

ÉPREUVE D'INFORMATIQUE

(Durée : 2 heures)

L'utilisation des calculatrices **n'est pas autorisée** pour cette épreuve.

Le langage de programmation choisi par le candidat doit être spécifié en tête de la copie et on supposera que ce langage contient une fonction `partieEntiere(x)` qui calcule la partie entière du réel x . On attachera une grande importance à la concision, à la clarté, et à la précision de la rédaction.

* * *

Lissage de vecteurs obliques

Un écran numérique est une matrice de $w \times h$ *pixels* (*picture elements* en anglais), petits carrés dont chacun s'affiche dans une couleur uniforme. Cette grille de pixels est mal adaptée aux tracés de vecteurs obliques pour lesquels un effet de crénelage peut se produire. Pour l'éliminer, on lisse ces tracés en peignant en gris certains pixels se trouvant aux bords de ces tracés.

Nous explorons une version simple du lissage (*anti-aliasing* en anglais) dans le cas du remplissage de demi-plans.



FIG. 1: Effet de crénelage



FIG. 2: Tracé avec lissage

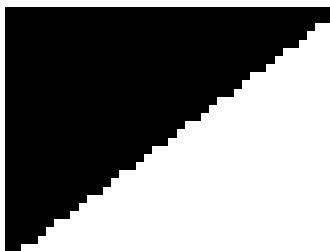


FIG. 3: Effet de crénelage (zoom)

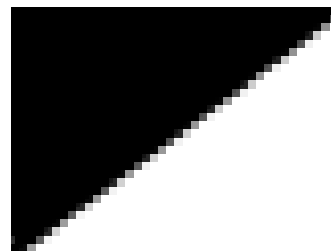


FIG. 4: Tracé avec lissage (zoom)

L'écran de largeur w et hauteur h sera considéré de deux manières. D'une part, l'écran idéal, continu, est la surface définie par le produit cartésien $[0, w[\times [0, h[$, incluse dans $\mathbb{R} \times \mathbb{R}$, où tous les tracés sont possibles. D'autre part, l'écran réel est composé de w pixels en largeur, et h pixels en hauteur. Un pixel est référencé par les coordonnées de son coin *inférieur gauche*. Ainsi l'écran idéal est constitué des pixels de coordonnées (x, y) où x et y sont des entiers satisfaisant $0 \leq x < w$ et $0 \leq y < h$. Comme indiqué sur la figure 5, le pixel (x, y) représente la surface $[x, x + 1[\times [y, y + 1[$. On suppose disposer d'une fonction `peindrePixel`(x, y, g) qui affiche le pixel de coordonnées (x, y) avec l'intensité de gris g (où $g \in \mathbb{R}$ et $0 \leq g \leq 1$, le blanc correspondant au 0 et le noir au 1).

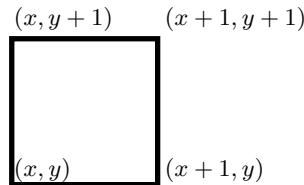


FIG. 5: Le pixel (x, y) représente la surface $[x, x + 1[\times [y, y + 1[$

L'écran est initialement blanc. La région \mathcal{T} à peindre est définie par les trois inégalités :

$$a * x + b \leq y \quad 0 \leq x < w \quad 0 \leq y < h$$

On suppose $0 \leq a \leq 1$ et $b \geq 0$. Cette région \mathcal{T} est donc la partie de l'écran contenue dans le demi-plan \mathcal{P} situé au dessus de la droite Δ d'équation $y = ax + b$.

Partie I. Remplissage simple

Dans cette partie, les pixels totalement contenus dans \mathcal{T} sont peints en noir, ainsi que les pixels à cheval sur la droite Δ .

Question 1 Écrire la procédure `peindreRegion`(w, h, a, b) qui peint la région \mathcal{T} en examinant tous les pixels de l'écran.

La procédure `peindreRegion` examine $w \times h$ pixels. C'est inutile puisqu'initialement l'écran est blanc. Il suffit de n'explorer que les pixels dont l'ordonnée y est minorée par $yMin(x)$ où $yMin$ est défini par :

$$yMin(x) = \text{partieEntière}(ax + b).$$

Question 2 Écrire la procédure `peindreRegionBis`(w, h, a, b) qui peint la région \mathcal{T} en n'examinant que les pixels de \mathcal{T} à noircir.

Partie II. Suréchantillonnage

Une technique simple de lissage est le suréchantillonnage. Elle est illustrée par la figure 6. Avant de noircir ou non le pixel (x, y) , on découpe la zone qu'il couvre en quatre carrés de côté $\frac{1}{2}$. Pour chaque paire d'indices (i, j) vérifiant $0 \leq i, j < 2$, on note $g_{i,j} = 1$ si $(x + \frac{i}{2}, y + \frac{j}{2})$ appartient à \mathcal{T} , et $g_{i,j} = 0$ sinon. L'intensité de gris utilisée pour afficher le pixel (x, y) est alors

$$g = \frac{1}{4} (g_{0,0} + g_{0,1} + g_{1,0} + g_{1,1}).$$

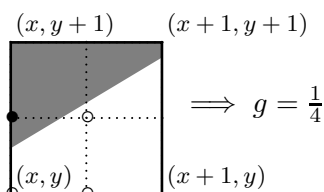


FIG. 6: Calcul d'intensité par suréchantillonnage

Question 3 Écrire la procédure `peindreRegionAA(w, h, a, b)` qui peint la région \mathcal{T} en appliquant le procédé de suréchantillonnage à tous les pixels de l'écran.

Comme à la question I.1, la question II.3 examine un trop grand nombre de pixels. On remarque que dans les cas où $y + 0.5 < ax + b$ ou bien $y \geq a(x + 0.5) + b$, l'intensité calculée est soit 0, soit 1.

Question 4 Écrire la procédure `peindreRegionAABis(w, h, a, b)` qui peint la région \mathcal{T} en n'examinant que les pixels de \mathcal{T} à noircir ou à griser.

Partie III. Calcul d'intensité par surface

Le lissage devient beaucoup plus précis en utilisant toute la gamme des niveaux de gris. Tout pixel (x, y) est peint avec un niveau de gris égal à la surface occupée par \mathcal{T} à l'intérieur de la zone $[x, x + 1[\times [y, y + 1[$.

Comme la pente de la droite Δ est comprise entre 0 et 1, on remarque que, pour x fixé, tous les pixels sont blancs ou noirs sauf les deux pixels $(x, yMin(x))$ et $(x, yMin(x) + 1)$ potentiellement à cheval sur Δ qui peuvent être peints en gris, cf. figure 7.

On note $s_0(x)$ et $s_1(x)$ les surfaces des intersections de ces deux pixels avec le demi-plan \mathcal{P} , et on pose $S(x) = s_0(x) + s_1(x)$.

Question 5 Montrer que l'on a :

$$S(x) = 2 - \frac{a}{2} - ax - b + yMin(x)$$



FIG. 7: Calculs de surfaces quand $y = yMin(x)$

On en déduit les inégalités :

$$1 - \frac{a}{2} \leq S(x) \leq 2 - \frac{a}{2}$$

et les deux cas de la figure 7 correspondant aux inégalités et égalités suivantes :

$S(x) \leq 1 + \frac{a}{2}$	$S(x) \geq 1 + \frac{a}{2}$
$S(x+1) = 1 + S(x) - a$	$S(x+1) = S(x) - a$
$s_0(x) = f(S(x))$	$s_0(x) = S(x) - 1$
$yMin(x+1) = yMin(x) + 1$	$yMin(x+1) = yMin(x)$.

où

$$f(S) = \frac{1}{2a} \left(S - 1 + \frac{a}{2} \right)^2.$$

Question 6 Écrire la procédure `peindreRegionAAA(w, h, a, b)` qui peint la région \mathcal{T} avec le calcul d'intensité par surface donné par les tables ci-dessus pour suivre la valeur de $S(x)$ et de $yMin(x)$ au cours de la variation de l'abscisse x .

La procédure précédente recalcule $f(S)$ trop fréquemment. Comme $f(S)$ est un polynôme de degré 2, et que S évolue de manière assez simple, on peut suivre la valeur de $f(S)$ de proche en proche grâce au calcul de sa dérivée $f'(S)$. Posons $f0 = f(S)$, $f1 = f'(S)$, $f2 = -af'(S)$.

Question 7 Donner deux suites de quatre additions/soustractions pour obtenir les triplets : $\langle f(S+1), f'(S+1), -af'(S+1) \rangle$ et $\langle f(S-a), f'(S-a), -af'(S-a) \rangle$ à partir de $\langle f0, f1, f2 \rangle$.

Question 8 Écrire une procédure `peindreRegionAAAbis(w, h, a, b)` qui peint la région \mathcal{T} avec le calcul d'intensité par surface qui n'utilise que des comparaisons et des additions au cours de la variation de l'abscisse x .

* *
*