

GOI ESKOLA
POLITEKNIKOA

ESCUELA
POLITÉCNICA
SUPERIOR



MONDRAGON
UNIBERTSITATEA

GOLF JOKO BATEN FISIKA

INGENIERIA INFORMATIKAKO GRADUA

EGILEAK:

ANE SAJERAS
AITOR LANDA
MIKEL CASTRO
JOSELYN GANCHOZO

2017ko azaroaren 22a

LABURPENA

Golf joko bat egiteko hainbat pausu jarraitu izan ditugu. Lehenengo pilotaren erreboteak eta horizontalean jarraitu behar duen bidea kalkulatu izan ditugu, hori egiten genuen bitartean, hainbat arazo sortu zaizkigu bidean, baina konponbideak topatu izan ditugu denon artean. Bukatzeko, egindako kalkulu eta azterketa guztiak programazioko app batean probatu izan ditugu, hauek frogatzeko eta azkenengo emaitza ikusi ahal izateko.

RESUMEN

Para crear un juego de golf hemos tenido que hacer varios pasos. Primero hemos sacado cálculos de los rebotes y del seguimiento de la pelota, junto con ello, nos han ido apareciendo problemas que hemos ido solucionando. Por último, hemos puesto todos nuestros conocimientos en una app de programación para probar nuestro juego y para ver el resultado obtenido.

ABSTRACT

In order to create a golf game, we have followed some steps. First of all, we have used maths to calculate the bounces and the tracking of the ball, solving all the problems we had. Finally, we have put all our knowledges in a programming app, to check our game and also to examine the result obtained.

EDUKIEN TAULA

1.	SARRERA.....	1
1.1.	ARAZOAREN AZTERKETA.....	1
1.2.	TXOSTENAREN EGITURA.....	2
2.	PLANGINTZA.....	3
3.	PILOTAREN MUGIMENDUAREN DISEINUA.....	4
3.1.	KALKULUAK.....	4
3.1.1.	PARABOLA	4
3.1.2.	ERREBOTEAK	4
4.	ARAZOAK	7
4.1.	KALKULUETAN	7
4.2.	AZKEN EMAITZA	7
5.	PILOTAREN MUGIMENDUA PROGRAMA	8
6.	BALIDAZIOA	12
7.	ONDORIOAK ETA ETORKIZUNERAKO ILDOAK.....	16
8.	BIBLIOGRAFIA	17

IRUDIEN TAULA

3.1 GOLF JAURTIKETA	4
3.2 ERREBOTEAK.....	4
6.1 Pantailako ardatzak ordenagailuaren errealitatean.....	15
6.2 Pantailako ardatzak gure errealitatean	15

1. SARRERA

Proiektuaren helburu nagusia, golf joko interesgarri bat egitea da. Horretarako dokumentu hau zazpi atal desberdinetan zatituta dago.

Lehenengo atalean arazoaren aurkezpen orokor bat egiten da, erreboteen arazoan ikuspuntu nagusia ipiniz. Horrez gain, txostenaren egitura ageri da.

Bigarren atalean lana egiteko jarraitu genuen plangintza azaltzen da.

Hirugarren atalean berriz, egin dugunaren azalpena modu labur batean ageri da. Gainera atal honetan ere golf jokoaren kalkuluak egiteko formula matematikoak eta azalpen teoriko garrantzitsuenak aurkezten dira, hala nola, erreboteak, energia galera, pilotak behar duen denbora etab.

Laugarren atalean lana egiterako orduan izan ditugun arazoak ageri dira.

Bostgarren atalean, pilotaren mugimenduaren programa ageri da.

Seigarren atalean berriz, golf jokoaren ondorioa nagusiak ageri dira, hau da gure programaren balidaketa agertzen da.

Zazpigarren atalean, etorkizuneko ildoak eta aterako ondorioak daude azalduta. Azkenik informazio guztiaren bibliografia ageri da.

1.1. ARAZOAREN AZTERKETA

Hasieran parabola bat egiten zuen programa eman ziguten, programa honekin pilota ez zituen boterik ematen, hau da, jaisterakoan ez zen inoiz gelditzen. Programa hau abiapuntuz harturik golf joko interesgarri bat sortu behar izan genuen.

Golf jokia egiterakoan hainbat arazo izan ditugu, baina deigarrienak programatzeko orduan izan dira, hain zuzen ere, pilotaren erreboteak eta koordinatuak ipintzeko orduan.

Hasiera batean, golf pilotaren mugimendua programatu genuen, baina Processing programan konprobatzerako orduan konturatu ginen pilotak parabola alderantziz egiten zuela. Ondorioz brainstorming bat egin behar izan genuen programan egin behar genituen aldaketak zeintzuk ziren jakiteko. Horren eraginez konturatu ginen hau gertatzearen arrazoi nagusia koordinatuen kokapena zela. Izan ere, gure (0,0) koordinatua programan goiko aldean baitzegoen eta haren orde (0,360) puntua jarri behar izan genuen gure pilotak parabola ondo egiteko.

Horrez gain, erreboteak egitea asko kostatu zitzaizkigun, pilotak, errebotatu ondoren hasierako balioa hartzen baitzuen eta ez aurreko botearena. Bigarren arazo honi ere irtenbidea bilatzea lortu genion.

1.2. TXOSTENAREN EGITURA

Lehenengo eta behin, golf jokoaren azalpen orokor bat ageri da, geneukan pseudokodea abiapuntu bezala hartuta eta erreboteen arazoan ikuspuntu nagusia ipiniz.

Ondoren gure helburua lortzeko egin genuenaren azalpena modu labur batean ageri da, hau da gure programa ondo funtzionatu zezan jarraitu behar izan genuen prozesua. Honen barruan erabilitako formulak agertzen dira eta hauek programan inplementatzeko sortu zizkiguten arazoak. Gure programa azkenik nola gelditu zen azaltzen du.

Azalpen guztiak bukatu ondoren, guk Visual Studio-en, C-n programatzeko aplikazio batean egindako kodea agertzen da. Aplikazio honetan pilotaren mugimendua zein zen ezin izan genuen ikusi, bakarrik datuak agertzen baitziren eta ez grafikoa edo marrazkia. Horregatik gero mugimendua ondo ikusteko simuladore bat erabili genuen, bi programek ez zutenez C-ko estruktura berdina erabiltzen zenbait aldaketa egin behar izan genizkion guk egindako programari. Azkenik dena bukatzerakoan ateratako ondorioak agertzen dira eta etorkizunerako pentsatutako ildoak.

2. PLANGINTZA

Lehenik eta behin, taldekide guztiek proiektua zeri buruz zihoan zekitela ziurtatu eta gaiari buruzko informazio garrantzitsu gehiago bilatzeari ekin genion. Batez ere honako argibideei erreparatu genien: pilotaren masari, golf zelaiaren dimentsioei, pilotaren mugimenduari, tiro parabolikoen formulei etab.

Hori egin ostean, denon artean brainstorming bat egiten hasi ginen, bakoitzak pentsatzen zituen ideiak esanez. Ondoren, gogoeta guzti horien baitan, denon artean ideia bakar batera heltzea lortu genuen, proiektuaren ideia garrantzitsuetan arreta handia ipiniz. Guzti hori egin bezain laster, gure golf joko interesgarria egiteari ekin genion.

Proiektua egiteko orduan golf pilotaren magnitudeak finkatu behar izan genituen pilotaren masa, abiadura eta angelua esaterako, beraz pilotaren mugimenduari kontu handiz erreparatu behar izan genion. Hasteko, lehenengo parabola egiten hasi ginen eta horren ondoren honek jarraitu beharreko ibilbide osoa, hau da, pilotaren ondorengo erreboteak eta pilotaren segidako desplazamendu txikia. Guzti hori egiteko hainbat kalkulu egin behar izan genituen. Azkenik, golf jokoari buruz gauzatutako guztia Visual Studio programan egin genuen eta ondoren Processing programara pasatu, jokoa ondo funtzionatzen zuela ziurtatzeko.

3. PILOTAREN MUGIMENDUAREN DISEINUA

Atal honetan garrantzitsua da jakitea gure jokoan dagoen golf pilota neurri eta pisu finko batzuk dituela eta pisu honen arabera jokalaria ematen dio abiadura eta angeluaren arabera distantzia batera edo beste batera helduko da. Gure kasuan pilotaren pisua 0,046kg-koa zen eta 43mm-ko diametroa zeukan.

3.1. KALKULUAK

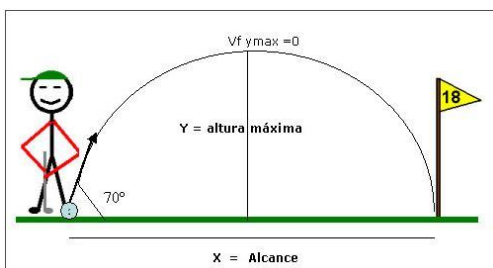
Atal honetan gure golf jokoa ondo funtzionatzeko erabilitako formulak azalduko ditugu. Programa lortzeko garrantzitsuenak parabolarena eta errebotearena izan ziren.

3.1.1. PARABOLA

Gure pilota parabola bat osatzeko bi formula erabili genituen.

$$x = x_0 + V_0 * \cos(\alpha) * t$$

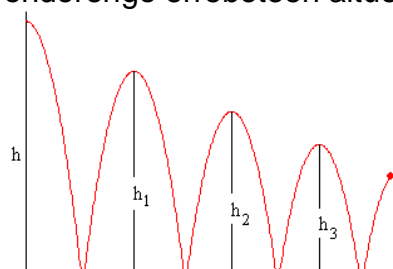
$$y = \frac{y_0 + V_0 * \sin(\alpha) * (t - 1)}{2 * 9,8 * t^2}$$



3.1 GOLF JAURTIKETA[4]

3.1.2. ERREBOTEAK

Demagun pilota bat erortzen uzten dela, hasieran h altueratik. Kalkula ditzagun ondorengo erreboteen altuerak:



LEHEN ERREBOTEA

Pilota erortzean bere energia kontserbatzen denez, pilotaren abiadura kalkula daiteke justu zoruaren kontra talka egitera doan unean:

$$mgh = \frac{1}{2}mu_1^2 \quad u_1 = \sqrt{2gh}$$

Talkan ordea ez da energia kontserbatzen, eta pilotaren abiadura justu talkaren ondoren hau da (modulua): $\mathbf{v}_1 = \mathbf{e} \cdot \mathbf{u}_1$

Orduan pilota gorantz abiatzen da, v_1 abiaduraz eta antzematen duen altuera maximoari dei diezaiogun h_1 . Berriz ere, energiaren kontserbazioaren printzipioa aplikatuz:

$$\frac{1}{2}mv_1^2 = mgh_1 \quad h_1 = e^2h$$

BIGARREN ERREBOTEA

Berriz ere erorketan, energia kontserbatzen denez, kalkulatu da pilotaren abiadura justu zoruaren kontra talka egin aurretik:

$$\frac{1}{2}mv_2^2 = mgh_1 \quad u_2 = \sqrt{2gh_1}$$

Orduan pilotaren abiadura justu talkaren ondoren: $\mathbf{v}_2 = \mathbf{e} \cdot \mathbf{u}_2$

Gero pilota gorantz abiatzen da, v_2 abiaduraz eta antzematen duen altuera maximoari dei diezaiogun h_2 . Berriz ere, energiaren kontserbazioaren printzipioa aplikatuz:

$$\frac{1}{2}mv_2^2 = mgh_1 \quad h_2 = e^2h_1 = e^4h$$

N-GARREN ERREBOTEA

Aurreko ekuazioen itxurak aztertuta, hona hemen n -garren errebotearen ondoren, pilotak antzematen duen altuera maximoa:

$$h_n = e^{2n}h$$

Lehen talkan pilotak honako energia galtzen du:

$$\Delta E_1 = \frac{1}{2}mv_1^2 - \frac{1}{2}mu_1^2 = (e^2 - 1)mgh$$

Bigarrenetan honako energia galtzen du:

$$\Delta E_2 = \frac{1}{2}mv_2^2 - \frac{1}{2}mu_2^2 = (e^2 - 1)mgh_1 = e^2(e^2 - 1)mgh$$

Eta n -garren talkan orduan:

$$\Delta E_n = \frac{1}{2}mv_n^2 - \frac{1}{2}mu_n^2 = (e^2 - 1)mgh_{(n-1)} = e^{2(n-1)}(e^2 - 1)mgh$$

Pilotak galdutako energia osoa, n erreboteren ondoren, energia-galeren batura da, $\Delta E_1 + \Delta E_2 + \Delta E_3 + \dots + \Delta E_n$. Bere hasierako energia osoa $m \cdot g \cdot h$ da eta osorik galduko du infinitu errebote burutu ondoren. Egiazta dezagun hori progresio

geometrikoaren terminoen batura kalkulatzeko. Progresioaren arrazoia e da eta lehen terminoa ΔE_1 .

$$\Delta E_\infty = \frac{(e^2 - 1)mgh}{1 - e^2} = -mgh$$

Pilota gelditu arte iragandako denbora.

Pilota h altueratik askatzen bada, eta pausagunetik abiatuta, hona hemen lurrera iristeko iragaten den denbora:

$$h = \frac{1}{2}gt_0^2 \quad t_0 = \sqrt{\frac{2h}{g}}$$

Pilotak errebotea egin, eta berriz ere h_1 altueraraino igotzen da. Gero berriro erortzen da lurrera. Igo eta jaisteko denbora hau da:

$$t_1 = 2\sqrt{\frac{2h_1}{g}} = 2\sqrt{\frac{2e^2h_0}{g}} = 2t_0e$$

Pilotak berriro errebotea eta berriro gora, h_2 altueraraino. Gero berriro lurrera. Horretarako iragandako denbora:

$$t_2 = 2\sqrt{\frac{2h_2}{g}} = 2\sqrt{\frac{2e^4h_0}{g}} = 2t_0e^2$$

Infinitu errebote burutu arte iragaten den denbora totala kalkulatzeko gehitu behar dira, t_0 eta progresio geometrikoaren infinitu terminoak. Progresioaren arrazoia e da eta lehen terminoa $2t_0e$.

$$t_\infty = t_0 + \frac{2et_0}{1 - e} = t_0 \frac{1 + e}{1 - e} = \frac{1 + e}{1 - e} \sqrt{\frac{2h}{g}}$$

Pilota pausagunetik abiatu beharrean, hasieran v_x abiadura horizontala badu, orduan infinitu erreboteen ondoren horizontalki desplazatuko da, honako distantzia:

$$x = v_x t_\infty$$

4. ARAZOAK

Hainbat arazo izan ditugu jokoaren funtzionamenduari dagokionez. Kalkuluak ateratzerako orduan eta gure programaren lengoaia aztertzerako orduan.

4.1. KALKULUETAN

Hasierako parabola eta erreboteak kalkulatzeko orduan, zalantzak sortzen hasi ziren, magnitude finkoei dagokienez. Horietako bat, paraboletan sortutako angeluak ziren, erreboteetan eragina baitzuen galdetzen hasi ginelako. Konturatu ginen ezetz, konstante jarraitzen zuela, beraz bertan ez dugu aldaketarik egin.

Beste arazoetako bat altuerari eta hasierako abiadurari dagokio. Gure hasierako altuera, hau da, "y" ardatza (0,0) puntuan aurkitzen zen, baina Internetetik aurkitutako informazioan goialdean aurkitzen zenez suposatu genuen hori altuera maximoa izango zela eta jatorrizko puntutik hasteko ez zuela arazorik sortuko, gure asmoa hasieratik erabiltzaileari abiadura eta angelua eskatzea zelako. Beraz, altuera ez zen arazo bat, guk kokatu behar genuelako pilota hasiera batean guk nahi genuen lekuan. Kontuan izan behar genuena hasierako abiadura zen, erabiltzaileak jarritakoa izango zelako.

Azkenik, dena eginda genuenean beste arazo bat sortu zitzaigun, baina hauek arazo teknikoagoak zirenez irakaslearekin geratzea erabaki genuen. Egun bat finkatu eta azaldu genion gure arazoa, hau da, ezin genuela pasa "visual" estrukturatik "processing"-era, hainbat errore ematen baitzituen.

4.2. AZKEN EMAITZA

Nahiz eta arazo horiek garrantzitsuak izan, denen arteko egindako lanarekin aldaketa gutxi behar izan dugu. Izan ere, zalantza horiek baliagarriak izan dira, kontzeptuak garbi edukitzeko eta proiektua aurrera eramateko.

Azken emaitza ona izan da, "Visual-en" gure kalkuluak jarri genituelako eta horrek, parabolaren eta pilotaren jarraipenak aztertzen lagundu zigulako. Hurrengoko pausua "precessing-eko" programara pasatzea izan zen. Horrek nahiz eta arazo gehiago ekarri zituen, programa ondo funtzionatzen zuela ikusita gogobetetze sentimendu bat eragin zigun.

5. PILOTAREN MUGIMENDUA PROGRAMA

Visual Studio-en egindako programaren zeregina jaurtiketa nahiz erreboteen parabolaren puntu guztiak pantailaratzea zen, hauek aztertuz parabolaren forma igarri genezake. Beraz, kode hau ez du inolako grafikorik sortzen, baina jokoaren mezuak idazten ditu eta jokoarekiko harreman bat sortu daiteke. Kode honen neurriak, gure ohiko ardatzetan daude jarrita eta dimentsio maximoak golf zelai batekoak dira ere. Visual Studio-ko kodea hurrengo hau da:

(Hala ere, dokumentu honekin, *Golf.c* atxikitzen da, bertan eran argian ikusteko eta bere funtzionamendua frogatzeko.)

```
#define _CRT_SECURE_NO_WARNINGS
#define PI 3.141592
#define MU 0.3
#include <stdio.h>
#include <string.h>
#include <math.h>
/*Berez golf zelai baten 'x' maximoa ((0,0)posiziotik zuloaraino),
5000m distantzia dago, pantailaren mugak direla eta, 640 izango
litzateke,
baina guk .c kode honetan bakarrik kontuan hartu ditugu neurri
maximoak*/

double Xmax(double v, double angelua); //Alcance máximo (solo en
la parabola del tiro)
double Ymax(double v, double angelua); //Altura máximo (solo en la
parabola del tiro), en los rebotes se sustituye por
"erreboteAltuera"
double XBerri(double t, double v, double angelua, double x); //Cada
punto en el eje X de la parabola cada 0.1s
double YBerri(double y0, double t, double v, double angelua);
//Cada punto en el eje Y de la parabola cada 0.1s
double radianetara(double radianetara); //Pasar el angulo de ° a
radianes
double Tmax(double angelua, double v); //El tiempo que está la
pelota en el aire
double Vbajada(double y, double m, double x); //La velocidad de la
pelota antes de tocar el suelo
double erreboteAltuera(double y, double restituzioa); //La altura
(máxima) del rebote
double r(double v, double t); //Coeficiente de restitución
double VX(double v1, double angelua); //Abiadura X ardatzan,
hurrengo funtziorako erabilgarria.
double rodamendua(double vx, double m); //Pilota egiten duen
desplazamendua boteak eman ondoren, marruskadura gelditu arte.
void erreboteak(double x, double x0, double y, double x1, double
y1, double t, double v, double v1, double tmax, double m, double
restituzioa, double angelua, double y0, double vx);
int main(int argc, char* argv[])
{
    double x=0, vx=0, x0 = 0, y=0, x1=0, y1=0, y0 = 0, t = 0, v=0,
```

```

v1=0, tmax=0, m = 0.00468673469387755102040816326531,
restituzioa=0, angelua=0;
char str[128];
erreboteak(x, x0, y, x1, y1, t, v, v1, tmax, m, restituzioa,
angelua, y0, vx);
printf("\n\n");
printf("Sakatu \"return\" amaitzeko .....");
fgets(str, 128, stdin);
return 0;
}

double Xmax(double v, double angelua)
{
double X;
X = (v*v * 2 * sin(angelua)*cos(angelua)) / (9.8);
return X;
}

double Ymax(double v, double angelua)
{
double Y;
Y = (v*v*sin(angelua)*sin(angelua))/(2*9.8);
return Y;
}

double XBerri(double t, double v, double angelua, double x0)
{
double x;
x = x0+v*cos(angelua)*t;
return x;
}

double YBerri(double y0, double t, double v, double angelua)
{
double y;
y = y0 + v*sin(angelua)*t-0.5*9.8*pow(t,2);
return y;
}

double Tmax(double angelua, double v)
{
double g = 9.8, tmax;
tmax = (2 * v*sin(angelua)) / g;
return tmax;
}

double radianetara(double angelua)
{
double radianak;
radianak = (angelua*PI) / 180;
return radianak;
}

double Vbajada(double y, double m, double x)
{
double ep, ez, Vbajada;
ep = m*9.8*y;
ez = ep;
Vbajada = sqrt(ez / (m*0.5));
return Vbajada;
}

```



```
}
double erreboteAltuera(double y, double e)
{
    double h, g=9.8;
    h = e*y;
    return h;
}
double r(double v, double t)
{
    double e, g = 9.8;
    e = 1 / (1 + (2 * v) / (g*t));
    return e;
}
double VX(double v1, double angelua)
{
    double vx;
    vx = v1*cos(angelua);
    return vx;
}
double rodamendua(double vx, double m)
{
    double d;
    d = (7 * m*pow(vx, 2)) / (10 * MU*m*9.8);
    return d;
}
void erreboteak(double x, double x0, double y, double x1, double
y1, double t, double v, double v1, double tmax, double m, double
restituzioa, double angelua, double y0, double vx)
{
    char str[128];
    printf("Emaidazu hurrengo balioak:\n");
    printf("Pilota jaurtitzeko angelua: \n");
    fgets(str, 128, stdin);
    sscanf(str, "%lf", &angelua);
    angelua = radianetara(angelua);
    printf("Pilota jaurtitzeko hasierako abiadura: \n");
    fgets(str, 128, stdin);
    sscanf(str, "%lf", &v);
    tmax = Tmax(angelua, v);
    v1 = Vbajada(y, m, x);
    x0 = x1;
    restituzioa = r(v, tmax);
    t = 0;
    do
    {
        x1 = XBerri(t, v, angelua, x0);
        y1 = YBerri(y0, t, v, angelua);
        t = t + 0.01;
        printf("(%lf, %lf)\n", x1, y1);
    } while (t <= tmax);
    printf("\n\n \t----- ERREBOTEAK ----- \n\n");
    do {
        v1 = Vbajada(y, m, x);
        y = erreboteAltuera(y, restituzioa);
```

```

x0 = x1;
v = v*r(v, t);
t = 0;
tmax = Tmax(angelua, v);
y = Ymax(v, angelua);
do
{
    x1 = XBerri(t, v, angelua, x0);
    y1 = YBerri(y0, t, v, angelua);
    t = t + 0.01;
    printf("(%lf, %lf)\n", x1, y1);
} while (t <= tmax);
printf("\n\n \t----- ERREBOTEAK ----- \n\n");
if (x1 > 5010)
    break;
} while (y1 > 0.01);
vx = VX(v1, angelua);
x1 += rodamendua(vx, m);

if ((x1 == 5000 && y1 == 0) || (x1 > 4999.99 && y1 < 0.02) && (x1
< 5000.1 && y1 < 0.02))
    printf("\tZorionak! Zuloan sartu duzu pilota");
else if (x1 > 5010)
    printf("Zulotik pasa zara, eta golf pilota laku batera erori
da.\n \tJokoa amaitu da.\n");
else if (5010 > x1 && x1 > 5000.1)
{
    double AX = 0;
    AX = 5000 - x1;
    printf("Pixka bat pasatu zara, baina oraindik Green-ean zaude,
zulotik %lfm-tara. \n", AX);
    erreboteak(x, x0, y, x1, y1, t, v, v1, tmax, m, restituzioa,
angelua, y0, vx);
}
else
{
    double AX = 0;
    AX = 5000 - x1;
    printf("Oraindik %.3lfm-tara zaude zuloarekiko, eman beste golpe
bat. \n", AX);
    erreboteak(x, x0, y, x1, y1, t, v, v1, tmax, m, restituzioa,
angelua, y0, vx);
}
}

```

6. **BALIDAZIOA**

Hasiera batean, “Visual-en” probatu genuen gure jokoa, kalkuluak ondo egiten zituela ziurtatzeko. Hori konprobatu ostean, jokoa behar den bezala funtziona dezan beste programa bat erabili dugu, “Processing” hain zuzen.

Horretarako, lehenik eta behin, bere funtzionamendua eta lengoaia aztertu ditugu. Dena burutu ostean, Visual-en egindakoa Processing-era pasatu dugu. Bitartean, kontuan izan behar genuen, guk egindako kalkuluak eta programarenak ez zirela berdinak, hau da, errealitatean gure jatorrizko puntua pantailaren goialdean aurkitzen zelako. Hori gutxi balitz, programak dena alderantziz egiten zuen, parabolak gora egin beharrean behera egiten zituztelako.

Arazo hau konpontzeko hainbat aldaketa egin behar izan ditugu kalkuluetan, horretarako moldatu izan behar genituen gure kalkuluak, errealitatean egin beharreko mugimenduak egiteko. Kalkulu horien artean, garrantzitsuena erreferentzia ardatzen aldaketa izan zen, honen bitartez gure ardatz sistemaren eta Processing-en ardatz sistemaren artean erlazio bat sortu genuen.

Processing-eko kodea hurrengo lerroetakoa da, baina dokumentu honekin batera *Golf_Sketch.pde* dokumentu bat gehitzen da programa frogatzeko eta kodea argiago ikusteko.

```
int rad = 10;           // Width of the shape
float xpos, ypos;      // Starting position of shape

float xspeed = 2.2;    // Speed of the shape
float yspeed = 2.8;    // Speed of the shape
float tiempo=0;
float v0=40;
float angelua=45;
float x0=0;
float y0=0;
float v1=0;
float vx, m = 0.00468673469387755102040816326531, restituzioa;

void setup()
{
  size(640, 360);
  noStroke();
  frameRate(30);

  xpos = 0;
  ypos = 360;
}
```

```

void draw()
{
    float angeluaRad;
    float xPantaila, yPantaila;
    float yMax;
    float tMax;
    background(102);
    angeluaRad= radianetara(angelua);
    yMax=Ymax (v0, angeluaRad);

    xpos = XBerri(tiempo, v0, angeluaRad, x0);
    ypos = YBerri(tiempo, v0, angeluaRad, y0);
    tMax= Tmax(angeluaRad, v0);
    xPantaila= xPantaila(xpos);
    yPantaila= yPantaila(ypos);
    ellipse(xPantaila, yPantaila, rad, rad);
    tiempo+=0.1;
    if (ypos<0)
        erreboteak(yMax, tMax);
}

void erreboteak(float yMax, float tMax)
{
    v1=Vbajada(yMax,m);
    restituzioa = r(v0, tMax);
    yMax=erreboteAltuera(yMax, restituzioa);
    x0=xpos;
    v0=v0*r(v0,tiempo);
    tiempo=0;
}

float XBerri(float t, float v, float angelua, float x0)
{
    float x;

    x = x0+v*cos(angelua)*t;
    return x;
}

float YBerri(float t, float v, float angelua, float y0)
{
    float y;
    y = y0 + (v * sin(angelua) * t-0.5 * 9.8 * pow(t,2));
    return y;
}

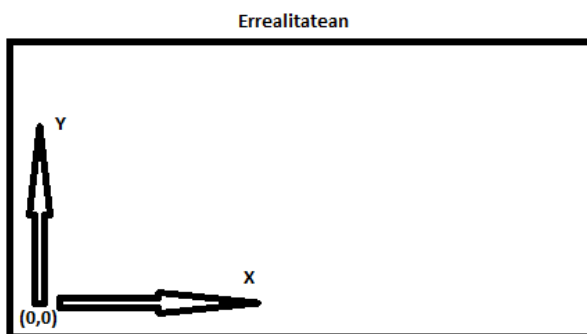
float radianetara(float angelua)
{
    float radianak;
    radianak = (angelua*PI) / 180;
    return radianak;
}

float xPantaila (float x)
{
    float xRef=0;

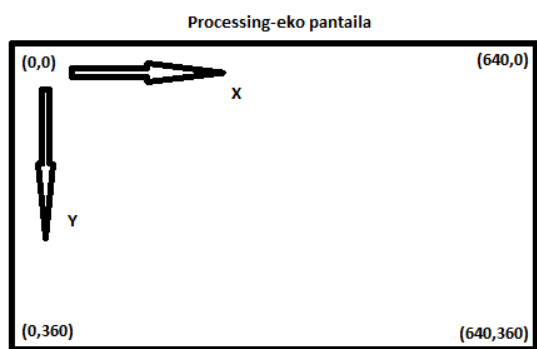
```

```
    int direccion=1;
    float ret=0;
    ret=(x-xRef)*direccion;
    return ret;
}
float yPantaila (float y)
{
    float yRef=360;
    int direccion=-1;
    float ret=0;
    ret=(y-yRef)*direccion;
    return ret;
}

float Ymax(float v, float angelua)
{
    float Y;
    Y = (v*v*sin(angelua)*sin(angelua))/(2*9.8);
    return Y;
}
float Tmax(float angelua, float v)
{
    float g = 9.8, tmax;
    tmax = (2 * v*sin(angelua)) / g;
    return tmax;
}
float Vbajada(float y, float m)
{
    float ep, ez, Vbajada;
    ep = m*9.8*y;
    ez = ep;
    Vbajada = sqrt(ez / (m*0.5));
    return Vbajada;
}
float erreboteAltuera(float y, float e)
{
    float h;
    h = e*y;
    return h;
}
float r(float v, float t)
{
    float e, g = 9.8;
    e = 1 / (1 + (2 * v) / (g*t));
    return e;
}
```



6.1 Pantailako ardatzak ordenagailuaren errealitatean



6.2 Pantailako ardatzak gure errealitatean

6. ONDORIOAK ETA ETORKIZUNERAKO ILDOAK.

Momentuz denboraren ondorioz, golf joko sinple bat egitea lortu dugu. Hau da, gure golf jokoan funtzionamendua sinplea daukala. Baina etorkizun batean gure jokoari hainbat gehigarri jartzea gustatuko litzaiguke. Boteen formulak lortuta gure hobekuntzak zeintzuk izan zezaketen pentsatu genituen, baina azkenean boteak gure programan sartzeko pentsatu genuena baino denbora gehiago erabili genuen. Horregatik hemen etorkizunean jarri nahi genituen gehigarriak, jokia errealago bihurtzeko:

Lehendabizi, oinarria edo lur mota desberdinak jartzea litzateke. Errealitatean golf zelaietan dauden bezala, harea, belar motak eta ura jartzea zen gure helburua. Horretarako, kontuan izan behar genuen lurreko marruskadura, pilotaren paraboletan eta bere jarraipenean. X- ardatzeko eremu desberdinetan marruskadura desberdinak jartzen lortuko genuke errealagoa bihurtzea gure jokia.

Beste gehigarrietako bat, haizea txertatzea zen. Bertan marruskadura indarra egon beharko litzake eta honek pilota airean zein lurrean eragina izango baitzuen.

Aurrekoarekin lotuta, angeluaren aldaketa ere hobetuko genuke, haizeak pilotarengan duen eragina kontuan hartuz, honek nola aldatuko luke parabolaren angeluak.

7. BIBLIOGRAFIA

- [1] **Fleita, Carlos.** Movimiento Parabólico con rebote. *Geogebra*. [Online]
<https://www.geogebra.org/m/JBj5WyY7>.
- [2] **García, Ángel Franco.** Caída libre y sucesivos rebotes. *EHU*. [Online]
http://www.sceu.frba.utn.edu.ar/dav/archivo/homovidens/Proyecto_final_huber/formulas.html.
- [3] **St. Jude School.** Tiro Oblicuo. *SCEU*. [Online]
http://www.sceu.frba.utn.edu.ar/dav/archivo/homovidens/Proyecto_final_huber/formulas.html.
- [4] **3.1.1.1 Golf Jaurtiketa.**
- [5] **3.1.2.1 Erreboteak.**