Five Games　5つのゲーム

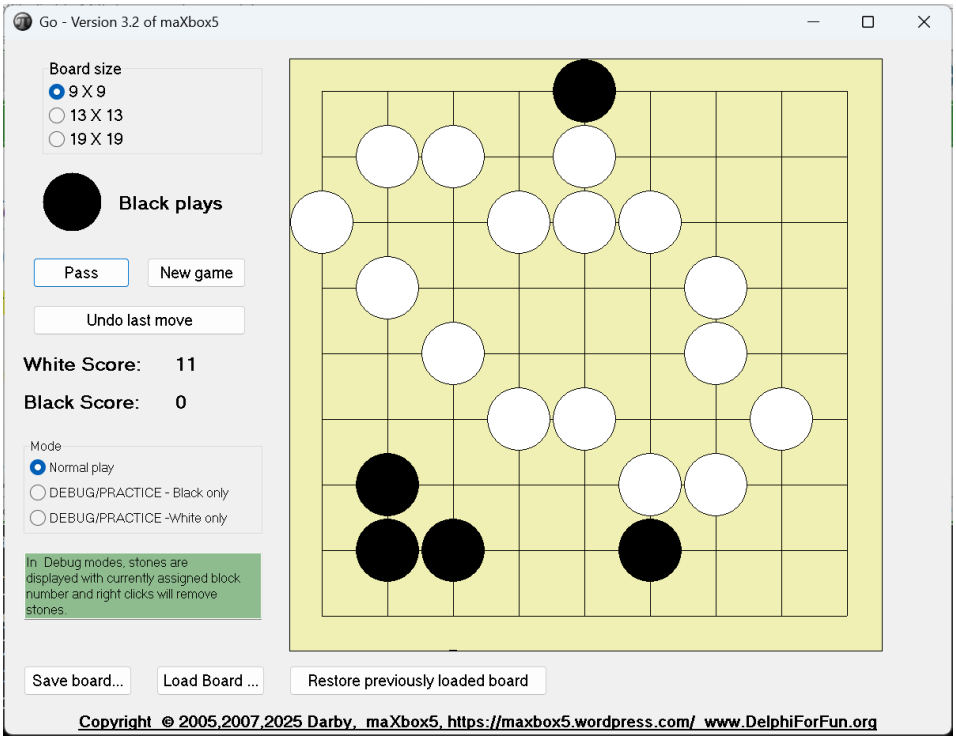
maXbox Starter 156 – 5 Games Showcase.

Source: Multi Code – Code Blog

Script_: Download 1435_5_games_showcase.txt (maXbox5)

Multi-game programming refers to the practice of integrating multiple games, models or game engines within a single system or application. This approach is particularly useful in scenarios where different games or boards full fill at specific tasks.
We check five games: Go – Chess – Othello – PacMan – 4Gewinnt (Fantastic4) - **GCOP4**

Go is a strategic board game for two players, where the objective is to control territory on the board by placing stones.



1435_mX5GuiScreenshot_GO_2025-09-27100346.png

The advantages of go-code (or multi-player-mode) are:

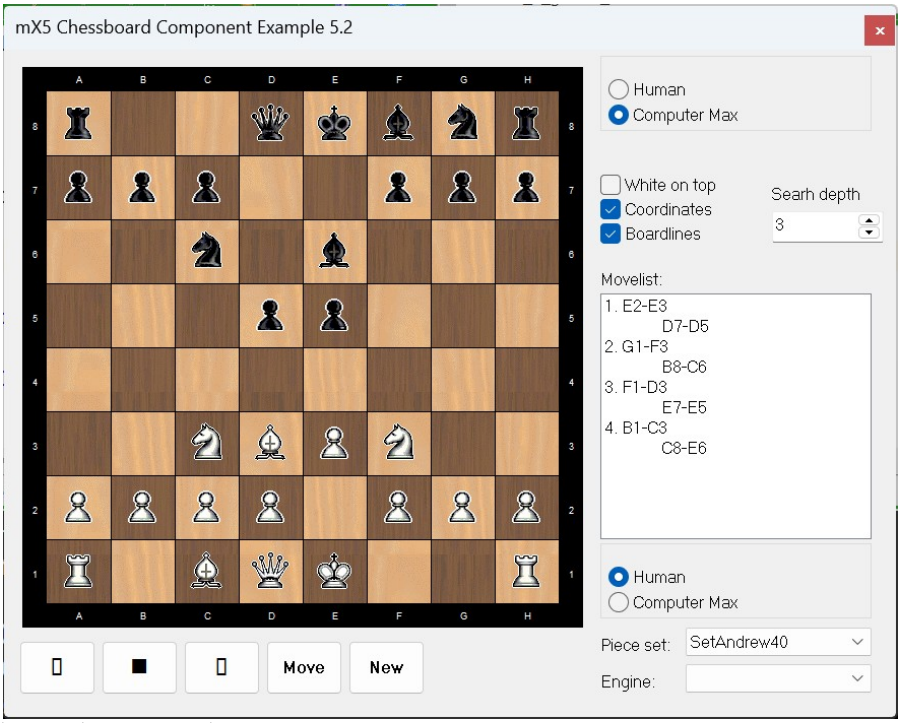
- **Optimized Performance**: Each game strategy or situation is used where it performs best.
  - **Flexibility**: Adapts to diverse project requirements.
  - **Scalability**: Supports complex, large-scale systems.
  - **Trainability**: more players learn from each other

The **Chessboard** component v3.03 for Delphi 2-7 and C++ Builder (or v 2.11 for C++ Builder only), is now available. The free Chessboard component (upgrade to 64-bit done) provides a 2-Dimensional chessboard with a drag and drop interface that can keep track of a game and allows full customization of the bitmaps for the pieces, squares and border.

Optionally the board can be resized at runtime. All common chess events are implemented (OnLegalMove, OnIllegalMove, OnCapture, OnCheck, OnMate, OnStaleMate, OnDraw etc.)

```
procedure startChessonBoard;
var chessbrd: TChessBrd;
 begin
  chessbrd:= TChessBrd.create;
   with chessbrd do begin
    top:= 50; left:= 290;
    //parent:= comfrm;   need to define a Form
    searchDepth:= 3;
    NewGame;
    ComputerPlaysBlack:=true;
    //onpromotion
    //ClearSquare(e_2);
    //borderbitmap:= getBitmapObject(Exepath+BACKMAP);
   end;
 end;
```

Now the component includes the engine of Tom's Simple Chess Program and calculates using its own thread. Optionally it allows you to use your own custom engine instead as well. Just drop a Chessbrd component on a form and you are very close to a complete multithreaded chess application. A Delphi example project has been included to demonstrate the common features (maxform1.chess41click(self).



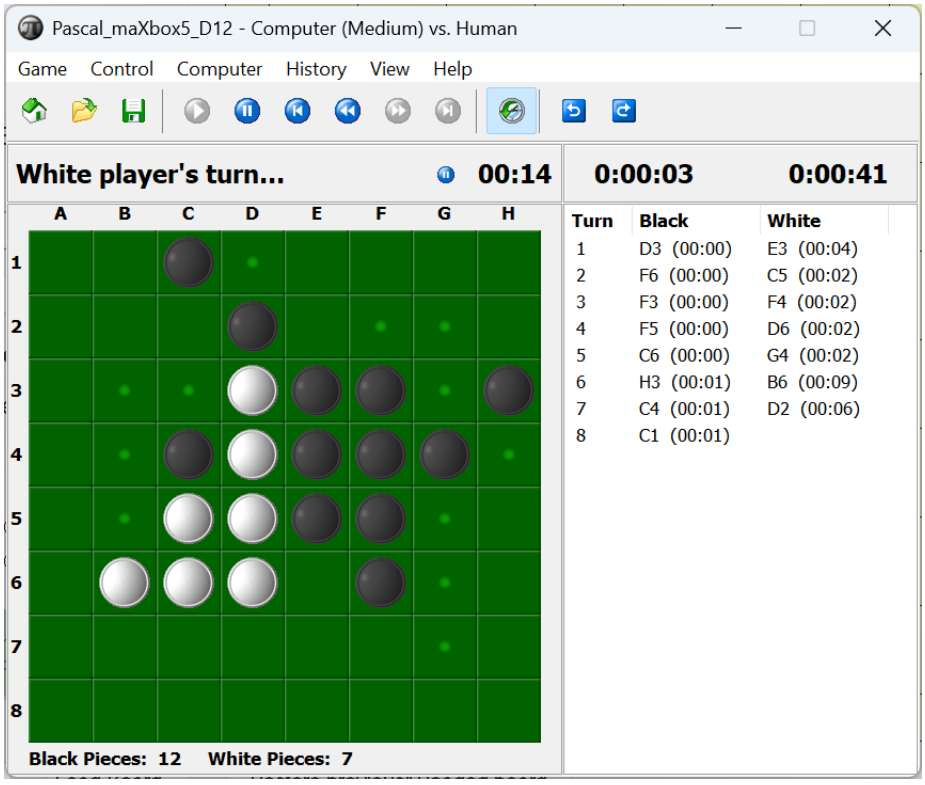
\menu\Options\Chess V52

To save and load a board in its simplest code we propose a *Stringlist* as a dataprovider with a matrix structure (double array):

```
Const
  BoardDimension = 8;
  BoardFileName = 'S8Board.txt';

Type
  TBoard = Array [1 .. BoardDimension] of array[1 .. BoardDimension] Of String;

procedure SaveBoard(Board: TBoard);
var
  i, j: Integer;
  Line, BoardFile: TStringList;
 begin
  BoardFile:= TStringList.Create;
  Line:= TStringList.Create;

  for i:= 1 to BoardDimension do begin
    Line.Clear;
    for j:= 1 to BoardDimension do
      Line.Add(Board[i][j]);
    BoardFile.Add(Line.CommaText+CRLF);
  end;

  Line.Free;
  BoardFile.SaveToFile(exepath+BoardFileName);
  BoardFile.Free;
 end;
```

It works with: Delphi version 6.0 and needs the Library SysUtils and StdCtrls, in maXbox there are precompiled on board.

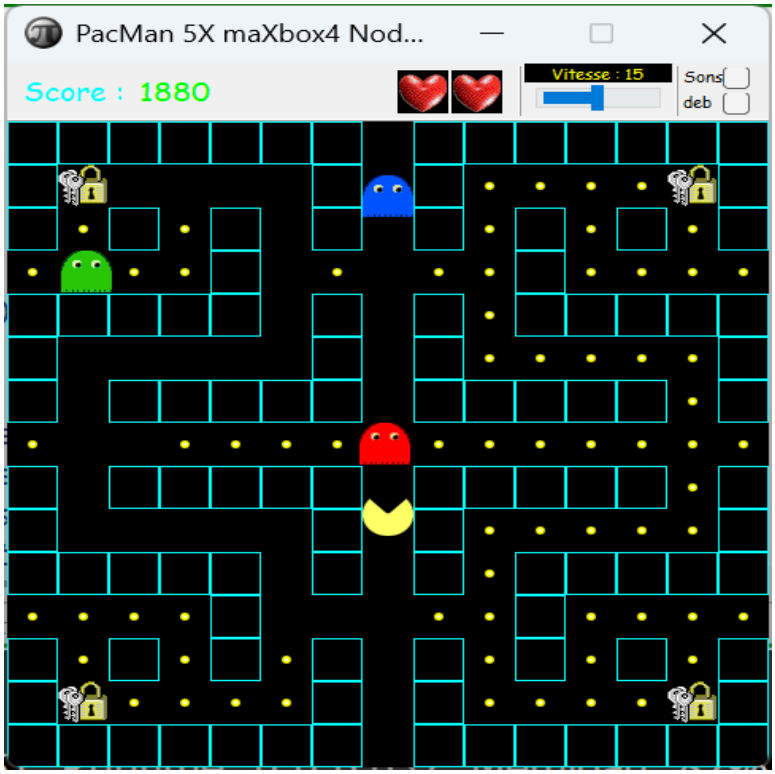
The third one is **Othello** and runs almost everywhere.



Reversi (also known as Othello) is a pretty simple game. It consists of a 8x8 square board, and pieces with one black and one white side. Each player has a colour, and the aim of the game is to get more of your pieces on the board than opponent's pieces(maxform1.reversi1click(self);

```
procedure startOthello;
begin
 with TOthelloMainForm.create(self) do  begin
    //formshow(self)
    formCreate(self);
    Icon.LoadFromResourceName(HInstance,'XDAOthello'); //NEWREPORT');
    caption:= 'call precompiled programmable from script host';
    //ViewFullScreen;
    writ(itoa(game.pieces[_D5]));
    //writ(itoa(game.board.pieces[d5]));
    writeln('PiecesCount: '+(PiecesCount.caption));
    showmodal
    free
  end;
end;
```

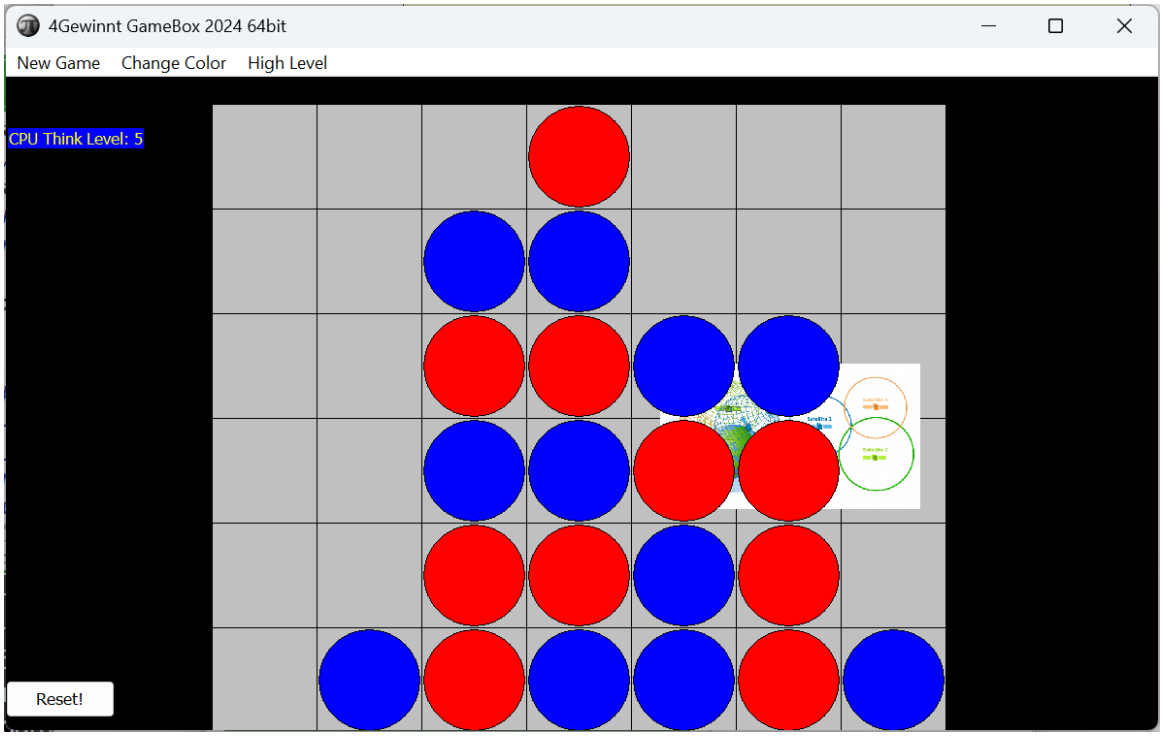
The next one is **PacMan** and runs also as Java Script and will also run in a browser but a local based one. With WebView2 in Microsoft Edge, developers can now embed web content directly within native Win apps. WebView2 offers a simple way to display web-based content using the Chromium engine inside your apps or scripts.
Specifically, WebView2 can run HTML, CSS, and JavaScript inside a standard non-web program, like something written in Visual C++, Delphi or .NET apps.
Openweb('https://raw.githack.com/breitsch2/maXbox4/master/assets/pacman2/pacman.html');



maxform1.PacManX51Click(self);

```
procedure JS_Solution;
begin
   with TEdgeViewForm.create(self) do begin
        width:= 1500;
        pagecontrol1.height:= 540;
        icon.loadfromresourcename(hinstance,'XJICON');
        Caption:= 'maXbox5 EdgeView2BrowserFeed_JScriptSolution' ;
        panel1.ParentBackground:= false;
        panel1.color:= clnavy;
        sleep(500);
      navigate('https://raw.githack.com/breitsch2/maXbox4/master/assets/
pacman2/pacman.html');
        // or local store
        memoJavaScript.text:= JSCRIPT_PACMAN;//JS_InitFunc;
          PageControl1.ActivePageIndex := 1;
          memoHTML.font.size:= 26;
          memoJavaScript.wordwrap:= true;
          memoJavaScript.lines.linebreak:= lf;
        tabsheet3.caption:= 'Result';
        panel3.caption:= 'Source: Result';
        ExecuteScript(JSCRIPT_PACMAN);
        sleep(1600);
        writeln('debug result of js webview2: '+memoHTML.text);
        writeln('scriptresult of js webview2: '+getScriptresult);
        showmodal;    //save resources
        free;
     end;
end;
```

The last in our showcase is my personal favorite, cause it implements a mini-max algorithm and its quite smart and effective too:

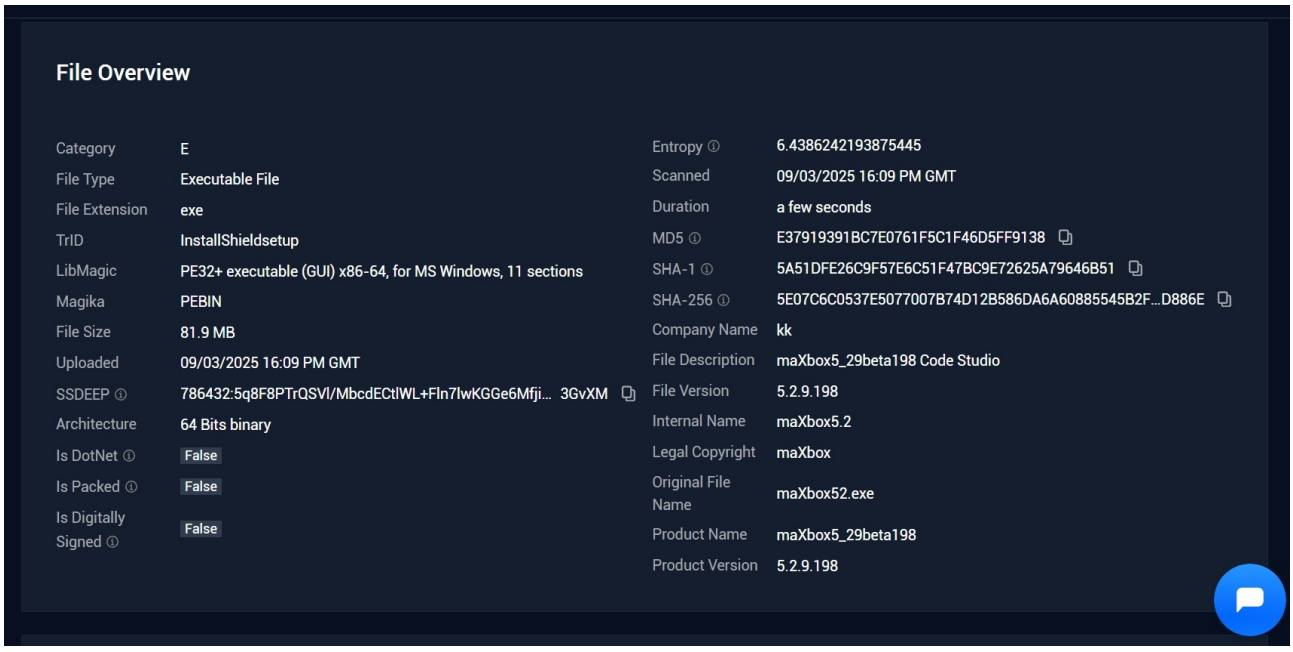



```
pic: TEdgeViewForm.create(self) do begin maxform1.N4GewinntGame1Click(self)
```

The Minimax algorithm is a decision-making algorithm used in game theory and artificial intelligence to determine the optimal move for a player, assuming that the opponent also plays optimally. It is widely used in two-player turn-based games such as Tic-Tac-Toe, Chess, and Backgammon. Mini-Max algorithm is a decision-making algorithm used in artificial intelligence, particularly in game theory and computer games. It is designed to minimize the possible loss in a worst-case scenario (hence "min") and maximize the potential gain (therefore "max").
The challenges are:

- Complexity: Managing multiple games can increase development and debugging effort.
- Interoperability Issues: Ensuring smooth communication between components and so called game engines.
- Learning Curve: Requires expertise in multiple programming paradigms.

Facilitates communication between different languages or frameworks, often through APIs, shared memory, or intermediate representations.



References:
http://www.delphiforfun.org.ws034.alentus.com/Programs/Go.htm

Pas2JS Examples:
https://raw.githack.com/breitsch2/maXbox4/master/assets/tetris2.html
https://raw.githack.com/breitsch2/maXbox4/master/assets/pacman2/pacman.html

How to Use WebView2 in Microsoft Edge Browser - GeeksforGeeks
Try It Online

Doc and Tool: maXbox5 - Manage Files at SourceForge.net
Release maXbox V5.2.9 · maxkleiner/maXbox5

Max Kleiner 07/09/2025

maXbox4exe
Digitally signed by maXbox4exe
Date: 2025.09.07 14:31:43 +02'00'