

UTEC - Ciencias de la Computación
Programación Orientada a Objetos I
2019-1 Fecha Elaboración: 14/05/2019
PROYECTO 1

Profesor: Rubén Rivas

1. DESCRIPCIÓN DEL PROYECTO

Diseñar y desarrollar un programa que permita reproducir el juego Senku es un juego de tablero que con una distribución como se muestra en la figura, inicialmente todos los espacios, representado por los vértices de los cuadrados, están ocupados, excepto el espacio central que se encuentra vacío, los espacios ocupados se representan por O y los espacios vacíos se representado por +. La regla del juego es mover los espacios ocupados hacia los espacios vacíos, pero para moverlo es requisito obligatorio saltar sobre solo un espacio que este ocupado, de otro modo el espacio ocupado no podrá moverse. Se gana el juego si al final se queda con solo un espacio ocupado y se pierde si no se puede realizar más movimientos y se tiene más de un espacio ocupado, los movimientos solo son verticales u horizontales, pero no diagonales.

1)

```
{ ' ', ' ', ' ', 'O', 'O', 'O', ' ', ' ', ' ' },
{ ' ', 'O', 'O', 'O', 'O', 'O', ' ', ' ', ' ' },
{ 'O', 'O', 'O', '+', 'O', 'O', 'O', ' ', ' ' },
{ 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', ' ' },
{ 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', ' ' },
{ ' ', 'O', 'O', 'O', 'O', 'O', 'O', ' ', ' ' },
{ ' ', ' ', ' ', 'O', 'O', 'O', ' ', ' ', ' ' },
```

2)

```
{ ' ', ' ', ' ', ' ', 'O', 'O', 'O', ' ', ' ', ' ', ' ' },
{ ' ', ' ', ' ', ' ', 'O', 'O', 'O', ' ', ' ', ' ', ' ' },
{ ' ', ' ', ' ', ' ', 'O', 'O', 'O', ' ', ' ', ' ', ' ' },
{ 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', ' ' },
{ 'O', 'O', 'O', 'O', '+', 'O', 'O', 'O', 'O', 'O', ' ' },
{ 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', ' ' },
{ ' ', ' ', ' ', ' ', 'O', 'O', 'O', ' ', ' ', ' ', ' ' },
{ ' ', ' ', ' ', ' ', 'O', 'O', 'O', ' ', ' ', ' ', ' ' },
{ ' ', ' ', ' ', ' ', 'O', 'O', 'O', ' ', ' ', ' ', ' ' },
```

3)

```
{ ' ', ' ', ' ', 'O', 'O', 'O', ' ', ' ', ' ' },
{ ' ', ' ', ' ', 'O', 'O', 'O', ' ', ' ', ' ' },
{ 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', ' ' },
{ 'O', 'O', 'O', '+', 'O', 'O', 'O', 'O', ' ' },
{ 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', ' ' },
{ ' ', ' ', ' ', 'O', 'O', 'O', ' ', ' ', ' ' },
{ ' ', ' ', ' ', 'O', 'O', 'O', ' ', ' ', ' ' },
```

Programación Orientada a Objetos I

Estilos de juego:

- Estilo Francés
- Estilo Alemán
- Estilo Inglés

Menú del programa:

```
int Menu() {
    int Selec;
    cout << "  Menu - Juego senku" << endl;
    cout << "  -----" << endl;
    cout << "  1. Estilo Frances" << endl;
    cout << "  2. Estilo Aleman" << endl;
    cout << "  3. Estilo Ingles" << endl;
    cout << "  -----" << endl;
    cout << "  0. Salir del programa" << endl << endl;
    cout << "  Seleccionar la opcion:" << endl;
    cin >> Selec;
    switch (Selec) {
    case 1:case 2:case 3:case 0:
        return Selec;
        break;
    default:
        cout << "  *****\n";
        cout << "  * > Opcion invalida < *\n";
        cout << "  *****\n\n";
        Menu();
    }
}
```

Main.cpp

Contenido:

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
#include <ctime>
#include <cstdlib>
#include <ctime>
#include <cstdlib>
#include <typeinfo>
#include "Funciones.h"
using namespace std;

int main() {
    int Selec;
    char M_ingles[7][7] = { {' ', ' ', ' ', '0', '0', '0', ' ', ' ', ' '},
                           {' ', ' ', ' ', '0', '0', '0', ' ', ' ', ' '},
                           {'0', '0', '0', '0', '0', '0', '0', '0', '0'},
                           {'0', '0', '0', '+', '0', '0', '0', '0', '0'},
                           {'0', '0', '0', '0', '0', '0', '0', '0', '0'},
                           {' ', ' ', ' ', '0', '0', '0', ' ', ' ', ' '},
                           {' ', ' ', ' ', '0', '0', '0', ' ', ' ', ' '}};

    char M_aleman[9][9] = { {' ', ' ', ' ', ' ', '0', '0', '0', ' ', ' ', ' '},
                           {' ', ' ', ' ', ' ', '0', '0', '0', ' ', ' ', ' '},
                           {' ', ' ', ' ', ' ', '0', '0', '0', ' ', ' ', ' '},
                           {'0', '0', '0', '0', '0', '0', '0', '0', '0', '0'},
                           {'0', '0', '0', '0', '+', '0', '0', '0', '0', '0'},
                           {'0', '0', '0', '0', '0', '0', '0', '0', '0', '0'},
                           {' ', ' ', ' ', ' ', '0', '0', '0', ' ', ' ', ' '},
                           {' ', ' ', ' ', ' ', '0', '0', '0', ' ', ' ', ' '},
                           {' ', ' ', ' ', ' ', '0', '0', '0', ' ', ' ', ' '}};

    char M_frances[7][7] = {{{' ', ' ', ' ', '0', '0', '0', ' ', ' ', ' '},
                              {' ', '0', '0', '0', '0', '0', ' ', ' ', ' '},
                              {'0', '0', '0', '+', '0', '0', '0', '0', '0'},
                              {'0', '0', '0', '0', '0', '0', '0', '0', '0'},
                              {'0', '0', '0', '0', '0', '0', '0', '0', '0'},
                              {' ', '0', '0', '0', '0', '0', ' ', ' ', ' '},
                              {' ', ' ', ' ', '0', '0', '0', ' ', ' ', ' '}}};

    Selec = Menu();
    switch (Selec) {
        case 1:
            Juego_Frances(M_frances);
            break;
        case 2:
            Juego_Aleman(M_aleman);
            break;
        case 3:
            Juego_Ingles(M_ingles);
            break;
        case 0:
    }
```

```

        return 0;
        break;
    }
}

```

Funciones.cpp

Contenido:

```

#include <iostream>
#include "Funciones.h"
using namespace std;

int Menu() {
    int Selec;
    cout << "  Menu - Juego senku" << endl;
    cout << "  -----" << endl;
    cout << "  1. Estilo Frances" << endl;
    cout << "  2. Estilo Aleman" << endl;
    cout << "  3. Estilo Ingles" << endl;
    cout << "  -----" << endl;
    cout << "  0. Salir del programa" << endl << endl;
    cout << "  Seleccionar la opcion:" << endl;
    cin >> Selec;
    switch (Selec) {
        case 1:case 2:case 3:case 0:
            return Selec;
            break;
        default:
            cout << "  *****\n";
            cout << "  * > Opcion invalida < *\n";
            cout << "  *****\n\n";
            Menu();
    }
}

//INGLES
int Juego_Ingles(char Matriz[7][7]) {
    int ZZZ = 0, Cont1, Cont2,Xi,Yi,Xf,Yf;
    while(ZZZ != 1) {
        for (int i = 1; i <= 7; i++) { cout << "  " << i << "  "; }
        cout << endl;
        for (int y = 1; y <= 7; y++) {
            cout << (y) << "  ";
            switch (y) {
                case 1:case 2:case 6:case 7:
                    Cont1 = 2;
                    break;
                default:
                    Cont1 = 6;
            }
            for (int x = 1; x <= 7; x++) {
                cout << Matriz[y - 1][x - 1];
                if (Matriz[y - 1][x - 1] == ' ') {
                    cout << "  ";
                }
            }
        }
    }
}

```

```

        else if (Cont1 > 0) {
            cout << "-----";
            Cont1--;
        }
    }
    if (y != 7) {
        cout << endl << " ";

        switch (y) {
            case 1: case 2: case 5: case 6:
                Cont2 = 3;
                cout << "          ";
                break;
            default:
                Cont2 = 7;
        }
        for (int k = 1; k <= Cont2; k++) {
            cout << "| ";
        }
        cout << endl;
    }
    //En proceso (Moviento) >>>
    bool V1 = true;
    while (V1 == true) {
        cout << endl << "Ingrese la posicion de origen (fila, columna):
" << endl;

        cin >> Yi >> Xi;
        if (Matriz[Yi - 1][Xi - 1] == 'O') {
            V1 = false;
        }
        else { cout << ">>>Movimiento invalido<<<" << endl; }
    }
    V1 = true;
    while (V1 == true) {
        cout << "Ingrese la posicion de destino (fila, columna): " <<
endl;

        cin >> Yf >> Xf;
        if (Matriz[Yf - 1][Xf - 1] == '+') {
            if (Xi + 2 == Xf || Xi - 2 == Xf || Xi == Xf) {
                if (Yi + 2 == Yf || Yi - 2 == Yf || Yi == Yf) {
                    V1 = false;
                }
            }
        }
        else { cout << ">>>Movimiento invalido<<<" << endl; }
    }
    V1 = true;
    Matriz[Yi - 1][Xi - 1] = '+';
    Matriz[Yf - 1][Xf - 1] = 'O';
    if (Yi == Yf) {
        if (Xf > Xi) {
            Matriz[Yi - 1][Xi] = '+';
        }
        else if (Xf < Xi) {
            Matriz[Yi - 1][Xi - 2] = '+';
        }
    }
}

```

```

        if (Xi == Xf) {
            if (Yf > Yi) {
                Matriz[Yi][Xi - 1] = '+';
            }
            else if (Yf < Yi) {
                Matriz[Yi - 2][Xi - 1] = '+';
            }
        }

        // <<< En proceso (Movimiento)
    }
    return 0;
}

//ALEMAN
int Juego_Aleman(char Matriz[9][9]) {
    int ZZZ = 0, Cont1, Cont2, Xi, Yi, Xf, Yf;
    while (ZZZ != 1) {
        for (int i = 1; i <= 9; i++) { cout << " " << i << " "; }
        cout << endl;
        for (int y = 1; y <= 9; y++) {
            cout << (y) << " ";
            switch (y) {
                case 1:case 2:case 3:case 7:case 8:case 9:
                    Cont1 = 2;
                    break;
                default:
                    Cont1 = 8;
            }
            for (int x = 1; x <= 9; x++) {
                cout << Matriz[y - 1][x - 1];
                if (Matriz[y - 1][x - 1] == ' ') {
                    cout << " ";
                }
                else if (Cont1 > 0) {
                    cout << "-----";
                    Cont1--;
                }
            }
            if (y != 9) {
                cout << endl << " ";

                switch (y) {
                    case 1:case 2:case 3:case 6:case 7:case 8:
                        Cont2 = 3;
                        cout << " ";
                        break;
                    default:
                        Cont2 = 9;
                }
                for (int k = 1; k <= Cont2; k++) {
                    cout << " | ";
                }
            }
            cout << endl;
        }
        //En proceso (Movimiento) >>>
    }
}

```

```

        bool V1 = true;
        while (V1 == true) {
            cout << endl << "Ingrese la posicion de origen (fila, columna):
" << endl;

            cin >> Yi >> Xi;
            if (Matriz[Yi - 1][Xi - 1] == 'O') {
                V1 = false;
            }
            else { cout << ">>>Movimiento invalido<<<" << endl; }
        }
        V1 = true;
        while (V1 == true) {
            cout << "Ingrese la posicion de destino (fila, columna): " <<
endl;

            cin >> Yf >> Xf;
            if (Matriz[Yf - 1][Xf - 1] == '+') {
                if (Xi + 2 == Xf || Xi - 2 == Xf || Xi == Xf) {
                    if (Yi + 2 == Yf || Yi - 2 == Yf || Yi == Yf) {
                        V1 = false;
                    }
                }
            }
            else { cout << ">>>Movimiento invalido<<<" << endl; }
        }
        V1 = true;
        Matriz[Yi - 1][Xi - 1] = '+';
        Matriz[Yf - 1][Xf - 1] = 'O';
        if (Yi == Yf) {
            if (Xf > Xi) {
                Matriz[Yi - 1][Xi] = '+';
            }
            else if (Xf < Xi) {
                Matriz[Yi - 1][Xi - 2] = '+';
            }
        }
        if (Xi == Xf) {
            if (Yf > Yi) {
                Matriz[Yi][Xi - 1] = '+';
            }
            else if (Yf < Yi) {
                Matriz[Yi - 2][Xi - 1] = '+';
            }
        }

        // <<< En proceso (Movimiento)
    }
    return 0;
}

```

//FRANCES

```

int Juego_Francis(char Matriz[7][7]) {
    int ZZZ = 0, Cont1, Cont2, Xi, Yi, Xf, Yf;
    while (ZZZ != 1) {
        for (int i = 1; i <= 7; i++) { cout << " " << i << " "; }
        cout << endl;
        for (int y = 1; y <= 7; y++) {
            cout << (y) << " ";

```

```

switch (y) {
case 1:case 7:
    Cont1 = 2;
    break;
case 2:case 6:
    Cont1 = 4;
    break;
default:
    Cont1 = 6;
}
for (int x = 1; x <= 7; x++) {
    cout << Matriz[y - 1][x - 1];
    if (Matriz[y - 1][x - 1] == ' ') {
        cout << " ";
    }
    else if (Cont1 > 0) {
        cout << "-----";
        Cont1--;
    }
}
if (y != 7) {
    cout << endl << " ";

    switch (y) {
case 1:case 6:
        Cont2 = 3;
        cout << " ";
        break;
case 2:case 5:
        Cont2 = 5;
        cout << " ";
        break;
default:
        Cont2 = 7;
    }
    for (int k = 1; k <= Cont2; k++) {
        cout << "| ";
    }
    cout << endl;
}
//En proceso (Moviento) >>>
bool V1 = true;
while (V1 ==true) {
    cout << endl << "Ingrese la posicion de origen (fila, columna):
" << endl;

    cin >> Yi >> Xi;
    if (Matriz[Yi - 1][Xi - 1] == '0') {
        V1=false;
    }
    else { cout << ">>>Movimiento invalido<<<" << endl; }
}
V1 = true;
while (V1 ==true) {
    cout << "Ingrese la posicion de destino (fila, columna): " <<
endl;

    cin >> Yf >> Xf;
    if (Matriz[Yf - 1][Xf - 1] == '+') {

```



```

        if (Xi + 2 == Xf || Xi - 2 == Xf || Xi == Xf) {
            if (Yi + 2 == Yf || Yi - 2 == Yf || Yi == Yf) {
                V1 = false;
            }
        }
    }
    else { cout << ">>>Movimiento invalido<<<" << endl; }
}
V1 = true;
Matriz[Yi - 1][Xi - 1] = '+';
Matriz[Yf - 1][Xf - 1] = 'O';
if (Yi == Yf) {
    if (Xf > Xi) {
        Matriz[Yi - 1][Xi] = '+';
    }
    else if (Xf < Xi) {
        Matriz[Yi - 1][Xi - 2] = '+';
    }
}
if (Xi == Xf) {
    if (Yf > Yi) {
        Matriz[Yi][Xi - 1] = '+';
    }
    else if (Yf < Yi) {
        Matriz[Yi - 2][Xi - 1] = '+';
    }
}

// <<< En proceso (Movimiento)
}
return 0;
}

```

Funciones.h

Contenido:

```

#ifndef POO_FUNCIONES_H
#define POO_FUNCIONES_H

int Menu();
int Juego_Ingles(char Matriz[7][7]);
int Juego_Aleman(char Matriz[9][9]);
int Juego_Frances(char Matriz[7][7]);

#endif //POO_FUNCIONES_H

```