#include <stdio.h>

#include <stdlib.h>

#include "math.h"

#include "boolean.h"

#include "gameproperty.h"

#include "mesinkar.h"

#include "mesinkata.h"

#include "point.h"

#include "map.h"

#include "pcolor.h"

#include "stackunit.h"

#include "turnqueue.h"

#include <time.h>

#include "gamefunction.h"

#include "loadkar.h"

#include "loadkata.h"

void Credits() {

printf("\nKelompok Seksi Gorengan\n");

printf("Aldo Azali........13516125\n");

printf("Ivan Jonathan.....13516059\n");

printf("Shevalda..........13516\n");

printf("Seperayo..........13516068\n");

printf("Hafizh Budiman....13516137\n\n");

printf("Dosen : Bu Yani\n");

char junk;

scanf("%c", &junk);

printf("\n");

}

void EmptyToEksternal() {

FILE \* pita;

pita = fopen("input.txt", "w");

putc('.', pita);

fclose(pita);

}

void SaveGame(Player P1, Player P2, addressunit X, Unit CurrUnit1, Unit CurrUnit2, ListUnit LU1, ListUnit LU2, Stack S1, Stack S2, TurnQueue Q, MATRIKS M) {

FILE \* save;

char spasi = ' ';

char enter = '\n';

int temp;

save = fopen("load.txt", "w");

// MAP

temp = NBrsEff(M);

putc(temp, save);

putc(spasi, save);

temp = NKolEff(M);

putc(temp, save);

putc(enter, save);

// PLAYER

putc('.', save);

fclose(save);

}

void LoadGame() {}

void InputToEksternal(){

FILE \* pita;

char ch;

pita = fopen("input.txt","w");

while((ch = getchar())!='\n'){

if(!(ch=='1') && !(ch=='2') && !(ch=='3') && !(ch=='4') && !(ch=='5') && !(ch=='6') && !(ch=='7') & !(ch=='8') && !(ch=='9') && !(ch=='0')){

ch&='\_';

}

putc(ch,pita);

}

putc('.',pita);

fclose(pita);

}

void EksternalToInput(Kata \*pilihan){

STARTKATA();

\*pilihan = CKata;

}

boolean IsKataSama(Kata K1, Kata K2){

int i;

boolean bool=false;

if(K1.Length==0 && K2.Length==0){

bool=true;

}

else if(K1.Length==K2.Length){

bool=true;

i=1;

while(i<=K1.Length && bool){

if(K1.TabKata[i]!=K2.TabKata[i]){

bool=false;

}

i++;

}

}

return bool;

}

void Display(){

printf("\nWelcome to Battle For Olympia !!\n\n");

}

void Menu(int \*option) {

Kata pilihan;

Kata start, load, exit, satu, dua, tiga;

Kata credit, empat;

char junk;

//BAGIAN ASSIGN STRING PILIHAN

start.TabKata[1] = 'S';

start.TabKata[2] = 'T';

start.TabKata[3] = 'A';

start.TabKata[4] = 'R';

start.TabKata[5] = 'T';

start.Length = 5;

load.TabKata[1] = 'L';

load.TabKata[2] = 'O';

load.TabKata[3] = 'A';

load.TabKata[4] = 'D';

load.Length = 4;

exit.TabKata[1] = 'E';

exit.TabKata[2] = 'X';

exit.TabKata[3] = 'I';

exit.TabKata[4] = 'T';

exit.Length = 4;

credit.TabKata[1] = 'C';

credit.TabKata[2] = 'R';

credit.TabKata[3] = 'E';

credit.TabKata[4] = 'D';

credit.TabKata[5] = 'I';

credit.TabKata[6] = 'T';

credit.Length = 6;

satu.TabKata[1] = '1';

satu.Length = 1;

dua.TabKata[1] = '2';

dua.Length = 1;

tiga.TabKata[1] = '3';

tiga.Length = 1;

empat.TabKata[1] = '4';

empat.Length = 1;

scanf("%c", &junk); // Membuang karakter enter berlebihan dari user agar Input menu tidak error

//BAGIAN MENU PROGRAM

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\* START........[1] \*\n\* LOAD.........[2] \*\n\* CREDIT.......[3] \*\n\* EXIT.........[4] \*\n"); //Menu Pilihan

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\nGame Command : ");

InputToEksternal();

EksternalToInput(&pilihan);

if (IsKataSama(start, pilihan) || IsKataSama(satu, pilihan)) {

printf("\nStarting Game\n\n");

\*option = 1;

}

else if (IsKataSama(load, pilihan) || IsKataSama(dua, pilihan)) {

printf("\nLoading Previous Game\n\n");

\*option = 2;

}

else if (IsKataSama(credit, pilihan) || IsKataSama(tiga, pilihan)) {

\*option = 3;

}

else if (IsKataSama(exit, pilihan) || IsKataSama(empat, pilihan)) {

\*option = 0;

}

else {

printf("Input ERROR !!!\n");

scanf("%c", &junk);

printf("\n\n");

Menu(option);

}

}

void NewGame(Player \*P1, Player \*P2, addressunit \*X, Unit \*CurrUnit1, Unit \*CurrUnit2, ListUnit \*LU1, ListUnit \*LU2, Stack \*S1, Stack \*S2, TurnQueue \*Q, MATRIKS \*M) {

int B, K; //Baris , Kolom, Current Turn

char junk;

printf("\* Insert Map Row Size = ");

scanf("%d", &B);

scanf("%c", &junk); // Membuang karakter enter berlebihan dari user agar Input menu tidak error

printf("\* Insert Map Column Size = ");

scanf("%d", &K);

scanf("%c", &junk); // Membuang karakter enter berlebihan dari user agar Input menu tidak error

MakeMATRIKS(B, K, &(\*M));

InitializeMap(&(\*M));

InitializeQueue(&(\*P1), &(\*P2), &(\*M), &(\*Q));

CreateEmptyS(S1);

CreateEmptyS(S2);

}

void checkWinner(ListUnit P1UnitList, ListUnit P2UnitList) {

if (IsEmpty(P1UnitList)) {

Dealokasi(&(FirstUnit(P2UnitList)));

printf("\n Hooraayy!! Player 2 wins the game..\nCongratulations Player 2!.");

}

else {

Dealokasi(&(FirstUnit(P1UnitList)));

printf("\n Hooraayy!! Player 1 wins the game..\nCongratulations Player 1!.");

}

}

void StartGame(Player P1, Player P2, addressunit X, Unit CurrUnit1, Unit CurrUnit2, ListUnit LU1, ListUnit LU2, Stack S1, Stack S2, TurnQueue Q, MATRIKS M) {

Kata save, menu, move, undo, recruit, change\_unit, attack, map, info, end\_turn;

//BAGIAN ASSIGN STRING PILIHAN

save.TabKata[1] = 'S';

save.TabKata[2] = 'A';

save.TabKata[3] = 'V';

save.TabKata[4] = 'E';

save.Length = 4;

move.TabKata[1] = 'M';

move.TabKata[2] = 'O';

move.TabKata[3] = 'V';

move.TabKata[4] = 'E';

move.Length = 4;

undo.TabKata[1] = 'U';

undo.TabKata[2] = 'N';

undo.TabKata[3] = 'D';

undo.TabKata[4] = 'O';

undo.Length = 4;

recruit.TabKata[1] = 'R';

recruit.TabKata[2] = 'E';

recruit.TabKata[3] = 'C';

recruit.TabKata[4] = 'R';

recruit.TabKata[5] = 'U';

recruit.TabKata[6] = 'I';

recruit.TabKata[7] = 'T';

recruit.Length = 7;

change\_unit.TabKata[1] = 'C';

change\_unit.TabKata[2] = 'H';

change\_unit.TabKata[3] = 'A';

change\_unit.TabKata[4] = 'N';

change\_unit.TabKata[5] = 'G';

change\_unit.TabKata[6] = 'E';

change\_unit.TabKata[7] = '\_';

change\_unit.TabKata[8] = 'U';

change\_unit.TabKata[9] = 'N';

change\_unit.TabKata[10] = 'I';

change\_unit.TabKata[11] = 'T';

change\_unit.Length = 11;

attack.TabKata[1] = 'A';

attack.TabKata[2] = 'T';

attack.TabKata[3] = 'T';

attack.TabKata[4] = 'A';

attack.TabKata[5] = 'C';

attack.TabKata[6] = 'K';

attack.Length = 6;

map.TabKata[1] = 'M';

map.TabKata[2] = 'A';

map.TabKata[3] = 'P';

map.Length = 3;

info.TabKata[1] = 'I';

info.TabKata[2] = 'N';

info.TabKata[3] = 'F';

info.TabKata[4] = 'O';

info.Length = 4;

end\_turn.TabKata[1] = 'E';

end\_turn.TabKata[2] = 'N';

end\_turn.TabKata[3] = 'D';

end\_turn.TabKata[4] = '\_';

end\_turn.TabKata[5] = 'T';

end\_turn.TabKata[6] = 'U';

end\_turn.TabKata[7] = 'R';

end\_turn.TabKata[8] = 'N';

end\_turn.Length = 8;

menu.TabKata[1] = 'M';

menu.TabKata[2] = 'E';

menu.TabKata[3] = 'N';

menu.TabKata[4] = 'U';

menu.Length = 4;

Kata pilihan;

int CTurn;

CurrUnit1 = InfoUnit(FirstUnit(P1.UNTLST));

CurrUnit2 = InfoUnit(FirstUnit(P2.UNTLST));

Push(&S1,CurrUnit1);

Push(&S2,CurrUnit2);

while (!IsKataSama(menu, pilihan)) {

while (!IsKataSama(end\_turn, pilihan)) {

CTurn = InfoHead(Q).ID;

if (CTurn == 1) {

printf("\nPlayer %d's Turn\n", P1.ID);

printf("Cash : "); printf("%d\n", P1.GLD);

printf("Income : "); printf("%d\n", P1.INC);

printf("Upkeep : "); printf("%d\n", P1.OUT);

printf("Unit : ");

if (CurrUnit1.TYP == 'K') {

printf("King \n");

}

else if (CurrUnit1.TYP == 'A') {

printf("Archer \n");

}

else if (CurrUnit1.TYP == 'S') {

printf("Swordsman \n");

}

else if (CurrUnit1.TYP == 'W') {

printf("White Mage \n");

}

printf("Health : "); printf("%d\n", CurrUnit1.HP);

printf("Movement Point : "); printf("%d\n", CurrUnit1.MOVEPTS);

printf("Can Attack? ");

if (CurrUnit1.CHN) {

printf("Yes\n");

}

else {

printf("No\n");

}

}

else if (CTurn == 2) {

printf("\nPlayer %d's Turn\n", P2.ID);

printf("Cash : "); printf("%d\n", P2.GLD);

printf("Income : "); printf("%d\n", P2.INC);

printf("Upkeep : "); printf("%d\n", P2.OUT);

printf("Unit : ");

if (CurrUnit2.TYP == 'K') {

printf("King \n");

}

else if (CurrUnit2.TYP == 'A') {

printf("Archer \n");

}

else if (CurrUnit2.TYP == 'S') {

printf("Swordsman \n");

}

else if (CurrUnit2.TYP == 'W') {

printf("White Mage \n");

}

printf("Health : "); printf("%d\n", CurrUnit2.HP);

printf("Movement Point : "); printf("%d\n", CurrUnit2.MOVEPTS);

printf("Can Attack? ");

if (CurrUnit2.CHN) {

printf("Yes\n");

}

else {

printf("No\n");

}

}

printf("\nGame Command : ");

InputToEksternal();

EksternalToInput(&pilihan);

if (IsKataSama(move, pilihan)) {

if (CTurn==1) {

if (CurrUnit1.MOVEPTS > 0) {

TulisMAPJarak(&M,CurrUnit1);

fmove(&P1,&M,&Q,&CurrUnit1);

Push(&S1,CurrUnit1);

} else {

printf("You ran out of movement points\n");

}

} else if (CTurn==2) {

if (CurrUnit2.MOVEPTS > 0) {

TulisMAPJarak(&M,CurrUnit2);

fmove(&P2,&M,&Q,&CurrUnit2);

Push(&S2,CurrUnit2);

} else {

printf("You ran out of movement points\n");

}

}

}

else if (IsKataSama(undo, pilihan)) {

if (CTurn==1) {

if (IsOneS(S1)) {

printf("No previous move, unable to undo.");

} else {

printf("You have succesfully undo the move.");

Pop(&S1,&M,&CurrUnit1);

}

} else if (CTurn==2) {

if (IsOneS(S2)) {

printf("No previous move, unable to undo.");

} else {

printf("You have succesfully undo the move.");

Pop(&S2,&M,&CurrUnit2);

}

}

}

else if (IsKataSama(recruit, pilihan)) {

printf("\n");

frecruit(&P1, &P2, &M, &Q);

}

else if (IsKataSama(change\_unit, pilihan)) {

if (CTurn == 1) {

CurrUnit1 = fchange\_unit(P1);

DelP(&(P1.UNTLST), CurrUnit1);

InsVFirst(&P1.UNTLST, CurrUnit1);

}

else if (CTurn == 2) {

CurrUnit2 = fchange\_unit(P2);

DelP(&(P2.UNTLST), CurrUnit2);

InsVFirst(&P2.UNTLST, CurrUnit2);

}

}

else if (IsKataSama(attack, pilihan)) {

fattack(&P1, &P2, &M, &Q);

CurrUnit1 = InfoUnit(FirstUnit(P1.UNTLST));

CurrUnit2 = InfoUnit(FirstUnit(P2.UNTLST));

}

else if (IsKataSama(map, pilihan)) {

if (CTurn == 1) {

int X1 = CurrUnit1.LOC.X;

int Y1 = CurrUnit1.LOC.Y;

TulisMAP(&M, X1, Y1);

}

else if (CTurn == 2) {

int X2 = CurrUnit2.LOC.X;

int Y2 = CurrUnit2.LOC.Y;

TulisMAP(&M, X2, Y2);

}

}

else if (IsKataSama(info, pilihan)) {

printf("\n");

finfo(M);

}

else if (IsKataSama(save, pilihan)) {

printf("\n");

SaveGame(P1, P2, X, CurrUnit1, CurrUnit2, LU1, LU2, S1, S2, Q, M);

}

else if (IsKataSama(menu, pilihan)) {

printf("\n");

} else {

printf("\nWarning : Masukan anda salah! Ulangi!\n");

}

if (IsKataSama(end\_turn, pilihan)) {

printf("\n");

CurrUnit1.CHN = true;

CurrUnit2.CHN = true;

InfoUnit(FirstUnit(P1.UNTLST)).CHN = CurrUnit1.CHN;

InfoUnit(FirstUnit(P2.UNTLST)).CHN = CurrUnit2.CHN;

RefreshMovePTS(&P1);

RefreshMovePTS(&P2);

CurrUnit1.MOVEPTS = 2;

CurrUnit2.MOVEPTS = 2;

ChangeTurn(&Q);

EmptyToEksternal();

EksternalToInput(&pilihan);

}

if (IsEmpty(P1.UNTLST) || IsEmpty(P2.UNTLST)) {

pilihan = end\_turn;

}

}

if (IsEmpty(P1.UNTLST) || IsEmpty(P2.UNTLST)) {

pilihan = menu;

}

}

checkWinner(P1.UNTLST, P2.UNTLST);

}

int main(){

int option; //Variabel untuk Menu

/\*Variabel untuk start game \*/

Player P1, P2;

addressunit X;

Unit CurrUnit1, CurrUnit2;

ListUnit LU1, LU2;

Stack S1,S2;

TurnQueue Q;

MATRIKS M;

char junk;

do {

Display();

Menu(&option);

if (option == 1) {

NewGame(&P1, &P2, &X, &CurrUnit1, &CurrUnit2, &LU1, &LU2, &S1, &S2, &Q, &M);

StartGame(P1, P2, X, CurrUnit1, CurrUnit2, LU1, LU2, S1, S2, Q, M);

}

else if (option == 2) {

LoadGame();

scanf("%c", &junk);

}

else if (option == 3) {

Credits();

}

else {

option = 0;

}

} while (option != 0);

printf("Exiting the Game\n");

}