Module Bezier

structure de Bezier (degrès 2)

```c
/*******************************/
/* TYPE DE BEZIER  DE DEGRES 2 */
/*******************************/

typedef struct Bezier2_
{
    Point C0;
    Point C1;
    Point C2;
}Bezier2;
```

```c
/*initialisation de courbe de bezier de degres 2*/
Bezier2 init_bezier2(Point C0, Point C1, Point C2)
{
   Bezier2 B = {C0, C1, C2};
   return B;
}

/* Cette fonction calcul C(t) pour t entre [0 et 1] et pour une courbe de
** bezier de degres 2
*/
Point calcul_bezier2(Bezier2 B, double t)
{
   if (t<0 || t > 1)
       ERREUR_FATALE(RED"Erreur dans calcul_bezier2 : t doit etre compris entre 0 et 1 (inclus)\n");
   return add_point(add_point(mult_scalaire_point(B.C0, pow(1-t, 2)), mult_scalaire_point(B.C1, 2*(1-t)*t)), mult_scalaire_point(B.C2, pow(t, 2)));
}
```

```c
/* CETTE fonction approxime le contour polygonal CONT par une courbe de bezier
** de degres 2
*/
Bezier2 approx_bezier2(Tableau_Point CONT, UINT j1, UINT j2)
{
    int n = j2 - j1;
    Point P, C1;
    double alpha, beta;

    if (n <= 0)
        ERREUR_FATALE(RED"Erreur approx_bezier2 : erreur dans le passage des indice (j1 < j2)\n"RESET);


    //cas n = 1
    if (n == 1)
    {
        C1 = mult_scalaire_point(add_point(CONT.tab[j1], CONT.tab[j2]), 0.5);
        return init_bezier2(CONT.tab[j1], C1, CONT.tab[j2]);
    }

    //si n ≥ 2

    //cacul de la somme des Pj
    P = set_point(0, 0);
    for (int i = 1; i <= n-1; i++)
    {
        P = add_point(CONT.tab[j1+i], P);
    }

    //calcul de alpha et beta
    alpha = (3* (double) n)/(pow((double) n, 2) - 1);
    beta = (1 - 2* (double) n )/(2*((double) n +1));

    C1 = add_point(mult_scalaire_point(P, alpha), mult_scalaire_point(add_point(CONT.tab[j1], CONT.tab[j2]), beta));

        return init_bezier2(CONT.tab[j1], C1, CONT.tab[j2]);


}
```

```c
/* Cette fonction calcul et renvoie la distance entre le point P et la courbe
** de bezier de degres 2 B
*/
double distance_point_bezier2(Point P, Bezier2 B, double t)
{
    Point Ct;
    Ct = calcul_bezier2(B, t);
    return distance(P, Ct);

}
```

```c
/*Cette fonction convertis une courbe de bezier de degres 2 en courbe de bezier
** de degres 3 par elevation au degres
*/
Bezier3 elevation_degre(Bezier2 B)
{
   Point D0, D1, D2, D3;
   D0 = B.C0;
   D3 = B.C2;
   D1 = add_point(mult_scalaire_point(B.C0,(double)1/3), mult_scalaire_point(B.C1, (double)2/3));
   D2 = add_point(mult_scalaire_point(B.C2,(double)1/3), mult_scalaire_point(B.C1, (double)2/3));

   return init_bezier3(D0, D1, D2, D3);
}
```

Module simplification

```
/*¬
- simplifier la partie du contour CONT compris entre les indices j1 et j2 avec la distance-seuil d¬
- la fonction renvoie la s´equence de segments L contenant des point destiné a etre des courbe de bezier de degres 2¬
- procedure recursive de type "diviser pour r´egner" ("divide and conquer")¬
*/¬
Contour simplification_douglas_peucker_bezier2(Tableau_Point CONT, UINT j1, UINT j2, UINT d)¬
{¬
¬
    // il faut que j1 < j2¬
    if (j1 >= j2)¬
        ERREUR_FATALE(RED"Erreur avec les indices j1 et j2 dans la fonction simplification_segemnt"RESET);¬
¬
    UINT k; //indice du point le plus éloigné du segment [Pj1 Pj2]¬
    double dmax, dj; //distance entre Pk et le segment [Pj1 Pj2]¬
    UINT i,j; //variable temporaire¬
    int m;¬
    double ti;¬
    Bezier2 B;¬
¬
    m = j2 - j1; //-- nombre de segments de CONT entre les indices j1 et j2¬
¬
    //-- (0) approcher la s´equence de n + 1 points CONT[j1..j2] par une B´ezier B de degr´e 2¬
    B = approx_bezier2(CONT, j1, j2);¬
¬
¬
/*¬
- (1) rechercher le point Pk le plus eloigne de la bezier B¬
- ainsi que la distance dmax correspondante¬
*/¬
    dmax = 0;¬
    k = j1;¬
¬
    for (j = j1 + 1; j <= j2; j++)¬
    {¬
        i = j-j1;¬
        ti = (double)i/(double)m;¬
¬
        dj = distance_point_bezier2(CONT.tab[j], B, ti);¬
¬
        if(dmax < dj)¬
        {¬
            dmax = dj;¬
            k = j;¬
        }¬
¬
    }¬
¬
```

```
¬
    if (dmax <= d)¬
    {¬
        //-- (2) dmax ≤ d : simplification suivant la B´ezier B¬
        Contour C = creer_liste_Point_vide();¬
        C = ajouter_element_liste_Point(C, B.C0);¬
        C = ajouter_element_liste_Point(C, B.C1);¬
        C = ajouter_element_liste_Point(C, B.C2);¬
        return C;¬
    }¬
    else¬
    {¬
        //- (3) dmax > d : "diviser pour regner"¬
¬
        //- (3.1) d´ecomposer le probl`eme en deux¬
        //- simplifier la partie du contour CONT compris entre les indices j1 et k avec la distance-seuil d¬
        Contour C1 = simplification_douglas_peucker_bezier2(CONT, j1, k, d);¬
¬
        //- simplifier la partie du contour CONT compris entre les indices k et j2 avec la distance-seuil d¬
        Contour C2 = simplification_douglas_peucker_bezier2(CONT, k, j2, d);¬
¬
        //- (3.2) fusionner les deux s´equences L1 et L2 et les renvoyer¬
        return concatener_liste_Point(C1, C2);¬
¬
    }¬
¬
}¬
¬
```

```
/* Cette procédure prend en argument :¬
    - le nom d'un fichier contour n1¬
    - le nom d'un fichier de sortie n2¬
    - un reel d¬
Puis calcul grace a l'algorithme de Douglas Peucker  pour le scourbe de bezier 2 la simplification de n1 et met le resultat dans n2.¬
Cette fonction fais les calcul a la volé : c'est a dire pour chacun des contour de n1 elle va calculer sa simplification puis le mettre dans n2 avant de passer au contour suivant.¬
*/¬
void simplification_bezier2(char* nom_fichier_entree, char* nom_fichier_sortie, float d)¬
{¬
¬
    UINT nb_contour, taille_contour;¬
    Tableau_Point Cont;¬
    Point P;¬
    Contour L;¬
    UINT i, j;¬
¬
    FILE* f = fopen(nom_fichier_entree, "r");¬
    if (!f)¬
        ERREUR_FATALE(RED"ERREUR simplification_bezier2 : le fichier d'entree n'a pas pu être ouvert"RESET);¬
¬
    fscanf(f, "%d\n\n", &nb_contour);¬
¬
¬
    FILE* fich = fopen(nom_fichier_sortie, "w"); //fichier de sortie avec le contour simplifier¬
    if (!f)¬
        ERREUR_FATALE(RED"ERREUR simplification_bezier2 : le fichier de sortie n'a pas pu être ouvert"RESET);¬
¬
    fprintf(fich, "%d                  \n\n", nb_contour);¬
¬
    for (i = 0; i < nb_contour; i++)¬
    {¬
        fscanf(f, "%d\n", &taille_contour);¬
        Cont = cree_Tableau_point_vide(taille_contour);¬
¬
        for (j = 0; j < taille_contour; j++)¬
        {¬
            fscanf(f, "%lf %lf\n", &P.x, &P.y);¬
            Cont.tab[j] = P;¬
        }¬
¬
        L = simplification_douglas_peucker_bezier2(Cont, 0, taille_contour-1, d);¬
        Cont = supprimer_Tableau_point(Cont);¬
        ecrire_Contour_fichier(&L, fich);¬
¬
    }¬
¬
    fclose(f);¬
    fclose(fich);¬
¬
}¬
¬
```

MODULE EPS

```
void cree_image_eps_bezier2(const char* fichier_contour, const UINT Largeur_Image, const UINT Hauteur_Image, const char* nom_fichier, const char mode)
{
    FILE* fich = fopen(nom_fichier, "w");

    if (!fich)
        ERREUR_FATALE(RED"ERREUR cree_image_eps : le fichier n'a pas pu être ouvert"RESET);

    //overtur du fichier qui contiens tous les contours
    FILE* fich_contour = fopen(fichier_contour, "r");

    if (!fich_contour)
        ERREUR_FATALE(RED"ERREUR cree_image_eps : le fichier contour n'a pas pu être ouvert"RESET);

    //declaration de variables
    int nb_contour, taille_contour, nb_courbe;
    double val_x, val_y;
    bool premier;
    Bezier2 B;
    Bezier3 B3;
    Point C0, C1, C2;


    fprintf(fich, "%%!PS-Adobe-3.0 EPSF-3.0\n");
    fprintf(fich, "%%%%BoundingBox: 0 0 %d %d\n\n", Largeur_Image, Hauteur_Image);


    fscanf(fich_contour, "%d\n\n", &nb_contour);
    nb_courbe = 0;

    for (int i = 0 ; i < nb_contour ; i++)
    {
       fscanf(fich_contour, "%d\n", &taille_contour);
       premier = true; //sert a ecrire le "moveto"

        for (int j = 0 ; j < taille_contour ; j+=3)
        {
            fscanf(fich_contour, "%lf %lf\n", &val_x, &val_y);
            C0 = set_point(val_x, val_y);

            fscanf(fich_contour, "%lf %lf\n", &val_x, &val_y);
            C1 = set_point(val_x, val_y);

            fscanf(fich_contour, "%lf %lf\n", &val_x, &val_y);
            C2 = set_point(val_x, val_y);

            B = init_bezier2(C0, C1, C2);
            //elevation au degres car il faut que se soit de degres 3
            B3 = elevation_degre(B);
            if (premier) //pour ecrive un seul moveto par contour
            {
                fprintf(fich, "%lf %lf moveto\n", B3.C0.x, (double)Hauteur_Image - B3.C0.y);
                premier = false;
            }

            fprintf(fich, "%lf %lf\n", B3.C1.x, (double)Hauteur_Image - B3.C1.y);
            fprintf(fich, "%lf %lf\n", B3.C2.x, (double)Hauteur_Image - B3.C2.y);
            fprintf(fich, "%lf %lf curveto\n\n", B3.C3.x, (double)Hauteur_Image - B3.C3.y);
            nb_courbe++;

        }
        fscanf(fich_contour, "\n");
        fprintf(fich, "\n");
    }

    fprintf(fich, "%% largeur du tracé\n");
    fprintf(fich, "0 setlinewidth\n");

    fprintf(fich, "%% Instruction de tracé\n");
    if (mode == 'F')
        fprintf(fich, "fill\n");
    else
        fprintf(fich, "stroke\n");

    fprintf(fich, "showpage\n");


    printf(BLU"Le contour simplifié contient %d courbe de bezier de degres 2\n"RESET, nb_courbe);
    fclose(fich);
    fclose(fich_contour);

}
```

TABLEAU POUR Asterix3

image initial

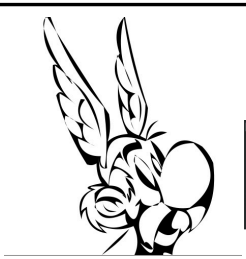
```
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 32 contour et un total de 12926 segments dans le fichier IMAGES_TEST/Asterix3.contour.txt
--------------------------------------------------------------------------------
```

d=1

```
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$ ./TEST/test_simplification_bezier IMAGES_TEST/Asterix3.pbm 1
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 32 contour et un total de 12926 segments dans le fichier IMAGES_TEST/Asterix3.contour.txt
--------------------------------------------------------------------------------
Le contour simplifié contient 966 courbe de bezier de degres 2
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$
```

d= 3

```
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$ ./TEST/test_simplification_bezier IMAGES_TEST/Asterix3.pbm 3
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 32 contour et un total de 12926 segments dans le fichier IMAGES_TEST/Asterix3.contour.txt
--------------------------------------------------------------------------------
Le contour simplifié contient 296 courbe de bezier de degres 2
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$
```



d = 10

```
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$ ./TEST/test_simplification_bezier IMAGES_TEST/Asterix3.pbm 10
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 32 contour et un total de 12926 segments dans le fichier IMAGES_TEST/Asterix3.contour.txt
--------------------------------------------------------------------------------
Le contour simplifié contient 158 courbe de bezier de degres 2
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$
```

d= 30

```
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$ ./TEST/test_simplification_bezier IMAGES_TEST/Asterix3.pbm 30
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 32 contour et un total de 12926 segments dans le fichier IMAGES_TEST/Asterix3.contour.txt
--------------------------------------------------------------------------------
Le contour simplifié contient 69 courbe de bezier de degres 2
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$
```

poun "l_cursif"

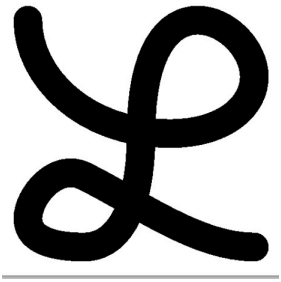
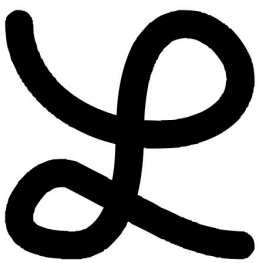


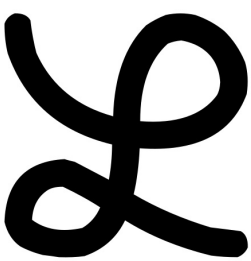
image initial

```
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 3 contour et un total de 4228 segments dans le fichier IMAGES_TEST/lettre-L-cursive.contour.txt
--------------------------------------------------------------------------------
```



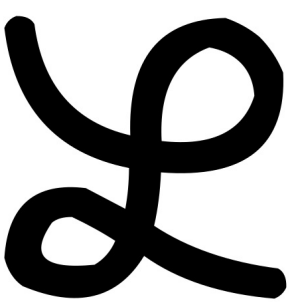
d = 1

```
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 3 contour et un total de 4228 segments dans le fichier IMAGES_TEST/lettre-L-cursive.contour.txt
--------------------------------------------------------------------------------
Le contour simplifié contient 255 courbe de bezier de degres 2
```



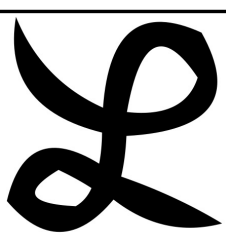
d = 3

```
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$ ./TEST/test_simplification_bezier IMAGES_TEST/lettre-L-cursive.pbm 3
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 3 contour et un total de 4228 segments dans le fichier IMAGES_TEST/lettre-L-cursive.contour.txt
--------------------------------------------------------------------------------
Le contour simplifié contient 40 courbe de bezier de degres 2
```



d = 10

```
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$ ./TEST/test_simplification_bezier IMAGES_TEST/lettre-L-cursive.pbm 10
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 3 contour et un total de 4228 segments dans le fichier IMAGES_TEST/lettre-L-cursive.contour.txt
--------------------------------------------------------------------------------
Le contour simplifié contient 26 courbe de bezier de degres 2
```



d = 30

```
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$ ./TEST/test_simplification_bezier IMAGES_TEST/lettre-L-cursive.pbm 30
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 3 contour et un total de 4228 segments dans le fichier IMAGES_TEST/lettre-L-cursive.contour.txt
--------------------------------------------------------------------------------
Le contour simplifié contient 15 courbe de bezier de degres 2
```

pour ColombeDePaix

image initial

```
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 106 contour et un total de 21764 segments dans le fichier IMAGES_TEST/ColombesDeLaPaix.contour.txt
--------------------------------------------------------------------------------
```

d = 1

```
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$ ./TEST/test_simplification_bezier IMAGES_TEST/ColombesDeLaPaix.pbm 1
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 106 contour et un total de 21764 segments dans le fichier IMAGES_TEST/ColombesDeLaPaix.contour.txt
--------------------------------------------------------------------------------
Le contour simplifié contient 1599 courbe de bezier de degres 2
```

d = 3

```
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$ ./TEST/test_simplification_bezier IMAGES_TEST/ColombesDeLaPaix.pbm 3
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 106 contour et un total de 21764 segments dans le fichier IMAGES_TEST/ColombesDeLaPaix.contour.txt
--------------------------------------------------------------------------------
Le contour simplifié contient 587 courbe de bezier de degres 2
```

d = 10

```
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$ ./TEST/test_simplification_bezier IMAGES_TEST/ColombesDeLaPaix.pbm 10
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 106 contour et un total de 21764 segments dans le fichier IMAGES_TEST/ColombesDeLaPaix.contour.txt
--------------------------------------------------------------------------------
Le contour simplifié contient 295 courbe de bezier de degres 2
```

d = 30

```
sagsag@MacSag[~/Desktop/SAGAR_MITTAL/SAGAR_MITTAL_2023_2024/S2/MAP401/projet/TACHE7]$ ./TEST/test_simplification_bezier IMAGES_TEST/ColombesDeLaPaix.pbm 30
--------------------------------------------------------------------------------
Les dimension de cette images sont : 500x500
Il y a 106 contour et un total de 21764 segments dans le fichier IMAGES_TEST/ColombesDeLaPaix.contour.txt
--------------------------------------------------------------------------------
Le contour simplifié contient 148 courbe de bezier de degres 2
```