

Compte-Rendu Tache 7.1

Mathis Grange & Mattéo Decorsaire

Ajouts et/ou modifications

Dans `geom2d.h`

```
// Implémentation d'un type pour chaque degré de Bézier
typedef struct Bezier2_{
    Point C0;
    Point C1;
    Point C2;
} Bezier2;

typedef struct Bezier3_{
    Point C0;
    Point C1;
    Point C2;
    Point C3;
} Bezier3;

double distance_point_bezier2(Point P, Bezier2 B, float t);

/*Conversion pour la syntaxe eps*/
Bezier3 Bezier2ToBezier3(Bezier2 B2);
```

Dans `structures.h`

```
/*On fait une liste de Bézier3 et ses fonctions associées*/
typedef struct Cellule_Liste_Bezier3_{
    Bezier3 data;
    struct Cellule_Liste_Bezier3_ *suiv;
} Cellule_Liste_Bezier3;

typedef struct Liste_Bezier3_{
    unsigned int taille;
    Cellule_Liste_Bezier3 *first;
    Cellule_Liste_Bezier3 *last;
} Liste_Bezier3;

typedef struct Tableau_Bezier3_{
    unsigned int taille;
    Bezier3 *tab;
} Tableau_Bezier3;

Liste_Bezier3 creer_liste_Bezier3_vide();
```

```

void ajouter_element_liste_Bezier3(Liste_Bezier3 *L, Bezier3 e);

Liste_Bezier3 supprimer_liste_Bezier3(Liste_Bezier3 L);

Liste_Bezier3 concatener_liste_Bezier3(Liste_Bezier3 L1, Liste_Bezier3 L2);

Tableau_Bezier3 sequence_bezier3_liste_vers_tableau(Liste_Bezier3 L);

```

Dans `bezier.h`

```

//donne l'image de t sur la courbe B
Point point_bezier2(Bezier2 B, double t);

//algo d'approximation
Bezier2 approx_bezier2(Tableau_Point P, int j1, int j2);

// algo de douglas peucker
Liste_Bezier3 simplification_douglas_peucker_bezier2(Tableau_Point T, int
j1, int j2, double d);

```

Dans `eps.h`

```

// Pour un contour on écrit la combinaison de curveto qui correspond
void ecrire_eps_Bezier(FILE *f, Tableau_Bezier3 T);

// Fonction "finale"
void PbmToEps_Bezier2(char *nom_entree, char *nom_sortie, double d);

```

Tests `approx_bezier2`

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include "bezier.h"
#include "structures.h"

#include"eps.h"

int main(int argc, char *argv[]) {
    if (argc != 9 && argc != 7 && argc != 3){
        printf("Usage : ./test_approx2 <j1> <j2> <Q0x> <Q0y> <Q1x> <Q1y>
<Q2x> <Q2y> \n ou bien ./test_approx2 <1> <P0x> <P0y> <P1x> <P1y> ou bien
on prend n=8 pour avoir l'exemple du cours");
        return 0;
    }
}

```

```

int j1 = atoi(argv[1]);
int j2 = atoi(argv[2]);
int n = j2 - j1;

printf("%d\n", n);
Liste_Point listeP = creer_liste_Point_vide();

if(n==0){
    ERREUR_FATALE("n=0");
}

if(n==1){
    Point P0 = set_point(atof(argv[3]), atof(argv[4]));
    Point P1 = set_point(atof(argv[5]), atof(argv[6]));
    printf("=====\n");
    printf("Point P0: (%.0f, %.0f)\n", P0.x, P0.y);
    printf("Point P1: (%.0f, %.0f)\n", P1.x, P1.y);
    printf("=====\n");
    ajouter_element_liste_Point(&listeP, P0);
    ajouter_element_liste_Point(&listeP, P1);
    printf("test\n");
}

if(n == 8){ // exemple page 49
    Point P0 = set_point(0,0);
    ajouter_element_liste_Point(&listeP, P0);
    Point P1 = set_point(1,0);
    ajouter_element_liste_Point(&listeP, P1);
    Point P2 = set_point(1,1);
    ajouter_element_liste_Point(&listeP, P2);
    Point P3 = set_point(1,2);
    ajouter_element_liste_Point(&listeP, P3);
    Point P4 = set_point(2,2);
    ajouter_element_liste_Point(&listeP, P4);
    Point P5 = set_point(3,2);
    ajouter_element_liste_Point(&listeP, P5);
    Point P6 = set_point(3,3);
    ajouter_element_liste_Point(&listeP, P6);
    Point P7 = set_point(4,3);
    ajouter_element_liste_Point(&listeP, P7);
    Point P8 = set_point(5,3);
    ajouter_element_liste_Point(&listeP, P8);
    printf("Attendu : C1 = (1.547619,2.452381)\n");
}

else if (n >= 2) {
    Point Q0 = set_point(atof(argv[3]), atof(argv[4]));
    Point Q1 = set_point(atof(argv[5]), atof(argv[6]));
    Point Q2 = set_point(atof(argv[7]), atof(argv[8]));

    printf("=====\n");
    printf("Point Q0: (%.0f, %.0f)\n", Q0.x, Q0.y);
    printf("Point Q1: (%.0f, %.0f)\n", Q1.x, Q1.y);
    printf("Point Q2: (%.0f, %.0f)\n", Q2.x, Q2.y);
}

```

```

printf("=====\n");

Bezier2 B;
B.C0 = Q0;
B.C1 = Q1;
B.C2 = Q2;

for(float i=0; i<=n; i++){
    float t = i/n;
    Point Pi = point_bezier2(B, t);
    ajouter_element_liste_Point(&listeP, Pi);
}

printf("test\n");

Tableau_Point P = sequence_points_liste_vers_tableau(listeP);
Bezier2 fB = approx_bezier2(P, j1, j2);

printf("=====\n");
printf("Point fB0: (%.6f, %.6f)\n", fB.C0.x, fB.C0.y);
printf("Point fB1: (%.6f, %.6f)\n", fB.C1.x, fB.C1.y);
printf("Point fB2: (%.6f, %.6f)\n", fB.C2.x, fB.C2.y);
printf("=====\n");
}

```

Tableaux

Image	d=1	d=3	d=10	d=15
Asterix3 : Contour = 32	Bezier2 = 962	Bezier2 = 295	Bezier2 = 161	Bezier2 = 70
lettre-L-cursive : Contour = 3	Bezier2 = 252	Bezier2 = 42	Bezier2 = 27	Bezier2 = 16
Picasso-ColombesDeLaPaix : Contour = 106	Bezier2 = 1601	Bezier2 = 589	Bezier2 = 306	Bezier2 = 154