UNIVERSITÉ Clermont Auvergne

Compte-rendu : Sokoban

HIAULT Lilian VALLET Baptiste

13 janvier 2020

Table des matières

Introduction

Le projet de programmation avancée à réaliser est le jeu du Sokoban. C'est un jeu dans lequel un personnage doit déplacer des caisses vers des points d'intérêts sans être bloqué par les murs. Nous avons programmé ce jeu en langage C en utilisant la librairie SDL 1.2 pour l'affichage graphique. Pour créer l'executable du jeu entrer : make, puis ./sokoban pour le lancer.

1 Modélisation du plateau

1.1 Modélisation d'un niveau

Les niveaux de jeu sont stockés sous forme de fichiers texte dont la première ligne donne la largeur du niveau et la seconde sa hauteur. Le fichier représente le niveau dont les objets sont représentés par des caractères :

— ' ' pour le sol
— '#' pour les murs
— 'I' pour les points d'intérêts
— 'C' pour les caisses et 'c' si elles sont sur des points d'intérêts
— 'P' pour le personnage et 'p' si il est sur un points d'intérêt

Par exemple ceci est un niveau valide :

```
10
8
##########
#        #
#   I    #
#   C    #
#        #
#   P    #
#        #
##########
```

FIGURE 1 – Niveau valide du jeu du Sokoban

On peut choisir grâce à un switch le niveau à charger :

```
FILE * level = NULL;
if((level = fopen(fileLevel, "r")) == NULL){
  perror("Problème d'ouverture du fichier de niveau");
  exit(EXIT_FAILURE);
}
```

1.2 Création du plateau

On extrait les deux premières lignes du fichier qu'on a ouvert pour connaître la taille du plateau de jeu.

```
fgets(buffer,BUFSIZ,level);
*largeur = strtol(buffer,NULL,10);
fgets(buffer,BUFSIZ,level);
*hauteur = strtol(buffer,NULL,10);
```

fgets permet de lire la ligne et d'avancer le descripteur à la ligne suivante puis on utilise strtol sur la chaîne de caractère extraite afin de la convertir en entier.

Le plateau de jeu est un tableau de caractères en deux dimensions

```
char ** TJeu = createArr2d(largeur, hauteur);
```

1.3 Remplir le plateau de jeu

On remplit le tableau avec les caractères du fichier du niveau :

```
for(int j =0; j< (*hauteur);j++){
  fgets(buffer,BUFSIZ,level);
    for(i=0;i< (*largeur);i++){
      TJeu[i][j]=buffer[i];
    }
  }
}
```

Le fichier est lu ligne par ligne puis les caractères sont copiés dans le tableau.

2 Fonctionnement du jeu

2.1 Trouver le personnage

Au début de la boucle principale on effectue une recherche dans le plateau de jeu pour trouver le personnage qu'il soit sur une case normale ou un point d'intérêt.

```
void trouvePerso(char ** TJeu,int hauteur,int largeur,pos * perso,char* caseInit){
  perso->x=0;
  perso->y=0;
  int i,j;
  for(i = 0; i < largeur; i++)
  {
    for(j = 0; j < hauteur; j++) {
      if(TJeu[i][j]=='p' || TJeu[i][j]=='P'){
         perso->x = j;
         perso->y = i;
        }
      }
    }
  caseInit=&TJeu[perso->y][perso->x];
  }
}
```

2.2 Pause

C'est unfonction qui met en pause le jeu jusqu'à une entrée du clavier, et qui renvoie la touche enfoncée.

```
  SDL_Event pause(){
  int cont=1;
  SDL_Event event;
  while( cont){
    SDL_WaitEvent(&event);
    switch(event.type){
    case SDL_QUIT:
      cont=0;
      break;
    case SDL_KEYDOWN:
      cont=0;
      break;
    default:
      cont=1;
    }
  }
  return(event);
}
```

2.3 Déplacement du personnage

La fonction deplacement() est centrale, c'est la boucle de jeu dans laquelle les saisies clavier sont enregistrées pour pour qu'il puisse y avoir un déplacement du personnage et qu'il soit affiché.

```
void deplacement(char ** TJeu, int hauteur, int largeur);
```

La variable cont restera vraie tant que l'utilisateur n'appuie pas sur 'q' ou ne reçoive le signal 'SDL_QUIT' et tant qu'elle est vraie, la boucle de jeu continue de tourner.

```
    event= pause();
    switch(event.type){
    ...
```

Ici, on utilise un switch pour trouver sur quelle touche l'utilisateur a appuyé. Puis s'il a appuyé sur une touche de direction déplace le personnage.

2.4 Bouge

```
void bouge(char** TJeu,char objet,int hauteur,int largeur, pos * posPrev, pos * posSuiv);
```

bouge() permet à partir de deux points (celui où est le personnage et sa destination) de verifier s'il est possible de se déplacer selon la présence d'obstacles et effectue ensuite le mouvement.

La fonction effectue des tests sur la position d'arrivée afin de déterminer la nature de la case.

— si c'est au-delà de la bordure du niveau ou que c'est un mur alors il ne peut pas se déplacer
— si c'est une caisse alors on vérifie une case plus loin afin de savoir s'il peut la déplacer
— sinon c'est soit un point d'intérêt soit une case vide donc le personnage peut y aller

Puis on applique le changement de position de joueur et/ou de la caisse en modifiant le contenu du plateau de jeu.

2.5 Victoire

La fonction victoire() verifie s'il existe encore des points d'intérêt non recouverts par des caisses. Elle parcours le tableau à la recherche de 'I' (points d'intérêts sans caisse dessus) ou 'p' (personnage sur un point d'intérêt).

```
 void victoire(char** TJeu, int hauteur,int largeur, int* cont){
  int i=0;
  int j=0;
  int reste = 1;
  *cont=0;
  while(i<largeur && reste){
    while(j<hauteur && reste){
      if((TJeu[i][j]=='I'|| TJeu[i][j]=='p') && reste){
reste=0;
*cont=1;
      }
      j++;
    }
    j=0;
    i++;
  }
}
```

3 Affichage graphique

3.1 Afficher la fenêtre

On défini dans le main() la taille de la fenêtre en fonction de la hauteur et largeur du niveau de manière a avoir 32 pixels par objet du niveau. On initialise alors la fenêtre et on charge les différentes images :

```
  if(SDL_Init(SDL_INIT_VIDEO) == 1){
    fprintf(stderr,"Erreur SDL : %s\n",SDL_GetError());
    return -1;
  }
  if(( ecran = SDL_SetVideoMode(WIDTH,HEIGHT,32,SDL_HWSURFACE)) == NULL){
    fprintf(stderr,"Erreur VideoMode : %s\n",SDL_GetError());
    exit(EXIT_FAILURE);}

  SDL_FillRect(ecran,NULL,SDL_MapRGB(ecran->format,150,200,175));
  SDL_Flip(ecran);
  OFF =  SDL_LoadBMP("interrupteurOff.bmp");
  ON =  SDL_LoadBMP("interrupteurOn.bmp");
  Begin = SDL_LoadBMP("Begin.bmp");
  Brique = SDL_LoadBMP("Bloc.bmp");
  Michel = SDL_LoadBMP("Michel2.bmp");
  MichelPoint = SDL_LoadBMP("MichelPoint.bmp");
  Point = SDL_LoadBMP("Point.bmp");
```

```
SDL_WM_SetCaption("SokobanBV",NULL);
SDL_Flip(ecran);
```

3.2 Afficher le niveau

dessine() permet d'afficher une surface à l'écran

dessineTJeu() est une fonction qui affiche le plateau de jeu a partir d'un tableau en deux dimensions de caractères. Elle colore la surface pour la remettre a zéro, puis associe a chaque case une image. Elle lit le plateau en mémoire et selon le caractère dans la case, elle appelle dessine() pour afficher l'image correspondant à l'objet.

Conclusion

Grâce à ce projet nous avons pu améliorer nos connaissances surtout au niveau de l'affichage graphique. Outre l'affichage il était intéressant de travailler à plusieurs surtout lorsqu'on a des idées différentes pour résoudre un problème. Enfin cela nous a permis de nous poser des questions sur comment modéliser le jeu ou alors quel est le lien entre ce qu'on voit et ce qui est programmé.