

### CANVI 1:

Trobo innecessària la variable auxiliar comptadora turno que s'iguala a 0.

```
// Variables auxiliarsS
int num_cartas_mano, carta_mano_maquina, puntos, contadorAs, puntosMaquina, turno = 0;
```

Es fa servir per començar el joc amb la repartició de les dues cartes de l'usuari però considero que si traiem aquesta variable juntament amb la seva condició, el programa funcionaria d'igual manera ja que sempre es comença repartint dos cartes.

```
if (turno == 0)
{
    cout << "\033[1;31m \nTus cartas son: \033[0m" << endl;
    puntos = 0;
    contadorAs = 0;
    for (int contadorMano = 0; contadorMano < 52; contadorMano++)
    {
        if (mano[contadorMano].palo != "")
        {
            // Si toca un AS suma 1 o 11 depenent dels punts que tingui l'usuari.
            if (mano[contadorMano].numero == 1)
            {
                cout << "\033[1;33m A de " << mano[contadorMano].palo << "\033[0m" << endl;
                contadorAs++;
                if (contadorAs >= 1 && puntos + 11 > 21)
                {
                    puntos = puntos + 1;
                }
            }
            else
            {

```

Borraré doncs la variable **turno = 0**, la condició **if (turno == 0)** i l'actualització de la variable **turno++**.

Tenim una variable menys.

### CANVI 2:

El següent canvi seria eliminar llibreries que no es fan servir.

```
#include <iostream>
#include <cstdlib> //for rand and srand
#include <cstdio>
#include <string>
#include <ctime> // time function for seed value
#include <chrono>
#include <thread>

using namespace std;
```

S'utilitzen iostream per l'ús de cout i cin amb using namespace std i el cstdlib i ctime per l'ús de les funcions random. La de chrono i thread s'utilitzen per no fer el joc instantani i que hi hagi entre moviment i moviment. Les que eliminem són les de **string**, perquè no fem cap ús a les funcions de cadena de caràcters i **cstdio** perquè ja utilitzem el iostream.

```
#include <iostream>
#include <cstdlib> //for rand and srand
#include <ctime> // time function for seed value
#include <chrono>
#include <thread>

using namespace std;
```

La cosa quedaria amb aquestes quatre llibreries.

### CANVI 3:

```
// Array amb els possibles números que poden sortir
int numeros[13] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13};
// Array amb els possibles pals de les cartes.
const string palos[4] = {"Corazones", "Diamantes", "Picas", "Treboles"};

int contador = 0;

// Tipus de carta (número i pal)
for (int i = 0; i < 13; i++)
{
    for (int j = 0; j < 4; j++)
    {
        baraja[contador].numero = numeros[i];
        baraja[contador].palo = palos[j];
        contador++;
    }
}
```

Podem eliminar l'**array de números** i en el moment de posar les cartes a la baralla podem posar el valor de **i + 1**

```
time_t t;
srand((unsigned)time(&t));
// Array amb els possibles pals de les cartes.
const string palos[4] = {"Corazones", "Diamantes", "Picas", "Treboles"};

int contador = 0;

// Tipus de carta (número i pal)
for (int i = 0; i < 13; i++)
{
    for (int j = 0; j < 4; j++)
    {
        baraja[contador].numero = i+1;
        baraja[contador].palo = palos[j];
        contador++;
    }
}
```

D'aquesta manera ens estalviem un array.

#### CANVI 4:

Recorre la mà de l'usuari 52 vegades és una tonteria. El jugador mai tindrà 52 cartes.

```
// Recorre la mà de l'usuari
for (int contadorMano = 0; contadorMano < 52; contadorMano++)
{
```

Ho canviem de la següent forma:

```
81
82 // Recorre la mà de l'usuari
83 for (int contadorMano = 0; contadorMano < num_cartas_mano + 1; contadorMano++)
84 {
```

Mirem fins a un màxim de cartes que té l'usuari inicialment (0) + 1. Així en la posició 0 de la mà, tindrem el valor de la primer carta, en la posició 1, el valor de la segona...

#### CANVI 5:

Aquí, per mostrar les cartes no cal recorre totes les 52. Hem de mostrar fins el número de cartes que té l'usuari.

```
// Mostrar cartes
case 2:
    system("clear");
    cout << "\033[1;31m \nTus cartas son: \033[0m" << endl;
    puntos = 0;
    contadorAs = 0;
    for (int contadorMano = 0; contadorMano < 52; contadorMano++)
    {
```

Canviem la condició de 52 al número de cartes que té l'usuari (**num\_cartas\_mano**).

```
// Mostrar cartes
case 2:
    system("clear");
    cout << "\033[1;31m \nTus cartas son: \033[0m" << endl;
    puntos = 0;
    contadorAs = 0;
    for (int contadorMano = 0; contadorMano < num_cartas_mano; contadorMano++)
    {
```

#### CANVI 6:

No sé si això canvia en el rendiment del programa però podríem escriure aquestes sentències:

```

if (contadorAs >= 1 && puntos
{
    puntos = puntos + 1;
}
else
{
    puntos = puntos + 11;
}

```

D'aquesta altra forma

```

if (contadorAs >= 1 && punto
{
    puntos += 1;
}
else
{
    puntos += 11;
}

```

### CANVI 7:

Una altra funció per veure el valor de la carta. Es fa servir per veure quina carta roba la màquina i per veure la baralla sencera.

```

if (manoMaquina[carta_mano_maquina].numero == 1)
{
    cout << "\033[1;33m A de " << manoMaquina[carta_mano_maquina].palo << "\033[0m" << endl;
}
else if (manoMaquina[carta_mano_maquina].numero == 11)
{
    cout << "\033[1;33m Sota de " << manoMaquina[carta_mano_maquina].palo << "\033[0m" << endl;
}
else if (manoMaquina[carta_mano_maquina].numero == 12)
{
    cout << "\033[1;33m Caballo de " << manoMaquina[carta_mano_maquina].palo << "\033[0m" << endl;
}
else if (manoMaquina[carta_mano_maquina].numero == 13)
{
    cout << "\033[1;33m Rey de " << manoMaquina[carta_mano_maquina].palo << "\033[0m" << endl;
}
else
{
}

```

Ho passem a una funció.

```
// Funcions
void mostrar_resultats(carta mano[], int contador)
{
    if (mano[contador].numero == 1)
    {
        cout << "\033[1;33m A de " << mano[contador].palo << "\033[0m" << endl;
    }
    else if (mano[contador].numero == 11)
    {
        cout << "\033[1;33m Sota de " << mano[contador].palo << "\033[0m" << endl;
    }
    else if (mano[contador].numero == 12)
    {
        cout << "\033[1;33m Caballo de " << mano[contador].palo << "\033[0m" << endl;
    }
    else if (mano[contador].numero == 13)
    {
        cout << "\033[1;33m Rey de " << mano[contador].palo << "\033[0m" << endl;
    }
    else
    {
        cout << "\033[1;33m " << mano[contador].numero << " de " << mano[contador].palo << "\033[0m" << endl;
    }
}
}
```

Ho passem a la main

```
// Plantar-se
case 3:
    system("clear");
    cout << "\033[1;32mHas conseguido " << puntos << " puntos \033[0m" << endl;
    cout << "\033[1;36mAhora juega la máquina\033[0m" << endl;
    while (puntosMaquina < puntos)
    {
        cout << "\033[1;36mLa maquina roba carta\033[0m" << endl;
        std::this_thread::sleep_for(std::chrono::milliseconds(1000));
        manoMaquina[carta_mano_maquina] = baraja[rand() % 52];
        mostrar_resultats(manoMaquina, carta_mano_maquina);
        puntosMaquina = puntosMaquina + manoMaquina[carta_mano_maquina].numero;
        cout << "\033[1;37m Puntos maquina: " << puntosMaquina << endl;
        carta_mano_maquina++;
    }

case 4:
    for (int contadorBaraja = 0; contadorBaraja < 52; contadorBaraja++)
    {
        mostrar_resultats(baraja, contadorBaraja);
    }
}
```

## CANVI 8:

Una funció per mostrar el resultat de la partida.

```

void resultat_punts(int puntos, bool repetir)
{
    // Més de 21 punts perds la partida
    if (puntos > 21)
    {
        cout << "\033[1;31mTe has pasado... puntos: " << puntos << "\033[0m" << endl;
        cout << "\033[1;31mHas perdido\033[0m" << endl;
        repetir = false;
    }

    // 21 punts guanyes o empates
    else if (puntos == 21)
    {
        cout << "\033[1;33mFELICIDADES tienes " << puntos << " puntos!(Blackjack)\033[0m" << endl;
        repetir = false;
    }

    // Menys de 21 punts mostres el punts que tens i continues jugant
    else
    {
        cout << "\033[1;32mTienes " << puntos << " puntos \033[0m" << endl;
    }
}

```

Cridem a la funció main

```

// Robar carta
case 1:
    system("clear");
    // Afegeixes una carta aleatòria a la mà
    mano[num_cartas_mano + 1] = baraja[rand() % 52];
    cout << "\033[1;32mTu carta es: \033[0m" << endl;

    // Es torna a comprovar la suma de punts depenent de la carta que et toca
    puntos = puntuacio(mano, num_cartas_mano, contadorAs, puntos, num_cartas_mano);

    // Tornem a mostrar els resultats
    resultat_punts(puntos, repetir);
    if (puntos >= 21)
    {
        break;
    }
}

```

Si els punts són 21 o més, s'acaba el joc per nosaltres.

### CANVI 9:

Fem un funció per calcular els punts que porta l'usuari.

```

int puntuacio(carta mano[], int contadorMano, int contadorAs, int puntos, int num_cartas_mano)
{
    if (mano[contadorMano].numero == 1)
    {
        cout << "\033[1;33m A de " << mano[contadorMano].palo << "\033[0m" << endl;
        contadorAs++;
        if (contadorAs >= 1 && puntos + 11 > 21)
        {
            puntos += 1;
        }
        else
        {
            puntos += 11;
        }
        num_cartas_mano++;
    }
    else if (mano[contadorMano].numero == 11)
    {
        cout << "\033[1;33m Sota de " << mano[contadorMano].palo << "\033[0m" << endl;
        puntos += 10;
        num_cartas_mano++;
    }
    else if (mano[contadorMano].numero == 12)
    {
        cout << "\033[1;33m Caballo de " << mano[contadorMano].palo << "\033[0m" << endl;
        puntos += 10;
        num_cartas_mano++;
    }
    else if (mano[contadorMano].numero == 13)
    {
        cout << "\033[1;33m Rev de " << mano[contadorMano].palo << "\033[0m" << endl;
    }
    else if (mano[contadorMano].numero == 13)
    {
        cout << "\033[1;33m Rey de " << mano[contadorMano].palo << "\033[0m" << endl;
        puntos += 10;
        num_cartas_mano++;
    }
    else
    {
        cout << "\033[1;33m " << mano[contadorMano].numero << " de " << mano[contadorMano].palo << "\033[0m" << endl;
        puntos += mano[contadorMano].numero;
        num_cartas_mano++;
    }
    return puntos;
}

```

Fem la funció perquè es repeteix el càlcul dels punts que té el jugador.

```

// Mostrar cartes
case 2:
    system("clear");
    cout << "\033[1;31m \nTus cartas son: \033[0m" << endl;
    puntos = 0;
    contadorAs = 0;
    for (int contadorMano = 0; contadorMano < num_cartas_mano; contadorMano++)
    {
        if (mano[contadorMano].palo != "")
        {
            puntos = puntuacio(mano, contadorMano, contadorAs, puntos, num_cartas_mano);
        }
    }
    cout << "\033[1;32mTienes " << puntos << " puntos \033[0m" << endl;
    break;

```

Així queda una part del codi main, per exemple.

```

81
82     // Recorre la mà de l'usuari
83     for (int contadorMano = 0; contadorMano < num_cartas_mano + 1; contadorMano++)
84     {
85         // Si la carta de l'usuari és un pal dels quatre pals
86         if (mano[contadorMano].palo != "")
87         {
88             puntos = puntuacio(mano, contadorMano, contadorAs, puntos, num_cartas_mano);
89             num_cartas_mano++;
90         }
91     }

```

Aquí una altra.

### CANVI 10:

La mà dels jugador mai tindrà 52 cartes.

```

// Variables per la mà de l'usuari i la màquina
struct carta mano[52], manoMaquina[52];
// Variables auxiliars

```

Lo seu seria fer que la mà tingués màxim 11 posicions. 11 perquè són les cartes màximes amb les que pot fer 21. 4 As amb valor de 1 = 4, 4 dosos i 3 tresos. Amb aquestes tres cartes de valor mínim fas 21 punts.

```

// Variables per la mà de l'usuari i la màquina
struct carta mano[11], manoMaquina[11];

```

Inicialitzem també els punts i les cartes de la màquina a 0 per evitar problemes i pugui entrar al bucle per calcular els punts.

```

case 3:
    system("clear");
    cout << "\033[1;32mHas conseguido " << puntos << " puntos \033[0m" << endl;
    cout << "\033[1;36mAhora juega la máquina\033[0m" << endl;
    carta_mano_maquina = 0;
    puntosMaquina = 0;
    while (puntosMaquina < puntos)
    {
        cout << "\033[1;36mLa máquina roba carta\033[0m" << endl;
        std::this_thread::sleep_for(std::chrono::milliseconds(1000));
        manoMaquina[carta_mano_maquina] = baraja[rand() % 52];
        mostrar_resultats(manoMaquina, carta_mano_maquina);
        puntosMaquina = puntosMaquina + manoMaquina[carta_mano_maquina].numero;
        cout << "\033[1;37m Puntos maquina: " << puntosMaquina << endl;
        carta_mano_maquina++;
    }

```