

# **Modelovacie a renderovacie techniky**

**Zuzana Berger Haladová**

# Homework 2

## Affine transformation

$$(x', y', 1) = (x, y, 1) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{pmatrix} = (x, y, 1) \cdot T(t_x, t_y)$$

$$(x', y', 1) = (x, y, 1) \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} = (x, y, 1) \cdot S(s_x, s_y)$$

$$(x', y', 1) = (x, y, 1) \begin{pmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} = (x, y, 1) \cdot R(\varphi)$$

# Homework 2

- ▶ Application
  - ▶ Draw primitives
  - ▶ Apply affine transformations
  - ▶ Possible modifications
  
- ▶ Dynamic control
  - ▶ Ctrl, shift ...
  - ▶ Use double buffering

# Homework 2

- ▶ Regular polygon
  - ▶ n-vertices
  - ▶ Possible to form primitives as
    - ▶ Triangle
    - ▶ Square
    - ▶ Circle
- ▶ Algorithm for mouse hovering detection
  - ▶ Iterate all edges, substitute mouse coordinates into edge equation
- ▶ For each primitive store transformations

# Homework 2

- ▶ Rotation
  - ▶ angle, (center of mass of the primitive)
- ▶ Translation
  - ▶ x, y vectors
- ▶ Scaling
  - ▶ x, y factors
- ▶ Store matrices
  - ▶ Array of 9 floats
- ▶ Matrix multiplication

# Homework 2

- ▶ `pictureBox1_Paint(System::Object^ sender, System::Windows::Forms::PaintEventArgs^ e) {`  
  
    `Pen^ blackPen = gcnew Pen( Color::Black, 3.0f );`  
  
    `e->Graphics->DrawLine(blackPen, sx1, sy1, sx2, sy2);`  
  
}
- ▶ `pictureBox1->Refresh();`
- ▶ `(Control::ModifierKeys == Keys::Control)`
- ▶ `(Control::ModifierKeys == Keys::Shift)`

# Homework 2

- ▶ Deadline
  - ▶ 27.11.2014 (exactly two weeks)
- ▶ Executable version + source code
- ▶ zhaladova@gmail.com