

OPŠTI PODACI O PROJEKTNOM TIMU				
Br grupe	Broj indeksa	Ime	Prezime	Email adresa
2	PR28/2022	Julijana	Ristić	julijanaristic223@gmail.com
2	PR25/2022	Katarina	Kalauz	kalauz.pr25.2022@uns.ac.rs
Github link				
https://github.com/julijanaristic/PRMuIS_projekat.git				

OPŠTI PODACI O PROJEKTU	
Naziv projekta:	Igre kviza TV Slagalice za dva igrača
TEHNIČKI OPIS PROJEKTA	
Sažetak:	Razvoj serverske aplikacije koja pruža klijentima mogućnost da odigraju tri igre kviza TV Slagalice: <i>Slagalicu</i> , <i>Skočko</i> i <i>Ko zna zna</i> . Dva klijenta se mogu prijaviti da igraju na jednom serveru. Poeni se čuvaju za svakog igrača i na kraju igre se prikazuje pobjednik sa osvojenim brojem poena. Jedan klijent se može prijaviti tako da odigra trening igru, gde ne mora igrati sve igre, već može da bira koje će igrati.
Primenjene metode:	<p>Multipleksiranje utičnica (socket multiplexing): Centralni server (igra) istovremeno upravlja sa dva klijenta (igrača), obezbeđujući efikasnu komunikaciju.</p> <p>UDP i TCP komunikacija:</p> <ul style="list-style-type: none"> • UDP se koristi za prijavu igrača (ako se prijavljuje samo jedan, šalje se i koje igre će igrati). • TCP se koristi slanje pitanja i odgovora tokom igre. <p>Interakcija i obrada unosa za svaku različitu igru: Zavisno od igre ponuditi adekvatnu interakciju (unos/izbor odgovora) i obradu unetih/izabranih vrednosti.</p> <p>Tok i stanje igre: praćenje trenutno aktivne igre, poena koji se osvajaju i koliko koji igrač ima osvojenih poena. Na kraju igre, ispisati rezultate oba igrača.</p>
Opis projekta:	<p>Potrebno je razviti serversku aplikaciju koja upravlja igrom i klijentske aplikacije koje korisnicima pružaju da igra igre kviza "TV Slagalice". Aplikacije poseduju <i>sledeće karakteristike</i>:</p> <p>Igre: Klijenti mogu igrati jednu od četiri igre:</p> <ul style="list-style-type: none"> • <i>Slagalice</i>: Unosi se najduža reč koja se sastavlja od ponuđenih slova • <i>Skočko</i>: Unosi se kombinacija četiri znaka (od mogućih šest) za koju se proverava da li se znaci nalaze na pravom mestu • <i>Ko zna zna</i>: Na osnovu prikazanog pitanja i ponuđenih odgovora, bira se tačan (unosom i slanjem broja/slova ispred tačnog odgovora) <p>Server koji upravlja igrom:</p> <ul style="list-style-type: none"> • <i>Reguliše prijavu igrača</i>: Ako se prijave dva igrača, priprema novu igru za obojicu. Ako se prijavi samo jedan, priprema igre samo za njega. • <i>Obrada odgovora i bodovanje</i>: na osnovu broja/tačnosti odgovora, dodeljuju se poeni svakom igraču. <p>Računanje ukupnog broja poena: osvojeni broj poena svakog igrača se prenosi u narednu igru.</p> <p>Prednost odgovaranja: Omogućiti da se prvi poslati odgovor prvi uzima u obzir prilikom računanja poena, ako je korektan i da igrač koji ga je poslao dobije poene. Ovo će se primenjivati u svim igrama osim igre "Skočko". Ako drugi igrač da korektan odgovor, dobija manje poena jer je bio sporiji.</p> <p>Primer upotrebe: Na svaki pokrenuti server mogu se povezati do dva klijenta i igraju, određenim redom, jednu po jednu igru, dobijaju poene za korektne odgovore. Na kraju igrač sa više osvojenih poena pobeđuje.</p>

Zadaci:			
1	Zadatak: Skica osnovne implementacije projekta	Rok: KT1	Broj poena: 2
	Opis: Napraviti blok-dijagram (šemu) koja prikazuje relacije između: <ul style="list-style-type: none"> • Kontrolora igre (servera), i • jednog klijenta (igrača). Dijagram treba da uključuje tokove podataka i komunikacione protokole (UDP za prijavu, TCP za praćenje toka kviza).		
2	Zadatak: Implementacija osnovnih funkcionalnosti servera - određivanje detalja o igri	Rok: KT1	Broj poena: 3
	Opis: Na serveru se otvara UDP utičnica koja služi za prijavu igrača. <p>Klijent se prijavljuje slanjem poruke u formatu "PRIJAVA: [ime/nadimak], [lista igara koje igrač želi da igra, odvojene zarezima]", gde ta lista igara može da sadrži konstante "sl", "sk" i/ili "kzz". Ako je klijent pravilno poslao poruku o prijavi, server mu odgovara slanjem informacija o njegovoj TCP utičnici (IP adresa i broj porta). Kada se uspostavi TCP konekcija, server šalje klijentu (igraču) poruku "Dobrodošli u trening igru kviza TV Slagalica, današnji takmičar je [uneto ime/nadimak]".</p> <p><i>Napomena:</i> reč trening se ne ispisuje u pozdravnoj poruci, ako učestvuje više igrača (KT2). U treningu, igrač može da traži da igra istu igru više puta.</p>		
3	Zadatak: Čuvanje podataka o igraču	Rok: KT1	Broj poena: 1
	Opis: Podaci o igraču se čuvaju u klasi Igrač . <p>Klasu Igrac čine polja:</p> <ul style="list-style-type: none"> • Identifikacioni broj igrača (tipa <i>int</i>): dodeljuje se prilikom prijave igrača • Ime/Nadimak igrača (tipa <i>string</i>) • <i>Niz</i> vrednosti tipa <i>int</i>: broj poena po svakoj igri. Niz ima broj elemenata u skladu sa time koliko igara je igrač odlučio da igra. 		
4	Zadatak: Početak igre za klijenta	Rok: KT1	Broj poena: 1
	Opis: Kada se uspostavi konekcija sa TCP utičnicom i klijent dobije poruku o početku igre, treba da pošalje poruku SPREMAN da bi započeo kviz.		
5	Zadatak: Igra "Slagalica"	Rok: KT1	Broj poena: 3
	Opis: Igra se implementira kao sopstvena klasa koja ima polja: <ul style="list-style-type: none"> • Ponuđena slova (tipa <i>string</i>): sadrži 12 slova koja se prikazuju igraču kako bi od njih sastavio reč • Sastavljena reč (tipa <i>string</i>) i metode: <ul style="list-style-type: none"> • Generiši slova: Popunjava string koji predstavlja ponuđena slova sa 12 nasumičnih slova. Treba da ima najviše četiri samoglasnika u ponudi. • Proveri reč: Proverava da li je sastavljena reč sačinjena od ponuđenih slova tako da se slova ne koriste više od broja ponavljanja. Ako je to tačno, dobija (5*broj slova u korektnoj reči) poena 		

	<p><i>Napomena:</i> Objekti ove klase se prave u okviru implementacije servera, a potrebne informacije za funkcionisanje igre se šalju klijentima. Klijent prosleđuje svoju reč serveru, a ona se proverava unutar funkcije za proveru reči.</p>		
6	Zadatak: Igra "Skočko"	Rok: KT2	Broj poena: 4
	<p>Opis: Igra se implementira kao sopstvena klasa koja ima polja:</p> <ul style="list-style-type: none"> • Tražena kombinacija (tipa <i>string</i>): sadrži 4 slova koja predstavljaju kombinaciju koja se traži (H-herc, T-tref, P-pik, K-karo, S-skočko, Z-zvezda) • Tekuća kombinacija (tipa <i>string</i>) <p>i metode:</p> <ul style="list-style-type: none"> • Generiši kombinaciju: Popunjava string koji predstavlja traženu kombinaciju sa 4 nasumična slova. Mora postojati barem dva različita slova u kombinaciji. • Proveri kombinaciju: Proverava da li tekuća kombinacija odgovara traženoj, a ako ne, vraća poruku o tome koliko znakova je na pravom mestu, koliko ih je deo kombinacije ali nisu na pravom mestu, kao i koliko njih nije deo kombinacije. <p><i>Primer:</i> Na ovakvu poruku: HZTP Server odgovara sa: dva znaka su na mestu i jedan nije na mestu.</p> <p><i>Napomena:</i> Objekti ove klase se prave u okviru implementacije servera, a potrebne informacije za funkcionisanje igre se šalju klijentima. Klijent prosleđuje jedan po jedan pokušaj kombinacije serveru, a on se proverava unutar funkcije za proveru kombinacije.</p> <p>Zavisno od broja pokušaja pri pogađanju kombinacije, igrač dobija 30 poena za pogodak na prvom pokušaju, 25 na drugom, 20 na trećem, 15 na četvrtom, 10 na petom i šestom.</p>		
7	Zadatak: Igra "Ko zna zna"	Rok: KT2	Broj poena: 3
	<p>Opis: Igra se implementira kao sopstvena klasa koja ima polja:</p> <ul style="list-style-type: none"> • Tekuće pitanje (tipa <i>string</i>): tekst pitanja koje se trenutno prikazuje sa ponuđenim odgovorima • Tačan odgovor (tipa <i>int</i>): broj opcije koja predstavlja tačan odgovor • Sva pitanja (Dictionary<string,int>): predstavlja skup svih pitanja i njima uparene tačne odgovore. <p>i metode:</p> <ul style="list-style-type: none"> • Učitavanje pitanja: popunjava rečnik svih pitanja vrednostima - učitavaju se iz tekstualne datoteke. • Proveri tačan odgovor: Proverava da li je igrač poslao tačan odgovor na traženo pitanje. <p><i>Primer:</i> Koje godine je pao Berlinski zid? 1 - 1961 2 - 1989 3 - kad god da je objavljen film "Atomic Blonde"</p> <p>Klijent unosi i šalje broj koji odgovara opciji koja je tačan odgovor</p> <p><i>Napomena:</i> Objekti ove klase se prave u okviru implementacije servera, a potrebne informacije za funkcionisanje igre se šalju klijentima. Klijent prosleđuje po jedan odgovor na svako od pet pitanja nazad serveru, gde se proverava tačan odgovor i klijent dobija poene. Za svaki tačan odgovor igrač dobija 10 poena. Za pogrešan, -5 poena.</p>		
8	Zadatak: Istovremeni, neblokirajući rad sa više klijenata	Rok: KT2	Broj poena: 3
	<p>Opis: Omogućiti polling model nad TCP utičnicama koje server koristi za komunikaciju sa više igrača. Ažurirati postojeće rešenje, tako da radi sa više od jednog klijenta i da čuva informacije za svakog od njih.</p>		

	Napomena: Ispisivati broj poena nakon svake igre na serveru za svakog od igrača.		
9	Zadatak: Bodovanje i određivanje pobjednika	Rok: KT2	Broj poena: 3
	<p>Opis: Ako je određeni igrač uspeo da pogodi i prvi da pošalje odgovor dobija ukupan broj poena koji se daje za pitanje. Igrač koji kasni sa odgovorom (po redosledu davanja odgovora, dobija po 15% manje poena). Ovo ne važi za igru "Skočko". Za negativne poene je uvek fiksna vrednost. Na kraju, potrebno je izračunati ukupan broj poena svakog igrača i ispisati konačnu tabelu poena na serveru, dok se klijentima šalje ko je pobjednik/gubitnik. Ako dva igrača imaju isti broj poena, pobjednik je onaj ko je ostvario više poena u igri "Skočko".</p>		
10	Zadatak: Skica ukupne implementacije projekta	Rok: KT2	Broj poena: 2
	<p>Opis: Ažurirati blok-dijagram (šemu) dodavanjem:</p> <ul style="list-style-type: none"> • Više igrača koji simultano šalju odgovore. • Način na koji server obrađuje sve zahteve. 		