Senku

César Bracamonte
Raúl Suarez
Maria José Linares

¿Qué es Senku?

- Senku es un juego de tablero solitario.
- El objetivo del Senku consiste en lograr a capturar todas las piezas hasta que quede solamente una.
- Al inicio del juego están todos los espacios ocupados excepto el central y el cual espacio tendrá que ser ocupado con el último movimiento.
- Las piezas se pueden mover solo capturando, la cual captura se realiza mediante el salto como con las Damas.

Inicio

Debe elegir uno de los tres tipos de juego.

Para que el usuario sepa que estilo es cada uno, se proporcionó las impresiones de diferentes tableros.

```
Bienvenidos a Senku! Escoge el tablero con el que quieras jugar.
1. Estilo Ingles.

      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

2. Estilo Aleman.

        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

3. Estilo Frances.

      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. Salir del programa

Seleccionar la opcion:
```

Código

Se usaron los archivos
main.cpp
senku.cpp
senku.h

main.cpp

```cpp
#include ...
using namespace std;
int main() {
    int nrow=0,ncol=0;
    char**tablero= nullptr;
    int tab;
    int termina=0;
    tipomatriz tablero1[9][9];
    tipomatriz tablero2[11][11];
    tipomatriz tablero3[9][9];
    //juego
    tab=menu_inicial(tablero1,tablero2,tablero3,nrow,ncol);
    //declarar tablero
    tablero=new char*[nrow];
    for (size_t i = 0; i <nrow ; ++i) {
        tablero[i] = new char[nrow];
    }
    crear_juego(tablero, nrow, ncol,tab);
    imprimir_tableroC(tablero,nrow,ncol);
    int mov=0;
    while(termina==0){
        mover_ficha(tablero,nrow,ncol);
        ++mov;
        cout<<"Numero movimientos: "<<mov<<endl;
        imprimir_tableroC(tablero,nrow,ncol);
        termina=comprobar_juego(tablero,nrow,ncol);
    }
    borrar_tablero(tablero,ncol);
    return 0;
}
```

senku.cpp

```cpp
using namespace std;

char O='O',x='+',esp=' ';//Con esto podemos cambiar la forma de las fichas y
void crear_tablero1(tipomatriz tablero[][9],size_t filas,size_t columnas){
    for(size_t f=0;f<filas;++f){
        for(size_t c=0;c<columnas;c++){
            if(f==0 || f==8 || c==0 ||c==8 ){
                tablero[f][c]=esp;
            }
            else if((f<3 || f>5) && (c<3 ||c>5)){
                tablero[f][c]=esp;

            }else if(f==4 && c==4){
                tablero[f][c]=x;
            }else{
                tablero[f][c]=0;
            }

        }
    }
}
void crear_tablero2(tipomatriz tablero[][11],size_t filas,size_t columnas){
    for(size_t f=0;f<filas;++f){
        for(size_t c=0;c<columnas;c++){
            if(f==0 || f==10 || c==0 ||c==10 ){
                tablero[f][c]=esp;
            }
            else if((f<4 || f>6) && (c<4 ||c>6)){
                tablero[f][c]=esp;

            }else if(f==5 && c==5){
                tablero[f][c]=x;
```

senku.h

```cpp
#ifndef SENKU_PROYECTO_POO_SENKU_H
#define SENKU_PROYECTO_POO_SENKU_H
#include ...
typedef char tipomatriz;
int menu_inicial(tipomatriz tablero1[][9],tipomatriz tablero2[][11],tipomatriz
void crear_tablero1(tipomatriz tablero[][9],size_t filas,size_t columnas);
void crear_tablero2(tipomatriz tablero[][11],size_t filas,size_t columnas);
void crear_tablero3(tipomatriz tablero[][9],size_t filas,size_t columnas);
void imprimir_tablero1(tipomatriz tablero[][9],size_t filas,size_t columnas);
void imprimir_tablero2(tipomatriz tablero[][11],size_t filas,size_t columnas);
void imprimir_tablero3(tipomatriz tablero[][9],size_t filas,size_t columnas);
int comprobar_juego(char **tablero,size_t filas,size_t columnas);
void mover_ficha(char **tablero,int fil, int col);
void crear_juego(char **tablero, int nrow, int ncol,int tab);
void imprimir_tableroC(char **tablero,int nrow,int ncol);
void borrar_tablero(char **tablero,int ncol);
#endif //SENKU_PROYECTO_POO_SENKU_H
```

Funciones

Solo se usaron 12 funciones para hacer Senku, de las cuales 6 se usaron para crear e imprimir los 3 tipos de juegos.

```
typedef char tipomatriz;
int menu_inicial(tipomatriz tablero1[][9],tipomatriz tablero2[][11],tipomatriz tablero
void crear_tablero1(tipomatriz tablero[][9],size_t filas,size_t columnas);
void crear_tablero2(tipomatriz tablero[][11],size_t filas,size_t columnas);
void crear_tablero3(tipomatriz tablero[][9],size_t filas,size_t columnas);
void imprimir_tablero1(tipomatriz tablero[][9],size_t filas,size_t columnas);
void imprimir_tablero2(tipomatriz tablero[][11],size_t filas,size_t columnas);
void imprimir_tablero3(tipomatriz tablero[][9],size_t filas,size_t columnas);
int comprobar_juego(char **tablero,size_t filas,size_t columnas);
void mover_ficha(char **tablero,int fil, int col);
void crear_juego(char **tablero, int nrow, int ncol,int tab);
void imprimir_tableroC(char **tablero,int nrow,int ncol);
void borrar_tablero(char **tablero,int ncol);
```

Jugar

Es muy fácil de entender como jugar Senku.

Se proporcionó indicaciones directas de lo que el usuario tiene que escribir para que funcione. También se incluyó los números de columnas y filas para que sea más fácil de leer la posición de las fichas.

```
  1 2 3 4 5 6 7
1     0 0 0
2     0 0 0
3 0 0 0 0 0 0 0
4 0 0 0 + 0 0 0
5 0 0 0 0 0 0 0
6     0 0 0
7     0 0 0
```

```
Ingrese la posicion de origen (fila):
4
Ingrese la posicion de origen (columna):
2
Ingrese la posicion de destino (fila):
4
Ingrese la posicion de destino (columna):
4
```

Movimientos erróneos

Si el usuario se equivoca en escribir los puntos de origen y/o destino o hace un movimiento que no es válido.

El programa va a responder con “movimiento inválido” y le va a solicitar nuevos movimientos.

```
  1 2 3 4 5 6 7
1       0 0 0
2       0 0 0
3 0 0 0 0 0 0 0
4 0 0 0 + 0 0 0
5 0 0 0 0 0 0 0
6       0 0 0
7       0 0 0
```

```
Ingrese la posicion de origen (fila):
4
Ingrese la posicion de origen (columna):
3
Ingrese la posicion de destino (fila):
2
Ingrese la posicion de destino (columna):
3
Movimiento invalido.
Ingrese un movimiento valido.
Ingrese la posicion de origen (fila):
```

Ganar o Perder

Cuando el usuario gana, el programa te dice con un anuncio que ganaste.

```
Numero movimientos: 31
  1 2 3 4 5 6 7
1     + + +
2     + + +
3 + + + + + + +
4 + + + 0 + + +
5 + + + + + + +
6     + + +
7     + + +

¡Felicitaciones ganaste!
```

Cuando el usuario pierde, el programa te dice con un anuncio que lo intentes de nuevo.

```
Numero movimientos: 27
  1 2 3 4 5 6 7
1     + + +
2     + + +
3 + + + 0 + + +
4 + + + + + + +
5 0 + 0 + 0 + 0
6     + + +
7     + + +

Has perdido, intentalo de nuevo
```

Característica

Una de las características del código es ágil porque usa matrices dinámicas.

```
char**tablero= nullptr;

tablero=new char*[nrow];
```