Nombre: Manuel Alejandro Aponte Ramírez Código: 1201940

Para desarrollar el simulador balístico, se recurrieron a varias fuerzas que interactúan en el entorno y se desarrolló para que este pudiese ser usado en cualquier dirección. El proyectil en este caso se asume como una esfera, además para enriquecer la escena en el instante de colisión con el suelo se da simulación de partículas en el ambiente.

Para el tiro del proyectil se suponen varios valores iniciales;

Posición inicial, velocidad inicial, la aceleración de la tierra, la dirección y magnitud del viento, el radio de la esfera, Angulo de desviación con respecto al piso, y Angulo de elevación.

Para el desarrollo de este se usaron un método de primer orden (Euler), de segundo (Euler mejorado) y de cuarto orden (Range Kutta), estando todos estos métodos en el script disparo.

Las descomposiciones de velocidades en el mismo script se dan de la siguiente forma.

Vel.y = V \* Mathf.Sin(adisparo \* degtorad);

Vel.x = V \* Mathf.Cos(adisparo \* degtorad) \* Mathf.Cos(apiso \* degtorad);

Vel.z = V \* Mathf.Cos(adisparo \* degtorad) \* Mathf.Sin(apiso \* degtorad);

El método de Euler

//POSICIONES

//eje x y z

pos = pos + Vel \* dt;

//VELOCIDADES

Vel = Vel + a \* dt;

//FUERZA Y ACELERACION

Fresistencia = Vel.normalized;

Fresistencia \*= -0.5f \* 1.23f \* Vel.magnitude \* Vel.magnitude \* (3.1416f \* 0.5f \* 0.5f) \* 0.1f;

Faire = Faire.normalized;

Faire \*= 0.5f \* 1.23f \* Vaire \* (3.1416f \* 0.5f \* 0.5f) \* 0.1f;

a.x = Fresistencia.x+Faire.x;

a.z = Fresistencia.z+Faire.z;

a.y = Fresistencia.y + g+Faire.y;

Metodo de Euler mejorado

//POSICIONES

pos.x = pos.x + (dt / 2) \* (Vel.x + Vel.x+dt);

pos.z = pos.z + (dt / 2) \* (Vel.z + Vel.z+dt);

pos.y = pos.y + (dt / 2) \* (Vel.y + Vel.y+dt);

//VELOCIDADES

Vel.x = Vel.x + (dt / 2) \* (a.x + a.x + dt);

Vel.y = Vel.y + (dt / 2) \* (a.y + a.y + dt);

Vel.z = Vel.z + (dt / 2) \* (a.z + a.z + dt);

//FUERZA Y ACELERACION

Fresistencia = Vel.normalized;

Fresistencia \*= -0.5f \* 1.23f \* Vel.magnitude \* Vel.magnitude \* (3.1416f \* 0.5f \* 0.5f) \* 0.1f;

Faire = Faire.normalized;

Faire \*= 0.5f \* 1.23f \* Vaire \* (3.1416f \* 0.5f \* 0.5f) \* 0.1f;

a.x = Fresistencia.x + Faire.x;

a.z = Fresistencia.z + Faire.z;

a.y = Fresistencia.y + g + Faire.y;

Metodo de Range Kutta

//POSICIONES

float kx1 = dt \* (Vel.x);

float kx2 = dt \* (Vel.x + dt / 2);

float kx3 = dt \* (Vel.x + dt / 2);

float kx4 = dt \* (Vel.x + dt);

float kz1 = dt \* (Vel.z);

float kz2 = dt \* (Vel.z + dt / 2);

float kz3 = dt \* (Vel.z + dt / 2);

float kz4 = dt \* (Vel.z + dt);

float ky1 = dt \* (Vel.y);

float ky2 = dt \* (Vel.y + dt / 2);

float ky3 = dt \* (Vel.y + dt / 2);

float ky4 = dt \* (Vel.y + dt);

pos.x = pos.x + (kx1 + 2 \* kx2 + 2 \* kx3 + kx4) / 6;

pos.y = pos.y + (ky1 + 2 \* ky2 + 2 \* ky3 + ky4) / 6;

pos.z = pos.z + (kz1 + 2 \* kz2 + 2 \* kz3 + kz4) / 6;

//VELOCIDAD

float kvx1 = dt \* (a.x);

float kvx2 = dt \* (a.x + dt / 2);

float kvx3 = dt \* (a.x + dt / 2);

float kvx4 = dt \* (a.x + dt);

float kvz1 = dt \* (a.z);

float kvz2 = dt \* (a.z + dt / 2);

float kvz3 = dt \* (a.z + dt / 2);

float kvz4 = dt \* (a.z + dt);

float kvy1 = dt \* (a.y);

float kvy2 = dt \* (a.y + dt / 2);

float kvy3 = dt \* (a.y + dt / 2);

float kvy4 = dt \* (a.y + dt);

Vel.x = Vel.x + (kvx1 + 2 \* kvx2 + 2 \* kvx3 + kvx4) / 6;

Vel.y = Vel.y + (kvy1 + 2 \* kvy2 + 2 \* kvy3 + kvy4) / 6;

Vel.z = Vel.z + (kvz1 + 2 \* kvz2 + 2 \* kvz3 + kvz4) / 6;

//Fuerzas Y aceleracion

Fresistencia = Vel.normalized;

Fresistencia \*= -0.5f \* 1.23f \* Vel.magnitude \* Vel.magnitude \* (3.1416f \* 0.5f \* 0.5f) \* 0.1f;

Faire = Faire.normalized;

Faire \*= 0.5f \* 1.23f \* Vaire \* (3.1416f \* 0.5f \* 0.5f) \* 0.1f;

a.x = Fresistencia.x + Faire.x;

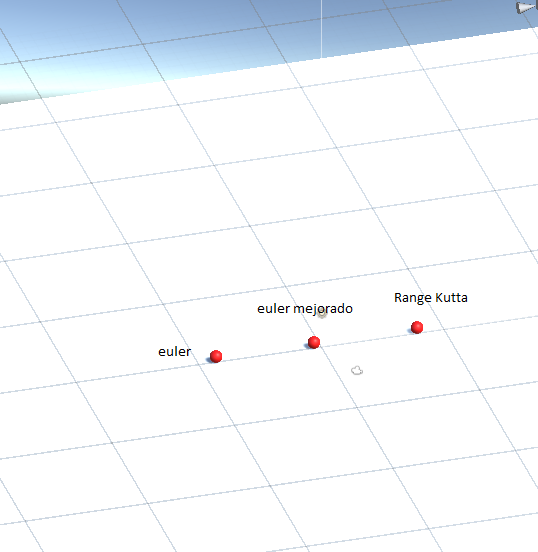
a.z = Fresistencia.z + Faire.z;

a.y = Fresistencia.y + g + Faire.y;

Las partículas que se dispersan después de la colisión tienen un lógica similar a esta, con la diferencia que los ángulos de dispersión son aleatorios, estando esto en el script parabólico, y por otro lado el integrador que maneja la instanciación de estas partículas se encuentra en el script controlador.

Por lado el script changemethod manipula los elementos de GUI, sin embargo este no funciona en su totalidad.

Se usaron 3 proyectiles cada uno con un método distinto, los video de su funcionamiento se encontrara anexo.



En los videos se puede evidenciar que los proyectiles se pueden disparar en cualquier dirección deseada, pero además se da cuenta que este mismo no toma un movimiento parabólico perfecto, por la acción de las fuerzas del viento y de resistencia, lo que hace que las partículas cambien de trayectoria, e inclusive se devuelvan.