DICEwin: DICE for Windows

Dietmar G. Schrausser

2023

Overview

Quasi random number generator via simulated dice roll. Application for MS Windows (Schrausser, 2023).

C++ Source of main functions

```
//------------------------------------------------------------------------| DICEDlg.cpp
//
//                                                                        | DICE,
Zufallswürfelzahl (deutsch)            //
//                                                                        | von Dietmar
Schrausser, (C) SCHRAUSSER 2011  //
//
#include "stdafx.h"
#include "DICE.h"
#include "DICEDlg.h"
#include "DICEeinst.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <math.h>

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

int dlg_x, dlg_y, x_=0, y_1=0, y_2=0, stp=1, logsw=0;
int  fb_hg,fb_hg_0,
        fb_wrf,fb_wrf_0,
        fb_ag,fb_ag_0,
        fb_zhl,fb_zhl_0,
        vms_, vms_0,
        von_, von_0,
        vwl_, vwl_0,
        z_frm,z_frm_0,
        sw_lg,
        ini_sw;

double dy_;
double SIGMA = 34.0/45;
double fn_erg;


//--------------------------------------------------------| variableninitialisierung About
About::About(): CDialog(About::IDD)
{
       //{{AFX_DATA_INIT(About)
       //}}AFX_DATA_INIT
}

//--------------------------------------------------------| steuerelement-initialisierung
About
void About::DoDataExchange(CDataExchange* pDX)
{
       CDialog::DoDataExchange(pDX);
       //{{AFX_DATA_MAP(About)
       //}}AFX_DATA_MAP
}
```

```
//--------------------------------------------------------| ereignis-deklaration About
BEGIN_MESSAGE_MAP(About, CDialog)
        //{{AFX_MSG_MAP(About)
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()


//--------------------------------------------------------| variableninitialisierung CDICEDlg
CDICEDlg::CDICEDlg(CWnd* pParent )
        : CDialog(CDICEDlg::IDD, pParent)
{
        //{{AFX_DATA_INIT(CDICEDlg)
        //}}AFX_DATA_INIT

        m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
        m_Csr1  = AfxGetApp()->LoadCursor(IDC_CURSOR1);
        m_Csr2  = AfxGetApp()->LoadCursor(IDC_CURSOR2);
        m_Csr3  = AfxGetApp()->LoadCursor(IDC_CURSOR3);
}

//--------------------------------------------------------| steuerelement-initialisierung
CDICEDlg
void CDICEDlg::DoDataExchange(CDataExchange* pDX)
{
        CDialog::DoDataExchange(pDX);
        //{{AFX_DATA_MAP(CDICEDlg)
        //}}AFX_DATA_MAP
}

//--------------------------------------------------------| ereignis-deklaration CDICEDlg
BEGIN_MESSAGE_MAP(CDICEDlg, CDialog)
        //{{AFX_MSG_MAP(CDICEDlg)
        ON_WM_SYSCOMMAND()
        ON_WM_QUERYDRAGICON()
        ON_WM_TIMER()
        ON_WM_PAINT()
        ON_WM_SIZE()
        ON_WM_LBUTTONDOWN()
        ON_BN_CLICKED(IDC_BUTTON1, OnButton1)
        ON_WM_MOUSEMOVE()
        ON_WM_LBUTTONUP()
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()


//--------------------------------------------------------| bei dialogstart CDICEDlg
BOOL CDICEDlg::OnInitDialog()
{
        CDialog::OnInitDialog();

        qzufall((time(0)-1234567890));//seed

        CWinApp*  pApp = AfxGetApp();  // ini profil lesen
        ini_sw  = pApp->GetProfileInt("INI","sw",0);

        fb_hg   = pApp->GetProfileInt("Farbe","Hintergrund",8421504);
        fb_wrf  = pApp->GetProfileInt("Farbe","Würfel",0);
        fb_ag   = pApp->GetProfileInt("Farbe","Augen",16777215);
        fb_zhl  = pApp->GetProfileInt("Farbe","Zahlen",16777215);
        vms_    = pApp->GetProfileInt("Geschwindigkeit","msec",100);
        von_    = pApp->GetProfileInt("Geschwindigkeit","onset",300);
        vwl_    = pApp->GetProfileInt("Geschwindigkeit","Würfel",10);
        z_frm   = pApp->GetProfileInt("Zahlenformat","R",1);
        sw_lg   = pApp->GetProfileInt("Logdatei","schreiben",1);


        if(ini_sw==0)//bei nicht vorhandener inidatei
        {
                                        pApp->WriteProfileInt("Farbe","Hintergrund",  fb_hg);
                                        pApp->WriteProfileInt("Farbe","Würfel",fb_wrf);
                                        pApp->WriteProfileInt("Farbe","Augen",fb_ag);
                                        pApp->WriteProfileInt("Farbe","Zahlen",fb_zhl);
                                        pApp->WriteProfileInt("Geschwindigkeit","msec",vms_);
                                        pApp->WriteProfileInt("Geschwindigkeit","onset",von_);
                                        pApp->WriteProfileInt("Geschwindigkeit","Würfel",vwl_ );
                if(z_frm==0)pApp->WriteProfileInt("Zahlenformat","N",1 );
                if(z_frm==1)pApp->WriteProfileInt("Zahlenformat","N",0 );
```

```
                                        pApp->WriteProfileInt("Zahlenformat","R",z_frm );
                                        pApp->WriteProfileInt("Logdatei","schreiben",sw_lg);
        }

        fb_hg_0=  fb_hg;
        fb_wrf_0= fb_wrf;
        fb_ag_0=  fb_ag;
        fb_zhl_0= fb_zhl;
        vms_0=    vms_;
        von_0=    von_;
        vwl_0=    vwl_;
        z_frm_0=z_frm;

        pApp->WriteProfileInt("INI","sw",  1); // ini datei aktivieren

        wt_sw=0;

        SetTimer(1,vwl_,0);  //timer1 lauf geschwindigkeit
        SetTimer(2,vms_,0);  //timer2 wurf geschwindigkeit
                           //timer3 onset siehe OnLButtonDown() funktion

        CMenu*  zgr = GetSystemMenu(0);

                        zgr->AppendMenu(MF_SEPARATOR);
                        zgr->AppendMenu(MF_STRING, 0x0010, "Einstellungen...");
                        zgr->AppendMenu(MF_SEPARATOR);
                        zgr->AppendMenu(MF_STRING, 0x0020, "Information");

        SetIcon(m_hIcon, 1);SetIcon(m_hIcon, 0);return 1;
}

//------------------------------------------------------------| system-menue-punkte CDICEDlg
void CDICEDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
        if (nID == 0x0010)
        {
                DICEeinst o; o.DoModal();
        }

        if (nID == 0x0020)
        {
                m_InfoDlg.DestroyWindow(); //<--

                m_InfoDlg.Create(IDD_AboutDlg );
                m_InfoDlg.ShowWindow(SW_SHOW);
                m_InfoDlg.BringWindowToTop();
        }

        else{CDialog::OnSysCommand(nID, lParam);}
}

HCURSOR CDICEDlg::OnQueryDragIcon(){return (HCURSOR) m_hIcon;}


//------------------------------------------------------------| OnPaint CDICEDlg
void CDICEDlg::OnPaint()
{
        UpdateWindow();

        CPaintDC ooo(this);

                                                 CRect oo(0, 0, dlg_x, dlg_y);//hintergrund
                         ooo.FillSolidRect(oo, fb_hg );

                         ooo.SetTextColor(fb_zhl); //textfarbe

                                                 CFont o6;

o6.CreateFont(13,4,0,0,300,0,0,0,0,OUT_DEFAULT_PRECIS, CLIP_DEFAULT_PRECIS, DEFAULT_QUALITY,
DEFAULT_PITCH,"Lucida Sans Unicode" );
                         ooo.SelectObject(&o6);

        if(wt_sw==1) //warteschleife

                     ooo.TextOut(5,0,"DICE ...");


        if(wt_sw==0)
        {

                              itoa(y_1,ccc,10);
                                       cc=ccc;
                         ooo.TextOut(5,0,cc); // rnd int

                         if(z_frm==1)
                         {
                         ooo.TextOut(11,0,'.'); //komma

                                 itoa(y_2,ccc,10); //rnd dezimale
                                          cc=ccc;
                         ooo.TextOut(14,0,cc);
                         }

                                  CRect o1(dlg_x-38-x_, dlg_y-38, dlg_x-9-x_, dlg_y-9);//würfel
                         ooo.FillSolidRect(o1,  fb_wrf  );

                                                 CRect o7(dlg_x-21-x_, dlg_y-21, dlg_x-26-x_,
dlg_y-26);//würfel auge mitte
                                                 CRect o3(dlg_x-33-x_, dlg_y-19, dlg_x-28-x_,
dlg_y-14);//würfel auge lu
                                                 CRect o4(dlg_x-19-x_, dlg_y-28, dlg_x-14-x_,
dlg_y-33);//würfel auge ro
                                                 CRect o2(dlg_x-19-x_, dlg_y-19, dlg_x-14-x_,
dlg_y-14);//würfel auge ru
                                                 CRect o5(dlg_x-33-x_, dlg_y-28, dlg_x-28-x_,
dlg_y-33);//würfel auge lo
                                                 CRect o9(dlg_x-33-x_, dlg_y-21, dlg_x-28-x_,
dlg_y-26);//würfel auge mitte l
                                                 CRect o8(dlg_x-19-x_, dlg_y-21, dlg_x-14-x_,
dlg_y-26);//würfel auge mitte r
                if(y_1==1)//rnd int == 1
                {
                         ooo.FillSolidRect(o7,  fb_ag );  SetCursor(m_Csr2);
                }

                if(y_1==2)//rnd int == 2
                {
                         ooo.FillSolidRect(o3,  fb_ag );  SetCursor(m_Csr3);
                         ooo.FillSolidRect(o4,  fb_ag );
                }

                if(y_1==3)//rnd int == 3
                {
                         ooo.FillSolidRect(o3,  fb_ag);        SetCursor(m_Csr2);
                         ooo.FillSolidRect(o7,  fb_ag );
                         ooo.FillSolidRect(o4,  fb_ag );
                }

                if(y_1==4)//rnd int == 4
                {
                         ooo.FillSolidRect(o2,  fb_ag );  SetCursor(m_Csr3);
                         ooo.FillSolidRect(o3,  fb_ag );
                         ooo.FillSolidRect(o4,  fb_ag );
                         ooo.FillSolidRect(o5,  fb_ag );
                }

                if(y_1==5)//rnd int == 5
                {
                         ooo.FillSolidRect(o2,  fb_ag );  SetCursor(m_Csr2);
                         ooo.FillSolidRect(o3,  fb_ag );
                         ooo.FillSolidRect(o7,  fb_ag );
                         ooo.FillSolidRect(o4,  fb_ag );
                         ooo.FillSolidRect(o5,  fb_ag );
                }

                if(y_1==6)//rnd int == 6
                {
                         ooo.FillSolidRect(o2,  fb_ag );  SetCursor(m_Csr3);
                         ooo.FillSolidRect(o3,  fb_ag );
                         ooo.FillSolidRect(o8,  fb_ag );
                         ooo.FillSolidRect(o9,  fb_ag );
                         ooo.FillSolidRect(o4,  fb_ag );
                         ooo.FillSolidRect(o5,  fb_ag );
                }
        }
}

//------------------------------------------------------------| timer ereignisse CDICEDlg
void CDICEDlg::OnTimer(UINT nIDEvent)
{
        if(nIDEvent==3 && wt_sw==1) //warteschaltung
        {
                   wt_sw++;
                if(wt_sw==2)wt_sw=0;
        }

        if(nIDEvent==1 && stp==1 && wt_sw==0) //timer1 lauf geschwindigkeit
        {
                  if(stp==1)x_+=2;
                if(x_>=150){x_=140; stp=0; logsw=1; }

                RedrawWindow();
        }

        if(nIDEvent==1 && (fb_hg_0 !=fb_hg  || // bei neueinstellungen
                               fb_wrf_0!=fb_wrf || //
                                         fb_ag_0 !=fb_ag  || //
                                         fb_zhl_0!=fb_zhl || //
                                         vms_0   !=vms_   || //
                                         von_0   !=von_   || //
                                         vwl_0   !=vwl_   || //
                                         z_frm_0 !=z_frm    ) //
          )                                     //
        {                                          //
                fb_hg_0= fb_hg;                    //
                fb_wrf_0=fb_wrf;                   //
                fb_ag_0= fb_ag;                    //
                fb_zhl_0=fb_zhl;                   //
                                                   //
                           vms_0 =vms_;            //
                SetTimer(2,vms_,0);                //
                                             //
                                  vwl_0 =vwl_;            //
                SetTimer(1,vwl_,0);                //
                                             //
                                  von_0 =von_;            //
                SetTimer(3,von_,0);                //
                                             //
                z_frm_0 =z_frm;                    //
                                                       //
                RedrawWindow();                    //
        }

        if(nIDEvent==2 && stp==1 && wt_sw==0) //timer2 wurf geschwindigkeit
        {
                     dy_=qzufall(fn_erg); // zufallszahl
      y_1=floor(dy_);                // zufallszahl int
          y_2=     (dy_-y_1)*1000;       // zufallszahl dezimale
        }

        if(nIDEvent==2 && stp==0 && logsw==1 && sw_lg==1) //logfile
        {
                if(fopen("DICE_log.txt","a")!=0)
                {
                                              FILE *log;
                                                       log=fopen("DICE_log.txt","a");
                  if(z_frm==1)fprintf(log,"%.16f\n",dy_);
                  if(z_frm==0)fprintf(log,"%i\n",y_1);
                                         fclose(log);

                         logsw=0;
                }
        }

        CDialog::OnTimer(nIDEvent);
}

//------------------------------------------------------------| OnSize CDICEDlg
void CDICEDlg::OnSize(UINT nType, int cx, int cy)
{
        CDialog::OnSize(nType, cx, cy);
        dlg_x= cx;
        dlg_y= cy;

        RedrawWindow();
}

void CDICEDlg::OnMouseMove(UINT nFlags, CPoint point)
{
        if(stp==0)SetCursor(m_Csr1);
        if(stp!=0)SetCursor(0);

        CDialog::OnMouseMove(nFlags, point);
}

//----------------------------------------------------------|  click links ereignisse
CDICEDlg
void CDICEDlg::OnLButtonDown(UINT nFlags, CPoint point)
{
        SetCursor(0);

        stp=1; x_=0; wt_sw=1; SetTimer(3,von_,0);RedrawWindow();

        CDialog::OnLButtonDown(nFlags, point);
}

void CDICEDlg::OnLButtonUp(UINT nFlags, CPoint point)
{
        SetCursor(0);

        CDialog::OnLButtonUp(nFlags, point);
}

//----------------------------------------------------------|  leertaste ereignis
(unsichtbarer button)
void CDICEDlg::OnButton1() {OnLButtonDown(1,1);}


//----------------------------------------------------------|  quasi-zufallszahl (1,6)
funktion
double CDICEDlg::qzufall(double seed)
{
        fn_erg =                10*( pow(seed,SIGMA) - floor( pow(seed,SIGMA) ) )
                     - floor( 10*( pow(seed,SIGMA) - floor( pow(seed,SIGMA) ) ) );

        fn_erg= 1 + 6*fn_erg;

        return fn_erg;
};

//----------------------------------------------------------| einstellungsvariablenübernahme
funktionen
void CDICEDlg::fhg_(int fhg){fb_hg=fhg;} //-----------------| hintergrundfarbenfunktion
void CDICEDlg::fwl_(int fwl){fb_wrf=fwl;} //----------------| würfelfarbenfunktion
void CDICEDlg::fag_(int fag){fb_ag=fag;} //-----------------| augenfarbenfunktion
void CDICEDlg::fzl_(int fzl){fb_zhl=fzl;} //----------------| zahlenfarbenfunktion
void CDICEDlg::v_ms(int vms){vms_=vms;} //------------------| geschwindigkeit (würf in msec)
void CDICEDlg::v_wl(int vwl){vwl_=vwl;} //------------------| geschwindigkeit (würfel in
msec)
void CDICEDlg::v_on(int von){von_=von;} //------------------| geschwindigkeit (onset in msec)
void CDICEDlg::zfrm(int zfrm_){z_frm=zfrm_;} //-------------| zahlenformat
void CDICEDlg::lg_(int lg){sw_lg=lg;} //--------------------| log schreiben
```

References

Schrausser, D. G. (2023). *Schrausser/DICEwin: DICE for Windows (v1.0.5).* Zenodo. DOI:10.5281/zenodo.7644967