

Entrega I

Decidimos separar el juego en varios archivos a fin de aislar los problemas y mejorar la navegacion del codigo. Más adelante se describiran cada uno de los archivos.

Nuestra matriz es un vector largo **lamatriz [26][7][2]** en donde las posiciones(i) funcionan con **lamatriz[1...25]**, en **lamatriz[i][0][0]** se guarda el numero del dado en esa posicion (j) funcionan con **lamatriz[i][0...6]**, en la posicion **lamatriz[i][j][0]** se guarda a quien le pertenece el dado de valor (j), si es 1 es al humano y si es 2 es a la máquina.

TPCphalopoid.c

Es el archivo principal donde se inicia el juego, y donde está la **función** que verifica el ganador del juego. El usuario elige si ir primero o si la computadora va primero. Luego entramos a un **while** que repetira los turnos de jugador y computadora.

Jugador.c/.h

- void dadoUno(int);
- void dadoMas(int,int,int);
- void sumarDado(int,int);
- int validarJugada(void);
- int jugador(int);
- void ponerDadojugador(int);

Se le pide al jugador una posicion, luego en la función **validarJugada** con un **while** se verifica mediante **if** y si es válido el dato introducido se llama a la función limpiarBuffer para luego llamar a la función **ponerDadojugador** el cual recibe como parámetro el valor de la posicion introducido por el usuario.

Se agrupa posiciones con características parecidas y se suma los dados alrededor de la posicion seleccionada. Se llama a la función **sumarDado** que permite al jugador elegir cuántos dados, que dados, y si se arrepiente o no de elegir los dados y los come y suma siempre y cuando la suma sea menor a 6. Se utilizaron 4 while y 2 for para lograr esto(no se representa en la foto porque la función **dadoMas** es muy larga). El algebra utilizado se basa en la manipulacion de las posiciones de la matriz donde si la jugada es “arriba” se le resta 5 a la posicion,”abajo” se le suma 5, “izquierda” se le resta uno y “derecha se le suma uno.

Si es que no existe un movimiento el cual permite aumentar el valor del dado, el dado puesto en el tablero será 1.

```
int derecha = lamatriz[jugada + 1][0][0];
int abajo = lamatriz[jugada + 5][0][0];
int izquierda = lamatriz[jugada - 1][0][0];
int arriba = lamatriz[jugada - 5][0][0];
int vector[] = { izquierda, derecha, arriba, abajo, '\0' };
int suma = 0;
int resp = 0;
int comercantidad = 0;
int terminar = 0;
int ref[] = { -1, +1, -5, +5, '\0' };

void dadoUno(int jugada) {
    lamatriz[jugada][0][0] = 1;
    lamatriz[jugada][1][0] = 1;
}

void sumarDado(int suma,int jugada){
    lamatriz[jugada][0][0] = suma;
    lamatriz[jugada][suma][0] = 1;
}
```

```
if (jugada == 6 || jugada == 11 || jugada == 16) {
    suma = derecha + arriba + abajo;
    int j = 0;
    for (int i = 0; i < 4; i++) {
        if (vector[i] == 6) {
            suma += vector[i];
        }
        if (vector[i] != 0 && i != 0 && vector[i] != 6) {
            dif++;
            ubi[j] = ref[i];
            j++;
        }
    }
    if (suma < 7 && dif == 2) {
        sumarDado(suma, jugada);
        for (int i = 0; i < dif; i++) {
            lamatriz[jugada + ubi[i]][0][0] = '\0';
        }
    } else if (dif > 2) {
        int invalido = 1;
        dadoMas(invalido, jugada, dif);
    } else {
        dadoUno(jugada);
    }
}
```

Computadora.c/.h

Al entrar en computadora.c, lo primero que se ve es la función “ponerDado”, que recibe como parámetro la posición la cual la compu eligió usar, que veremos después cómo lo hace, y lo que hace es revisar si se puede sumar dados a su alrededor, y lo hace dividiendo en secciones:



- 1) las esquinas 1,5,21,25;
- 2) las filas de arriba: $2 < x < 5$; las filas de abajo $21 < x < 25$:

- 3) las columnas del costado izquierdo y derecho respectivamente: $x == 6 \parallel 11 \parallel 16$ y $x == 10 \parallel 15 \parallel 20$:
- 4) y la sección más grande que sería el bloque del medio: $6 < x < 10; 11 < x < 15; 16 < x < 20$

Cada uno con sus límites correspondientes (en la posición 1 no se le puede restar -1 como sí se puede en la posición 25). En la función, se entra en muchos if, que ven primero en que posición están, segundo ven si se puede sumar algo y si suman menor que 6.

Dentro del main, la “estrategia” que usa la compu es la siguiente: mira si puede poner arriba, abajo, a la izquierda o derecha del dado de la posición que eligió el jugador; y si no puede, tira uno random. Cada vez que lo hace llama la función “ponerDado”.

Funciones .c/.h

Aquí se encuentran funciones de uso diverso que hacen lo que su nombre indica cómo limpiar el buffer; aplicar un ente a la consola, imprimir la matriz entre otros