

Projet Pavage de plan

1 Description du projet

Le but de ce projet consiste à réaliser un éditeur de pavés d'une sous-partie de la classe PG (cf. le poly de cours de Guy Cousineau, que nous nommerons par la suite **POLY**). Plus précisément, vous devez programmer une application graphique en C qui vous permet i) d'éditer des motifs similaires à celui de la page 40 du **POLY**; et ii) de paver le plan grâce à ces motifs.

1.1 L'éditeur de pavés

Les pavés que vous réaliserez sont ceux décrits sur les pages 38–40 du **POLY**. Vous partirez donc de trois droites verticales équi-espacées représentant les miroirs, ainsi que deux droites horizontales représentant les translations verticales, comme expliqué en haut de la page 39 du **POLY**(cf. la figure 1). Afin de pouvoir modifier la forme et la taille des pavés à votre convenance, vous mettrez des poignées sur les intersections des ces droites. En cliquant sur les poignées, vous ferez en sorte de pouvoir déplacer ces dernières sur l'écran.

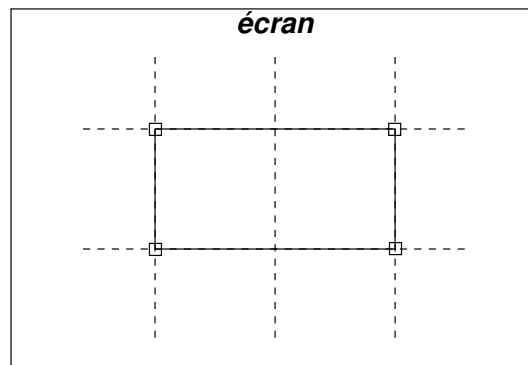
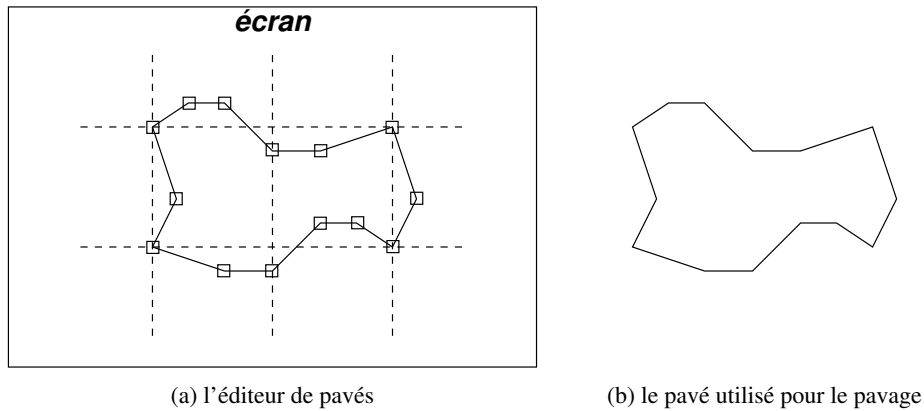


FIG. 1 – Le rectangle de base de PG.

Le cœur de l'éditeur de pavés réside dans les poignées : il s'agit de pouvoir rajouter/supprimer/déplacer des poignées. Celles-ci seront reliées entre elles par des droites et ce sont ces dernières qui définiront le motif de pavage. Par exemple, le motif de pavage correspondant au pavé édité dans la figure 2.(a) est représenté sur la figure 2.(b). Aussi créez-vous des boutons sur la gauche de l'écran de votre éditeur pour sélectionner les différentes actions (rajouter/supprimer/déplacer) possibles sur les poignées. À ceux-ci, vous rajouterez un dernier bouton afin de passer en mode «pavage de tout le plan avec votre pavé». Vous obtiendrez donc un écran similaire à celui de la figure 3.

Lorsque le bouton «ajouter» sera sélectionné, en cliquant sur une des droites de votre pavé, vous rajouterez une poignée à cet endroit, ainsi qu'aux endroits symétriques selon PG (cf. figure 4). Par exemple, sur la figure 4, à gauche, le fait de cliquer en haut à gauche du rectangle pour rajouter une poignée, en rajoute aussi une en bas à droite puisqu'il y a glissement entre la ligne horizontale du bas du rectangle et la ligne horizontale du haut.

La suppression d'une poignée se fait en sélectionnant le bouton «supprimer» et en cliquant sur ladite poignée. Bien Évidemment, cela supprime aussi la poignée symétrique (selon PG). Attention : les 4 poignées du rectangle d'origine ne sont pas supprimables (en effet elles sont là uniquement pour déplacer les axes de symétries et de translation de PG); toutes les autres poignées (c'est-à-dire celles rajoutées explicitement par l'utilisateur) peuvent être éliminées. La suppression se déroule de la manière suivante : si l'on veut éliminer la poignée B de la figure 5.(a), on note que celle-ci se trouve entre les poignées A et C ; on remplace donc les segments $[A, B]$ et $[B, C]$ par un unique segment $[A, C]$. Bien entendu, par symétrie, les segments $[D, E]$ et $[E, F]$ sont remplacés par le segment $[D, F]$.



(a) l'éditeur de pavés

(b) le pavé utilisé pour le pavage

FIG. 2 – Les pavés et leur édition.

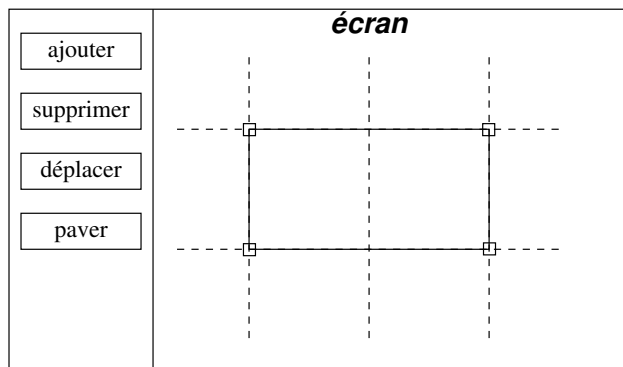


FIG. 3 – Les boutons de l'éditeur de pavés.

Lorsque le bouton «déplacer» est sélectionné, le déplacement d'une poignée s'effectue en cliquant sur la poignée en question et en déplaçant la souris tout en conservant le bouton de la souris pressée. L'affichage est remis à jour en «temps réel». Évidemment, la poignée symétrique selon PG est déplacée elle aussi de manière à ce que le pavé satisfasse les contraintes de PG.

1.2 Le pavage du plan

Lorsque l'utilisateur sélectionne le bouton «paver», l'éditeur de pavés, c'est-à-dire la partie de l'écran à droite des boutons, est remplacé par un écran entièrement pavé avec le motif de l'utilisateur. Ainsi l'éditeur de la figure 6.(a) serait remplacé par l'écran de la figure 6.(b).

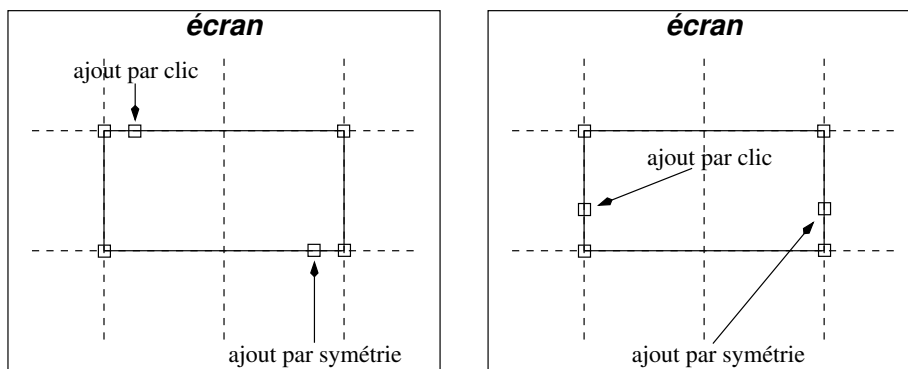


FIG. 4 – Les poignées rajoutées par clic de l'utilisateur et par symétrie sur PG.

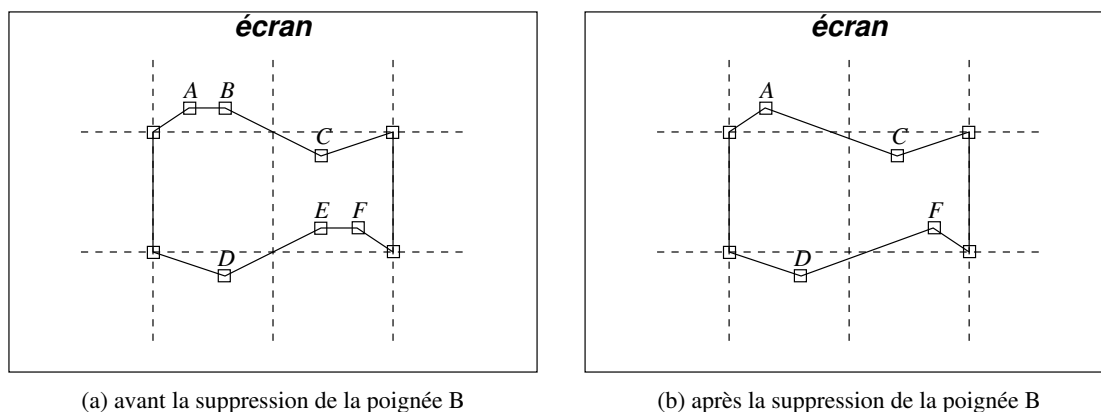


FIG. 5 – La suppression de la poignée B.

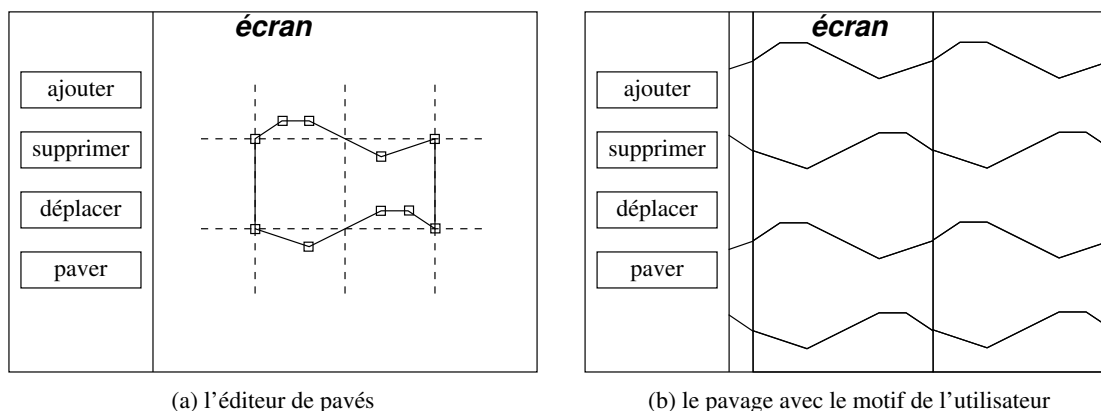


FIG. 6 – Le pavage du plan selon le motif de l'utilisateur.

2 La mise en œuvre et le travail à rendre

Vous programmerez votre projet en Visual C++ en sélectionnant le type «win32 application». Vous avez libre choix des structures de données que vous utiliserez, cela dit réfléchissez bien à celles-ci avant d'entreprendre la programmation du projet : faites en sorte que votre structure prenne en compte les symétries de PG.

À l'issue de ce projet, c'est-à-dire début juillet, vous me rendrez une disquette avec l'ensemble de votre projet (pour que cela tienne sur la disquette, détruisez votre répertoire Debug, mais pas les autres fichiers), ainsi qu'une version papier de votre code et un rapport papier m'indiquant ce que vous avez fait. Le rapport est destiné à permettre i) à un éventuel programmeur de compléter/mettre à jour votre logiciel ; ii) à un utilisateur de savoir comment utiliser votre logiciel. Il y a donc deux parties distinctes. En ce qui concerne la première, vous devez donner toutes les informations utiles pour comprendre rapidement vos algorithmes, le découpage du programme, les variables importantes/globales, la structure des données, pourquoi vous avez utilisé ces structures, bref tout ce qui permettrait à un autre programmeur de ne pas se prendre la tête pendant des jours à essayer de comprendre ce que vous avez fait.

En ce qui concerne votre listing, vous devez soigner votre listing, bien le présenter, vérifier l'indentation avec un notepad ou un wordpad (l'indentation de Visual C++ laisse parfois à désirer quand on sort le listing à l'imprimante). Je rappelle qu'il faut respecter les règles énoncées dans le memento. Sont pris en considération dans la note de projet les aspects suivants :

- la lisibilité du code (indentation, aération, commentaires) ;
- la beauté du code (j'aime les codes simples à lire) ;
- les bugs éventuels ;
- dans une *insignifiante mesure*, la beauté du rendu du logiciel : ce qui m'intéresse dans un projet, ce n'est pas l'interface, mais ce qu'il y a derrière.