Tema: *Realizacija igre šaha*

Predmet: Praktikum iz objektno-orijentisanog programiranja

Ja: Marko Mladenović 72-2022

Programski jezik: C++

Radjeno je po uzoru na pravu igru šaha, sa par malih modifikacija koje ne uticu toliko na samu igru. Pravila po kojima se igra:

* Potezi se unose u formatu **[A-H][1-8] [A-H][1-8]**, primer: A2 A3.
* Figure su obelezene u formatu *BOJA\_FIGURE TIP\_FIGURE*, primer: B T: Beli Top.
* BOJA\_FIGURE moze biti C – Crna, B – Bela.
* TIP\_FIGURE moze biti: P – Pijun, T – Top, S – Skakac(Konj), L – Lovac, Q – Kraljica(Queen), K – Kralj.
* Prvi na potezu je uvek BELI igrac, ako uspesno odigra potez onda igra CRNI igrac.
* U slucaju da potez nije validan onda igrac ponavlja potez sve dok ne odigra validan potez.
* Igra se zavrsava kada je POJEDEN jedan od kraljeva.
* Tabla se stampa, tj. prikazuje nakon svakog poteza, kao i na pocetku.

**Figura** je apstraktni tip, koji predstavlja osnovu za kreiranje ostalih šahovskih figura. Kreira se tako sto se prosledi *BOJA\_FIGURE i TIP\_FIGURE*(B – Bela, C - Crna). Moguce je proveriti da li data figura moze odigrati odredjeni potez tako sto se proslede pocetne i zavrsne koordinate figure funkcijom *proveriPotez(...).* Moguce je stampati figuru u formatu *BOJA\_FIGURE TIP\_FIGURE*. Moguce je dohvatiti tip i boju figure funkcijama *getTip()* i *getBoja()*.

**Pijun** je *figura* koja ima prepisanu funkciju *proveriPotez(...)*, moze da se krece samo jedno polje napred, ili jedno polje dijagonalno u slucaju da je tu figura suprotne boje, a u slucaju da se taj pijun prvi pu pomera, onda moze 2 polja u napred. Ispsisuje se u formatu *BOJA\_FIGURE P*.

**Top** je *figura* koja ima prepisanu funkciju *proveriPotez(...)*, moze da se krece po X ili Y u bilo kom smeru. Ispisuje se u formatu *BOJA\_FIGURE T*.

**Lovac** je *figura* koja ima prepisanu funkciju *proveriPotez(...)*, moze da se krece dijagonalno u sva 4 smera. Ispisuje se u formatu *BOJA\_FIGURE L*.

**Skakac** je *figura* koja ima prepisanu funkciju *proveriPotez(...)*, moze da se krece u obliku *Г* u 8 razlicitih smerova. Ispisuje se u formatu *BOJA\_FIGURE S*. Skakac predstavlja konja, ali slovo K je zauzeto za kralja pa sam odlucio da koristim skakac zbog bolje jasnoce.

**Kraljica** je *figura* koja ima prepisanu funkciju *proveriPotez(…)*, moze da se krece dijagonalno u sva 4 smera, kao I po X/Y osi u sva cetri smera. Ispisuje se u formatu *BOJA\_FIGURE* *Q*. Q – Queen, da ne bi doslo do mesanja sa kraljem.

**Kralj** je *figura* koja ima prepisanu funkciju *proveriPotez(…)*, moze da se krece samo jedno polje, ali u bilo kom smeru. Ispisuje se u formatu *BOJA\_FIGURE K*. U slucaju da je kralj pojeden igra se zavrsava I pobedio je igrac koji je pojeo protivnickog kralja.

**Potez** je tip koji sadrzi informacije o jednom uspesno odigranom potezu. Moguce je dobiti tekstualnu reprezentaciju poteza funkcijom *getPotez()*, funkcija vraca string u format *RBR\_POTEZA:POCETNO\_POLJE -> ODIGRANO\_POLJE*. Moguce je proslediti potez na standardni output stream(*cout*) u istom format.

**Polje** je tip koji sadrzi informacije o jednom polju, tj kvadratu na tabli, sadrzi x I y koordinatu, kao I informaciju o tome koja figura se nalazi trenutno(nullptr ako je prazno). Metoda *getFigura(...)* vraca figuru koja se nalazi na datom polju.

**Tabla** je tip koji vodi racuna o toku igre. Tabla sadrzi matricu figura 8x8, ta matrica predstavlja tablu, u slucaju da nema figura na nekom polju onda je *nullptr*. Moguce je inicijalizovati igru pozivanjem metode *inicijalizujIgru()*, metoda postavlja belog igraca kao pocetnog i postavlja broj odigranih poteza(*odigranoPoteza*) na 0, takodje postavlja figure na svoje odredjene pozicije. Poseduje metodu *proveriPolje(…)* koja proverava da li polje koje odgovara prosledjenim koordinatama ima figuru na sebi I da li koje je boje ta figura. Moguce je dohvatiti bilo koje polje sa table metodom *getPolje(…)*. Ima metodu *odigrajPotez(…)* koja za prosledjene pocetne I zavrsne koordinate prvo proverava da li je moguce odigrati dati potez, ukoliko je moguce, potez biva odigran, a ako nije moguce onda baca jedan od exception-a, u zavisnosti od problema. Moguce je stampati tablu, figure se stampaju u formatu *BOJA\_FIGURE TIP\_FIGURE*, odvojene su uspravnim crtama ‘|’, crna polja su obelezena sa tri znaka ‘#’, tj. ‘###’, a bela polja su obelezena praznim prostorom izmedju dve uspravne crte, tj. ‘|’. Svako se nalazi izmedju dve ‘|’. Nakon odigrane partije moguce je odstampati sve odigrane poteze metodom *prikaziPoteze()*. Metoda *stampajPojedene()* omogucava prikaz figura koje su igraci pojeli. Metoda prikaziPoteze() omogucava igracima da na kraju igre prikazu sve uspesno odigrane poteze. Metoda *proveriSah(…)* proverava da li je neki od kraljeva pod napadom, u slucaju da jeste onda se ispisuje informacija o tome koji je kralj u opasnosti, u slucaju da igrac ne pomeri kralja onda protivnicki igrac moze da ga pojede u narednom potezu.

**NevalidanPotezException** je *exception* koji ima prepisanu metodu *what()*. Javlja se u slucaju da odredjena figura ne moze odigrati taj potez.

**PotezIzvanTableException** je *exception* koji ima prepisanu metodu *what()*. Javlja se u slucaju da su koordinate za potez van table(npr A9, J4…).

**IgracNijeNaPotezuException** je *exception* koji ima prepisanu metodu *what()*. Javlja se u slucaju da igrac pokusa da pomeri protivnicku figure.

**PraznoPoljeException** je *exception* koji ima prepisanu metodu *what()*. Javlja se u slucaju da igrac odabere polje na kome nema figure.