

Projet de JAVA : Gestionnaire de Stroke

Bertrand Le Saux

1 Présentation

Le but de ce projet est d'implémenter un gestionnaire de stroke ¹. Un stroke est une alternative, dans certains cas plus rapide, aux boutons ou aux menus. L'utilisateur dessine un symbole avec la souris, et le logiciel effectue une action correspondante. Par exemple, le dessin d'un point d'interrogation va déclencher l'ouverture de la fenêtre d'aide, de la même manière que le bouton "aide".

2 Les différentes parties du logiciel

2.1 Interface graphique

Il faut tout d'abord créer une interface graphique qui permette de dessiner un stroke avec la souris. Un modèle rudimentaire est présenté sur la figure 1.

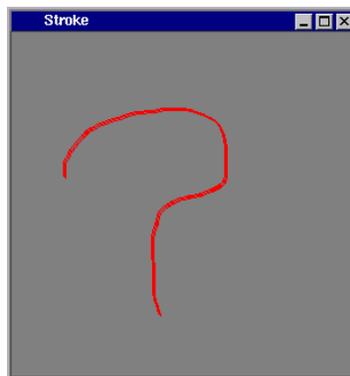


FIG. 1 – Fenêtre permettant le dessin de "strokes".

2.2 Reconnaissance des symboles

A partir du symbole dessiné par l'utilisateur, le système doit identifier une action à effectuer. L'approche utilisée est la suivante : on considère le carré dans lequel est inscrit le

¹La société Sensiva ® a développé un logiciel utilisable sous les différents types usuels de plateformes pour PC (y compris Linux) et gratuit (cf. <http://www.sensiva.com>).

stroke, et on le découpe en $n * n$ cases, comme indiqué sur la figure 2 pour $n = 3$. n représente le nombre de cases par côté. Votre logiciel devra permettre de changer la résolution de reconnaissance en jouant sur le paramètre n , $n = 3$ constituant une valeur par défaut (que nous emploieront dans la suite du sujet). À chaque stroke, on associe la suite de cases par lesquelles passe le trait. Ainsi, le stroke *point d'interrogation* correspond au code 4123658, et le stroke *diagonale* au code 159.

L'ordre dans lequel les cases ont été traversées est important : le stroke 159 (couramment utilisé pour faire un agrandissement) est différent du stroke 951 (qui correspond à une réduction).

La reconnaissance ne doit pas dépendre de la taille du symbole dessiné : elle doit se centrer par rapport au dessin, et s'adapter à sa taille.

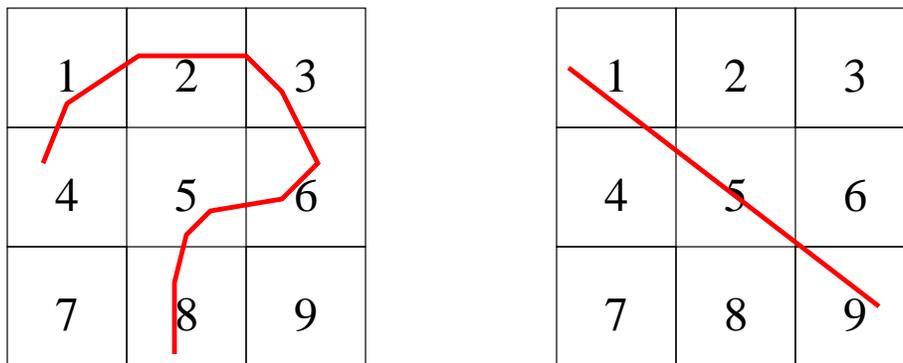


FIG. 2 – Exemples de “strokes”, avec leurs codes associés.

2.3 Déclenchement d'une action

Voici des exemples d'actions qui peuvent être déclenchées par un stroke :

- 4123658 ; le point d'interrogation déclenche l'ouverture d'une fenêtre d'aide, qui contient les principaux codes.
- 159 ; la diagonale descendante de gauche à droite agrandit la fenêtre.
- 951 ; la diagonale montante de droite à gauche réduit la fenêtre.
- 456 ; un trait horizontal ferme l'application.
- 896321 ; un arc de cercle tournant dans le sens anti-horaire annule la dernière action effectuée (ce qui correspond au bouton “annuler” ou “undo” de nombreuses applications).
- dans des applications de type traitement de texte ou logiciel de dessin, après sélection d'un élément, un “C” peut correspondre au bouton “copy”, “D” à “delete” et “P” à “paste”.

Les codes (et donc les dessins correspondants) sont donnés à titre indicatif, et correspondent à une vision usuelle ou intuitive de l'action correspondante.

3 Travail demandé

3.1 Travail minimal

le projet final devra comporter au minimum :

- une interface graphique permettant de :
 1. dessiner un stroke.
 2. afficher le code correspondant au stroke dessiné.
- un programme traduisant le stroke sous forme de code.
- la réalisation des actions suivantes : agrandissement et réduction de la fenêtre, arrêt de l'application, aide simple mais rappelant le principe du stroke et donnant les strokes utilisables. Vous pouvez utiliser d'autres codes que ceux donnés en 2.3, si les nouveaux vous paraissent plus parlants, à condition de donner la relation code-action dans l'aide.

Pour écrire le programme complet, il est recommandé de créer de nombreuses classes, pour séparer les parties distinctes du projet. Par exemple, la partie reconnaissance de stroke doit se trouver dans une classe distincte de la partie graphique, qui peut elle-même être séparée en plusieurs classes.

Pour gérer les actions de façon simple, vous pourrez utiliser la notion d'interface Java (à ne pas confondre avec les interfaces graphiques). Vous verrez les détails avec les responsables du projet.

3.2 Extensions

Plusieurs directions permettent de poursuivre et d'améliorer le projet (cela est vivement recommandé pour obtenir une bonne note) :

- La réalisation d'actions plus complexes comme :
 1. le "annuler" ("undo").
 2. une aide dans une nouvelle fenêtre, portant plus spécifiquement sur la dernière action entreprise.
 3. dans le cas d'un stroke ne correspondant à aucun code connu, la proposition du code le plus proche.
- L'algorithme de reconnaissance des strokes devra également être robuste. La figure 2 montre un dessin du stroke 159. Il serait difficile de passer exactement par le coin pour passer du carré 1 au carré 5. Il faut donc choisir un critère permettant de différencier le stroke 159 du stroke 1459. Le perfectionnement de l'algorithme de reconnaissance pour augmenter sa robustesse est une des extensions possibles.
- selon votre imagination et les besoins qui vous apparaîtront lors de la réalisation du projet...