

M3, Infographie 2D. Deuxième session

mardi 9 septembre 2003

La précision et la clarté de votre rédaction sont *fondamentales*. Cours et TD sont autorisés. Durée 1 heure.

Exercice 1. [13pts] En 2010 un nouveau processeur optimisé pour le graphisme est sorti. Avec ce nouveau processeur, en une seule instruction on fait des opérations sur des millions de pixels ! Il travaille avec des “vraies” couleurs (RVB) qui peuvent être plus ou moins transparentes (A). Malheureusement son concepteur n’avait plus assez de place sur la puce électronique pour le terminer et certaines opérations devront être réalisées de manière logicielle. Votre mission (que vous devez accepter pour réussir l’examen) est de donner pour chacune des fonctions manquantes une idée générale sur la manière de réaliser l’opération et un algorithme précis s’il est simple.

FUNCTIONALITÉS DU PROCESSEUR GRAPHIQUE : les fonctions suivantes du processeurs sont cablées. Chaque algorithme spécifie ce que fait la fonction, vous n’avez pas d’accès direct à la mémoire image.

– *Allocation de mémoire graphique :*

```
RVBA** allouer_image(largeur, hauteur : entiers)
void liberer_image(image : RVBA**)
```

– *Écriture d’un pixel dans une matrice en tenant compte de la transparence :*

```
ecritPixel(image : RVBA**, x, y : entiers, couleur : RVBA)
debut
  image[y][x].r <-
    ( couleur.a*couleur.r + (256-couleur.a)*image[y][x].r ) / 256
  // idem vert et bleu
  image[y][x].a <- couleur.a
fin
```

– *Copie d’une matrice de pixels en tenant compte de la transparence :*

```
void copier(depart, arrivee : RVBA**, Xdebut, Ydebut, Xfin, Yfin : entiers
           X, Y : entiers)
debut
  i, j : entiers
  pour i de Xdebut a Xfin-1
    pour j de Ydebut a Yfin-1
      ecritPixel(arrivee, i-Xdebut+X, j-Ydebut+Y, debut[j][i])
fin
```

FUNCTIONALITÉS QUE VOUS DEVEZ AJOUTER : pour ajouter les fonctionnalités suivantes, vous ne devez pas utiliser directement le contenu de la matrice de pixel, vous devez passer par les 3 fonctions cablées décrites précédemment.

1. [2pt] Copie d’une même couleur *opaque* dans une zone rectangulaire de pixels. Il est important que cette fonction soit rapide, donnez sa complexité en nombre d’appels aux fonctions cablées.

```
void efface(image : RVBA**, Xdebut, Ydebut, Xfin, Yfin : entiers,
           couleur : RVBA) ; /* Le A n’est pas utilise */
```

2. [2pt] Même question, mais en faisant un effacement qui tienne compte de la transparence.

3. [2pt] On désire faire de l’affichage de caractères de couleur noire anti-aliassés. Comment procéderiez-vous ?

4. [2pt] Quelle fonction devrait-on ajouter au processeur pour pouvoir faire des textes de toutes les couleurs ?

5. [2pt] On désire faire un tracé de droite, comment l’implémenter ?

6. [2pt] On désire faire un tracé de droite anti-aliassé, comment l’implémenter ?

Remarque : si vous avez répondu de la même manière aux deux dernières questions, il y a un problème.

Exercice 2. [7pts] On considère la courbe de Bézier cubique \mathcal{C} associée aux points de contrôle $M_0 = (0, 0)$, $M_1 = (2, 1)$, $M_2 = (2, -2)$ et $M_3 = (-2, -2)$.

1. [2pts] Calculez les coordonnées du point générique $M(u)$ de la courbe \mathcal{C} .

2. [1pt] En quel point X autre que l’origine la courbe de Bézier coupe-t-elle l’axe des abscisses ? En quel point Y autre que l’origine la courbe de Bézier coupe-t-elle l’axe des ordonnées ?

3. [2pts] Calculez l’équation de la tangente à \mathcal{C} en X et l’équation de la tangente à \mathcal{C} en Y .

4. [1pt] Calculez le point d’intersection I entre ces deux tangentes.

5. [1pt] Tracez la courbe de Bézier et les tangentes en question.