/\* AniTool.h \*/

#ifndef \_ANI\_TOOL\_

#define \_ANI\_TOOL\_

/\* 헤더 파일 \*/

#include <stdio.h>

#include <fcntl.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <stdlib.h>

#include <termios.h>

#include <time.h>

#include <string.h>

#include <dirent.h>

#include <signal.h>

/\* MAP 상수 \*/

#define Y\_START 10 // 화면에서 맵이 시작하는 지점 ( 테두리 제외 )

#define X\_START 10

#define Y\_SIZE 30 // 맵 크기( 테두리 제외 )

#define X\_SIZE 60

#define OUT 1 // Excute할때 커서를 치워두는 위치

/\* SIZE \*/

#define S\_BUFSIZE 64

#define F\_BUFSIZE (X\_SIZE\*Y\_SIZE\*SIZEOFINT+SIZEOFINT\*2) // for File I/O

#define VERTICAL (49) // 한 화면의 길이

#define HORIZON 93

#define XYS (X\_SIZE\*Y\_SIZE\*sizeof(int))

#define SIZEOFINT 4

/\* OPENING MENU \*/

#define OPENING\_MIN 1

#define OPENING\_MAX 3

#define BANNER 3 // Number of BANNER Line

/\* MAP INTERFACE \*/

// MAP Tool 오른쪽의 인터페이스 관련 정보

#define MENU\_X (X\_START+X\_SIZE)

#define MENU\_ONE\_Y (Y\_START) // Save

#define MENU\_TWO\_Y (Y\_START+1) // Add

#define MENU\_THREE\_Y (Y\_START+2) // Copy

#define MENU\_FOUR\_Y (Y\_START+3) // Erase

#define MENU\_FIVE\_Y (Y\_START+4) // Delete

#define MENU\_SIX\_Y (Y\_START+5) // Load

#define MENU\_SEVEN\_Y (Y\_START+6) // Quit

#define MENU\_EIGHT\_Y (Y\_START+7) // Reset

#define MENU\_END\_Y MENU\_EIGHT\_Y

#define MENU\_TAIL\_Y (Y\_START+Y\_SIZE-1)

#define COMMAND\_LINE (Y\_START+Y\_SIZE+3)

/\* ASKII CODE \*/

#define UP 4283163

#define DOWN 4348699

#define LEFT 4479771

#define RIGHT 4414235

#define SBLEFT 91 // '['

#define SBRIGHT 93 // ']'

#define SPACE 32

#define ENTER 10

#define Y\_KEY 121

#define N\_KEY 110

/\* FILE NAME \*/

#define SAVEDIR "save\_ani"

/\* Initial Value \*/

#define INITIAL\_SHOWTIME 1

/\* BOOL 상수 \*/

#define TRUE 1

#define FALSE 0

/\* 전역 변수 \*/

extern char filewd[S\_BUFSIZE]; // 프로그램의 원래 디렉터리 저장

extern struct termios default\_tm;

/\* Type 정의 \*/

typedef enum { false = 0, true } bool;

typedef enum { off = 1, on, cur\_on, cur\_off } shape\_t;

typedef enum { \_default, \_red, \_yellow, \_green, \_blue } color\_t;

typedef struct \_Point {

int x;

int y;

} Point;

typedef struct \_Map {

shape\_t shapeOfMap[Y\_SIZE][X\_SIZE];

} Map;

typedef struct \_AniList {

int number; // 순서

Map MapData; // Data

clock\_t showTime; // ms

struct \_AniList\* Left; // 연결 고리

struct \_AniList\* Right;

} AniList;

typedef AniList\* Ptr\_AniList; /\* Header 포인터가 이중포인터가 되면 보기 어렵기 때문에

따로 정의\*/

union Askii {

int int\_buf;

char char\_buf[4];

};

/\* Thread 선언( AniTool.c에 정의 ) \*/

void \*thr\_NewFile(void\* arg);

void \*thr\_OpenFile(void\* arg);

void \*thr\_ExcuteFile(void\* arg);

void \*thr\_Kbhit();

// thr\_ExcuteFile에 종속

/\* SIGNAL 처리 함수 선언( AniTool.c에 정의 ) \*/

void sig\_int(int signo);

/\* 함수 원형화 선언 \*/

/\* 리스트 관련 \*/

void init\_AniList(AniList\* List, Ptr\_AniList\* pHeader);// 첫 번째 List를 초기화

AniList\* add\_AniList(AniList\* pCurrent, Ptr\_AniList const Header);

// List의 뒤에 새로운 AniList 생성, return - 새로 생성된 AniList\*

AniList\* copy\_AniList(AniList\* pCurrent, Ptr\_AniList const Header);

AniList\* delete\_AniList(AniList\* List, Ptr\_AniList\* pHeader);

/\* pCurrent에는 pCurrent이전 혹은 pCurrent 자신을 집어넣고,

Header는여전히 맨 앞의 값을 가리키도록함 \*/

AniList\* delete\_All(Ptr\_AniList \*pHeader);

void fill\_Number(Ptr\_AniList const Header);

AniList\* free\_AllList(Ptr\_AniList\* pHeader); // freeAllList, pHeader <- NULL

// 반환 - NULL

bool is\_Empty(const Ptr\_AniList Header); // 반환 - 비어있으면 true

/\* 초기화 \*/

void init\_Map(Map\* MapData);

void init\_Cur(Point\* Cur);

/\* 터미널 제어 \*/

void gotoxy(const int x, const int y); // 관용적으로 쓰이는 이름

void gotoUnder(const int num);

void gotoRight(const int num);

void gotoLeft(const int num);

void hide\_Cur();

// 커서 숨기기

void reveal\_Cur();

// 커서 드러내기

void default\_Cur();

// 커서 기본 색 설정

void set\_Color(const color\_t color);

// 글씨 색, arg: \_blue, \_red, \_yellow, \_default, \_green

void clear();

void erase\_Line(const int line);

void erase\_Back(const int fromleft);

// 현재 커서전부터 fromleft만큼 지우고 커서를 그곳에 둠

/\* 출력 \*/

void screen\_Tool(const AniList List, const Point Cur); // 모든 print함수를 포함

AniList\* screen\_Excute(const AniList List, const Ptr\_AniList Header, const Point Cur);

// 반환: 다음 node의 포언터

void print\_Map(const AniList List, const Point Cur);

void print\_a\_Line(const Map MapData, const Point Cur, const int y);

// printMap에 종속

void print\_Menu(const AniList List, const Point Cur);

void show\_AtCmdLine(const char\* commandLine);

void print\_Menual();

shape\_t get\_Shape(const Map MapData, const Point Cur, const int x, const int y);

/\* 조작 \*/

int kbhit();

void set\_default\_tm();

void up\_Tool(Point\* Cur);

void down\_Tool(Point\* Cur);

void left\_Tool(Point\* Cur);

void right\_Tool(Point\* Cur);

AniList\* space\_Tool(AniList\* pCurrent, Ptr\_AniList \*pHeader, Point\* Cur);

// 반환 - pCurrent

AniList\* control\_Tool(AniList\* pCurrent, Ptr\_AniList \*pHeader, Point\* Cur);

// 실수로 control\_Tool만 AniTool\_F.c에서 밑에다 정의함

void press\_Key(); // 그냥 키 입력받을때까지 대기

/\* 기능 - tool \*/

void file\_List(const char\* dirname);

// ls와 비슷하지만 조금 더 기능이 확장 됨

int check\_File(const char\* filename, const char\* dirname);

// 디렉터리 내에 같은파일이 있으면 0, 없으면 1을 반환

void default\_wd();

// 기본 디렉터리로 working directory 변경, atexit에 등록

void save(const Ptr\_AniList Header);

AniList\* load(AniList\* pCurrent, Ptr\_AniList\* pHeader);

// 반환 - 항상 pCurrent,

/\* opening \*/

void scan(char string[]);

void print\_Opening();

void print\_OpeningCur(const short int opening\_cur);

void cntl\_Opening(short int\* opening\_cur, pthread\_t\* nthread);

void up\_Opening(short int\* opening\_cur);

void down\_Opening(short int\* opening\_cur);

void space\_Opening(short int opening\_cur, pthread\_t\* nthread);

/\* 기타 \*/

bool isAbleToSaveLoad(const char\* filename);

// 입출력 할 수 있는 파일인지 체크, 예외) 파일이 없으면 true

#else

#endif