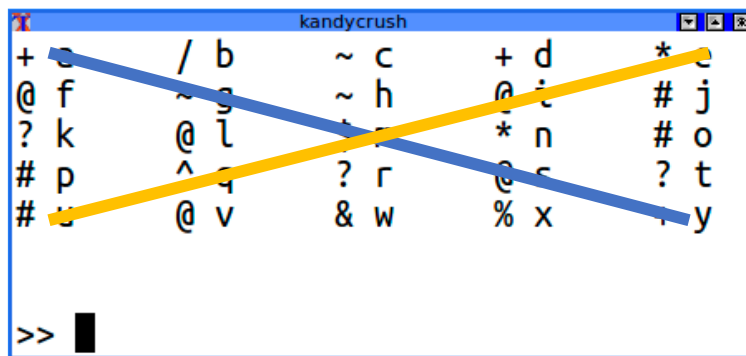


IDEA DE COMO DESARROLLAR EL CANDY CRUSH.

INTEGRANTES: William Aguirre Zapata y Juan Esteban Cardona.

La creación de la matriz 5x5 utilizando caracteres del código ascii y al lado los literales de las letras de la A hasta la Y del alfabeto, los cuales servirán para intercambiar la posición de las fichas (caracteres). El puntaje se mostrará en pantalla durante la partida, cada vez que se forme 3 caracteres en línea aumentará 10 puntos. La interacción de los movimientos de las posiciones se realizaran por entradas de teclado digitando sus respectivos literales: AB – BA.

Otras Jugadas ganadoras: Adicionaremos una jugada especial que consiste en formar la diagonal (0,0) (1,1) (2,2) (3,3) (4,4) o de las posiciones (0,4) (1,3) (2,2) (3,1) (4,0) donde jugador ganará 100 puntos.



POSIBLES ESTRUCTURAS A UTILIZAR.

```
struct ficha{
    char simbolo;
    char posicion;
};
typedef struct ficha tipoFicha;

struct tablero{
    int N=5, M=5;
    tipoFicha matrix[N][M];
};
typedef struct tablero tipoTablero;

struct juego{
    int score;
    int movimientos;
    char * nomJugador;
    tipoTablero box;
};
```

FUNCIONES NECESARIAS CON ENTRADAS Y SALIDAS.

```
void inicializarJuego (tipoJuego j, char candys[]);
/*Descripción: Esta función se encarga de inicializar el puntaje, los movimientos,
y llenar la matriz de caracteres con un Random.
Entradas: Le entra a la estructura tipoJuego, y un arreglo con los caracteres
del juego "@ # $ & %"
Salidas: El tablero dibujado junto con el puntaje y los límites de movimientos.*/

void inicializarTablero (tipoTablero t, int N, int M);
/*Descripción: Esta función se encarga de inicializar el tablero con sus respectivo
tamaño N x M.
Entradas: Le entra a la estructura tipoTablero, con dos enteros N para el tamaño
las filas, y M para el tamaño de las columnas.
Salidas: El tablero inicializado de N x M.*/

void dibujarFicha(tipoFicha f):
/*Descripción: Esta función imprime el símbolo en la posición respectiva, la cual
es llamada en inicializarJuego();
Entradas: Le entra una estructura tipoFicha.
Salidas: Imprime el símbolo en su posición [i][j].*/

void dibujarTablero(tipoTablero t);
/*Descripción: Esta función imprime todos los símbolos de las fichas, formando
el tablero.
Entradas: Le entra una estructura tipoTablero.
Salidas: Imprime el tablero dibujado dentro del juego. */

void leerMovimiento();
/*Descripción: Esta función se encarga de obtener por teclado los valores de los movimientos*/

void obtenerFicha(tipoTablero t, char cn);
/*Descripción: Esta función se encarga de obtener por teclado las posiciones de las
fichas.
Entradas: Le entra una estructura tipoTablero, y un carácter obtenido por leerMovimiento()
en la posición [0] o [1].
Salidas: Obtiene la posición de la ficha del tipoTablero. */

void intercambiarFichas(tipoFicha *f1, tipoFicha *f2);
/*Descripción: Esta función se encarga de intercambiar las posiciones del valor
de las fichas entradas por teclado.
Entradas: Le entra dos estructuras tipoFichas que son guardadas en variables c1 y c2
implementadas en obtenerFicha(t, cn).
Salidas: devuelve las posiciones nuevas con sus cambios respectivos en fichas. */
```

```

void mostrarValoresJuegos(tipoJuego j);
/*Descripción: Esta función se encarga de mostrar el estado del Juego en pantalla con sus
| | | | | respectivos valores.
Entradas: Le dentra una estructura tipoJuego.
Salidas: Imprime en panalla el estado del juego. Su Puntaje y los movimientos restantes.*/

void finalizarJuego (tipoJuego j);
/*Descripción: Esta función se encarga de verificar cuando un jugador termina
| | | | | el juego verificando el puntaje si los movimientos llegan a cero.
Entradas: Le dentra una estrutura tipoJuego.
Salidas: Imprime en pantalla si el Juego a terminado junto con su respectivo
| | | Puntaje. */

void resetearJuego (tipoJuego j);
/*Descripción: Esta función se encarga de reiniciar el juego para una nueva
| | | | | partida escribiendo "reset" durante el juego.
Entradas: Le dentra una estructura tipoJuego.
Salidas: Devuelve a su estado inicial la estructura tipoJuego con sus estructuras
| | | anidadas. */

void cerrarVentanas (tipoJuego j);
/*Descripción: Esta función se encarga de cerrar el juego y su ventana
| | | | | escribiendo "end" durante el juego.
Entradas: Le dentra una estructura tipoJuego. */

```