

Trabajo Práctico Final de complejidad temporal estructura de datos y algoritmos

Carrera: Ingeniería Informática

TEMA: Desarrollo de un juego

PROFESOR: Alejandro Fontan

ALUMNO: Víctor Balbuena

COMISION: 4

AÑO: 2020

Detalles de implementación:

Para la realización del trabajo se utilizó la estructura de datos vista durante la cursada en este caso el árbol general, a modo de facilitar la codificación del mismo se le proveyó al alumno todas las clases que se necesitan para finalizar el trabajo práctico así como también la interfaz gráfica para poder visualizar los cambios realizados.

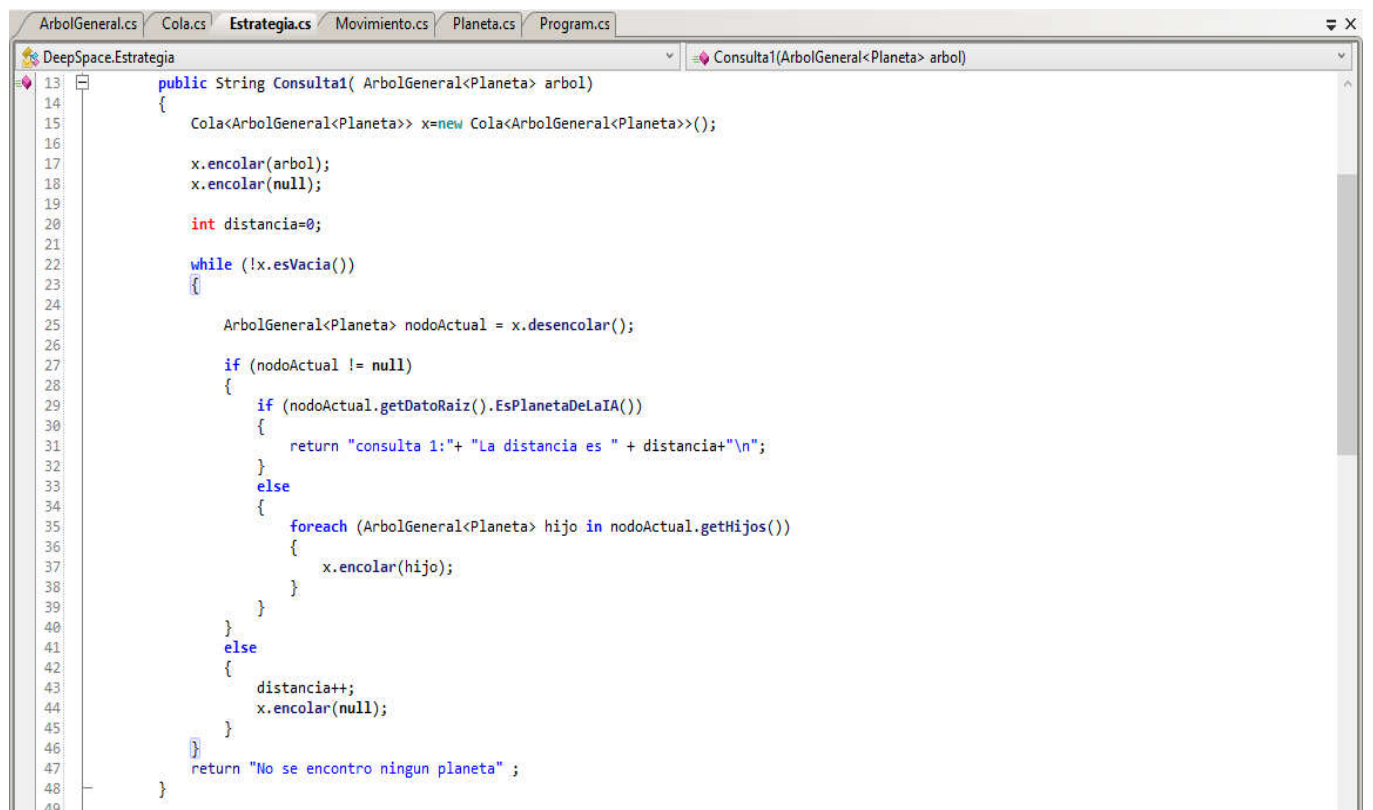
La implementación que se debe llevar a cabo es en la clase estrategia del proyecto en la cual el alumno debe completar los métodos definidos por el profesor en este caso son los de: (Calcular movimiento, consulta1, consulta2, consulta3). Cada uno de estos métodos representan el movimiento que debería realizar el bot del juego para ganar.

Todo el código está realizado en el lenguaje c# con lo cual para su correcta ejecución se debe hacer uso de un IDE el cual cuente con soporte para este lenguaje como por ejemplo Visual Studio o Sharp Develop en este caso se utilizó el segundo el cual logra ejecutarlo sin ningún inconveniente.


Durante la ejecución del proyecto se produce una excepción en la siguiente línea de código (game.target.**DrawTextLayout**(position, textLayout, brush);) la cual es producida al momento de presionar el botón de inicio, salir al menú principal hacer una consulta y volver al juego con lo cual el programa debe finalizarse.

Imágenes del código

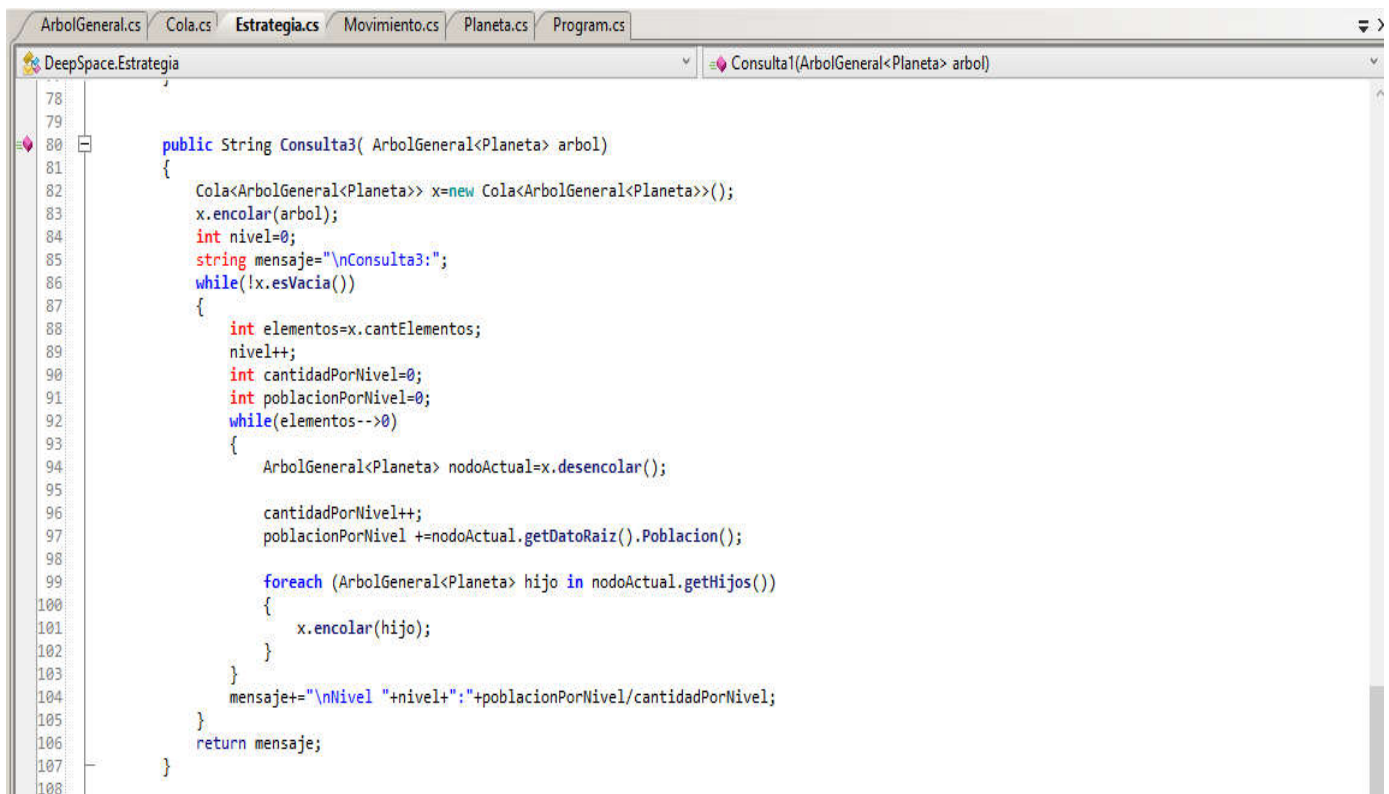
Clase estrategia es la clase donde había que implementar los métodos de consultas



```
ArbolGeneral.cs Cola.cs Estrategia.cs Movimiento.cs Planeta.cs Program.cs
DeepSpace.Estrategia Consultar1(ArbolGeneral<Planeta> arbol)
13 public String Consultar1( ArbolGeneral<Planeta> arbol)
14 {
15     Cola<ArbolGeneral<Planeta>> x=new Cola<ArbolGeneral<Planeta>>();
16
17     x.encolar(arbol);
18     x.encolar(null);
19
20     int distancia=0;
21
22     while (!x.esVacia())
23     {
24
25         ArbolGeneral<Planeta> nodoActual = x.desencolar();
26
27         if (nodoActual != null)
28         {
29             if (nodoActual.getDatoRaiz().EsPlanetaDeLaIA())
30             {
31                 return "consulta 1:" + "La distancia es " + distancia+"\n";
32             }
33             else
34             {
35                 foreach (ArbolGeneral<Planeta> hijo in nodoActual.getHijos())
36                 {
37                     x.encolar(hijo);
38                 }
39             }
40         }
41         else
42         {
43             distancia++;
44             x.encolar(null);
45         }
46     }
47     return "No se encontro ningun planeta" ;
48 }
49
```



```
ArbolGeneral.cs Cola.cs Estrategia.cs Movimiento.cs Planeta.cs Program.cs
DeepSpace.Estrategia Consultar2(ArbolGeneral<Planeta> arbol)
49
50
51 public String Consultar2( ArbolGeneral<Planeta> arbol)
52 {
53     Cola<ArbolGeneral<Planeta>> x=new Cola<ArbolGeneral<Planeta>>();
54     x.encolar(arbol);
55     int nivel=0;
56     string mensaje="\nConsulta2:";
57     while(!x.esVacia())
58     {
59         int elementos=x.cantElementos;
60         nivel++;
61         int cantidadPorNivel=0;
62         while(elementos-->0)
63         {
64             ArbolGeneral<Planeta> nodoActual=x.desencolar();
65
66             if (nodoActual.getDatoRaiz().Poblacion())>10) {
67                 cantidadPorNivel++;
68             }
69             foreach (ArbolGeneral<Planeta> hijo in nodoActual.getHijos())
70             {
71                 x.encolar(hijo);
72             }
73         }
74         mensaje+="Nivel "+nivel+": "+cantidadPorNivel+"-";
75     }
76     return mensaje;
77 }
78
```



```
78
79
80 public String Consulta3( ArbolGeneral<Planeta> arbol)
81 {
82     Cola<ArbolGeneral<Planeta>> x=new Cola<ArbolGeneral<Planeta>>();
83     x.encolar(arbol);
84     int nivel=0;
85     string mensaje="\nConsulta3:";
86     while(!x.esVacia())
87     {
88         int elementos=x.cantElementos;
89         nivel++;
90         int cantidadPorNivel=0;
91         int poblacionPorNivel=0;
92         while(elementos-->0)
93         {
94             ArbolGeneral<Planeta> nodoActual=x.desencolar();
95
96             cantidadPorNivel++;
97             poblacionPorNivel +=nodoActual.getDatoRaiz().Poblacion();
98
99             foreach (ArbolGeneral<Planeta> hijo in nodoActual.getHijos())
100             {
101                 x.encolar(hijo);
102             }
103         }
104         mensaje+="\nNivel "+nivel+" "+poblacionPorNivel/cantidadPorNivel;
105     }
106     return mensaje;
107 }
108
```

Conclusión

Con la realización de este trabajo practico se puede observar como los conocimientos adquiridos durante la cursada se pueden implementar en un determinado proyecto, específicamente se puede ver el funcionamiento de un árbol el uso de los nodos y como recorrerlo para obtener información.