|  |  |
| --- | --- |
| uses crt,graph,sysutils;  var  ej: array[1..20,1..30] of integer;  gd,gm,  i,j,  k,  a,  b,  x,y,  figure,  saj,sdzax,  delete\_row,  koxm,  score: integer;  ra,  rb,rx,ry: real;  ch: char;  poxel,sfgr,clear\_row: boolean; | Ej – այս փոփոխականում պահվում է խաղի քարտեզը  Gd,gm – փոփոխականներ գրաֆիկական ռեժիմը միացնելու համար  i, j,a – ցիկլերի պարամետրեր  k – ֆիգուրայի 4 տեսքերից մեկի ընտրության համար նախատեսված փոփոխական  x,y – ընկնող ֆիգուրայի կոորդինատները  figure – ֆիգուրայի համարը  saj,sdzax – աջ և ձախ կողմերից նախատեսված սահմանները  delete\_row – ջնջվող տողի համարը  koxm – ֆիգուրայի շարժման ուղղություն  score – խաղացողի միավորներ |
| procedure nkarel();  var i,j: integer;  begin  for i:=1 to 18 do  for j:=1 to 25 do begin  if ej[i,j]=1 then begin  setfillstyle(Solidfill,5);  bar(i\*20,j\*20,i\*20+15,j\*20+15); end;  end;  setfillstyle(emptyfill,green);  Rectangle(5,5,380,515);  end; | Խաղի քարտեզի արտապատկերման համար նախատեսված պրոցեդուրա |
| function qarakusi(x,y: integer) : boolean;  var x1,x2,x3,x4,y1,y2,y3,y4: integer;  begin  x1:=x;  x2:=x+1;  x3:=x;  x4:=x+1;  y1:=y;  y2:=y;  y3:=y+1;  y4:=y+1;  setfillstyle(solidfill,0);  case koxm of  1: begin bar((x1-1)\*20,y1\*20,(x1-1)\*20+15,y1\*20+15);bar((x3-1)\*20,y3\*20,(x3-1)\*20+15,y3\*20+15); end;  2: begin bar((x2+1)\*20,y2\*20,(x2+1)\*20+15,y2\*20+15);bar((x4+1)\*20,y4\*20,(x4+1)\*20+15,y4\*20+15); end;  3: begin bar((x1)\*20,(y1-1)\*20,(x1)\*20+15,(y1-1)\*20+15);bar((x2)\*20,(y2-1)\*20,(x2)\*20+15,(y2-1)\*20+15); end;  end;  setfillstyle(solidfill,6);  bar(x1\*20,y1\*20,x1\*20+15,y1\*20+15);  bar(x2\*20,y2\*20,x2\*20+15,y2\*20+15);  bar(x3\*20,y3\*20,x3\*20+15,y3\*20+15);  bar(x4\*20,y4\*20,x4\*20+15,y4\*20+15);  if (ej[x3,y3+1]=1) or (ej[x4,y4+1]=1) or (y=24) then begin  ej[x1,y1]:=1;  ej[x2,y2]:=1;  ej[x3,y3]:=1;  ej[x4,y4]:=1;  qarakusi:=true;  end  else  qarakusi:=false;  end; | Քառակուսի ֆիգուրայի արտապատկերման համար նախատեսված ֆունկցիա |
| function erankyun(x,y: integer) : boolean;  var x1,x2,x3,x4,y1,y2,y3,y4: integer;  begin  case k of  1: begin  x1:=x-1;  x2:=x;  x3:=x;  x4:=x+1;  y1:=y;  y2:=y;  y3:=y-1;  y4:=y;  setfillstyle(solidfill,0);  case koxm of  1: begin bar((x1-1)\*20,y1\*20,(x1-1)\*20+15,y1\*20+15);bar((x3-1)\*20,y3\*20,(x3-1)\*20+15,y3\*20+15); end;  2: begin bar((x3+1)\*20,y3\*20,(x3+1)\*20+15,y3\*20+15);bar((x4+1)\*20,y4\*20,(x4+1)\*20+15,y4\*20+15); end;  3: begin bar((x1)\*20,(y1-1)\*20,(x1)\*20+15,(y1-1)\*20+15);bar((x3)\*20,(y3-1)\*20,(x3)\*20+15,(y3-1)\*20+15);bar((x4)\*20,(y4-1)\*20,(x4)\*20+15,(y4-1)\*20+15); end;  end;  if sfgr then bar((x)\*20,(y+1)\*20,(x)\*20+15,(y+1)\*20+15);  setfillstyle(solidfill,6);  bar(x1\*20,y1\*20,x1\*20+15,y1\*20+15);  bar(x2\*20,y2\*20,x2\*20+15,y2\*20+15);  bar(x3\*20,y3\*20,x3\*20+15,y3\*20+15);  bar(x4\*20,y4\*20,x4\*20+15,y4\*20+15);  if (ej[x2,y2+1]=1) or (ej[x4,y4+1]=1) or (ej[x1,y1+1]=1) or (y=25) then begin  ej[x1,y1]:=1;  ej[x2,y2]:=1;  ej[x3,y3]:=1;  ej[x4,y4]:=1;  erankyun:=true;  end  else  erankyun:=false;  end;  2: begin  x1:=x;  x2:=x;  x3:=x;  x4:=x+1;  y1:=y+1;  y2:=y;  y3:=y-1;  y4:=y;  setfillstyle(solidfill,0);  case koxm of  1: begin bar((x1-1)\*20,y1\*20,(x1-1)\*20+15,y1\*20+15);bar((x2-1)\*20,y2\*20,(x2-1)\*20+15,y2\*20+15);bar((x3-1)\*20,y3\*20,(x3-1)\*20+15,y3\*20+15); end;  2: begin bar((x3+1)\*20,y3\*20,(x3+1)\*20+15,y3\*20+15);bar((x4+1)\*20,y4\*20,(x4+1)\*20+15,y4\*20+15);bar((x1+1)\*20,y1\*20,(x1+1)\*20+15,y1\*20+15); end;  3: begin bar((x4)\*20,(y4-1)\*20,(x4)\*20+15,(y4-1)\*20+15);bar((x3)\*20,(y3-1)\*20,(x3)\*20+15,(y3-1)\*20+15); end;  end;  if sfgr then bar((x-1)\*20,(y)\*20,(x-1)\*20+15,(y)\*20+15);  setfillstyle(solidfill,6);  bar(x1\*20,y1\*20,x1\*20+15,y1\*20+15);  bar(x2\*20,y2\*20,x2\*20+15,y2\*20+15);  bar(x3\*20,y3\*20,x3\*20+15,y3\*20+15);  bar(x4\*20,y4\*20,x4\*20+15,y4\*20+15);  if (ej[x2,y2+1]=1) or (ej[x4,y4+1]=1) or (ej[x1,y1+1]=1) or (y=24) then begin  ej[x1,y1]:=1;  ej[x2,y2]:=1;  ej[x3,y3]:=1;  ej[x4,y4]:=1;  erankyun:=true;  end  else  erankyun:=false;  end;  3: begin  x1:=x;  x2:=x;  x3:=x-1;  x4:=x+1;  y1:=y+1;  y2:=y;  y3:=y;  y4:=y;  setfillstyle(solidfill,0);  case koxm of  1: begin bar((x1-1)\*20,y1\*20,(x1-1)\*20+15,y1\*20+15);bar((x3-1)\*20,y3\*20,(x3-1)\*20+15,y3\*20+15); end;  2: begin bar((x4+1)\*20,y4\*20,(x4+1)\*20+15,y4\*20+15);bar((x1+1)\*20,y1\*20,(x1+1)\*20+15,y1\*20+15); end;  3: begin bar((x4)\*20,(y4-1)\*20,(x4)\*20+15,(y4-1)\*20+15);bar((x3)\*20,(y3-1)\*20,(x3)\*20+15,(y3-1)\*20+15);bar((x2)\*20,(y2-1)\*20,(x2)\*20+15,(y2-1)\*20+15); end;  end;  if sfgr then bar((x)\*20,(y-1)\*20,(x)\*20+15,(y-1)\*20+15);  setfillstyle(solidfill,6);  bar(x1\*20,y1\*20,x1\*20+15,y1\*20+15);  bar(x2\*20,y2\*20,x2\*20+15,y2\*20+15);  bar(x3\*20,y3\*20,x3\*20+15,y3\*20+15);  bar(x4\*20,y4\*20,x4\*20+15,y4\*20+15);  if (ej[x1,y1+1]=1) or (ej[x2,y2+1]=1) or (ej[x3,y3+1]=1) or (ej[x4,y4+1]=1) or (y=24) then begin  ej[x1,y1]:=1;  ej[x2,y2]:=1;  ej[x3,y3]:=1;  ej[x4,y4]:=1;  erankyun:=true;  end  else  erankyun:=false;  end;  4: begin  x1:=x;  x2:=x;  x3:=x-1;  x4:=x;  y1:=y+1;  y2:=y;  y3:=y;  y4:=y-1;  setfillstyle(solidfill,0);  case koxm of  1: begin bar((x1-1)\*20,y1\*20,(x1-1)\*20+15,y1\*20+15);bar((x3-1)\*20,y3\*20,(x3-1)\*20+15,y3\*20+15);bar((x4-1)\*20,y4\*20,(x4-1)\*20+15,y4\*20+15); end;  2: begin bar((x4+1)\*20,y4\*20,(x4+1)\*20+15,y4\*20+15);bar((x1+1)\*20,y1\*20,(x1+1)\*20+15,y1\*20+15);bar((x2+1)\*20,y2\*20,(x2+1)\*20+15,y2\*20+15); end;  3: begin bar((x4)\*20,(y4-1)\*20,(x4)\*20+15,(y4-1)\*20+15);bar((x3)\*20,(y3-1)\*20,(x3)\*20+15,(y3-1)\*20+15); end;  end;  if sfgr then bar((x+1)\*20,(y)\*20,(x+1)\*20+15,(y)\*20+15);  setfillstyle(solidfill,6);  bar(x1\*20,y1\*20,x1\*20+15,y1\*20+15);  bar(x2\*20,y2\*20,x2\*20+15,y2\*20+15);  bar(x3\*20,y3\*20,x3\*20+15,y3\*20+15);  bar(x4\*20,y4\*20,x4\*20+15,y4\*20+15);  if (ej[x1,y1+1]=1) or (ej[x2,y2+1]=1) or (ej[x3,y3+1]=1) or (ej[x4,y4+1]=1) or (y=24) then begin  ej[x1,y1]:=1;  ej[x2,y2]:=1;  ej[x3,y3]:=1;  ej[x4,y4]:=1;  erankyun:=true;  end  else  erankyun:=false;  end;  end;  end; | Եռանկյունաձև ֆիգուրայի արտապատկերման համար նախատեսված ֆունկցիա |
| function kosch(x,y: Integer) : boolean;  var x1,x2,x3,x4,y1,y2,y3,y4: integer;  begin  case k of  1: begin  x1:=x;  x2:=x+1;  x3:=x;  x4:=x;  y1:=y;  y2:=y;  y3:=y-1;  y4:=y-2;  setfillstyle(solidfill,0);  case koxm of  1: begin bar((x1-1)\*20,y1\*20,(x1-1)\*20+15,y1\*20+15);bar((x3-1)\*20,y3\*20,(x3-1)\*20+15,y3\*20+15);bar((x4-1)\*20,y4\*20,(x4-1)\*20+15,y4\*20+15); end;  2: begin bar((x4+1)\*20,y4\*20,(x4+1)\*20+15,y4\*20+15);bar((x3+1)\*20,y3\*20,(x3+1)\*20+15,y3\*20+15);bar((x2+1)\*20,y2\*20,(x2+1)\*20+15,y2\*20+15); end;  3: begin bar((x4)\*20,(y4-1)\*20,(x4)\*20+15,(y4-1)\*20+15);bar((x2)\*20,(y2-1)\*20,(x2)\*20+15,(y2-1)\*20+15); end;  end;  if sfgr then begin bar((x+2)\*20,(y-1)\*20,(x+2)\*20+15,(y-1)\*20+15);bar((x+2)\*20,(y)\*20,(x+2)\*20+15,(y)\*20+15); end;  setfillstyle(solidfill,6);  bar(x1\*20,y1\*20,x1\*20+15,y1\*20+15);  bar(x2\*20,y2\*20,x2\*20+15,y2\*20+15);  bar(x3\*20,y3\*20,x3\*20+15,y3\*20+15);  bar(x4\*20,y4\*20,x4\*20+15,y4\*20+15);  if (ej[x1,y1+1]=1) or (ej[x2,y2+1]=1) or (y=25) then begin  ej[x1,y1]:=1;  ej[x2,y2]:=1;  ej[x3,y3]:=1;  ej[x4,y4]:=1;  kosch:=true;  end  else  kosch:=false;  end;  2: begin  x1:=x;  x2:=x;  x3:=x+1;  x4:=x+2;  y1:=y;  y2:=y-1;  y3:=y-1;  y4:=y-1;  setfillstyle(solidfill,0);  case koxm of  1: begin bar((x1-1)\*20,y1\*20,(x1-1)\*20+15,y1\*20+15);bar((x2-1)\*20,y2\*20,(x2-1)\*20+15,y2\*20+15); end;  2: begin bar((x4+1)\*20,y4\*20,(x4+1)\*20+15,y4\*20+15);bar((x1+1)\*20,y1\*20,(x1+1)\*20+15,y1\*20+15); end;  3: begin bar((x4)\*20,(y4-1)\*20,(x4)\*20+15,(y4-1)\*20+15);bar((x3)\*20,(y3-1)\*20,(x3)\*20+15,(y3-1)\*20+15);bar((x2)\*20,(y2-1)\*20,(x2)\*20+15,(y2-1)\*20+15); end;  end;  if sfgr then begin bar((x)\*20,(y-2)\*20,(x)\*20+15,(y-2)\*20+15);bar((x+1)\*20,(y)\*20,(x+1)\*20+15,(y)\*20+15); end;  setfillstyle(solidfill,6);  bar(x1\*20,y1\*20,x1\*20+15,y1\*20+15);  bar(x2\*20,y2\*20,x2\*20+15,y2\*20+15);  bar(x3\*20,y3\*20,x3\*20+15,y3\*20+15);  bar(x4\*20,y4\*20,x4\*20+15,y4\*20+15);  if (ej[x1,y1+1]=1) or (ej[x3,y3+1]=1) or (ej[x4,y4+1]=1) or (y=25) then begin  ej[x1,y1]:=1;  ej[x2,y2]:=1;  ej[x3,y3]:=1;  ej[x4,y4]:=1;  kosch:=true;  end  else  kosch:=false;  end;  3: begin  x1:=x;  x2:=x+1;  x3:=x+1;  x4:=x+1;  y1:=y;  y2:=y;  y3:=y+1;  y4:=y+2;  setfillstyle(solidfill,0);  case koxm of  1: begin bar((x1-1)\*20,y1\*20,(x1-1)\*20+15,y1\*20+15);bar((x3-1)\*20,y3\*20,(x3-1)\*20+15,y3\*20+15);bar((x4-1)\*20,y4\*20,(x4-1)\*20+15,y4\*20+15); end;  2: begin bar((x4+1)\*20,y4\*20,(x4+1)\*20+15,y4\*20+15);bar((x3+1)\*20,y3\*20,(x3+1)\*20+15,y3\*20+15);bar((x2+1)\*20,y2\*20,(x2+1)\*20+15,y2\*20+15); end;  3: begin bar((x1)\*20,(y1-1)\*20,(x1)\*20+15,(y1-1)\*20+15);bar((x2)\*20,(y2-1)\*20,(x2)\*20+15,(y2-1)\*20+15); end;  end;  if sfgr then begin bar((x+2)\*20,(y-1)\*20,(x+2)\*20+15,(y-1)\*20+15);bar((x+1)\*20,(y)\*20,(x+1)\*20+15,(y)\*20+15); bar((x)\*20,(y)\*20,(x)\*20+15,(y)\*20+15); end;  setfillstyle(solidfill,6);  bar(x1\*20,y1\*20,x1\*20+15,y1\*20+15);  bar(x2\*20,y2\*20,x2\*20+15,y2\*20+15);  bar(x3\*20,y3\*20,x3\*20+15,y3\*20+15);  bar(x4\*20,y4\*20,x4\*20+15,y4\*20+15);  if (ej[x1,y1+1]=1) or (ej[x4,y4+1]=1) or (y=23) then begin  ej[x1,y1]:=1;  ej[x2,y2]:=1;  ej[x3,y3]:=1;  ej[x4,y4]:=1;  kosch:=true;  end  else  kosch:=false;  end;  4: begin  x1:=x;  x2:=x+1;  x3:=x+2;  x4:=x+2;  y1:=y;  y2:=y;  y3:=y;  y4:=y-1;  setfillstyle(solidfill,0);  case koxm of  1: begin bar((x1-1)\*20,y1\*20,(x1-1)\*20+15,y1\*20+15);bar((x4-1)\*20,y4\*20,(x4-1)\*20+15,y4\*20+15); end;  2: begin bar((x4+1)\*20,y4\*20,(x4+1)\*20+15,y4\*20+15);bar((x3+1)\*20,y3\*20,(x3+1)\*20+15,y3\*20+15); end;  3: begin bar((x1)\*20,(y1-1)\*20,(x1)\*20+15,(y1-1)\*20+15);bar((x2)\*20,(y2-1)\*20,(x2)\*20+15,(y2-1)\*20+15);bar((x4)\*20,(y4-1)\*20,(x4)\*20+15,(y4-1)\*20+15); end;  end;  if sfgr then begin bar((x+1)\*20,(y+2)\*20,(x+1)\*20+15,(y+2)\*20+15);bar((x+1)\*20,(y+1)\*20,(x+1)\*20+15,(y+1)\*20+15); end;  setfillstyle(solidfill,6);  bar(x1\*20,y1\*20,x1\*20+15,y1\*20+15);  bar(x2\*20,y2\*20,x2\*20+15,y2\*20+15);  bar(x3\*20,y3\*20,x3\*20+15,y3\*20+15);  bar(x4\*20,y4\*20,x4\*20+15,y4\*20+15);  if (ej[x1,y1+1]=1) or (ej[x4,y4+1]=1) or (y=25) then begin  ej[x1,y1]:=1;  ej[x2,y2]:=1;  ej[x3,y3]:=1;  ej[x4,y4]:=1;  kosch:=true;  end  else  kosch:=false;  end;  end;  end; | Լ – ձև ֆիգուրայի համար նախատեսված ֆունկցիա |
| procedure verify\_LR(x,y,figure:integer) ;  var x1,x2,x3,x4,y1,y2,y3,y4: integer;  begin  case figure of  0: begin  sdzax:=1;  saj:=17;  x1:=x;  x2:=x+1;  x3:=x;  x4:=x+1;  y1:=y;  y2:=y;  y3:=y+1;  y4:=y+1;  if x1>1 then  if (ej[x1-1,y1]=1) or (ej[x3-1,y3]=1) then sdzax:=x1+1;  if (ej[x2+1,y2]=1) or (ej[x4+1,y4]=1) then saj:=x2-1;  end;  1: begin {Erankyun}  case k of  1: begin  if x=1 then x:=2;  sdzax:=2;  saj:=17;  x1:=x-1;  x2:=x;  x3:=x;  x4:=x+1;  y1:=y;  y2:=y;  y3:=y-1;  y4:=y;  if x1>1 then  if ej[x1-1,y1]=1 then sdzax:=x1+1;  if ej[x4+1,y4]=1 then saj:=x4-1;  end;  2: begin  sdzax:=1;  saj:=17;  x1:=x;  x2:=x;  x3:=x;  x4:=x+1;  y1:=y+1;  y2:=y;  y3:=y-1;  y4:=y;  if x1>1 then  if (ej[x1-1,y1]=1) or (ej[x2-1,y2]=1) or (ej[x3-1,y3]=1) then sdzax:=x1+1;  if ej[x4+1,y4]=1 then saj:=x4-1;  end;  3: begin  if x=1 then x:=2;  sdzax:=2;  saj:=17;  x1:=x;  x2:=x;  x3:=x-1;  x4:=x+1;  y1:=y+1;  y2:=y;  y3:=y;  y4:=y;  if x3>1 then  if ej[x3-1,y3]=1 then sdzax:=x3+1;  if ej[x4+1,y4]=1 then saj:=x4-1;  end;  4: begin  if x=1 then x:=2;  sdzax:=2;  saj:=18;  x1:=x;  x2:=x;  x3:=x-1;  x4:=x;  y1:=y+1;  y2:=y;  y3:=y;  y4:=y-1;  if x3>1 then  if ej[x3-1,y3]=1 then sdzax:=x3+1;  if (ej[x1+1,y1]=1) or (ej[x2+1,y2]=1) or (ej[x4+1,y4]=1) then saj:=x1-1;  end;  end;  end;  2: begin {Kosh}  case k of  1: begin  sdzax:=1;  saj:=17;  x1:=x;  x2:=x+1;  x3:=x;  x4:=x;  y1:=y;  y2:=y;  y3:=y-1;  y4:=y-2;  if (x1>1) and (y4>1) then  if (ej[x1-1,y1]=1) or (ej[x3-1,y3]=1) or (ej[x4-1,y4]=1) then sdzax:=x1+1;  if ej[x2+1,y2]=1 then saj:=x2-1;  end;  2: begin  sdzax:=1;  saj:=16;  x1:=x;  x2:=x;  x3:=x+1;  x4:=x+2;  y1:=y;  y2:=y-1;  y3:=y-1;  y4:=y-1;  if x1>1 then  if (ej[x1-1,y1]=1) or (ej[x2-1,y2]=1) then sdzax:=x2+1;  if ej[x4+1,y4]=1 then saj:=x4-2;  end;  3: begin  sdzax:=1;  saj:=17;  x1:=x;  x2:=x+1;  x3:=x+1;  x4:=x+1;  y1:=y;  y2:=y;  y3:=y+1;  y4:=y+2;  if x1>1 then  if ej[x1-1,y1]=1 then sdzax:=x1+1;  if(ej[x2+1,y2]=1) or (ej[x3+1,y3]=1) or (ej[x4+1,y4]=1) then saj:=x4-1;  end;  4: begin  sdzax:=1;  saj:=16;  x1:=x;  x2:=x+1;  x3:=x+2;  x4:=x+2;  y1:=y;  y2:=y;  y3:=y;  y4:=y-1;  if x1>1 then  if ej[x1-1,y1]=1 then sdzax:=x1+1;  if (ej[x3+1,y3]=1) or (ej[x4+1,y4]=1) then sdzax:=x4-2;  end;  end;  end;  end;  end; | Աջ և ձախ կողմերից ֆիգուրայի համար նախատեսվող սահմանների վորոշման ֆունկցիա |
| procedure spusk();  begin  case figure of  0: begin poxel:=qarakusi(x,y); end;  1: begin poxel:=erankyun(x,y); end;  2: begin poxel:=kosch(x,y); end;  end;  end; | Շարժվող ֆիգուրայի ֆունկցիայի վորոշման պրոցեդուրա |
| procedure onkeypressfin();  begin  if keypressed then begin  ch:=readkey;  verify\_LR(x,y,figure);  if ch=#77 then if x<saj then begin x:=x+1; koxm:=1; end;  if ch=#75 then if x>sdzax then begin x:=x-1; koxm:=2; end;  if ch=#27 then halt;  if ch=#72 then begin  if k<4 then k:=k+1 else k:=1;  sfgr:=true;  end;  if (ch=#80) and (y<25) then y:=y+1;  spusk();  sfgr:=false;  end;  end; | Այս պրոցեդուրան նախատեսված է ստեղն սխմելիս նրա համար նախատեսված ֆունկցիայի կատարման համար |
| function verify\_rows() : boolean;  var ii,jj,kk: integer;  begin  if clear\_row=false then  for jj:=1 to delete\_row-1 do  for ii:=1 to 18 do begin  ej[ii,delete\_row-jj+1]:=ej[ii,delete\_row-jj];  end;  for ii:=1 to 25 do begin  kk:=0;  for jj:=1 to 18 do  if ej[jj,ii]=1 then kk:=kk+1;  if kk=18 then begin verify\_rows:=false; delete\_row:=ii; break; end else verify\_rows:=true;  end;  end; | Ստուգում է տողերը |
| procedure delete\_rows();  var iii: integer;  begin  for iii:=1 to 18 do begin  setfillstyle(solidfill,0);  ej[iii,delete\_row]:=0;  bar(iii\*20,delete\_row\*20,iii\*20+15,delete\_row\*20+15);  delay(50);  end;  setfillstyle(solidfill,0);  bar(0,0,1000,1000);  score:=score+1;  end; | Լրացված տողերի ջնջման համար նախատեսված պրոցեդուրա |
| Begin  a:=0; | Ամբողջ ծրագրի սկիզբ  a փոփոխականին վերագրվում է 0 |
| k:=0; | K փոփոխականին վերագրվում է 0 |
| score:=0; | Score փոփոխականին վերագրվում է 0 |
| randomize; |  |
| gd:=detect;  initgraph(gd,gm,'..\bgi'); | Գրաֆիկական ռեժիմի միացում |
| saj:=17; | Saj փոփոխականին վերագրվում է 17 արժեք |
| sdzax:=1; | sdzax փոփոխականին վերագրվում է 1 արժեք |
| while true do begin | Անվերջ ցիկլի սկիզբ, որը կվերջանա Esc ստեղնը սխմելու դեպքում |
| figure:=random(3); | figure փոփոխականին վերագրվում է 0-2 ինչ-որ արժեք |
| x:=12; | x փոփոխականին վերագրվում է 12 արժեք |
| y:=1; | y փոփոխականին վերագրվում է 1 արժեք |
| k:=1; | k փոփոխականին վերագրվում է 1 արժեք |
| verify\_LR(x,y,figure); | verify\_LR պրոցեդուրայի կանչ x,y,figure պարամետրերով |
| setfillstyle(solidfill,0); | Ուղղանկյուն նկարելու ձևը տվող պրոցեդուրա |
| bar(0,0,1000,1000); | Գծում է 1000 պիկսել կողմ ունեցող քառակուսի 0,0 սկզբնական կետից |
| nkarel(); | Արտածում է էկրանին խաղի քարտեզը |
| while y<25 do begin | Ցիկլ y<25 պայմանով |
| verify\_LR(x,y,figure); | verify\_LR պրոցեդուրայի կանչ x,y,figure պարամետրերով |
| if (a<100) and (poxel=false) then begin | Պայման, եթե a<100 և poxel=false, ապա |
| a:=a+1; | a փոփոխականը ավելանում է 1 արժեքով |
| verify\_LR(x,y,figure); | verify\_LR պրոցեդուրայի կանչ x,y,figure պարամետրերով |
| onkeypressfin(); | Ինչ-որ ստեղնի սխման դեպքում կատարվում է այս պրոցեդուրան |
| if ej[x,y+1]=1 then break; | Եթե ej[x,y+1]=1 ապա ծրագիրը դուրս է գալիս ցիկլից |
| delay(1); end | Սպասում է 1 միլիվայրկյան |
| Else begin | a<100 և poxel=false պայմանի հակառակ դեպում կկատարվի ներքևը նշված մի քանի տողը |
| poxel:=false; | Poxel փոփոխականին վերագրվում է false արժեք |
| koxm:=3; | koxm փոփոխականին վերագրվում է 3 արժեք |
| y:=y+1; | y փոփոխականի արժեքը ավելանում է 1-ով |
| a:=0; | a փոփոխականին վերագրվում է 3 արժեք |
| nkarel(); | Արտածում է էկրանին խաղի քարտեզը |
| spusk(); | Նկարում է տվյալ պահին իջնող ֆիգուրան |
| end; | Պայմանի ավարտ |
| outtextxy(590,100,inttostr(score)); | 590,100 կոորդինատներով էկրանին տպում է խաղացողի ջնջած տողերի քանակը |
| if poxel then break; | Եթե poxel=true, ապա ծրագիրը դուրս է գալիս ցիկլից |
| end; | y<25 պայմանով ցիկլի ավարտ |
| repeat  clear\_row:=verify\_rows(); | Ցիկլի սկիզբ  clear\_row փոփոխականին վերգրում է verify\_rows() ֆունկցիայի կատարած աշխատանքի վերադարձրած արդյունքը |
| if clear\_row=false then Begin | Եթե clear\_row=false ապա |
| delete\_rows(); | Ջնջում է լցված տողը |
| nkarel(); | Նկարում է ստացված քարտեզը |
| end; | Պայմանի ավարտ |
| until clear\_row; | Ցիկլի ավարտ եթե clear\_row=true |
| end; | Ցիկլի ավարտ |
| End. | Ընդհանուր ծրագրի ավարտ |