

Analiza BlackJack strategij

Poročilo

Zala Stopar Špringer

Maj 2022

1 Kratek opis

Za projekt pri predmetu Matematika z računalnikom sem raziskovala različne strategije stavljanja. Svoj program sem napisala v programskem jeziku *Python*. Aplikacijo sem oblikovala s pomočjo paketa *Pygame*. Glavni del igre je napisan v datoteki **gaming.py**

Na začetku sem se lotila programiranja in dizajniranja navadne igre BlackJack, ki sem jo tudi dokončala. Posameznik se lahko pomeri proti hiši. Skupaj je v igri 6 kompletov kart. Na vsakem koraku ima igralec na voljo akciji HIT in STAND, v pravih okoliščinah pa tudi INSURANCE, DOUBLE DOWN in SPLIT. V različih igralnicah se pravila malo razlikujejo. Za moj program sem določila, da po tem, ko igralec izbere opcijo SPLIT, nima več na voljo opciji DOUBLE in INSURANCE.

Opazovala sem rezultate sledečih strategij:

- Paroli system
- 1 3 2 6 system
- Reverse Labouchere
- Card counting
- Martingale
- Oscar's Grind
- Labouchere

Te so zbrane v mapi **functions** v datoteki **strategies.py**, njihov opis pa sledi kasneje v poročilu.

2 Opis strategij

2.1 Paroli system

Kot pri večini strategij, je tudi pri Paroli sistemu potrebno določiti enoto stavljanja. Pri tem sistemu spreminjamo stavo glede na število zaporednih iger,

ki smo jih zmagali. Po zmagi, se stava v naslednji igri podvoji. V mojem primeru sem stavo podvojevala do treh zaporednih zmag, nato pa sem se vrnila na začetno enoto. Seveda je lahko meja števila zaporednih zmag, ko se vrnemo na začetno stavo višja, je pa zato taka igra tudi bolj tvegana.

2.2 1 3 2 6 System

Tudi 1326 sistem spreminja velikost stave glede na število zaporednih zmag. Po zmagi, velikost stave v naslednji igri znaša $enota \times 3$, po dveh zaporednih stavah v naslednji igri stavimo $enota \times 2$ in po treh zaporednih zmagah stavimo $enota \times 6$. Nato se vrnemo na začetek in postopek ponovimo. Od tod izvira ime 1326. Seveda se vrnemo nazaj na prvotno enoto, če zadnjo igro izgubimo.

2.3 Reverse Labouchere

Pri tej strategiji začnemo z oblikovanjem zaporedja števil, katerih vsota se mora sešteti v zaželen dobiček. Za prvo stavo seštejemo prvo in zadnjo število v zaporedju. Če mešanje zmagamo, vsoto dodamo na konec našega zaporedja, sicer pa števili odstranimo iz seznama. nato zopet stavimo vsoto prvega in zadnjega števila. Ko (Če) je naše zaporedje prazno, smo zaslužili željeno vsoto.

2.4 One half increase

Podobno kot prvi dve strategiji, ta strategija spreminja stavo glede na število zaporednih zmag. Prav tako pa začnemo stavljati z eno enoto, ki je v naprej izbrana. Ko zmagamo dve mešanji zapored, povečamo stavo za polovico enote. To ponavljamo, dokler ne izgubimo mešanja. Takrat se vrnemo na prvotno enoto in ponovno začnemo cikel.

2.5 Card counting

Tako kot zgoraj opisane strategije, deluje tudi ta glede na število zaporednih zmag. Pred začetkom igranja se vsaki karti določi svojo vrednost, v mojem primeru so karte v vrednosti od 2 do 6 znašale $+1$, karte 7, 8, 9 so predstavljale vrednost 0, 10 ali višje pa -1 . Igralec po vsaki karti, ki pade na mizo spremeni štetje glede na vrednost karte. Ker igralnice igrajo ogro z večimi paketi kart, je treba vrednost štetja spremeniti, tako da ga delimo s številom preostalih paketov kart. Višje, ko je štetje, višja je stava. V mojem primeru sem stavila osnovno enoto, če je bilo štetje 1 ali manjše, sicer pa $(\text{štetje} - 1) \times enota \times 10$.

2.6 Martingale

Pri tej strategiji se višina stave spreminja glede na število zaporednih porazov. Po vsakem porazu, je stava v naslednji igri dvakrat večja. Ko prvič zmagamo, si povrnemo našo izgubo in nekaj malega pridobimo. Seveda pa stave lahko zelo hitro narastejo, zato je čim boljše začeti z minimalno stavo, da imamo možnost zmagati eno igro, preden nam zmanjka denarja.

2.7 Oscar's Grind

Tudi ta strategija deluje na podlagi števila zaporednih izgubljenih iger. Dokler ne zmagamo stavljamo v višini ene enote. Ob zmagi stavimo edve enoti in stavljamo dokler ne zmagamo, nato stavimo 3 enote, Postopek ponavljamo, dokler skupno ne zaslužimo ene enote. Nato vse ponovimo.

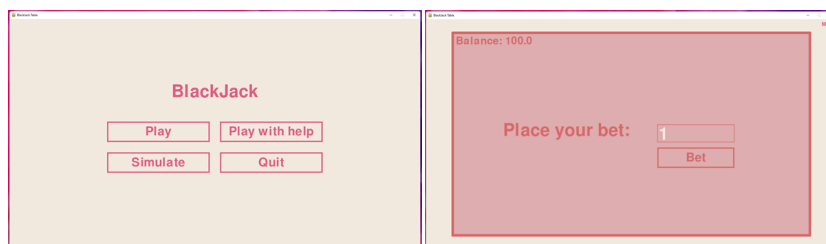
2.8 Labouchere

Ta strategija deluje zelo podobno kot Reverse Labouchere. Prav tako se začne z zaporedjem števil. Prva stava je enaka vsoti prve in zadnje številke v zaporedju. Če igro zmagamo, številki prečrtamo, sicer pa vsoto dodamo na konec seznama. Naslednja stava pa zopet znaša vsoto prve in zadnje številke.

3 Oblikovanje vmesnika

Pri tem projektu sem se prvič spoznala s knjižnjico *Pygame*. Večina objektov v aplikaciji je sestavljena iz različnih kvadratov, z različnimi lastnostmi. Poleg tega si delo olajšala s tem, da sem ustvarila razred **Button**, ki mu podamo pozicijo, besedilo v gumbu, 3 različne barve, velikost in obrobo. Pri ustvarjanju razreda **InputBox** sem imela več težav. Zato sem si kodo izposodila iz spletne strani Stack Overflow, dodala pa sem nekaj vrstic, ki sem jih potrebovala za svojo igro.

Ker sem se z omenjeno knjižnjico šele spoznavala, sem se za začetek lotila oblikovanja lažjih oken. Po tem, ko sem izbrala ustrezno bravno paleto sem oblikovala začetno stran igre, ki je shranjena v mapi **windows** pod imenom *menu_window.py*. Kot je vidno na sliki, imamo na uvodi strani na voljo nekaj možnosti, lahko igramo navadno igro, lahko igramo igro s pomočjo ali pa simuliramo igro.



Slika 1: Menu and Bet windows

Ne glede na možnost, ki jo izberemo, nas aplikacija postavi na stran, kjer položimo vsoto, s katero bi radi igrali. V primeru navadne igre in igre s pomočjo se nato najdemo na strani, kjer stavimo prvo stavo. To stran lahko tudi vidimo na sliki 1. V nasprotnem primeru se odpravimo na stran (izbira simulacije), kjer poleg vsote, s katero bi radi igrali izberemo tudi število igre, ki bi jih radi igrali in število mešanj v posamezni igri. Poleg tega izberemo tudi strategijo stavljanja, ki bi jo radi igrali in osnovno enoto stave. Ko končamo z igranjem igre nas vrže na zaključno stran, kjer lahko prevzamemo dobiček.

Nekaj novih stvari sem se tudi nučila pri oblikovanju kart. Da nisem risala usake karte posebej, sem ustvarila razred po imenu **Card** s parametri brava, vrednost in pozicija karte. Z metodo *.draw()* lahko karto narišemo, z metodo *.card_back()* pa narišemo hrbtno stran karte. Za sliko na hrbtni strani karte in za simbole barv sem poiskala ikone na spletni strani <https://www.flaticon.com/>.

4 Še sledi

- dokončati igro, kjer je navoljo pomoč
- napisati funkcijo za računanje verjetnosti zmage v določeni situaciji
- dodati strategijo štetja kart
- napisati funkcijo za hranjenje podatkov iz simulacije različnih strategij
- analizirati dobljene podatke