

Lissage de vecteurs obliques

Question 1 : On parcourt tous les pixels de l'écran par une double boucle et on peint tous les pixels de coordonnées (x, y) vérifiant $\lfloor ax + b \rfloor \leq y$.

```

1 def peindreRegion(w, h, a, b):
2     for x in range(w):
3         for y in range(h):
4             if partieEntiere(a*x+b) <= y:
5                 peindrePixel(x, y, 1)

```

Question 2 : Pour chaque ordonnée x , on peint les points de coordonnées (x, y) pour y compris entre $ax + b$ et $h - 1$.

```

1 def partieEntiere(x):
2     return(int(x))
3
4 def peindreRegionbis(w, h, a, b):
5     def yMin(x):
6         return(partieEntiere(a*x+b))
7     for x in range(w):
8         for y in range(yMin(x), h):
9             peindrePixel(x, y, 1);

```

(PSI*)

Question 3 : On parcourt tous les pixels de l'écran par une double boucle et on peint tous les pixels de coordonnées (x, y) vérifiant $\lfloor ax + b \rfloor \leq y$. Pour chaque pixel, on calcule l'intensité de gris à utiliser (double boucle des lignes 4 à 11).

```

1 def peindreRegionAA(w, h, a, b):
2     def yMin(x):
3         return(partieEntiere(a*x+b))
4     for x in range(w):
5         for y in range(yMin(x), h):
6             g = 0
7             for i in range(2) :
8                 for j in range(2):
9                     if a*(x+.5)+b < y+.5*j :
10                        g = g+.25
11                peindrePixel(x, y, g)

```

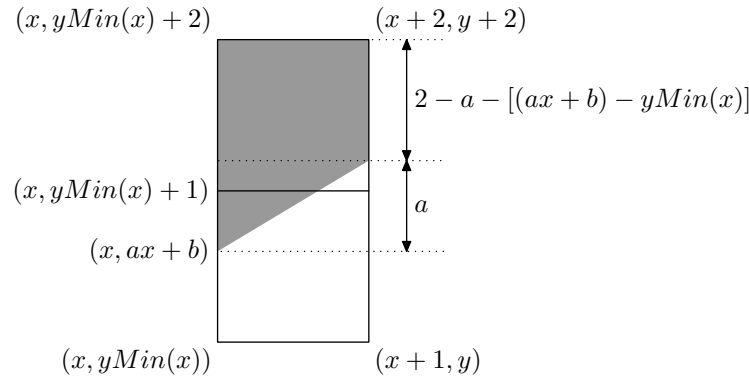
Question 4 : On ne regarde que les pixels de coordonnées (x, y) vérifiant $ax + b \leq y$ et on ne traite de façon particulière que ceux qui vérifient $ax + b - 0.5 \leq y \leq a(x + 0.5) + b$ pour calculer l'intensité du gris.

```

1 def peindreRegionAAbis(w, h, a, b):
2     def yMin(x):
3         return partieEntiere(a*x+b)
4     for x in range(w) :
5         for y in range(yMin(x), h):
6             g = 0.
7             if a*(x+.5)+b <= y:
8                 g = 1
9             else:
10                for i in range(2):
11                    for j in range(2):
12                        if a*(x+.5*i)+b < y+.5*j:
13                            g = g+.25
14                peindrePixel(x, y, g)

```

Question 5 :



En reprenant les données du dessin ci-dessus, $S(x)$ apparaît comme la somme des aires d'un triangle et d'un rectangle dont l'un des cotés a pour mesure 1 et l'autre respectivement a et $2 - a - [(ax + b) - yMin(x)]$ ce qui nous donne :

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

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

Question 6 : Lignes 3 à 7, on écrit la fonction f donnée par l'énoncé. Ligne 5, on calcule ce qui correspond à $yMin(0)$ et ligne 6 ce qui correspond à $S(0)$. La boucle des lignes 7 à 22 permet de traiter tous les points d'abscisses x . Pour les points d'abscisses x , on ne peint que les points d'ordonnées comprises entre $yMin(x)$ et $h-1$, $yMin(x)$ étant représenté par la variable `yMin`. On commence par peindre les pixels noirs (boucle des lignes 8 à 9). Puis on traite le cas des points à la frontière qui vont être peints en gris (lignes 10 à 17) en prenant soin de vérifier qu'ils sont dans la fenêtre graphique. On calcule $S(x+1)$ et $yMin(x+1)$ représentés par les variables respectives `S` et `yMin` (lignes 18 à 22) en suivant les indications données par les tableaux de la page 4 de l'énoncé.

```

1 def peindreRegionAAAd(w, h, a, b):
2     def f(x):
3         temp = x-1.+a/2.
4         return(temp*temp*0.5/a)
5     yMin = partieEntiere(b)
6     S = 2. - a/2. - b + partieEntiere(b)
7     for x in range(w) :
8         for y in range(yMin+2,h):
9             peindrePixel(x, y, 1.)
10        if S <= 1.+a/2.:
11            s0 = f(S)
12        else :
13            s0 = S-1.
14        if yMin < h:
15            peindrePixel(x, yMin, s0)
16        if yMin+1 < h:
17            peindrePixel(x, yMin+1, S-s0)
18        if S <= 1.+a/2.:
19            S = 1.+S-a
20            yMin = yMin+1
21        else :
22            S = S-a

```

Question 7 : En posant $\alpha = S - 1 + \frac{a}{2}$, on a :

$$f(S) = f_0 = \frac{1}{2a}\alpha^2, \quad f'(S) = \frac{1}{a}\left(S - 1 + \frac{a}{2}\right) = \frac{1}{a}\alpha = f_1,$$

$f_2 = -a.f'(S) = -\alpha$. D'où :

$$f(S+1) - f(S) = \frac{1}{2a}((\alpha+1)^2 - \alpha^2) = \frac{1}{2a}(2\alpha+1) = \frac{1}{a}\alpha + \frac{1}{2a} = f_1 + \frac{1}{2a}$$

$$\Rightarrow f(S+1) = f_0 + f_1 + \frac{1}{2a}$$

Comme f' est un polynôme de degré 1 de coefficient dominant $\frac{1}{a}$, on a :

$$f'(S+1) = f'(S) + \frac{1}{a} \Rightarrow f'(S+1) = f_1 + \frac{1}{a}$$

$$\text{et } -af'(S+1) = -af'(S) - 1 \Rightarrow -af'(S+1) = f_2 - 1$$

$$f(S-a) - f(S) = \frac{1}{2a}((\alpha-a)^2 - \alpha^2) = \frac{1}{2a}(2\alpha-a).(-a) = -\alpha + \frac{a}{2} = f_2 + \frac{a}{2}$$

$$\Rightarrow f(S-a) = f_0 + f_2 + \frac{a}{2}$$

Comme f' est un polynôme de degré 1 de coefficient dominant $\frac{1}{a}$, on a :

$$f'(S-a) = f'(S) - 1 \Rightarrow f'(S-a) = f_1 - 1$$

$$\text{et } -af'(S-a) = -af'(S) - a \Rightarrow -af'(S-a) = f_2 + a$$

Question 8 : On commence par calculer une fois pour toutes les quantités $\frac{1}{a}$, $\frac{1}{2a}$, $\frac{a}{2}$ (lignes 2 à 4). La variable `minimum` correspond à `yMin(x)` pour le x courant. On initialise les variables `f0`, `f1` et `f2` (ligne 8-10). Dans la boucle des lignes 11-35, on traite les différentes abscisses possibles. Lignes 12-13, on traite les pixels noirs. Lignes 14-35, on traite les pixels du bord de la région $\mathcal{T}$. On traite les deux cas $S(x) \leq 1 + \frac{a}{2}$ (lignes 14-21) et $S(x) > 1 + \frac{a}{2}$ (lignes 23-26) Dans le premier cas, on calcule `f0`, `f1` et `f2` qui correspondent à $f(1+S(x)-a) = f(S(x+1))$ en deux étapes, en utilisant les propriétés démontrées à la question 8. Dans le deuxième cas, on calcule `f0`, `f1` et `f2` qui correspondent à $f(S(x)-a) = f(S(x+1))$ en une étape. Lignes 27-30, on peint les deux pixels à la limite du demi plan avec une couleur proportionnelle à la surface couverte pas le demi plan.

```

1 def peindreRegionAAAbis(w, h, a, b):
2     inv_a = 1./a
3     inv_a_div_2 = 1./((2.*a))
4     a_sur_2 = a/2.
5     yMin = partieEntiere(b)
6     S = 2. - a/2. - b + yMin
7     alpha = S - 1. + a_sur_2
8     f0 = alpha*alpha/(2.*a)
9     f1 = alpha/a
10    f2 = -alpha
11    for x in range(w):
12        for y in range(yMin+2,h):
13            peindrePixel(x, y, 1.)
14        if S <= 1.+a_sur_2:
15            s0 = f0
16            f0 = f0+f1+inv_a_div_2
17            f1 = f1+inv_a
18            f2 = f2-1.
19            f0 = f0+f2+a_sur_2
20            f1 = f1-1.
21            f2 = f2+a
22        else :
23            s0 = S-1.
24            f0 = f0+f2+a_sur_2
25            f1 = f1-1.
26            f2 = f2+a
27        if yMin < h:
28            peindrePixel(x, yMin, s0)
29        if yMin+1 < h:
30            peindrePixel(x, yMin+1, S-s0)
31        if S <= 1.+a_sur_2:
32            S = 1.+S-a
33            yMin = yMin+1
34        else :
35            S = S-a

```