Blackjack roz 💗🎰

Dumea Mihaela

AIA- anul 1

Grupa a 2 a

Cuprins:

1.Detalii despre aplicație și motivul ales.

2. Regulile jocului .

3.Scopul proiectului.

4.Tehnologii și structuri utilizate .

5 .Analizarea codului .

6.Elemente vizuale .

7 . Bibliografie .

1.Ce reprezintă blackjack-ul ❓🎰💗

Blackjack (cunoscut și sub numele de „21 puncte”) este unul dintre cele mai populare jocuri de cărți din lume, jucat atât în cazinouri, cât și în formă recreativă , eu am adus o îmbunătățire a aspectului acestui joc , cu culori și emojiuri , făcându-l mai estetic .Scopul jocului este simplu: jucătorul trebuie să obțină un total cât mai apropiat de 21 de puncte, fără să îl depășească.

Am ales să fac acest joc pentru proiectul meu de final de semestru deoarece , consider ca ,jocurile de intuire și alegere pot fii esențiale pentru dezvoltarea inteligenței logice , dar și pentru prevenirea bolii Alzheimer , deoarece aceste jocuri de logică , activează regiuni din creier (cortexul frontal , hipocampul , etc.) activând astfel sinapsele active și pot forma rețele neuronale de rezervă .

Fiecare jucător joacă împotriva dealerului, nu împotriva celorlalți jucători. Cărțile numerotate (de la 2 până la 10) valorează cifra pe care o reprezintă, iar  aceste cărți : valet (J), damă (Q) și rege (K) , valorează fiecare câte 10 puncte. Asul (A) poate valora fie 1, fie 11 puncte , în funcție de avantajul jucătorului.

Acest joc folosește un mecanism de bază , ce constă în alegerea dintre două acțiuni :

1. “Hit” - ce înseamnă mai tragi o carte

2. ⁠”Stand ”- oprește-te și păstrează scorul actual .

Dacă totalul punctelor jucătorului depășește 21, acesta pierde automat runda. După ce jucătorul își finalizează tura, dealerul trage automat cărți până ajunge la un scor de minimum 17 puncte .

Blackjack este apreciat pentru simplitatea regulilor, dar și pentru strategia implicată în luarea deciziilor ,când să mai tragi o carte și când să te oprești. Este un joc care îmbină norocul cu analiza și intuiția, motiv pentru care a devenit un clasic al jocurilor de noroc din întreaga lume.

2.Regulile jocului:💗🎰

1. Scopul jocului

Obiectivul este să obții un scor cât mai apropiat de 21, fără să îl depășești. Jucătorul joacă împotriva dealerului (calculatorul), nu împotriva altor jucători.

2. Valoarea cărților

Cărțile de la 2 la 10 au valoarea pe care o poartă.

Valetul , Dama și Regele valorează fiecare câte 10 puncte.

Asul valorează 1 punct sau 11 în funcție de cum îl ajută pe utilizator .

3. Desfășurarea jocului

Jucătorul trage cărți pe rând și poate decide să continue sau să se oprească în funcție de opțiunea aleasă .

Dacă scorul jucătorului depășește 21, acesta pierde automat runda.

După ce jucătorul se oprește, dealerul trage automat cărți până ajunge la un scor de cel puțin 17.

4. Determinarea câștigătorului

Dacă jucătorul are peste 21, pierde runda.

Dacă dealerul are peste 21, jucătorul câștigă.

Dacă niciunul nu a depășit 21 există 3 posibilități :

Jucătorul câștigă dacă are un scor mai mare decât dealerul.

Dealerul câștigă dacă are un scor mai mare.

Egalitate dacă scorurile sunt egale.

5. Sistemul de scor:

La fiecare rundă, se actualizează scorul total al sesiunii:

Victorie pentru jucător

Înfrângere

Egalitate

3.Scopul proiectului:

Scopul principal al acestui proiect este realizarea unei aplicații interactive în linie de comandă, care simulează jocul de cărți Blackjack, prin limbajul de programare C. Proiectul este conceput ca o modalitate practică de a exersa și demonstra cunoștințele fundamentale dobândite în cadrul laboratorului de programare, în special în ceea ce privește utilizarea funcțiilor, a condițiilor, a buclelor și a variabilelor globale.

Prin intermediul acestui proiect, se urmărește dezvoltarea următoarelor competențe:

Structurarea unui program în C : Codul este împărțit în funcții clare și bine definite, fiecare responsabilă pentru o parte specifică a jocului (tragerea cărților, afișarea scorului, rularea turei jucătorului, etc.). Această abordare sprijină înțelegerea principiilor de organizare logică a codului și crește lizibilitatea programului.

Utilizarea funcțiilor din bibliotecile standard: Proiectul folosește funcții din bibliotecile stdio.h, stdlib.h și time.h pentru a gestiona input/output, generarea de numere aleatoare și manipularea timpului.

Aplicarea structurilor de control: acest joc implică luarea de decizii pe baza scorurilor curente și a alegerilor utilizatorului, astfel încât se folosesc condiții if, else și switch, dar și bucle while și do...while, pentru a permite repetarea acțiunilor până la o anumită condiție.

Interacțiunea cu utilizatorul: Printr-un meniu simplu și intuitiv afișat în terminal, utilizatorul poate alege să joace o rundă, să vizualizeze scorul, să reseteze scorurile sau să iasă din joc. Această interacțiune este realizată prin comenzi printf și scanf.

Folosirea culorilor în terminal: Pentru a face aplicația mai atractivă vizual, sunt utilizate coduri ANSI pentru a colora textul în roz. Acest aspect adaugă o notă personalizată și creativă proiectului, și evidențiază faptul că programarea poate fi și expresivă, nu doar funcțională.

Gândire logică și decizională: În jocul Blackjack, utilizatorul trebuie să analizeze scorul și să decidă dacă să mai tragă o carte sau să se oprească. Acest mecanism adaugă un nivel de interactivitate și simulează luarea de decizii într-un context logic.

În concluzie, proiectul urmărește să îmbine partea tehnică (programarea în C) cu elemente de joc și estetică, într-un mod simplu, dar eficient. Este o oportunitate de a pune în practică noțiunile teoretice învățate, într-un cadru creativ și distractiv, care încurajează învățarea activă și aprofundarea logicii programării.

4.Tehnologii și structuri utilizate:🎰🃏♥️

Pentru realizarea acestui joc de Blackjack, am folosit câteva lucruri de bază din limbajul C și câteva funcții din bibliotecile standard. Iată ce am folosit mai exact:

Limbaj și biblioteci

->Limbajul C , tot jocul e scris în C, pentru că e un limbaj foarte bun ca să înveți cum funcționează lucrurile „de la bază”.

-> stdio.h , pentru afișări pe ecran și citirea opțiunilor de la utilizator (printf, scanf).

-> stdlib.h , pentru funcția rand(), care ne ajută să tragem cărți aleatoare.

-> time.h , ca să folosim srand(time(NULL)), care face ca fiecare joc să aibă cărți diferite de fiecare dată (altfel ar fi mereu aceeași ordine).

Structurile de control folosite :

-> Bucle while și do...while , ca să putem repeta runde de joc, meniuri sau să tragem cărți până la un anumit scor.

-> Instrucțiuni if, else if, else : pentru a lua decizii, cum ar fi cine a câștigat sau dacă jucătorul a depășit 21 de puncte.

-> switch , folosit în meniul principal, ca să alegi ușor între opțiuni.

Funcții proprii

-> Am scris funcții separate pentru fiecare parte a jocului: tragerea unei cărți, afișarea scorului, turul jucătorului, turul dealerului, verificarea câștigătorului, etc. Asta ajută la organizare și face codul mai ușor de înțeles și modificat.

Pentru a personaliza acest joc am adăugat culori în terminal:

-> Am folosit coduri ANSI ca să colorez textul în roz. E un detaliu simplu, dar face jocul mai plăcut vizual și îi dă un stil aparte.

Variabile folosite :

-> Am folosit 3 variabile globale pentru scor: victorii, infrangeri și egaluri, care se păstrează pe tot parcursul sesiunii de joc. Așa putem ține minte rezultatele din mai multe runde.

5.Analizarea codului:🎰🕹️

1.Bibliotecii incluse și definiții :

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

-> stdio.h: a fost folosit pentru funcțiile de input/output (printf, scanf).

-> stdlib.h: pentru rand() și srand() (generarea numerelor aleatoare).

-> time.h: acesta l-am inclus pentru a putea folosi time(NULL) pentru inițializarea randomului (altfel am primi aceleași valori de fiecare dată).

#define ROZ "\033[1;35m"

#define NORMAL "\033[0m"

Aici definim două constante:

-> Culoarea ROZ : un cod ANSI care colorează textul în roz (mov intens).

-> NORMAL: codul care resetează culoarea la normal.

Astfel ajută la stilizarea textului în terminal.

int victorii = 0, infrangeri = 0, egaluri = 0;

-> Conține 3 variabile globale care țin scorul sesiunii curente:

-> Câte runde ai câștigat, pierdut sau încheiat la egalitate.

-> Sunt accesibile de oriunde în program.

2.Functia trage\_carte() :

int trage\_carte() {

int c = rand() % 13 + 1;

if (c > 10) return 10;

return c;

}

-> Aceasta funcție generează o valoare între 1 și 13.

-> Dacă e mai mare de 10 (adică J, Q, K), returnează 10 – exact ca în regulile clasice de Blackjack.

-> Această funcție simulează tragerea unei cărți.

3.Functia afiseaza\_carte(int c):

void afiseaza\_carte(int c) {

if (c == 1)

printf(ROZ "A " NORMAL);

else

printf(ROZ "%d " NORMAL, c);

}

Primește o valoare de carte și o afișează în roz.

-> Dacă e 1, o tratează ca As (A).

-> Nu gestionează valori grafice (ex: ♠️, ♥️), dar e clar și eficient pentru terminal.

4. Funcția tura\_jucator():

int tura\_jucator() {

int total = 0, carte, optiune;

do {

carte = trage\_carte();

printf("Ai tras: ");

afiseaza\_carte(carte);

total += carte;

printf("\nScorul tău e acum: " ROZ "%d\n" NORMAL, total);

if (total > 21) {

printf(ROZ "Ai depasit 21, ai pierdut runda 😢\n" NORMAL);

return total;

}

printf("1. Mai trage o carte\n2. Stai\nAlege: ");

scanf("%d", &optiune);

} while (optiune == 1);

return total;

}

Inițializează scorul jucătorului (total).

-> Într-o buclă do...while, trage o carte, o afișează și actualizează scorul.

-> Dacă scorul depășește 21, jucătorul pierde automat.

-> Jucătorul poate alege să continue sau să se oprească.

5. Funcția tura\_dealer():

int tura\_dealer() {

int total = 0, carte;

printf("\nDealerul joaca...\n");

while (total < 17) {

carte = trage\_carte();

printf("Dealerul a tras: ");

afiseaza\_carte(carte);

total += carte;

printf("(total: %d)\n", total);

}

return total;

}

Simulează comportamentul dealerului.

-> Dealerul continuă să tragă până ajunge la un scor de minim 17, conform regulilor clasice ale acestui joc .

-> Nu are opțiuni , acțiunea e complet automată.

6. Funcția verifica\_castigator():

void verifica\_castigator(int scor\_jucator, int scor\_dealer) {

printf("\nScor final: Tu = %d | Dealer = %d\n", scor\_jucator, scor\_dealer);

if (scor\_jucator > 21) {

printf(ROZ "Ai pierdut (ai depasit 21)\n" NORMAL);

infrangeri++;

} else if (scor\_dealer > 21 || scor\_jucator > scor\_dealer) {

printf(ROZ "Ai castigat! 😎\n" NORMAL);

victorii++;

} else if (scor\_jucator == scor\_dealer) {

printf(ROZ "Remiza 🤝\n" NORMAL);

egaluri++;

} else {

printf(ROZ "Dealerul a castigat 😬\n" NORMAL);

infrangeri++;

}

}

Primește scorurile și decide câștigătorul.

Reguli clare:

-> Dacă jucătorul sare de 21, pierde.

-> Dacă dealerul sare de 21 sau scorul jucătorului e mai mare, câștigă jucătorul.

-> Egalitate = remiză.

->Actualizează scorurile în funcție de rezultat.

7. Funcția afiseaza\_scor():

void afiseaza\_scor() {

printf(ROZ "\nScor total sesiune:\n" NORMAL);

printf("Victorii: %d\n", victorii);

printf("Infrangeri: %d\n", infrangeri);

printf("Egaluri: %d\n\n", egaluri);

}

-> Afișează scorul total acumulat în sesiunea actuală.

-> Datele provin din variabilele globale.

8. Funcția reset\_scor():

void reset\_scor() {

victorii = infrangeri = egaluri = 0;

printf(ROZ "Scor resetat 🔄\n" NORMAL);

}

-> Resetează toate scorurile la zero.

-> Se poate folosi când utilizatorul vrea să reînceapă.

9. Funcția joaca():

void joaca() {

int jucator = tura\_jucator();

if (jucator <= 21) {

int dealer = tura\_dealer();

verifica\_castigator(jucator, dealer);

} else {

infrangeri++;

}

}

-> Coordonează o rundă completă.

-> Apelează tura\_jucator() și, dacă jucătorul nu a pierdut direct, se apelează și tura\_dealer(), urmat de verifica\_castigator().

10. Funcția meniu():

void meniu() {

int opt;

do {

printf(ROZ "\n--- Meniu Blackjack Roz ---\n" NORMAL);

printf("1. Joaca o runda\n");

printf("2. Arata scor\n");

printf("3. Reseteaza scor\n");

printf("4. Iesire\n");

printf("Optiune: ");

scanf("%d", &opt);

switch (opt) {

case 1: joaca(); break;

case 2: afiseaza\_scor(); break;

case 3: reset\_scor(); break;

case 4: printf(ROZ "Pa pa! 👋\n" NORMAL); break;

default: printf("Alegere invalida.\n");

}

} while (opt != 4);

}

Afișează un meniu repetitiv până utilizatorul alege să iasă.

-> Oferă acces la toate funcțiile jocului.

-> Este „nucleul” interacțiunii cu utilizatorul.

11. Funcția principală main():

int main() {

srand(time(NULL));

printf(ROZ "\n\*\*\* Bine ai venit la Blackjack Roz! \*\*\*\n\n" NORMAL);

meniu();

return 0;

}

-> Inițializează randomul cu srand() folosind timpul actual.

-> Afișează un mesaj de bun venit.

-> Apelează meniul principal.

6.Elementele vizuale ale aplicației:

Unul dintre aspectele care face ca acest proiect să iasă în evidență față de un simplu joc de consolă este accentul pus pe elementele vizuale, adică modul în care informațiile sunt afișate pe ecran. Deși aplicația nu are o interfață grafică cu butoane sau ferestre, ea reușește să creeze o experiență vizuală plăcută direct în terminal , folosind culori, simboluri și structură de afișare.

1.Culoarea roz: Pentru a oferi un aspect unic, și tematic, interfața jocului folosește culoarea roz ca temă principală.

Această culoare este aplicată la:

->Titlurile afișate (ex: „Meniu Blackjack Roz”)

-> Scorul afișat

->Cărțile trase

-> Mesajele importante de stare (victorie, înfrângere, remiză)

Această culoare este adăugată cu ajutorul codurilor ANSI, specifice terminalului, mai exact:

#define ROZ "\033[1;35m"

#define NORMAL "\033[0m"

->ROZ setează culoarea textului la roz.

->NORMAL revine la culoarea implicită (alb sau gri, în funcție de sistem).

Astfel, orice text afișat între aceste două coduri va apărea colorat, ceea ce ajută la citirea rapidă a informațiilor importante.

2. Emoticoanele :

Pentru a face jocul mai interactiv în loc de texte simple am decis să includ și emoticoane precum :

-> 😎 la victorie

-> 😬 când dealerul câștigă

-> 🤝 la remiză

-> 😢 când jucătorul pierde

-> 👋 la ieșire

Aceste elemente de efect adaugă caracter și fac ca experiența să fie mai distractivă și mai puțin formală.

3 Afișarea cărților :

Afișarea cărților cu denumiri (A, J, Q,K) :

-> 1 → „A”

-> 11 → „J”

-> 12 → „Q”

-> 13 → „K”

Acest lucru este realizat cu o funcție specială:

const char\* denumire\_carte(int c) {

...

}

Prin această funcție , jucătorul înțelege mai ușor ce carte a tras.

4.Structura clară a meniului :

--- Meniu Blackjack Roz ---

1. Joaca o runda

2. Arata scor

3. Reseteaza scor

4. Iesire

Optiune:

Această organizare simplă, dar eficientă, oferă o navigare ușoară chiar și pentru cei care nu sunt obișnuiți cu jocurile în linia de comandă.

7.Concluzie : ♣️♥️♦️♠️

Proiectul „Blackjack Roz” reprezintă mai mult decât o simplă implementare a unui joc clasic în limbajul C — este o dovadă clară că și aplicațiile în consolă pot fi atractive, interactive și prietenoase pentru utilizator, atunci când sunt construite cu atenție la detalii, logică bine structurată și un strop de creativitate.

Acest proiect a fost gândit ca un exercițiu complet de programare, în care am aplicat noțiuni învățate la cursuri și laboratoare: de la utilizarea funcțiilor, variabilelor globale, buclelor și condiționalelor, până la lucrul cu generatoare de numere aleatoare (rand()), coduri de culoare ANSI pentru stilizarea terminalului și organizarea codului în mod modular.

Pe lângă partea tehnică, s-a pus accent pe interacțiunea utilizatorului și pe o experiență cât mai prietenoasă: folosirea culorii roz ca temă vizuală, adăugarea de emoji-uri sugestive, afişarea cărţilor într-un mod cât mai apropiat de realitate (A, J, Q, K), precum şi mesaje clar formulate pentru fiecare situaţie din joc. Astfel, jucătorul se poate bucura de un joc care pare simplu la prima vedere, dar care oferă o experienţă completă şi captivantă.

Pe tot parcursul dezvoltării, s-au urmărit principii de organizare logică și claritate: codul este împărțit în funcții cu roluri bine definite, fiecare acțiune este ușor de urmărit, iar structura generală a aplicației permite extinderea ușoară pe viitor. De exemplu, pot fi adăugate opțiuni suplimentare precum salvarea scorului într-un fișier, joc în doi jucători sau chiar un mod cu dificultate variabilă.

Un alt aspect important a fost adaptarea jocului la contextul unui terminal deși nu este o aplicație cu interfață grafică (GUI), jocul reușește să creeze o atmosferă unică și plăcută prin simplitate și stil.

Realizarea acestui proiect a contribuit semnificativ la înțelegerea modului în care putem transpune o idee concretă (un joc de noroc cunoscut) în cod funcțional, pas cu pas, într-un mod controlat și logic. Totodată, a evidențiat importanța testării, a gândirii din perspectiva utilizatorului și a dorinței de a adăuga un plus de personalitate propriului proiect.

În concluzie, „Blackjack Roz” este un proiect care îmbină eficient logica programării cu creativitatea .. Este un exemplu de proiect în care programarea nu este doar parte tehnica , ci și o formă de exprimare.

8.Bibliografie:

1. Wikipedia. (2024). Blackjack. Disponibil la: https://en.wikipedia.org/wiki/Blackjac

2. Chatgpt

3.ANSI Escape Codes Reference. Terminal Text Formatting. Disponibil la: https://en.wikipedia.org/wiki/ANSI\_escape\_code

4. infolearning.ro. (2023). Limbajul C – Tutoriale pentru începători. Disponibil la: https://infolearning.ro/cursuri