SEMINARKSA NALOGA PRI PREDMETU PROGRAMIRANJE 1

IGRA:

TRI V VRSTO / TIC TAC TOE

Šola: **Višja strokovna šola Kranj**

Smer: **Informatika**

Predmet: **Programiranje 1**

Študijsko leto: **2016/2017**

Študent: **Gašper Črtalič**

Mentor: **Srečko Uranič**  **Tržič, 4.4.2017**

KAZALO VSEBINE

[1. UVOD 1](#_Toc479201534)

[2. METODA – pritisk na gumb 2](#_Toc479201535)

[2](#_Toc479201536)

[3. METODA – preveri zmago 3](#_Toc479201537)

[4. METODA –onemogoci gumbe 5](#_Toc479201538)

[5. ZAVIHKI NA APLIKACIJI 5](#_Toc479201539)

[6](#_Toc479201540)

[6. METODA – v gumb 7](#_Toc479201541)

[7. VSEBINA SEKUNDARNEGA OKNA 8](#_Toc479201542)

[8. ČASOVNIK 9](#_Toc479201544)

[9. ZAKLJUČEK 11](#_Toc479201545)

[10. IZJAVA O AVTORSTVU 11](#_Toc479201546)

[Slika 1: form2 1](file:///C:\Users\GEP\Documents\PROGRAMS\Seminarska%20Naloga\Gasper_Crtalic_SN_TicTacToe.docx#_Toc479201563)

[Slika 2: form1 1](file:///C:\Users\GEP\Documents\PROGRAMS\Seminarska%20Naloga\Gasper_Crtalic_SN_TicTacToe.docx#_Toc479201564)

[Slika 3: Izdelava gumbov 2](file:///C:\Users\GEP\Documents\PROGRAMS\Seminarska%20Naloga\Gasper_Crtalic_SN_TicTacToe.docx#_Toc479201565)

[Slika 4: Najboljša zmaga 4](#_Toc479201566)

[Slika 5: Zavihki 5](file:///C:\Users\GEP\Documents\PROGRAMS\Seminarska%20Naloga\Gasper_Crtalic_SN_TicTacToe.docx#_Toc479201567)

[Slika 6: O aplikaciji 6](#_Toc479201568)

[Slika 7: user friendly 7](#_Toc479201569)

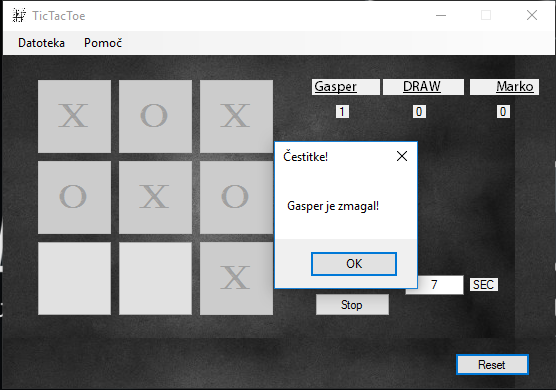
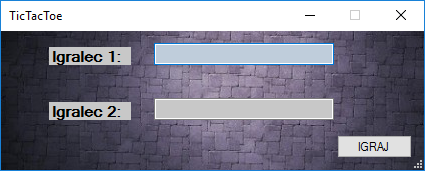
[Slika 8: form2 8](#_Toc479201570)

[Slika 9: časovnik 9](file:///C:\Users\GEP\Documents\PROGRAMS\Seminarska%20Naloga\Gasper_Crtalic_SN_TicTacToe.docx#_Toc479201571)

# UVOD

Za seminarsko nalogo z naslovom »Tri v vrsto« sem se odločil, ker je to prva izdelana računalniška igra na svetu. Ker predhodno nisem imel toliko znanja o samem programiranju, mi je projekt predstavljal iziv. Program sem izdelal s pomočjo Microsoftovega orodja Visual Studio 2015. Izbiral sem med konzolno aplikacijo ter okensko aplikacijo. Na koncu sem se odločil za okensko (windows form application), ker me je zanimalno grafično obdelovanje same aplikacije. S problemi sem se sočal postopkovno. Glavni problem aplikacije Tri v vrsto je preverjanje zmage igralcev. Zmago sem preverjal skozi metodo »jeZmagovalec«, ki je obsegala preverjanje kombinacij na vodoravni, navpični in diagonalni ravni. Cilj moje seminarske naloge je bil, da izdelam aplikacijo, ki je vizualno sprejemljiva in hkrati zabavna za igranje in krajšanje prostega časa uporabnikom. Med izdelovanjem seminarke naloge sem naletel na določene napake, ki sem jih s pomočjo medmrežja znal popraviti. Seminarski nalogi sem posvečil veliko časa, saj me izdelovanje programov zelo zabava.

Skozi svojo seminarsko nalogo bom predstavil izdelovanje same aplikacije. Ponazoril bom probleme na katere sem naletel in načine kako sem probleme rešil. Poskušal bom predstaviti spoznanja, ki so pomembna za razumevanje problema naloge.



Slika : form2

Slika : form1

# METODA – pritisk na gumb

Ta metoda je zadolžena da, ob pritisku na gumb vrne X oz. O odvisno od tega kdo je navrsti. Ob sami izdelavi sem tej metodi dodal časovnik, ki se začne ko prvi igralec pritisne na gumb. Program ve kdo je na vrsti s pomočjo vrednosti bool. Pomembno je, da tisti, ki je navrsti – ko pritisne na gumb, da program zamenja igralca.

bool Navrsti = true; // true= X; false=Y

int Navrsti\_stej = 0

private void pritisk\_gumb(object sender, EventArgs e)

{

Button b = (Button)sender;

//če pritisnes na gumb, glede na to kdo je na vrsti se izpiše x ali o

timer1.Start();

if (Navrsti)

{

b.Text = "X";

timer1.Start();

}

else

b.Text = "O";

Navrsti = !Navrsti;

b.Enabled = false; //Ko igralec klikne na gumb in mu da vrednost, se le ta ko znova kliknem nesprmemeni

Navrsti\_stej++; //vsakic ko pritisnemo na gumb spremenljivka steje

preveri\_Zmago();

}

# 

Slika : Izdelava gumbov

# METODA – preveri zmago

Metoda vsako potezo sproti preverja, če imamo zmagovalca. Kako vemo, da je zmagovalec? Čisto preprosto. Tako da pregledamo vodoravne, navpične in diagonalne kombinacije. To naredimo s pomočjo if gnezednih stavkov. Če so znaki na gumbih enaki in so postavljeni skupaj v stolpec, vrstico ali v diagonalo – to pomeni, da imamo zmagovalca in boolean vrednost jeZmagovalec postavimo na true. Če v devetih potezah ni zmagovalca to pomeni, da je igra neodločena. Ob vsaki zmagi ali remiju se v novem oknu izpiše ime zmagovalca.

private void preveri\_Zmago() //preverjanje ali imamo zmagovalca

{

bool jeZmagovalec = false;

//preverjanje vodoravnih kombinacij

if ((A1.Text == A2.Text) && (A2.Text == A3.Text) && (!A1.Enabled))

/\*Če je gumb A1 ugasnen in so vsi trije v isti vrsti enaki potem je zmagovalec\*/

jeZmagovalec = true;

else if ((B1.Text == B2.Text) && (B2.Text == B3.Text) && (!B1.Enabled))

jeZmagovalec = true;

else if ((C1.Text == C2.Text) && (C2.Text == C3.Text) && (!C1.Enabled))

jeZmagovalec = true;

//preverjanje navpičnih kombinacij

if ((A1.Text == B1.Text) && (B1.Text == C1.Text) && (!A1.Enabled))

/\*Če je gumb A1 ugasnen in so vsi trije v istem stolpcu enaki potem je zmagovalec\*/

jeZmagovalec = true;

else if ((A2.Text == B2.Text) && (B2.Text == C2.Text) && (!A2.Enabled))

jeZmagovalec = true;

else if ((A3.Text == B3.Text) && (B3.Text == C3.Text) && (!A3.Enabled))

jeZmagovalec = true;

//preverjanje diagonalnih kombinacij

else if ((A1.Text == B2.Text) && (B2.Text == C3.Text) && (!A1.Enabled))

jeZmagovalec = true;

else if ((A3.Text == B2.Text) && (B2.Text == C1.Text) && (!C1.Enabled))

jeZmagovalec = true;

**if (jeZmagovalec)**

**{**

**Onemogoci\_gumbe();**

**timer1.Stop();**

**string zmagovalec = "";**

**if (Navrsti)**

**{**

**magovalec = igralec2;**

**stetjeZmagaO.Text = (Int32.Parse(stetjeZmagaO.Text) + 1).ToString();**

**}**

**else**

**{**

zmagovalec = igralec1;

stetjeZmagaX.Text = (Int32.Parse(stetjeZmagaX.Text) + 1).ToString();

} MessageBox.Show(zmagovalec + " je zmagal!", "Čestitke!");

Da si delo pri iskanju zmagovalca olajšamo lahko polja smiselno poimenujemo npr. A1 A2 A3 B1 B2 B3 C1 C2 C3. Poleg zgoraj naštetih stvari sem v to metodo vključil bleženje najhitrejših zmag. Najhitrejša zmaga se skupaj z igralcem in njgovim časom vpiše v tabelo. Poleg tega pa se tekstovno datoteko shrani najhitrejša zmaga.

string[] najZmaga = new string[5];

if (jeZmagovalec)

{

int najCas = 0;

if (cas > najCas)

{

najCas = cas;

for (int i = 0; i < 1; i++)

{

najZmaga[i] = " | " + zmagovalec + " | " + cas + " sekund";

}

//beleženeje v tekstovno datoteko

StreamWriter SW = new StreamWriter(@"najZmaga.txt");

for (int i = 0; i < 1; i++)

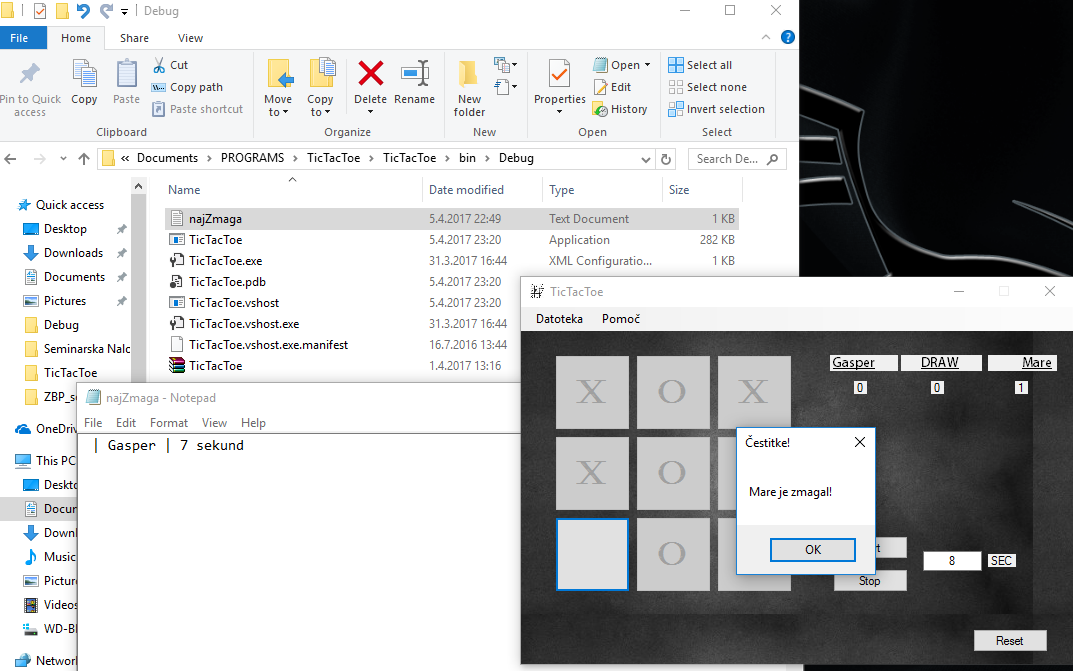
{

SW.WriteLine(najZmaga[i]);

}

SW.Close();

}



Slika : Najboljša zmaga

# METODA –onemogoci gumbe

Metodo onemogoči gumbe sem uporabil, ker je pomembno, da se ob kliku na polje gumb onemogoči, da nanj nemoremo vnesti vrednsoti. Prav tako, se ob koncu igre zaklenejo vsi preostali gumbi. To sem naredil s pomočjo varovalnega bloka.

private void Onemogoci\_gumbe() //ko se igra konca se onemogočijo gumbi

{

try

{

foreach (Control c in Controls)

//Control je generična spremenljivka za to jo convertamo v gumb

{

Button b = (Button)c;

//za vsak gumb na nasi aplikcaiji ga ugasnemo

b.Enabled = false;

}

}

catch { } //ni edinstvene vrednsoti za to catch pustimo prazen

}

# ZAVIHKI NA APLIKACIJI

Pod zavihkom datoteka sem izbral možnosti – metode: igraj znova, štej od začetka in zapri aplikacijo. Pod zavihkom pomoč so informacije o avtorstvu.

Za objekt igraj znova sem prav tako uporabil varovalni blok. Ob kliku na izbirno možnost se vsa polja resetirajo.

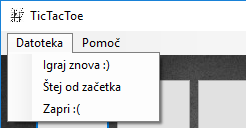
private void igrajZnovaToolStripMenuItem\_Click(object sender, EventArgs e)

{

Navrsti = true; //zavihek igraj znova

Navrsti\_stej = 0;

foreach (Control c in Controls)

 {

Slika : Zavihki

try

{

Button b = (Button)c;

b.Enabled = true;

b.Text = "";

}

catch { }

}

Možnost štej od začetka resetira vse seštevke zmag in remijev igralcev.To enostavno naredimo tako, da polje v katerem se seštevjo zmage ali remiji vstavimo string vrednost »0«;

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

{

stetjeDraw.Text = "0";

stetjeZmagaO.Text = "0";

stetjeZmagaX.Text = "0";

}

Če želimo zapreti aplikacijo lahko enostavno izberemo možnost zapri. Metoda izgledano tako:

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

{

Application.Exit();

}

Pod zavihkom pomoč se najaja močnost O meni. Če pritisnemo na to se pojavi sporočilno okno, v katerem pišejo podatki.

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

{

MessageBox.Show("Seminarska naloga narejena v okviru predmeta Programiranje 1 \n\nAvtor: Gašper Črtalič, 1l", "O aplikaciji");

}

# 

Slika : O aplikaciji

# METODA – v gumb

Metodo vGumb sem uporabil zaradi tega, ker sem želel aplikacijo narediti bolj »user friendly«. Kar metoda naredi je to, da ob premiku miške na polje oz. gumb prikaže vrednost igralne poteze ( X ali O). Najlprej moramo seveda poskrbeti, da inicializiramo gumb v spremenljivko, katero lahko kasneje obdelujemo.

private void vGumb(object sender, EventArgs e)

{

Button b = (Button)sender;

if (b.Enabled)

{

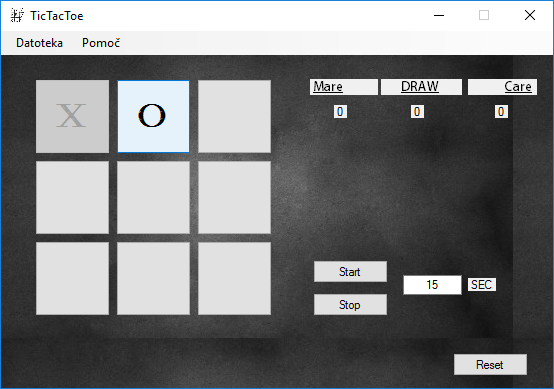
if (Navrsti)

b.Text = "X";

else

b.Text = "O";

}



Slika : user friendly

Prav tako je bilo treba poskrbeti, da ko uporabnik miško odmakne iz polja se vsebina izbriše. Če to nebi implementiral bi se na vseh mestih, kjer bi se dotaknili gumba, znašla igralčeva figura ( X ali O ). Za to sem naredil metodo, ki naredi ravno to.

private void izGumb(object sender, EventArgs e)

{

Button b = (Button)sender;

if (b.Enabled)

{

b.Text = "";

}

}

# VSEBINA SEKUNDARNEGA OKNA

Okno, ki se prvo prikaže uporabniku, je namenjeno poizvedbi po igralcih, za lažjo in bolj uporabno vsebino aplikacije. Funkcionalnosti okna sem napisal v drugem razredu. Okno vsebuje dva tekstovna pripomočka in gumb igraj. Poleg gumba igraj sem v kodo implemetiral, da se igra lahko prične tudi s stiskom na enter. Igralcu 1 se dodeli vrednost X in obratno velja za igralca 2.

# 

Slika : form2

namespace TicTacToe

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

TicTacToe.NastaviImenaIgralcev(igralec1.Text, igralec2.Text);

this.Close();

}

private void igralec2\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar.ToString() == "\r")

//če uporabnik pritisne enter se igra zacne

button1.PerformClick();

}

}

}Ker sem izdelal prvo oken za samo igro sem moral napisati tudi metodo, ki bo zagotovila, da se bo oken za vnos igralcev odpru pred samo igro. Prav tako dodeli vrednosti, ki smo jih vpisali.

private void TicTacToe\_Load(object sender, EventArgs e)

{

Form2 login = new Form2();

login.ShowDialog(); //odpre prvo login form nato sele igrico

label1.Text = igralec1;

label3.Text = igralec2;

}

# ČASOVNIK

Za nadrgadjo k aplikaciji sem se odločil, da bom izdelal časovnik, ki se prične izvajati, ko prvič pritisnemo na gumb in se preneha izvajati, ko se igra zaključi. Najboljši izmerjen čas se nato zabeleži v tabelo – iz tabele pa se podatki prepišejo v tekstovno datoteko.

int cas = 0; //štoparica

private void timer1\_Tick(object sender, EventArgs e)

{

cas++;

casomer.Text = cas.ToString();

}

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

{

timer1.Enabled = true;

timer1.Start();

}

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

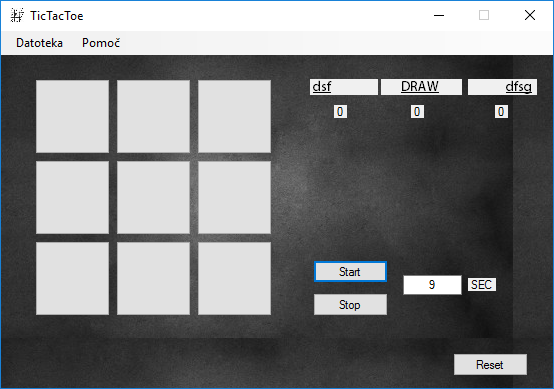
{

timer1.Stop();

}

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

{

 casomer.Text = "0";

}

Slika : časovnik

Najhitrejša zmaga se preverja s pogoji in s pomočjo funkcije stream writer zapisuje v tekstovno datoteko.

//beleženje najhitrejših zmag

string[] najZmaga = new string[5];

if (jeZmagovalec)

{

int najCas = 0;

if (cas > najCas)

{

najCas = cas;

for (int i = 0; i < 1; i++)

{

najZmaga[i] = " | " + zmagovalec + " | " + cas + " sekund";

}

//beleženeje v tekstovno datoteko

StreamWriter SW = new StreamWriter(@"najZmaga.txt");

for (int i = 0; i < 1; i++)

{

SW.WriteLine(najZmaga[i]);

}

SW.Close();

}

# ZAKLJUČEK

S svojo seminarsko nalogo sem zadovoljen, ker dela tako kot želim. Imel sem veliko testnih preizkušen s strani uporanikov, saj sem želel izvedeti če dela vse tako kot sem pričakoval. Idej za izboljšavo je še veliko. Morda bom v nadaljne izkoristil čas in seminarsko še bolj izboljšal. Implementirati si želim umetno inteligenco, vendar nimam dovolj predznanja zaenkrat, da bi idejo zrealiziral. Časovnik ima pomankljivost, da se ob kliku na izbirno možnost »igraj znova« skrijejo napisi na ukaznih gumbih. Objektivno oceno mojega projekta bi opredelil z 9.

# IZJAVA O AVTORSTVU

Izjavljam, da sem nalogo opravil samostojno in da sem njen avtor. Zavedam se, da v primeru, če izjava prvega stavka ni resnična, kršim disciplinska pravila.