Lab 4

#include "library.h"

const double g = 9.807;

double projectileh(int v, double t, double a){

double h = v\*t\*(sin(a)) - (0.5\*g)\*(t\*t);

return h;

}

void tabulation(int v, double t, double a){

double d = v\*t\*cos(a);

int t\_elapsed = ((2)\*(v) / g);

move\_to(0,300);

if (t <= t\_elapsed, (int)projectileh(v, t, a) >= 0){

cout << " After " << t << " second(s), height is " << (int)projectileh(v, t,a) << " metres" << endl;

set\_pen\_width\_color(5,color::red);

draw\_point((t)\*50, (500-projectileh(v, t,a)/2));

tabulation(v, t + 0.05,a);

}

}

void drawside(double angle, int length) {

turn\_right\_by\_degrees(angle);

draw\_distance(length);

}

void circle(int n, int numsides, int len) {

double pi = acos(-1.0);

if (n > 0) {

drawside(360.0 / numsides, len);

circle(n - 1, numsides, len);

}

}

void cannon(){

set\_pen\_width(3);

move\_to(45, 495);

draw\_to(0, 460);

draw\_to(100, 400);

move\_to(45, 495);

draw\_to(120, 420);

draw\_to(100, 400);

move\_to(60, 470);

circle(36, 36, 3);

}

void castle(int x){

move\_to(880, 500);

draw\_distance(x);

turn\_right\_by\_degrees(90);

draw\_distance(x);

turn\_right\_by\_degrees(90);

draw\_distance(x);

move\_relative(0, -x / (5/2));

move\_relative(-30, 0);

draw\_distance(10);

turn\_right\_by\_degrees(90);

draw\_distance(15);

turn\_right\_by\_degrees(90);

draw\_distance(10);

turn\_right\_by\_degrees(90);

draw\_distance(15);

move\_relative(10, 0);

draw\_distance(15);

turn\_right\_by\_degrees(90);

draw\_distance(10);

turn\_right\_by\_degrees(90);

draw\_distance(15);

turn\_right\_by\_degrees(90);

draw\_distance(10);

}

void main(){

make\_window(950, 500);

cannon();

castle(50);

print("Enter wanted velocity: ");

const int velocity = read\_double();

print("Enter wanted angle: ");

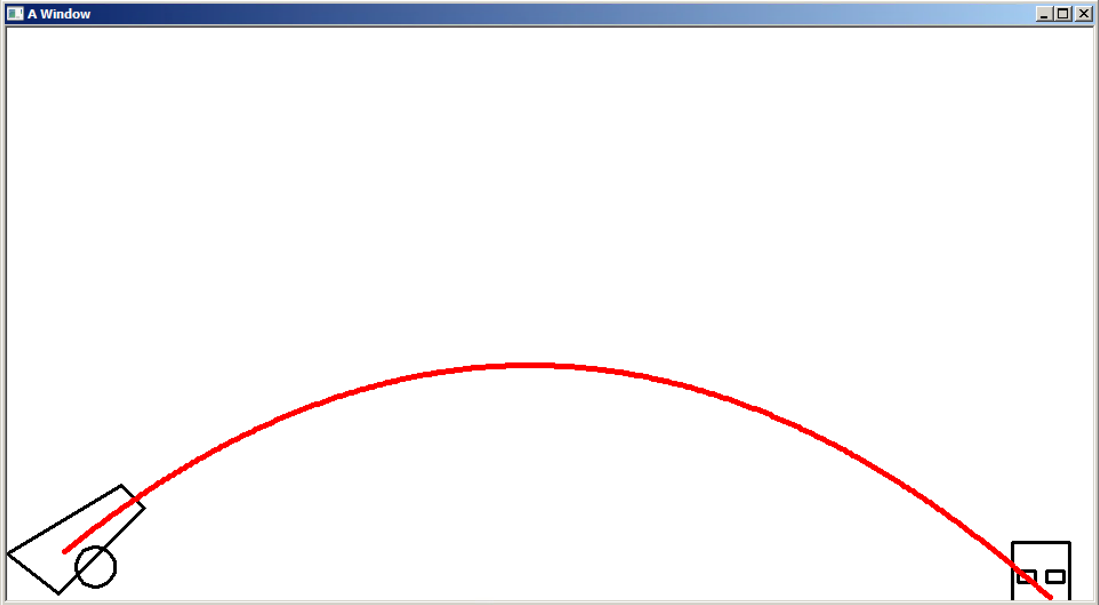
double Piangle = read\_double()\*(3.1415 / 180);

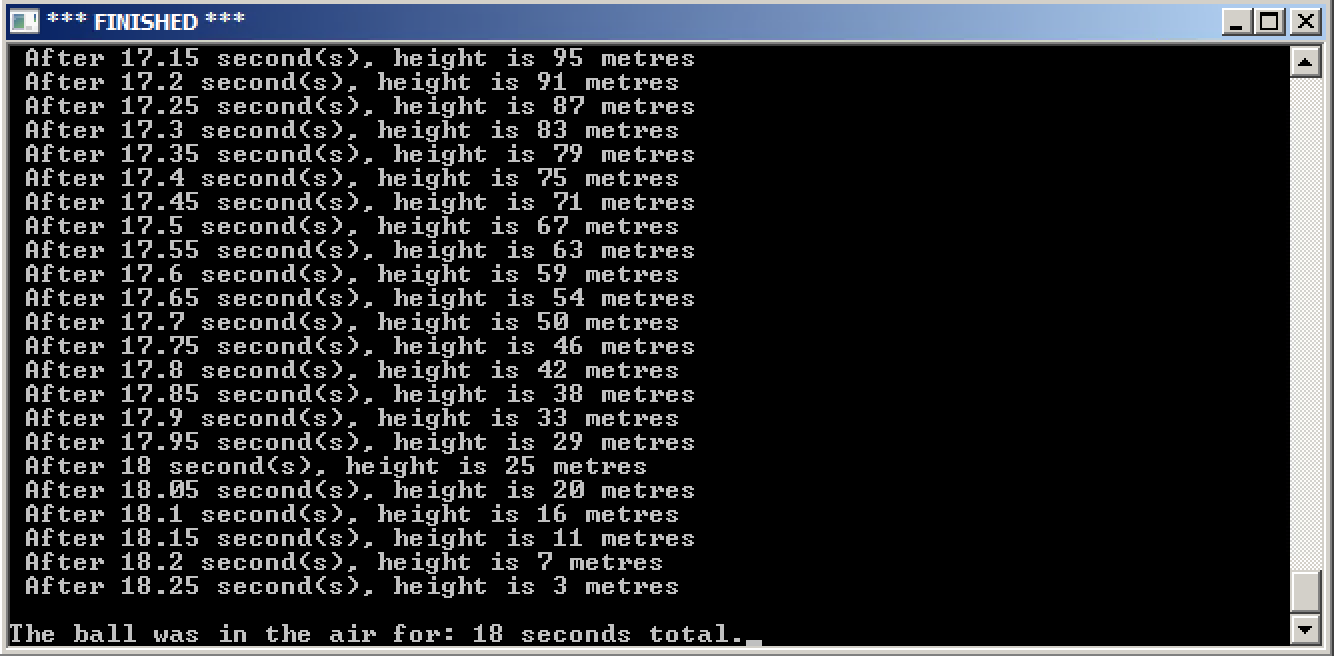
move\_to(98, 415);

tabulation(velocity, 1.0,Piangle);

new\_line();

cout << "The ball was in the air for: " << (int)((2)\*(velocity) / g) << " seconds total.";





}

I couldn’t get the projectile to shoot from the center of the cannon barrel, and couldn’t find out how to create a function that acknowledges the ending of the projectile relevant to the placement of the castle.