

```

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
  Graphics, Controls, Forms, Dialogs, IdBaseComponent,
  IdComponent, IdTCPServer, StdCtrls;

type
  TForm1 = class(TForm)
    TCPServer: TIdTCPServer;
    btStart: TButton;
    btStop: TButton;
    Memo1: TMemo;
    procedure btStartClick(Sender: TObject);
    procedure btStopClick(Sender: TObject);
    procedure TCPServerConnect(AThread: TIdPeerThread);
    procedure TCPServerDisconnect(AThread: TIdPeerThread);
    procedure TCPServerNoCommandHandler(ASender: TIdTCPServer;
      const AData: String; AThread: TIdPeerThread);
    procedure TCPServerMessagesFromBoard(rec, msgtype: Char; msg: String);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  DPlayerID,
  FPlayerID,
  DViewerID,
  FViewerID: Cardinal;
  PlayerID: Char;
  MSGType: String;
  // vars for testing purposes
  NoPlayers : Integer;

implementation

{$R *.dfm}

function NotRegistered(id: Cardinal): Boolean;
begin
  if DPlayerID = id then result := False
  else if FPlayerID = id then result := False
  else if DViewerID = id then result := False
  else if FViewerID = id then result := False
  else result := True;
end;

procedure TForm1.btStartClick(Sender: TObject);
begin
  TCPServer.Active := True;
  NoPlayers := 0;
end;

```

```

procedure TForm1.btStopClick(Sender: TObject);
begin
    TCPServer.Active := False;
end;

procedure TForm1.TCPServerConnect(AThread: TIdPeerThread);
begin
    AThread.Connection.WriteLn('Connected_to_server!');
end;

procedure TForm1.TCPServerDisconnect(AThread: TIdPeerThread);
begin
    AThread.Connection.WriteLn('Disconnected_from_server!_Have_a_nice_life ...');
end;

procedure TForm1.TCPServerNoCommandHandler(ASender: TIdTCPServer;
    const AData: String; AThread: TIdPeerThread);
begin
    // debugging in memo ^_^
    Memo1.Lines.Add('INCOMING:_ ' + AData);
    Memo1.Lines.Add('FROM:_ ' + inttostr(AThread.ThreadID));
    Memo1.Lines.Add(' ');

    // message type is determined by the first three characters
    MSGType := AData[1] + AData[2] + AData[3];

    // incoming message from P, D or F
    case AData[1] of

        // new player join request
        'P' : begin
            if (AData = 'PBJ') and NotRegistered(AThread.ThreadID) then
                begin
                    PlayerID := Join();
                    Case PlayerID of
                        'F' : begin
                            FPlayerID := AThread.ThreadID;
                            AThread.Connection.WriteLn('BPT_F');
                        end;
                        'D' : begin
                            DPlayerID := AThread.ThreadID;
                            AThread.Connection.WriteLn('BPT_D');
                        end;
                        'R' : begin
                            AThread.Connection.WriteLn('BPT_R');
                        end;
                    end;
                end;
            end;
        end; // end new player messages

    // messages from Dolphin player
    'D' : begin
        if AThread.ThreadID = DPlayerID then
            begin
                if MSGType = 'DBM' then
                    begin

```

```

        TryMove('D', strtoint(AData[5]), strtoint(AData[7]),
                strtoint(AData[9]), strtoint(AData[11]));
    end
    else if MSGType = 'DBR' then
    begin
        TCPServerMessagesFromBoard('E', 'B', UpdateViews)
    end
    end;
end; // end messages from D

// messages from Fox player (almost identical to dolphin player)
'F' : begin
    if AThread.ThreadID = DPlayerID then
    begin
        if MSGType = 'FBM' then
        begin
            TryMove('F', strtoint(AData[5]), strtoint(AData[7]),
                    strtoint(AData[9]), strtoint(AData[11]));

        end
        else if MSGType = 'FBR' then
        begin
            TCPServerMessagesFromBoard('G', 'B', UpdateViews)
        end
        end;
    end; // end messages from F
end; // end case
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    Memo1.Clear;
end;

procedure TForm1.TCPServerMessagesFromBoard(rec, msgtype: Char; msg: String);
var clients: TList;
    i: Integer;
    recID: Cardinal;
begin
    // built list with all connected clients
    clients:=TCPServer.Threads.LockList;

    // check which client we need to contact
    case rec of
        'D' : recid := DPlayerID;
        'E' : recid := DViewerID;
        'F' : recid := FPlayerID;
        'G' : recid := FViewerID
    end;

    // now find the threadid belonging to that client
    for i:=0 to clients.Count - 1 do
    begin
        if TIDPeerThread(clients[i]).ThreadID = recid then
        begin
            case msgtype of
                // and send what we want to sent,

```

```

        // only messages of type B and S need extra information
        'B' : TIdPeerThread(clients[i]).Connection.WriteLn('B' + rec + 'B' + msg);
        'F' : TIdPeerThread(clients[i]).Connection.WriteLn('B' + rec + 'F');
        'N' : TIdPeerThread(clients[i]).Connection.WriteLn('B' + rec + 'N');
        'S' : TIdPeerThread(clients[i]).Connection.WriteLn('B' + rec + 'S' + msg);
    end;
end;
// needs to be done apparently
TCPServer.Threads.UnlockList;
end;

```