TEHNIČKA ŠKOLA NIKOLE TESLE

VUKOVAR

ZAVRŠNI RAD

PREDMET: NAPREDNO I OBJEKTNO PROGRAMIRANJE

MENTOR: MIROSLAV KARAN, dipl. inž. el.

TEMA: IZRADA LOGIČKE IGRICE KAO WINDOWS APLIKACIJE

UČENIK: BOJAN PUVAČA

ZANIMANJE: TEHNIČAR ZA RAČUNALSTVO

RAZRED: $4._2$ TR

DATUM OBRANE:_____

OCJENA:_____

OCJENA ZAVRŠNOG RADA:_____

KONAČNA OCJENA:_____

Komisija:_____

_____

_____

LJETNI ISPITNI ROK, šk. god. 2020./2021.

Sadržaj

1 UVOD 3
2 IGRA POGAĐANJA REČI 4
2.1 OPIS I PRAVILA IGRE 4
3 IZRADA IGRE 5
3.1 INTERFEJS 5
3.2 OPIS RAZVOJA PROGRAMA 6
3.3 OBJAŠNJENJE KODA 8
4 IZGLED ZAVRŠENOG PROGRAMA 19
5 ZAKLJUČAK 20
6 PRILOG 21
7 LITERATURA 22

1 UVOD

Rad se odnosi na igru pogađanja reči. Tačnije u radu je realizovana računarska igra napisana u programskom jeziku C# pomoću alata MS *Visual Studio IDE*. Rad će se sastojati od tekstualnih i grafičkih informacija kao i od delova koda uz objašnjenja.

Program je zamišljen kao minimalistički program pogađanja reči jedan na jedan tj. igra je namenjena za jednog igrača koji igra protiv računara. U radu će biti objašnjen proces osmišljavanja programa, postupak pred razvoj i sam razvoj koda.

Za razvoj programa će se koristiti jezik C# (en. izgovor C Sharp). C# spada među mlađe programske jezike, nastao 2000. godine kao sastavni deo Microsoftovog razvojnog okruženja .NET Framework. Jezik C# nema ograničenja kada je u pitanju tip aplikacije koja se pravi. Razvojno okruženje koje će se koristiti je Microsoft Visual Studio. Podržava 36 različitih programskih jezika, među kojima se nalazi i C#. Rad će biti urađen koristeći Windows Froms razvojnu platformu.

Ovu temu sam odabrao iz više razloga. Smatram da je tema na kojoj bih mogao najviše isprobati svoje znanje programiranja, kao i steći nova. Iako je na izgled jednostavan zadatak, sastoji se od mnogih kompleksnih delova čija je izrada predstavljala izazov koji sam bio spreman prihvatiti.

2 IGRA POGAĐANJA REČI

2.1 OPIS I PRAVILA IGRE

Ova igra, pogađanja reči, namenjena je za dva ili više igrača. Jedan igrač zamišlja reč, frazu ili rečenicu, a drugi ih pokušava pogoditi predlažući slova unutar određenog broja nagađanja. Reči koje se pogađaju predstavljene su redom crtica, svaka crtica predstavlja jedno slovo reči.

U većini verzija ove igre, vlastite imenice kao što su vlastita imena, mesta i brendovi su zabranjeni. Slengovi, tj. neformalne ili skraćene reči također nisu dopuštene za pogađanje.

Ako igrač koji pogađa predloži slovo koje se nalazi u reči, drugi igrač piše to slovo na sva odgovarajuća mesta.

U slučaju da se predloženo slovo ne nalazi u reči, drugi igrač crta jedan deo tela karikiranog čovečuljka.

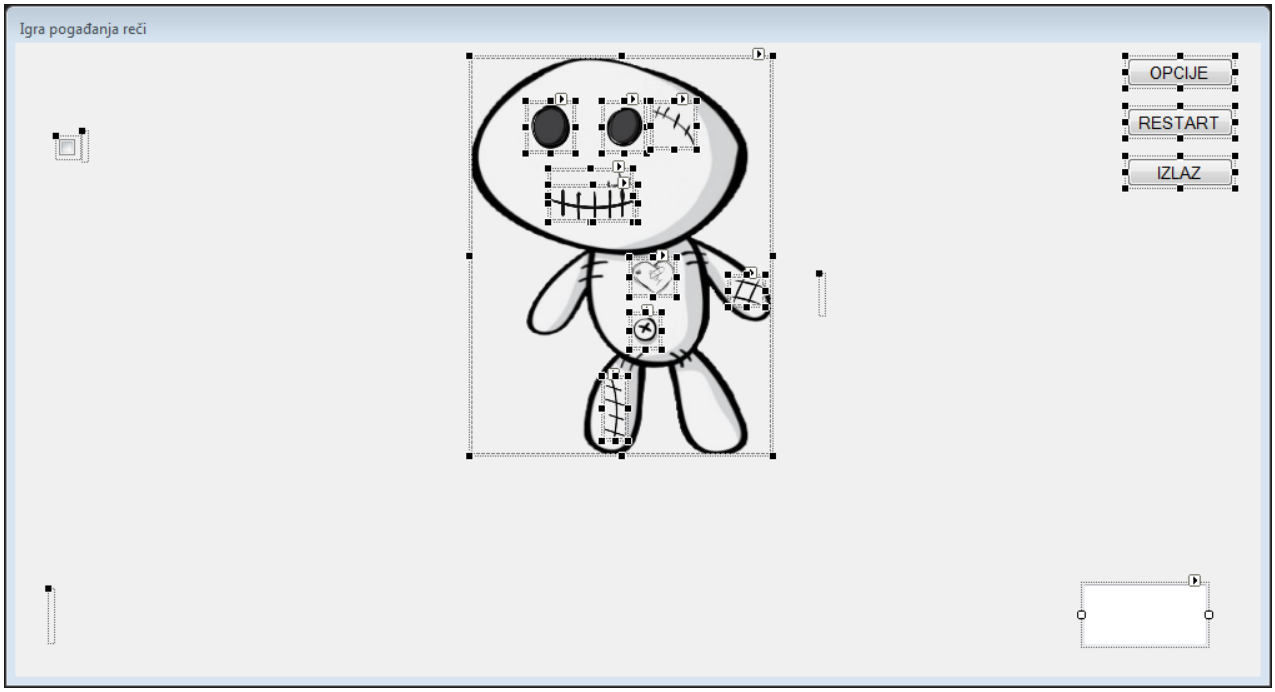
Igrač koji pogađa reč može u bilo kojem trenutku pokušati pogoditi celu reč. Ako je reč pogođena igra je gotova i igrač koji pogađa je pobedio.

S druge strane, ako igrač napravi dovoljno netačnih nagađanja kako bi omogućio protivniku da dovrši telo čovečuljka, igra je također gotova, ovoga puta igrač koji pogađa gubi.

3 IZRADA IGRE

3.1 INTERFEJS

Interfejs programa je zamišljen kao minimalistički, bez preterane količine informacija jednostavan za korišćenje. Program će biti urađen kao forma. Formu možemo zamisliti kao komad papira ili školsku tablu, sa posebnim delovima za određene podatke. Jedan deo zauzima prostor za unos slova koje pogađamo. Drugi deo služi za prikazivanje teksta pobede “POBEDA”, poraza “KRAJ IGRE”, a takođe se koristi u slučaju da igrač ponovo unese već korišteno slovo “Slovo je već bilo u upotrebi!”. Treći deo rezervisan je za crtice koje će se generisati u zavisnosti od broja slova koje reč sadrži. Četvrti deo odvojen je za tastere restarta tj. ponovnog pokretanja igre, izlaza iz aplikacije i opcija koje otvara novu formu u kojoj možemo dodati nove ili pregledati trenutne reči. Peti deo služi za ispis svih prethodno iskorišćenih slova. Šesti deo odvojen je za crtanje delova lutke. Sedmi i zadnji deo služi kako bi igrač kad to poželi proverio koja reč mu je bila zadata.



Slika 3.1.1 Izgled interfejsa igre

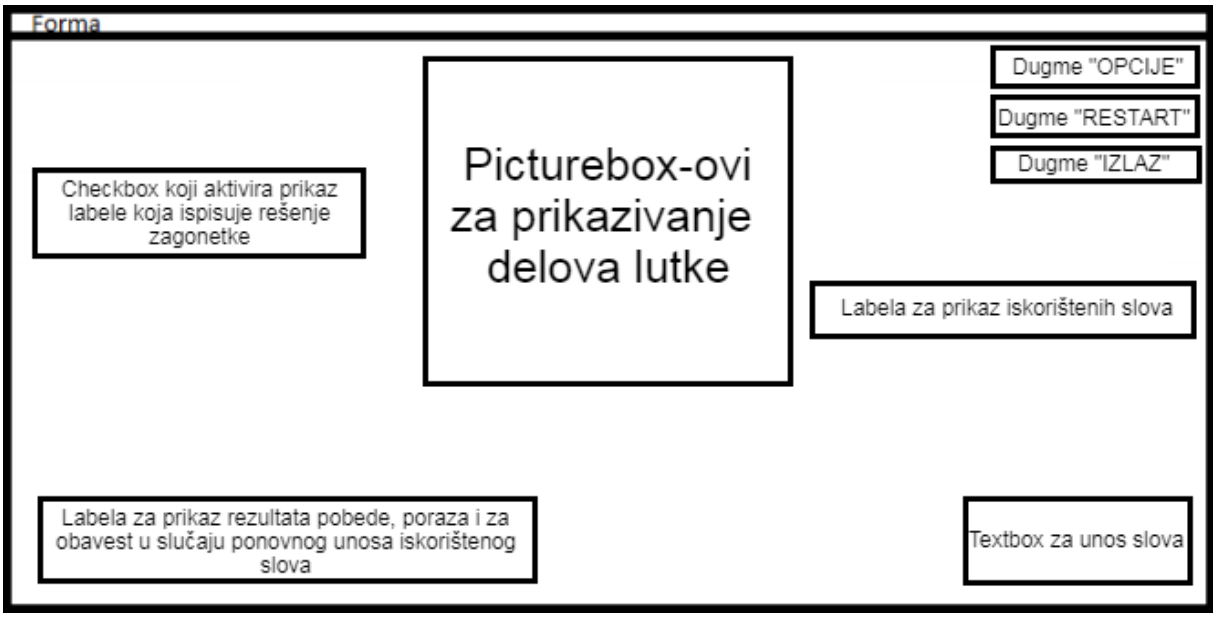
3.2 OPIS RAZVOJA PROGRAMA

Prvi korak pri izradi programa je započinjanje novog projekta u Visual Studiu. Projekat je Windows forma u jeziku C#. Sledeći korak je bilo dodavanje potrebnih objekata i uređivanje forme. Na formi je promenjen atribut Text na “Igra pogađanja reči” kao i atribut Icon iz stilskih razloga, ControlBox postavljen je na false, time uklanjamo dugmiće za minimiziranje, maksimiziranje i zatvaranje prozora.

Direktno u dizajn prozora su ubačeni sledeći objekti:

- *Textbox* za unos slova koje pogađamo
- *Labela* za prikaz rezultata pobede ili poraza, a takođe i za obavest u slučaju ponovnog unosa iskorišćenog slova
- *Labela* za prikaz iskorišćenih slova
- *Dugme* za opcije upravljanja rečima
- *Dugme* za ponovno pokretanje igre tj. “restart”

- *Dugme* za izlaz iz aplikacije
- *Picturebox*-ovi za prikazivanje delova lutke
- *Checkbox* koji aktivira prikaz labele koja ispisuje rešenje zagonetke

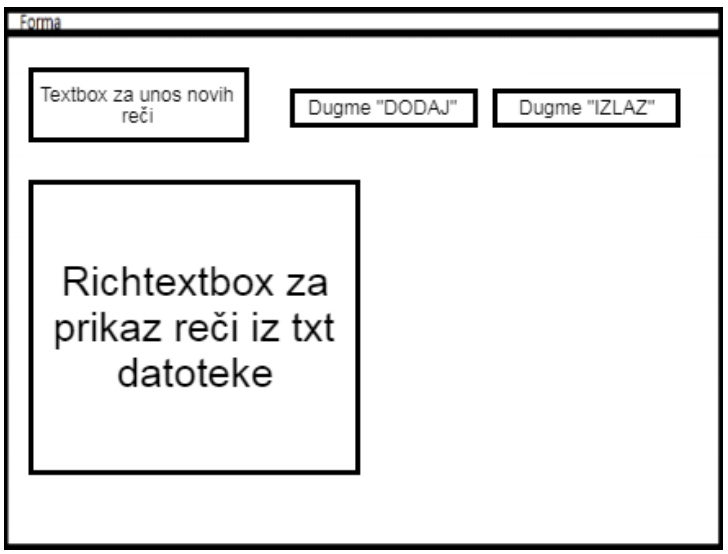


Slika 3.2.1 Šematski prikaz dizajna forme igre

Klikom na opcije otvara se novi form koji sadrži:

- *Textbox* za unos novih reči
- *Dugme* koje će uneti novu reč u txt datoteku
- *Richtextbox* za prikaz reči iz txt datoteke
- *Dugme* koje će zatvoriti prozor opcija

Slika 3.2.2 Šematski prikaz dizajna forme opcije



3.3 OBJAŠNJENJE KODA

Nakon dodavanja svih potrebnih objekata u dizajn prozoru, bilo je potrebno napisati kôd koji bi im dao svrhu i na osnovu kojeg bi igra funkcionisala. Kod je osmišljen da bude modularan i jednostavan za izmenjivanje. U slučaju da se javi potreba za promenom u toku proizvodnje, bilo bi jednostavno tu promenu napraviti bez velikih izmena na celokupnom programu.

Na početku je bilo potrebno deklarisati određene varijable i koje su neophodne za razvoj programa.

```
StreamReader srd = new StreamReader("words.txt");
List<string> ucitaneReci = new List<string>();
List<Igrica> iskoristenoSlovo = new List<Igrica>();
SoundPlayer zvukDobijenaIgra = new SoundPlayer(@"pobeda.wav");
SoundPlayer zvukIzgubljenaIgra = new SoundPlayer(@"izgubljena.wav");
SoundPlayer zvukPogodjenoSlovo = new SoundPlayer(@"pogodjena.wav");
SoundPlayer zvukPromasenoSlovo = new SoundPlayer(@"greska.wav");
Random randBr = new Random();
PictureBox[] slike = new PictureBox[10];
```

Svrha sledećih varijabli i objekata:

- Varijabla *srd* predstavlja objekat klase *StreamReader* koji će čitati reči iz datoteke *words.txt*
- Objekat *ucitaneReci* je lista stringova koja predstavlja listu reči učitanu iz dokumenta *words.txt*.
- Lista *iskoristenoSlovo* je lista čiji će elementi biti objekti klase *Igrica*, a služi da bi se u nju upisivala sva prethodno iskoristena slova.
- Varijable *zvukDobijenaIgra*, *zvukIzgubljenaIgra*, *zvukPogodjenoSlovo*, *zvukPromasenoSlovo* predstavljaju objekte

klase *SoundPlayer* koji će reproducirati zvukove u sledećim situacijama:

1. *zvukDobijenaIgra* – reproduciraće zvuk pobeda.wav kada igrač koji pogađa reč popuni sva prazna mesta tražene reči.
2. *zvukIzgubljenaIgra* – reproduciraće zvuk izgubljena.wav kada igrač koji pogađa reč napravi devet grešaka.
3. *zvukPogodjenoSlovo*– reproduciraće zvuk pogodjena.wav kada igrač koji pogađa reč pogodi slovo koje se nalazi na jednom ili više mesta u reči.
4. *zvukPromasenoSlovo*– reproduciraće zvuk greska.wav kada igrač koji pogađa reč predloži slovo koje se ne nalazi u reči.

Ovi zvukovi uzeti su iz igrice Super Mario Bros.

- Varijabla *randBr* predstavlja objekat klase *Random* koji služi za generisanje nasumičnih brojeva tokom rada programa. „Seed“, početna varijabla koja određuje niz brojeva koje generator izbacuje, koji generator koristi je vezan za vrijeme kada je program pokrenut.

Varijabla *slike* je niz od najviše 10 objekta *PictureBox* klase koji služi za grafički prikaz delova crteža lutke.

```
while ((jednaRec = srd.ReadLine()) != null){
      jednaRec = jednaRec.ToUpper();
      if (jednaRec.Length > 0)
      {
      reciIzTxt += jednaRec + "\n";
      ucitaneReci.Add(jednaRec);
      }
}
srd.Close();
int brRijeci = ucitaneReci.Count();

izabranaRec = randBr.Next(0, brRijeci);
lblResenja.Text = Convert.ToString(izabranaRec);
brojKaraktera = ucitaneReci[izabranaRec].Length -

brojRazmakaIzmedjuReci();
```

- *While* petljom čitamo sadržaj txt datoteke liniju po liniju koristeći *ReadLine* metodu *StreamReader* klase. Svaka linija pohranjena je u string *jednaRec*. U slučaju da je u txt datoteci reč napisana malim slovima metoda *ToUpper*() će prebaciti sva mala slova u velika. If postavlja uslov tako da će u listu *ucitaneReci* biti unet sav sadržaj veći od 0 tj. na ovaj način sprečavamo unos praznog stringa iz txt datoteke u listu. Kada pročita i zadnji znak u datoteci zatvoriće *StreamReader*.
- Deklarišemo varijablu *brRijeci* u koju ćemo smestiti ukupan broj elemenata liste pomoću metode *Count*().
- Varijabla *izabranaRec* koja je tipa int, nasumično će odabrati vrednost indexa neke reči pomoću metode *Next*.
- Text *lblResenja* pridružujemo vrednost stringa indexa izabrane reči koji prebacujemo u string.

Varijabli *brojKaraktera* biće pridružena duljina stringa izabrane reči koju će nam vratiti svojstvo *Leght*. Pri tom u slučaju da pogađamo frazu tj. da imamo minimalno dve reči u jedom stringu, duljina stringa oduzeće se sa metodom *brojRazmakaIzmedjuReci*.

```
private int brojRazmakaIzmedjuReci()
        {
            string a = ucitaneReci[izabranaRec];
            int brRazmaka = 0;
            for (int i = 0; i < a.Length; i++)
            {
                if (a[i] == ' ')
                    brRazmaka++;
            }
            return brRazmaka;
        }
```

Ova metoda će pomoću for petlje proći kroz string izabrane reči tražeći space tj. prazan znak između reči. Na kraju će vratiti broj razmaka među rečima.

```
slike[0] = pictureBox1;
slike[1] = pictureBox2;
slike[2] = pictureBox3;
slike[3] = pictureBox4;
slike[4] = pictureBox5;
slike[5] = pictureBox6;
slike[6] = pictureBox7;
slike[7] = pictureBox8;
slike[8] = pictureBox9;
slike[9] = pictureBox10;

sakrijSlike();
crtajLinije();
ispisSlova();
```

- Svakom objektu iz niza slike pridružujemo pictureBox
- Na kraju pozivamo metode

```
private void sakrijSlike()
        {
            for (int i = 0; i < 10; i++)
            {
                slike[i].Visible = false;
            }
        }
```

Ova metoda će pomoću for petlje postaviti vrednost svih slika na false tj. slike neće biti prikazane u Formu kada se aplikacija otvori.

```
private void crtajLinije()
        {
            Label[] labels = new
Label[ucitaneReci[izabranaRec].Length];
            string s = ucitaneReci[izabranaRec];
            lblResenja.Text = s;
            for (int i = 0; i < s.Length; i++)
            {
                labels[i] = new Label();
                if (s[i] != ' ')
                    labels[i].Text = "_";
                else
                    labels[i].Text = " ";
                labels[i].Font = new Font("Arial", 30);
                labels[i].AutoSize = true;
                labels[i].Top = topPozCrte;
```

```
                labels[i].Left = leftPozCrte;
                labels[i].Height = 1;
                this.Controls.Add(labels[i]);
                leftPozCrte += 50;
            }
        }
```

Sledeća metoda *crtajLinije* napraviće potreban broj labela koje će prikazati kao crtice ili prazno polje između više reči ako se radi o frazi. Varijabla labels je niz objekta klase Label u koju će biti pridružen broj karaktera pomoću metode *Length* reči koje pogađamo. Kroz for petlju proveramo svaki karakter. Ako je prazno polje labela će biti prazna tj. nama nevidljiva, inače ispisaće crticu.

```
private void ispisSlova()
        {
            slova = new Label[ucitaneReci[izabranaRec].Length];
            string s = ucitaneReci[izabranaRec];

            for (int i = 0; i < ucitaneReci[izabranaRec].Length;
i++)
            {
                if (s[i] != ' ')
                {
                    slova[i] = new Label();
                    slova[i].Font = new Font("Arial", 20);
                    slova[i].AutoSize = true;
                    slova[i].Top = topPozSlova;
                    slova[i].Left = leftPozSlova;
                    this.Controls.Add(slova[i]);
                }
                leftPozSlova += 50;
            }
        }
```

Sledeća metoda *ispisSlova* slična je prethodnoj. Ovoga puta if-om proveravamo da li je neki znak u reči prazan i ako je kreiramo labelu koja će biti pozicionirana iznad crtice koje kreiraćemo u prethodno objašnjenoj metodi.

```
private void povecajSlovo(object sender, KeyEventArgs e)
        {
            lblMessidzBox.BackColor = Color.Gray;
            lblMessidzBox.Text = "";
            textBox1.Text = textBox1.Text.ToUpper();
            Igrica a = new Igrica();
            a.x = textBox1.Text;
            textBox1.Text = "";
            a.x = a.x.ToUpper();
            if (!a.ValidacijaIskoristenihSlova(iskoristenoSlovo))
            {
               lblMessidzBox.BackColor = Color.Red;
               lblMessidzBox.Text = "Slovo je već bilo u upotrebi!";
            }
            else
            {
                if (a.x.Length == 1)
                {
                    unesenoSlovo = char.Parse(a.x);
                    proveraKoristenosti();
                    iskoristenoSlovo.Add(a);
                    if (!koristenoUReci())
                    {
                        svaSlova[p] = unesenoSlovo;
                        p++;
                    }
                    unesenoSlovo = '\0';
                    textBox1.Text = "";
                }
                else
                {
                    unesenoSlovo = '\0';
                    textBox1.Text = "";
                }

                if (brojPogodjenih == brojKaraktera)
                {
                    textBox1.Enabled = false;
                    lblMessidzBox.Text = "POBEDA";
                    zvukDobijenaIgra.Play();

                    slike[0].Visible = true;
                    slike[1].Visible = true;
                    slike[2].Visible = true;
                    slike[3].Visible = false;
                    slike[4].Visible = true;
                    slike[5].Visible = false;
                    slike[6].Visible = false;
                    slike[7].Visible = false;
```

```
                    slike[8].Visible = false;
                    slike[9].Visible = true;
                }
                else
                {
                    crtajNaGresku();
                }
            }
            string svaIskoristenaSlova = "";
            foreach (Igrica clan in iskoristenoSlovo)
            {
                svaIskoristenaSlova += clan.x;
            }
            lblPotrosenaSlova.Text = svaIskoristenaSlova;
        }
```

Sledeća metoda *povecajSlovo* automatski će uneti slovo koje pritisnemo sa tastature u textbox. Pre nego što proveri da li je slovo korišteno u reči proveriće da li smo mi već prethodno probali uneti to slovo pozivanjem metode *ValidacijaIskoristenihSlova* iz klase Igrica.

```
public bool ValidacijaIskoristenihSlova(List<Igrica>
iskoristenoSlovo)
        {

            foreach (Igrica a in iskoristenoSlovo)
            {
                if (this.x == a.x)
                {
                    return false;
                }
            }
            return true;
        }
```

Ova metoda će za svaki objekat klase *Igrica* koji ćemo za ovu priliku nazvati *a* koji se nalaze u listi *iskoristenSlovo* testirati da li je uneseni karakter jednak nekom od karaktera iz liste. Ako je, onda će se u labeli lblMessidzBox ispisati “Slovo je već bilo u upotrebi!” i postaviće se pozadina labele na crvenu boju. Inače proveravamo da li je unet samo jedan karakter u Textbox i potom pozivamo metodu *proveraKoristenosti.*

```
private void proveraKoristenosti()
        {
```

```
            string s = ucitaneReci[izabranaRec];
            int j = 0;

            if (koristenoUReci() == false)
            {
                for (int i = 0; i < ucitaneReci[izabranaRec].Length;
i++)
                {
                    if (s[i] == unesenoSlovo)
                    {
                        slova[i].Text = unesenoSlovo.ToString();
                        brojPogodjenih++;
                        j++;
                    }
                }
            }
            if (j == 0)
            {
                brojGresaka++;
                zvukPromasenoSlovo.Play();
            }
            else
                zvukPogodjenoSlovo.Play();
        }
```

Metoda *proveraKoristenosti* će proveriti da li je karakter korišten u reči pozivanjem metode *koristenoUReci*.

```
private bool koristenoUReci()
        {
            for (int i = 0; i < 30; i++)
            {
                if (unesenoSlovo == svaSlova[i])
                    return true;
            }
            return false;
        }
```

Ukoliko je *unesenoSlovo* jedanko nekom od tih slova vratiće se true (što znači da je to slovo već korišteno) a u suprotnom će se vratiti false.

Ako vrati false proveriće na koliko se mesta nalazi uneti karakter te će se reproducirati *zvukPogodjenoSlovo.* Ako karakter ne postoji u reči tada će se brojač *brojGresaka* povećati za jedan i reproduciraće se *zvukPromasenoSlovo*.

Ako to slovo nije korišteno u reči to slovo će se uneti u niz *svaSlova* na poziciji *p* i potom će postaviti vrednost na prazan char i isprazniti Textbox.

Svaki put proveravamo da li je broj pogođenih slova jednak ukupnom broju karaktera (slova). Ako je, blokiraće se textbox da više ne možemo unositi slova, u labelu messidzbox ispisace “POBEDA”, reproduciraće se zvuk dobijene igre i postaviti vidljivost slika tako da dobijemo srećnog lutka. Inače ako broj pogođenih nije jednak broju karaktera slova pozvaće se metoda crtajNaGresku. Na kraju metode foreach petljom u labelu *lblPotrosenaSlova* upisuje se novo iskorišteno slovo.

```
private void crtajNaGresku()
        {
            for (int i = 0; i < brojGresaka; i++)
            {
                slike[i].Visible = true;
            }

            if (brojGresaka == 9)
            {
                lblMessidzBox.Text = "KRAJ IGRE";
                textBox1.Enabled = false;
                zvukIzgubljenaIgra.Play();
            }
        }
```

Sledeća metoda *crtajNaGresku* će postavljati sliku za svaku pogresku koju napravimo. Kada broj gresaka bude 9 u labeli messidzobox će se ispisati “KRAJ IGRE”, blokiraće se textbox na način da više nećemo moći upisavati slova te će se reproducirati zvuk izgubljene igre.

```
private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            lblResenja.Visible = !lblResenja.Visible;
        }
```

Sledeća metoda će prikazati ili sakriti labelu rešenja u zavisnosti da li je checkbox označen ili ne. Na početku vrednost vidljivosti labele rešenja će biti false tj. biće nevidljiva.

```
      private void restartIgre(object sender, EventArgs e)
        {
            var x = MessageBox.Show(null, "Želite li ponovno
pokrenuti igru?", "Restart?", MessageBoxButtons.YesNo);
            if (x == DialogResult.Yes)
                Application.Restart();
        }

        private void izlazIgre(object sender, EventArgs e)
        {
            var x = MessageBox.Show(null, "Želite li izaći iz
igre?", "Izađi?", MessageBoxButtons.YesNo);
            if (x == DialogResult.Yes)
                Application.Exit();
        }
```

Slede dve vrlo slično napisane metode: *restartIgre* i *izlazIgre*. U obe metode koristićemo MessageBox prozor sa mogućim izborom da/ne (yes/no) Ako stisnemo da, metoda *restartIgre* pozvace metodu *Restart*() te restartovati aplikaciju, dok će metoda *izlazIgre* pozvati metodu *Exit*() te izaći iz aplikacije.

```
TextBox tbOp1 = new TextBox();
RichTextBox rtOp1 = new RichTextBox();
Form opcije = new Form();
private void opcijeMenu(object sender, EventArgs e)
        {
            opcije.Width = 550;
            opcije.Height = 400;
            opcije.Left = 100;
            opcije.Top = 100;
            opcije.ControlBox = false;
            opcije.Show();
            opcije.Text = "Opcije dodavanja i pregledavanja reči";
            opcije.MaximizeBox = true;

            opcije.Controls.Add(tbOp1);
```

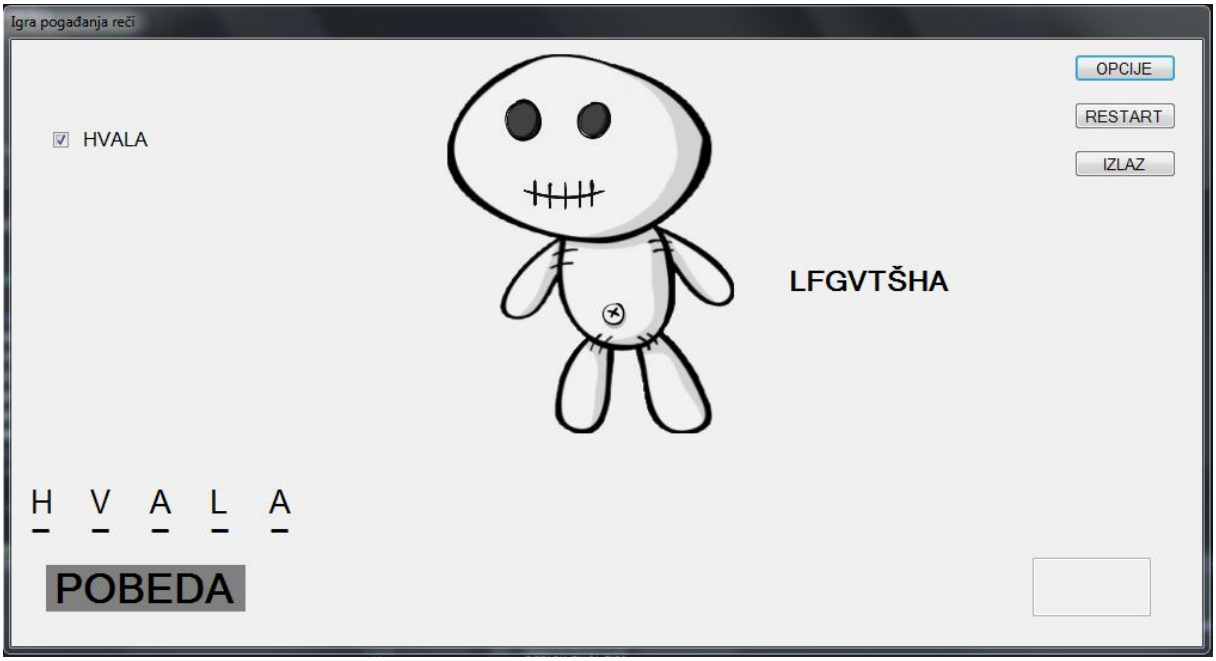
```
            tbOp1.Left = 50;
            tbOp1.Top = 50;

            rtOp1.Left = 50;
            rtOp1.Top = 80;
            rtOp1.Width = 80;
            rtOp1.Height = 200;
            rtOp1.Width = 300;
            opcije.Controls.Add(rtOp1);
            rtOp1.Text = reciIzTxt;
            rtOp1.Text = rtOp1.Text.ToUpper();

            Button btnOp1 = new Button();
            btnOp1.Top = 50;
            btnOp1.Left = 200;
            btnOp1.Text = "DODAJ";
            opcije.Controls.Add(btnOp1);
            btnOp1.Click += new EventHandler(dodajNovuRec);

            Button btnOp2 = new Button();
            btnOp2.Top = 50;
            btnOp2.Left = 400;
            btnOp2.Text = "IZLAZ";
            opcije.Controls.Add(btnOp2);
            btnOp2.Click += new EventHandler(zatvori);
        }

        private void zatvori(object s, EventArgs e)
        {
            opcije.Hide();
        }
        private void dodajNovuRec(object s, EventArgs e)
        {
            string novaRec = tbOp1.Text;
            StreamWriter sw = new StreamWriter("words.txt", true);
            sw.WriteLine(novaRec);
            sw.Close();

            tbOp1.Text = "";

            reciIzTxt += novaRec + "\n";
            rtOp1.Text = reciIzTxt;
            rtOp1.Text = rtOp1.Text.ToUpper();
        }
```

Sledeća metoda *opcijeMeni* će biti pozvana klikom na dugme OPCIJE. Ispred metode pozivamo kreiranje objekata za *textbox, richtextbox* i *form*

opcije. Postavljamo parametere za pozicije i veličine ovih objekata. Prvo dugme “DODAJ” će pozvati metodu *dodajNovuRec* koja će tekst upisan u *textbox* pomoću *StreamWritera* dodati u *txt* datoteku. Takođe upisaće reč u *richtextbox*. Drugo dugme “IZLAZ” služi za izlaz iz prozora opcije tj. dugme će pozvati metodu zatvori koja će sakriti prozor opcija.

4 IZGLED ZAVRŠENOG PROGRAMA

Izgled završenog programa je prikazan na slici ispod.



Slika 4.1 Izgled prozora aplikacije nakon završetka pogađanja reči

5 ZAKLJUČAK

Izrada ovog rada i programa je bilo vrlo poučno iskustvo za mene. Proširio sam staro znanje i stekao nove veštine i kada god sam se susreo sa više mogućih načina izrade nekog dela odabrao sam teži.

Program nije savršen i moglo bi se dodati nekoliko stvari koje bi dodale kvaliteti ove Windows aplikacije, poput:

- pogađanja potpune reči,
- promena broja mogućih grešaka,
- prijava više korisnika,
- mogućnost spremanja korisničkih rezultata nakon svake igre,
- sistem rangiranja među ostalim korisnicima,
- top tabela najboljih troje,
- mogućnost igre u dvoje na način da jedan igrač zadaje reč a drugi pogađa.

Ova Windows aplikacija veoma lako bi se mogla realizovati i kao web aplikacija.

6 PRILOG

Na CD-u je elektronski dokument ovog Završnog rada plus celokupni projekat ove aplikacije (**Igra pogađanja reči**).

7 LITERATURA

- Knjiga:

[1] M. P., OBJEKTNO ORIJENTIRANO PROGRAMIRANJE, Abacastudio d.o.o., Zagreb,2011

- Web stranice:

[2] https://docs.microsoft.com/en-us/dotnet/api/system.random.next?view=net-5.0

[3] https://docs.microsoft.com/en-us/dotnet/api/system.media.soundplayer?view=net-5.0

[4] https://stackoverflow.com/questions/16991387/messageboxbutton-yesno-tutorial/16991427