

1 WPF – IV

1.1 Display Primitive Graphics (Shapes)

- graphic shapes can be added to components like *Canvas*
- ➔ define canvas in XAML and
- perhaps include a picture

a) XAML

```
<Grid Margin="0,0,-237,-179">
  <Canvas x:Name="imgMap" HorizontalAlignment="Left" Height="436" Margin="20,49,0,0" VerticalAlignment="Top" Width="994">
    <Image x:Name="imgMap" MouseLeftButtonDown="OnMouseClick" Source="C:\Users\gerald\source\repos\WpfMush04\images\stockenboi.png" HorizontalAlignment="Left" Height="436"
  </Canvas>
  <Label x:Name="lblMessage" Content="with LEFT MOUSE CLICK set new discovery of selected mushroom" HorizontalAlignment="Left" Height="32" Margin="20,558,0,0" VerticalAlign
  <ComboBox x:Name="cboxMushroom" ToolTip="select a mushroom" SelectionChanged="OnSelectMushroom" HorizontalAlignment="Left" Height="35" Margin="20,585,0,0" VerticalAlignm
  <DatePicker x:Name="dpDiscovery" Text="select date" ToolTip="select date of discovery" HorizontalAlignment="Left" Height="35" Margin="298,585,0,0" VerticalAlignment="Top"
  <Menu HorizontalAlignment="Left" Height="34" Margin="20,10,0,0" VerticalAlignment="Top" Width="228">
    <MenuItem Header="Simple Commands" Height="41" Width="121">
      <MenuItem x:Name="mitemShowDiscovery" Header="Show Discoveries" HorizontalAlignment="Left" Height="31" Width="187" Click="OnSelectMenu" Margin="0,0,-42,0"/>
    </MenuItem>
  </Menu>
</Grid>
```

important hint: “image” is a child of “canvas”;

b) Controller

important hint: all eg. ellipses are children of “canvas”

```
private void PaintDiscoveries(List<Discovery> collDiscovery)
{
  canvasImage.Children.RemoveRange(1, canvasImage.Children.Count - 1); //remove all children (= ellipses) except image
  foreach (Discovery d in collDiscovery)
  {
    Ellipse myEllipse = new Ellipse();
    myEllipse.Stroke = System.Windows.Media.Brushes.Red;
    myEllipse.Fill = new SolidColorBrush() { Color = Colors.Red, Opacity = 0.75f };
    myEllipse.Width = Discovery.SIZE_OF_POINT;
    myEllipse.Height = Discovery.SIZE_OF_POINT;
    myEllipse.Margin = new Thickness(d.PositionOfDiscovery.ImgPoint_X - Discovery.SIZE_OF_POINT / 2, //set position in relation to parent GUI
                                     d.PositionOfDiscovery.ImgPoint_Y - Discovery.SIZE_OF_POINT / 2, //which is canvas
                                     0, 0); //=> distance left, top, bottom, right to canvas
    myEllipse.ToolTip = d.MushroomFound.Name + " found on " + d.DateOfDiscovery;
    canvasImage.Children.Add(myEllipse);
  }
}
```

2 SQL-Express

2.1 Settings

2.2 Connection-String

2.3 Working with Spatial (MS SQL-Server)

2.3.1 Overview

SQL Server supports two spatial data types:

- The geometry type represents data in a Euclidean (flat) coordinate system.
- The geography type represents data in a round-earth coordinate system.

2.3.2 SQL – Commands

```
CREATE TABLE discoveries (
  id INT NOT NULL IDENTITY,
  idMushroom INT,
  dateDiscovery DATE,
  geoint POINT,
  x NUMERIC(4),
  y NUMERIC(4),
  CONSTRAINT pkDiscovery PRIMARY KEY(id),
  CONSTRAINT fkDiscovery FOREIGN KEY (idMushroom) REFERENCES mushrooms
);

INSERT INTO dbo.discoveries (idMushroom, dateDiscovery, geoint, x, y)
VALUES(1, CONVERT(DATE, '14.04.2018', 104), geography::STPointFromText('POINT(46.743941 13.579347)', 4326), 332, 122);
INSERT INTO dbo.discoveries (idMushroom, dateDiscovery, geoint, x, y)
VALUES(1, CONVERT(DATE, '14.04.2018', 104), geography::STPointFromText('POINT(46.726884 13.615478)', 4326), 640, 342);
INSERT INTO dbo.discoveries (idMushroom, dateDiscovery, geoint, x, y)
VALUES(1, CONVERT(DATE, '14.04.2018', 104), geography::STPointFromText('POINT(46.712541 13.675658)', 4326), 1153, 527);
```

Latitude Longitude SRID

SRID...The ***Spatial Reference Identifier*** corresponds to a spatial reference system based on the specific ellipsoid used for either flat-earth mapping or round-earth mapping. It is defined by the European Petroleum Survey Group.

4326 is code for:GEOGCS["WGS 84", DATUM["World Geodetic System 1984", ELLIPSOID["WGS 84", 6378137, 298.257223563]], PRIMEM["Greenwich", 0], UNIT["Degree", 0.0174532925199433]]

id	idMushroom	dateDiscovery	geopoint	x	y
1	1	14.04.2018	POINT (46.743941 13.579347)	332	122
2	1	14.04.2018	POINT (46.726884 13.615478)	640	342
3	1	14.04.2018	POINT (46.712541 13.675658)	1153	527

2.3.3 ADO.Net

reading

```
private static readonly string SELECT_ALL_DISCOVERY = "SELECT ... geopoint.Long, geopoint.Lat, x, y " +
    " FROM discoveries ... ";

public List<Discovery> GetAllDiscoveries()
{
    List<Discovery> retValue = new List<Discovery>();
    SqlCommand cmd = new SqlCommand(SELECT_ALL_DISCOVERY, conn, trx);
    SqlDataReader reader = cmd.ExecuteReader();
    while (reader.Read())
    {
        Mushroom mush = new Mushroom(reader.GetString(1));
        Position pos = new Position(reader.GetDouble(3), reader.GetDouble(4), reader.GetInt32(5), reader.GetInt32(6));
        Discovery d = new Discovery(reader.GetInt32(0), reader.GetDateTime(2), pos, mush);
        retValue.Add(d);
    }
    reader.Close();
    return retValue;
}
```

writing

Steps:

- define a Geography – object (bind DLL – Library to your references)
- with the static method "Point"
- define a "user-defined" SQL-parameter (Udt)
- which holds Geography – object and
- transfers it to the database

```
private static readonly string INSERT_DISCOVERY = "INSERT INTO discoveries (idMushroom, dateDiscovery, geopoint, x, y) " +
    " VALUES(@idMushroom, @dateDiscovery, @geopoint, @x, @y)";

public void AddDiscovery(Discovery d)
{
    SqlCommand cmd = new SqlCommand(INSERT_DISCOVERY, conn, trx);

    cmd.Parameters.Add("idMushroom", SqlDbType.Int).Value = d.MushroomFound.Id;
    cmd.Parameters.Add("dateDiscovery", SqlDbType.Date).Value = d.DateOfDiscovery;
    SqlParameter para = new System.Data.SqlClient.SqlParameter
    {
        ParameterName = "geopoint",
        SqlDbType = System.Data.SqlDbType.Udt,
        UdtTypeName = "geography",
        Value = SqlGeography.Point(d.PositionOfDiscovery.Latitude, d.PositionOfDiscovery.Longitude, 4326)
        //SRID 4326...Definition of Earth-Ellipsoid of 1984, Greenwich, unit = Meter
    };
    cmd.Parameters.Add(para);
    cmd.Parameters.Add("x", SqlDbType.Int).Value = d.PositionOfDiscovery.ImgPoint_X;
    cmd.Parameters.Add("y", SqlDbType.Int).Value = d.PositionOfDiscovery.ImgPoint_Y;

    cmd.ExecuteNonQuery();
}
```