**//低空低速弹道仿真;**

**#include "IHF.h"**

**#include "Window.h"**

**//图形绘制相关参数;**

**Window Win;**

**#define Win\_width 1200     //窗口宽度;**

**#define Win\_heigh 800      //窗口高度;**

**double asp = 1;            //图形坐标轴伸缩比例;**

**#define g -9.8             //重力加速度;**

**#define PI 3.14159265      //圆周率;**

**#define iterate 100000     //最大迭代次数;**

**#define Points\_Num 100000  //显示允许的最大微元点数;**

**//弹体参数;**

**double M, R, C, dt;        //弹体质量(g),弹体半径(m),弹形空气阻力系数,弹道精度(越小越精确);**

**//弹体运动学参数;**

**double X, Y;                                          //弹体位置(m);**

**double v, vx, vy, vx\_last, vy\_last, V, θ;             //弹体速度(m/s) [直角坐标,极坐标(θ为弧度制)];**

**double ax, ay;                                        //弹体加速度(m/s^2);**

**double air\_resistance, air\_temper;                    //空气阻力(N),空气温度(℃);**

**double X\_last[Points\_Num], Y\_last[Points\_Num];**

**//弹体目标;**

**double TargetX, TargetY, Tolerance;  //坐标位置,容许误差半径(m)(以目标为圆心以R为半径的一个域,如果弹道穿过这个域则视为击中目标);**

**double Mark\_V, Mark\_θ;**

**//空气阻力系数,空气温度(摄氏度),弹体迎风面积(m^2),相对空气速度矢量(m/s);**

**void Air\_resistance(double C, double T, double S) {**

**v = sqrt(pow(vx, 2) + pow(vy, 2));                                 //计算合速度;**

**double air\_pressure = 101.29 \* pow((T + 273.1) / 288.08, 5.256);   //计算大气压强;**

**double ρ = air\_pressure / (0.2869 \* (T + 273.1));                  //计算大气密度;**

**air\_resistance = (C \* ρ \* S \* pow(v, 2)) / 2;                      //计算空气阻力;**

**}**

**//弹道瞬时三角计算;**

**double d\_sin() { return vy / v; }**

**double d\_cos() { return vx / v; }**

**double d\_slope() { return vy / vx; }**

**//水平积分器;**

**double level() {**

**double d\_vx, dx;**

**//通过力计算加速度;**

**ax = (d\_cos() \* -air\_resistance) / M;**

**vx\_last = vx;**

**//对vx积分;**

**d\_vx = ax \* dt;**

**vx = d\_vx + vx;**

**//对X积分;**

**dx = vx \* dt;**

**X = dx + X;**

**return 0;**

**}**

**//铅锤积分器;**

**double vertical() {**

**double d\_vy, dy;**

**//计算加速度;**

**ay = (d\_sin() \* -air\_resistance) / M;**

**vy\_last = vy;**

**//对vy积分;**

**d\_vy = (ay + g) \* dt;**

**vy = d\_vy + vy;**

**//对Y积分;**

**dy = vy \* dt;**

**Y = dy + Y;**

**return 0;**

**}**

**//初始化图形界面;**

**void Initialize\_Graph() {**

**Win.Initialize\_Window(Win\_width, Win\_heigh, EW\_SHOWCONSOLE);**

**setlinecolor(RGB(31, 31, 31)); setlinestyle(PS\_SOLID || PS\_ENDCAP\_ROUND, 1);**

**}**

**//绘制弹道图形;**

**void Graph\_Drawing() {**

**//弹道轨迹;**

**setfillcolor(RGB(104, 33, 122));**

**solidcircle(X \* asp, Y \* asp, 2);**

**//随X的水平速度;**

**setfillcolor(RGB(10, 89, 247));**

**solidcircle(X \* asp, abs(vx \* asp), 1);**

**//随X的铅锤速度;**

**setfillcolor(RGB(24, 148, 63));**

**solidcircle(X \* asp, abs(vy \* asp), 1);**

**//随X的合加速度;**

**setfillcolor(RGB(244, 120, 34));**

**solidcircle(X \* asp, abs(sqrt(pow(ax \* asp, 2) + pow(ay \* asp, 2))), 1);**

**//随X的空气阻力;**

**setfillcolor(RGB(237, 24, 64));**

**solidcircle(X \* asp, abs(air\_resistance \* asp), 1);**

**}**

**double S, K = 2;**

**void value\_evolution() {**

**X = 0; Y = 0;                      //设置弹体射出瞬间初速度初始位置恒为0;**

**θ = PI / K; K += dt;               //极坐标(θ为弧度制)**

**vx = V \* cos(θ); vy = V \* sin(θ);  //极坐标转换为直角坐标;**

**}**

**//弹体参数初始值设置初始化;**

**void initialize\_value() {**

**asp = 1;                        //绘图缩放比例;**

**air\_temper = 15;                 //大气温度(℃);**

**//弹体质量(g),弹体半径(m),弹形空气阻力系数,弹道精度(越小越精确);**

**M = 1, R = 0.06, C = 0.2; dt = 0.03;**

**V = 1;                          //弹体初速度(m/s);**

**S = PI \* pow(R, 2);              //计算弹体迎风面积(默认为圆柱型刚弹);**

**X = 0; Y = 0;                    //弹体射出瞬间初速度初始位置;**

**TargetX = 100, TargetY = 200**

**, Tolerance = 8;   //目标位置,容差半径(m);**

**}**

**int main() {**

**initialize\_value();**

**for (int i = 1; i < iterate; i++) {//速度变化;**

**V += 1; cout << TargetX - X << endl;**

**for (int i1 = 0; i1 < iterate; i1++) {//角度变化;**

**value\_evolution();**

**while (1) {//弹道积分;**

**Air\_resistance(C, air\_temper, S);**

**level();**

**vertical();**

**//cout << "(" << X << "," << Y << ")" << endl;**

**//弹体击中目标的判定条件;**

**if (X > TargetX - Tolerance && X < TargetX + Tolerance) {**

**if (Y > TargetY - Tolerance && Y < TargetY + Tolerance) {**

**//记录弹体击中目标的条件(V:弹体初速度,θ:弹体发射角度);**

**Mark\_V = V;**

**Mark\_θ = θ;**

**cout << "Mark\_V:" << Mark\_V << " , Mark\_θ:" << Mark\_θ << endl;**

**i = iterate; i1 = iterate;  break;//结束弹道迭代;**

**}**

**}**

**if (Y < 0) { break; }      //弹体落入海平面终止本次弹道积分;**

**}**

**if (K > 20) { K = 2; break; }   //当发射角度小于某值重设为90°,并结束角度迭代;**

**}**

**}**

**//初始化图形窗口;**

**Initialize\_Graph();**

**setlinecolor(RGB(30, 30, 30));**

**line(0, -20, 0, Win\_heigh); line(-20, 0, Win\_width, 0);                                              //绘制坐标轴;**

**setlinecolor(RGB(120, 120, 120));**

**line(0, TargetY \* asp, Win\_width, TargetY \* asp); line(TargetX \* asp, 0, TargetX \* asp, Win\_heigh);  //绘制目标X,Y轴;**

**circle(TargetX \* asp, TargetY \* asp, Tolerance \* asp);                                               //绘制目标容许误差半径;**

**dt = 0.001;**

**X = 0; Y = 0;                      //弹体射出瞬间初速度初始位置;**

**V = Mark\_V, θ = Mark\_θ;**

**vx = V \* cos(θ); vy = V \* sin(θ);  //极坐标转换为直角坐标;**

**while (1) {**

**Air\_resistance(C, air\_temper, S);**

**level();**

**vertical();**

**Graph\_Drawing();**

**if (Y < 0) { break; }      //弹体落入海平面终止本次弹道积分;**

**}**

**system("pause");**

**}**