

Informe Técnico

Escuela de ingeniería eléctrica: Programación bajo plataformas abiertas

2022

Nombre del curso: Programación bajo plataformas abiertas

Sebastián Jara Solano C13862

Número de grupo: 15

Juego a desarrollar: Poker

Fecha: 8 de diciembre del 2022

ÍNDICE

Introducción 3
Diseño general 4
Principales Retos 8
Conclusión 9

Introducción

En el presente trabajo se presentará el proceso y retos de la realización del proyecto final del curso, dicho proyecto consiste en la elaboración de un programa capaz de comenzar un juego de poker y permitir al jugador saber si ganó o perdió contra la computadora, para esto fue necesario crear distintas instancias tales como una función que crease el mazo, funciones para evaluar al ganador, entre otras. También es importante mencionar que la realización de este proyecto fue posible gracias a las bases presentadas en las lecciones del curso.

Diseño General

I Creación De La Carta

Esta primera sección del proyecto se basó en la creación de la carta, la cual supone el primer reto del ejercicio, ya que un naipe en la vida real porta tanto un valor numérico (as, dos, tres, ..., J = 11, Q = 12, K = 13) y un símbolo (Diamante, Pica, Trébol, Corazón) el cual es mejor almacenar como un carácter. Para esto se hizo uso de lo aprendido en clase y aprendiendo de ejercicios anteriores, se definieron primero dos enumeraciones para almacenar los valores de los números y símbolos y luego se usaron estas para definir la estructura de carta, la cual almacena a ambos valores (emulando a un naipe real).

II Creación Del Mazo

Una vez definida una estructura capaz de almacenar dos valores distintos, se necesitaba crear todas las cartas posibles dados trece números y cuatro símbolos, dicha cantidad serían 52 cartas por lo que definiendo una estructura para almacenar las cartas totales y las cartas ya repartidas, el mazo estaría creado.

III Creación y Repartición De La Baraja

Haciendo uso de todas las estructuras antes definidas, se crearía una función para definir los valores específicos que un naipe tendría (por ejemplo generar un as de corazones), movería dicha combinación específica a la variable de cartas ya repartidas y seguiría asignando valores a las siguientes, mediante otras funciones se reparten una cantidad x ; fijada a $x = 5$ para emular un juego real, a cada jugador para así dar comienzo al juego.

IV Evaluador de Ganadores

Esta sección fue la más extensa de todo el proyecto y probablemente la que más problemas me generó, el poker tiene ciertas reglas bien definidas a la hora de definir un ganador, estas se basan en la repetición de valores de la carta en mano, por ejemplo, dos cartas con un mismo número son consideradas un par, la jugada más básica del poker, estas jugadas tienen una jerarquía, significando que una jugada “más alta” gana a una “más baja”, dicha jerarquía es la siguiente

- Nada : el jugador no consigue ninguna combinación en su mano
- Par : El jugador consigue dos cartas con el mismo valor en la mano
- Doble Par : El jugador consigue dos pares de cartas con el mismo tipo (ejemplo dos 3 y dos ases)
- Trío : El jugador consigue tres cartas del mismo valor en la mano
- Escalera : El jugador en su mano obtiene cartas cuyos valores son consecutivos (osea el primer valor es igual al segundo valor - 1 y así para los demás)
- Color : Todas las cartas del jugador poseen el mismo símbolo
- Full House : El jugador consigue un trío y una pareja
- Poker : El jugador consigue las cuatro cartas con el mismo valor (ejemplo de esto sería encontrar el as de diamantes, as de picas, as de flores y el as de corazones en una misma mano)
- Escalera de Color : Jugada que ocurre cuando el jugador consigue una escalera cuyas cartas poseen el mismo símbolo todas
- Escalera Real : Escalera de color con las cartas de mayor valor del juego

Definidas estas reglas, me propuse a crear una función que evaluaría una mano separando los valores numéricos de los símbolos de las cartas, y tratando como un arreglo la mano, la recorrería en busca de las jugadas. Para esto, se definieron variables con el nombre de las distintas jugadas cuyo valor inicial sería de 0, esto con el fin de luego a la hora de evaluar las distintas jugadas, en caso de cumplirse alguna de estas, entonces solo tendría que sumar un dígito a la variable correspondiente a dicha variable, tal y como se aprecia en la siguiente figura.

```

int Evaluador_de_mano (Carta **mano)
{
    int pareja=0, trio=0 , escalera=0, mismo_simbolo=0, poker=0, puntos_in_game=0;

    int i;
    int NumMano[13];
    int SimMano[4];

    memset(NumMano,0,sizeof(NumMano));
    memset(SimMano,0,sizeof(SimMano));

    for (i=0; i<5; i++)
    {
        NumMano[mano[i] ->numero]++;
        SimMano[mano[i] ->mazo]++;
    }
}

```

Figura 1. Evaluador de mano

Tras esto, para evitar el anidado de ifs, implementé distintas condiciones que evaluarían si se cumplían las jugadas que dependen de toda la mano, las cuales son la escalera y el Color, para ambas use el mismo procedimiento, el cual consistía en buscar en los arrays las similitudes deseadas, en el caso de escalera el programa busca en los valores numéricos de la mano si los valores de las cartas están en orden ascendente, en caso de que sí, se sumará uno al valor de la variable escalera, si no, el programa sigue. Por otro lado para evaluar color, el programa busca en la mano si el valor de símbolo de todas las cartas es igual, de ser el caso, se suma uno al valor de la variable “mismo_símbolo”, caso contrario, el programa sigue.

Para el caso de las jugadas que dependen de la repetición de valores numéricos, el programa busca en la mano cual es la cantidad de cartas que poseen el mismo valor, si solo encuentra dos, suma uno a la variable par (el programa suma 1 por cada pareja que encuentra, por lo que si encuentra dos parejas, el valor de la variable par será 2), de encontrar tres cartas cuyo valor numérico es igual, el programa suma uno a la variable trío, y en el caso de que se logre encontrar 4 cartas cuyo valor numérico es el mismo, el programa suma 1 a la variable poker.

Luego de todo esto, ya con todas las variables habiendo obtenido un valor, o haber permanecido en 0, el programa recorre todas las posibles posibilidades, esto lo hace de la siguiente manera:

```
if (escalera == 1 && mismo_simbolo == 1)
|   puntos_in_game = 8;
else if (poker == 1)
|   puntos_in_game = 7;
else if (trio == 1 && pareja == 1)
|   puntos_in_game = 6;
else if (escalera == 1)
|   puntos_in_game = 5;
else if (mismo_simbolo == 1)
|   puntos_in_game = 4;
else if (trio == 1)
|   puntos_in_game = 3;
else if (pareja == 2)
|   puntos_in_game = 2;
else if (pareja == 1)
|   puntos_in_game = 1;
else
|   puntos_in_game = 0;
```

Figura 2. Evaluación de jugada

Nótese que esta sección pudo haberse realizado también con la herramienta switch, sin embargo decidí realizarla con un anidado de ifs. Otro punto importante a mencionar es la variable “puntos_in_game”, esta variable es la que el evaluador de mano devuelve, tal y como se aprecia entre mayor sea la jugada que el jugador obtenga, mayor será los puntos que este obtendrá, el ganador será el jugador que obtenga la mayor cantidad de puntos, esto evaluado en la función “Determinar_ganador (int Puntos1, int Puntos2)”, la cual verifica cual de los dos jugadores obtuvo mayor puntaje

V Función menú

Por último la función menú permite al usuario elegir entre tres opciones, 1-jugar, 2-ver la tabla de puntuaciones y por último 3-salir del programa, si cualquier otro dígito es ingresado el programa imprime un mensaje de error, esta función opera gracias a un ciclo de while, el cual mantiene abierto el menú siempre y cuando la opción elegida sea distinta de 3.

Principales Retos

Este proyecto definitivamente fue todo un reto y me enseñó muchas lecciones importantes. De entre los retos más difíciles que encontré durante la escritura del código fueron, entre otros

- El manejo de punteros : Este complejo tema realmente estuvo presente durante todo el proceso ya que la gran mayoría de errores que encontré al compilar el programa se encontraban en un mal uso o posicionamiento de punteros
- Evaluar la mano : Separar los valores numéricos de carta y su símbolo también fue un reto ya que esto tenía que hacerse dentro de un arreglo, esta parte del proyecto saltó varios errores que me enseñaron a tener más cuidado con mi escritura
- La definición de las distintas jugadas : tener una mano de cartas en la vida real y decir que jugada de poker tengo es muy sencillo sin embargo cómo hacer que un programa fuese capaz de distinguirlas y decirme con certeza cual correspondía a cada caso fue un gran desafío que conllevó bastante tiempo.
- Llamada de funciones que requerían a estructuras como argumentos : esto presentó un desafío ya que me obligaba a hacer uso de punteros, lo cual como ya expuse antes, aún es un tema que me resulta complejo.
- Errores con el menú : ya que el menú existe gracias a un ciclo de while, al cambiar ciertos parámetros el ciclo se repetía una y otra vez sin dejar espacio a insertar una de las opciones, descubrir que causaba este error me llevó bastante tiempo, solucionarlo fue bastante fácil, pero al inicio, la incógnita de saber que estaba causando este error fue algo sin duda desafiante

Conclusión

Ya con anterioridad había cursado un curso de programación, en específico el curso de principios de informática, en donde aprendí un poco sobre el lenguaje python, sin embargo con este curso he realmente tenido una experiencia completamente distinta a lo ya experimentado antes, esto debido a que c, el lenguaje tratado en el curso, al tratarse de un lenguaje de más bajo nivel que python realmente permite ver distintos procesos que el compilador realiza, permitiendo un mayor control sobre el código, no he logrado terminar el proyecto en su totalidad, ciertas especificaciones me fueron imposibles de implementar al código y aún hay demasiadas funciones que son extrañas para mí y definitivamente este ha sido un proyecto que me ha dado bastantes problemas, sin embargo gracias a este proyecto y a este curso me he decidido a aprender a profundidad a como programar.