

Proyecto Final

Edwar Yamir Forero Blanco (2259664)

Santiago Aníbal Carrillo Torres (2259465)

Faber Alexis Solís Gamboa (2259714)

Programación Orientada a Objetos

Semestre #2

Tecnología en desarrollo de software-2724

Universidad del valle

Tuluá-Valle

Diciembre de 2022

Informe Proyecto Final

En este informe se hablará sobre los problemas o las adversidades que se presentaron durante la construcción del código y la ejecución del programa, estos problemas se expondrán en tres sesiones, la primera que habla sobre diseño y algunas características generales del código; la segunda, hablará respecto al jugador; la tercera se basará en el invasor, donde cada sesión tendrá la debida descripción de sus errores. Además, se especificará que videojuego se deseó crear.

El docente presentó varias opciones sobre los programas a realizar como exposición del proyecto final, todas las opciones consistían en un tipo de videojuego las cuales eran las siguientes:

- Space Invaders.
- Damas.
- Ajedrez.

Sesión 1. Diseño y generalidades:

Entre el grupo elegimos la opción de realizar y desarrollar el videojuego **Space Invaders**, debido a que era del que más conocimiento y más claridad teníamos sobre cómo era la dinámica del mismo, cuáles eran sus propósitos y de qué forma se podría desarrollar el mismo.

El diseño del mismo se basará en cuatro ventanas, la primera será la de inicio donde se presenta un botón de iniciar el juego y tres niveles (seleccionados por un radiobutton) los cuales aumentan su dificultad de forma ascendente, luego de elegir el nivel el botón de inicio se habilitará y el juego podrá iniciarse.

La segunda ventana, será la que tendrá la ejecución del juego, en esta ventana están todas las propiedades y eventos que se realizaron para su ejecución; en la tercera ventana se mostrará un mensaje, ya sea de victoria o derrota. Además tendrá los botones de reiniciar el juego, volver al menú o cerrar el programa.

El inicio del programa se dio con la escogencia de las imágenes las cuales iban a estar presentes durante la ejecución del mismo, para se eligieron imágenes las cuales fueran acorde con la temática que deseamos realizar, la imagen seleccionada tendría la condición de mantenerse en formato con fondo transparente o png. Para cargar las imágenes de los diferentes personajes se usó la herramienta del PictureBox.

En lo que corresponde al movimiento automático las balas disparadas por el jugador y los enemigos y el movimiento horizontal y vertical de los enemigos, se usó la herramienta de los Timer, los cuales a partir de un lapso de tiempo permiten realizar un cambio en las propiedades que se seleccionen.

El único problema interesante que se presentó fue al momento de cambiar las opciones del juego respecto a los niveles elegidos, debido a que al seleccionar el nivel estas propiedades no cambiaban. El problema se solucionó instanciando la clase en su respectivo radiobutton, es decir que en cualquiera de los tres botones se deberían instanciar la clase al inicio para que de esta forma sus valores se modifiquen.

Sesión 2. Problemas sobre el Jugador:

Iniciando con el código, se inició la búsqueda de cómo realizar el movimiento y disparo de la nave, el cual no tuvo mucha dificultad debido a diversas fuentes en internet y la asesoría del docente. El problema vino, al momento de tener que realizar dos movimientos al mismo tiempo, los cuales eran el disparo de la nave y el movimiento de la ella, debido a que si se presionaba un botón el otro se deshabilitaría.

La solución que encontramos fue la de cambiar el botón de disparo el cual inicialmente estaba configurado con el botón de espacio, luego de cierta investigación se decidió cambiar por el click derecho del mouse, el cual permite que los botones se logren presionar al mismo tiempo, mejorando así la comodidad al momento de jugar el videojuego.

Luego de este apartado, se inició con un tipo de validación para el disparo del jugador, el cual consistía en desaparecer el invasor al momento de que la bala lo golpeará o en otras palabras remover el invasor cuando la bala y el mismo estuviera en las mismas coordenadas.

Acerca de la vida del jugador, este podrá ser interceptado por una bala enemiga tres veces hasta que termine el juego, es decir que si la bala del enemigo golpea al jugador la tercera vez el juego terminará y se dará como perdedor.

Sesión 3. Problemas sobre Invasores y sus disparos:

Con el tema de los invasores si se presentó un grado mayor de dificultad, debido a que todos sus movimientos se realizan de manera automática en el momento de la ejecución, para este apartado se necesitó de una investigación mayor, para lograr mover los invasores y que los mismos dispararan. Además, se necesitaba agrupar a los invasores de forma dinámica.

Iniciando con la agrupación de los invasores, inicialmente se pensó en usar un panel de agrupación con el que se haría más sencillo mover las naves, pero la dificultad se presentó al momento de validar si la bala golpeaba a alguno de estos invasores, debido a que no hayamos la forma de realizarlo con esta herramienta de agrupación.

De allí se buscó una solución por medio de consultas en internet y recomendaciones del docente, la forma que se encontró para terminar con el problema, fue la de cargar PictureBox a partir de matrices y la función foreach. De esta forma se logró hallar la manera de verificar si el PictureBox era interceptado por el disparo del jugador.

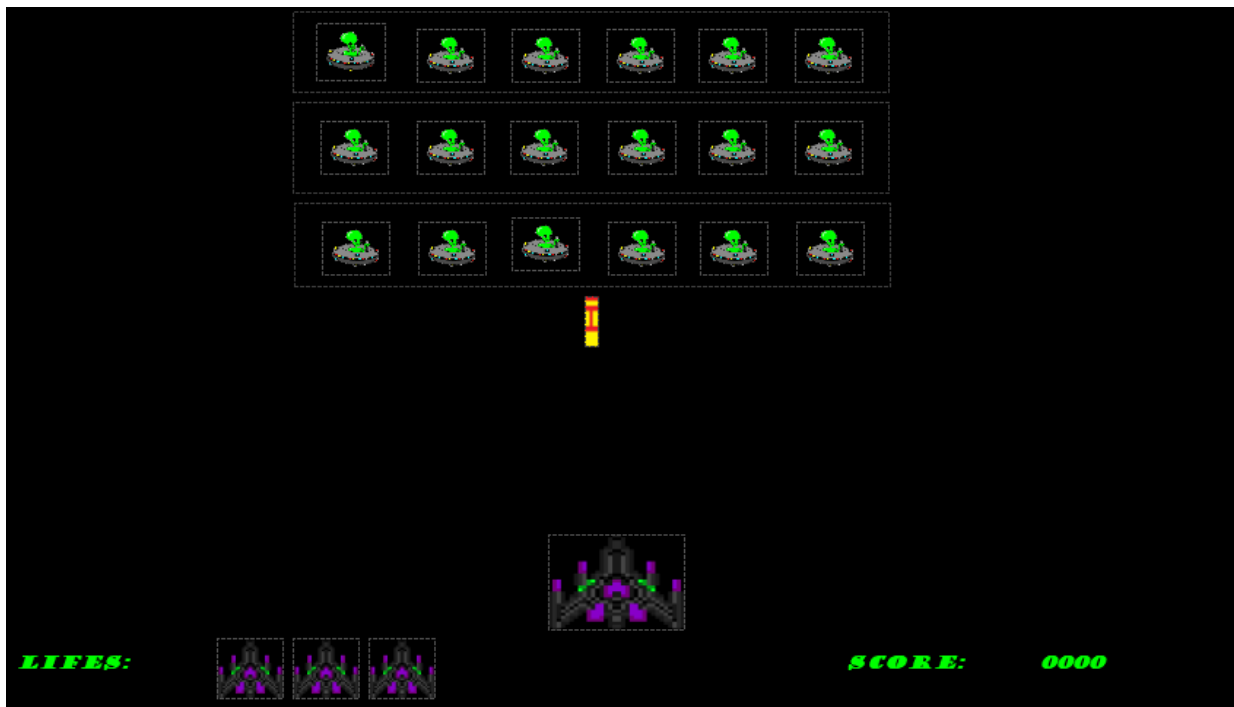
Sobre el movimiento de los invasores, se tuvo un problema y es que solo se lograba mover la última columna y las demás filas y columnas no recibían el movimiento, pero luego de un análisis se logró hallar el error, que estaba solamente en intercambiar la posición en la que se sumaba y restaba las coordenadas de los invasores dentro del timer.

Respecto a las validaciones de los invasores, no se obtuvo una gran dificultad, ya que funcionaría de forma similar a la que se realizó con la nave. Una condición que se tiene es que si los invasores descienden hasta cierto punto, el jugador será derrotado.

A continuación se mostrará una versión antigua del proyecto, donde se tuvieron los primeros errores, para así mejorar y realizarlo de una mejor manera. En otras palabras, la primera versión del programa.

Formulario:

Diseño de la pantalla de juego:



Clase jugador:

```
Form1.cs*  INVADER.cs  PLAYER.cs  Form1.cs [Diseño]*
SpacInvaders_1  naveMovimientoder()

13  public int X;
14  public int Y;
15  public int speed;
16  public int balax;
17  public int balay;
18  public bool seguir;
19  //PictureBox bala = new PictureBox();
20
21
22
23  1 referencia
24  public PLAYER()
25  {
26      X = 0;
27      Y = 0;
28      speed = 20;
29      balax = 0;
30      balay = 0;
31      seguir = true;
32  }
33
34  1 referencia
35  public void naveMovimientoder()
36  {
37      if (seguir == true)
38      {
39          X -= 20;
40          Y = Y;
41      }
42  }
43
44  1 referencia
45  public void naveMovimientoder()
46  {
47      if (seguir == true)
48      {
49          X += 20;
50          Y = Y;
51      }
52  }
53
54  1 referencia
55  public void balaMovimientoder()
56  {
57      //bala.Image = Properties.Resources.bullet;
58      //bala.Size = new Size(5, 20);
59      //bala.Tag = "bala";
60      //bala.Top = X + 20;
61      //this.Controls.Add(bala);
62      //bala.BringToFront();
63
64      balax = X + 47;
65      balay = Y;
66  }
```

Clase Invasor:

```
Form1.cs* INVADER.cs x PLAYER.cs Form1.cs [Diseño]*
SpacInvaders_1 SpacInvaders_1.INVADER invaderMovimientoizq()

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Numerics;
5 using System.Security.Cryptography.Xml;
6 using System.Text;
7 using System.Threading.Tasks;
8
9 namespace SpacInvaders_1
10 {
11     3 referencias
12     internal class INVADER
13     {
14         public int fila1X;
15         public int fila1Y;
16         public int fila2X;
17         public int fila2Y;
18         public int fila3X;
19         public int fila3Y;
20         public int speed;
21         public bool seguir;
22         public int balaX;
23         public int balaY;
24
25         1 referencia
26         public INVADER()
27         {
28             speed = 30;
29             seguir = true;
30         }
31
32         1 referencia
33         public void invaderMovimientoizq()
34         {
35             if (seguir == true)
36             {
37                 fila1X -= 10;
38                 fila1Y = fila1Y;
39                 fila2X -= 10;
40                 fila2Y = fila2Y;
41                 fila3X -= 10;
42                 fila3Y = fila3Y;
43             }
44         }
45     }
46 }
```

```
Form1.cs* INVADER.cs x PLAYER.cs Form1.cs [Diseño]*
SpacInvaders_1 SpacInvaders_1.INVADER invaderMovimientoizq()

49
50 1 referencia
51 public void invaderMovimientoizq()
52 {
53     if (seguir == true)
54     {
55         fila1X += 10;
56         fila1Y = fila1Y;
57         fila2X += 10;
58         fila2Y = fila2Y;
59         fila3X += 10;
60         fila3Y = fila3Y;
61     }
62 }
63
64
65 public void invaderRebote()
66 {
67     if (seguir == true)
68     {
69         fila1X = fila1X;
70         fila1Y += speed;
71         fila2X = fila2X;
72         fila2Y += speed;
73         fila3X = fila3X;
74         fila3Y += speed;
75     }
76 }
77
78
79
80 public void movimientoBala()
81 {
82     if (seguir == true)
83     {
84         //balaX = balaX;
85         balaY = balaY + 20 ;
86     }
87 }
88
89
90
91 }
```

Código de la ventana de juego:

```
Form1.cs*  INVASER.cs  PLAYER.cs  Form1.cs [Diseño]*
SpacInvaders_1  SpacInvaders_1.Form1  timer2_Tick(object sender, EventArgs e)

1 namespace SpacInvaders_1
2 {
3     public partial class Form1 : Form
4     {
5         public Form1()
6         {
7             InitializeComponent();
8             //timer1.Start();
9             //timer2.Start();
10            timer3.Start();
11        }
12
13        PLAYER player = new PLAYER();
14        INVADER invader = new INVADER();
15        Random random = new Random();
16        //Form form2 = new Form();
17
18
19
20
21        1 referencia
22        private void Form1_KeyPress(object sender, KeyPressEventArgs e)
23        {
24        }
25
26        1 referencia
27        private void Form1_KeyDown(object sender, KeyEventArgs e) /*MOVENCION DE LA NAVE*/
28        {
29
30            player.Y = pictureBox1.Location.Y;
31            player.X = pictureBox1.Location.X;
32
33
34            if (player.X > 12) /*MOVENCION HACIA LA IZQUIERDA*/
35            {
36                if (e.KeyCode == Keys.Left)
37                {
38                    player.naveMovimientozq();
39                }
40            }
41
42        }
43    }
44 }
```

```
Form1.cs*  INVASER.cs  PLAYER.cs  Form1.cs [Diseño]*
SpacInvaders_1  SpacInvaders_1.Form1  timer2_Tick(object sender, EventArgs e)

44
45
46            if (player.X < 739) /*MOVENCION HACIA LA DERECHA*/
47            {
48                if (e.KeyCode == Keys.Right)
49                {
50                    player.naveMovimientoder();
51                }
52            }
53
54            if (e.KeyCode == Keys.Space) /*MOVENCION DE LA BALAS*/
55            {
56
57                player.balaMovimiento();
58                pictureBox3.Visible = true;
59                timer3.Enabled = invader.seguir;
60
61                //ConsBulletNave();
62
63            }
64            Point nuevo = new Point(player.X, player.Y);
65            pictureBox1.Location = nuevo;
66            Point nuevo2 = new Point(player.balax, player.balay);
67            pictureBox3.Location = nuevo2;
68
69        }
70
71        //public void ConsBulletNave()
72        //{
73            // PictureBox bala = new PictureBox();
74            // bala.Image = Properties.Resources.bullet;
75            // bala.Size = new Size(5, 20);
76            // bala.Tag = "bala";
77            // bala.Top = player.balay;
78            // this.Controls.Add(bala);
79            // bala.BringToFront();
80        }
81        //}
82
83        1 referencia
84        private void timer1_Tick(object sender, EventArgs e) /*MOVENCION DE LOS ALIENS HACIA LA IZQUIERDA*/
85        {
86            invader.filalX = panelInvader.Location.X;
87            invader.filalY = panelInvader.Location.Y;
88        }
89    }
90 }
```

```
Form1.cs*  INVADER.cs  PLAYER.cs  Form1.cs [Diseño]*
SpacInvaders_1  SpacInvaders_1.Form1  timer2_Tick(object sender, EventArgs e)

88
89 invader.fila2X = panel2Invader.Location.X;
90 invader.fila2Y = panel2Invader.Location.Y;
91
92 invader.fila3X = panel3Invader.Location.X;
93 invader.fila3Y = panel3Invader.Location.Y;
94
95
96 if (invader.fila1X > 12)
97 {
98     invader.invaderMovimientoIzq();
99 }
100
101 else
102 {
103     timer2.Start();
104     timer1.Stop();
105 }
106
107 Point puntoFila1 = new Point(invader.fila1X, invader.fila1Y);
108 Point puntoFila2 = new Point(invader.fila2X, invader.fila2Y);
109 Point puntoFila3 = new Point(invader.fila3X, invader.fila3Y);
110 panel1Invader.Location = puntoFila1;
111 panel2Invader.Location = puntoFila2;
112 panel3Invader.Location = puntoFila3;
113
114
115 }
116
117
118 0 referencias
119 private void pictureBox6_Click(object sender, EventArgs e)
120 {
121 }
122
123
124 1 referencia
125 private void Form1_KeyUp(object sender, KeyEventArgs e)
126 {
127 }
128
129 1 referencia
130 private void timer2_Tick(object sender, EventArgs e) /*MOVER LOS ALIENS HACIA LA DERECHA Y REBOTAR*/
131 {
```

```
Form1.cs*  INVADER.cs  PLAYER.cs  Form1.cs [Diseño]*
SpacInvaders_1  SpacInvaders_1.Form1  timer2_Tick(object sender, EventArgs e)

131
132 invader.fila1X = panel1Invader.Location.X;
133 invader.fila1Y = panel1Invader.Location.Y;
134
135 invader.fila2X = panel2Invader.Location.X;
136 invader.fila2Y = panel2Invader.Location.Y;
137
138 invader.fila3X = panel3Invader.Location.X;
139 invader.fila3Y = panel3Invader.Location.Y;
140
141 if (invader.fila1X < 420)
142 {
143     invader.invaderMovimientoDer();
144 }
145
146 else
147 {
148     timer1.Enabled = true;
149     timer2.Enabled = false;
150     invader.invaderRebote();
151 }
152 Point puntoFila1 = new Point(invader.fila1X, invader.fila1Y);
153 panel1Invader.Location = puntoFila1;
154 Point puntoFila2 = new Point(invader.fila2X, invader.fila2Y);
155 panel2Invader.Location = puntoFila2;
156 Point puntoFila3 = new Point(invader.fila3X, invader.fila3Y);
157 panel3Invader.Location = puntoFila3;
158
159 if (invader.fila1Y + 195 >= player.Y && invader.fila1Y + 195 <= player.Y + 66) /*VERIFICA GAME OVER*/
160 {
161     timer1.Enabled = false;
162     timer2.Enabled = false;
163     invader.seguir = false;
164     player.seguir = false;
165     pictureBox27.Visible = false;
166     pictureBox28.Visible = false;
167     pictureBox29.Visible = false;
168 }
169
170
171 //if (invader.fila2Y + 195 >= player.Y && invader.fila2Y + 195 <= player.Y + 66) /*VERIFICA GAME OVER*/
172 //
173 // if (invader.fila3Y + 195 >= player.Y && invader.fila3Y + 195 <= player.Y + 66)
174 // timer1.Enabled = false;
175 // timer2.Enabled = false;
176 // invader.seguir = false;
```

```
Form1.cs* x INVADER.cs PLAYER.cs Form1.cs [Diseño]*
SpacInvaders_1 SpacInvaders_1.Form1 timer2_Tick(object sender, EventArgs e)

176 // invader.seguir = false;
177 // player.seguir = false;
178 // pictureBox27.Visible = false;
179 // pictureBox28.Visible = false;
180 // pictureBox29.Visible = false;
181
182 //}
183
184 //if (invader.filasY + 195 >= player.Y && invader.filasY + 195 <= player.Y + 66) /*VERIFICA GAME OVER*/
185 //{
186 //    timer1.Enabled = false;
187 //    timer2.Enabled = false;
188 //    invader.seguir = false;
189 //    player.seguir = false;
190 //    pictureBox27.Visible = false;
191 //    pictureBox28.Visible = false;
192 //    pictureBox29.Visible = false;
193 //}
194
195
196
197
198
199
200 0 referencias
201 private void pictureBox3_Click(object sender, EventArgs e)
202 {
203 }
204
205 1 referencia
206 private void timer3_Tick(object sender, EventArgs e) /*MOVIMIENTO BALAS*/
207 {
208     player.balax = pictureBox3.Location.X;
209     player.balay = pictureBox3.Location.Y;
210
211     if (player.balay >= 0)
212     {
213         player.balax = pictureBox3.Location.X;
214         player.balay -= 80;
215     }
216     else
217     {
218         pictureBox3.Visible = false;
219         timer3.Enabled = false;
220     }
221 }
```

```
Form1.cs* x INVADER.cs PLAYER.cs Form1.cs [Diseño]*
SpacInvaders_1 SpacInvaders_1.Form1 timer2_Tick(object sender, EventArgs e)

219 }
220 Point punto2 = new Point(player.balax, player.balay);
221 pictureBox3.Location = punto2;
222
223
224
225
226 1 referencia
227 private void label1_Click(object sender, EventArgs e)
228 {
229 }
230
231 1 referencia
232 private void Form1_Load(object sender, EventArgs e)
233 {
234 }
235
236 1 referencia
237 private void panellInvader_Paint(object sender, PaintEventArgs e)
238 {
239 }
240
241 0 referencias
242 private void panel4Invader_Paint(object sender, PaintEventArgs e)
243 {
244 }
245
246 1 referencia
247 private void timer4_Tick(object sender, EventArgs e)
248 {
249     //invader.balax = pictureBox7.Location.X;
250     invader.balay = pictureBox7.Location.Y;
251
252     if (invader.balay < 461)
253     {
254         invader.movimientoBala();
255     }
256     else
257     {
258         invader.balax = random.Next(12, 823);
259         invader.balay = 0;
260     }
261 }
```



```

Form1.cs*  INVASER.cs  PLAYER.cs  Form1.cs [Diseño]*
SpaceInvaders_1
timer2_Tick(object sender, EventArgs e)

239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275

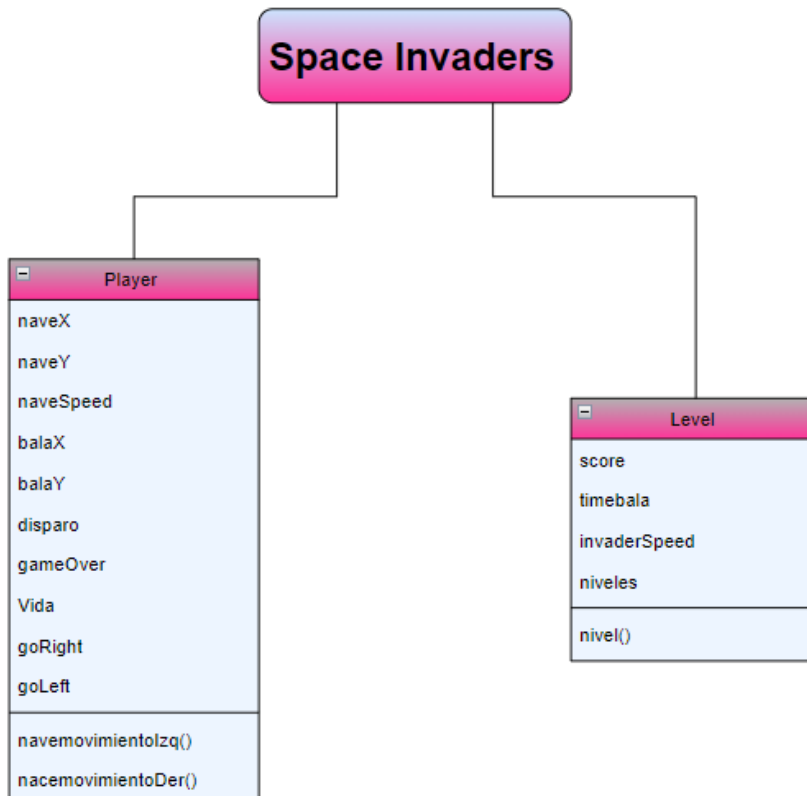
0 referencias
private void panelInvader_Paint(object sender, PaintEventArgs e)
{
}

1 referencias
private void timer4_Tick(object sender, EventArgs e)
{
    //invader.balaX = pictureBox7.Location.X;
    invader.balaY = pictureBox7.Location.Y;
    if (invader.balaY < 461)
    {
        invader.movimientoBala();
    }
    else
    {
        invader.balaX = random.Next(12, 823);
        invader.balaY = 0;
    }
    Point balaEnemigo = new Point(invader.balaX, invader.balaY);
    pictureBox7.Location = balaEnemigo;

    if (invader.balaY + 35 >= player.Y && invader.balaY + 35 <= player.Y + 66)
    {
        if (invader.balaX + 5 >= player.X && invader.balaX + 47 <= player.X + 5)
        {
            pictureBox27.Visible = false;
        }
    }
}
}

```

Diagrama UML:



REFERENCIAS:

[How to make a simple space invaders style game in windows form with C# coding](#)

[Movimiento de PictureBox de izquierda derecha en C#](#)

[Movimiento de PictureBox con el Teclado](#)

[COMO PASAR DATOS DE UN FORMULARIO A OTRO EN C# !\[\]\(0f848bbd71cef6b345273b16f905912a_img.jpg\) \[FACIL y SENCILLO\]](#)