

CENTRO UNIVERSITÁRIO SENAC

Lucas Teles

Rodrigo Mendonça

**AC – Algoritmo de Balística**

São Paulo

1º semestre de 2012

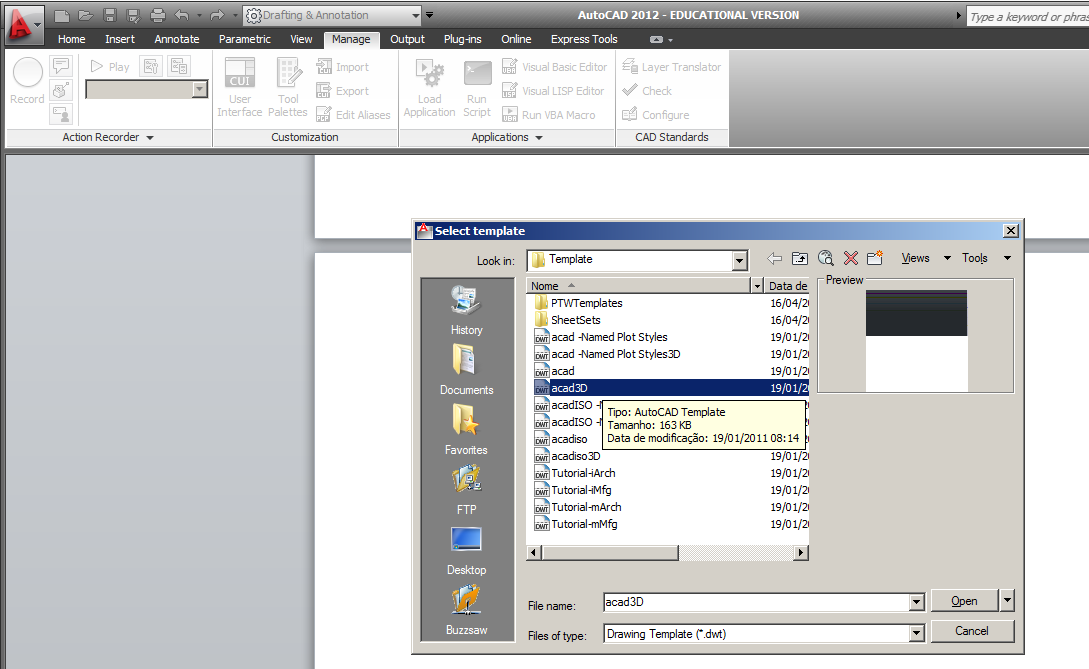
**O programa**

Este programa gera um arquivo no formato SCR para ser importado no AutoCad, contendo as coordenadas dos pontos referentes ao lançamento de um projetil de acordo com os parâmetros selecionados.

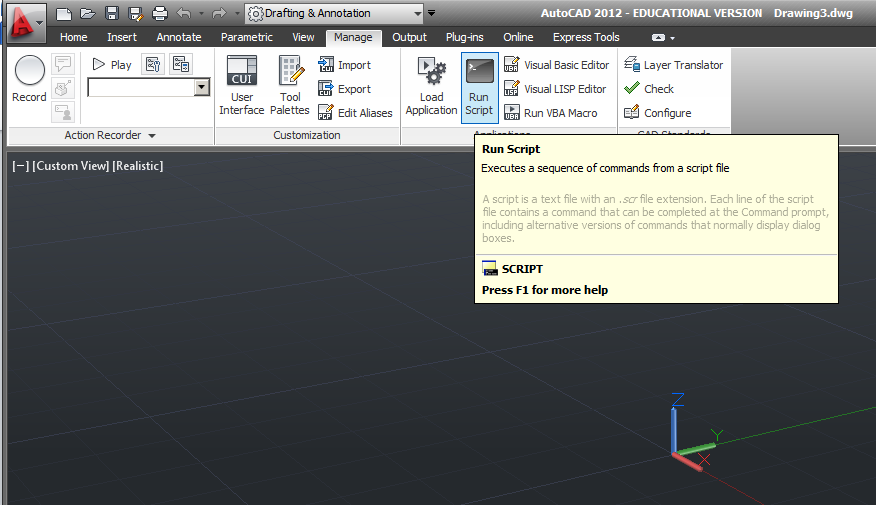
**Como importar**

Após selecionar os parâmetros, o programa gera um arquivo “Exporte.scr” na pasta da executável.

No AutoCad clique em novo, e selecione “acad3d”:



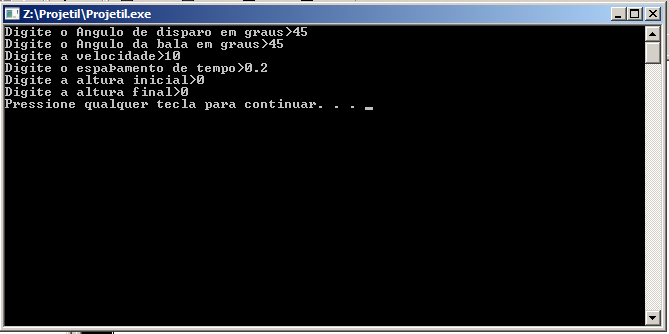
Em seguida na aba Manage, clique em “Run Script”:

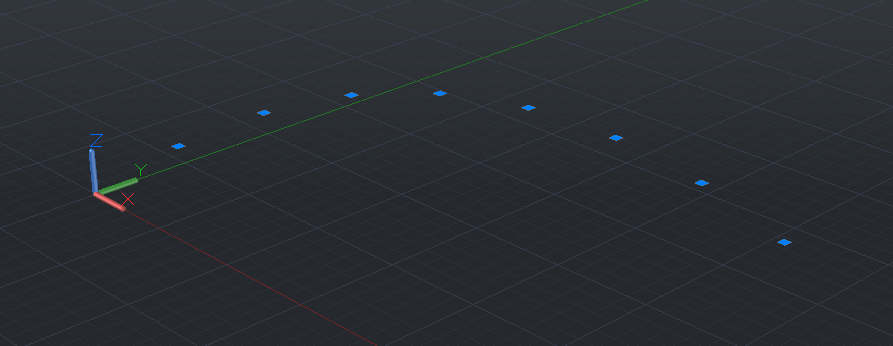


Ao apertar o botão selecione o arquivo gerado pelo programa C

**Testes**

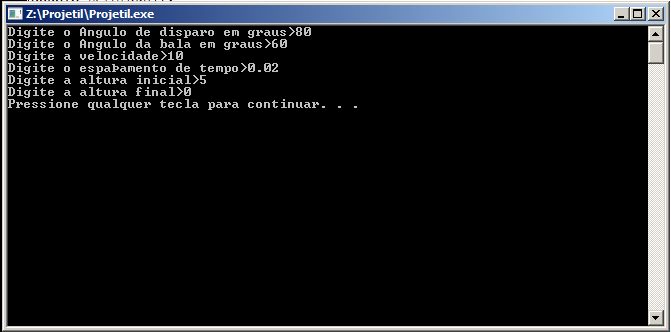
Teste 1

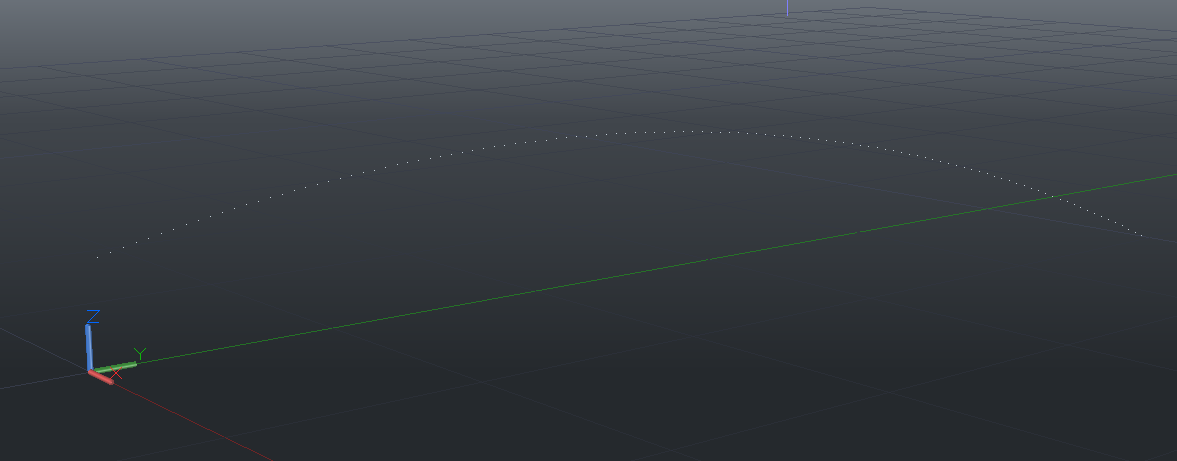


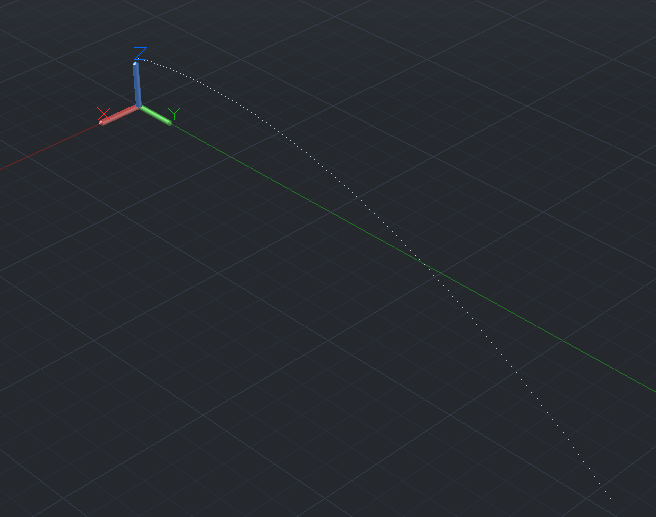


Teste 2

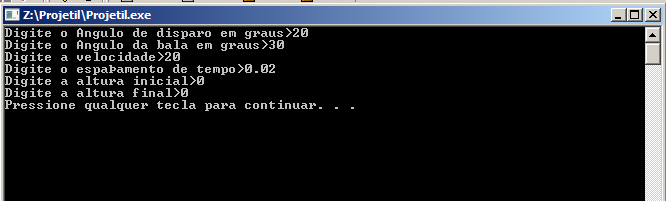
Quanto menor o espaçamento de tempo mais visível é a trajetória, pois são mais pontos

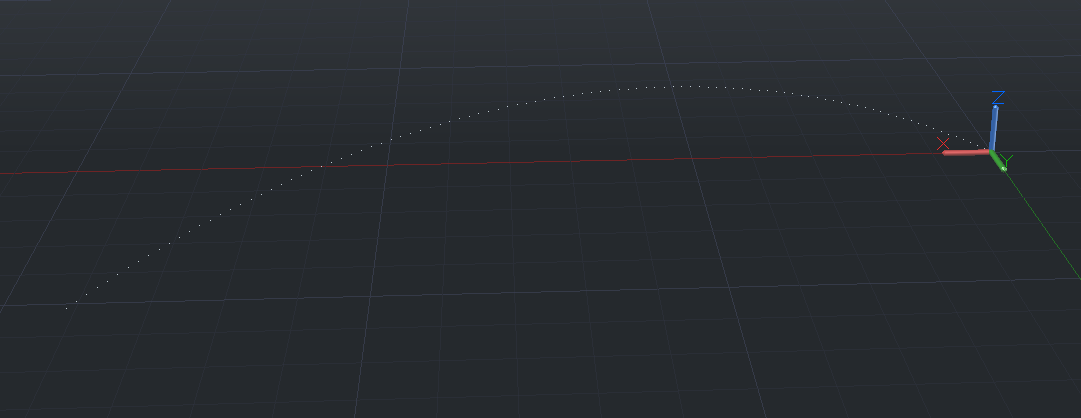






Teste 3







**Codigo Fonte:**

#include <cstdlib>

#include <iostream>

#include <math.h>

#include <String.h>

#include <fstream>

using namespace std;

// metodos

double LerGraus();

double d2r(double nGraus);

void CalculaPontos();

ofstream myFile("Exporte.scr");

// variaveis

const double nG = 9.80665; // aceleraçao da gravidade

double nGrausGiro,nGrausBala,nSpeed,nPasso,nZi,nZf;

int main(int argc, char \*argv[])

{

// lê angulo de disparo

cout << "Digite o Angulo de disparo em graus>";

nGrausGiro = LerGraus();

cout << "Digite o Angulo da bala em graus>";

// lê angulo da bala

nGrausBala = LerGraus();

cout << "Digite a velocidade>";

// lê velocidade

cin >> nSpeed;

cout << "Digite o espaçamento de tempo>";

// lê espaçãmentos entre os pontos

cin >> nPasso;

cout << "Digite a altura inicial>";

// lê altura inicial Z

cin >> nZi;

cout << "Digite a altura final>";

// lê altura final Z

cin >> nZf;

CalculaPontos();

myFile.close();

system("PAUSE");

return EXIT\_SUCCESS;

}

void CalculaPontos()

{

char cBuffer [50];

//cordenadas basicas

float nX=0,nY=0,nZ=0;

//velocidades

float nVxy=0, nVz;

//Auxiliar

float nRaiz=0;

//tempo

float nT = 0;

//controle de final de loop

float nFinal;

//verifica onde ficara o final do Z

if (nZf == 0)

{ nFinal = 0-nZi; }

else

{ nFinal = nZf - nZi; }

while( (nZ >= nFinal ) || nX==0 )

{

//calcula o proximo tempo

nT = nT + nPasso;

//velocidade no plano

nVxy = cos(nGrausBala) \* nSpeed;

nVz = sin(nGrausBala) \* nSpeed;

//cordenadas do X

nX = nT \* nVxy;

// pela equação da reta acha o Y

nY = tan(nGrausGiro) \* nX;

//calculo de balistica

nZ = nVz \* nT - 0.5 \* nG \* pow(nT,2);

sprintf(cBuffer,"point %f,%f,%f\n",nX,nY,nZ+nZi);

cout << "";

myFile << cBuffer;

}

}

/\*

FUNÇÔES ALXILIARES

\*/

double LerGraus()

{

int nGraus;

bool lOK = false;

while(!lOK)

{

cin >> nGraus;

if ( !(nGraus < 0 || nGraus > 360) )

lOK = true;

}

return d2r(nGraus);

}

double d2r(double nGraus)

{

return (nGraus/180)\* M\_PI;

}