#include <stdio.h>

#include <stdlib.h>

#include <SDL2/SDL.h>

#include <SDL2/SDL\_image.h>

#include <math.h>

void first\_components\_of\_the\_background (SDL\_Renderer\* renderer);

void carre\_terre(SDL\_Renderer \*ren, int x, int y);

void mvt\_tank(SDL\_Surface \*Tank\_i, SDL\_Renderer \*ren, int x, int y );

void fond\_terre(SDL\_Renderer\* renderer); // updates

void directions\_panel(SDL\_Renderer \*ren, int x, int y);

void small\_direction (SDL\_Renderer \*ren, int x, int y, int dd);

int choix\_direction(int mx, int my);

int main(int argc, char \*argv[]){

SDL\_Window\* window = NULL;

SDL\_Renderer\* renderer = NULL;

SDL\_Surface\* Tank = NULL;

SDL\_Event event;

int ON = 1;

//Initialisation

//Ajouter la gestion de l'erreur

SDL\_Init(SDL\_INIT\_VIDEO);

IMG\_Init(IMG\_INIT\_PNG);

//Génération de la fenêtre

window = SDL\_CreateWindow("TankRider : Level 00", SDL\_WINDOWPOS\_CENTERED, SDL\_WINDOWPOS\_CENTERED, 900, 720, SDL\_WINDOW\_SHOWN);

renderer = SDL\_CreateRenderer( window, 0, SDL\_RENDERER\_TARGETTEXTURE);

first\_components\_of\_the\_background (renderer);

fond\_terre(renderer);

//Trouver une façon plus elegante permettant de charger les surfaces

SDL\_Surface \*t\_1 = IMG\_Load("IMG/tank1.png");

SDL\_Surface \*t\_2 = IMG\_Load("IMG/tank2.png");

SDL\_Surface \*t\_3 = IMG\_Load("IMG/tank3.png");

SDL\_Surface \*t\_4 = IMG\_Load("IMG/tank4.png");

SDL\_Surface \*t\_5 = NULL;

SDL\_Surface \*t\_6 = IMG\_Load("IMG/tank6.png");

SDL\_Surface \*t\_7 = IMG\_Load("IMG/tank7.png");

SDL\_Surface \*t\_8 = IMG\_Load("IMG/tank8.png");

SDL\_Surface \*t\_9 = IMG\_Load("IMG/tank9.png");

SDL\_Surface\* Tanks[9]= {t\_1,t\_2,t\_3,t\_4,t\_5,t\_6,t\_7,t\_8,t\_9};

//Coord du premier Tank

int x = 35;

int y = 35;

//Orientation du premier Tank

Tank = t\_6;

//Affichage du Premier tank

mvt\_tank(Tank,renderer,x,y);

//Initialisation de la position du curseur

int MouseX = 0;

int MouseY = 0;

//Ces 3 tab enregistrentles coords et le bouton qui est down

//36 est un nombre quelconque, à redefinir en fonction des courte

int xcible[36];

int ycible[36];

int d\_on\_tab[36];

//Nombre de fleche posée

int nb=0;

//Element de la boucle de dessin

int nbj;

//indice de la fleche selected

int d\_on = 0;

//Debut du jeu

int action =0;

//Position intial

int set = 6;

//tableau de direction

int dx[9]={-5,0,5,-5,0,5,-5,0,5};

int dy[9]={5,5,5,0,0,0,-5,-5,-5};

while( ON == 1 ) {

SDL\_WaitEvent(&event);

switch( event.type){

//Gestion de la fermeture

case SDL\_QUIT :

ON = 0;

break;

case SDL\_MOUSEMOTION :

MouseX=event.motion.x;

MouseY=event.motion.y;

break;

case SDL\_MOUSEBUTTONDOWN:

//detection de la fleche selected

d\_on = choix\_direction(MouseX,MouseY);

break;

case SDL\_MOUSEBUTTONUP:

//#### AJOUTER LA RESTRICTION DE LA POSITION DE LA FLECHE ########

if (event.button.button == SDL\_BUTTON\_LEFT){

//Placement dans "la grille"

//Pas encore dessinée

if(MouseX<735 && MouseX>35 && MouseY< 735 && MouseY>35 ){

xcible[nb]= 35+50\*((MouseX/50)-1);

ycible[nb]= 35+50\*((MouseY/50)-1);

d\_on\_tab[nb]=d\_on;

small\_direction (renderer,xcible[nb],ycible[nb],d\_on\_tab[nb]);

nb++;

}

}

if (event.button.button == SDL\_BUTTON\_LEFT){

if(MouseX>731 && MouseX<879 && MouseY>601 && MouseY<649){

action = 1;

}

}

break;

}

//}

while (action==1){

fond\_terre(renderer);

Tank = Tanks[set-1];

int i;

for(i=0;i<10;i++){

fond\_terre(renderer);

for(nbj = 0; nbj<nb;nbj++){

if(nb!=0){

small\_direction (renderer,xcible[nbj],ycible[nbj],d\_on\_tab[nbj]);

}

}

x=x+dx[set-1];

y=y+dy[set-1];

mvt\_tank(Tank,renderer, x ,y);

if((x==635 || x==25 || y==635 || y==)){ //borne a verifier

action=0;

SDL\_ShowSimpleMessageBox(0, "out", "vous etes hors champs ", window);

}

}

for(nbj=0;nbj<nb;nbj++){

if(x==xcible[nbj] && y==ycible[nbj]){

set= d\_on\_tab[nbj];

}

}

}

}

SDL\_FreeSurface(Tank); //tableau de direction

SDL\_DestroyRenderer(renderer);

SDL\_DestroyWindow(window);

SDL\_Quit();

return EXIT\_SUCCESS;

}

void first\_components\_of\_the\_background (SDL\_Renderer\* renderer ){

//coloration de toutes la fenetes en blanc

SDL\_SetRenderDrawColor( renderer, 255, 255, 255, 0 );

SDL\_Rect rect\_BaseBackground; rect\_BaseBackground.x = 0; rect\_BaseBackground.y = 0; rect\_BaseBackground.w = 900; rect\_BaseBackground.h = 720;

SDL\_RenderFillRect( renderer, &rect\_BaseBackground );

SDL\_RenderPresent(renderer);

SDL\_SetRenderDrawColor( renderer, 120, 165, 255, 0 );

SDL\_Rect rect\_G; rect\_G.x = 10; rect\_G.y = 10; rect\_G.w = 700; rect\_G.h = 700;

SDL\_RenderFillRect( renderer, &rect\_G );

SDL\_RenderPresent(renderer);

SDL\_SetRenderDrawColor( renderer, 213, 196, 128, 0);

SDL\_Rect rect\_D; rect\_D.x = 720; rect\_D.y = 10; rect\_D.w = 170; rect\_D.h = 700;

SDL\_RenderFillRect( renderer, &rect\_D );

SDL\_RenderPresent(renderer);

//Placement du bouton start

SDL\_Texture \*texture\_start = NULL;

SDL\_Surface \*start = IMG\_Load("IMG/start.png");

texture\_start = SDL\_CreateTextureFromSurface(renderer, start);

SDL\_FreeSurface(start);

SDL\_Rect dst\_start;

dst\_start.x = 730;

dst\_start.y = 600;

SDL\_QueryTexture(texture\_start, NULL, NULL, &dst\_start.w, &dst\_start.h);

SDL\_RenderCopy(renderer, texture\_start, NULL, &dst\_start);

SDL\_RenderPresent(renderer);

//Placement du direction\_pannel

//En changeant les coordonnées(730,80), il faut updater ##choix\_direction

SDL\_Texture \*texture\_directions = NULL;

SDL\_Surface \*Directions = IMG\_Load("IMG/Directions.png");

texture\_directions = SDL\_CreateTextureFromSurface(renderer,Directions);

SDL\_FreeSurface(Directions);

SDL\_Rect dst\_directions;

dst\_directions.x = 730;

dst\_directions.y = 80;

SDL\_QueryTexture(texture\_directions, NULL, NULL, &dst\_directions.w, &dst\_directions.h);

SDL\_RenderCopy(renderer, texture\_directions, NULL, &dst\_directions);

SDL\_RenderPresent(renderer);

}

void carre\_terre(SDL\_Renderer \*ren, int x, int y){

SDL\_Texture \*texture = NULL;

SDL\_Surface \*terre5050 = IMG\_Load("IMG/Base\_terre.png");

texture = SDL\_CreateTextureFromSurface(ren, terre5050);

SDL\_FreeSurface(terre5050);

SDL\_Rect dst;

dst.x = x;

dst.y = y;

SDL\_QueryTexture(texture, NULL, NULL, &dst.w, &dst.h);

SDL\_RenderCopy(ren, texture, NULL, &dst);

}

void mvt\_tank(SDL\_Surface \*Tank\_i, SDL\_Renderer \*ren, int x, int y ){

SDL\_Texture \*texture = NULL;

texture = SDL\_CreateTextureFromSurface(ren, Tank\_i);

SDL\_Rect dst;

dst.x = x;

dst.y = y;

SDL\_QueryTexture(texture, NULL, NULL, &dst.w, &dst.h);

SDL\_RenderCopy(ren, texture, NULL, &dst);

SDL\_RenderPresent(ren);

}

void fond\_terre(SDL\_Renderer\* ren){

int jx, jy;

for(jx = 0; jx<13;jx++){

for(jy = 0; jy<13;jy++){

carre\_terre(ren, 35+50\*jx, 35+50\*jy);

}

}

}

void directions\_panel(SDL\_Renderer \*ren, int x, int y){

SDL\_Texture \*texture = NULL;

SDL\_Surface \*Directions = IMG\_Load("IMG/Directions.png");

texture = SDL\_CreateTextureFromSurface(ren,Directions);

SDL\_FreeSurface(Directions);

SDL\_Rect dst;

dst.x = x;

dst.y = y;

SDL\_QueryTexture(texture, NULL, NULL, &dst.w, &dst.h);

SDL\_RenderCopy(ren, texture, NULL, &dst);

}

void small\_direction (SDL\_Renderer \*ren, int x, int y, int dd){

SDL\_Texture \*texture = NULL;

SDL\_Surface \*d = NULL;

switch(dd){

case(1):d = IMG\_Load("IMG/d1.png");break;

case(2):d = IMG\_Load("IMG/d2.png");break;

case(3):d = IMG\_Load("IMG/d3.png");break;

case(4):d = IMG\_Load("IMG/d4.png");break;

case(6):d = IMG\_Load("IMG/d6.png");break;

case(7):d = IMG\_Load("IMG/d7.png");break;

case(8):d = IMG\_Load("IMG/d8.png");break;

case(9):d = IMG\_Load("IMG/d9.png");break;

}

texture = SDL\_CreateTextureFromSurface(ren,d);

SDL\_FreeSurface(d);

SDL\_Rect dst;

dst.x = x;

dst.y = y;

SDL\_QueryTexture(texture, NULL, NULL, &dst.w, &dst.h);

SDL\_RenderCopy(ren, texture, NULL, &dst);

SDL\_RenderPresent(ren);

}

int choix\_direction(int mx, int my){

if ( mx< 779 && mx> 731 && my>181 && my<229){

return 1;

}

if ( mx< 829 && mx> 781 && my>181 && my<229){

return 2;

}

if ( mx< 879 && mx> 831 && my>181 && my<229){

return 3;

}

if ( mx< 779 && mx> 731 && my>131 && my<179){

return 4;

}

if ( mx< 879 && mx> 831 && my>131 && my<179){

return 6;

}

if ( mx< 779 && mx> 731 && my> 81 && my<189){

return 7;

}

if ( mx< 829 && mx> 781 && my> 81 && my<189){

return 8;

}

if ( mx< 879 && mx> 831 && my> 81 && my<189){

return 9;

}else{

return 0;

}

}

//Sources:

//http://www.willusher.io/sdl2%20tutorials/2013/08/17/lesson-2-dont-put-everything-in-main/