https://github.com/Krispy101/CSI-211-computer-programming-projects/blob/main/Snake%20Game.cpp

#include <iostream>

#include<conio.h>

#include <windows.h>

using namespace std;

bool gameover;

const int width = 40;

const int height = 20;

int snake\_head, snake\_bottom, fruit\_x, fruit\_y, score;//x is head , y is tail

int TailX[100], TailY[100];

int nTail;

enum direction { STOP = 0, LEFT, RIGHT, UP, DOWN };

edirection direct;

void setup() {

gameover = false;

direct = STOP;

snake\_head = width / 2;// so the head is in the middle

snake\_bottom = height / 2;// so the tail is in the middle

fruit\_x = rand() % width;

fruit\_y = rand() % height;

score = 0;

}

void draw() {

system("cls");

for (int i = 0; i < (width + 4); i++)

{

cout << "\*";

}

cout << endl;

for (int i = 0; i < (width + 4); i++)

{

cout << "@";

}

cout << endl;

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

for (int j = 0; j < width; j++)

{

if (j == 0)

cout << "\*@";

if (i == snake\_bottom && j == snake\_head)

cout << "8";

else if (i == fruit\_y && j == fruit\_x)

cout << "$";

else {

bool print = false;

for (int k = 0; k < nTail; k++) {

if (TailX[k] == j && TailY[k] == i)

{

cout << "s";

print = true;

}

}

if (!print)

{

cout << " ";

}

}

if (j == width - 1)

cout << "@\*";

}

cout << endl;

}

for (int i = 0; i < (width + 4); i++)

{

cout << "@";

}

cout << endl;

for (int i = 0; i < (width + 4); i++)

{

cout << "\*";

}

cout << endl;

cout << "Money: " << score << endl;

}

void input() {

if (\_kbhit()) { //w is up , s is down, a is left, d is right

switch (\_getch()) {

case 'a':

direct = LEFT;

break;

case 'd':

direct = RIGHT;

break;

case 's':

direct = DOWN;

break;

case 'w':

direct = UP;

break;

case 'x':

gameover = true;

break;

}

}

}

int logic() {

int prevX= TailX[0];

int prevy = TailY[0];

int prev2X, prev2Y;

TailX[0] = snake\_head;

TailY[0] = snake\_bottom;

for (int i = 1; i < nTail; i++) {

prev2X = TailX[i];

pre v2Y = TailY[i];

TailX[i] = prevX;

TailY[i] = privy;

prevX = prev2X;

prevy = prev2Y;

}

switch (direct) {

case LEFT:

snake\_head--; //head

break;

case RIGHT:

snake\_head++;

break;

case UP:

snake\_bottom--;

break;

case DOWN:

snake\_bottom++;

break;

default:

break;

}

//if (x > width || x < 0|| y>height ||y<0)

//gameover = true;

if (snake\_head >= width)

snake\_head = 0;

else if (snake\_head < 0)

snake\_head = width - 1;

if (snake\_bottom >= height)

snake\_bottom = 0;

else if (snake\_bottom < 0)

snake\_bottom = height - 1;

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

if(TailX[i]== snake\_head &&TailY[i]== snake\_bottom)

gameover = true;

}

if (snake\_head == fruit\_x && snake\_bottom == fruit\_y)

{

score = score + 10;

fruit\_x = rand() % width;

fruit\_y = rand() % height;

nTail++;

}

return score;

}

void ShowConsoleCursor(bool showFlag)

{

HANDLE out = GetStdHandle(STD\_OUTPUT\_HANDLE);

CONSOLE\_CURSOR\_INFO cursorInfo;

GetConsoleCursorInfo(out, &cursorInfo);

cursorInfo.bVisible = showFlag; // set the cursor visibility

SetConsoleCursorInfo(out, &cursorInfo);

}

int main() {

system("color 0E");

int difficuty\_level;

cout << "--------Welcome to The Snake Game---------" << endl << endl;

cout << "If you collect enough money your difficulty level will increase." <<endl<< "Select your difficulty level (1-3): “;

cin >> difficuty\_level;

if (difficuty\_level < 1 || difficuty\_level > 3)

{

cout << "Incorrect Input" << endl;

cout << "Select your difficulty level (1-3): ";

cin >> difficuty\_level;

}

ShowConsoleCursor(false);

if (difficuty\_level == 1)

{

setup();

while (!gameover) {

draw();

input();

if (logic() == 200) {

cout << "The level of difficulty is: 2" << endl;

Sleep(50);

}

else if (logic() == 400) {

cout << "The level of difficulty is: 3" << endl;

Sleep(1);

}

else

{

cout << "The level of difficulty is: 1" << endl;

Sleep(300);

}

logic();

}

}

else if (difficuty\_level == 2)

{

setup();

while (!gameover) {

draw();

input();

if (logic() == 200) {

cout << "The level of difficulty is: 3" << endl;

Sleep(1);

}

else {

cout << "The level of difficulty is: 2" << endl;

Sleep(50);

}

logic();

}

}

else if (difficuty\_level == 3)

{

setup();

while (!gameover) {

draw();

input();

logic();

}

}

return 0;

}