#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include "game.h"

#include "level.h"

#include "gui.h"

#include "highscores.h"

#ifdef \_WIN32

#include <crtdbg.h>

#endif

/\*

(optional) parameters:

- Number of the level to start in.

- Random seed for level generation.

\*/

int main(int argc, char \*argv[])

{

int i;

LevelInfo info = {10,10,0,CLASSIC,0.5,5,0,0.9};

Enemy uno={1\*TILE\_SIZE,1\*TILE\_SIZE,EAST,3,2,0,0};

Obstacle dos0={OBSTACLE,2,1,0};

Obstacle dos1={OBSTACLE,4,2,0};

Obstacle dos2={OBSTACLE,5,3,0};

Obstacle dos3={OBSTACLE,6,4,0};

Obstacle dos4={OBSTACLE,1,2,1};

Obstacle dos5={OBSTACLE,2,4,1};

Obstacle dos6={OBSTACLE,3,6,1};

Obstacle dos7={OBSTACLE,4,0,1};

Obstacle dos8={OBSTACLE,4,1,1};

Obstacle dos9={OBSTACLE,4,2,1};

int total\_score = 0;

Game game;

int end\_of\_game = 0;

/\* Initialize level number \*/

int level = argc >= 2 ? atoi(argv[1]) : 1;

/\* Initialize random seed \*/

int seed = argc == 3 ? atoi(argv[2]) : (int)time(NULL);

#ifdef \_WIN32

//\_CrtSetBreakAlloc(180);

#endif

srand(seed);

printf("Seed is %d...\n\n", seed);

/\* Initialize the gui \*/

//uno.frozen=0;

//uno.is\_boss=1;

info.level\_nr=level;

gui\_initialize(); //eerst enkel dit

gui\_set\_level\_info(&info);

gui\_draw\_buffer();

gui\_add\_enemy(&uno);

gui\_draw\_buffer();

for(i=1;i<9;i++){

uno.x=i\*TILE\_SIZE;

//uno.y=i\*TILE\_SIZE;

gui\_add\_enemy(&uno);

gui\_draw\_buffer();

\_sleep(150);

}

uno.move\_direction=SOUTH;

for(i=1;i<9;i++){

uno.y=i\*TILE\_SIZE;

//uno.y=i\*TILE\_SIZE;

gui\_add\_enemy(&uno);

gui\_draw\_buffer();

\_sleep(150);

}

uno.move\_direction=WEST;

for(i=1;i<8;i++){

uno.x=uno.x-TILE\_SIZE;

//uno.y=i\*TILE\_SIZE;

gui\_add\_enemy(&uno);

gui\_draw\_buffer();

\_sleep(150);

}

uno.move\_direction=NORTH;

for(i=1;i<8;i++){

uno.y=uno.y-TILE\_SIZE;

//uno.y=i\*TILE\_SIZE;

gui\_add\_enemy(&uno);

gui\_draw\_buffer();

\_sleep(500);

}

gui\_add\_obstacle(&dos0);

gui\_draw\_buffer();

gui\_add\_obstacle(&dos2);

gui\_draw\_buffer();

gui\_clean();

gui\_add\_obstacle(&dos3);

gui\_draw\_buffer();

gui\_add\_obstacle(&dos4);

gui\_draw\_buffer();

gui\_add\_obstacle(&dos5);

gui\_draw\_buffer();

gui\_add\_obstacle(&dos6);

gui\_draw\_buffer();

gui\_add\_obstacle(&dos7);

gui\_draw\_buffer();

gui\_add\_obstacle(&dos8);

gui\_draw\_buffer();

gui\_add\_obstacle(&dos9);

gui\_draw\_buffer();

while(!end\_of\_game) {

/\* Initialize the next game instance \*/

init\_game(&game, level);

/\* Start the actual game loop \*/

do\_game\_loop(&game, total\_score);

total\_score += game.score;

/\* Check if player cleared the level \*/

if(!game.game\_over)

level++;

else

end\_of\_game = 1;

/\* Clean up \*/

destroy\_game(&game);

/\* Ask for user input \*/

wait\_for\_space();

}

gui\_clean();

printf("\nYour total score is %d\n", total\_score);

/\* highscores \*/

load\_highscores();

check\_highscore\_entry(total\_score);

display\_highscores();

save\_highscores();

#ifdef \_WIN32

getchar();getchar();

\_CrtDumpMemoryLeaks();

#endif

return 0;

}