```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Windows.Media.Animation;
using System.Windows.Media.Media3D;
using System.Threading;
using System.IO;

namespace WpfGames
{
    /// <summary>
    /// Interaction logic for Window1.xaml
    /// </summary>
    public partial class RubikWnd : Window
    {
        bool bClockWise = true;
        int nShuffle = 0;
        string LogFileName = "";
        int nGlobleIndex = 0;
        bool bLogFile = false;
        Random ran = new Random();
        bool bUndoState = false;

        CubeObject[, ,] FrontCubeObjectX = new CubeObject[3, 3, 3];
        CubeObject[, ,] FrontCubeObjectY = new CubeObject[3, 3, 3];
        CubeObject[, ,] FrontCubeObjectZ = new CubeObject[3, 3, 3];

        CubeObject[, ,] BackCubeObjectX = new CubeObject[3, 3, 3];
        CubeObject[, ,] BackCubeObjectY = new CubeObject[3, 3, 3];
        CubeObject[, ,] BackCubeObjectZ = new CubeObject[3, 3, 3];

        Stack<RubikMove> UndoStack = new Stack<RubikMove>();
        Stack<RubikMove> RedoStack = new Stack<RubikMove>();


        Transform3DGroup FrontTransGroupX0Y0Z0 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX0Y0Z1 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX0Y0Z2 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX0Y1Z0 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX0Y1Z1 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX0Y1Z2 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX0Y2Z0 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX0Y2Z1 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX0Y2Z2 = new Transform3DGroup();

        Transform3DGroup FrontTransGroupX1Y0Z0 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX1Y0Z1 = new Transform3DGroup();
```

```
        Transform3DGroup FrontTransGroupX1Y0Z2 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX1Y1Z0 = new Transform3DGroup();
        //   Transform3DGroup TransGroupX1Y1Z1 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX1Y1Z2 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX1Y2Z0 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX1Y2Z1 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX1Y2Z2 = new Transform3DGroup();

        Transform3DGroup FrontTransGroupX2Y0Z0 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX2Y0Z1 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX2Y0Z2 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX2Y1Z0 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX2Y1Z1 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX2Y1Z2 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX2Y2Z0 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX2Y2Z1 = new Transform3DGroup();
        Transform3DGroup FrontTransGroupX2Y2Z2 = new Transform3DGroup();


        Transform3DGroup BackTransGroupX0Y0Z0 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX0Y0Z1 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX0Y0Z2 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX0Y1Z0 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX0Y1Z1 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX0Y1Z2 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX0Y2Z0 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX0Y2Z1 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX0Y2Z2 = new Transform3DGroup();

        Transform3DGroup BackTransGroupX1Y0Z0 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX1Y0Z1 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX1Y0Z2 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX1Y1Z0 = new Transform3DGroup();
        //   Transform3DGroup TransGroupX1Y1Z1 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX1Y1Z2 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX1Y2Z0 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX1Y2Z1 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX1Y2Z2 = new Transform3DGroup();

        Transform3DGroup BackTransGroupX2Y0Z0 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX2Y0Z1 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX2Y0Z2 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX2Y1Z0 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX2Y1Z1 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX2Y1Z2 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX2Y2Z0 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX2Y2Z1 = new Transform3DGroup();
        Transform3DGroup BackTransGroupX2Y2Z2 = new Transform3DGroup();

        public RubikWnd()
        {
            InitializeComponent();
            CreateLogFile();

            FrontCubeObjectX[0, 0, 0] = new CubeObject(FrontTransGroupX0Y0Z0,
CUBEX0Y0Z0, "CUBEX0Y0Z0");
```

```
            FrontCubeObjectX[0, 0, 1] = new CubeObject(FrontTransGroupX0Y0Z1,
CUBEX0Y0Z1, "CUBEX0Y0Z1");
            FrontCubeObjectX[0, 0, 2] = new CubeObject(FrontTransGroupX0Y0Z2,
CUBEX0Y0Z2, "CUBEX0Y0Z2");
            FrontCubeObjectX[0, 1, 0] = new CubeObject(FrontTransGroupX0Y1Z0,
CUBEX0Y1Z0, "CUBEX0Y1Z0");
            FrontCubeObjectX[0, 1, 1] = new CubeObject(FrontTransGroupX0Y1Z1,
CUBEX0Y1Z1, "CUBEX0Y1Z1");
            FrontCubeObjectX[0, 1, 2] = new CubeObject(FrontTransGroupX0Y1Z2,
CUBEX0Y1Z2, "CUBEX0Y1Z2");
            FrontCubeObjectX[0, 2, 0] = new CubeObject(FrontTransGroupX0Y2Z0,
CUBEX0Y2Z0, "CUBEX0Y2Z0");
            FrontCubeObjectX[0, 2, 1] = new CubeObject(FrontTransGroupX0Y2Z1,
CUBEX0Y2Z1, "CUBEX0Y2Z1");
            FrontCubeObjectX[0, 2, 2] = new CubeObject(FrontTransGroupX0Y2Z2,
CUBEX0Y2Z2, "CUBEX0Y2Z2");

            FrontCubeObjectX[1, 0, 0] = new CubeObject(FrontTransGroupX1Y0Z0,
CUBEX1Y0Z0, "CUBEX1Y0Z0");
            FrontCubeObjectX[1, 0, 1] = new CubeObject(FrontTransGroupX1Y0Z1,
CUBEX1Y0Z1, "CUBEX1Y0Z1");
            FrontCubeObjectX[1, 0, 2] = new CubeObject(FrontTransGroupX1Y0Z2,
CUBEX1Y0Z2, "CUBEX1Y0Z2");
            FrontCubeObjectX[1, 1, 0] = new CubeObject(FrontTransGroupX1Y1Z0,
CUBEX1Y1Z0, "CUBEX1Y1Z0");
            FrontCubeObjectX[1, 1, 1] = null;
            FrontCubeObjectX[1, 1, 2] = new CubeObject(FrontTransGroupX1Y1Z2,
CUBEX1Y1Z2, "CUBEX1Y1Z2");
            FrontCubeObjectX[1, 2, 0] = new CubeObject(FrontTransGroupX1Y2Z0,
CUBEX1Y2Z0, "CUBEX1Y2Z0");
            FrontCubeObjectX[1, 2, 1] = new CubeObject(FrontTransGroupX1Y2Z1,
CUBEX1Y2Z1, "CUBEX1Y2Z1");
            FrontCubeObjectX[1, 2, 2] = new CubeObject(FrontTransGroupX1Y2Z2,
CUBEX1Y2Z2, "CUBEX1Y2Z2");

            FrontCubeObjectX[2, 0, 0] = new CubeObject(FrontTransGroupX2Y0Z0,
CUBEX2Y0Z0, "CUBEX2Y0Z0");
            FrontCubeObjectX[2, 0, 1] = new CubeObject(FrontTransGroupX2Y0Z1,
CUBEX2Y0Z1, "CUBEX2Y0Z1");
            FrontCubeObjectX[2, 0, 2] = new CubeObject(FrontTransGroupX2Y0Z2,
CUBEX2Y0Z2, "CUBEX2Y0Z2");
            FrontCubeObjectX[2, 1, 0] = new CubeObject(FrontTransGroupX2Y1Z0,
CUBEX2Y1Z0, "CUBEX2Y1Z0");
            FrontCubeObjectX[2, 1, 1] = new CubeObject(FrontTransGroupX2Y1Z1,
CUBEX2Y1Z1, "CUBEX2Y1Z1");
            FrontCubeObjectX[2, 1, 2] = new CubeObject(FrontTransGroupX2Y1Z2,
CUBEX2Y1Z2, "CUBEX2Y1Z2");
            FrontCubeObjectX[2, 2, 0] = new CubeObject(FrontTransGroupX2Y2Z0,
CUBEX2Y2Z0, "CUBEX2Y2Z0");
            FrontCubeObjectX[2, 2, 1] = new CubeObject(FrontTransGroupX2Y2Z1,
CUBEX2Y2Z1, "CUBEX2Y2Z1");
            FrontCubeObjectX[2, 2, 2] = new CubeObject(FrontTransGroupX2Y2Z2,
CUBEX2Y2Z2, "CUBEX2Y2Z2");

            // back
            BackCubeObjectX[0, 0, 0] = new CubeObject(BackTransGroupX0Y0Z0,
BCUBEX0Y0Z0, "CUBEX0Y0Z0");
```

```
            BackCubeObjectX[0, 0, 1] = new CubeObject(BackTransGroupX0Y0Z1,
BCUBEX0Y0Z1, "CUBEX0Y0Z1");
            BackCubeObjectX[0, 0, 2] = new CubeObject(BackTransGroupX0Y0Z2,
BCUBEX0Y0Z2, "CUBEX0Y0Z2");
            BackCubeObjectX[0, 1, 0] = new CubeObject(BackTransGroupX0Y1Z0,
BCUBEX0Y1Z0, "CUBEX0Y1Z0");
            BackCubeObjectX[0, 1, 1] = new CubeObject(BackTransGroupX0Y1Z1,
BCUBEX0Y1Z1, "CUBEX0Y1Z1");
            BackCubeObjectX[0, 1, 2] = new CubeObject(BackTransGroupX0Y1Z2,
BCUBEX0Y1Z2, "CUBEX0Y1Z2");
            BackCubeObjectX[0, 2, 0] = new CubeObject(BackTransGroupX0Y2Z0,
BCUBEX0Y2Z0, "CUBEX0Y2Z0");
            BackCubeObjectX[0, 2, 1] = new CubeObject(BackTransGroupX0Y2Z1,
BCUBEX0Y2Z1, "CUBEX0Y2Z1");
            BackCubeObjectX[0, 2, 2] = new CubeObject(BackTransGroupX0Y2Z2,
BCUBEX0Y2Z2, "CUBEX0Y2Z2");

            BackCubeObjectX[1, 0, 0] = new CubeObject(BackTransGroupX1Y0Z0,
BCUBEX1Y0Z0, "CUBEX1Y0Z0");
            BackCubeObjectX[1, 0, 1] = new CubeObject(BackTransGroupX1Y0Z1,
BCUBEX1Y0Z1, "CUBEX1Y0Z1");
            BackCubeObjectX[1, 0, 2] = new CubeObject(BackTransGroupX1Y0Z2,
BCUBEX1Y0Z2, "CUBEX1Y0Z2");
            BackCubeObjectX[1, 1, 0] = new CubeObject(BackTransGroupX1Y1Z0,
BCUBEX1Y1Z0, "CUBEX1Y1Z0");
            BackCubeObjectX[1, 1, 1] = null;
            BackCubeObjectX[1, 1, 2] = new CubeObject(BackTransGroupX1Y1Z2,
BCUBEX1Y1Z2, "CUBEX1Y1Z2");
            BackCubeObjectX[1, 2, 0] = new CubeObject(BackTransGroupX1Y2Z0,
BCUBEX1Y2Z0, "CUBEX1Y2Z0");
            BackCubeObjectX[1, 2, 1] = new CubeObject(BackTransGroupX1Y2Z1,
BCUBEX1Y2Z1, "CUBEX1Y2Z1");
            BackCubeObjectX[1, 2, 2] = new CubeObject(BackTransGroupX1Y2Z2,
BCUBEX1Y2Z2, "CUBEX1Y2Z2");

            BackCubeObjectX[2, 0, 0] = new CubeObject(BackTransGroupX2Y0Z0,
BCUBEX2Y0Z0, "CUBEX2Y0Z0");
            BackCubeObjectX[2, 0, 1] = new CubeObject(BackTransGroupX2Y0Z1,
BCUBEX2Y0Z1, "CUBEX2Y0Z1");
            BackCubeObjectX[2, 0, 2] = new CubeObject(BackTransGroupX2Y0Z2,
BCUBEX2Y0Z2, "CUBEX2Y0Z2");
            BackCubeObjectX[2, 1, 0] = new CubeObject(BackTransGroupX2Y1Z0,
BCUBEX2Y1Z0, "CUBEX2Y1Z0");
            BackCubeObjectX[2, 1, 1] = new CubeObject(BackTransGroupX2Y1Z1,
BCUBEX2Y1Z1, "CUBEX2Y1Z1");
            BackCubeObjectX[2, 1, 2] = new CubeObject(BackTransGroupX2Y1Z2,
BCUBEX2Y1Z2, "CUBEX2Y1Z2");
            BackCubeObjectX[2, 2, 0] = new CubeObject(BackTransGroupX2Y2Z0,
BCUBEX2Y2Z0, "CUBEX2Y2Z0");
            BackCubeObjectX[2, 2, 1] = new CubeObject(BackTransGroupX2Y2Z1,
BCUBEX2Y2Z1, "CUBEX2Y2Z1");
            BackCubeObjectX[2, 2, 2] = new CubeObject(BackTransGroupX2Y2Z2,
BCUBEX2Y2Z2, "CUBEX2Y2Z2");

            ////////////// Bottom to up for Y-axis (Layer, x and z)
            CopyXSlicesToYSlices();
            CopyXSlicesToZSlices();
```

```
        }

        void CopyXSlicesToYSlices()
        {
            for (int x = 0; x < 3; x++)
            {
                for (int y = 0; y < 3; y++)
                {
                    for (int z = 0; z < 3; z++)
                    {
                        FrontCubeObjectY[y, x, z] = (FrontCubeObjectX[x, y, z]
== null) ? null : new CubeObject(FrontCubeObjectX[x, y, z]);
                        BackCubeObjectY[y, x, z] = (BackCubeObjectX[x, y, z]
== null) ? null : new CubeObject(BackCubeObjectX[x, y, z]);
                    }
                }
            }
        }

        void CopyXSlicesToZSlices()
        {
            for (int x = 0; x < 3; x++)
            {
                for (int y = 0; y < 3; y++)
                {
                    for (int z = 0; z < 3; z++)
                    {
                        FrontCubeObjectZ[z, x, y] = (FrontCubeObjectX[x, y, z]
== null) ? null : new CubeObject(FrontCubeObjectX[x, y, z]);
                        BackCubeObjectZ[z, x, y] = (BackCubeObjectX[x, y, z]
== null) ? null : new CubeObject(BackCubeObjectX[x, y, z]);
                    }
                }
            }
        }

        void CopyYSlicesToXSlices()
        {
            for (int x = 0; x < 3; x++)
            {
                for (int y = 0; y < 3; y++)
                {
                    for (int z = 0; z < 3; z++)
                    {
                        FrontCubeObjectX[x, y, z] = (FrontCubeObjectY[y, x, z]
== null) ? null : new CubeObject(FrontCubeObjectY[y, x, z]);
                        BackCubeObjectX[x, y, z] = (BackCubeObjectY[y, x, z]
== null) ? null : new CubeObject(BackCubeObjectY[y, x, z]);
                    }
                }
            }
        }

        void CopyYSlicesToZSlices()
        {
            for (int x = 0; x < 3; x++)
            {
```

```
                for (int y = 0; y < 3; y++)
                {
                    for (int z = 0; z < 3; z++)
                    {
                        FrontCubeObjectZ[z, x, y] = (FrontCubeObjectY[y, x, z]
== null) ? null : new CubeObject(FrontCubeObjectY[y, x, z]);
                        BackCubeObjectZ[z, x, y] = (BackCubeObjectY[y, x, z]
== null) ? null : new CubeObject(BackCubeObjectY[y, x, z]);
                    }
                }
            }
        }

        void CopyZSlicesToXSlices()
        {
            for (int x = 0; x < 3; x++)
            {
                for (int y = 0; y < 3; y++)
                {
                    for (int z = 0; z < 3; z++)
                    {
                        FrontCubeObjectX[x, y, z] = (FrontCubeObjectZ[z, x, y]
== null) ? null : new CubeObject(FrontCubeObjectZ[z, x, y]);
                        BackCubeObjectX[x, y, z] = (BackCubeObjectZ[z, x, y]
== null) ? null : new CubeObject(BackCubeObjectZ[z, x, y]);
                    }
                }
            }
        }

        void CopyZSlicesToYSlices()
        {
            for (int x = 0; x < 3; x++)
            {
                for (int y = 0; y < 3; y++)
                {
                    for (int z = 0; z < 3; z++)
                    {
                        FrontCubeObjectY[y, x, z] = (FrontCubeObjectZ[z, x, y]
== null) ? null : new CubeObject(FrontCubeObjectZ[z, x, y]);
                        BackCubeObjectY[y, x, z] = (BackCubeObjectZ[z, x, y]
== null) ? null : new CubeObject(BackCubeObjectZ[z, x, y]);
                    }
                }
            }
        }
        private void CreateLogFile()
        {
            string FileName = "RubLog.txt";
            int nVertion = 0;

            if (bLogFile == true && LogFileName == "")
            {
                while (File.Exists(FileName) == true)
                {
                    FileName = "RubLog" + nVertion.ToString() + ".txt";
                    nVertion++;
```

```
            }
            LogFileName = FileName;
            FileStream fs = new FileStream(LogFileName, FileMode.Create);
            fs.Close();
            ErrorLog(DateTime.Now.ToString());
        }
    }
    public void ErrorLog(string sErrMsg)
    {
        StreamWriter sw = new StreamWriter(LogFileName, true);
        sw.WriteLine(sErrMsg);
        sw.Flush();
        sw.Close();
    }
    private void OnAboutClick(object sender, RoutedEventArgs e)
    {
        WpfGames.AboutDialog about = new AboutDialog();
        about.ShowDialog();

    }
    private void RotateClickCW(object sender, RoutedEventArgs e)
    {
        bClockWise = true;
        RotateTheCube();
    }
    private void RotateClickCCW(object sender, RoutedEventArgs e)
    {
        bClockWise = false;
        RotateTheCube();
    }
    private void RotateClickShuffle(object sender, RoutedEventArgs e)
    {
        EnableControl(false);
        nShuffle = 25;
        Automate();
    }

    private void OnAxisChange(object sender, RoutedEventArgs e)
    {
        int Index = AxisCB.SelectedIndex;
        if (SliceCB != null)
        {
            SliceCB.Items.Clear();
            if (Index == 0)
            {
                SliceCB.Items.Add("Left");
                SliceCB.Items.Add("Middle");
                SliceCB.Items.Add("Right");
                SliceCB.Items.Add("All");
            }
            else if (Index == 1)
            {
                SliceCB.Items.Add("Bottom");
                SliceCB.Items.Add("Middle");
                SliceCB.Items.Add("Top");
                SliceCB.Items.Add("All");
            }
```

```
                else
                {
                    SliceCB.Items.Add("Back");
                    SliceCB.Items.Add("Middle");
                    SliceCB.Items.Add("Front");
                    SliceCB.Items.Add("All");
                }
                SliceCB.SelectedIndex = 0;
            }
        }
        private void OnCheckLogFile(object sender, RoutedEventArgs e)
        {
            bLogFile = (bool)CheckBtnLogFile.IsChecked;
            CreateLogFile();
        }
        private void OnUndoClick(object sender, RoutedEventArgs e)
        {
            if (UndoStack.Count > 0 )
            {
                bUndoState = true;
                RubikMove rm = new RubikMove(UndoStack.Pop());
                rm._bMoveCW = !rm._bMoveCW;
                RedoStack.Push(rm);
                simpleButtonRedo.IsEnabled = true;
                AxisCB.SelectedIndex = rm._nAxis;
                SliceCB.SelectedIndex = rm._nSlice;
                bClockWise = rm._bMoveCW;
                RotateTheCube();
            }
        }
        private void OnRedoClick(object sender, RoutedEventArgs e)
        {
            if (RedoStack.Count > 0)
            {
                bUndoState = true;
                RubikMove rm = new RubikMove(RedoStack.Pop());
                rm._bMoveCW = !rm._bMoveCW;
                UndoStack.Push(rm);
                simpleButtonUndo.IsEnabled = true;

                AxisCB.SelectedIndex = rm._nAxis;
                SliceCB.SelectedIndex = rm._nSlice;
                bClockWise = rm._bMoveCW;
                RotateTheCube();
            }
        }

        private void RotateTheCube()
        {
            using(Mutex mutex = new Mutex(false,"RotateTheCube"))
            {
                int AxisIndex = AxisCB.SelectedIndex;
                int SliceIndex = SliceCB.SelectedIndex;

                if (bUndoState == false)
                    UndoStack.Push(new RubikMove(AxisIndex, SliceIndex,
bClockWise));
```

```
                if (AxisIndex == 0)
                {
                    AxisAngleRotation3D Rotation = new AxisAngleRotation3D();
                    Rotation.Axis = (bClockWise == true) ? new Vector3D(1, 0,
0) : new Vector3D(-1, 0, 0);

                    if (SliceIndex != 3)
                    {
                        DoubleAnimation da = new DoubleAnimation();
                        da.Completed += new EventHandler(da_Completed);

                        Transform3D t3d = new RotateTransform3D(Rotation);

                        FrontCubeObjectX[SliceIndex, 0,
0].SetTransform3D(t3d);
                        FrontCubeObjectX[SliceIndex, 0,
1].SetTransform3D(t3d);
                        FrontCubeObjectX[SliceIndex, 0,
2].SetTransform3D(t3d);

                        FrontCubeObjectX[SliceIndex, 1,
0].SetTransform3D(t3d);
                        if (FrontCubeObjectX[SliceIndex, 1, 1] != null)
                            FrontCubeObjectX[SliceIndex, 1,
1].SetTransform3D(t3d);
                        FrontCubeObjectX[SliceIndex, 1,
2].SetTransform3D(t3d);

                        FrontCubeObjectX[SliceIndex, 2,
0].SetTransform3D(t3d);
                        FrontCubeObjectX[SliceIndex, 2,
1].SetTransform3D(t3d);
                        FrontCubeObjectX[SliceIndex, 2,
2].SetTransform3D(t3d);

                        BackCubeObjectX[SliceIndex, 0, 0].SetTransform3D(t3d);
                        BackCubeObjectX[SliceIndex, 0, 1].SetTransform3D(t3d);
                        BackCubeObjectX[SliceIndex, 0, 2].SetTransform3D(t3d);

                        BackCubeObjectX[SliceIndex, 1, 0].SetTransform3D(t3d);
                        if (BackCubeObjectX[SliceIndex, 1, 1] != null)
                            BackCubeObjectX[SliceIndex, 1,
1].SetTransform3D(t3d);
                        BackCubeObjectX[SliceIndex, 1, 2].SetTransform3D(t3d);

                        BackCubeObjectX[SliceIndex, 2, 0].SetTransform3D(t3d);
                        BackCubeObjectX[SliceIndex, 2, 1].SetTransform3D(t3d);
                        BackCubeObjectX[SliceIndex, 2, 2].SetTransform3D(t3d);

                        da.By = 90;
                        da.Duration = new Duration(new TimeSpan(0, 0, 0, 0,
500));
                        da.RepeatBehavior = new RepeatBehavior(1);

Rotation.BeginAnimation(AxisAngleRotation3D.AngleProperty, da);
                    }
```

```
                    else
                    {
                        DoubleAnimation da = new DoubleAnimation();
                        da.Completed += new EventHandler(da_Completed);

                        Transform3D t3d = new RotateTransform3D(Rotation);
                        t3d = new RotateTransform3D(Rotation);
                        for (int i = 0; i < 3; i++)
                        {
                            FrontCubeObjectX[i, 0, 0].SetTransform3D(t3d);
                            FrontCubeObjectX[i, 0, 1].SetTransform3D(t3d);
                            FrontCubeObjectX[i, 0, 2].SetTransform3D(t3d);

                            FrontCubeObjectX[i, 1, 0].SetTransform3D(t3d);
                            if (FrontCubeObjectX[i, 1, 1] != null)
                                FrontCubeObjectX[i, 1, 1].SetTransform3D(t3d);
                            FrontCubeObjectX[i, 1, 2].SetTransform3D(t3d);

                            FrontCubeObjectX[i, 2, 0].SetTransform3D(t3d);
                            FrontCubeObjectX[i, 2, 1].SetTransform3D(t3d);
                            FrontCubeObjectX[i, 2, 2].SetTransform3D(t3d);

                            BackCubeObjectX[i, 0, 0].SetTransform3D(t3d);
                            BackCubeObjectX[i, 0, 1].SetTransform3D(t3d);
                            BackCubeObjectX[i, 0, 2].SetTransform3D(t3d);

                            BackCubeObjectX[i, 1, 0].SetTransform3D(t3d);
                            if (BackCubeObjectX[i, 1, 1] != null)
                                BackCubeObjectX[i, 1, 1].SetTransform3D(t3d);
                            BackCubeObjectX[i, 1, 2].SetTransform3D(t3d);

                            BackCubeObjectX[i, 2, 0].SetTransform3D(t3d);
                            BackCubeObjectX[i, 2, 1].SetTransform3D(t3d);
                            BackCubeObjectX[i, 2, 2].SetTransform3D(t3d);
                        }

                        da.By = 90;
                        da.Duration = new Duration(new TimeSpan(0, 0, 0, 0,
500));
                        da.RepeatBehavior = new RepeatBehavior(1);

Rotation.BeginAnimation(AxisAngleRotation3D.AngleProperty, da);
                    }

                }
                else if (AxisIndex == 1) // y Rotation
                {
                    AxisAngleRotation3D Rotation = new AxisAngleRotation3D();
                    Rotation.Axis = (bClockWise == true) ? new Vector3D(0, -1,
0) : new Vector3D(0, 1, 0);

                    if (SliceIndex != 3)
                    {
                        DoubleAnimation da = new DoubleAnimation();
                        da.Completed += new EventHandler(da_Completed);

                        Transform3D t3d = new RotateTransform3D(Rotation);
```

```
                        t3d = new RotateTransform3D(Rotation);

                        FrontCubeObjectY[SliceIndex, 0,
0].SetTransform3D(t3d);
                        FrontCubeObjectY[SliceIndex, 0,
1].SetTransform3D(t3d);
                        FrontCubeObjectY[SliceIndex, 0,
2].SetTransform3D(t3d);

                        FrontCubeObjectY[SliceIndex, 1,
0].SetTransform3D(t3d);
                        if (FrontCubeObjectY[SliceIndex, 1, 1] != null)
                            FrontCubeObjectY[SliceIndex, 1,
1].SetTransform3D(t3d);
                        FrontCubeObjectY[SliceIndex, 1,
2].SetTransform3D(t3d);

                        FrontCubeObjectY[SliceIndex, 2,
0].SetTransform3D(t3d);
                        FrontCubeObjectY[SliceIndex, 2,
1].SetTransform3D(t3d);
                        FrontCubeObjectY[SliceIndex, 2,
2].SetTransform3D(t3d);


                        BackCubeObjectY[SliceIndex, 0, 0].SetTransform3D(t3d);
                        BackCubeObjectY[SliceIndex, 0, 1].SetTransform3D(t3d);
                        BackCubeObjectY[SliceIndex, 0, 2].SetTransform3D(t3d);

                        BackCubeObjectY[SliceIndex, 1, 0].SetTransform3D(t3d);
                        if (BackCubeObjectY[SliceIndex, 1, 1] != null)
                            BackCubeObjectY[SliceIndex, 1,
1].SetTransform3D(t3d);
                        BackCubeObjectY[SliceIndex, 1, 2].SetTransform3D(t3d);

                        BackCubeObjectY[SliceIndex, 2, 0].SetTransform3D(t3d);
                        BackCubeObjectY[SliceIndex, 2, 1].SetTransform3D(t3d);
                        BackCubeObjectY[SliceIndex, 2, 2].SetTransform3D(t3d);

                        da.By = 90;
                        da.Duration = new Duration(new TimeSpan(0, 0, 0, 0,
500));
                        da.RepeatBehavior = new RepeatBehavior(1);

Rotation.BeginAnimation(AxisAngleRotation3D.AngleProperty, da);
                    }
                    else
                    {
                        DoubleAnimation da = new DoubleAnimation();
                        da.Completed += new EventHandler(da_Completed);

                        Transform3D t3d = new RotateTransform3D(Rotation);
                        t3d = new RotateTransform3D(Rotation);
                        for (int i = 0; i < 3; i++)
                        {
                            FrontCubeObjectY[i, 0, 0].SetTransform3D(t3d);
                            FrontCubeObjectY[i, 0, 1].SetTransform3D(t3d);
```

```
                            FrontCubeObjectY[i, 0, 2].SetTransform3D(t3d);

                            FrontCubeObjectY[i, 1, 0].SetTransform3D(t3d);
                            if (FrontCubeObjectY[i, 1, 1] != null)
                                FrontCubeObjectY[i, 1, 1].SetTransform3D(t3d);
                            FrontCubeObjectY[i, 1, 2].SetTransform3D(t3d);

                            FrontCubeObjectY[i, 2, 0].SetTransform3D(t3d);
                            FrontCubeObjectY[i, 2, 1].SetTransform3D(t3d);
                            FrontCubeObjectY[i, 2, 2].SetTransform3D(t3d);

                            BackCubeObjectY[i, 0, 0].SetTransform3D(t3d);
                            BackCubeObjectY[i, 0, 1].SetTransform3D(t3d);
                            BackCubeObjectY[i, 0, 2].SetTransform3D(t3d);

                            BackCubeObjectY[i, 1, 0].SetTransform3D(t3d);
                            if (BackCubeObjectY[i, 1, 1] != null)
                                BackCubeObjectY[i, 1, 1].SetTransform3D(t3d);
                            BackCubeObjectY[i, 1, 2].SetTransform3D(t3d);

                            BackCubeObjectY[i, 2, 0].SetTransform3D(t3d);
                            BackCubeObjectY[i, 2, 1].SetTransform3D(t3d);
                            BackCubeObjectY[i, 2, 2].SetTransform3D(t3d);
                        }

                        da.By = 90;
                        da.Duration = new Duration(new TimeSpan(0, 0, 0, 0,
500));
                        da.RepeatBehavior = new RepeatBehavior(1);

Rotation.BeginAnimation(AxisAngleRotation3D.AngleProperty, da);
                    }

                }
                else
                {
                    AxisAngleRotation3D Rotation = new AxisAngleRotation3D();
                    Rotation.Axis = (bClockWise == true) ? new Vector3D(0, 0,
1) : new Vector3D(0, 0, -1);

                    if (SliceIndex != 3)
                    {
                        DoubleAnimation da = new DoubleAnimation();
                        da.Completed += new EventHandler(da_Completed);

                        Transform3D t3d = new RotateTransform3D(Rotation);
                        t3d = new RotateTransform3D(Rotation);

                        FrontCubeObjectZ[SliceIndex, 0,
0].SetTransform3D(t3d);
                        FrontCubeObjectZ[SliceIndex, 0,
1].SetTransform3D(t3d);
                        FrontCubeObjectZ[SliceIndex, 0,
2].SetTransform3D(t3d);

                        FrontCubeObjectZ[SliceIndex, 1,
0].SetTransform3D(t3d);
```

```
                        if (FrontCubeObjectZ[SliceIndex, 1, 1] != null)
                            FrontCubeObjectZ[SliceIndex, 1,
1].SetTransform3D(t3d);
                        FrontCubeObjectZ[SliceIndex, 1,
2].SetTransform3D(t3d);

                        FrontCubeObjectZ[SliceIndex, 2,
0].SetTransform3D(t3d);
                        FrontCubeObjectZ[SliceIndex, 2,
1].SetTransform3D(t3d);
                        FrontCubeObjectZ[SliceIndex, 2,
2].SetTransform3D(t3d);

                        BackCubeObjectZ[SliceIndex, 0, 0].SetTransform3D(t3d);
                        BackCubeObjectZ[SliceIndex, 0, 1].SetTransform3D(t3d);
                        BackCubeObjectZ[SliceIndex, 0, 2].SetTransform3D(t3d);

                        BackCubeObjectZ[SliceIndex, 1, 0].SetTransform3D(t3d);
                        if (BackCubeObjectZ[SliceIndex, 1, 1] != null)
                            BackCubeObjectZ[SliceIndex, 1,
1].SetTransform3D(t3d);
                        BackCubeObjectZ[SliceIndex, 1, 2].SetTransform3D(t3d);

                        BackCubeObjectZ[SliceIndex, 2, 0].SetTransform3D(t3d);
                        BackCubeObjectZ[SliceIndex, 2, 1].SetTransform3D(t3d);
                        BackCubeObjectZ[SliceIndex, 2, 2].SetTransform3D(t3d);

                        da.By = 90;
                        da.Duration = new Duration(new TimeSpan(0, 0, 0, 0,
500));
                        da.RepeatBehavior = new RepeatBehavior(1);

Rotation.BeginAnimation(AxisAngleRotation3D.AngleProperty, da);
                    }
                    else
                    {
                        DoubleAnimation da = new DoubleAnimation();
                        da.Completed += new EventHandler(da_Completed);

                        Transform3D t3d = new RotateTransform3D(Rotation);
                        t3d = new RotateTransform3D(Rotation);

                        for (int i = 0; i < 3; i++)
                        {
                            FrontCubeObjectZ[i, 0, 0].SetTransform3D(t3d);
                            FrontCubeObjectZ[i, 0, 1].SetTransform3D(t3d);
                            FrontCubeObjectZ[i, 0, 2].SetTransform3D(t3d);

                            FrontCubeObjectZ[i, 1, 0].SetTransform3D(t3d);
                            if (FrontCubeObjectZ[i, 1, 1] != null)
                                FrontCubeObjectZ[i, 1, 1].SetTransform3D(t3d);
                            FrontCubeObjectZ[i, 1, 2].SetTransform3D(t3d);

                            FrontCubeObjectZ[i, 2, 0].SetTransform3D(t3d);
                            FrontCubeObjectZ[i, 2, 1].SetTransform3D(t3d);
                            FrontCubeObjectZ[i, 2, 2].SetTransform3D(t3d);
```

```
                            BackCubeObjectZ[i, 0, 0].SetTransform3D(t3d);
                            BackCubeObjectZ[i, 0, 1].SetTransform3D(t3d);
                            BackCubeObjectZ[i, 0, 2].SetTransform3D(t3d);

                            BackCubeObjectZ[i, 1, 0].SetTransform3D(t3d);
                            if (BackCubeObjectZ[i, 1, 1] != null)
                                BackCubeObjectZ[i, 1, 1].SetTransform3D(t3d);
                            BackCubeObjectZ[i, 1, 2].SetTransform3D(t3d);

                            BackCubeObjectZ[i, 2, 0].SetTransform3D(t3d);
                            BackCubeObjectZ[i, 2, 1].SetTransform3D(t3d);
                            BackCubeObjectZ[i, 2, 2].SetTransform3D(t3d);
                        }

                        da.By = 90;
                        da.Duration = new Duration(new TimeSpan(0, 0, 0, 0,
500));
                        da.RepeatBehavior = new RepeatBehavior(1);

Rotation.BeginAnimation(AxisAngleRotation3D.AngleProperty, da);
                    }
                }
            }
        }

        void da_Completed(object sender, EventArgs e)
        {
            // use Mutex, so only one thread can access that code else if you
rotate
            // many times (push Rotate button very fast) it will break the
application
            using (Mutex mutex = new Mutex(false, "da_Completed"))
            {
                if (bClockWise == true)
                    da_CompletedCW();
                else
                    da_CompletedCCW();

                if (nShuffle > 0)
                    Automate();
                else
                    EnableControl(true);
            }
        }

        private void EnableControl(bool bEnable)
        {
            simpleButtonCW.IsEnabled = bEnable;
            simpleButtonCCW.IsEnabled = bEnable;
            AxisCB.IsEnabled = bEnable;
            SliceCB.IsEnabled = bEnable;
            simpleButtonShuffle.IsEnabled = bEnable;

            simpleButtonUndo.IsEnabled = (bEnable && UndoStack.Count > 0);
            simpleButtonRedo.IsEnabled = (bEnable && RedoStack.Count > 0);

        }
```

```
        void Automate()
        {
            int Axis = ran.Next(3);
            AxisCB.SelectedIndex = Axis;
            int Slice = ran.Next(4);
            SliceCB.SelectedIndex = Slice;
            int rotation = ran.Next(2);
            bClockWise = rotation != 0 ? true : false;
            RotateTheCube();
            nShuffle--;
        }

        void da_CompletedCW()
        {
            int AxisIndex = AxisCB.SelectedIndex;
            int slice = SliceCB.SelectedIndex;


            CubeObject[, ,] FrontTempCubeObjectX = new CubeObject[3, 3, 3];
            CubeObject[, ,] FrontTempCubeObjectY = new CubeObject[3, 3, 3];
            CubeObject[, ,] FrontTempCubeObjectZ = new CubeObject[3, 3, 3];

            CubeObject[, ,] BackTempCubeObjectX = new CubeObject[3, 3, 3];
            CubeObject[, ,] BackTempCubeObjectY = new CubeObject[3, 3, 3];
            CubeObject[, ,] BackTempCubeObjectZ = new CubeObject[3, 3, 3];

            // first copy every thing
            for (int i = 0; i < 3; i++)
            {
                for (int j = 0; j < 3; j++)
                {
                    for (int k = 0; k < 3; k++)
                    {
                        FrontTempCubeObjectX[i, j, k] = new
CubeObject(FrontCubeObjectX[i, j, k]);
                        FrontTempCubeObjectY[i, j, k] = new
CubeObject(FrontCubeObjectY[i, j, k]);
                        FrontTempCubeObjectZ[i, j, k] = new
CubeObject(FrontCubeObjectZ[i, j, k]);

                        BackTempCubeObjectX[i, j, k] = new
CubeObject(BackCubeObjectX[i, j, k]);
                        BackTempCubeObjectY[i, j, k] = new
CubeObject(BackCubeObjectY[i, j, k]);
                        BackTempCubeObjectZ[i, j, k] = new
CubeObject(BackCubeObjectZ[i, j, k]);
                    }
                }
            }

            if (AxisIndex == 0) // x-axis
            {
                if (slice == 3)
                {
                    string strLog = nGlobleIndex.ToString() + "  CW X-axis
All";
                    if (bLogFile == true)
```

```
                        ErrorLog(strLog);

                    //da_CompletedCW
                    for (int i = 0; i < 3; i++)
                    {
                        FrontCubeObjectX[i, 0, 0] = new
CubeObject(FrontTempCubeObjectX[i, 0, 2]);
                        FrontCubeObjectX[i, 1, 0] = new
CubeObject(FrontTempCubeObjectX[i, 0, 1]);
                        FrontCubeObjectX[i, 2, 0] = new
CubeObject(FrontTempCubeObjectX[i, 0, 0]);
                        FrontCubeObjectX[i, 0, 1] = new
CubeObject(FrontTempCubeObjectX[i, 1, 2]);
                        FrontCubeObjectX[i, 1, 1] = (FrontTempCubeObjectX[i,
1, 1] == null) ? null : new CubeObject(FrontTempCubeObjectX[i, 1, 1]);
                        FrontCubeObjectX[i, 2, 1] = new
CubeObject(FrontTempCubeObjectX[i, 1, 0]);
                        FrontCubeObjectX[i, 0, 2] = new
CubeObject(FrontTempCubeObjectX[i, 2, 2]);
                        FrontCubeObjectX[i, 1, 2] = new
CubeObject(FrontTempCubeObjectX[i, 2, 1]);
                        FrontCubeObjectX[i, 2, 2] = new
CubeObject(FrontTempCubeObjectX[i, 2, 0]);

                        BackCubeObjectX[i, 0, 0] = new
CubeObject(BackTempCubeObjectX[i, 0, 2]);
                        BackCubeObjectX[i, 1, 0] = new
CubeObject(BackTempCubeObjectX[i, 0, 1]);
                        BackCubeObjectX[i, 2, 0] = new
CubeObject(BackTempCubeObjectX[i, 0, 0]);
                        BackCubeObjectX[i, 0, 1] = new
CubeObject(BackTempCubeObjectX[i, 1, 2]);
                        BackCubeObjectX[i, 1, 1] = (BackTempCubeObjectX[i, 1,
1] == null) ? null : new CubeObject(BackTempCubeObjectX[i, 1, 1]);
                        BackCubeObjectX[i, 2, 1] = new
CubeObject(BackTempCubeObjectX[i, 1, 0]);
                        BackCubeObjectX[i, 0, 2] = new
CubeObject(BackTempCubeObjectX[i, 2, 2]);
                        BackCubeObjectX[i, 1, 2] = new
CubeObject(BackTempCubeObjectX[i, 2, 1]);
                        BackCubeObjectX[i, 2, 2] = new
CubeObject(BackTempCubeObjectX[i, 2, 0]);
                    }
                    CopyXSlicesToYSlices();
                    CopyXSlicesToZSlices();
                }
                else
                {
                    string strLog = nGlobleIndex.ToString() + "  CW  X-axis
Slice = ";
                    if (slice == 0)
                        strLog += "First";
                    else if (slice == 0)
                        strLog += "Second";
                    else strLog += "Third";

                    if (bLogFile == true)
```

```
                    ErrorLog(strLog);

                //da_CompletedCW
                FrontCubeObjectX[slice, 0, 0] = new
CubeObject(FrontTempCubeObjectX[slice, 0, 2]);
                FrontCubeObjectX[slice, 1, 0] = new
CubeObject(FrontTempCubeObjectX[slice, 0, 1]);
                FrontCubeObjectX[slice, 2, 0] = new
CubeObject(FrontTempCubeObjectX[slice, 0, 0]);
                FrontCubeObjectX[slice, 0, 1] = new
CubeObject(FrontTempCubeObjectX[slice, 1, 2]);
                FrontCubeObjectX[slice, 1, 1] =
(FrontTempCubeObjectX[slice, 1, 1] == null) ? null : new
CubeObject(FrontTempCubeObjectX[slice, 1, 1]);
                FrontCubeObjectX[slice, 2, 1] = new
CubeObject(FrontTempCubeObjectX[slice, 1, 0]);
                FrontCubeObjectX[slice, 0, 2] = new
CubeObject(FrontTempCubeObjectX[slice, 2, 2]);
                FrontCubeObjectX[slice, 1, 2] = new
CubeObject(FrontTempCubeObjectX[slice, 2, 1]);
                FrontCubeObjectX[slice, 2, 2] = new
CubeObject(FrontTempCubeObjectX[slice, 2, 0]);

                FrontCubeObjectY[0, slice, 0] = new
CubeObject(FrontTempCubeObjectY[0, slice, 2]);
                FrontCubeObjectY[0, slice, 1] = new
CubeObject(FrontTempCubeObjectY[1, slice, 2]);
                FrontCubeObjectY[0, slice, 2] = new
CubeObject(FrontTempCubeObjectY[2, slice, 2]);
                FrontCubeObjectY[1, slice, 0] = new
CubeObject(FrontTempCubeObjectY[0, slice, 1]);
                FrontCubeObjectY[1, slice, 1] = (FrontTempCubeObjectY[1,
slice, 1] == null) ? null : new CubeObject(FrontTempCubeObjectY[1, slice, 1]);
                FrontCubeObjectY[1, slice, 2] = new
CubeObject(FrontTempCubeObjectY[2, slice, 1]);
                FrontCubeObjectY[2, slice, 0] = new
CubeObject(FrontTempCubeObjectY[0, slice, 0]);
                FrontCubeObjectY[2, slice, 1] = new
CubeObject(FrontTempCubeObjectY[1, slice, 0]);
                FrontCubeObjectY[2, slice, 2] = new
CubeObject(FrontTempCubeObjectY[2, slice, 0]);

                FrontCubeObjectZ[0, slice, 0] = new
CubeObject(FrontTempCubeObjectZ[2, slice, 0]);
                FrontCubeObjectZ[0, slice, 1] = new
CubeObject(FrontTempCubeObjectZ[1, slice, 0]);
                FrontCubeObjectZ[0, slice, 2] = new
CubeObject(FrontTempCubeObjectZ[0, slice, 0]);
                FrontCubeObjectZ[1, slice, 0] = new
CubeObject(FrontTempCubeObjectZ[2, slice, 1]);
                FrontCubeObjectZ[1, slice, 1] = (FrontTempCubeObjectZ[1,
slice, 1] == null) ? null : new CubeObject(FrontTempCubeObjectZ[1, slice, 1]);
                FrontCubeObjectZ[1, slice, 2] = new
CubeObject(FrontTempCubeObjectZ[0, slice, 1]);
                FrontCubeObjectZ[2, slice, 0] = new
CubeObject(FrontTempCubeObjectZ[2, slice, 2]);
```

```
                    FrontCubeObjectZ[2, slice, 1] = new
CubeObject(FrontTempCubeObjectZ[1, slice, 2]);
                    FrontCubeObjectZ[2, slice, 2] = new
CubeObject(FrontTempCubeObjectZ[0, slice, 2]);


                    BackCubeObjectX[slice, 0, 0] = new
CubeObject(BackTempCubeObjectX[slice, 0, 2]);
                    BackCubeObjectX[slice, 1, 0] = new
CubeObject(BackTempCubeObjectX[slice, 0, 1]);
                    BackCubeObjectX[slice, 2, 0] = new
CubeObject(BackTempCubeObjectX[slice, 0, 0]);
                    BackCubeObjectX[slice, 0, 1] = new
CubeObject(BackTempCubeObjectX[slice, 1, 2]);
                    BackCubeObjectX[slice, 1, 1] = (BackTempCubeObjectX[slice,
1, 1] == null) ? null : new CubeObject(BackTempCubeObjectX[slice, 1, 1]);
                    BackCubeObjectX[slice, 2, 1] = new
CubeObject(BackTempCubeObjectX[slice, 1, 0]);
                    BackCubeObjectX[slice, 0, 2] = new
CubeObject(BackTempCubeObjectX[slice, 2, 2]);
                    BackCubeObjectX[slice, 1, 2] = new
CubeObject(BackTempCubeObjectX[slice, 2, 1]);
                    BackCubeObjectX[slice, 2, 2] = new
CubeObject(BackTempCubeObjectX[slice, 2, 0]);

                    BackCubeObjectY[0, slice, 0] = new
CubeObject(BackTempCubeObjectY[0, slice, 2]);
                    BackCubeObjectY[0, slice, 1] = new
CubeObject(BackTempCubeObjectY[1, slice, 2]);
                    BackCubeObjectY[0, slice, 2] = new
CubeObject(BackTempCubeObjectY[2, slice, 2]);
                    BackCubeObjectY[1, slice, 0] = new
CubeObject(BackTempCubeObjectY[0, slice, 1]);
                    BackCubeObjectY[1, slice, 1] = (BackTempCubeObjectY[1,
slice, 1] == null) ? null : new CubeObject(BackTempCubeObjectY[1, slice, 1]);
                    BackCubeObjectY[1, slice, 2] = new
CubeObject(BackTempCubeObjectY[2, slice, 1]);
                    BackCubeObjectY[2, slice, 0] = new
CubeObject(BackTempCubeObjectY[0, slice, 0]);
                    BackCubeObjectY[2, slice, 1] = new
CubeObject(BackTempCubeObjectY[1, slice, 0]);
                    BackCubeObjectY[2, slice, 2] = new
CubeObject(BackTempCubeObjectY[2, slice, 0]);

                    BackCubeObjectZ[0, slice, 0] = new
CubeObject(BackTempCubeObjectZ[2, slice, 0]);
                    BackCubeObjectZ[0, slice, 1] = new
CubeObject(BackTempCubeObjectZ[1, slice, 0]);
                    BackCubeObjectZ[0, slice, 2] = new
CubeObject(BackTempCubeObjectZ[0, slice, 0]);
                    BackCubeObjectZ[1, slice, 0] = new
CubeObject(BackTempCubeObjectZ[2, slice, 1]);
                    BackCubeObjectZ[1, slice, 1] = (BackTempCubeObjectZ[1,
slice, 1] == null) ? null : new CubeObject(BackTempCubeObjectZ[1, slice, 1]);
                    BackCubeObjectZ[1, slice, 2] = new
CubeObject(BackTempCubeObjectZ[0, slice, 1]);
```

```
                    BackCubeObjectZ[2, slice, 0] = new
CubeObject(BackTempCubeObjectZ[2, slice, 2]);
                    BackCubeObjectZ[2, slice, 1] = new
CubeObject(BackTempCubeObjectZ[1, slice, 2]);
                    BackCubeObjectZ[2, slice, 2] = new
CubeObject(BackTempCubeObjectZ[0, slice, 2]);
                }

            }
            else if (AxisIndex == 1) // y-axis
            {
                if (slice == 3)
                {
                    string strLog = nGlobleIndex.ToString() + " CCW Y-axis
All";
                    if (bLogFile == true)
                        ErrorLog(strLog);

                    for (int i = 0; i < 3; i++)
                    {
                        FrontCubeObjectY[i, 0, 0] = new
CubeObject(FrontTempCubeObjectY[i, 0, 2]);
                        FrontCubeObjectY[i, 1, 0] = new
CubeObject(FrontTempCubeObjectY[i, 0, 1]);
                        FrontCubeObjectY[i, 2, 0] = new
CubeObject(FrontTempCubeObjectY[i, 0, 0]);
                        FrontCubeObjectY[i, 0, 1] = new
CubeObject(FrontTempCubeObjectY[i, 1, 2]);
                        FrontCubeObjectY[i, 1, 1] = (FrontTempCubeObjectY[i,
1, 1] == null) ? null : new CubeObject(FrontTempCubeObjectY[i, 1, 1]);
                        FrontCubeObjectY[i, 2, 1] = new
CubeObject(FrontTempCubeObjectY[i, 1, 0]);
                        FrontCubeObjectY[i, 0, 2] = new
CubeObject(FrontTempCubeObjectY[i, 2, 2]);
                        FrontCubeObjectY[i, 1, 2] = new
CubeObject(FrontTempCubeObjectY[i, 2, 1]);
                        FrontCubeObjectY[i, 2, 2] = new
CubeObject(FrontTempCubeObjectY[i, 2, 0]);

                        BackCubeObjectY[i, 0, 0] = new
CubeObject(BackTempCubeObjectY[i, 0, 2]);
                        BackCubeObjectY[i, 1, 0] = new
CubeObject(BackTempCubeObjectY[i, 0, 1]);
                        BackCubeObjectY[i, 2, 0] = new
CubeObject(BackTempCubeObjectY[i, 0, 0]);
                        BackCubeObjectY[i, 0, 1] = new
CubeObject(BackTempCubeObjectY[i, 1, 2]);
                        BackCubeObjectY[i, 1, 1] = (BackTempCubeObjectY[i, 1,
1] == null) ? null : new CubeObject(BackTempCubeObjectY[i, 1, 1]);
                        BackCubeObjectY[i, 2, 1] = new
CubeObject(BackTempCubeObjectY[i, 1, 0]);
                        BackCubeObjectY[i, 0, 2] = new
CubeObject(BackTempCubeObjectY[i, 2, 2]);
                        BackCubeObjectY[i, 1, 2] = new
CubeObject(BackTempCubeObjectY[i, 2, 1]);
                        BackCubeObjectY[i, 2, 2] = new
CubeObject(BackTempCubeObjectY[i, 2, 0]);
```

```
                    }
                    CopyYSlicesToXSlices();
                    CopyYSlicesToZSlices();
                }
                else
                {
                    //da_CompletedCCW() y,x,z
                    string strLog = nGlobleIndex.ToString() + " CCW  Y-axis
Slice = ";
                    if (slice == 0)
                        strLog += "First";
                    else if (slice == 0)
                        strLog += "Second";
                    else strLog += "Third";

                    if (bLogFile == true)
                        ErrorLog(strLog);

                    FrontCubeObjectY[slice, 0, 0] = new
CubeObject(FrontTempCubeObjectY[slice, 0, 2]);
                    FrontCubeObjectY[slice, 1, 0] = new
CubeObject(FrontTempCubeObjectY[slice, 0, 1]);
                    FrontCubeObjectY[slice, 2, 0] = new
CubeObject(FrontTempCubeObjectY[slice, 0, 0]);
                    FrontCubeObjectY[slice, 0, 1] = new
CubeObject(FrontTempCubeObjectY[slice, 1, 2]);
                    FrontCubeObjectY[slice, 1, 1] =
(FrontTempCubeObjectY[slice, 1, 1] == null) ? null : new
CubeObject(FrontTempCubeObjectY[slice, 1, 1]);
                    FrontCubeObjectY[slice, 2, 1] = new
CubeObject(FrontTempCubeObjectY[slice, 1, 0]);
                    FrontCubeObjectY[slice, 0, 2] = new
CubeObject(FrontTempCubeObjectY[slice, 2, 2]);
                    FrontCubeObjectY[slice, 1, 2] = new
CubeObject(FrontTempCubeObjectY[slice, 2, 1]);
                    FrontCubeObjectY[slice, 2, 2] = new
CubeObject(FrontTempCubeObjectY[slice, 2, 0]);

                    FrontCubeObjectX[0, slice, 0] = new
CubeObject(FrontTempCubeObjectX[0, slice, 2]);
                    FrontCubeObjectX[0, slice, 1] = new
CubeObject(FrontTempCubeObjectX[1, slice, 2]);
                    FrontCubeObjectX[0, slice, 2] = new
CubeObject(FrontTempCubeObjectX[2, slice, 2]);
                    FrontCubeObjectX[1, slice, 0] = new
CubeObject(FrontTempCubeObjectX[0, slice, 1]);
                    FrontCubeObjectX[1, slice, 1] = (FrontTempCubeObjectX[1,
slice, 1] == null) ? null : new CubeObject(FrontTempCubeObjectX[1, slice, 1]);
                    FrontCubeObjectX[1, slice, 2] = new
CubeObject(FrontTempCubeObjectX[2, slice, 1]);
                    FrontCubeObjectX[2, slice, 0] = new
CubeObject(FrontTempCubeObjectX[0, slice, 0]);
                    FrontCubeObjectX[2, slice, 1] = new
CubeObject(FrontTempCubeObjectX[1, slice, 0]);
                    FrontCubeObjectX[2, slice, 2] = new
CubeObject(FrontTempCubeObjectX[2, slice, 0]);
```

```
                    FrontCubeObjectZ[0, 0, slice] = new
CubeObject(FrontTempCubeObjectZ[2, 0, slice]);
                    FrontCubeObjectZ[0, 1, slice] = new
CubeObject(FrontTempCubeObjectZ[1, 0, slice]);
                    FrontCubeObjectZ[0, 2, slice] = new
CubeObject(FrontTempCubeObjectZ[0, 0, slice]);
                    FrontCubeObjectZ[1, 0, slice] = new
CubeObject(FrontTempCubeObjectZ[2, 1, slice]);
                    FrontCubeObjectZ[1, 1, slice] = (FrontTempCubeObjectZ[1,
1, slice] == null) ? null : new CubeObject(FrontTempCubeObjectZ[1, 1, slice]);
                    FrontCubeObjectZ[1, 2, slice] = new
CubeObject(FrontTempCubeObjectZ[0, 1, slice]);
                    FrontCubeObjectZ[2, 0, slice] = new
CubeObject(FrontTempCubeObjectZ[2, 2, slice]);
                    FrontCubeObjectZ[2, 1, slice] = new
CubeObject(FrontTempCubeObjectZ[1, 2, slice]);
                    FrontCubeObjectZ[2, 2, slice] = new
CubeObject(FrontTempCubeObjectZ[0, 2, slice]);

                    BackCubeObjectY[slice, 0, 0] = new
CubeObject(BackTempCubeObjectY[slice, 0, 2]);
                    BackCubeObjectY[slice, 1, 0] = new
CubeObject(BackTempCubeObjectY[slice, 0, 1]);
                    BackCubeObjectY[slice, 2, 0] = new
CubeObject(BackTempCubeObjectY[slice, 0, 0]);
                    BackCubeObjectY[slice, 0, 1] = new
CubeObject(BackTempCubeObjectY[slice, 1, 2]);
                    BackCubeObjectY[slice, 1, 1] = (BackTempCubeObjectY[slice,
1, 1] == null) ? null : new CubeObject(BackTempCubeObjectY[slice, 1, 1]);
                    BackCubeObjectY[slice, 2, 1] = new
CubeObject(BackTempCubeObjectY[slice, 1, 0]);
                    BackCubeObjectY[slice, 0, 2] = new
CubeObject(BackTempCubeObjectY[slice, 2, 2]);
                    BackCubeObjectY[slice, 1, 2] = new
CubeObject(BackTempCubeObjectY[slice, 2, 1]);
                    BackCubeObjectY[slice, 2, 2] = new
CubeObject(BackTempCubeObjectY[slice, 2, 0]);

                    BackCubeObjectX[0, slice, 0] = new
CubeObject(BackTempCubeObjectX[0, slice, 2]);
                    BackCubeObjectX[0, slice, 1] = new
CubeObject(BackTempCubeObjectX[1, slice, 2]);
                    BackCubeObjectX[0, slice, 2] = new
CubeObject(BackTempCubeObjectX[2, slice, 2]);
                    BackCubeObjectX[1, slice, 0] = new
CubeObject(BackTempCubeObjectX[0, slice, 1]);
                    BackCubeObjectX[1, slice, 1] = (BackTempCubeObjectX[1,
slice, 1] == null) ? null : new CubeObject(BackTempCubeObjectX[1, slice, 1]);
                    BackCubeObjectX[1, slice, 2] = new
CubeObject(BackTempCubeObjectX[2, slice, 1]);
                    BackCubeObjectX[2, slice, 0] = new
CubeObject(BackTempCubeObjectX[0, slice, 0]);
                    BackCubeObjectX[2, slice, 1] = new
CubeObject(BackTempCubeObjectX[1, slice, 0]);
                    BackCubeObjectX[2, slice, 2] = new
CubeObject(BackTempCubeObjectX[2, slice, 0]);
```

```
                    BackCubeObjectZ[0, 0, slice] = new
CubeObject(BackTempCubeObjectZ[2, 0, slice]);
                    BackCubeObjectZ[0, 1, slice] = new
CubeObject(BackTempCubeObjectZ[1, 0, slice]);
                    BackCubeObjectZ[0, 2, slice] = new
CubeObject(BackTempCubeObjectZ[0, 0, slice]);
                    BackCubeObjectZ[1, 0, slice] = new
CubeObject(BackTempCubeObjectZ[2, 1, slice]);
                    BackCubeObjectZ[1, 1, slice] = (BackTempCubeObjectZ[1, 1,
slice] == null) ? null : new CubeObject(BackTempCubeObjectZ[1, 1, slice]);
                    BackCubeObjectZ[1, 2, slice] = new
CubeObject(BackTempCubeObjectZ[0, 1, slice]);
                    BackCubeObjectZ[2, 0, slice] = new
CubeObject(BackTempCubeObjectZ[2, 2, slice]);
                    BackCubeObjectZ[2, 1, slice] = new
CubeObject(BackTempCubeObjectZ[1, 2, slice]);
                    BackCubeObjectZ[2, 2, slice] = new
CubeObject(BackTempCubeObjectZ[0, 2, slice]);
                }
            }
            else // z-axis
            {
                if (slice == 3)
                {
                    string strLog = nGlobleIndex.ToString() + "  CW Z-axis
All";
                    if (bLogFile == true)
                        ErrorLog(strLog);

                    for (int i = 0; i < 3; i++)
                    {
                        //da_CompletedCW
                        FrontCubeObjectZ[i, 0, 0] = new
CubeObject(FrontTempCubeObjectZ[i, 0, 2]);
                        FrontCubeObjectZ[i, 1, 0] = new
CubeObject(FrontTempCubeObjectZ[i, 0, 1]);
                        FrontCubeObjectZ[i, 2, 0] = new
CubeObject(FrontTempCubeObjectZ[i, 0, 0]);
                        FrontCubeObjectZ[i, 0, 1] = new
CubeObject(FrontTempCubeObjectZ[i, 1, 2]);
                        FrontCubeObjectZ[i, 1, 1] = (FrontTempCubeObjectZ[i,
1, 1] == null) ? null : new CubeObject(FrontTempCubeObjectZ[i, 1, 1]);
                        FrontCubeObjectZ[i, 2, 1] = new
CubeObject(FrontTempCubeObjectZ[i, 1, 0]);
                        FrontCubeObjectZ[i, 0, 2] = new
CubeObject(FrontTempCubeObjectZ[i, 2, 2]);
                        FrontCubeObjectZ[i, 1, 2] = new
CubeObject(FrontTempCubeObjectZ[i, 2, 1]);
                        FrontCubeObjectZ[i, 2, 2] = new
CubeObject(FrontTempCubeObjectZ[i, 2, 0]);

                        BackCubeObjectZ[i, 0, 0] = new
CubeObject(BackTempCubeObjectZ[i, 0, 2]);
                        BackCubeObjectZ[i, 1, 0] = new
CubeObject(BackTempCubeObjectZ[i, 0, 1]);
                        BackCubeObjectZ[i, 2, 0] = new
CubeObject(BackTempCubeObjectZ[i, 0, 0]);
```

```
                        BackCubeObjectZ[i, 0, 1] = new
CubeObject(BackTempCubeObjectZ[i, 1, 2]);
                        BackCubeObjectZ[i, 1, 1] = (BackTempCubeObjectZ[i, 1,
1] == null) ? null : new CubeObject(BackTempCubeObjectZ[i, 1, 1]);
                        BackCubeObjectZ[i, 2, 1] = new
CubeObject(BackTempCubeObjectZ[i, 1, 0]);
                        BackCubeObjectZ[i, 0, 2] = new
CubeObject(BackTempCubeObjectZ[i, 2, 2]);
                        BackCubeObjectZ[i, 1, 2] = new
CubeObject(BackTempCubeObjectZ[i, 2, 1]);
                        BackCubeObjectZ[i, 2, 2] = new
CubeObject(BackTempCubeObjectZ[i, 2, 0]);

                    }
                    CopyZSlicesToXSlices();
                    CopyZSlicesToYSlices();

                }
                else
                {
                    string strLog = nGlobleIndex.ToString() + "  CW  Z-axis
Slice = ";
                    if (slice == 0)
                        strLog += "First";
                    else if (slice == 0)
                        strLog += "Second";
                    else strLog += "Third";
                    if (bLogFile == true)
                        ErrorLog(strLog);

                    //da_CompletedCW
                    FrontCubeObjectZ[slice, 0, 0] = new
CubeObject(FrontTempCubeObjectZ[slice, 0, 2]);
                    FrontCubeObjectZ[slice, 1, 0] = new
CubeObject(FrontTempCubeObjectZ[slice, 0, 1]);
                    FrontCubeObjectZ[slice, 2, 0] = new
CubeObject(FrontTempCubeObjectZ[slice, 0, 0]);
                    FrontCubeObjectZ[slice, 0, 1] = new
CubeObject(FrontTempCubeObjectZ[slice, 1, 2]);
                    FrontCubeObjectZ[slice, 1, 1] =
(FrontTempCubeObjectZ[slice, 1, 1] == null) ? null : new
CubeObject(FrontTempCubeObjectZ[slice, 1, 1]);
                    FrontCubeObjectZ[slice, 2, 1] = new
CubeObject(FrontTempCubeObjectZ[slice, 1, 0]);
                    FrontCubeObjectZ[slice, 0, 2] = new
CubeObject(FrontTempCubeObjectZ[slice, 2, 2]);
                    FrontCubeObjectZ[slice, 1, 2] = new
CubeObject(FrontTempCubeObjectZ[slice, 2, 1]);
                    FrontCubeObjectZ[slice, 2, 2] = new
CubeObject(FrontTempCubeObjectZ[slice, 2, 0]);

                    FrontCubeObjectX[0, 0, slice] = new
CubeObject(FrontTempCubeObjectX[0, 2, slice]);
                    FrontCubeObjectX[0, 1, slice] = new
CubeObject(FrontTempCubeObjectX[1, 2, slice]);
                    FrontCubeObjectX[0, 2, slice] = new
CubeObject(FrontTempCubeObjectX[2, 2, slice]);
```

```
                    FrontCubeObjectX[1, 0, slice] = new
CubeObject(FrontTempCubeObjectX[0, 1, slice]);
                    FrontCubeObjectX[1, 1, slice] = (FrontTempCubeObjectX[1,
1, slice] == null) ? null : new CubeObject(FrontTempCubeObjectX[1, 1, slice]);
                    FrontCubeObjectX[1, 2, slice] = new
CubeObject(FrontTempCubeObjectX[2, 1, slice]);
                    FrontCubeObjectX[2, 0, slice] = new
CubeObject(FrontTempCubeObjectX[0, 0, slice]);
                    FrontCubeObjectX[2, 1, slice] = new
CubeObject(FrontTempCubeObjectX[1, 0, slice]);
                    FrontCubeObjectX[2, 2, slice] = new
CubeObject(FrontTempCubeObjectX[2, 0, slice]);

                    FrontCubeObjectY[0, 0, slice] = new
CubeObject(FrontTempCubeObjectY[2, 0, slice]);
                    FrontCubeObjectY[0, 1, slice] = new
CubeObject(FrontTempCubeObjectY[1, 0, slice]);
                    FrontCubeObjectY[0, 2, slice] = new
CubeObject(FrontTempCubeObjectY[0, 0, slice]);
                    FrontCubeObjectY[1, 0, slice] = new
CubeObject(FrontTempCubeObjectY[2, 1, slice]);
                    FrontCubeObjectY[1, 1, slice] = (FrontTempCubeObjectY[1,
1, slice] == null) ? null : new CubeObject(FrontTempCubeObjectY[1, 1, slice]);
                    FrontCubeObjectY[1, 2, slice] = new
CubeObject(FrontTempCubeObjectY[0, 1, slice]);
                    FrontCubeObjectY[2, 0, slice] = new
CubeObject(FrontTempCubeObjectY[2, 2, slice]);
                    FrontCubeObjectY[2, 1, slice] = new
CubeObject(FrontTempCubeObjectY[1, 2, slice]);
                    FrontCubeObjectY[2, 2, slice] = new
CubeObject(FrontTempCubeObjectY[0, 2, slice]);

                    ////
                    BackCubeObjectZ[slice, 0, 0] = new
CubeObject(BackTempCubeObjectZ[slice, 0, 2]);
                    BackCubeObjectZ[slice, 1, 0] = new
CubeObject(BackTempCubeObjectZ[slice, 0, 1]);
                    BackCubeObjectZ[slice, 2, 0] = new
CubeObject(BackTempCubeObjectZ[slice, 0, 0]);
                    BackCubeObjectZ[slice, 0, 1] = new
CubeObject(BackTempCubeObjectZ[slice, 1, 2]);
                    BackCubeObjectZ[slice, 1, 1] = (BackTempCubeObjectZ[slice,
1, 1] == null) ? null : new CubeObject(BackTempCubeObjectZ[slice, 1, 1]);
                    BackCubeObjectZ[slice, 2, 1] = new
CubeObject(BackTempCubeObjectZ[slice, 1, 0]);
                    BackCubeObjectZ[slice, 0, 2] = new
CubeObject(BackTempCubeObjectZ[slice, 2, 2]);
                    BackCubeObjectZ[slice, 1, 2] = new
CubeObject(BackTempCubeObjectZ[slice, 2, 1]);
                    BackCubeObjectZ[slice, 2, 2] = new
CubeObject(BackTempCubeObjectZ[slice, 2, 0]);

                    BackCubeObjectX[0, 0, slice] = new
CubeObject(BackTempCubeObjectX[0, 2, slice]);
                    BackCubeObjectX[0, 1, slice] = new
CubeObject(BackTempCubeObjectX[1, 2, slice]);
```

```
                    BackCubeObjectX[0, 2, slice] = new
CubeObject(BackTempCubeObjectX[2, 2, slice]);
                    BackCubeObjectX[1, 0, slice] = new
CubeObject(BackTempCubeObjectX[0, 1, slice]);
                    BackCubeObjectX[1, 1, slice] = (BackTempCubeObjectX[1, 1,
slice] == null) ? null : new CubeObject(BackTempCubeObjectX[1, 1, slice]);
                    BackCubeObjectX[1, 2, slice] = new
CubeObject(BackTempCubeObjectX[2, 1, slice]);
                    BackCubeObjectX[2, 0, slice] = new
CubeObject(BackTempCubeObjectX[0, 0, slice]);
                    BackCubeObjectX[2, 1, slice] = new
CubeObject(BackTempCubeObjectX[1, 0, slice]);
                    BackCubeObjectX[2, 2, slice] = new
CubeObject(BackTempCubeObjectX[2, 0, slice]);

                    BackCubeObjectY[0, 0, slice] = new
CubeObject(BackTempCubeObjectY[2, 0, slice]);
                    BackCubeObjectY[0, 1, slice] = new
CubeObject(BackTempCubeObjectY[1, 0, slice]);
                    BackCubeObjectY[0, 2, slice] = new
CubeObject(BackTempCubeObjectY[0, 0, slice]);
                    BackCubeObjectY[1, 0, slice] = new
CubeObject(BackTempCubeObjectY[2, 1, slice]);
                    BackCubeObjectY[1, 1, slice] = (BackTempCubeObjectY[1, 1,
slice] == null) ? null : new CubeObject(BackTempCubeObjectY[1, 1, slice]);
                    BackCubeObjectY[1, 2, slice] = new
CubeObject(BackTempCubeObjectY[0, 1, slice]);
                    BackCubeObjectY[2, 0, slice] = new
CubeObject(BackTempCubeObjectY[2, 2, slice]);
                    BackCubeObjectY[2, 1, slice] = new
CubeObject(BackTempCubeObjectY[1, 2, slice]);
                    BackCubeObjectY[2, 2, slice] = new
CubeObject(BackTempCubeObjectY[0, 2, slice]);

                }
            }
            nGlobleIndex++;
        }
        void da_CompletedCCW()
        {
            int AxisIndex = AxisCB.SelectedIndex;
            int slice = SliceCB.SelectedIndex;


            CubeObject[, ,] FrontTempCubeObjectX = new CubeObject[3, 3, 3];
            CubeObject[, ,] FrontTempCubeObjectY = new CubeObject[3, 3, 3];
            CubeObject[, ,] FrontTempCubeObjectZ = new CubeObject[3, 3, 3];

            CubeObject[, ,] BackTempCubeObjectX = new CubeObject[3, 3, 3];
            CubeObject[, ,] BackTempCubeObjectY = new CubeObject[3, 3, 3];
            CubeObject[, ,] BackTempCubeObjectZ = new CubeObject[3, 3, 3];


            // first copy every thing
            for (int i = 0; i < 3; i++)
            {
                for (int j = 0; j < 3; j++)
```

```
                {
                    for (int k = 0; k < 3; k++)
                    {
                        FrontTempCubeObjectX[i, j, k] = new
CubeObject(FrontCubeObjectX[i, j, k]);
                        FrontTempCubeObjectY[i, j, k] = new
CubeObject(FrontCubeObjectY[i, j, k]);
                        FrontTempCubeObjectZ[i, j, k] = new
CubeObject(FrontCubeObjectZ[i, j, k]);

                        BackTempCubeObjectX[i, j, k] = new
CubeObject(BackCubeObjectX[i, j, k]);
                        BackTempCubeObjectY[i, j, k] = new
CubeObject(BackCubeObjectY[i, j, k]);
                        BackTempCubeObjectZ[i, j, k] = new
CubeObject(BackCubeObjectZ[i, j, k]);
                    }
                }
            }

            if (AxisIndex == 0) // x-axis
            {
                if (slice == 3)
                {
                    string strLog = nGlobleIndex.ToString() + " CCW X-axis
All";
                    if (bLogFile == true)
                        ErrorLog(strLog);

                    for (int i = 0; i < 3; i++)
                    {
                        //da_CompletedCCW() // x,y,z
                        FrontCubeObjectX[i, 0, 0] = new
CubeObject(FrontTempCubeObjectX[i, 2, 0]);
                        FrontCubeObjectX[i, 1, 0] = new
CubeObject(FrontTempCubeObjectX[i, 2, 1]);
                        FrontCubeObjectX[i, 2, 0] = new
CubeObject(FrontTempCubeObjectX[i, 2, 2]);
                        FrontCubeObjectX[i, 0, 1] = new
CubeObject(FrontTempCubeObjectX[i, 1, 0]);
                        FrontCubeObjectX[i, 1, 1] = (FrontTempCubeObjectX[i,
1, 1] == null) ? null : new CubeObject(FrontTempCubeObjectX[i, 1, 1]);
                        FrontCubeObjectX[i, 2, 1] = new
CubeObject(FrontTempCubeObjectX[i, 1, 2]);
                        FrontCubeObjectX[i, 0, 2] = new
CubeObject(FrontTempCubeObjectX[i, 0, 0]);
                        FrontCubeObjectX[i, 1, 2] = new
CubeObject(FrontTempCubeObjectX[i, 0, 1]);
                        FrontCubeObjectX[i, 2, 2] = new
CubeObject(FrontTempCubeObjectX[i, 0, 2]);

                        BackCubeObjectX[i, 0, 0] = new
CubeObject(BackTempCubeObjectX[i, 2, 0]);
                        BackCubeObjectX[i, 1, 0] = new
CubeObject(BackTempCubeObjectX[i, 2, 1]);
                        BackCubeObjectX[i, 2, 0] = new
CubeObject(BackTempCubeObjectX[i, 2, 2]);
```

```
                        BackCubeObjectX[i, 0, 1] = new
CubeObject(BackTempCubeObjectX[i, 1, 0]);
                        BackCubeObjectX[i, 1, 1] = (BackTempCubeObjectX[i, 1,
1] == null) ? null : new CubeObject(BackTempCubeObjectX[i, 1, 1]);
                        BackCubeObjectX[i, 2, 1] = new
CubeObject(BackTempCubeObjectX[i, 1, 2]);
                        BackCubeObjectX[i, 0, 2] = new
CubeObject(BackTempCubeObjectX[i, 0, 0]);
                        BackCubeObjectX[i, 1, 2] = new
CubeObject(BackTempCubeObjectX[i, 0, 1]);
                        BackCubeObjectX[i, 2, 2] = new
CubeObject(BackTempCubeObjectX[i, 0, 2]);
                    }

                    CopyXSlicesToYSlices();
                    CopyXSlicesToZSlices();
                }
                else
                {
                    string strLog = nGlobleIndex.ToString() + " CCW  X-axis
Slice = ";
                    if (slice == 0)
                        strLog += "First";
                    else if (slice == 0)
                        strLog += "Second";
                    else strLog += "Third";

                    if (bLogFile == true)
                        ErrorLog(strLog);

                    //da_CompletedCCW() // x,y,z
                    FrontCubeObjectX[slice, 0, 0] = new
CubeObject(FrontTempCubeObjectX[slice, 2, 0]);
                    FrontCubeObjectX[slice, 1, 0] = new
CubeObject(FrontTempCubeObjectX[slice, 2, 1]);
                    FrontCubeObjectX[slice, 2, 0] = new
CubeObject(FrontTempCubeObjectX[slice, 2, 2]);
                    FrontCubeObjectX[slice, 0, 1] = new
CubeObject(FrontTempCubeObjectX[slice, 1, 0]);
                    FrontCubeObjectX[slice, 1, 1] =
(FrontTempCubeObjectX[slice, 1, 1] == null) ? null : new
CubeObject(FrontTempCubeObjectX[slice, 1, 1]);
                    FrontCubeObjectX[slice, 2, 1] = new
CubeObject(FrontTempCubeObjectX[slice, 1, 2]);
                    FrontCubeObjectX[slice, 0, 2] = new
CubeObject(FrontTempCubeObjectX[slice, 0, 0]);
                    FrontCubeObjectX[slice, 1, 2] = new
CubeObject(FrontTempCubeObjectX[slice, 0, 1]);
                    FrontCubeObjectX[slice, 2, 2] = new
CubeObject(FrontTempCubeObjectX[slice, 0, 2]);


                    FrontCubeObjectY[0, slice, 0] = new
CubeObject(FrontTempCubeObjectY[2, slice, 0]);
                    FrontCubeObjectY[0, slice, 1] = new
CubeObject(FrontTempCubeObjectY[1, slice, 0]);
```

```
                    FrontCubeObjectY[0, slice, 2] = new
CubeObject(FrontTempCubeObjectY[0, slice, 0]);
                    FrontCubeObjectY[1, slice, 0] = new
CubeObject(FrontTempCubeObjectY[2, slice, 1]);
                    FrontCubeObjectY[1, slice, 1] = (FrontTempCubeObjectY[1,
slice, 1] == null) ? null : new CubeObject(FrontTempCubeObjectY[1, slice, 1]);
                    FrontCubeObjectY[1, slice, 2] = new
CubeObject(FrontTempCubeObjectY[0, slice, 1]);
                    FrontCubeObjectY[2, slice, 0] = new
CubeObject(FrontTempCubeObjectY[2, slice, 2]);
                    FrontCubeObjectY[2, slice, 1] = new
CubeObject(FrontTempCubeObjectY[1, slice, 2]);
                    FrontCubeObjectY[2, slice, 2] = new
CubeObject(FrontTempCubeObjectY[0, slice, 2]);


                    FrontCubeObjectZ[0, slice, 0] = new
CubeObject(FrontTempCubeObjectZ[0, slice, 2]);
                    FrontCubeObjectZ[0, slice, 1] = new
CubeObject(FrontTempCubeObjectZ[1, slice, 2]);
                    FrontCubeObjectZ[0, slice, 2] = new
CubeObject(FrontTempCubeObjectZ[2, slice, 2]);
                    FrontCubeObjectZ[1, slice, 0] = new
CubeObject(FrontTempCubeObjectZ[0, slice, 1]);
                    FrontCubeObjectZ[1, slice, 1] = (FrontTempCubeObjectZ[1,
slice, 1] == null) ? null : new CubeObject(FrontTempCubeObjectZ[1, slice, 1]);
                    FrontCubeObjectZ[1, slice, 2] = new
CubeObject(FrontTempCubeObjectZ[2, slice, 1]);
                    FrontCubeObjectZ[2, slice, 0] = new
CubeObject(FrontTempCubeObjectZ[0, slice, 0]);
                    FrontCubeObjectZ[2, slice, 1] = new
CubeObject(FrontTempCubeObjectZ[1, slice, 0]);
                    FrontCubeObjectZ[2, slice, 2] = new
CubeObject(FrontTempCubeObjectZ[2, slice, 0]);

                    BackCubeObjectX[slice, 0, 0] = new
CubeObject(BackTempCubeObjectX[slice, 2, 0]);
                    BackCubeObjectX[slice, 1, 0] = new
CubeObject(BackTempCubeObjectX[slice, 2, 1]);
                    BackCubeObjectX[slice, 2, 0] = new
CubeObject(BackTempCubeObjectX[slice, 2, 2]);
                    BackCubeObjectX[slice, 0, 1] = new
CubeObject(BackTempCubeObjectX[slice, 1, 0]);
                    BackCubeObjectX[slice, 1, 1] = (BackTempCubeObjectX[slice,
1, 1] == null) ? null : new CubeObject(BackTempCubeObjectX[slice, 1, 1]);
                    BackCubeObjectX[slice, 2, 1] = new
CubeObject(BackTempCubeObjectX[slice, 1, 2]);
                    BackCubeObjectX[slice, 0, 2] = new
CubeObject(BackTempCubeObjectX[slice, 0, 0]);
                    BackCubeObjectX[slice, 1, 2] = new
CubeObject(BackTempCubeObjectX[slice, 0, 1]);
                    BackCubeObjectX[slice, 2, 2] = new
CubeObject(BackTempCubeObjectX[slice, 0, 2]);


                    BackCubeObjectY[0, slice, 0] = new
CubeObject(BackTempCubeObjectY[2, slice, 0]);
```

```
                    BackCubeObjectY[0, slice, 1] = new
CubeObject(BackTempCubeObjectY[1, slice, 0]);
                    BackCubeObjectY[0, slice, 2] = new
CubeObject(BackTempCubeObjectY[0, slice, 0]);
                    BackCubeObjectY[1, slice, 0] = new
CubeObject(BackTempCubeObjectY[2, slice, 1]);
                    BackCubeObjectY[1, slice, 1] = (BackTempCubeObjectY[1,
slice, 1] == null) ? null : new CubeObject(BackTempCubeObjectY[1, slice, 1]);
                    BackCubeObjectY[1, slice, 2] = new
CubeObject(BackTempCubeObjectY[0, slice, 1]);
                    BackCubeObjectY[2, slice, 0] = new
CubeObject(BackTempCubeObjectY[2, slice, 2]);
                    BackCubeObjectY[2, slice, 1] = new
CubeObject(BackTempCubeObjectY[1, slice, 2]);
                    BackCubeObjectY[2, slice, 2] = new
CubeObject(BackTempCubeObjectY[0, slice, 2]);

                    BackCubeObjectZ[0, slice, 0] = new
CubeObject(BackTempCubeObjectZ[0, slice, 2]);
                    BackCubeObjectZ[0, slice, 1] = new
CubeObject(BackTempCubeObjectZ[1, slice, 2]);
                    BackCubeObjectZ[0, slice, 2] = new
CubeObject(BackTempCubeObjectZ[2, slice, 2]);
                    BackCubeObjectZ[1, slice, 0] = new
CubeObject(BackTempCubeObjectZ[0, slice, 1]);
                    BackCubeObjectZ[1, slice, 1] = (BackTempCubeObjectZ[1,
slice, 1] == null) ? null : new CubeObject(BackTempCubeObjectZ[1, slice, 1]);
                    BackCubeObjectZ[1, slice, 2] = new
CubeObject(BackTempCubeObjectZ[2, slice, 1]);
                    BackCubeObjectZ[2, slice, 0] = new
CubeObject(BackTempCubeObjectZ[0, slice, 0]);
                    BackCubeObjectZ[2, slice, 1] = new
CubeObject(BackTempCubeObjectZ[1, slice, 0]);
                    BackCubeObjectZ[2, slice, 2] = new
CubeObject(BackTempCubeObjectZ[2, slice, 0]);
                }

            }
            else if (AxisIndex == 1) // y-axis
            {
                if (slice == 3)
                {
                    string strLog = nGlobleIndex.ToString() + "  CW Y-axis
All";
                    if (bLogFile == true)
                        ErrorLog(strLog);

                    for (int i = 0; i < 3; i++)
                    {
                        //da_CompletedCW
                        FrontCubeObjectY[i, 0, 0] = new
CubeObject(FrontTempCubeObjectY[i, 2, 0]);
                        FrontCubeObjectY[i, 1, 0] = new
CubeObject(FrontTempCubeObjectY[i, 2, 1]);
                        FrontCubeObjectY[i, 2, 0] = new
CubeObject(FrontTempCubeObjectY[i, 2, 2]);
```

```
                        FrontCubeObjectY[i, 0, 1] = new
CubeObject(FrontTempCubeObjectY[i, 1, 0]);
                        FrontCubeObjectY[i, 1, 1] = (FrontTempCubeObjectY[i,
1, 1] == null) ? null : new CubeObject(FrontTempCubeObjectY[i, 1, 1]);
                        FrontCubeObjectY[i, 2, 1] = new
CubeObject(FrontTempCubeObjectY[i, 1, 2]);
                        FrontCubeObjectY[i, 0, 2] = new
CubeObject(FrontTempCubeObjectY[i, 0, 0]);
                        FrontCubeObjectY[i, 1, 2] = new
CubeObject(FrontTempCubeObjectY[i, 0, 1]);
                        FrontCubeObjectY[i, 2, 2] = new
CubeObject(FrontTempCubeObjectY[i, 0, 2]);

                        BackCubeObjectY[i, 0, 0] = new
CubeObject(BackTempCubeObjectY[i, 2, 0]);
                        BackCubeObjectY[i, 1, 0] = new
CubeObject(BackTempCubeObjectY[i, 2, 1]);
                        BackCubeObjectY[i, 2, 0] = new
CubeObject(BackTempCubeObjectY[i, 2, 2]);
                        BackCubeObjectY[i, 0, 1] = new
CubeObject(BackTempCubeObjectY[i, 1, 0]);
                        BackCubeObjectY[i, 1, 1] = (BackTempCubeObjectY[i, 1,
1] == null) ? null : new CubeObject(BackTempCubeObjectY[i, 1, 1]);
                        BackCubeObjectY[i, 2, 1] = new
CubeObject(BackTempCubeObjectY[i, 1, 2]);
                        BackCubeObjectY[i, 0, 2] = new
CubeObject(BackTempCubeObjectY[i, 0, 0]);
                        BackCubeObjectY[i, 1, 2] = new
CubeObject(BackTempCubeObjectY[i, 0, 1]);
                        BackCubeObjectY[i, 2, 2] = new
CubeObject(BackTempCubeObjectY[i, 0, 2]);

                    }
                    CopyYSlicesToXSlices();
                    CopyYSlicesToZSlices();

                }
                else
                {
                    string strLog = nGlobleIndex.ToString() + "  CW  Y-axis
Slice = ";
                    if (slice == 0)
                        strLog += "First";
                    else if (slice == 0)
                        strLog += "Second";
                    else strLog += "Third";

                    if (bLogFile == true)
                        ErrorLog(strLog);

                    //da_CompletedCW
                    FrontCubeObjectY[slice, 0, 0] = new
CubeObject(FrontTempCubeObjectY[slice, 2, 0]);
                    FrontCubeObjectY[slice, 1, 0] = new
CubeObject(FrontTempCubeObjectY[slice, 2, 1]);
                    FrontCubeObjectY[slice, 2, 0] = new
CubeObject(FrontTempCubeObjectY[slice, 2, 2]);
```

```
                    FrontCubeObjectY[slice, 0, 1] = new
CubeObject(FrontTempCubeObjectY[slice, 1, 0]);
                    FrontCubeObjectY[slice, 1, 1] =
(FrontTempCubeObjectY[slice, 1, 1] == null) ? null : new
CubeObject(FrontTempCubeObjectY[slice, 1, 1]);
                    FrontCubeObjectY[slice, 2, 1] = new
CubeObject(FrontTempCubeObjectY[slice, 1, 2]);
                    FrontCubeObjectY[slice, 0, 2] = new
CubeObject(FrontTempCubeObjectY[slice, 0, 0]);
                    FrontCubeObjectY[slice, 1, 2] = new
CubeObject(FrontTempCubeObjectY[slice, 0, 1]);
                    FrontCubeObjectY[slice, 2, 2] = new
CubeObject(FrontTempCubeObjectY[slice, 0, 2]);

                    FrontCubeObjectX[0, slice, 0] = new
CubeObject(FrontTempCubeObjectX[2, slice, 0]);
                    FrontCubeObjectX[0, slice, 1] = new
CubeObject(FrontTempCubeObjectX[1, slice, 0]);
                    FrontCubeObjectX[0, slice, 2] = new
CubeObject(FrontTempCubeObjectX[0, slice, 0]);
                    FrontCubeObjectX[1, slice, 0] = new
CubeObject(FrontTempCubeObjectX[2, slice, 1]);
                    FrontCubeObjectX[1, slice, 1] = (FrontTempCubeObjectX[1,
slice, 1] == null) ? null : new CubeObject(FrontTempCubeObjectX[1, slice, 1]);
                    FrontCubeObjectX[1, slice, 2] = new
CubeObject(FrontTempCubeObjectX[0, slice, 1]);
                    FrontCubeObjectX[2, slice, 0] = new
CubeObject(FrontTempCubeObjectX[2, slice, 2]);
                    FrontCubeObjectX[2, slice, 1] = new
CubeObject(FrontTempCubeObjectX[1, slice, 2]);
                    FrontCubeObjectX[2, slice, 2] = new
CubeObject(FrontTempCubeObjectX[0, slice, 2]);

                    FrontCubeObjectZ[0, 0, slice] = new
CubeObject(FrontTempCubeObjectZ[0, 2, slice]);
                    FrontCubeObjectZ[0, 1, slice] = new
CubeObject(FrontTempCubeObjectZ[1, 2, slice]);
                    FrontCubeObjectZ[0, 2, slice] = new
CubeObject(FrontTempCubeObjectZ[2, 2, slice]);
                    FrontCubeObjectZ[1, 0, slice] = new
CubeObject(FrontTempCubeObjectZ[0, 1, slice]);
                    FrontCubeObjectZ[1, 1, slice] = (FrontTempCubeObjectZ[1,
1, slice] == null) ? null : new CubeObject(FrontTempCubeObjectZ[1, 1, slice]);
                    FrontCubeObjectZ[1, 2, slice] = new
CubeObject(FrontTempCubeObjectZ[2, 1, slice]);
                    FrontCubeObjectZ[2, 0, slice] = new
CubeObject(FrontTempCubeObjectZ[0, 0, slice]);
                    FrontCubeObjectZ[2, 1, slice] = new
CubeObject(FrontTempCubeObjectZ[1, 0, slice]);
                    FrontCubeObjectZ[2, 2, slice] = new
CubeObject(FrontTempCubeObjectZ[2, 0, slice]);

                    BackCubeObjectY[slice, 0, 0] = new
CubeObject(BackTempCubeObjectY[slice, 2, 0]);
                    BackCubeObjectY[slice, 1, 0] = new
CubeObject(BackTempCubeObjectY[slice, 2, 1]);
```

```
                    BackCubeObjectY[slice, 2, 0] = new
CubeObject(BackTempCubeObjectY[slice, 2, 2]);
                    BackCubeObjectY[slice, 0, 1] = new
CubeObject(BackTempCubeObjectY[slice, 1, 0]);
                    BackCubeObjectY[slice, 1, 1] = (BackTempCubeObjectY[slice,
1, 1] == null) ? null : new CubeObject(BackTempCubeObjectY[slice, 1, 1]);
                    BackCubeObjectY[slice, 2, 1] = new
CubeObject(BackTempCubeObjectY[slice, 1, 2]);
                    BackCubeObjectY[slice, 0, 2] = new
CubeObject(BackTempCubeObjectY[slice, 0, 0]);
                    BackCubeObjectY[slice, 1, 2] = new
CubeObject(BackTempCubeObjectY[slice, 0, 1]);
                    BackCubeObjectY[slice, 2, 2] = new
CubeObject(BackTempCubeObjectY[slice, 0, 2]);

                    BackCubeObjectX[0, slice, 0] = new
CubeObject(BackTempCubeObjectX[2, slice, 0]);
                    BackCubeObjectX[0, slice, 1] = new
CubeObject(BackTempCubeObjectX[1, slice, 0]);
                    BackCubeObjectX[0, slice, 2] = new
CubeObject(BackTempCubeObjectX[0, slice, 0]);
                    BackCubeObjectX[1, slice, 0] = new
CubeObject(BackTempCubeObjectX[2, slice, 1]);
                    BackCubeObjectX[1, slice, 1] = (BackTempCubeObjectX[1,
slice, 1] == null) ? null : new CubeObject(BackTempCubeObjectX[1, slice, 1]);
                    BackCubeObjectX[1, slice, 2] = new
CubeObject(BackTempCubeObjectX[0, slice, 1]);
                    BackCubeObjectX[2, slice, 0] = new
CubeObject(BackTempCubeObjectX[2, slice, 2]);
                    BackCubeObjectX[2, slice, 1] = new
CubeObject(BackTempCubeObjectX[1, slice, 2]);
                    BackCubeObjectX[2, slice, 2] = new
CubeObject(BackTempCubeObjectX[0, slice, 2]);

                    BackCubeObjectZ[0, 0, slice] = new
CubeObject(BackTempCubeObjectZ[0, 2, slice]);
                    BackCubeObjectZ[0, 1, slice] = new
CubeObject(BackTempCubeObjectZ[1, 2, slice]);
                    BackCubeObjectZ[0, 2, slice] = new
CubeObject(BackTempCubeObjectZ[2, 2, slice]);
                    BackCubeObjectZ[1, 0, slice] = new
CubeObject(BackTempCubeObjectZ[0, 1, slice]);
                    BackCubeObjectZ[1, 1, slice] = (BackTempCubeObjectZ[1, 1,
slice] == null) ? null : new CubeObject(BackTempCubeObjectZ[1, 1, slice]);
                    BackCubeObjectZ[1, 2, slice] = new
CubeObject(BackTempCubeObjectZ[2, 1, slice]);
                    BackCubeObjectZ[2, 0, slice] = new
CubeObject(BackTempCubeObjectZ[0, 0, slice]);
                    BackCubeObjectZ[2, 1, slice] = new
CubeObject(BackTempCubeObjectZ[1, 0, slice]);
                    BackCubeObjectZ[2, 2, slice] = new
CubeObject(BackTempCubeObjectZ[2, 0, slice]);
                }
            }
            else // z-axis
            {
                if (slice == 3)
```

```
                {
                    string strLog = nGlobleIndex.ToString() + " CCW Z-axis
All";
                    if (bLogFile == true)
                        ErrorLog(strLog);

                    for (int i = 0; i < 3; i++)
                    {
                        FrontCubeObjectZ[i, 0, 0] = new
CubeObject(FrontTempCubeObjectZ[i, 2, 0]);
                        FrontCubeObjectZ[i, 1, 0] = new
CubeObject(FrontTempCubeObjectZ[i, 2, 1]);
                        FrontCubeObjectZ[i, 2, 0] = new
CubeObject(FrontTempCubeObjectZ[i, 2, 2]);
                        FrontCubeObjectZ[i, 0, 1] = new
CubeObject(FrontTempCubeObjectZ[i, 1, 0]);
                        FrontCubeObjectZ[i, 1, 1] = (FrontTempCubeObjectZ[i,
1, 1] == null) ? null : new CubeObject(FrontTempCubeObjectZ[i, 1, 1]);
                        FrontCubeObjectZ[i, 2, 1] = new
CubeObject(FrontTempCubeObjectZ[i, 1, 2]);
                        FrontCubeObjectZ[i, 0, 2] = new
CubeObject(FrontTempCubeObjectZ[i, 0, 0]);
                        FrontCubeObjectZ[i, 1, 2] = new
CubeObject(FrontTempCubeObjectZ[i, 0, 1]);
                        FrontCubeObjectZ[i, 2, 2] = new
CubeObject(FrontTempCubeObjectZ[i, 0, 2]);

                        BackCubeObjectZ[i, 0, 0] = new
CubeObject(BackTempCubeObjectZ[i, 2, 0]);
                        BackCubeObjectZ[i, 1, 0] = new
CubeObject(BackTempCubeObjectZ[i, 2, 1]);
                        BackCubeObjectZ[i, 2, 0] = new
CubeObject(BackTempCubeObjectZ[i, 2, 2]);
                        BackCubeObjectZ[i, 0, 1] = new
CubeObject(BackTempCubeObjectZ[i, 1, 0]);
                        BackCubeObjectZ[i, 1, 1] = (BackTempCubeObjectZ[i, 1,
1] == null) ? null : new CubeObject(BackTempCubeObjectZ[i, 1, 1]);
                        BackCubeObjectZ[i, 2, 1] = new
CubeObject(BackTempCubeObjectZ[i, 1, 2]);
                        BackCubeObjectZ[i, 0, 2] = new
CubeObject(BackTempCubeObjectZ[i, 0, 0]);
                        BackCubeObjectZ[i, 1, 2] = new
CubeObject(BackTempCubeObjectZ[i, 0, 1]);
                        BackCubeObjectZ[i, 2, 2] = new
CubeObject(BackTempCubeObjectZ[i, 0, 2]);
                    }

                    CopyZSlicesToXSlices();
                    CopyZSlicesToYSlices();
                }
                else
                {
                    string strLog = nGlobleIndex.ToString() + " CCW  Z-axis
Slice = ";
                    if (slice == 0)
                        strLog += "First";
                    else if (slice == 0)
```

```
                        strLog += "Second";
                    else strLog += "Third";
                    if (bLogFile == true)
                        ErrorLog(strLog);

                    //da_CompletedCCW()
                    FrontCubeObjectZ[slice, 0, 0] = new
CubeObject(FrontTempCubeObjectZ[slice, 2, 0]);
                    FrontCubeObjectZ[slice, 1, 0] = new
CubeObject(FrontTempCubeObjectZ[slice, 2, 1]);
                    FrontCubeObjectZ[slice, 2, 0] = new
CubeObject(FrontTempCubeObjectZ[slice, 2, 2]);
                    FrontCubeObjectZ[slice, 0, 1] = new
CubeObject(FrontTempCubeObjectZ[slice, 1, 0]);
                    FrontCubeObjectZ[slice, 1, 1] =
(FrontTempCubeObjectZ[slice, 1, 1] == null) ? null : new
CubeObject(FrontTempCubeObjectZ[slice, 1, 1]);
                    FrontCubeObjectZ[slice, 2, 1] = new
CubeObject(FrontTempCubeObjectZ[slice, 1, 2]);
                    FrontCubeObjectZ[slice, 0, 2] = new
CubeObject(FrontTempCubeObjectZ[slice, 0, 0]);
                    FrontCubeObjectZ[slice, 1, 2] = new
CubeObject(FrontTempCubeObjectZ[slice, 0, 1]);
                    FrontCubeObjectZ[slice, 2, 2] = new
CubeObject(FrontTempCubeObjectZ[slice, 0, 2]);

                    FrontCubeObjectX[0, 0, slice] = new
CubeObject(FrontTempCubeObjectX[2, 0, slice]);
                    FrontCubeObjectX[0, 1, slice] = new
CubeObject(FrontTempCubeObjectX[1, 0, slice]);
                    FrontCubeObjectX[0, 2, slice] = new
CubeObject(FrontTempCubeObjectX[0, 0, slice]);
                    FrontCubeObjectX[1, 0, slice] = new
CubeObject(FrontTempCubeObjectX[2, 1, slice]);//
                    FrontCubeObjectX[1, 1, slice] = (FrontTempCubeObjectX[1,
1, slice] == null) ? null : new CubeObject(FrontTempCubeObjectX[1, 1, slice]);
                    FrontCubeObjectX[1, 2, slice] = new
CubeObject(FrontTempCubeObjectX[0, 1, slice]);
                    FrontCubeObjectX[2, 0, slice] = new
CubeObject(FrontTempCubeObjectX[2, 2, slice]);
                    FrontCubeObjectX[2, 1, slice] = new
CubeObject(FrontTempCubeObjectX[1, 2, slice]);
                    FrontCubeObjectX[2, 2, slice] = new
CubeObject(FrontTempCubeObjectX[0, 2, slice]);

                    FrontCubeObjectY[0, 0, slice] = new
CubeObject(FrontTempCubeObjectY[0, 2, slice]);
                    FrontCubeObjectY[0, 1, slice] = new
CubeObject(FrontTempCubeObjectY[1, 2, slice]);
                    FrontCubeObjectY[0, 2, slice] = new
CubeObject(FrontTempCubeObjectY[2, 2, slice]);
                    FrontCubeObjectY[1, 0, slice] = new
CubeObject(FrontTempCubeObjectY[0, 1, slice]);
                    FrontCubeObjectY[1, 1, slice] = (FrontTempCubeObjectY[1,
1, slice] == null) ? null : new CubeObject(FrontTempCubeObjectY[1, 1, slice]);
                    FrontCubeObjectY[1, 2, slice] = new
CubeObject(FrontTempCubeObjectY[2, 1, slice]);
```

```
                    FrontCubeObjectY[2, 0, slice] = new
CubeObject(FrontTempCubeObjectY[0, 0, slice]);
                    FrontCubeObjectY[2, 1, slice] = new
CubeObject(FrontTempCubeObjectY[1, 0, slice]);
                    FrontCubeObjectY[2, 2, slice] = new
CubeObject(FrontTempCubeObjectY[2, 0, slice]);

                    BackCubeObjectZ[slice, 0, 0] = new
CubeObject(BackTempCubeObjectZ[slice, 2, 0]);
                    BackCubeObjectZ[slice, 1, 0] = new
CubeObject(BackTempCubeObjectZ[slice, 2, 1]);
                    BackCubeObjectZ[slice, 2, 0] = new
CubeObject(BackTempCubeObjectZ[slice, 2, 2]);
                    BackCubeObjectZ[slice, 0, 1] = new
CubeObject(BackTempCubeObjectZ[slice, 1, 0]);
                    BackCubeObjectZ[slice, 1, 1] = (BackTempCubeObjectZ[slice,
1, 1] == null) ? null : new CubeObject(BackTempCubeObjectZ[slice, 1, 1]);
                    BackCubeObjectZ[slice, 2, 1] = new
CubeObject(BackTempCubeObjectZ[slice, 1, 2]);
                    BackCubeObjectZ[slice, 0, 2] = new
CubeObject(BackTempCubeObjectZ[slice, 0, 0]);
                    BackCubeObjectZ[slice, 1, 2] = new
CubeObject(BackTempCubeObjectZ[slice, 0, 1]);
                    BackCubeObjectZ[slice, 2, 2] = new
CubeObject(BackTempCubeObjectZ[slice, 0, 2]);

                    BackCubeObjectX[0, 0, slice] = new
CubeObject(BackTempCubeObjectX[2, 0, slice]);
                    BackCubeObjectX[0, 1, slice] = new
CubeObject(BackTempCubeObjectX[1, 0, slice]);
                    BackCubeObjectX[0, 2, slice] = new
CubeObject(BackTempCubeObjectX[0, 0, slice]);
                    BackCubeObjectX[1, 0, slice] = new
CubeObject(BackTempCubeObjectX[2, 1, slice]);//
                    BackCubeObjectX[1, 1, slice] = (BackTempCubeObjectX[1, 1,
slice] == null) ? null : new CubeObject(BackTempCubeObjectX[1, 1, slice]);
                    BackCubeObjectX[1, 2, slice] = new
CubeObject(BackTempCubeObjectX[0, 1, slice]);
                    BackCubeObjectX[2, 0, slice] = new
CubeObject(BackTempCubeObjectX[2, 2, slice]);
                    BackCubeObjectX[2, 1, slice] = new
CubeObject(BackTempCubeObjectX[1, 2, slice]);
                    BackCubeObjectX[2, 2, slice] = new
CubeObject(BackTempCubeObjectX[0, 2, slice]);

                    BackCubeObjectY[0, 0, slice] = new
CubeObject(BackTempCubeObjectY[0, 2, slice]);
                    BackCubeObjectY[0, 1, slice] = new
CubeObject(BackTempCubeObjectY[1, 2, slice]);
                    BackCubeObjectY[0, 2, slice] = new
CubeObject(BackTempCubeObjectY[2, 2, slice]);
                    BackCubeObjectY[1, 0, slice] = new
CubeObject(BackTempCubeObjectY[0, 1, slice]);
                    BackCubeObjectY[1, 1, slice] = (BackTempCubeObjectY[1, 1,
slice] == null) ? null : new CubeObject(BackTempCubeObjectY[1, 1, slice]);
                    BackCubeObjectY[1, 2, slice] = new
CubeObject(BackTempCubeObjectY[2, 1, slice]);
```

```
                    BackCubeObjectY[2, 0, slice] = new
CubeObject(BackTempCubeObjectY[0, 0, slice]);
                    BackCubeObjectY[2, 1, slice] = new
CubeObject(BackTempCubeObjectY[1, 0, slice]);
                    BackCubeObjectY[2, 2, slice] = new
CubeObject(BackTempCubeObjectY[2, 0, slice]);
                }
            }
            nGlobleIndex++;

        }

    }
    internal class RubikMove
    {
        internal RubikMove(int nAxis, int nSlice, bool bMoveCW)
        {
            _nAxis = nAxis;
            _nSlice = nSlice;
            _bMoveCW = bMoveCW;
        }

        internal RubikMove(RubikMove rm)
        {
            _nAxis = rm._nAxis;
            _nSlice = rm._nSlice;
            _bMoveCW = rm._bMoveCW;
        }

        internal int _nAxis;
        internal int _nSlice;
        internal bool _bMoveCW;

    };

    internal class CubeObject
    {
        internal CubeObject(Transform3DGroup transGroup, ModelVisual3D
modelVisual3D, string Name)
        {
            _transGroup = transGroup;
            _modelVisual3D = modelVisual3D;
            _Name = Name;
        }

        internal CubeObject(CubeObject co)
        {
            if (co != null)
            {
                _transGroup = co._transGroup;
                _modelVisual3D = co._modelVisual3D;
                _Name = co._Name;
            }
        }
        internal void SetTransform3D(Transform3D transform)
        {
            if (_transGroup != null)
```

```
            {
                _transGroup.Children.Add(transform);
                _modelVisual3D.Transform = _transGroup;
            }
        }
        internal System.Windows.Media.Media3D.ModelVisual3D _modelVisual3D;
        internal Transform3DGroup _transGroup;
        internal string _Name;
    };
}
```