**UNIVERZITET U BEOGRADU**

**FAKULTET ORGANIZACIONIH NAUKA**

**SEMINARSKI RAD**

Tema: Igra za ucenje brajevog pisma

Katedra za elektronsko poslovanje

Predmet: Mobilno poslovanje

Student:

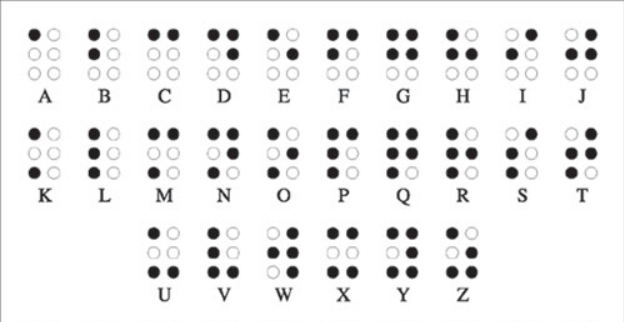
Milan Dedjanski 0193/2018

Beograd, februar 2022

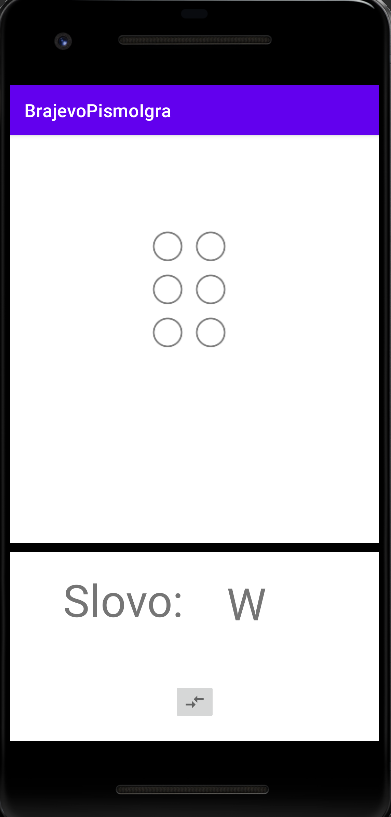
Brajevo pismo:

Brajevo pismo je reljefno pismo za slepe. **Brajeva azbuka se sastoji od šest tačaka**, koje su raspoređene u dve kolone po tri tačke. Kombinacijom ispupčenih tačaka, kojih ukupno ima 63, dobijaju se pojedina slova. Znaci interpunkcije, kao što su zapete (zarezi), tačke, znakovi pitanja, usklika (uzvičnika) i slično, raspoređeni su u dve kolone po dve tačke. Brajevi znakovi za brojeve i matematičke funkcije doživele su nekoliko modifikacija.

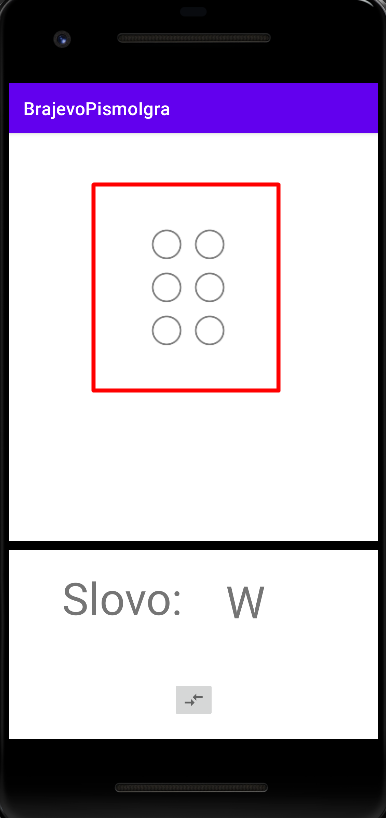
**Slepi dodirom, jabučicom desnog kažiprsta, čitaju sleva nadesno, a pišu zdesna nalevo.** Jagodica prsta može osetiti celu jednu ćeliju, jedno slovo od jednom.



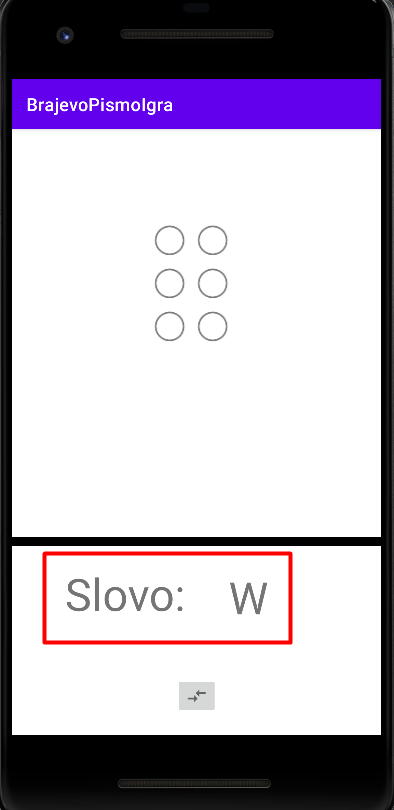
Izgled I funkcionalnosti:



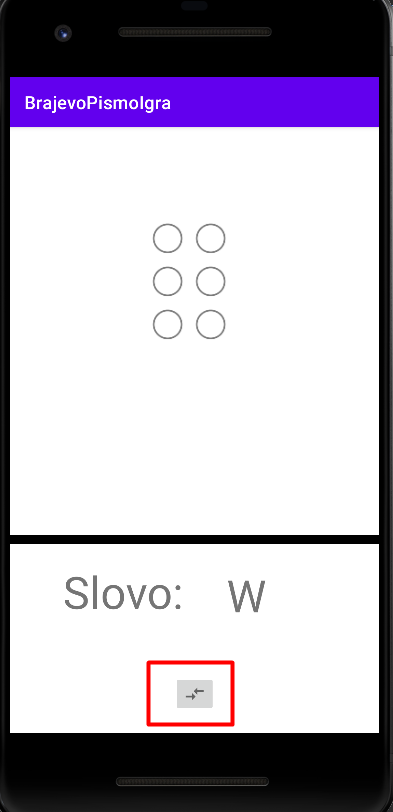
Odmah nakon pokretanja aplikacije ucitava se ovaj ekran.



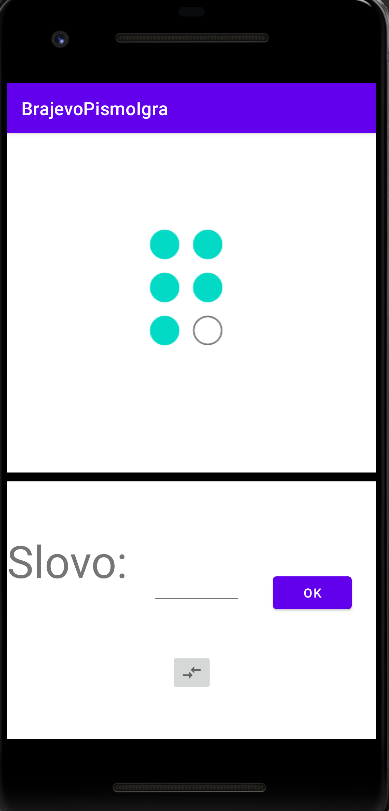
U gorenjem delu nalaze se “tacke” koje je neophodno popuniti na odgovarajuci nacin kako bi se nastavilo sa igrom.



Ispod njih se nalazi nasumicno generisano slovo. Na osnovu ovog slova se I odredjuje ispravna kombinacija “tacaka”.



Na samom dnu nalazi se dugme koje okrece igru otvarajuci novu stranu.



Nova strana prilikom ucitavanja popunjava “tacke” na nasumican nacin. Potrebno je da Korisnik unese ispravnu slovnu vrednost tacaka kako bi se igra nastavila.

Kod:

public class Parovi {  
 private char slovo;  
 private boolean cb1;  
 private boolean cb2;  
 private boolean cb3;  
 private boolean cb4;  
 private boolean cb5;  
 private boolean cb6;  
  
  
 public Parovi(char slovo, boolean cb1, boolean cb2, boolean cb3, boolean cb4, boolean cb5, boolean cb6) {  
 this.slovo = slovo;  
 this.cb1 = cb1;  
 this.cb2 = cb2;  
 this.cb3 = cb3;  
 this.cb4 = cb4;  
 this.cb5 = cb5;  
 this.cb6 = cb6;  
 }

Za pocetak je kreirana nova klasa koja ce predstavljati kombinacije slova I tacaka. Iz tog razloga sadrzi jedan char za slovo I sest Boolean vrednosti koje predstavljaju vrednosti checkboxa ( true – stikliran, false – nije stikliran ). Naravno, tu se nalazi konstruktor za inicijalizaciju svih vrednosti.

Pored prikazanog koda, u okviru klase su definisani I geteri I seteri za sve attribute kao I override-ovanu metodu equals.

List<Character> slova = Arrays.*asList*('A','B','C','D','E','F','G','H','I','J','K','L','M','N'  
 ,'O','P','Q','R','S','T','U','V','W','X','Y','Z');  
  
// true su popunjene tackice. Posmatrao si sa leva na desno!  
// 1 2  
// 3 4  
// 5 6  
List<Parovi> parovi = Arrays.*asList*(new Parovi('A',true,false,false,false,false,false),  
 new Parovi('B',true,false,true,false,false,false),  
 new Parovi('C',true,true,false,false,false,false),  
 new Parovi('D',true,true,false,true,false,false),  
 new Parovi('E',true,false,false,true,false,false),  
 new Parovi('F',true,true,true,false,false,false),  
 new Parovi('G',true,true,true,true,false,false),  
 new Parovi('H',true,false,true,true,false,false),  
 new Parovi('I',false,true,true,false,false,false),  
 new Parovi('J',false,true,true,true,false,false),  
 new Parovi('K',true,false,false,false,true,false),  
 new Parovi('L',true,false,true,false,true,false),  
 new Parovi('M',true,true,false,false,true,false),  
 new Parovi('N',true,true,false,true,true,false),  
 new Parovi('O',true,false,false,true,true,false),  
 new Parovi('P',true,true,true,false,true,false),  
 new Parovi('Q',true,true,true,true,true,false),  
 new Parovi('R',true,false,true,true,true,false),  
 new Parovi('S',false,true,true,false,true,false),  
 new Parovi('T',false,true,true,true,true,false),  
 new Parovi('U',true,false,false,false,true,true),  
 new Parovi('V',true,false,true,false,true,true),  
 new Parovi('W',false,true,true,true,false,true),  
 new Parovi('X',true,true,false,false,true,true),  
 new Parovi('Y',true,true,false,true,true,true),  
 new Parovi('Z',true,false,false,true,true,true));

Kod iza MainActivity-ja pocinje kreiranjem liste Parova ( klasa koja je prethodno objasnjena ). Bilo je neophodno rucno uneti vrednosti u ovu listu. Takodje, kreirana je lista slova. Iz nje ce se nasumicnim putem birati slovo.

Inicijalizaciju elementa sa grafickog interfejsa, zbog svoje jednostavnosti, necu prikazivati.

private String randomSlovo(){  
 return slova.get(new Random().nextInt(slova.size())).toString();  
}

Funkcija koja iz liste slova vraca jedno nasumicno.

void slusajPromene(){  
 cb1.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 logika();  
 }  
 });  
 cb2.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 logika();  
 }  
 });  
 cb3.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 logika();  
 }  
 });  
 cb4.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 logika();  
 }  
 });  
 cb5.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 logika();  
 }  
 });  
 cb6.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 logika();  
 }  
 });  
  
}

Funkcija slusajPromene() je zaduzena za pracenje promena stanja “tacaka”. Ona se poziva u okviru OnCreate(). U slucaju da detektira promenu poziva funkciju logika().

void logika(){  
 boolean b1 = false;  
 boolean b2= false;  
 boolean b3= false;  
 boolean b4= false;  
 boolean b5= false;  
 boolean b6= false;  
  
 if(cb1.isChecked()) b1=true;  
 if(cb2.isChecked()) b2=true;  
 if(cb3.isChecked()) b3=true;  
 if(cb4.isChecked()) b4=true;  
 if(cb5.isChecked()) b5=true;  
 if(cb6.isChecked()) b6=true;  
  
 Parovi p = new Parovi(slovo.getText().charAt(0),b1,b2,b3,b4,b5,b6);  
  
 for(int i =0;i<parovi.size();i++){  
 if(p.equals(parovi.get(i))){  
 restartujIgru();  
 }  
 }  
}

Metoda logika kreira nov objekat p koj dobija vrednosti trenutno stikliranih tacaka. Proverava da li se ovaj novi objekat nalazi u listi parova, I ukoliko se nalazi poziva metodu restartujIgru().

void restartujIgru(){  
 cb1.setChecked(false);  
 cb2.setChecked(false);  
 cb3.setChecked(false);  
 cb4.setChecked(false);  
 cb5.setChecked(false);  
 cb6.setChecked(false);  
  
 slovo.setText(randomSlovo());  
}

Ova metoda jednostavno postavlja vrednosti na pocetne( false ) I ucitava novo slovo.

public void next(View view) {  
 Intent i = new Intent(this,MainActivity3.class);  
 startActivity(i);  
}

Konacno, metoda next se poziva pritiskom na dugme na dnu ekrana. Ona otvara novu stranu.

private String ucitajTacke() {  
 Parovi par = parovi.get(new Random().nextInt(parovi.size()));  
 boolean c1 = par.isCb1();  
 boolean c2 = par.isCb2();  
 boolean c3 = par.isCb3();  
 boolean c4 = par.isCb4();  
 boolean c5 = par.isCb5();  
 boolean c6 = par.isCb6();  
  
 cb1.setChecked(c1);  
 cb2.setChecked(c2);  
 cb3.setChecked(c3);  
 cb4.setChecked(c4);  
 cb5.setChecked(c5);  
 cb6.setChecked(c6);  
  
 String a = par.getSlovo() + "";  
 return a;  
}

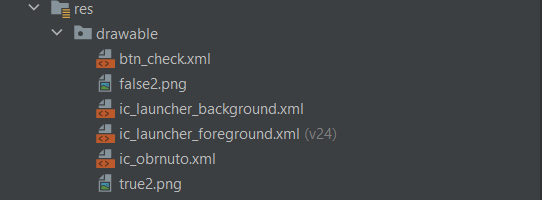
Prva metoda koja se poziva prilikom otvaranja strane ( OnCreate() ). Postavlja vrednost CheckBox-ova na kombinaciju koja postoji u listi parova. Takodje, vraca vrednost prvog atributa objekta, odnosno slovo ciju je kombinaciju izabrao.

public void uporedi(View view) {  
 String uneto = String.*valueOf*(slovo.getText());  
 if(uneto.toUpperCase().equals(unetoSlovo)){  
 restartujIgru();  
 }  
}

Ova metoda se poziva pritiskom na dugme “OK”. Uporedjuje da li je slovo koje je uneo Korisnik isto kao I slovo za koje je izbacena kombinacija tacaka. Ukoliko jeste poziva metodu restartujIgru().

private void restartujIgru() {  
 unetoSlovo = ucitajTacke();  
 slovo.setText("");  
}

Ova metoda iznova postavlja vrednosti tacaka I brise tekst koji je Korisnik uneo.



Finalno, u drawable folderu se nalaze koriscene slike. Btn\_check.xml zajedno sa true2 I false2 su korisceni za postizanje ovog izgleda checkboxa. Ic\_obrnuto je koriscena za dugme koje otvara novu stranu.