TEHNIČKO VELEUČILIŠTE U ZAGREBU  
STRUČNI STUDIJ INFORMATIKE

„Hangman“  
Objektno orijentirano programiranje 2

Ivan Vukosav

Zagreb, lipanj 2020.

Sadržaj

[Uvod 3](#_Toc42142720)

[Teme korištene u izradi aplikacije 3](#_Toc42142721)

[Opis aplikacije 3](#_Toc42142722)

[Unos podataka igrača 6](#_Toc42142723)

[Realizacija 8](#_Toc42142724)

[Zaključak 10](#_Toc42142725)

Uvod

U seminaru sam odlučio napraviti popularnu igru iz djetinjstva koja se igrala na papiru “Hangman”. Igra je ostvarena kao Windows Forms .NET aplikacija pisana u C# jeziku. Za izradu aplikacije korišten je „Microsoft Visual Studio“.

Teme korištene u izradi aplikacije

* Upoznavanje sa razvojnim okruženjem Vs, osnove c# objektnog jezika, klase i UML
* XML, iznimke, tekstualne datoteke, log, korisničke postavke
* Ugrađene strukture podataka i generičke klase
* Jednostavno grafičko sučelje

Opis aplikacije

„Hangman“ igra je veoma poznate igra s kojima je većina ljudi upoznata. Za one koji ne znaju ili su zaboravili ova igra je rađena igra tako da računalo generira jednu nasumičnu riječ koja se nalazi unutar niza svih riječi i igrač mora pogađati slova izgenerirane riječi, za svaki krivo odabrano slovo igrač se približava kraju igre, a ukoliko pogodi slova približava se pobjedi igre.

Kada korisnik prvi puta pokrene aplikaciju moći će vidjet sljedeće:

A screenshot of a social media post

Description automatically generated

Čim se korisniku otvori aplikacija riječ je izgenerirana i može se odmah početi igrat. Korisnik će znati koliko slova ima riječi po broju crtica ispod slike vješala.

Slova će se unositi preko konzole u “textbox” pored gumba “Unesi znak” , u “textbox” elementu je dozvoljen upis samo jednog karaktera i pritiskom na sami gumb “Unesi znak” program će provjeriti nalazi li se slovo unutar zadane riječi.

Sve riječi su pisane malim slovom da igrač ne mora pogađati da li je zadana riječ pisana velikim ili malim slovima te ako će korisnik i htjet upisati velika slova ona će se automatski upisat kao mala. Slova koja su već pokušana će postati onemogućena tj. ponovni unos pogrešnog slova aplikacija neće gledati kao gubitak “života” već će javiti poruka za ponovni pokušaj unosa istog slova. Isto tako neće biti penala za unos “ničega”.

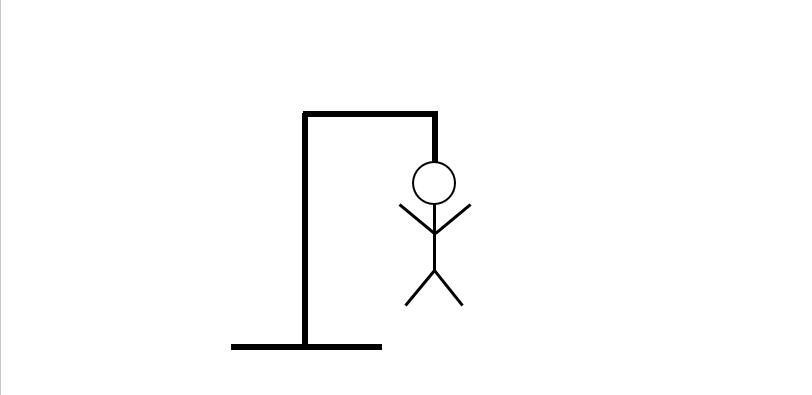
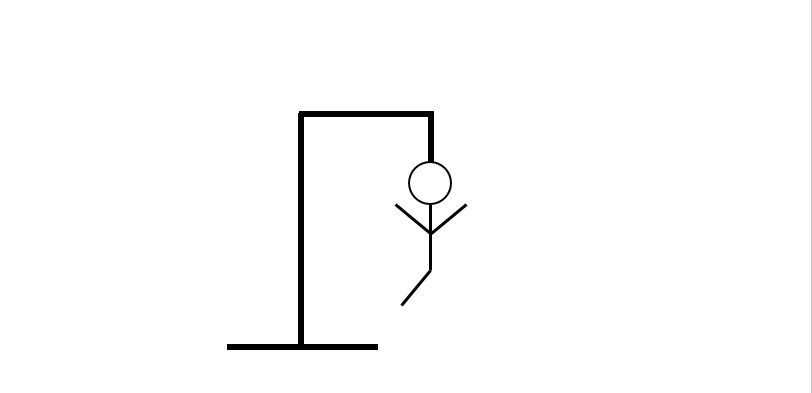
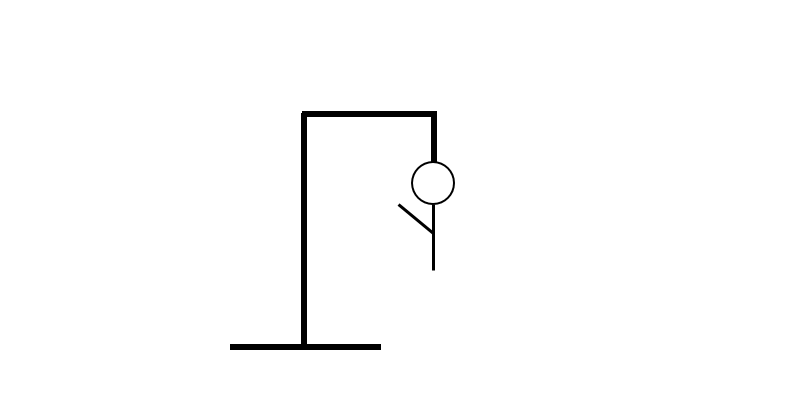
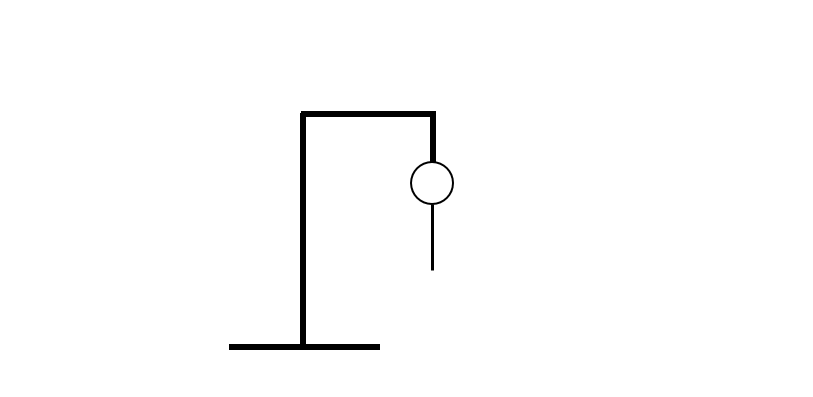
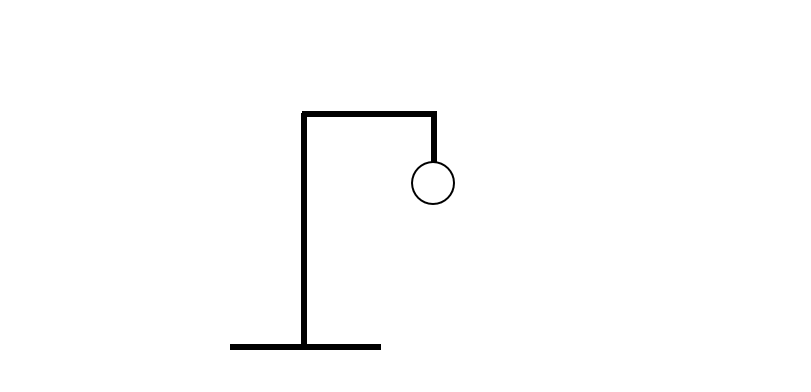
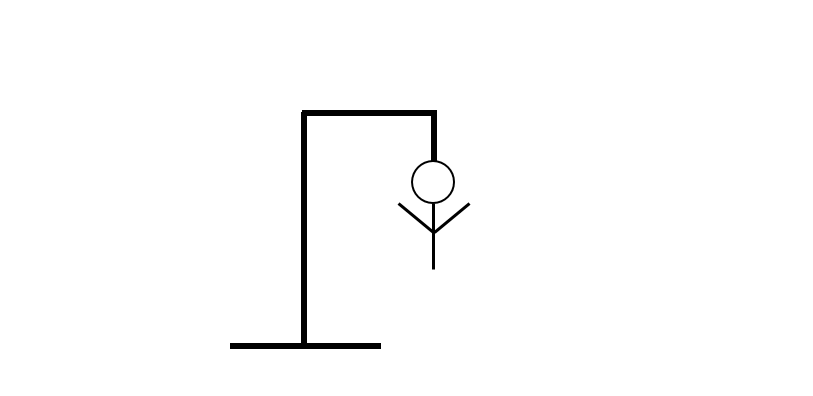
A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Ako se slovo nalazi unutar zadane riječi onda će na mjestu gdje se slovo nalazi u samoj riječi prikazati slovo umjesto crtice, a ako se ne nalazi početi će se crtati tijelo obješene osobe. Tijelo dok ne igra završi se iscrtava ud po ud te se sastoji od pet mogućih pogrešaka jer na šestom kriviom unosu slova je osoba nacrtana I to označava kraj igre.



Unos podataka igrača

Informacije igrača poput imena i prezima moći će svaki igrač unijeti po izboru te će se te informacije spremati u posebnu xml datoteku, a igraču će biti priložene u “listbox-u” u desnom kutu.

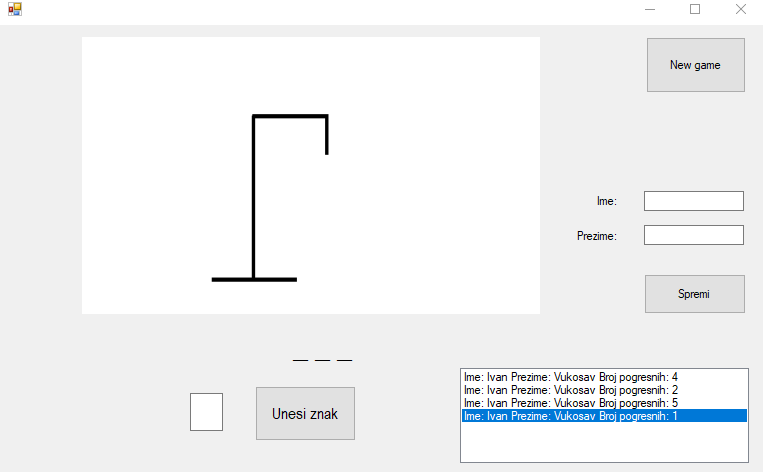
A screenshot of a social media post

Description automatically generated

Ako će igrač htjeti započeti novu igru dovoljno je samo stisnuti na gumb “New game” i program će generirati novu riječ, te resetirati broj pogrešaka i unosa istog slova. Za svaku novu igru informacije igrača(ime, prezime i broj pogrešnih unosa) se neće brisati već će ostati unutar “listbox” elemenata i biti će unutar njega sve dok se iz aplikacije ne izađe.

A screenshot of a cell phone

Description automatically generated



Realizacija

public class Spremanje

{

public string ime { get; set; }

public string prezime { get; set; }

public int Pogresni{ get; set; }

public override string ToString()

{

return "Ime: " + ime +

" Prezime: " + prezime +

" Broj pogresnih: " + Pogresni;

}

}

private void Serialize()

{

try

{

// serijalizacija tipa List<Korisnik>

XmlSerializer serializer = new XmlSerializer(Korisnici.GetType());

using (StreamWriter sw = new StreamWriter("Hangman1.xml"))

{

serializer.Serialize(sw, Korisnici);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Deserialize()

{

try

{

if (File.Exists("Hangman1.xml"))

{

XmlSerializer serializer = new XmlSerializer(Korisnici.GetType());

using (StreamReader sr = new StreamReader("Hangman1.xml"))

{

Korisnici = (List<Spremanje>)serializer.Deserialize(sr);

}

}

else throw new FileNotFoundException("File not found");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

Ovaj programski odsječak se odnosi na spremanje podataka u XML datoteku te prikaz istih podataka na sučelju korisnika.

Sljedeći kod se odnosi na samo slovo. Za vrijeme pogađanja će kod ispitivati da li slovo postoji u riječi, ako slovo postoji onda se slovo zapisuje umjesto “\_”, a ako ne postoji kreira se slika.

private void Unesi\_Click(object sender, EventArgs e)

{

string[] a = UnosSlova.Lines;//textbox slovo

string b = String.Join(" ", a);//pretvori u string

char[] privremenaProm = slova.ToCharArray();//crtice

char[] Slova = rijeci[NRijec].ToCharArray();//slova

if (String.IsNullOrEmpty(UnosSlova.Text))

{

MessageBox.Show("Niste unijeli vrijednost");

}

else

{

if (rijeci[NRijec].Contains(a[0]))

{

//razdvoji rijec u slova //update

char Pogadanje = b.ElementAt(0);// u char

for (int i = 0; i <2; i++)

{

if (Slova[i] == Pogadanje)

{

privremenaProm[i] = Pogadanje;

}

}

slova = new string(privremenaProm);

Prikaz();//update prikaza

}

else

{

if (Nesmije.Contains(b))

{

MessageBox.Show("Vec ste pokušali to slovo!");

}

else

{

index++;

}

}

}

Nesmije.Add(b);//spremi slovo u niz i prolazimo nizom i kazemo da se ne smije stisnut ako se nalazi u ovom nizu

pogresni = index;//global

if (index <= 6)

{

HangSlike.Image = Slike[index];

}

if (index == 6)

{

MessageBox.Show(String.Format("Game Over!\nRijec je: {0}", rijeci[NRijec]));

}

if (!slova.Contains("\_"))

{

MessageBox.Show("Pobjeda!");

}

}

Zaključak

Aplikacija nema puno mogućnosti za unaprijeđenje osim samog fokusa na riječi. Za daljne unaprijeđenje smatram kako bi veća baza riječi tj. postojala mogućnost u odabira različitih kategorija riječi.