

# **PRÁCTICA 2**

## **INTELIGENCIA ARTIFICIAL**



**UNIVERSIDAD  
DE GRANADA**

**DAVID MUÑOZ SÁNCHEZ**

## Comportamiento implementado

Se han implementado los niveles 1 y 2 de la práctica así como el tutorial.

El nivel 1 es muy parecido al tutorial, ya que este implementa la búsqueda en profundidad y ahora tenemos que implementar la búsqueda en anchura, es decir, tenemos que cambiar la pila de abiertos (LIFO), por una cola de abiertos (FIFO).

```
cout << "Calculando plan\n";
plan.clear();
set<estado, ComparaEstados> Cerrados; // Lista de Cerrados
queue<nodo> Abiertos;                  // Lista de Abiertos
```

Para el nivel 2 sí que he necesitado más modificaciones. La esencia del algoritmo es muy parecida a los anteriores, pero ahora tenemos que tratar la cuestión del bikini y las zapatillas. En la función que calcula el algoritmo, antes de generar descendientes, se observa si se puede activar las zapatillas o el bikini en el nodo, para a la hora de calcular los descendientes, se tenga en cuenta esta cuestión, ya que la función `calculoCoste`, hace distinción según el objeto que se posea y no se nos puede olvidar que estamos minimizando el gasto de batería para llegar a un objetivo. Antes de introducir el estado en abiertos, se calcula el coste, para el que se ha implementado la función **`calculoCoste(int fila, int columna, bool bikini, bool zapatillas, int accion)`** en la que se tiene en cuenta la tabla proporcionada en el guión de esta práctica en referencia a los gastos de batería según el terreno, los movimientos y los objetos que se posean.

```
int ComportamientoJugador::calculoCoste(int fila, int columna, bool bikini, bool
zapatillas, int accion)
{
    char letra_nodo = mapaResultado[fila][columna];
    int coste = 0;

    if (accion != 6)
    {
        coste = 1;
    } //No es idle

    if (letra_nodo == 'A')
    {
        if (accion == 0)
        { //forward
            coste = 200;
            if (bikini)
            {
                coste = 10;
            }
        }
    }
}
```

```

    }
}
//giros
if (accion == 1 || accion == 2)
{
    coste = 500;
    if (bikini)
    {
        coste = 5;
    }
}
//semigiros
if (accion == 3 || accion == 4)
{
    coste = 300;
    if (bikini)
    {
        coste = 2;
    }
}
}

else if (letra_nodo == 'B')
{
    if (accion == 0)
    {
        coste = 100;
        if (zapatillas)
        {
            coste = 15;
        }
    }
    if (accion == 1 || accion == 2)
    {
        coste = 3;
        if (zapatillas)
            coste = 1;
    }

    if (accion == 3 || accion == 4)
    {
        coste = 2;
        if (zapatillas)
        {
            coste = 1;
        }
    }
}

else if (letra_nodo == 'T')
{

```

```

//Todo lo que no sea moverse en diagonal en la tierra vale 2 por defecto
coste = 2;

if (accion == 3 || accion == 4)
{
    coste = 1;
}

return coste;
}

```

Por último, se ha creado una struct parecida a ComparaEstados, pero que tiene en cuenta si se tiene o no bikini o zapatillas para hacer esta comparación.

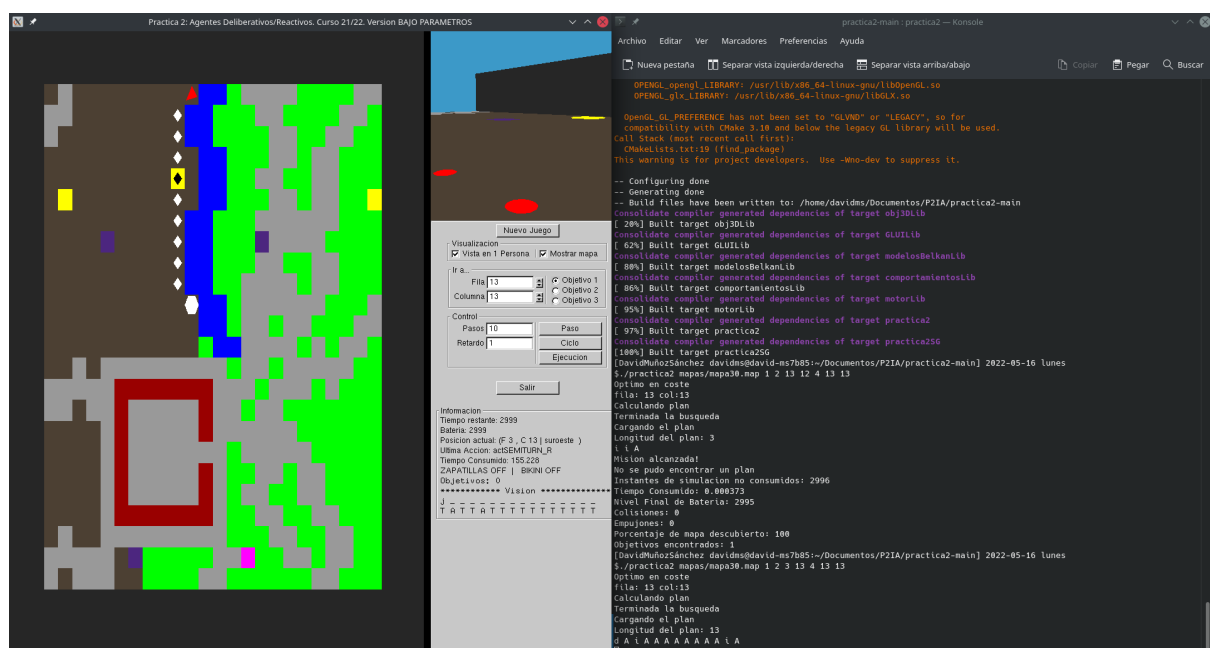
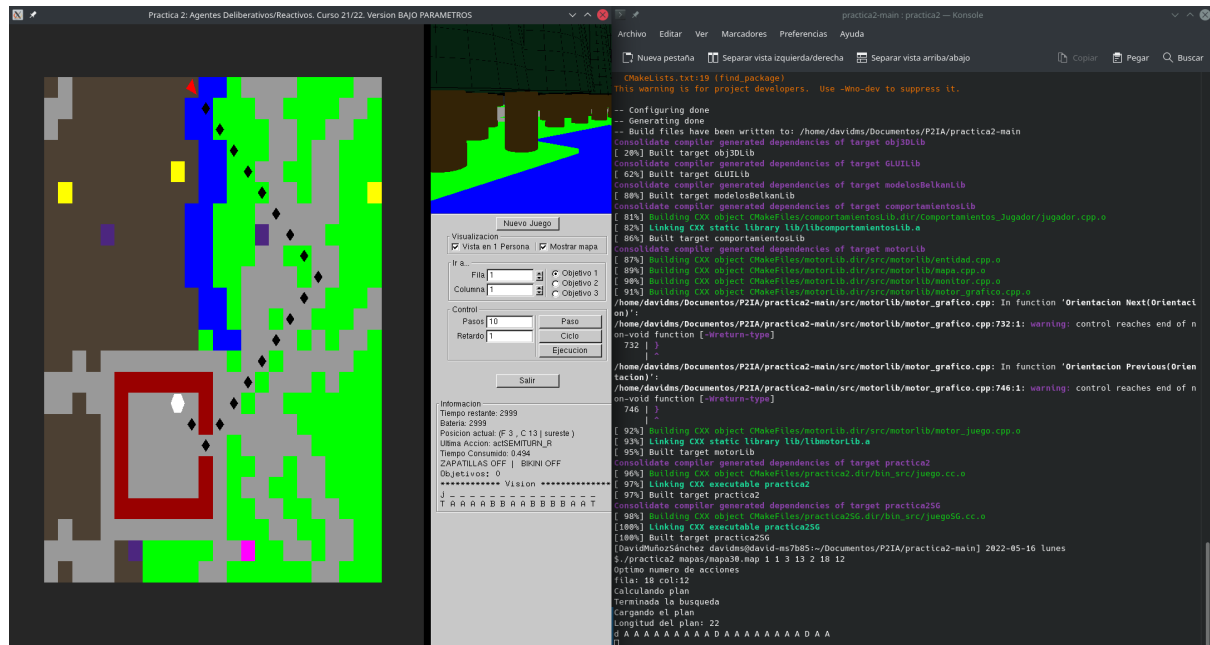
```

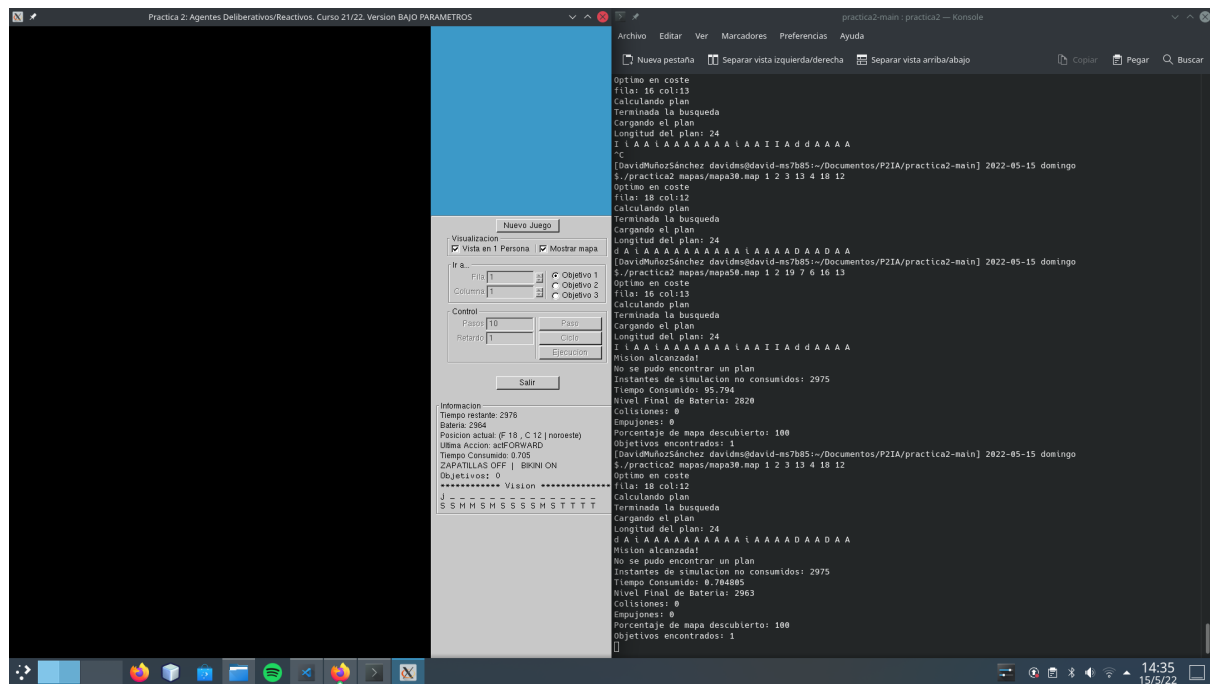
struct ComparaEstadosCU
{
    bool operator()(const estado &a, const estado &n) const
    {
        if ((a.fila > n.fila) or (a.fila == n.fila and a.columna > n.columna) or
            (a.fila == n.fila and a.columna == n.columna and a.orientacion >
n.orientacion) or
            (a.fila == n.fila and a.columna == n.columna and a.orientacion ==
n.orientacion and
            (!a.bikini and n.bikini) or (a.bikini and !n.bikini) and
a.zapatillas == n.bikini) or
            (a.fila == n.fila and a.columna == n.columna and a.orientacion ==
n.orientacion and
            a.bikini == n.bikini and
            (!a.zapatillas and n.zapatillas) or (a.zapatillas and
!n.zapatillas)))
            return true;
        else
            return false;
    }
};

```

## Ejemplos ejecución

A continuación, se muestran varios ejemplos de ejecución en base a los ejemplos proporcionados en Prado.





En este caso la ruta es diferente a la de los ejemplos, pero el gasto de batería es igual (se queda en 2821), si miramos la interfaz gráfica.