

Control nave juego FMX Realizado con 10.1 Berlín.

Utilizaremos las animaciones FloatAnimation - Rotate. Utilizaremos el Timer1 para desplazar la nave y el Timer2 para desplazar la bola/meteorito.

Al mostrar el formulario:

```
void __fastcall TForm1::FormShow(-----)
{
    StartClick(Sender);
}
```

Al finalizar la animación de rotación de la nave la desactivamos:

```
void __fastcall
TForm1::FloatAnimationRotarFinish(-----)
{
    FloatAnimationRotar->Enabled=False;
}
```

Al pulsar las flechas del teclado en el formulario llamaremos a las funciones de dirección:

```
void __fastcall TForm1::FormKeyDown(-----)
{
    switch (Key)
    {
        case 37: Girar(BIz); //Botón izquierda
        break;
        case 38: propulsion(BAtras);
        break;
        case 39: Girar(BDer); //Botón derecha
        break;
        case 40: propulsion(BAdelante);
        break;
    }
}
```

Crearemos varios objetos (en un bucle for) como rocas o formas, con posiciones y tamaños aleatorias (Random) con la función New:

Para el botón de Start:

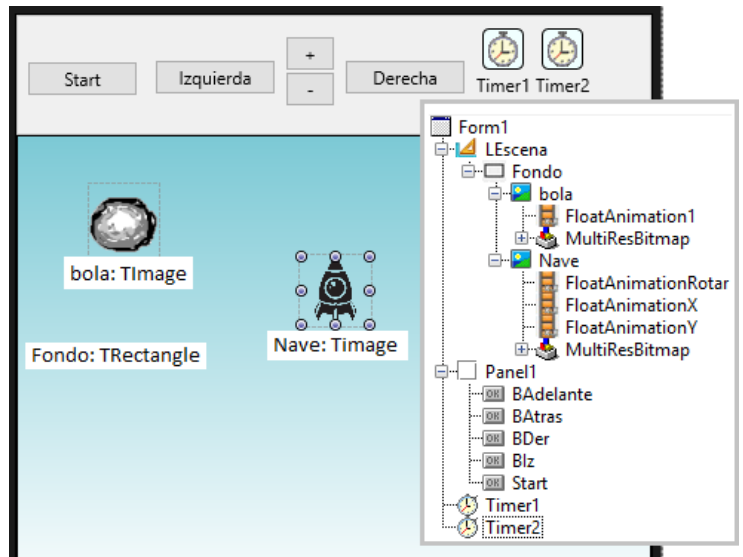
```
void __fastcall TForm1::StartClick(-----)
{
    int c,i,j,k;
    Timer2->Enabled=True; //movimiento bola Nave-
    >Tag=0; Nave->Position->X=Int(Fondo->Width/2);
    Nave->Position->Y=Int(Fondo->Height/2);
    //----Generamos rocas-----
    TCircle * roca; //creamos nombre para nuevo objeto
    randomize; //baraja
    for (c=1;c<15;c++) // Bucle finito: repite el bucle 15 veces
    {
        roca = new TCircle(NULL); //creo nuevo objeto del tipo TCircle
        roca->Parent= Fondo;
        i=Random(Fondo->Width);
        j=Random(Fondo->Height);
        k=Random(100);
        roca->SetBounds(i, j, k, k); } //recuadro posición y tamaño
}
```

Para girar la nave con animación +15 grados:

```
void __fastcall TForm1::Girar(-----)
{
    int grados; FloatAnimationRotar-
    >Enabled=False;if (Sender == BIz)
    {grados=-15;}
    else if (Sender == BDer) {grados=+15;}
    FloatAnimationRotar->StartValue=Nave->RotationAngle;
    FloatAnimationRotar->StopValue=Nave->
    RotationAngle+grados; FloatAnimationRotar-
    >Enabled=True; }
}
```

Para desplazar la nave, usamos su propiedad Tag:

```
void __fastcall TForm1::propulsion(-----)
{
    float horiz,verti,velocidad;
    if (Sender==BAdelante) Nave->Tag++;
    else Nave->Tag--; Timer1-
    >Enabled=True; }
void __fastcall TForm1::Timer1Timer(-----)
{
    float veloc,horiz,verti,angulo;
    veloc=Nave->Tag;
    angulo=((Nave->RotationAngle)+90)*3.141592/180 ;
    horiz=cos(angulo);
    verti=sin(angulo); Nave->Position->X=Nave->Position-
```



```
>X+horiz*veloc/10;Nave->Position->Y=Nave->Position-
>Y+verti*veloc/10;
//comprobamos fuera límites -----
```

```

if ((Nave->Position->X > LEscena->Width+10) || (Nave-
>Position->X < LEscena->Position->X) || (Nave->Position
->Y > LEscena->Height+10) || (Nave->Position->
Y < LEscena->Position->Y))
{
    Timer1-
    >Enabled=False;
    Timer2-
    >Enabled=False;
    ShowMessage("Fuera de límites. Pulse
Start");}}Para mover la bola/meteorito por la
pantalla:
void __fastcall TForm1::Timer2Timer(-----)
{
    switch (mover) //control de la bola meteorito
    {
        FloatAnimation1->Enabled=False;
        case 1: //abajoder-----
            ---bola->Position->X++;
            bola->Position->Y++;
            if (bola->Position->X>=Fondo->Width-bola->Width)
            {mover=2;};
            if (bola->Position->Y>=Fondo->Height-bola->Height)
            {mover=3;}; break;
        case 2: //abajoiz-----
            ---bola->Position->X--;
            bola->Position->Y++;
            if (bola->Position->X<=0) {mover=1;};
            if (bola->Position->Y>=Fondo->Height-bola->Height)
            {mover=4;}; break;
        case 3: //arribader-----
            ---bola->Position->X++;
            bola->Position->Y--;
            if (bola->Position->X>=Fondo->Width-bola->Width)
            {mover=4;};
            if (bola->Position->Y<=0) {mover=1;};
        break;case 4: // arribaiz-----
            --bola->Position->X--;
            bola->Position->Y--;
            if (bola->Position->X<=0) {mover=3;};
            if (bola->Position->Y<=0) {mover=2;};
        break; }FloatAnimation1->Enabled=True;
        //comprobamos colisión-----
        ---TRect R1,R2; //Región cuadrada
        int a,b,c,d;
        a = Int(bola->Position-
        >X);b = Int(bola-
        >Position->Y);
        c = Int(bola->Position->X+bola-
        >Width); d = Int(bola->Position-
        >Y+bola->Height);R1=Rect(a,b,c,d);
        a = Int(Nave->Position-
        >X);b = Int(Nave-
        >Position->Y);
        c = Int(Nave->Position->X+Nave-
        >Width); d = Int(Nave->Position-
        >Y+Nave->Height);R2=Rect(a,b,c,d);
        if (IntersectRect(R1,R2)==True)
        {
            Timer1->Enabled=False; Timer2-
            >Enabled=False; ShowMessage("Colisión.
            Pulse Start"); }}
    
```