

Projet Informatique

Modélisation de la trajectoire d'un point

Rapport de seconde phase

Nous avons, afin de mener le projet à terme, rajouté des fonctions au programme, non prévues au début. Nous avons également supprimé la fonction lire_fichier, celle-ci n'étant utilisée à aucun moment de notre programme. Toutes les fonctions qui ne sont pas citées dans la suite sont les fonctions qui n'ont pas été modifiées par rapport au rapport de première phase.

I. Librairie notation polonaise

Cette librairie contient toutes les fonctions qui permettent de lire et de faire les calculs grâce à une notation polonaise.

1) Fonction conversion_chaine_tableau

La fonction conversion_chaine_tableau prend en entrée l'adresse d'un caractère, l'adresse d'un tableau et l'adresse d'un int.

Elle permet de mettre chaque caractère de la chaîne de caractères dans un tableau en supprimant les espaces.

```
char conversion_chaine_tableau(char*, float*, int*);
```

Elle renvoie le caractère 'e' s'il y a une erreur c'est à dire si l'utilisateur a rentré d'autres caractères que x,y,z,a,b,c,*,+,-,/ et renvoie 't' si tout c'est bien passé.

2) Fonction napi_initialisation

Cette fonction permet à l'utilisateur de rentrer ses équations sous notation polonaise inversée. Elle fait appel à la fonction conversion_chaine_tableau.

Elle prend en entrée trois adresses de tableaux, trois adresses d'entiers où il faudra stocker la taille des tableaux.

```
void napi_initialisation(float*, float*, float*, int*, int*, int*);
```

Elle ne renvoie rien car elle utilise des pointeurs.

3) Fonction remplace_valeur

Cette fonction prend en entrée l'adresse d'un tableau, deux structures de trois float qui correspondent aux coordonnées positions et aux paramètres de la trajectoire.

Cette fonction remplace dans le tableau les lettres par leurs valeurs.

```
void remplace_valeur(float*, tri, tri, int );
```

Cette fonction ne renvoie rien, elle utilise un pointeur sur un tableau.

4) Fonctions depiler et empiler

Ces deux fonctions utilisent le principe des piles elles permettent d'ajouter ou d'enlever une valeur dans un tableau.

La fonction depiler prend en entrée l'adresse d'un tableau et un entier qui correspond à la taille du tableau. Elle renvoie le nombre retiré du tableau.

```
float depiler(float*, int);
```

La fonction empiler prend en entrée l'adresse d'un tableau, la valeur du nombre à ajouter et la taille du tableau. Elle ne renvoie rien.

```
int empiler(float*, float , int );
```

5) Fonction npv

Cette fonction prend en entrée l'adresse d'un tableau et un entier qui correspond à la taille du tableau. Cette fonction permet de calculer la valeur de la vitesse grâce à l'expression donnée par l'utilisateur. Elle utilise la fonction depiler et empiler.

```
float npv(float*, int );
```

Elle renvoie la valeur finale de la vitesse calculée sous forme de float.

6) fonction calcul

Cette fonction permet de calculer les trois valeurs de la vitesse. Elle prend en entrée les coordonnées et les paramètres de la trajectoire. Elle prend aussi l'adresse de trois tableaux et trois entiers qui correspondent à leur taille. Cette fonction utilise les fonctions remplace_valeur et npv.

```
tri calcul(tri, tri , float* , float* , float* , int , int , int );
```

Elle renvoie les trois valeurs de la vitesse sous forme d'un triplets de float.

II. Une fonction simul_npv

```
void simul_npv(FILE* fichier , float tpssimulation, float dt, tri coordposition, tri coordparametre)
```

Elle se charge de faire l'ensemble de la simulation à partir des formules rentrées par l'utilisateur en notation polonaise (elle fait les mises à jour des points et écrit dans le fichier final qui servira à gnuplot). Elle utilise toutes les fonctions liées à la notation polonaise. Elle est ajoutée à la librairie simulation.

III. Librairie gnuplot i, créée par Nicolas Devillard

Utilisée pour une communication directe entre le programme en C et gnuplot, sans avoir besoin de repasser par des lignes de commande dans la console.

On utilisera les fonctions de cette librairie suivantes.

1) Fonction gnuplot_init();

```
gnuplot_ctrl * gnuplot_init();
```

Elle est utilisée pour ouvrir une session gnuplot en renvoyant un pointeur de type `gnuplot_ctrl*` vers cette session pour l'utiliser, de manière un peu similaire à l'ouverture d'un fichier.

2) Fonction gnuplot_cmd(handle, "commande");

```
void gnuplot_cmd(gnuplot_ctrl * handle, char const * cmd, ...);
```

Elle est utilisée pour envoyer directement du texte à gnuplot et modifier des paramètres, tracer des courbes par exemple.

3) Fonction gnuplot_close(g);

```
void gnuplot_close(gnuplot_ctrl * handle);
```

Permet de refermer la session utilisée, afin de libérer de la place en mémoire, comme on ferait avec un fichier.

IV. Architecture du main:

Le fichier `main.c` contient le programme C principal, il sera compilé avec les différentes bibliothèques par le `makefile`.

- Il commence par demander le système à utiliser, et les différents paramètres, et va afficher un récapitulatif.
- Ensuite il initialise un fichier vide du nom de "courbe.dat" et va l'ouvrir.
- Puis il va lancer la simulation pour le système choisi et refermer le fichier lorsque celle-ci est finie.
- Enfin, il va par l'intermédiaire des fonctions de la bibliothèque `gnuplot_i`, ouvrir une session gnuplot et tracer la courbe (avec les vecteurs vitesses si l'utilisateur le souhaite) grâce aux coordonnées contenues dans le fichier "courbe.dat".