**MAIN.CPP**

#include <iostream>

#include <string.h>

#include <conio.h>

#include <stdlib.h>

#include <time.h>

#include <SDL.h>

#include <SDL\_image.h>

#include <SDL\_ttf.h>

#include "Game.h"

#include "Vagon.h"

#include "Bandido.h"

#include "Mina.h"

#include "Moneda.h"

#include "Item.h"

#include "Lista.h"

#include "Parametros.h"

using namespace std;

#define milisegundos 30 //30

void correrGame(Game &game,int anchoVentana,int altoVentana);

void imprimirMinas(Game &game);

void imprimirComanda(Game &game);

void imprimirParametros(Game &game);

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

int anchoCelda = 40;

int altoCelda = 40;

int anchoVentana = 800;

int altoVentana = 600;

int filas = altoVentana/altoCelda; //15

int columnas = anchoVentana/anchoCelda; //20

Game game;

crearGame(game,filas,columnas,anchoCelda,altoCelda);

correrGame(game,anchoVentana,altoVentana);

cout<<"presione una tecla para salir..."<<endl;

getch();

return 0;

}

void imprimirMinas(Game &game){

if(!listaVacia(game.minas)){

cout << "======================" << endl;

cout<<"Lector de minas"<<endl;

cout << "======================" << endl;

PtrNodoLista nodo=primero(game.minas);

while(nodo !=finLista()){

//El nodo apunta al dato, y casteo el dato a mina

Mina\* mina =(Mina\*) nodo->ptrDato;

cout<<"posX:"<<getColumna(mina)<<";posY:"<<getFila(mina)<<";codItem:"<<getCodItem(mina)<<";IP:"<<getIP(\*mina)<<";";

for(int i=0;i<5;i++){

cout<<"seq["<<i+1<<"]:"<<getSecuencia(\*mina)[i]<<" ";

}

cout<<endl;

nodo = siguiente(game.minas, nodo);

}

}

// Agrego un par de líneas a la consola

cout << endl;

cout << endl;

cout << endl;

}

void imprimirComanda(Game &game){

if(!listaVacia(game.comanda)){

cout << "======================" << endl;

cout << "= Lector de Comanda =" << endl;

cout << "======================" << endl;

PtrNodoLista nodo=primero(game.comanda);

while(nodo !=finLista()){

//El nodo apunta al dato, y casteo el dato a Comanda

Comanda\* comanda =(Comanda\*) nodo->ptrDato;

cout<<"codItem:"<<comanda->codItem<<";cantidad:"<<comanda->cantidad<<endl;

nodo = siguiente(game.comanda, nodo);

}

}

// Agrego un par de líneas a la consola

cout << endl;

cout << endl;

cout << endl;

}

void imprimirParametros(Game &game){

cout << "======================" << endl;

cout<<"Lector de Parametros"<<endl;

cout << "======================" << endl;

cout<<"S:"<<game.parametros.claveS<<"\nP:"<<game.parametros.claveP<<"\nA:"<<game.parametros.claveA<<endl;

cout<<"posXE:"<<game.parametros.claveposXE<<"\nposYE:"<<game.parametros.claveposYE<<"\nIM:"<<game.parametros.claveIM<<endl;

cout<<"VM:"<<game.parametros.claveIM<<"\nIB:"<<game.parametros.claveIB<<"\nVB:"<<game.parametros.claveVB<<"\nIP:"<<game.parametros.claveIP<<endl;

// Agrego un par de líneas a la consola

cout << endl;

cout << endl;

cout << endl;

}

void correrGame(Game &game,int anchoVentana,int altoVentana){

if(SDL\_Init(SDL\_INIT\_EVERYTHING)>=0){

int anchoCelda=game.anchoCelda;

int altoCelda=game.altoCelda;

SDL\_Window \*window;

SDL\_Renderer \*renderer;

window = SDL\_CreateWindow( "Clash of UNLa", SDL\_WINDOWPOS\_CENTERED,SDL\_WINDOWPOS\_CENTERED,

anchoVentana,altoVentana, SDL\_WINDOW\_RESIZABLE | SDL\_RENDERER\_PRESENTVSYNC );

renderer = SDL\_CreateRenderer(window,-1,SDL\_RENDERER\_ACCELERATED);

IMG\_Init(IMG\_INIT\_PNG);

TTF\_Init();

SDL\_Event event;

const unsigned char \*keys;

keys = SDL\_GetKeyboardState(NULL);

leerParametros(game); //Archivo

imprimirParametros(game);

leerComanda(game); //Archivo

imprimirComanda(game);

setTerreno(game, renderer);

leerMinas(game, renderer); //Archivo

imprimirMinas(game);

int posXE=atoi(game.parametros.claveposXE.c\_str()); //Posicion de la estación

int posYE=atoi(game.parametros.claveposYE.c\_str());

Estacion estacion;

Estacion \*ptrEstacion=&estacion;

crearEstacion(estacion,renderer,posXE,posYE,anchoCelda,altoCelda);

ubicarEstacion(game,ptrEstacion);

Lista locomotora;

crearLista(locomotora,NULL);

Vagon vagon;

crearVagon(vagon,"c1", 0, 7,"der",anchoCelda, altoCelda,0,"");//

ubicarVagon(game,adicionarPrincipio(locomotora,&vagon));

setDireccion(game,getDireccion(\*(Vagon\*)primero(locomotora)->ptrDato));

Lista bandidos;

crearLista(bandidos,NULL);

Lista monedas;

crearLista(monedas,NULL);

dibujarTerreno(game,renderer);

dibujarEntidades(game,renderer);

SDL\_RenderPresent(renderer);

SDL\_Delay(1000);

int IM=numeroStringRand(game.parametros.claveIM);//Numero random para IntervaloMoneda

int IB=numeroStringRand(game.parametros.claveIB);//Numero random para IntervaloBandido

int contadorFps = 0;//Contador de frames por intervalo

while(!getGameOver(game)){

controlarEventos(game,event,keys);

if(getIntervalo(game)==10){

setIntervalo(game,0);

setContadorSegundo(game,getContadorSegundo(game)+1);//Cada vez que pasan los 10 intervalos, suma un segundo

Vagon \* vagon = (Vagon\*)primero(locomotora)->ptrDato;

if(!(vagon->detenido)){

cambiarColumna(game,primero(locomotora));

cambiarFila(game,primero(locomotora));

}

cambiarDireccion(game, locomotora);

//Revisa si las minas ya tienen que producir un item

actualizarMinas(game);

///Revisa si se acabo el tiempoDeVida de las monedas y/o bandidos

actualizarMonedas(game,monedas,renderer);

actualizarBandidos(game,bandidos,renderer);

/// GeneracionDeMonedas, la vidaMoneda se calcula con la VM + el segundoActual

if(getContadorSegundo(game)==IM){

int VM=numeroStringRand(game.parametros.claveVM);

generarMonedas(game,monedas,renderer,VM);

IM=getContadorSegundo(game)+numeroStringRand(game.parametros.claveIM); ///Le sumo los segundos actuales al IM para saber el segundo final

///En la cual aparece una nueva moneda

}

///GeneracionDeBandido. la vidaBandido se calcula con la VB + el segundo actual

if(getContadorSegundo(game)==IB){

int VB=numeroStringRand(game.parametros.claveVB);

generarBandidos(game,bandidos,renderer,VB);

IB=getContadorSegundo(game)+numeroStringRand(game.parametros.claveIB);

}

evaluarColision(game,locomotora,monedas,bandidos,renderer);

}

if(!getGameOver(game)){

if(getIntervalo(game)==2){

setGameOver(game,evaluarColisiones(game,primero(locomotora)));

encuentroConBandido(game,locomotora,bandidos);

}

SDL\_RenderClear(renderer);

dibujarTerreno(game,renderer);

dibujarEntidades(game,renderer);

dibujarPuntuacion(game,renderer,window,locomotora,monedas,bandidos);

SDL\_RenderPresent(renderer);

SDL\_Delay(milisegundos);

if(contadorFps % ((100/milisegundos)\*atoi(getClaveS(game.parametros).c\_str())) == 0){

setIntervalo(game,getIntervalo(game)+1);

}

}

contadorFps++;

}

SDL\_Delay(500);

cout<<"Destruimos las instancias"<<endl;

destruirGame(game,renderer,window,locomotora,monedas,bandidos);

}

}