

Fox and geese

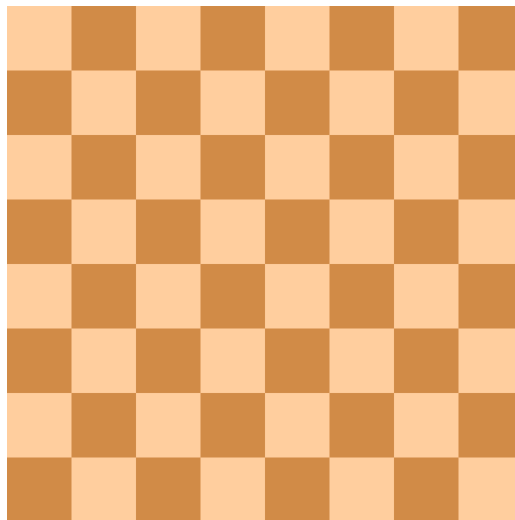
Prvi zadatak iz Androida (školska 2023/2024)

Cilj zadatka:

Cilj ovog zadatka je implementirati igru „fox and geese“ koja se igra između dva igrača na jednom uređaju ili između dva igrača preko mreže (klijent-server). Realizacija sa dva igrača na jednom uređaju nosi 15 poena dok realizacija igre preko mreže nosi 30 poena. Igra se u oba slučaja odvija isto, a opis igre je dat u nastavku.

Pravila igre:

„Fox and geese“ je igra koja postoji u više varijanti, zavisno od toga na kakvoj tabli se igra, ali će u zadatku biti potrebno implementirati samo jednu varijantu, koja će se igrati na šahovskoj tabli:



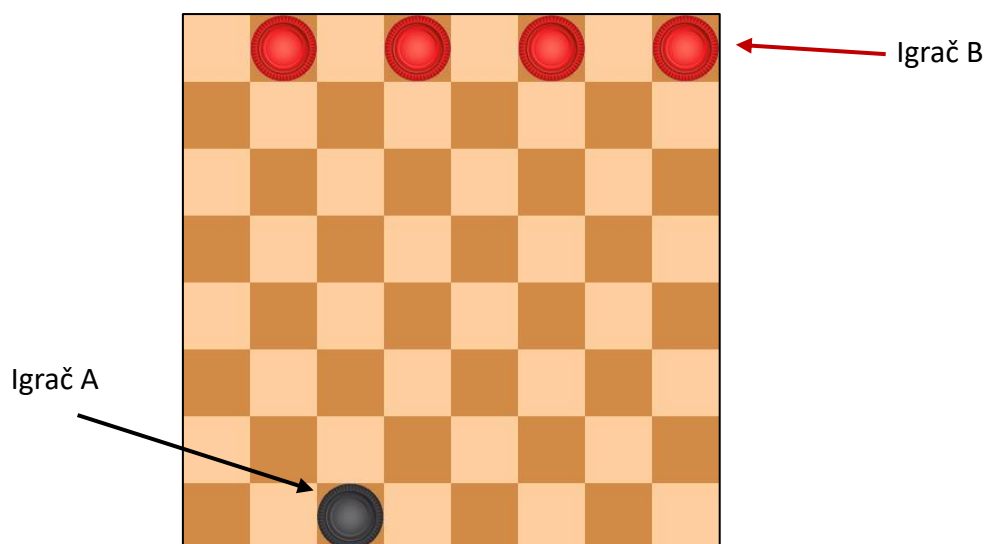
U zadatku nije nužno da koristite iste boje i dizajn sa slika koje su vam date u ovom tekstu zadatka. Bitno je da tabla sadrži 8x8 polja, da se na njoj smenjuju naizmenično dve različite vrste boja i da su figure na tabli jasno vidljive u odnosu na nju.

Početno stanje igre:

U igri učestvuju dva igrača, koji upravljaju svojim figurama (možete ih predstaviti proizvoljno, kao diskove, ili drugačije):

- **Igrač A – upravlja lisicom (fox).** Ovaj igrač upravlja samo jednom figurom koja predstavlja lisicu i ona se na početku igre nalazi na dnu table, na proizvoljnom polju, koje samo mora biti iste boje kao i polja na kojima se inicijalno nalaze figure igrača B.
- **Igrač B – upravlja guskama (geese).** Ovaj igrač upravlja sa 4 figure koje predstavljaju guske i one se na početku igre nalaze na vrhu table i sve 4 moraju biti na istoj boji (dakle, sve figure su ili na „tamnoj“ boji ili su sve figure na „svetloj“ boji – ovo možete sami odabrati da li da bude uvek isto ili proizvoljno).

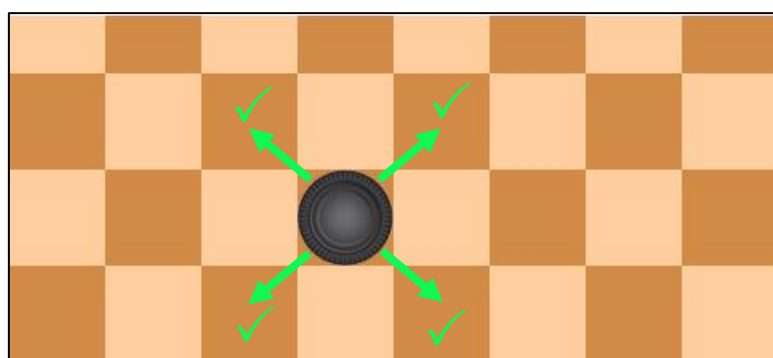
Na osnovu prethodnog, jedno, validno, početno stanje bi bilo sledeće:



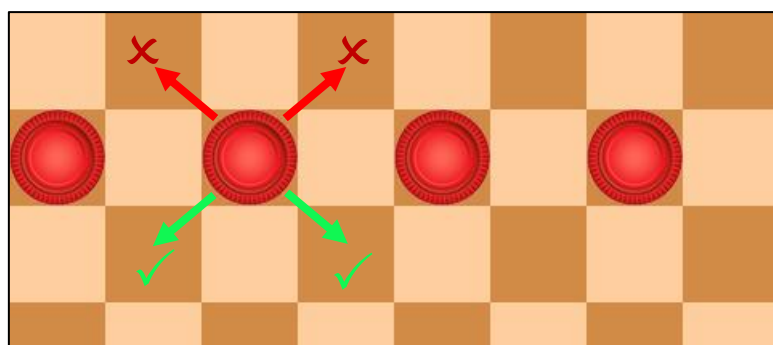
Pravila pomeranja figura:

Oba igrača pomeraju svoje figure **za jedno polje** i to samo **po dijagonalama**, odnosno na naredno polje koje je iste boje, sa bitnom razlikom:

- Igrač A (*fox*) sme da pomera svoju figuru u sva četiri smeru po dijagonalama:



- Igrač B (*geese*) sme da pomera samo jednu figuru po potezu i to samo prema dnu table (ne sme da se vraćaju figure prema vrhu table):



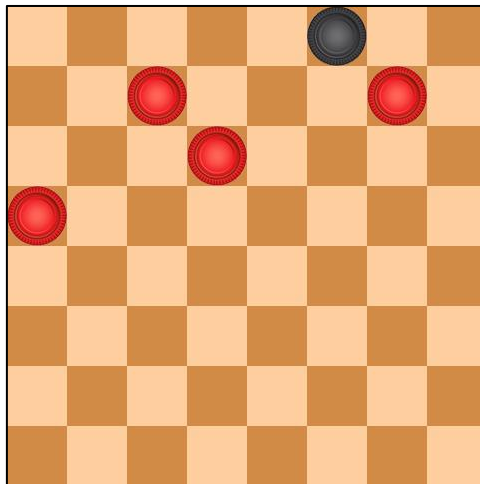
Nije dozvoljeno pomeranje figure na isto polje na kojem se već nalazi neka figura, bilo da je ona od istog igrača ili od protivničkog igrača.

Igru započinje igrač A (*fox*), nakon čega se igrači naizmenično smenjuju u potezima (dakle, sledi igrač B, pa ponovo igrač A, pa igrač B, itd...).

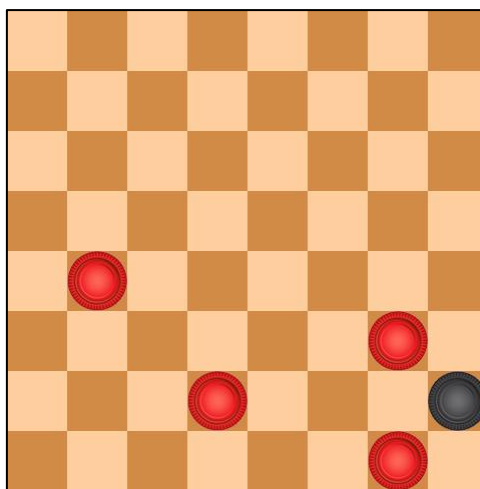
Cilj igre:

Cilj Igrača A (*fox*) je da dođe do vrha table, tj. do nekog od početnih polja igrača B. Cilj igrača B (*geese*) je da “zarobi” igrača A, tako što će ga pomeranjem svojih figura dovesti u poziciju da nema nijedno slobodno polje za pomeranje svoje figure.

Ukoliko igrač A dođe do vrha table, onda je on pobedio:



Ukoliko igrač B “zarobi” igrača A, onda je igrač B pobedio:



Ukoliko vam je lakše da razumete pravila igre putem video objašnjenja, možete pogledati kratak, jednominutni tutorijal na sledećem linku: <https://www.youtube.com/watch?v=6VjNwmGAyzo>

Opis zadatka (dva igrača na istom uređaju):

1. Android aplikacija se sastoji od jedne aktivnosti u kojoj odmah počinje igra: prikazana je tabla u kojoj su figure postavljene na inicijalne pozicije. Prvi igrač igra ulogu igrača A iz ranije opisa (tj. *fox*), a drugi igra ulogu igrača B iz ranijeg opisa (tj. *geese*);
2. Prvo igra igrač A, nakon toga igra drugi igrač i tako naizmenično sve dok jedan igrač ne pobedi;
3. Posle svakog poteza, proverava se da li je igra završena;
4. Po završetku igre, ispisuje se poruka o tome koji igrač je pobedio i pokreće se nova igra u kojoj će prvi igrati igrač koji nije bio prvi na početku trenutne igre.

Opis zadatka (preko mreže):

1. Server je zadužen da registruje igrače koji se prijave, a komunikacija dva igrača se odvija preko servera;
2. Sami odaberite kako ćete implementirati serversku Java aplikaciju – konzolna ili GUI aplikacija. Nije nužno, ali najpraktičnije bi bilo da se serverska aplikacija pokreće na identičan način kao u slučaju distribuirane „Kviz“ Java aplikacije;
3. Klijentska aplikacija je Android aplikacija koja se sastoji od dve aktivnosti:
 - a. Prva aktivnost omogućava korisniku da unese IP adresu servera i port, kao i da ostvari inicijalnu komunikaciju sa serverom (slanje korisničkog imena u cilju registracije), kao i odabir igrača sa kojim želi da igra nakon što od servera dobije spisak raspoloživih registrovanih igrača);
 - b. Druga aktivnost se pojavljuje kada korisnik odabere sa kim želi da igra igru i u njoj je realizovan sam tok igre: prikazana je tabla sa figurama na inicijalnim pozicijama i igrači dobijaju uloge koje će igrati (igrač koji je inicirao igru će biti *fox*, a drugi igrač će biti *geese*). Nakon svakog odigranog poteza, ažurira se stanje table tako što se odgovarajuća figura od igrača pojavi na poziciji koju je selektovao;
4. Kada se korisnik poveže sa serverom, šalje mu svoje korisničko ime i na taj način se registruje za igranje igre;
5. Nakon registrovanja novog korisnika na server, server svim korisnicima koji trenutno ne igraju šalje spisak svih korisnika koji su dostupni (registrovani a ne igraju trenutno);
6. Svaki korisnik (npr. X) može da odabere sa kime želi da igra igru (npr. sa Y), a server će taj upit proslediti korisniku Y i ako on prihvati - igra počinje, a korisnici X i Y su označeni kao nedostupni (registrovani ali zauzeti);
7. Za primer od gore, ako korisnik Y ne prihvati igru sa korisnikom X, korisnik X mora biti obavešten o tome i mora mu se omogućiti da, ako želi, ponovo bira sa kime želi da igra;
8. Kada igra počne, prvi odigrava korisnik koji ima ulogu igrača A (*fox*) iz ranijeg opisa, nakon toga igra drugi igrač i tako naizmenično sve dok jedan igrač ne pobjedi;
9. Svaki odigran potez se šalje serveru, a on je dalje prosleđuje drugom korisniku, čiji ekran se ažurira u skladu sa prethodno odigranim potezom protivnika;
10. Ako igrač koji je odigrao poslednji pokušaj da ponovo odigra pre nego što to uradi njegov protivnik, njegov pokušaj treba odbaciti i prikazati korisniku adekvatnu poruku da treba da sačeka da protivnik ubaci svoj disk;
11. Posle svakog poteza, proverava se da li je igra završena. Na kraju igre, ispisuje se poruka koji igrač je pobjedio, a igračima se ponudi opcija da ponovo igraju i samo ako obojica prihvate, igra kreće iz početka, a u njoj će prvi igrati igrač koji nije bio prvi na početku trenutne igre;
12. U slučaju da barem jedan od igrača ne želi da igra novu igru, server će oba igrača ponovo označiti kao dostupne i poslati svim dostupnim igračima ažurirani spisak dostupnih igrača nakon čega se mogu odigravati nove igre.

Napomene:

1. Slike za predstavljanje figura, kao i njihove boje, odaberite sami;
2. Sam izgled table ne mora da bude isti kao u ovom tekstu zadatka;
3. Sami odaberite layout koji ćete koristiti za predstavljanje sadržaja table (kombinaciju Linear layouta, Relative layout, Constraint);
4. Inicijalne pozicije figura igrača B (*geese*) možete staviti da uvek idu na istu boju (npr. "tamnu"), a poziciju igrača A (*fox*) postavite na **nasumično** polje **iste boje** u donjem redu;
5. Raspored komponenti na ekranu možete kreirati statički u okviru layout XML datoteke, a dozvoljeno je kreirati ih i dinamički obzirom na to da su pozicije na koje se postavljaju figure pravilno raspoređene;
6. Podrazumeva se da igrači koji se registruju imaju svi različita imena – možete zanemariti situaciju u kojoj se dva igrača sa istim imenom registruju za igranje igrice;
7. U slučaju da realizujete opciju igranja preko mreže možete istovremeno pokrenuti više emulatora u okviru Android Studio-a, te na taj način možete testirati svoju aplikaciju u slučaju da nemate Android telefon;
8. U slučaju da realizujete opciju igranja preko mreže, ukoliko imate Android telefon, možete i njega koristiti prilikom testiranja aplikacije (npr. jedan korisnik koristi Android emulator, a drugi igra preko telefona);

Ocenjivanje – oba igrača na istom uređaju (maksimalno 15 poena):

- Korektno iscrtavanje početnog stanja table i ponovnog iscrtavanja početnog stanja nakon što se završi igra – **5 poena**
- Implementirani svi koraci tokom odigravanja igre (naizmenično omogućavanje jednom pa drugom igraču da pomeraju svoje figure i adekvatno ažuriranje table) – **5 poena**
- Detekcija završetka igre, zamena igrača koji igra prvi u narednoj igri – **5 poena**

Ocenjivanje – realizacija preko mreže (maksimalno 30 poena)

- Serverska aplikacija sa više niti (slična kao što smo koristili kod server aplikacije za "Kviz") – **5 poena**
- Povezivanje sa serverom, registrovanje korisnika za igru i početak igre (odabir protivnika, prihvatanje igre, itd.) – **5 poena**
- Korektno iscrtavanje početnog stanja table i ponovnog iscrtavanja početnog stanja nakon što se završi igra i odabere ponovno igranje – **5 poena**
- Komunikacija sa serverom u pozadinskoj niti – **5 poena**
- Implementirani svi koraci tokom odigravanja igre (naizmenično omogućavanje jednom pa drugom igraču da pomera svoje figure i adekvatno ažuriranje table, kontrola koji igrač u kom trenutku može da odigra svoj potez) – **5 poena**
- Detekcija završetka igre, zamena igrača koji igra prvi u narednoj igri – **5 poena**