

REGIONALNI CENTAR ZA TALENTE IZ PRIRODNIH
I TEHNIČKIH NAUKA U VRANJU

PROGRAM PC SLAGALICA

Autor:

Nikola Stojiljković učenik II godine Gimnazije “Svetozar Marković”,
član FDV “Hristifor Crnilović-Kica”, Vlasotince

Mentor:

Novica Stojilković, dipl. eng. el. samostalni stručni saradnik pri F.D.V. “H. Crnilović-Kica”
Vlasotince

PROGRAM PC SLAGALICA

AUTOR

NIKOLA STOJILJKOVIĆ, učenik II godine gimnazije “Svetozar Marković”

MENTOR:

Novica Stojilković, dipl. eng. el. samostalni stručni saradnik pri F.D.V. “ H. Crnilović-Kica”
Vlasotince

REZIME

Program **PC Slagalica** je pisan u okruženju Delphi. Delphi je okruženje koje omogućava brzi razvoj aplikacija (RAD okruženje) i poseduje veliki broj vizuelnih komponenata (VCL) Osnova okruženja Delphi čini Object Pascal, tako da poznavanje bilo Object Pascal-a ili Turbo Pascal-a može da Vam bude od velike koristi.

PC Slagalica je simulacija popularnog kviza “TV Slagalica” koji se prikazuje na RTS 1. Sastavljena je od četiri igara: “**Moj Broj**”, “**Spojnice**”, “**Dodavanje bojeva**” i “**Ko zna zna**”. Skoro sve igre su urađene identično osim igre “**Ko zna zna**” u kojoj imate tri ponuđena odgovora umesto da upišete tačno rešenje. Takođe u igri “**Moj broj**” možete da vidite najbolje rešenje koje je moguće dobiti od ponuđenih brojeva. Na ideju da napravim igru “**PC Slagalica**” došao sam tako što sam kao mali svaki dan gledao kviz “TV Slagalica” i imao želju da učestvujem, a kako nisam bio u mogućnosti odlučio sam da napravim simulaciju u kojoj ću moći da učestvujem svaki dan.

Ključne reči: Program, Unit, Procedure, class, VCL, Slagalica, Stack, Pointer, Math

SUMMARY

Program **PC Slagalica** has been written in Delphi. Delphi makes possible fast development application and it has many visual components (VCL). Root of Delphi is Object Pascal, so knowledge Object Pascal or Turbo Pascal can be a big help.

PC Slagalica is simulation of popular quiz “TV Slagalica” which is shown on RTS 1. It contains four different games: “**Moj broj**”, “**Spojnice**”, “**Dodavanje bojeva**” and “**Ko zna zna**”. Almost each of the games has been made identically except the game “**Ko zna zna**” where you have got three offered answers instead of writing a solution. Also, in the game “**Moj broj**” you can see the best solution that can be made with the given numbers. The idea to make “**PC Slagalica**” fell on my mind when I was a kid. Then I was watching “TV Slagalica” everyday, and I had a wish to participate in the quiz. That was impossible for me because I was too young, so I made it myself. I made a program “**PC Slagalica**” where I am able to participate anytime I want.

keywords: program, unit, procedure, class, VCL, Slagalica, Stack, Pointer, Math

UPUTSTVO ZA KORISNIKA

Da biste pokrenuli program potrebno je da dva puta kliknete primarnim tasterom miša na ikonu programa. Da ne bi dobijali grešku po pokretanju programa, potrebno je instalirati BDE (setup se nalazi na disku). Po pokretanju programa na ekranu se vidi jedna forma. Na njoj se nalaze dva dugmeta. Dugme za izlaz i za početak igre (slika 1).



Slika 1, izgled glavne forme PC Slagalica

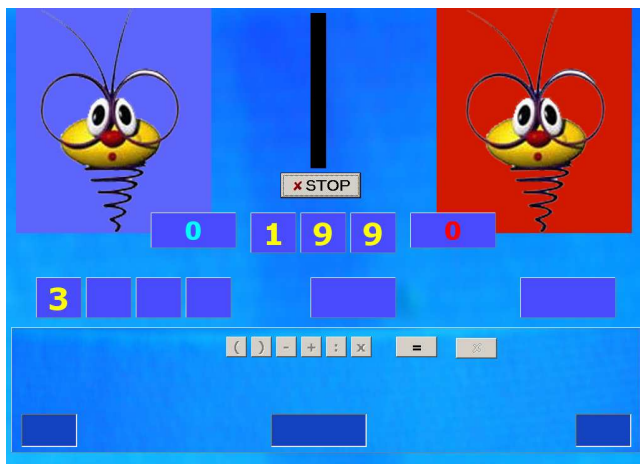


Slika 2, izgled dugmeta kada je kursor na njemu

Kada pređete mišem preko dugmeta, dugme postaje drugačije (slika 2). Isto se dešava sa svim dugmićima.

Pravila za sve igre su objašnjena u Help-u (nalazi se na glavnoj strani, Slika 1).

Klikom na dugme počni igru otvara se nova forma u kojoj upisujete ime i prezime za dva takmičara koja se međusobno takmiče a zatim se otvara forma na kojoj je simulirana igra „Moj broj“ (slika 3). To je početak igre.



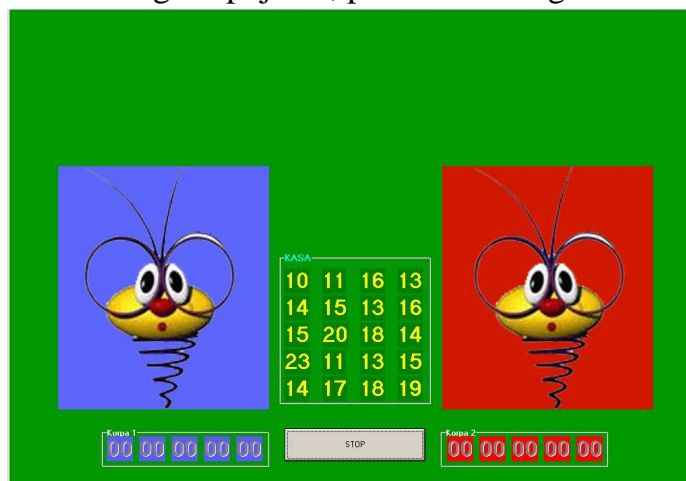
Slika 3, početak igre Slagalica. Izgled igre Moj Broj.

Posle završetka igre Moj Broj, prelazimo na igru Spojnice (Slika 3b).

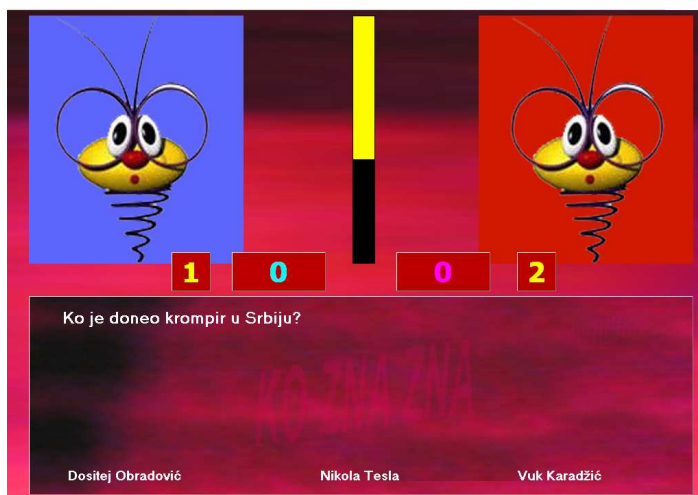


Slika 3b, izgled igre “Spojnice”

Posle završetka igre Spojnice, prelazimo na igru Dodavanje brojeva (Slika 4).



Slika 4, izgled igre Dodavanje Brojeva. Posle završetka igre “Dodavanje Brojeva”, prelazimo na igru “Ko Zna Zna” (Slika 5).



Slika 5, izgled igre “Ko zna zna”

TEHNIČKA DOKUMENTACIJA

Koristio sam programski jezik Delphi 7, zato što je objektno orijentisan, ima veliki broj VCL komponenti i zato što je najbolje i najpopularnije programsko okruženje za pisanje ozbiljnih aplikacija.

NEŠTO O DELPHI-u

Godine 1994 Borland počinje rad na RAD alatu pod šifrovanim nazivom Delphi. Tada je odlučeno da arhitektura modela komponenti predstavlja najbolji način za implementaciju RAD. Bilo je neophodno opredeliti se za programski jezik koji će biti srce sistema.

U to vreme Borland je bio jedina firma na tržištu prevodilaca koja je prodavao Pascal prevodioce. Borland je isto tako bio poznat kao kompanija koja proizvodi najbolje alate za Pascal. Ako ste bili Pascal programer, verovatno ste koristili Borlandov Turbo Pascal u jednom ili drugom obliku. Borland je više ili manje “posedovao” Pascal. Naravno legalno gledano Borland nije posedovao Pascal, ali nije bilo sumnji da njegova pozicija u svetu Pascal-a dozvoljava određene slobode u implementaciji novih mogućnosti i usavršavanja. Dodatno, nije postojao komitet za Pascal standarde, miti je bilo koji pisani standard definisao Pascal jezik. Zato je Borland napravio Delphi koristeći Pascal kao osnovni jezik (Borlandov interni šifrovani naziv je ujedno postao službeni naziv proizvoda).

Pre nego što je Delphi počeo da postoji, Borland je već izmenio Pascal u pozitivnom pravcu. Na primer Borland je već proširio Pascal kreiranjem novog jezika pod nazivom Object Pascal. Moglo bi se reći da je Object Pascal u odnosu na Pascal isto što i C++ u odnosu na C. Object Pascal je dodao klase Pascalu i tako svstao Pascal u svet objektno orijentisanih jezika (Object Oriented Programming Language-OOP). Kada je Delphi bio razvijen, dodata su nova ponašanja i ključne reči koje su imale veze sa modelom komponenti. Dodate su ključne reči poput published između ostalog. Ovo je omogućilo Borlandu da u potpunosti implementira snagu modela komponenti. Modifikujući jezik Pascal tako da se uklapi u model komponenti, Borland je bio u mogućnosti da na ispravan način implementira RAD. U osnovi jezik Object Pascal je modifikovan koliko je bilo potrebno, dok su stavke vezane za dizajn došle nakon razvoja tada nepoznatog proizvoda nazvanog Delphi. Kao rezultat dobio se jezik koji radi sa modelom komponenti.

Iako je izmena Pascal –a bila razmatrana kao veliki korak za Borland, se ovo nije dogodilo kao presedan. Prethodno je Microsoft modifikovao jezik Basic i napravio novi jezik pod nazivom Visual Basic. Ovaj novi jezik je bio

skoro neprepoznatljiv u poređenju sa originalnim jezikom Basic, koji je poslužio kao njegova osnova.

Borland je preuzeo rizik da modifikuje Pascal. Nakon svega imao je osnovu lojalnih kupaca koji možda nisu mogli da prihvate izmene jezika koji su upoznali i zavoleli. Ipak Borland je bio u dobroj poziciji na tržištu Pascal-a i krenuo je napred sa svojim planovima. Rezultat je bio pravi hit, naravno.

Strukture koje su korišćene: program ima složen algoritam koji sadrži veliki broj razgranatih (if), linijskih i cikličnih struktura, takođe sadrži i stekove i povezane liste pomoću kojih se se računa vrednost izraza.

Između ostalog poziva se veliki broj objekata posebne namene, kao i BDE (Modul programa za rad sa bazama podataka).

Kod pomoću kog se izračunava vrednost izraza se nalazi u unitu
CalcUnit:

```
unit CalcUnit;
```

```
interface
```

```
type
```

```
  TType = (operand, op);  
  TOperand = Real;
```

```
  PNode = ^TNode;  
  TNode = record  
    PNTType: TType;  
    operand: TOperand;  
    op: Char;  
    Next: PNode;  
  end;
```

```
function CalculateExpression(S: String): integer; //funkcija koja vraca resenje izraza S kao rezultat
```

```
implementation
```

```
var
```

```
  Stack: PNode;
```

```
procedure push(x: PNode; var Stack: PNode);
```

```
begin
```

```
  x^.Next := Stack;  
  Stack := x;
```

```
end;
```

```

function pop(var Stack: PNode): PNode;
begin
    pop := Stack;
    if Stack <> nil then
        Stack := Stack^.Next;
    end;
end;

function Calc(operand1, operand2: TOperand; op: Char): TOperand;
begin
    case op of
        '+': Result := Operand1 + Operand2;
        '-': Result := Operand1 - Operand2;
        '*': Result := Operand1 * Operand2;
        '/': Result := Operand1 / Operand2;
    end;
end;

function Calculate(PN: PNode): Real;
var
    i, n: integer;
    x: PNode;
    operand1, operand2: TOperand;
    Stack: PNode;
begin
    while PN <> nil do begin
        if PN^.PNTType = operand then begin
            New(x);
            x^.operand := PN^.operand;
            x^.PNTType := operand;
            Push(x, Stack);
        end
        else begin
            New(x);
            x := pop(Stack); operand2 := x^.operand;
            x := pop(Stack); operand1 := x^.operand;

            x^.operand := Calc(operand1, operand2, PN^.op);
            Push(x, Stack);
        end;
        PN := PN^.Next;
    end;

    Result := pop(Stack)^(operand);
end;

procedure Add(x: PNode; var Root, PN: PNode);
begin
    if Root = nil then begin
        New(Root);
        PN := Root;
    end
    else if Root <> nil then begin

```

```

    New(Root^.Next);
    Root := Root^.Next;
end;

Root^.operand := x^.operand;
Root^.op := x^.op;
Root^.PNTType := x^.PNTType;

Root^.Next := nil;
end;

procedure MakePN(var PN: PNode; V: String);
var
    p, tmp: PNode;
    opera: Char;
    Num: Real;
    e: integer;
    S: String;
    PNTType: TType;
    pri1, pri2: integer;
    i, n: integer;

function prioritet(c: Char): integer;
begin
    case c of
        '+', '-': Result := 1;
        '*', '/': Result := 2;
        '(': Result := 0;
    end;
end;

begin
    p := nil;
    i := 0; n := Length(V);
    while i < n do begin
        i := i + 1; //read(c);
        if {c}V[i] in ['0'..'9'] then PNTType := operand
        else PNTType := op;

        S := "";
        while V[i] in ['0'..'9'] do begin
            S := S + V[i];
            i := i + 1; //read(c);
        end;

        if PNTType = operand then begin
            Val(S, Num, e);
            New(tmp);
            tmp^.operand := Num; tmp^.PNTType := operand;
            Add(tmp, p, PN);
        end;

        if {c}V[i] = '(' then begin

```



```

    New(tmp);
    tmp^.op := '(';
    tmp^.PNType := op;
    push(tmp, Stack);
end
else if {c}V[i] = ')' then begin
    tmp := pop(Stack);
    while tmp^.op <> '(' do begin
        Add(tmp, p, PN);
        tmp := pop(Stack);
    end;
end
else if ({c}V[i] in ['+', '-', '*', '/']) and (Stack = nil) then begin
    New(tmp);
    tmp^.op := V[i]{c}; tmp^.PNType := op;
    push(tmp, Stack);
end
else if (V[i]{c} in ['+', '-', '*', '/']) then begin
    pri1 := prioritet(V[i]{c});
    pri2 := prioritet(Stack^.op);

    while pri1 <= pri2 do begin
        tmp := pop(Stack);
        Add(tmp, p, PN);
        if Stack <> nil then
            pri2 := prioritet(Stack^.op)
        else pri2 := 0;
    end;
    New(tmp);
    tmp^.op := V[i]{c}; tmp^.PNType := op;
    push(tmp, Stack);
end;
end;

tmp := pop(Stack);
while tmp <> nil do begin
    New(p^.Next);
    p := p^.Next;
    p^.op := tmp^.op;
    p^.PNType := op;
    p^.Next := nil;
    tmp := pop(Stack);
end;
end;

function CalculateExpression(S: String): integer;
var
    PN: PNode;
begin
    Stack := nil;
    MakePN(PN, S);

    Stack := nil;

```

```
Result := Round(Calculate(PN));  
end;  
  
end.
```

ZAKLJUČAK

Kviz PC Slagalice je jednostavan za rukovanje. Ako ste ikad igledali kviz “**TV Slagalice**” neće vam biti potrebno nikakvo objašnjenje. Ako ipak niste gledali imate help u kome su detaljno objašnjena pravila kviza. Kviz je zanimljiv i odrađen je za dva igrača.

Izgled kviza nije isti kao i na pravog kvizu “TV Slagalice”, ali ima lepe boje i prijatne su za oči.

Imam u planu da za nekoliko nedelja napravim i veštačku inteligenciju za kviz, koja će omogućiti igru u Singleplayer modu.

Program je inače rađen u okruženju za brzi razvoj aplikacija Delphi koje je jedno od više okruženja koja se danas mogu naći na svetksom tržištu.

Kao dodatne programe koristio sam: PhotoShop 7.0, Adobe Premiere, Paint, 3Ds Max,...

LITERATURA

- Savić D. (1990) Turbo Pascal naredbe i objekti, izdavačko preduzeće PC Program, Beograd
- Bratko I, i Rajković V. (1984) Računarstvo sa programskim jezikom Paskal, izdavač Nolit Beograd
- Stephen O'Brien (1991) Turbo Pascal kompletan vodič, izdavač Mikro knjiga, Beograd.
- Kent R. (1999) Delphi 4, izdavač Kompjuterska biblioteka Čačak.
5. M. Cantu (2001) Delphi 5 detaljni izvornik, Kompjuterska biblioteka Čačak
- Osnove informatike i računarstva za 8. razred Miodrag Stojanović, Dragoljub Vasić
- Osnovi računarskih sistema Nedeljko S. Parezanović
- Informatika i Računarstvo za 1. razred srednje škole, Nikola Klem
- Pascal I (za matematičku gimnaziju) Milan Čabarkapa