

Hello all,

I propose to change a little zoom and scroll mechanism:

1. scroll horizontal: When mouse (wheel) and the buttons "alt" "shift" and "ctrl" is not pressed
2. scroll vertical: when mouse (wheel) and is pressed the "alt" button
3. zoom horizontal proportional by mouse position: when mouse (wheel) and is pressed the "shift" button
4. zoom vertical proportional by mouse position: when mouse (wheel) and is pressed the "ctrl" button

The future: in the same time can zoom by X and Y axes by pressing "ctrl" "shift" and mouse wheel. The zoom is proportional by mouse position like google picassa and other photo viewers.

The code is:

```
void Scatter::MouseWheel(Point p_point, int p_zdelta, dword p_other){
    double v_scale = p_zdelta/10000.;
    if(GetAlt()){
        SetXYMin(GetXMin(), GetYMin()+GetYRange()*v_scale, GetYMin2());
        SetMinUnits(GetXMinUnit(), GetYMinUnit()-GetYRange()*v_scale);
    }
    else{
        bool v_is_ctrl = GetCtrl();
        bool v_is_shift = GetShift();

        if(v_is_ctrl){ // zoom y
            double v_zoom_v = GetYRange()*v_scale;
            double v_mouse_pos_v = GetYByPoint(p_point) - GetYMin();
            if(v_mouse_pos_v>0){
                SetXYMin(GetXMin(), GetYMin()-v_zoom_v*v_mouse_pos_v/GetYRange(), GetYMin2());
            }
            SetRange(GetXRange(), GetYRange()+v_zoom_v, GetY2Range());
        }
        if(v_is_shift){ // zoom x
            double v_zoom_v = GetXRange()*v_scale;
            double v_mouse_pos_v = GetXByPoint(p_point) - GetXMin();
            if(v_mouse_pos_v>0){
                SetXYMin(GetXMin()-v_zoom_v*v_mouse_pos_v/GetXRange(), GetYMin(), GetYMin2());
            }
            SetRange(GetXRange()+v_zoom_v, GetYRange(), GetY2Range());
        }

        if(!((v_is_ctrl)||v_is_shift)){
            SetMinUnits(GetXMinUnit()-GetXRange()*v_scale, GetYMinUnit());
        }
    }
}
```

```
    SetXYMin(GetXMin()+GetXRange()*v_scale, GetYMin(), GetYMin2());  
}  
}  
Refresh();  
}
```

I'm waiting for your opinions.

If need, I can upload a compiled test-case
