]	c2 [27;126]
American du Nord Alberta	c1 [102;163]	c2 [127;165]
American du Nord Ontario	c1 [154;224]	c2 [129;169]
American du Nord Québec	c1 [223;200]	c2 [124;178]
American du Nord Etat de l'ouest	c1 [64;151]	c2 [167;226]
American du Nord Etat de l'est	c1 [154;261]	c2 [179;242]
American du Nord Amérique Centrale	c1 [65;161]	c2 [259;188]
American du Sud Vénézuela	c1 [180;263]	c2 [136;368]
American du sud Pérou	c1 [163;298]	c2 [371;452]

Pour ceux qui n'auraient pas réussi à capturer les clics, voici les instructions permettant de renseigner les coordonnées de tous les territoires : `clickauto.txt`

3 : Affichage du nom du territoire sélectionné

Maintenant que nous avons un fichier comportant les différentes zones de chaque Territoire, nous allons écrire un programme qui affichera le nom du territoire lorsque nous cliquerons sur la carte (cette fois-ci sans le quadrillage).

Vous remarquerez également qu'à partir de cet exercice la Frame et le Panel ne posséderont plus une référence de la classe principale, mais une référence d'une interface `Controlable`.

Travail à Faire

- Dans un répertoire **exercice 3**,

Récupérez le fichier `territoriaire.xml` de l'exercice précédent.

et les 4 fichiers : `SelectionZone.java` `FrameCarte.java` `PanelCarte.java` `Controlable.java`

Complétez l'ensemble des sources Java pour répondre à ce besoin.

4 : Sélection par quadrillage

La sélection par zone rectangulaire est loin d'être précise.

La sélection du Pérou en est une parfaite illustration.

Travail à Faire

- Dans un répertoire **exercice 4**,

Récupérez le fichier `territoriaire.xml` de l'exercice précédent.

et les 4 fichiers : `SelectionZone.java` `FrameCarte.java` `PanelCarte.java` `Controlable.java`

Complétez l'ensemble des sources Java pour répondre à ce besoin.

5 : Sélection par couleur

La méthode du quadrillage est certes plus précise, mais la constitution du fichier est fastidieuse, même à l'aide d'un programme de saisie.

Cette méthode sera encore plus fastidieuse, si nous souhaitons augmenter la précision, en diminuant la taille de chaque case.

Nous ne souhaitions pas que le rendu de l'image soit déformé pour l'utilisateur.

Ainsi devons-nous tous les pays de :

- l'Amérique du Nord en jaune

- l'Amérique du sud en rouge

- l'Europe en bleu

- l'Afrique en Marron

- l'Asie en vert

- l'Océanie en Mauve

Comme vous le savez l'obtention d'une couleur se fait par synthèse additive des trois couleurs primaires rouge vert et bleu.

Pour chacune des trois couleurs primaires, la valeur s'exprime dans un intervalle entre 0 et 255 (FF en hexadécimal).

Un écart de 5 unités dans une des valeurs est imprévisible.

Il nous suffit de déterminer pour chaque territoire la couleur dominante et de l'afficher.

Vous n'avez pas à faire de programme pour cela, mais vous allez écrire un programme :

- qui placera l'ensemble des couleurs dans une chaîne de caractères.

- cette chaîne sera placée dans le presse-papier.

- chaque couleur sera stockée sous sa forme décimale (forme décimal R + G * 256 + B * 256*256)

- chaque entier sera séparé par un \n

L'association du code couleur (une clé), à un Territoire(valeur), devrait vous faire penser à une structure particulière.

A partir des données contenues dans le presse-papier et des fichiers précédemment manipulés, vous allez adapter votre programme de la question précédente :

- pour construire en dur la structure,

- afficher le nom du Territoire, en fonction de la couleur du pixel, sur lequel l'utilisateur a cliqué.

Travail à Faire

- Dans un répertoire **exercice 5**,

Récupérez le fichier `territoriaire.xml` de l'exercice précédent.

et les 4 fichiers : `SelectionZone.java` `FrameCarte.java` `PanelCarte.java` `Controlable.java`

Complétez l'ensemble des sources Java pour répondre à ce besoin.

Modification d'une image

6 : Ajout de cercle sur la carte

Nous vous proposons dans cette dernière partie de créer une nouvelle version du programme de la question 5.2.

Le centre sera déterminé à partir des zones définies à la question 2. L'image alors modifiée, devra être sauvegardée.

Voici une illustration du résultat attendu.

Nous vous proposons de consulter le : [Tutorial Oracle](#)

Travail à Faire

- Dans un répertoire **exercice 6**,

Récupérez le fichier `RiskCircle.java`

et les fichiers `carte_variation_risk.gif` et `PanelCarte.java`

Complétez l'ensemble des sources Java pour répondre à ce besoin.

7 : Sélection par couleur

La méthode du quadrillage est certes plus précise, mais la constitution du fichier est fastidieuse, même à l'aide d'un programme de saisie.

Cette méthode sera encore plus fastidieuse, si nous souhaitons augmenter la précision, en diminuant la taille de chaque case.

Nous ne souhaitions pas que le rendu de l'image soit déformé pour l'utilisateur.

Ainsi devons-nous tous les pays de :

- l'Amérique du Nord en jaune

- l'Amérique du sud en rouge

- l'Europe en bleu

- l'Afrique en Marron

- l'Asie en vert

- l'Océanie en Mauve

Comme vous le savez l'obtention d'une couleur se fait par synthèse additive des trois couleurs primaires rouge vert et bleu.

Pour chacune des trois couleurs primaires, la valeur s'exprime dans un intervalle entre 0 et 255 (FF en hexadécimal).

Un écart de 5 unités dans une des valeurs est imprévisible.

Il nous suffit de déterminer pour chaque territoire la couleur dominante et de l'afficher.

Vous n'avez pas à faire de programme pour cela, mais vous allez écrire un programme :

- qui placera l'ensemble des couleurs dans une chaîne de caractères.

- cette chaî