Algoritmi za rulet u programskom jeziku C

Tokić, Mateo

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni diplomski studij

ALGORITMI ZA RULET U PROGRAMSKOM JEZIKU C

Diplomski rad

Mateo Tokić

Osijek, 2018.

Sadržaj

1. UVOD

Novac pokreće svijet i bilo da se na tu rečenicu gleda pozitivno ili negativno, ona ostaje činjenica. Vodeći se time nije teško shvatiti zašto ga ljudi toliko žele i zašto postoji vrlo malo stvari koje ljudi neće napraviti kako bi zaradili nešto više. Igre poput ruleta ili drugih oblika klađenja glavni su prečac (shortcut) do bogatstva. No, kao i sve u životu ni ovaj prečac ne osigurava uvijek ono što igrač želi, ali u prirodi je čovjeka da pokuša naći rupu u pravilu.

Prema tome, postavlja se bitno pitanje; postoje li pobjedničke strategije ruleta? Iako je kuća uvijek u maloj prednosti, postoji veliki broj ljudi koji su tu prednost pokušali okrenuti u svoju korist. S druge strane, mnogi smatraju kako oni koji pobjede jednostavno imaju sreće. Međutim, iako manje poznate od strategija prisutnih kod kartaških igara, određene strategije igranja ruleta ipak postoje; od igranja šansi i duplanja uloga nakon gubitka poput Martingale metode, do ulaganja na jedan broj poput Makarov Biarritz strategije. Problem nastaje kada treba odabrati najbolju, odnosno onu koja će uvijek dovesti do pobjede, ukoliko takva uistinu postoji.

Tema ovog diplomskog rada je opisati algoritme koji se koriste prilikom igranja ruleta te u programskom jeziku C napisati kod koji će simulirati određene metode igranja ruleta.

1.1. Zadatak diplomskog rada

Zadatak ovog diplomskog rada je opisati rulet, pokazati metode kojima je moguće zaraditi na ruletu te napraviti programsku implementaciju za neke od metoda.

2. POVIJEST RULETA

Prvi oblik ruleta osmišljen je u 18. stoljeću u Francuskoj. Stoljeće prije, Blaise Pascal uveo je primitivni oblik ruleta u potrazi za perpetuum mobileom. Vjeruje se da je kotač ruleta spoj engleske igre Roly-Poly, Reiner, Ace of Hearts i EO, talijanske igre na ploči Hoca i Biribi, te "Roulette" iz već postojeće francuske igre na ploči tog imena [1]. Igra se igrala u svom sadašnjem obliku još od 1796. u Parizu. Rani opis igre ruleta u svom trenutnom obliku nalazi se u francuskom romanu “La Roulette , ou le Jour” od Jaques Lablee, koji opisuje rulet kotač u Palais Royalu u Parizu 1796. godine. Opis je "... dva mjesta za klađenje na kojima se nalaze dva broja banke, nula i dvostruka nula", odakle dobiva svoju jedinu matematičku prednost. Ova knjiga objavljena je 1801. godine.

Kockanje nije bilo vrlo popularno u to doba, ponajviše zbog činjenice da su igre na kocku bile ilegalne u mnogim zemljama Europe. No, krajem 18. stoljeća, uvođenjem strogih zakona o kockanju, klađenje je oživjelo u Francuskoj, kao i u ostatku Europe. U međuvremenu, princ Charles od Monaka se suočio s većim novčanim problemima te je odlučio iskoristiti rastuću popularnost kockanja kako bi zaradio. Otvorio je nekoliko kockarnica u Monaku, gdje je upravo rulet bio glavna igra. Slijedom toga, igra je postala vrlo popularna među aristokratima i plemićima.

S obzirom kako je postao popularna igra, rulet je kroz godine mijenjao svoj izgled. Kotači ruleta koji su bili korišteni u kasinima u Parizu kasnih 1790-ih imali su crvenu boju za jednu nulu i crnu za dvostruku nulu. Kako bi se izbjegla zbrka, zelena boja bila je odabrana za nule u kotačima ruleta počevši od 1800. godine

Godine 1843. u kasinu u njemačkom gradu Bad Homburgu, François i Louis Blanc predstavili su jedinstveni kotač s jednom nulom kako bi se natjecao protiv drugih kockarnica koje nude tradicionalni kotač s jednom i dvostrukom nulom. Ova promjena utjecala je na uravnoteženje odnosa između igrača i kasina. Međutim polje s dvostrukom nulom se još i danas koristi u nekim dijelovima svijeta kao u SAD, Kanadi, Južnoj Americi, Karibima ali i na Internetu.

U 19. stoljeću, rulet se proširio diljem Europe i SAD-a, postajući jedna od najpoznatijih i najpopularnijih kasino igara. Kada je njemačka vlada ukinula kockanje u 1860-ima, obitelj Blanc preselila se u Monte Carlo; jedino mjesto gdje je kasino bio legalan, gdje su uspostavili

kockarnicu za elitu Europe. Legenda kaže da je François Blanc navodno pregovarao s vragom kako bi dobio tajne ruleta. Legenda se temelji na činjenici da je zbroj svih brojeva na ruletu (od 0 do 36) 666, što je "broj zvijer"(vražji broj).

U prvom dijelu dvadesetog stoljeća, jedini gradovi kockarnice bili su Monte Carlo s tradicionalnim kotačem s jednom nulom i Las Vegas s kotačem koji ima dvije nule. U sedamdesetim godinama kasina su počela cvjetati širom svijeta. Do 2008. godine bilo je nekoliko stotina kockarnica diljem svijeta koje nude igre ruleta.

3. DANAŠNJI RULET

3.1. Izgled ruleta

Rulet je kompleks sastavljen od rulet kola i rulet stola. Rulet kolo je osnovni dio i sastoji se od stacioniranog drvenog kućišta sa žlijebom unutar kojeg kruži kuglica te unutrašnjeg rotirajućeg cilindra na kojem se nalazi 37 udubljenja (rupa). Svaka je rupa označena crnom ili crvenom bojom od 1 do 36, a samo 0 je označena zelenom. Rupe su ravnomjerno raspoređene na vanjskom dijelu ploče cilindra. Rulet kolo uvijek se vrti u jednom smjeru a kuglica u suprotnom.

Sl. 3.1. Izgled stola za rulet

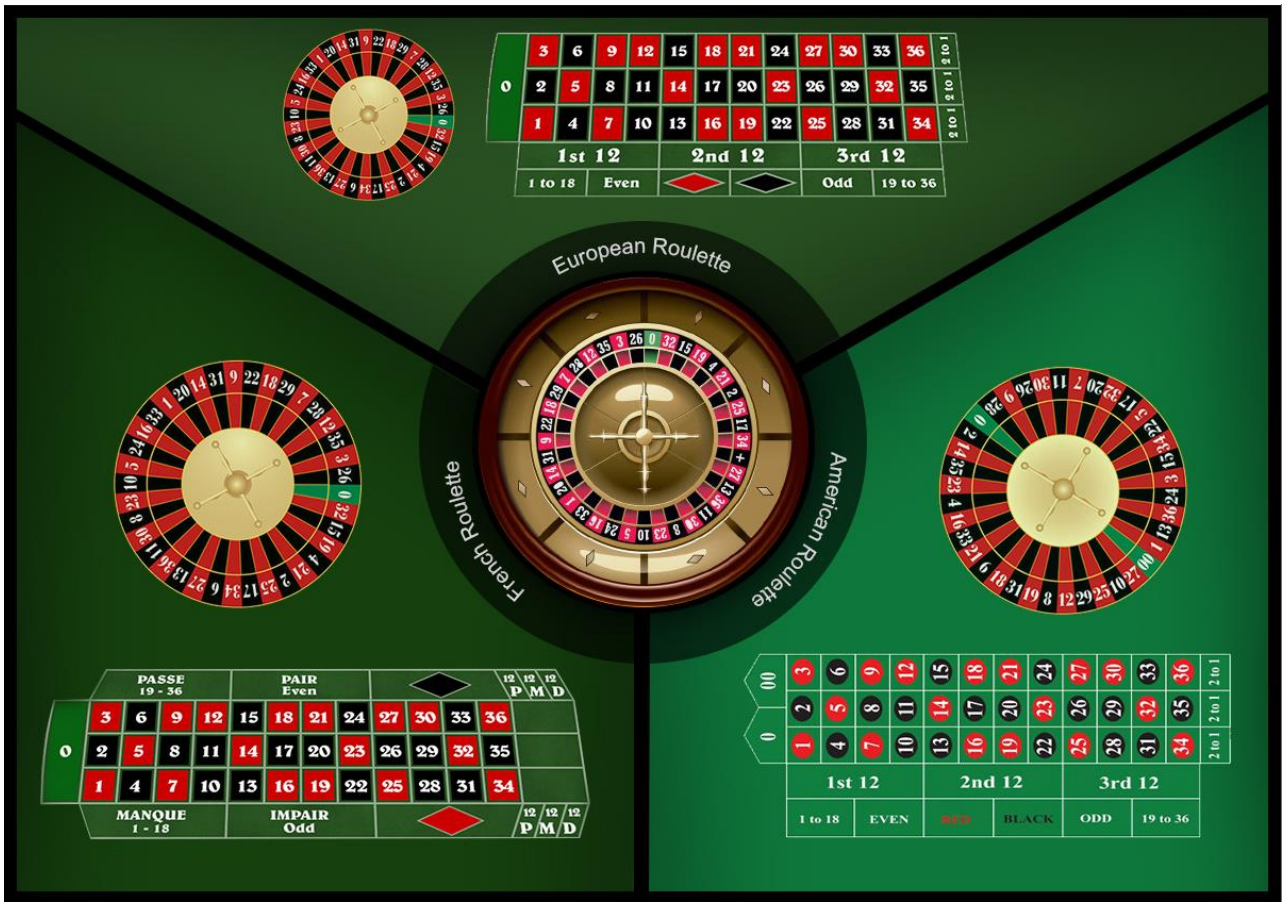
Razmještaj brojeva će se najlakše zapamtiti ako se pokretna ploča podijeli na dva dijela simetralom koja prolazi od 0 kroz sredinu tako da siječe ploču između brojeva 5 i 10. Tada su na desnoj strani u smjeru kazaljke na satu brojevi: 32, 15, 19, 4, 21, 2, 25, 17, 34, 6, 27, 13, 36, 11, 30, 8, 23 i 10, a na lijevoj strani u suprotnom smjeru kazaljke na satu: 26, 3, 35, 12, 28, 7, 29, 18, 22, 9, 31, 14, 20, 1, 33, 16, 24 i 5

Sl. 3.2. Europski i američki cilindar

Pri tom treba primijetiti da se crveni i crni brojevi, kao i veliki i mali brojevi pravilno izmjenjuju. Ali to nije slučaj s parnim i neparnim brojevima. Na desnoj strani imamo samo pet skupina po dva parna odnosno neparna broja, dok ih je na lijevoj osam skupina.

Rulet stol redovito je prekriven zelenom tkaninom, a u sredini su upisani brojevi tako da je 0 na vrhu, a zatim je redom u tri stupca i dvanaest redova upisano 36 brojeva (polja za ulaganje).

Krupije okreće cilindar koji se vrti oko svoje osi. Rulet kolo je izuzetno precizno napravljeno kako bi ispunilo svoju funkciju, ali i zaštitilo kasino. Rulet stol i rulet kolo integrirani su u jednu cjelinu. Međutim, dok je rulet kolo standardno izvedeno, rulet stolovi dolaze u tri varijante: francuski, američki i europski rulet stol. Francuski je veći te je prilaz stolu i stavljanje uloga jednostavno, ali je igra prilično spora dok je kod američke varijante pristup za ulaganje dosta slab, no igra teče brzo, često i s više od 30 zavrtaja po satu što dovodi do toga da je igra puno dinamičnija i zanimljivija. Europski stol je vrlo sličan američkom, ali sadrži samo jednu nulu.



Sl. 3.3. Američki, europski i francuski stol

3.2. Pravila igre, ulozi i dobici

Igra na rulet stolu započinje tako da krupije prije svakog zavrtaja pozove igrače na postavljanje uloga. Igrači postavljaju žetone na odgovarajuća polja prikazanim na slici gore. Nakon određenog vremena kad igrači završe s ulaganjem, krupije oglašava kraj ulaganja i zavrti rulet kolo u jednu stranu polagano i kuglicu u suprotnu vrlo energično, kako bi kuglica napravila mnogo krugova prije nego li padne na neki broj. Kad kuglica padne u svoju rupu krupije oglašava broj i boju broja koji je izašao te nakon toga kupi gubitne uloge i isplaćuje dobitke. Dobici u odnosu na ulog prikazani su u tablici 1.

Tab 3.1. Ulozi i isplate

Naziv uloga	Dobitne kombinacije	Isplata	Izgledi za dobitak
0	0	35 za 1	1:37
Puni broj	bilo koji pojedinačni broj	35 za 1	1:37
Split	bilo koja dva susjedna broja vertikalna ili vodoravna	17 za 1	2:37
Street	bilo koja tri vodoravna broja (1, 2, 3 or 4, 5, 6, itd.)	11 za 1	3:37
Korner	bilo koja 4 susjedna broja u bloku (1, 2, 4, 5 or 17, 18, 20, 21, itd.)	8 za 1	4:37
Linija	bilo kojih 6 brojeva iz 2 vodoravna reda (1, 2, 3, 4, 5, 6 or 28, 29, 30, 31, 32, 33 itd.)	5 za 1	6:37
1. kolona	1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34	2 za 1	12:37
2. kolona	2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35	2 za 1	12:37

3. kolona	3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36	2 za 1	12:37
1. dozina	1 do 12	2 za 1	12:37
2. dozina	13 do 24	2 za 1	12:37
3. dozina	25 do 36	2 za 1	12:37
Neparni	1, 3, 5, ..., 35	1 za 1	18:37
Parni	2, 4, 6, ..., 36	1 za 1	18:37
Crveno	1, 3, 5, 7, 9, 12,14, 16, 18, 19, 21, 23,25, 27, 30, 32, 34, 36	1 za 1	18:37
Crno	2, 4, 6, 8, 10, 11,13, 15, 17, 20, 22, 24,26, 28, 29, 31, 33, 35	1 za 1	18:37
1 do 18	1, 2, 3, ..., 18	1 za 1	18:37
19 do 36	19, 20, 21, ..., 36	1 za 1	18:37

Prednost kuće (kasina) izračunava se kao iznos u prosjeku koji igrač izgubi za bilo koju okladu. Ako se igrač kladi na jedan broj postoji vjerojatnost 1/37 da igrač osvoji 35 puta vrijednost uloženog i 36/37 priliku da igrač izgubi ulog. Očekivana vrijednost je:

$$-1 * \frac{36}{37} + 35 * \frac{1}{37} = -0.027$$ (2.70% prednost kuće)

To za igrača znači da u kasinu igra s vjerojatnosti 48.65 % , a kasino s vjerojatnosti 51.35%

4. METODE IGRE

Od metode se traži da osigura što više pogodaka, a da zbroj promašaja između dva pogotka bude što manji kako bi se gubici na tim segmentima mogli poništiti odgovarajućom taktikom ulaganja. Ali treba biti svjestan da je rulet ipak igra na sreću. Nije moguće manipuliranje u koje polje će kuglica pasti (barem ne legalno), stoga su rezultati slučajni - ili dovoljno blizu da se ne mogu predvidjeti. No, korištenjem određenih metoda i strategija moguće je povećati šanse za dobitak.

4.1. Shaun metoda

Metoda se temelji na velikoj vjerojatnosti pobjede (90%). Rulet ima 37 rupa za kuglicu, 36 brojeva i nulu. Cilj metode je uložiti na bilo koja 33 broja te uložiti na željenu boju broja (crno ili crveno). Ulaže se 66 žetona na način da ostanu slobodna samo 4 mjesta na stolu (broja).

Shaun metoda izgleda:

1. Ulaže se 36 žetona na boju, npr. crvenu čime je pokriveno 18 brojeva

2. Po 4 žetona uložiti na linije koje razdvajaju brojeve: 0-2, 8-11, 10-13, 17-20, 26-29, 28-31 te je pokriveno 12 crnih brojeva

3. Uložiti po 2 žetona na bilo koja 3 broja: 4, 6, 15, 22, 24, 33, 35

Sl. 4.1. Ulozi Shaun metode

Ako padne jedan od odigranih brojeva profit je šest žetona, a gubi se samo ako padne jedan od četiri neodigrana broja.

4.2. Zakon trećine

Zakon trećine je metoda koja se temelji na činjenici da se nakon određenog broja okretaja neki brojevi neće pojaviti. Ako zamislimo da je cilindar ruleta kolo sreće s 37 polja te ako kolo sreće napravi puni krug podrazumijeva se da je svako polje prošlo jedanput. Međutim, samo zato što se cilindar ruleta vrti u punim okretajima i ima 37 polja (brojeva) koji imaju istu vjerojatnost pojavljivanja, ne znači da će se svi brojevi pojaviti jednak broj puta. To znači da se u 37 okretaja trećina brojeva neće pojaviti zato što će se neki brojevi pojaviti dvaput ili više. Od 37 brojeva 25 će ih se pojaviti, a 12 neće - otprilike trećina. Prema ovoj metodi ne treba se igrati svaka vrtnja. Potrebno je zapisivati svaki broj koji izlazi sve dok se ne dobije devet različitih brojeva. Nakon toga, potrebno je uložiti na svaki od tih napisanih brojeva po jedan žeton u naredne tri vrtnje. Ako sve bude dobro, jedan će se broj pojaviti po drugi put (od vrtnje kad smo taj broj zapisali) i isplata će biti 35 za 1, a riskira se 27 žetona. Profit će biti 9 žetona (jedan žeton ostaje na pogođenom broju) tj. profit je trećina uloženih žetona.

4.3. Martingale progresija

Jedna od najkorištenijih poznatih metoda postoji gotovo od kad postoji i sama igra. Martingale je napredni sustav klađenja koji zahtijeva udvostručenje svoje prethodne oklade svaki put kada igrač izgubi. Ideja počiva na tome da udvostručavanjem nakon gubitka jednom mora pasti pogodak i tako omogućiti vraćanje gubitaka. Nažalost, sustav zahtijeva mnogo novca za igranje, a iako to ima savršen smisao tijekom vremena, to može biti opasno ako dođe serija gubitaka. Može se dosegnuti maksimalni limit stola ili ograničenje vlastitog budžeta prije nego što se mogu nadoknaditi gubici. Postoji također i nekoliko varijacija Martingale, uključujući obrnutu ili Anti-Martingale, što zahtijeva povećanje uloga nakon pobjede i smanjenje nakon gubitka. Grand Martingale je standardni Martingale, ali uključuje stavljanje malo većeg uloga od normalnog, nakon svakog gubitka (dvostruki ulog plus jedan).

4.4. Guetting metoda

Ova metoda poštuje načela da ulozi u fazi gubitka moraju biti minimalni, a ulozi u fazi dobitka moraju odgovarajuće rasti. Ulog - gubitak ne smije dovesti do gubitka uloženog, kao i do tada skupljenog dobitka. Počinje se igrati s dva žetona. U slučaju dobitka opet se ulaže dva žetona. U slučaju ponovnog dobitka poveća se ulog na tri žetona. Ako se opet pogodi ponovi se ulog i ako padne pogodak, ulog se povećava na četiri žetona itd. Taj proces ulaganja ima pet stupnjeva:

Tab 4.1. Stupnjevi ulaganja

1.stupanj	2.stupanj	3.stupanj	4.stupanj	5.stupanj
2	3	8	20	77
	4	15	30	100
	6	16	50	150

Kod dobitka se za prelazak na viši stupanj ulaganja treba ostvariti dva pogotka na nižem stupnju. U slučaju gubitka stalno se ulažu dva žetona dok se ne dobije dva puta pogodak na nižem stupnju. Tada se prelazi na drugi stupanj i ulažu se tri žetona. Ako se promaši odmah se vraća na niži stupanj i igra se s dva žetona. Ako se pogodi s tri žetona ponavlja se ulog. Bude li gubitak od tri žetona, opet se ponavlja ulog i ako se opet izgubi vraća se na prvi stupanj. Ako je bilo više uzastopnih pogodaka i došlo se do trećeg stupnja i prvi put se ulaže 16 žetona i promaši se, odmah se vraća na drugi stupanj s tri žetona. Ponovi li se gubitak vraća se na ulog od dva žetona.

4.5. D'Alembert metoda

Sustav je razvijen iz teorije ravnoteže francuskog matematičara Jeana Le Rond d'Alemberta. Njegova je teorija bila da postoje dva događaja s jednakom šansom da se dogode (npr. bacanje novčića, ili crveni, crni broj na cilindru ruleta), te ukoliko počne izlaziti puno jednakih ishoda (npr. puno crvenih brojeva), tada se suprotan ishod mora u nekom trenutku dogoditi, tako da se vrati prirodni poredak stvari, odnosno ravnoteža. Metoda je vrlo jednostavna jer nalaže povećanje uloga za jedan ako se izgubi te smanji ulog ako se dobije. Također postoji obrnuta D'Alembert metoda.

4.6. Fibonacci metoda

Ova se strategija temelji se na Fibonaccijevom nizu- redoslijedu gdje je sljedeći broj jednak zbroju prethodna dva, počevši od jedan. Metoda se koristi samo za šanse (parni/neparni broj, crveni/crni broj, 1-18/19-36) gdje je vjerojatnost dobitka i gubitka 50:50. Ako se pogodi prva oklada, niz počinje od početka. Ako se dobije oklada kasnije u nizu, jednostavno se treba vratiti dva broja i uložiti taj iznos. To se nastavlja sve dok se ne dođe do početka niza ili dok se ne ostvari željeni profit. U tablici 4.2. nalazi se primjer kako će izgledati 10 vrtnji koristeći ovu metodu:

Tab 4.2. Primjer ulaganja

Runda	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Ulog	**1**	1	2	3	5	2	3	5	2	3
Ishod	Gubitak **(-1)**	Gubitak (-1)	Gubitak (-2)	Gubitak (-3)	Dobitak (+10)	Gubitak (-2)	Gubitak (-3)	Dobitak (+10)	Gubitak (-2)	Dobitak (+6)
Stanje	**-1**	-2	-4	-7	3	1	-2	8	6	12

4.7. Makarov Biarritz metoda

Ideja metode je odabir jednog broja te ulog na odabrani puni broj nadajući se da će pasti taj broj u roku od 35 krugova. Što ranije padne, to će biti više profita. Kasnije se mogući profit smanjuje, manja je zarada, a ako se broj ne pojavi u roku od 36 krugova, pojedinac može izgubiti sve ukoliko tada prekine igru. Ili će izgubiti određenu svotu, ovisno o tome kada loptica padne na broj. Većina ljudi će se zaustaviti nakon 36 ili 37 vrtnji (jer ima 37 rupa na cilindru europskog ruleta).

4.8. Labouchere metoda

Sistem funkcionira tako da se napiše niz brojeva koji označavaju željeni ulog. Duljinu niza kao i pojedine brojeve tog niza određuje svatko za sebe. Visina uloga se računa tako da se zbroje prvi i zadnji broj u nizu. U slučaju dobitka ti se brojevi brišu iz niza, a idući ulog je opet zbroj prvog i zadnjeg broja. U slučaju gubitka visina izgubljenog uloga se dodaje na kraj niza te se kreće ispočetka. Cilj ovog sistema je da se prekriže svi brojevi, a ukoliko se to dogodi znači da je ostvaren profit, nakon čega se može započeti s novim nizom.

Ova metoda veliku popularnost duguje tomu što ga svatko može vrlo lako prilagoditi svojim potrebama. Niz se može napraviti kratkim ili dugim, a lako se određuje koliko će sistem biti riskantan. Ako se riskantnost želi držati niskom, tada se na početak niza stavlja nekoliko jedinica pa u slučaju negativnih rezultata (kada se brojevi dopisuju na kraj niza) niz neće jako rasti, tj. ni ulozi neće jako rasti.

4.9. James Bond metoda

U ovoj metodi ulaže se uvijek ista svota novca (žetona), ali po potrebi moguće je ulagati i progresivno kao u Martingale metodi. Ova metoda pokriva više od pola brojeva na stolu te se tako pokušava povećati šansa za dobitak. Za igru je potrebno 20 žetona koji se ulože kao na slici

Sl. 4.2. Ulozi

14 žetona na polje 19-36

5 žetona na liniju koja pokriva brojeve 13-18

1 žeton na broj 0

Ovakvom igrom, dobitnih će biti 25 brojeva a gubitnih 12. Mogući ishodi su:

Ako kuglica padne između 19-36 dobitak je 28 žetona.

Ako kuglica padne između 13-18 dobitak je 25 žetona.

Ako kuglica padne na 0 dobitak je 36 žetona.

Ali ako kuglica padne na brojeve od 1-12 gubitak je potpun.

5. PROGRAMSKI KOD

Programski dio ovog diplomskog rada napisan je u programskom jeziku C. On spada u proceduralne programske jezike, a nastao je 70-ih godina prošlog stoljeća. Stvoren je za rješavanje problema kodiranja sistemskih programa i jezgre operacijskog sustava UNIX. Smatra se jednim od najvažnijih jezika računalne industrije te je jedini ostao prilagođen za sve računalne platforme. C je jezik opće namjene pa se tako u njemu može napraviti: rješenje zadataka, operacijski sustavi, igre, driveri.

Radi lakšeg kodiranja te preglednosti koda, program je razdvojen na pet cjelina: Common, Roulette, Martingale, Makarov i D'Alambert. Te će u nastavku biti objašnjeni pojedini dijelovi koda.

5.1. Kod

U Common.h definirane su varijable i biblioteke koje se koriste u cijelom programu

```
4   /*************************************************************
5   *                  Includes
6   *************************************************************/
7   #include <stdint.h>
8   #include <stdio.h>
9   #include <time.h>
10  #include <Windows.h>
11
12
13  /*************************************************************
14  *                 Defines
15  *************************************************************/
16  #define ZERO_ASCII                                  48
17  #define NINE_ASCII                                  57
18
19  #define DISABLE_STAKE_LIMIT_STRING_LENGTH           3
20  #define KEYWORD_TO_DISABLE_STAKE_LIMIT              ("NE")
21  #define KEYWORD_TO_QUIT_GAME                        ("NE")
22  #define STRING_LENGTH_FOR_BUDGET_INPUT              10
23
24  #define END_GAME_STRING_LENGTH                      5
25  #define KEYWORD_TO_END_GAME                         ("KRAJ")
26
27  #define MAX_USER_STAKE_LIMIT_ENABLED                1600
28  #define MIN_USER_BUDGET                             100
29  #define MAX_USER_BUDGET                             100000
30  #define vrijeme                                     0.75
31
32
33  #define TRUE                                        1
34  #define FALSE                                       0
35
36  #define EUROPEAN_ROULETTE_NUMBER_COUNT              37
37
```

Sl. 5.1. Definicija varijabli i biblioteka

Također su definirane strukture i tipovi podataka unutar pojedinih struktura koji će se dalje koristiti u funkcijama

```
typedef enum
{
    eCommon_Martingale_Game = 1,
    eCommon_Makarov_Game,
    eCommon_Dalembert_Game
} Common_Selected_Method;      /*definirana je struktura za odabir metode
                                 te je kao prva postavljena Martingale*/

typedef enum
{
    eCommon_Roulette_Green_Field = 0,
    eCommon_Roulette_Red_Field,
    eCommon_Roulette_Black_Field
} Common_Field_Color;           /*definirana je struktura za boje a kako na ruletu
                                 ima samo jedan broj zelene boje odmah mu je dodijeljena njegova vrijednost*/

typedef struct
{
    uint8_t index;
    uint8_t value;
    BOOL even;
    Common_Field_Color color;
} Common_Roulette_Field;     /*struktura za dodijeljivanje boje, vrijednosti i indeksa broju */
```

```
typedef struct
{
    BOOL askUserForStake;                /* zastavica je postavljena u TRUE ako je igrac pogodio zadnju vrtnju,
    int64_t currentStake;                /* trenutni ulog */
    uint64_t lostSinceLastWin;           /* ukupno izgubljeno od posljednje pobjede */
    Common_Field_Color lastStakeColor;   /* posljednja igrana boja */
    int64_t startingBudget;              /* početni budžet */
    int64_t currentBudget;               /* trenutni budžet */
    int64_t currentBudgetPerStake;       /* Makarov metoda- ukupni budžet se dijeli na 37 */
    int64_t stakeLimit;                  /* limit uloga */
    BOOL playingWithStakeLimit;          /* zastavica koja odreduje igra li se s limitom ili bez */
    uint64_t numberOfRoundsPlayed;       /* broj odigranih vrtnji */
    BOOL finishGame;                     /* zastavica koja označava da li igra gotova ili ne */
} Common_PlayerStatus;
```

Sl. 5.2. Definicija struktura i tipova podataka

U Common.c napisan je dio programa koji je zajednički za sve metode i korisnika se pita da izabere metodu s kojom želi igrati rulet:

```
/**=========================================================================**
* Function      : Common_AskUserToSelectAMethod
*
* Description   : Korisnika se pita da odabere metodu s kojom će igrati rulet
*
**=========================================================================**/
void Common_AskUserToSelectAMethod(int *selectedMethod)
{
    printf("Odaberite metodu s kojom zelite igrati rulet.\n");
    printf("%d. Martingale metoda\n%d. Makarov Biarritz metoda\n%d. D'Alembert metoda\n",
        eCommon_Martingale_Game, eCommon_Makarov_Game, eCommon_Dalembert_Game);
    printf("Metoda: ");
    scanf(" %d", selectedMethod);

    PREPARE_INPUT_BUFFER();

    switch (*selectedMethod)
    {
    case eCommon_Martingale_Game:
        printf("Odabrali ste Martingale metodu!\n\n");
        break;
    case eCommon_Makarov_Game:
        printf("Odabrali ste Makarov Biarritz metodu!\n\n");
        break;
    case eCommon_Dalembert_Game:
        printf("Odabrali ste D'Alembert metodu!\n\n");
        break;
    default:
        printf("Odabrali ste metodu koja ne postoji, te je odabrana Martingale metoda!\n\n");
        *selectedMethod = eCommon_Martingale_Game;
        break;
    }
}
```

Sl. 5.3. Odabir metode

Funkcija kojom se korisnika pita želi li igrati s limitom uloga:

```
/**=========================================================================**
* Function      : Common_AskUserToDisableStakeLimit
*
* Description   : Korisnika se pita želi li igrati s limitom uloga ili bez
*
**=========================================================================**/
void Common_AskUserToDisableStakeLimit(BOOL *isStakeLimitEnabled)
{
    char userDisableLimit[DISABLE_STAKE_LIMIT_STRING_LENGTH] = {0};
    uint8_t i = 0;
    printf("Rulet se igra sa odredjenim limitom uloga. U ovom slucaju, limit je %d[HRK].\n",
        MAX_USER_STAKE_LIMIT_ENABLED);
    printf("Ako zelite igrati bez limita, unesite %s kad to budete mogli. Inace stisnuti enter za nastavak.\n",
        KEYWORD_TO_DISABLE_STAKE_LIMIT);
    printf("Zelite li igrati s limitom? ");
    fgets(userDisableLimit, DISABLE_STAKE_LIMIT_STRING_LENGTH, stdin);

    if (_stricmp(userDisableLimit, KEYWORD_TO_DISABLE_STAKE_LIMIT) == 0)
    {
        *isStakeLimitEnabled = FALSE;
    }

    PREPARE_INPUT_BUFFER();
}
```

Sl. 5.4. Igranje s limitom

Od igrača se traži da unese svoj budžet. Također se provjerava ispravnost unesenih brojeva (duljina broja, jesu li svi znakovi brojevi ili ima i slova):

```
/**============================================================================
* Function      : Common_AskUserToInputBudget
*
* Description   : Igrača se pita da unese svoj budžet te se provjerava
                  ispravnost unesenog budžeta (dužina stringa, jesu li svi znakovi brojke)
*
**============================================================================**/
void Common_AskUserToInputBudget(int64_t *userBudget)
{
    char inputString[STRING_LENGTH_FOR_BUDGET_INPUT] = {0};
    int i;
    int digit;
    int stringLength;
    BOOL validInput = TRUE;
    char *stringToInt;

    printf("\nUnesite svoj budget.\n");
    printf("Minimalni budget je %d[HRK]. Maksimalni budget je %d[HRK].", MIN_USER_BUDGET, MAX_USER_BUDGET);
    do
    {
        printf("\nUnesite budget u rasponu od [%d,%d][HRK]\n", MIN_USER_BUDGET, MAX_USER_BUDGET);
        printf("Budget: ");

        fgets(inputString, STRING_LENGTH_FOR_BUDGET_INPUT, stdin);

        PREPARE_INPUT_BUFFER();

        stringLength = strlen(inputString);

        for (i = 0; i < stringLength-1; i++)
        {
            digit = (int)inputString[i];

            if (digit < ZERO_ASCII || digit > NINE_ASCII)
            {
                printf ("Krivi unos. Otkriveno \'%c\' na indeksu %d.\n", inputString[i], i);
                validInput = FALSE;
                break;
            }
        }

        if (validInput == FALSE)
        {
            validInput = TRUE;
            continue;
        }
        *userBudget = strtol(inputString, &stringToInt, 10);
    } while ((*userBudget) < MIN_USER_BUDGET || (*userBudget) > MAX_USER_BUDGET);
}
```

Sl. 5.5. Unos budžeta

Ispunjavanje cilindra brojevima te njihovim pripadajućim bojama i obavještava se igrač da se radi o europskom ruletu:

```
/**============================================================================**
* Function      : Common_PopulateEuropeanRouletteCylinder
*
* Description   : Popunjavanje cilindra odgovarajućim brojevima i pripadajućim bojama
*
**============================================================================**/
void Common_PopulateEuropeanRouletteCylinder(Common_Roulette_Cylinder *cylinder)
{
    uint8_t i;
    uint8_t idx;

    for (i = 1; i < EUROPEAN_ROULETTE_NUMBER_COUNT; i++)
    {
        idx = Common_GetFieldIndexFromValue(i);
        cylinder->field[idx].index = idx;
        cylinder->field[idx].value = i;
        cylinder->field[idx].even = i % 2 ? 0 : 1;
        cylinder->field[idx].color = Common_GetFieldColorFromValue(i);
    }

    cylinder->field[0].index = 0;
    cylinder->field[0].value = 0;
    cylinder->field[0].even = 0;
    cylinder->field[0].color = eCommon_Roulette_Green_Field;

    printf("Igrate na europskom stolu!\n\n");
}
```

Sl. 5.6. Popunjavanje cilindra

Funkcija za dobivanje slučajnog broja

```
/**============================================================================**
* Function      : Common_SpinRouletteWheel
*
* Description   : Izbacivanje slučajnog broja
*
**============================================================================**/
void Common_SpinRouletteWheel(uint8_t lastFieldIndex, uint8_t *newFieldIndex)
{
    uint16_t numberOfCylinderFieldsPassedInASpin = 0;

    numberOfCylinderFieldsPassedInASpin = (rand() % EUROPEAN_ROULETTE_NUMBER_COUNT) + 1;

    *newFieldIndex = (lastFieldIndex + numberOfCylinderFieldsPassedInASpin) % (EUROPEAN_ROULETTE_NUMBER_COUNT);
}
```

Sl. 5.7. Izbacivanje slučajnog broja

Funkcija za ispis statistike igrača (broj vrtnji, željeni profit, gubitak) nakon što igrač završi s igrom:

```
/**=============================================================================**
* Function      : Common_PrintGameStatistics
*
* Description   : Ispis statistike igrača nakon završetka igre
*
**=============================================================================**/
void Common_PrintAfterGameStatistics(Common_PlayerStatus *player)
{
    printf("\nKraj igre!\nBroj odigranih vrtnji: %llu\nVrijeme igranja: %f minuta\n",
           player->numberOfRoundsPlayed,player->numberOfRoundsPlayed*vrijeme);
    printf("Pocetni budget: %lld[HRK]\nTrenutni budget: %lld[HRK]\n", player->startingBudget, player->currentBudget);
    if (player->currentBudget > player->startingBudget)
    {
        printf("Profit: %lld[HRK]\n", player->currentBudget - player->startingBudget);
    }
    else if (player->startingBudget > player->currentBudget)
    {
        printf("Izgubljeno: %lld[HRK]\n", player->startingBudget - player->currentBudget);
    }
    else
    {
        printf("Zavrsili ste igru s jednakim budgetom kao sto ste i poceli.\n");
    }
```

Sl. 5.8. Statistika igrača

Budući da je programski kod vrlo sličan za sve metode, u ovom diplomskom radu bit će prikazan dio koda iz D'Alambert metode.

Prije početka igre, postavljaju se inicijalne vrijednosti na nulu

```
void DAlembert_PlayRoulette(Common_Roulette_Cylinder *cylinder, Common_PlayerStatus *player)
{
    uint8_t lastFieldIndex = 0;
    uint8_t newFieldIndex = 0;
    int64_t step = 0;
    int64_t desiredProfit = 0;
    int64_t rowCounter = 0;
    char endGame[END_GAME_STRING_LENGTH] = {0};
    BOOL askForColor = TRUE;
    BOOL askForDesiredProfit = TRUE;
```

Sl. 5.9. Definiranje inicijalnih vrijednosti

U ovom dijelu koda od igrača se traži da unese koliki mu je željeni profit te mu se na ekranu ispišu određene poruke ovisno igra li s limitom uloga ili bez njega:

```
if (player->playingWithStakeLimit == TRUE)
{
    player->stakeLimit = MAX_USER_STAKE_LIMIT_ENABLED;
    printf("Igrate D'Alembert metodom s limitom uloga od %llu[HRK] i budzetom od %llu[HRK].\n",
     player->stakeLimit, player->startingBudget);
}
else
{
    player->stakeLimit = player->startingBudget;
    printf("Igrate D'Alembert metodom bez limita uloga i budzetom od %llu[HRK].\n", player->stakeLimit);
}

if (askForDesiredProfit == TRUE)
{
    printf("\nUnesite zeljeni profit (igra se dok se ne dostigne zeljeni profit ili dok ne se izgubi sve): ");
    scanf(" %d", &desiredProfit);
    printf("\nUnijeli ste %d HRK kao zeljeni profit.\n", desiredProfit);
    askForDesiredProfit = FALSE;
}
```

Sl. 5.10. Unos profita

Nakon što je igrač unio budžet s kojim raspolaže, željeni profit i s kojim ulogom će igrati, pita ga se na koju boju želi uložiti. Nakon što je odabrao jednu boju, u slučaju promašaja boja se automatski mijenja u suprotnu sve dok se ne pogodi boja. Također se ažurira stanje budžeta i trenutnog uloga:

```
if (askForColor == TRUE)
{
    do
    {
        printf("Odaberite boju [1 - crveno; 2 - crno]: ");
        scanf(" %u", &player->lastStakeColor);

        PREPARE_INPUT_BUFFER();
    } while (player->lastStakeColor != eCommon_Roulette_Red_Field && player->lastStakeColor != eCommon_Roulette
    askForColor = FALSE;
}
else
{
    if (player->lastStakeColor == eCommon_Roulette_Red_Field)
    {
        player->lastStakeColor = eCommon_Roulette_Black_Field;
        Sleep(DALEMBERT_SPIN_SLEEP_TIME);
    }
    else
    {
        player->lastStakeColor = eCommon_Roulette_Red_Field;
        Sleep(DALEMBERT_SPIN_SLEEP_TIME);
    }
}
```

Sl. 5.11. Odabir boje

Od igrača se traži da unese početni ulog te mu je taj ulog ograničen ili limitom uloga ili budžetom koji ima:

```
while (!player->finishGame)
{
    if (player->askUserForStake)
    {
        do
        {
            if (player->playingWithStakeLimit == TRUE)
            {
                printf("Unesite pocetni ulog. Iduci ulog bit ce uvecan za pocetni ulog. Limit uloga = %llu[HRK]: ",
                       player->stakeLimit);
            }
            else
            {
                printf("Unesite pocetni ulog. Iduci ulog bit ce uvecan za pocetni ulog. Ograniceni ste trenutnim
                       budzetom = %llu[HRK]: ", player->currentBudget);
            }
            scanf(" %llu", &player->currentStake);

            /* prepare input buffer by ignoring unused characters */
            PREPARE_INPUT_BUFFER();
        } while ((player->currentStake > player->stakeLimit) || (player->currentStake > player->currentBudget) ||
        (player->currentStake == 0));
        player->askUserForStake = FALSE;
        step = player->currentStake;
    }
```

Sl. 5.12. Unos početnog uloga

Provjerava se je li igrač pogodio boju te ako je, je li ostvario željeni profit. Ako je, ispisat će mu se poruka i igra će završiti.

```
if (cylinder->field[newFieldIndex].color == player->lastStakeColor)
{
    player->currentBudget += player->currentStake * DALEMBERT_STAKE_MULTIPLIER;
    player->currentStake -= step;
    player->lostSinceLastWin = 0;

    if ((player->currentBudget - player->startingBudget) >= desiredProfit )
    {
        printf("\nOstvarili ste zeljeni profit od %d HRK. Cestitamo!\n", desiredProfit);
        player->finishGame = TRUE;
    }
}
```

Sl. 5.13. Provjera ostvarenog profita

Ako igrač nije ostvario željeni profit, povećava se idući ulog te se provjerava ima li još novaca u budžetu, je li idući ulog veći od budžeta te prelazi li idući ulog limit uloga:

```
else
{
    player->currentStake += step;

    if (player->currentStake > player->currentBudget)
    {
        if (player->currentBudget == 0)
        {
            player->finishGame = TRUE;
            printf("\nPotrosili ste sve i za vas je igra zavrsena. Hvala i dovidenja.\n");
        }
        else
        {
            player->askUserForStake = TRUE;
            printf("\nIduci ulog = %llu[HRK] je veci od trenutnog budzeta = %llu[HRK]. Unesite novi ulog.\n",
    player->currentStake, player->currentBudget);
        }
    }
    else if (player->currentStake > player->stakeLimit)
    {
        printf("\nIduci ulog = %llu[HRK] je veci od limita uloga = %llu[HRK].\n", player->currentStake,
   player->stakeLimit);

        player->askUserForStake = TRUE;
    }
}
```

Sl. 5.14. Praćenje iznosa budžeta

6. ZAKLJUČAK

Rulet, kao jedna od najstarijih i najprivlačnijih igara na sreću, dobro je poznat vrlo širokoj populaciji. Usporedno s razvitkom popularnosti ove igre, tijekom godina razvijale su se i razne strategije igranja, odnosno povećanja šanse za pobjedom. Strategije, bilo da se odnosile na boju koju treba izabrati nakon gubitka ili dobitka, ili koliko treba povećati ili smanjiti ulog nakon svakog igranja, sve imaju isti cilj - povećati prednost igrača nad kasinom. Uz to, strategije su dobre i po tome što potiču igrača da igra metodičnije te odredi najveći gubitak koji može pretrpiti, kao i koliki iznos želi zaraditi. Iako određivanje najvećeg gubitka ima smisla, logika nalaže kako ne bi trebalo biti gornje granice. No, iako je glavni cilj svih igara na sreću zaraditi što više, određivanjem svote koju pojedinac želi zaraditi, smanjuje se rizik da izgubi sve.

Koliko god svaka metoda imala svoje individualne prednosti potrebno je imati na umu da niti jedna strategija ruleta nije sigurna oklada. Bez obzira na to koliko uvjerljivo izgledaju, niti jedna od njih ne može osigurati nikakvu stalnu statističku prednost nad kućom.

LITERATURA

[1] Wikipedia, rulet https://en.wikipedia.org/wiki/Roulette [19.6.2017.]

[2] Pravila ruleta, https://www.roulettephysics.com/how-to-play-roulette/ [19.6.2017.]

[3] Izgled ruleta, https://www.mastersofgames.com/rules/roulette-rules.htm [19.6.2017.]

[4] M. Kuljiš, Škola igre na ruletu, Hrvatska lutrija, Zagreb 1998. [20.6.2017.]

[5] Metode za rulet, http://onlineroulette.org.uk/systems/ [21.6.2017.]

[6] Zakon trećine metoda, http://www.rouletteonline.net/roulette-strategy/law-of-the-third-betting-system/ [21.9.2018.]

[7] James Bond metoda, https://www.roulettesites.org/strategies/james-bond/ [15.9.2018.]

[8] Programski jezik C, https://www.cprogramming.com/ [18.8.2017.]

[9] Tutorial za C, https://www.tutorialspoint.com/cprogramming [18.8.2017.]

[10] Shaun metoda, http://www.roulettetoken.com/shaun-roulette-system/ [21.9.2018.]

[11] Izgled za dobitak na ruletu, https://www.roulettesites.org/rules/odds/ [21.6.2017.]

SAŽETAK

ALGORITMI ZA RULET U PROGRAMSKOM JEZIKU C

U ovom radu obrađena je jedna od najpopularnijih igara na sreću- rulet. U prvom dijelu rada opisana je povijest ruleta, pravila igre te metode koje postoje kako bi se maksimizirao profit igrajući rulet. Drugi dio rada posvećen je programskoj implementaciji Martingalove, Makarov Biarritz i D'Alambert metode. Implementacija je odrađena u programskom jeziku C.

Ključne riječi: rulet, ulog, metoda

ABSTRACT

ALGORITHMS FOR ROULETTE IN C

This paper deals with one of the most popular games of chance roulette. The first part of the article describes the history of roulette, game rules and methods that exist to maximize profit by playing roulette. The second part of the paper is devoted to the program implementation of Martingale, Makarov Biarritz and D'Alambert methods. Implementation is done in programming language C.

Keywords: roulette, bet, method

ŽIVOTOPIS

Mateo Tokić rođen je 20.7.1992. u Virovitici te živi u Osijeku. Prva četiri razreda osnovne škole završava u područnoj školi u Briješću, a osnovnoškolsko obrazovanje završava u OŠ Vladimira Nazora u Čepinu. Obrazovanje nastavlja završavanjem I. Gimnazije Osijek te se upisuje na Elektrotehnički fakultet u Osijeku, smjer računarstvo, gdje je redoviti student. Također radi četiri godine kao djelitelj Texas holdem pokera u Luckia kasinu te je odradio stručnu praksu kao tester poslovnih aplikacija u Crozu.

Mateo Tokić

_____