

# Įvadas į žaidimų specialiųjų efektų programavimą

T120B167 Žaidimų grafinių specialiųjų efektų kūrimas ir  
programavimas

Rytis Maskeliūnas

Skype: rytmask

[Rytis.maskeliunas@ktu.lt](mailto:Rytis.maskeliunas@ktu.lt)

© R. Maskeliūnas >2013

© A. Noreika <2013

# Apie spec. efektus



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA



<https://www.youtube.com/watch?v=Tfh0ytz8S0k>

# Apie spec. efektus



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA



<https://www.youtube.com/watch?v=rCm3FYp4hdI>



- 1 sav. Įvadas
- 2 sav. Apšvietimas
- 3 sav. „Žemėlapiai“ / Jūsų kursinio planas
- 4 sav. Specialieji efektai 1
- 5 sav. Specialieji efektai 2
- 6 sav. 3D scenos (žemė, dangus, vanduo)
- 7 sav. Iškilumų žemėlapiai
- **8 sav. 1'as kursinio pristatymas / 1'ojo laboratorinio darbo ataskaita**
- 9 sav. Deformacijos
- 10 sav. Cel (Toon) shading
- 11 sav. Paveikslu paremtas renderiavimas
- 12 sav. Dalelių sistemos
- 13 sav. Šešėlių žemėlapiai
- **14 sav. Kursinio (pabaigto) pristatymas/ antrojo laboratorinio darbo ataskaita**
- 15 sav. rezervas
- **EGZ. [Egzaminas]**

# Atsiskaitymų planas



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA

Atsiskaitymo forma	Temos(ų) nr.	Val. sk.	Įtaka vertinimui	Užduoties pateikimo (*) ir atsiskaitymo savaitė (o)																
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17-20
<a href="#">Individualus darbas</a>	1-16	40	35	*							0						0			
<a href="#">Egzaminas žodžiu</a>	1-16	16	15	*															0	
<a href="#">Laboratorinio darbo gynimas</a>	1-16	40	50		*						0	*					0			
Iš viso	-	96	100																	

*\*Tikėtina egzamino užduotis – sukurti ir pademonstruoti dėstytojo nurodytą specialųjį efektą.*

- Turėti supratimą kas tie šeideriai / spec. Efektų skriptai ir kaip tai veikia, mokėti „gyvai“ padaryti kažką gražaus ir ne copy-paste‘inti.
- Sukurti ir implementuoti 10 unikalių (5 paremtų skriptais + 5 šeideriais) specialiųjų efektų KURSINIO DARBO PROJEKTE;
- 2 laboratoriniai darbai (skriptai / šeideriai) – atliekami KURSINIO DARBO PROJEKTE
- **Naudoti kursinio darbo projektą iš T120B169 arba T120B166 (kas neturėjo modulio – teks kurtis sceną);**
- Galima tęsti kaip bakalauro darbą!



- Kiekvienai lab. Užduočiai / kursinio darbo efektui:
- 33,3% - UŽ PADARYMĄ;
- 33,3% - UŽ MOKĖJIMĄ PAAIŠKINTI (kaip veikia) IR GYVAI PADARYTI DĖSTYTOJO PAPRAŠYTUS PAKEITIMUS;
- 33,3% - UŽ UNIKALUMĄ (dėstytojas tokio dar nematė, neatgooglino).

# .EGZAMINO METU REIKĖS PRISTATYTI LAIKMENĄ SU:



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA

- 1. Dokumentacija [DOC ar PDF], kurioje yra
  - *Turinys*
  - *Specialiųjų efektų aprašas su iliustracijomis;*
  - *Išeities tekstas su aprašytų funkcijų rinkiniu (komentaruose kode, išryškinta kita spalva, funkcijos indeksuojamos turinyje)*
  - *Viskas kas ne jūsų – tikslios nuorodos literatūros šaltinių skyriuje.*
- 2. Katalogas su pilnu projektu (*Išeities kodai su visomis bibliotekomis*)
- 3. Katalogas su sukompiliuotu projektu\* ir papildomomis bibliotekomis instaliacijai (*jei reikia*)
- *\*veikimas turės būti parodytas **GYVAI IR NE ANT SAVO KOMPIUTERIO***
- 4. LABORATORINIŲ DARBŲ ATASKAITOS IR PILNI PROJEKTAI (sukompiliuotų nereikia)

**LAB. / KURSINIO GYNIMO METU – prieš ginant BŪTINA PARODYTI „ATASKAITAS“ (1,2,3 punktai)!!!**





# Įrankiai šeideriams

```
1  #include <GPL.h>
2  #include <free_software.h>
3
4  void notepad4ever()
5  {
6      while (true)
7      {
8          Notepad++;
9      }
10 }
11
```

# NVIDIA Nsight Visual Studio Edition



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA

The screenshot displays the NVIDIA Nsight Visual Studio Edition interface. The main window shows the source code for a shader, `SledDarkMetal - Copy.psh`. The code is written in HLSL and includes various normalization and displacement calculations. The `Locals` window at the bottom left shows the current state of variables, including `worldNormal`, `worldCoord`, `worldBinormal`, `worldTangent`, `tex`, `projVelocity`, and `N`. The `Shaders` window at the bottom right lists the loaded shaders, including `main` and `main` (Pixel), with their respective types and symbolic status. The `Graphics Focus Picker` window on the right shows the current render target, `RT0 - 1280x720 - (R16G16B16A16_FLOAT)`, and the number of pixels, `576,302`. The `Graphics Inspector` window at the top right shows the current thread, `[0] Graphics Default Context`, and the stack frame, `Shader: SledDarkMetal - Copy.psh - [0] m...`.

Locals

Name	Value	Type
worldNormal	{x = -0.0099726664, y = -0.03755236, z = 0.99924493}	float3
worldCoord	{x = -0.51191813, y = 3.4421973, z = -0.18642822, w = 1}	float4
worldBinormal	{x = 0.13819657, y = 0.98963982, z = 0.038570616}	float3
worldTangent	{x = 0.99034107, y = -0.13847685, z = 0.0046797399}	float3
tex	{x = 0.076345407, y = 0.86925811}	float2
projVelocity	{x = 0.00010310233, y = 6.8565554e-05}	float2
N	{x = -0.0099726655, y = -0.037552357, z = 0.99924481}	float3
x	-0.0099726655	float
y	-0.037552357	float
z	0.99924481	float

Shaders

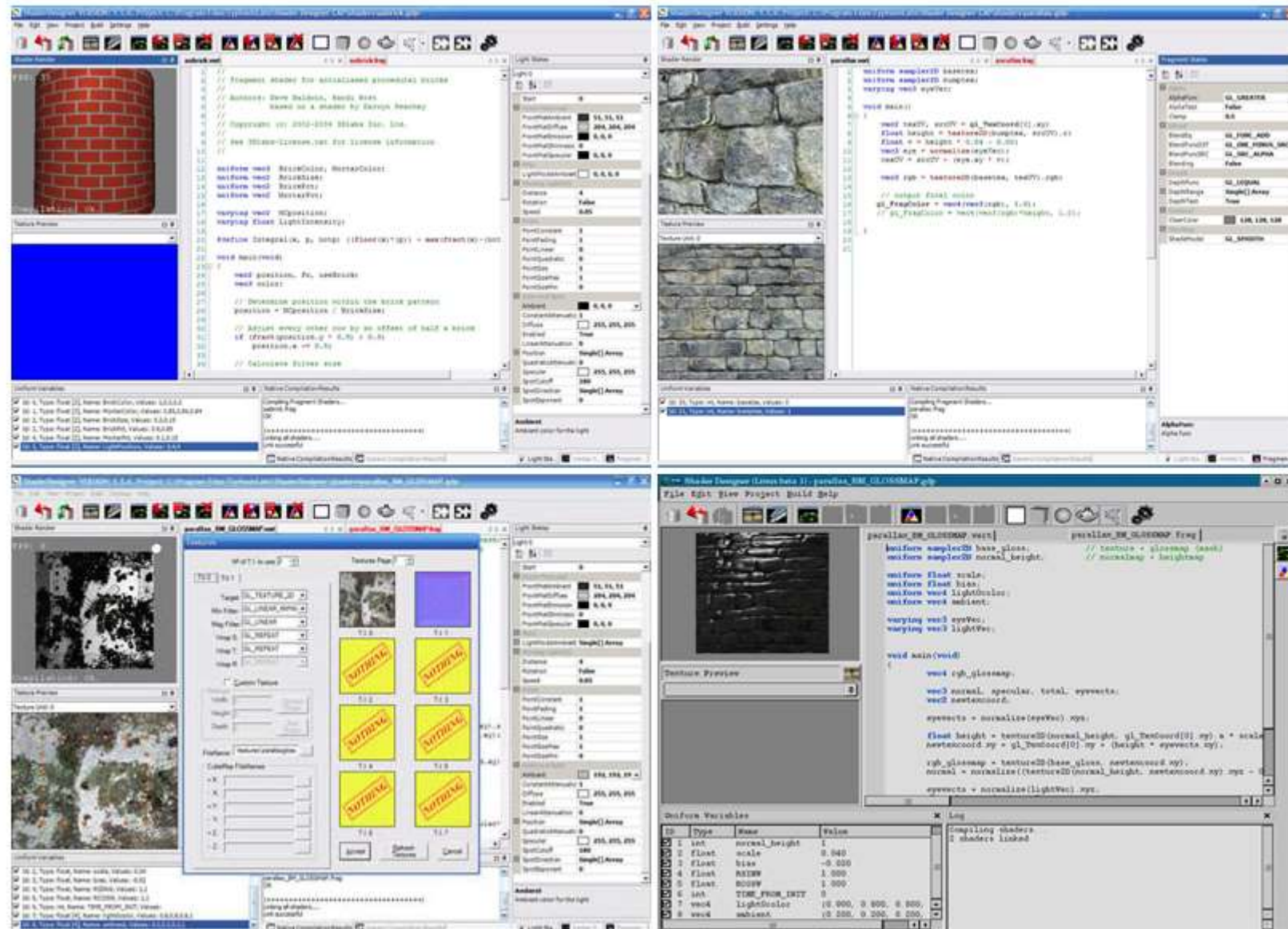
Name	Shader Type	Symbolics Status	Language	Tec
69 main	Vertex	Ready.	HLSL	No
70 main	Pixel	Ready.	HLSL	No
71 main	Vertex	Ready.	HLSL	No
72 main	Pixel	Ready.	HLSL	No
73 main	Vertex	Shader has not yet been used by the application.	HLSL	No
74 main	Pixel	Shader has not yet been used by the application.	HLSL	No
75 main	Vertex	Shader has not yet been used by the application.	HLSL	No
76 main	Pixel	Shader has not yet been used by the application.	HLSL	No
77 main	Vertex	Shader has not yet been used by the application.	HLSL	No

<https://developer.nvidia.com/nvidia-nsight-visual-studio-edition>

# OpenGL Shader Designer



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA



<https://www.opengl.org/sdk/tools/ShaderDesigner/>



# Online – Kick.js (WebGL)



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA

KickJS Documentation Examples Source Bugs Ask a question

New

Load

Save

Examples

Share

Full screen

Login

Kick.js | Shader editor | Default shader



Shader error console

Collapse

☐ Pause

Clear

Vertex shader

Fragment shader

Textures

Uniforms

Settings

About

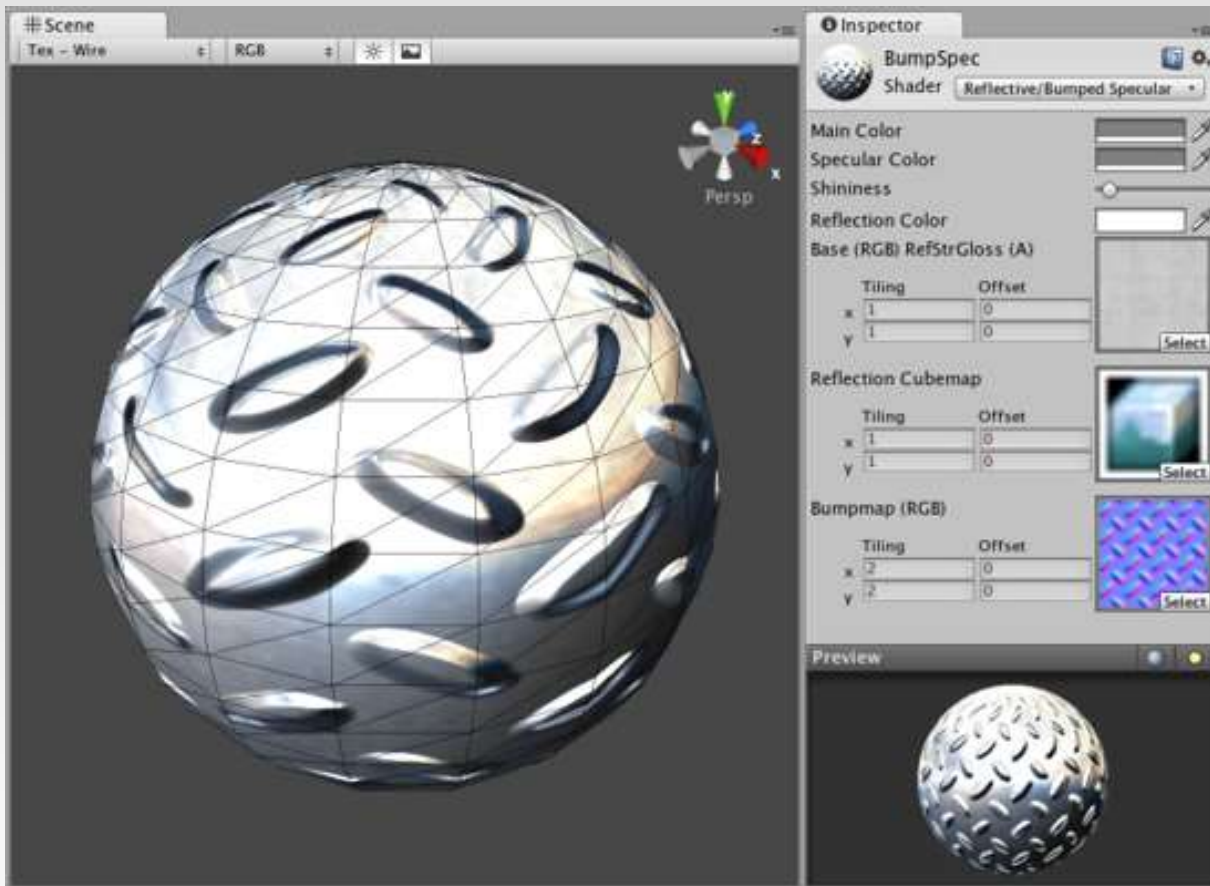
```
1 attribute vec3 vertex;  
2 attribute vec3 normal;  
3 attribute vec2 uv1;  
4  
5 uniform mat4 _mvProj;  
6 uniform mat3 _norm;  
7 uniform float _time;  
8  
9 varying vec2 uv;  
10 varying vec3 vColor;  
11  
12 #pragma include "light.glsl"  
13  
14 // constants  
15 vec3 materialColor = vec3(1.0,0.1,0.8);  
16 vec3 specularColor = vec3(1.0,0.5,1.0);  
17  
18 void main(void) {  
19     // compute position  
20     gl_Position = _mvProj * vec4(vertex, 1.0);  
21  
22     uv = uv1;  
23     // compute light info  
24     vec3 n = normalize(_norm * normal);  
25     vec3 diffuse;  
26     float specular;  
27     float glowingSpecular = sin(_time*0.003)*20.0+20.0;  
28     getDirectionalLight(n, _dlight, glowingSpecular, diffuse, specular);  
29     vColor = max(diffuse,_ambient.xyz)*materialColor+specular*specularColor;  
30 }
```

<http://www.kickjs.org/>

# Unity standartiniai šeideriai



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA



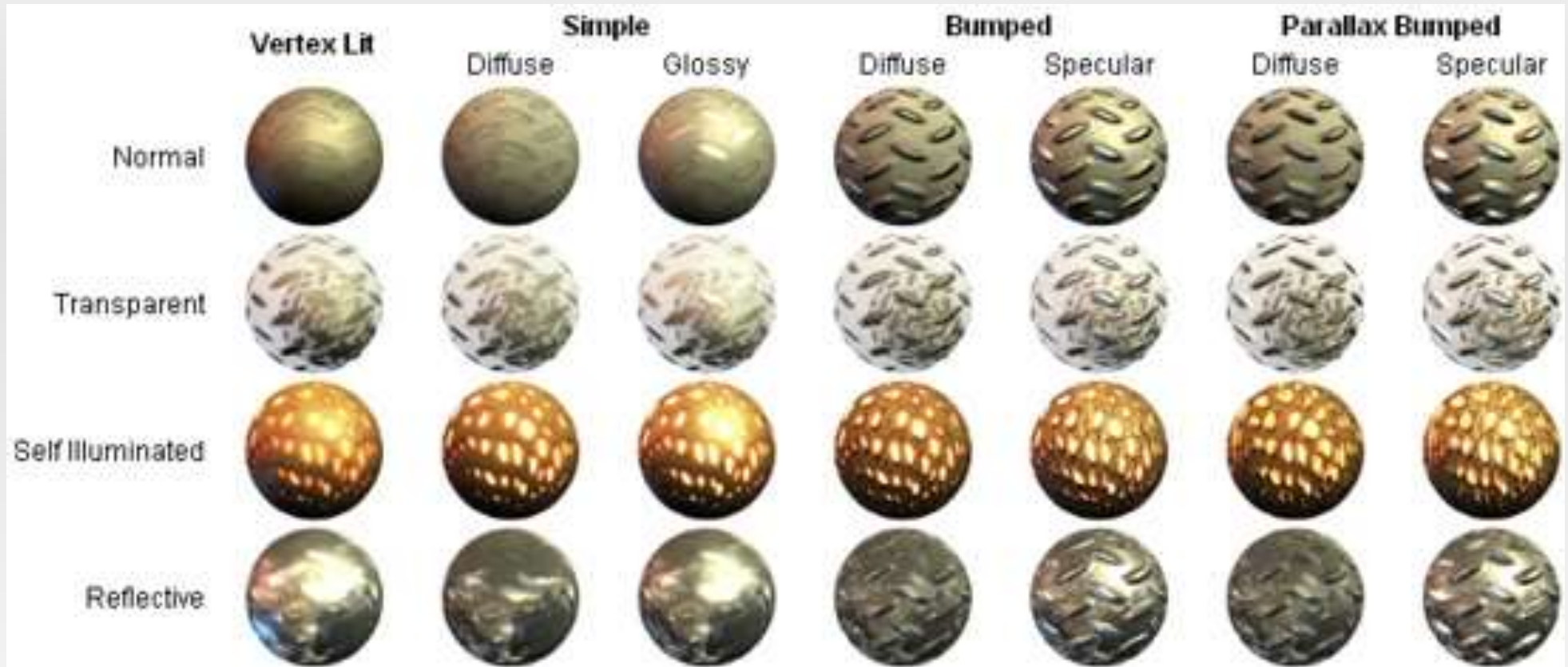
- Dauguma standartinių shader jau parašyti
- Jie pasirenkami pažymėjus „Materials“
- Galima shader parašyti:
  - Cg
  - ShaderLab
  - GLSL
  - HLSL
- Unity parašytus shader pati konvertuoja



# Unity standartiniai šeideriai



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA



Paviršių vaizdavimo tipai:

- Įprastinis (normal), naudojamas daugumai objektų
- Permatomas (transparent/cutout) – objektams naudojami tekstūros su alpha kanalu
- Self-illuminated – save apšviečiantys objektams, pvz. sienoms prie šviestuvų

# Unity3D strumpy



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA



- <http://u3d.as/content/strumpy-games/strumpy-shader-editor/1C4>



# IR ?

# Maya / Max / Blender



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA



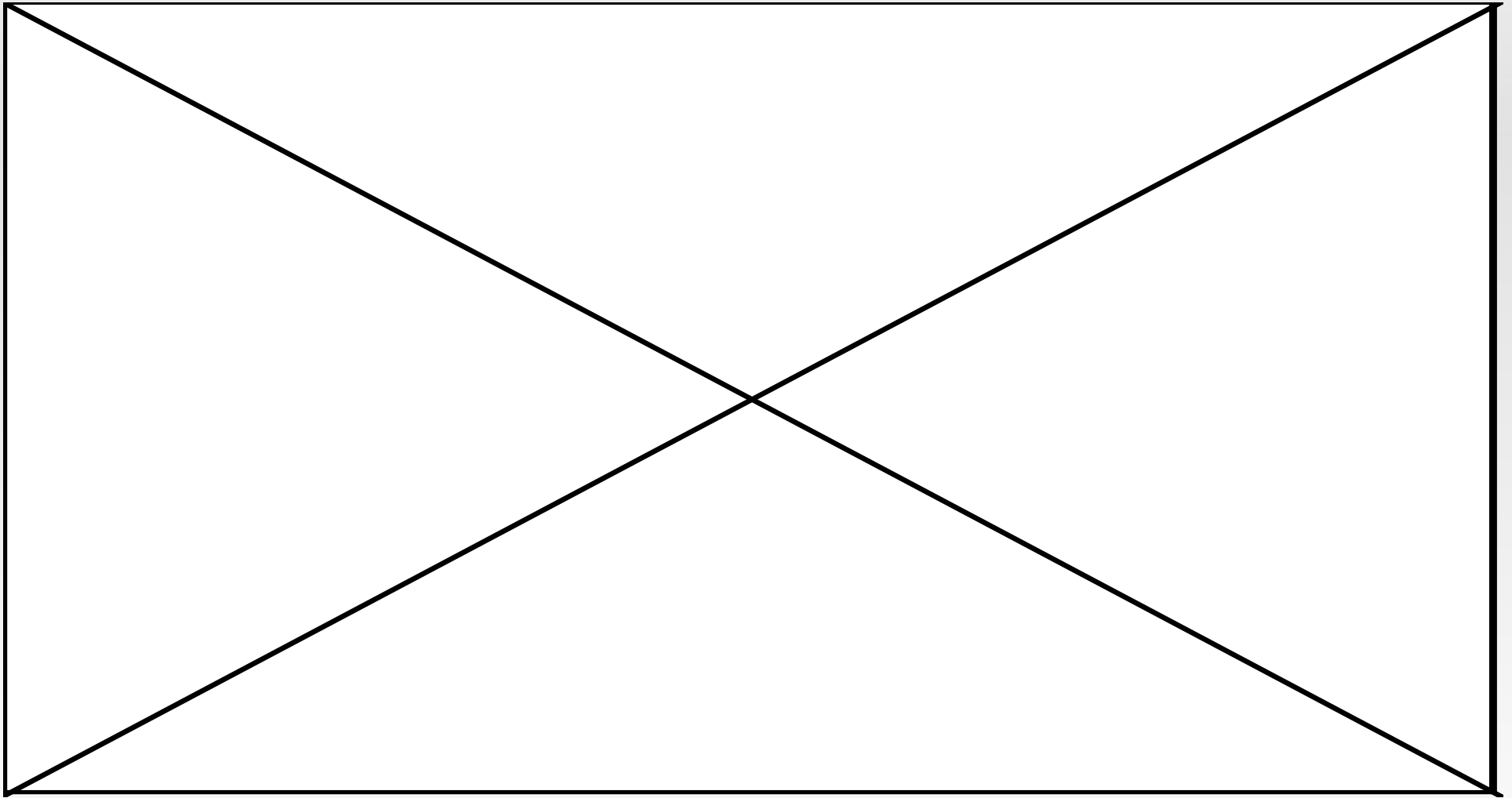


# **Vaizdo generavimo procesas KAIP TAI VEIKIA?**

# John Carmack paskaita (ND)

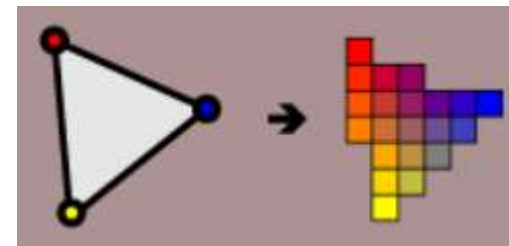
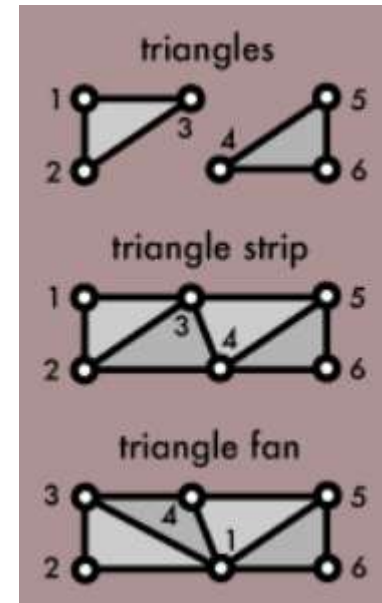
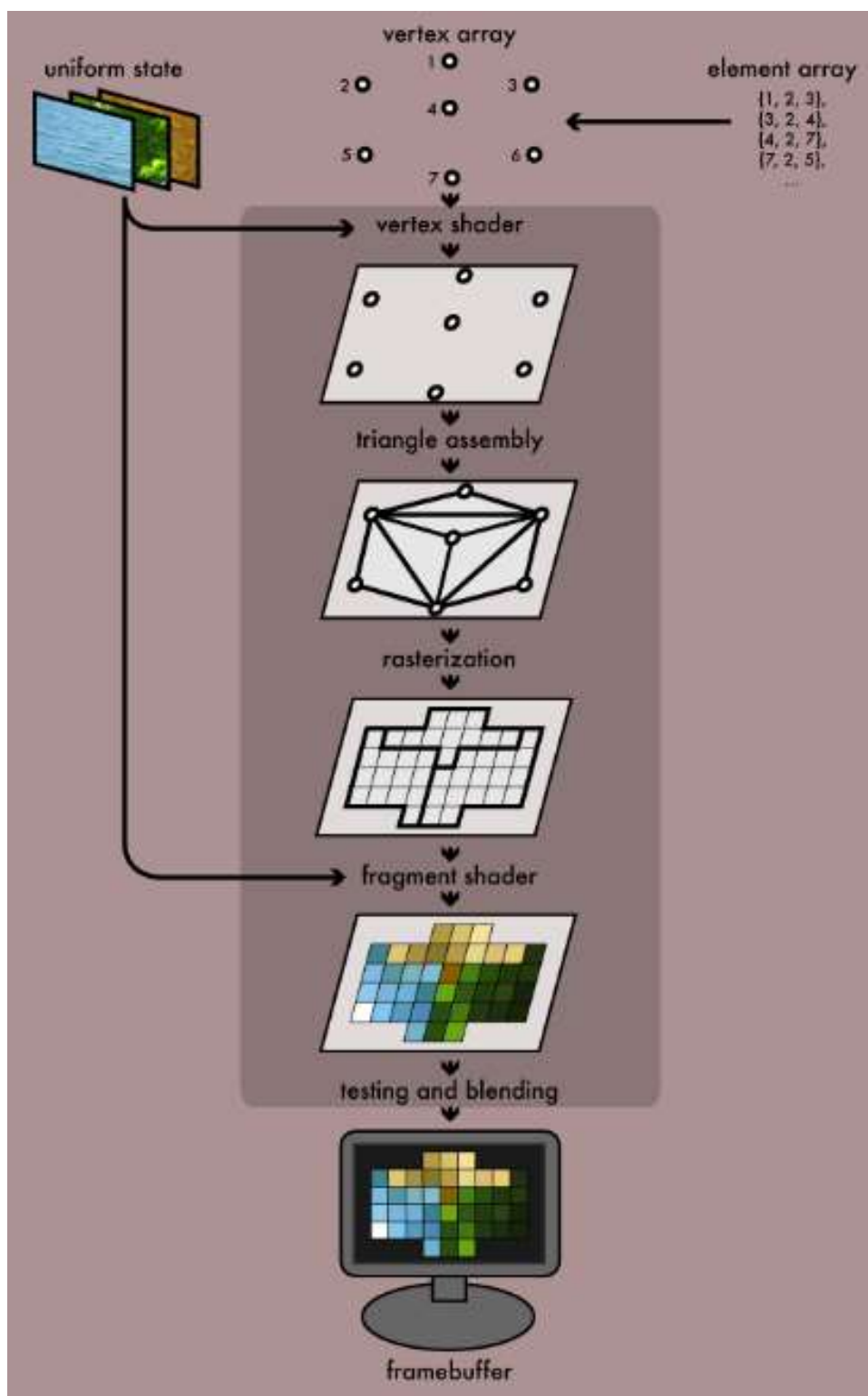


MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA



<https://www.youtube.com/watch?v=MG4QuTe8aUw>

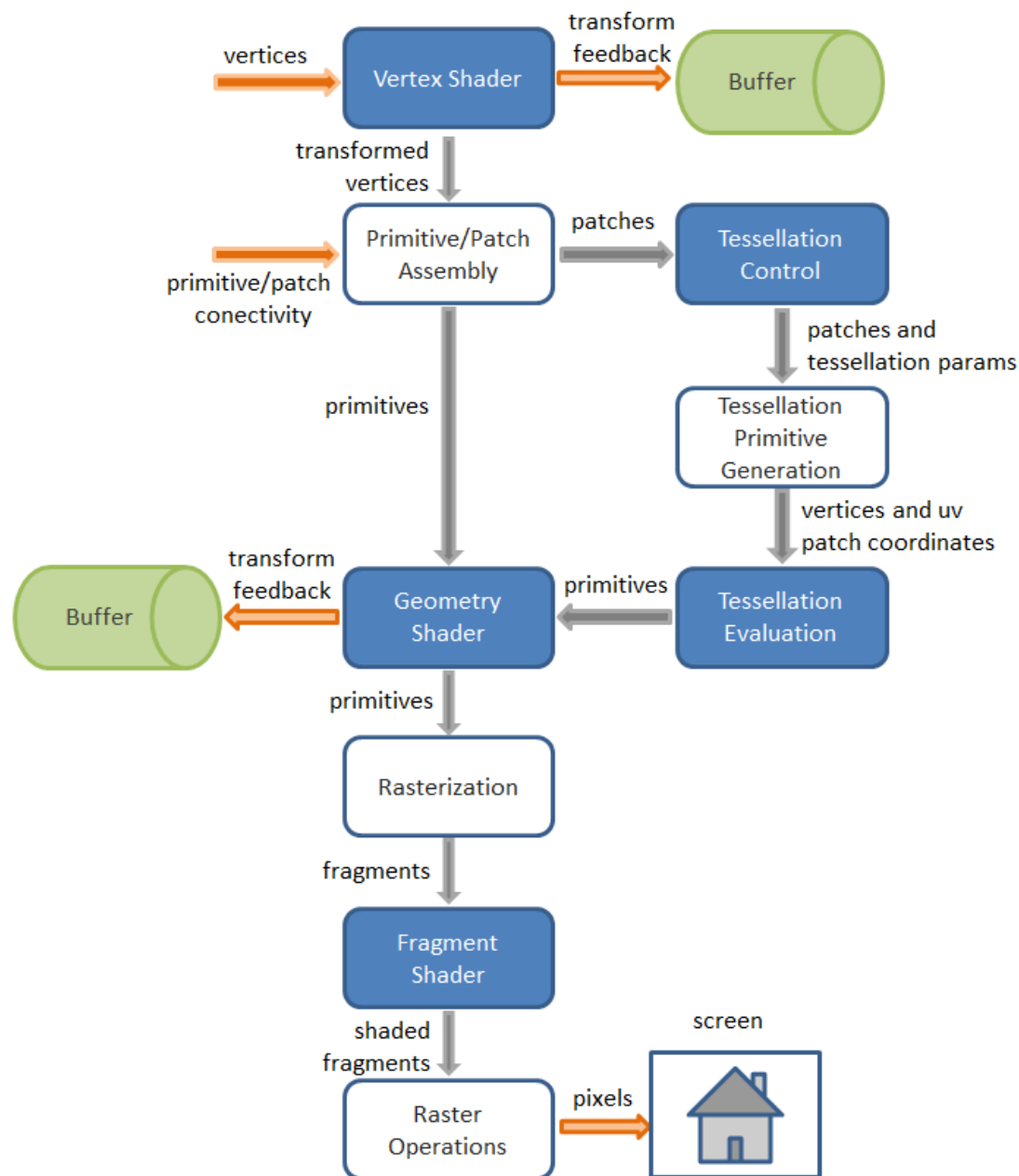




# Vaizdo generavimo procesas



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA



# Sheideriai...



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA



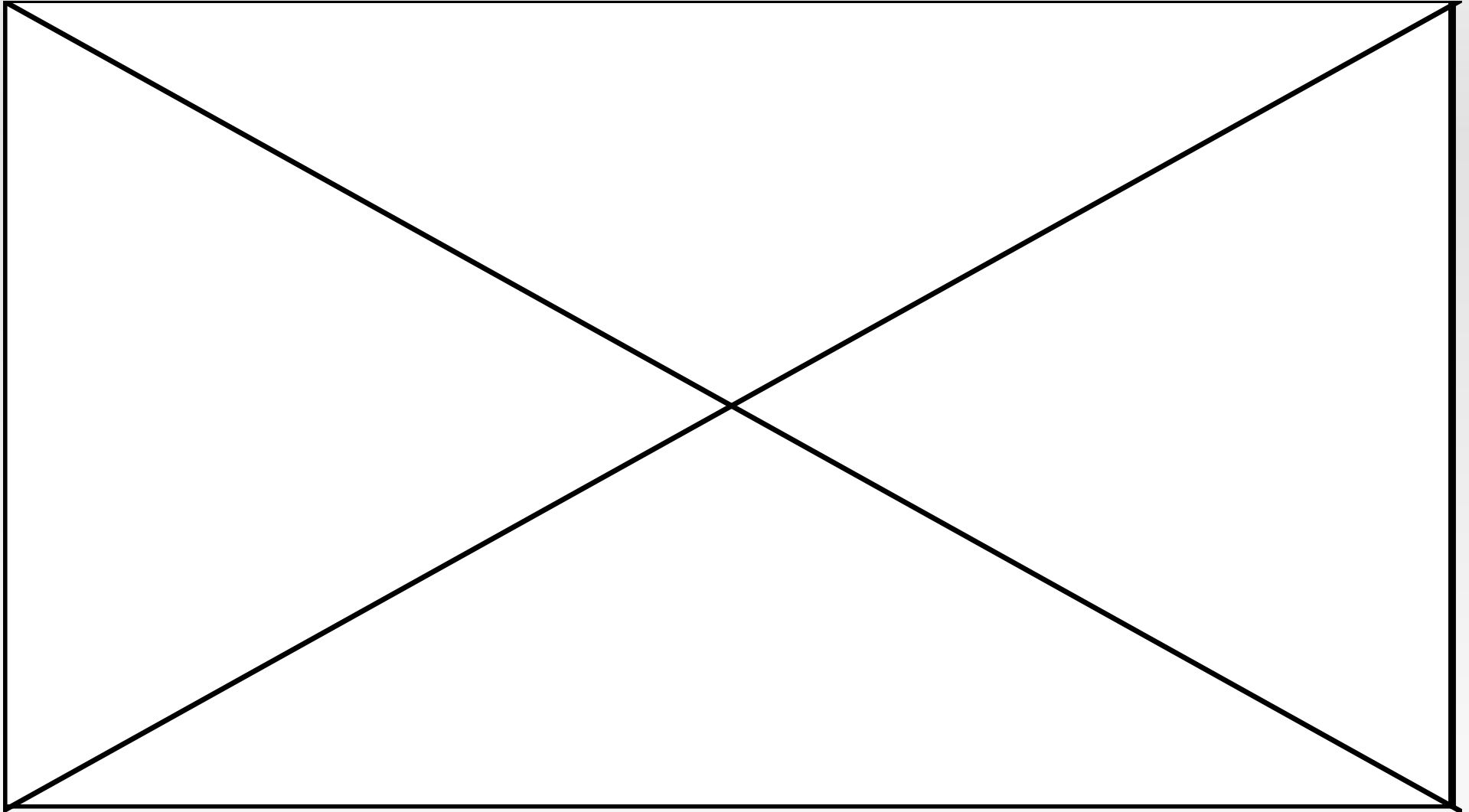
<https://www.youtube.com/watch?v=gC0PdspqNYk>

- Iki 2001 (iki DX8) – prasta grafika, fixed pipeline, nėra manipuliacijų bei transformacijų su pikseliais ir vertexais;
- Nuo 2011 (nuo DX8) – atsiranda vertex ir pixel šeideriai. Programuojama assembleriu... Fun...
- DX9 – atsiranda HLSL.
- DX10 – naujas modelis (4), naujas geometrijos šeideris ir Windows Višta ☺;
- DX11 – teseliavimas – ala realistiški paviršiai in hw...

# NEW vs OLD



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA

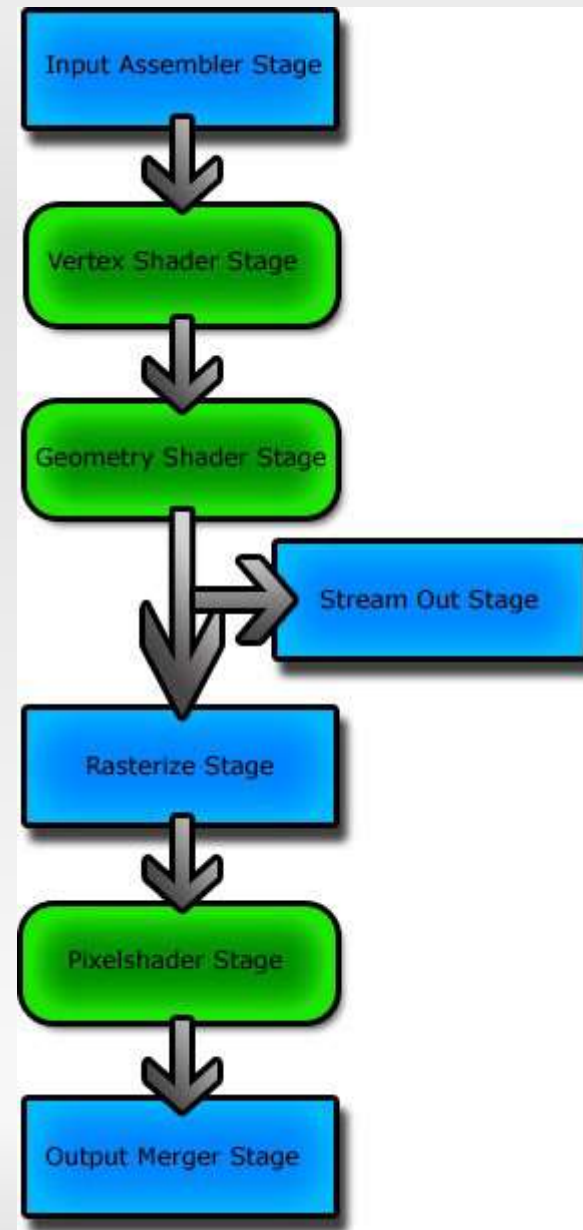


<https://www.youtube.com/watch?v=ISfN7OTUOTA>

# Kas tas šeideris?



- Viso labo instrukcija GPU ką daryti 😊
- Mažomis programytėmis galima valdyti tris grafikos kelio stadijas: vertex, geometrinių, pikselių šeiderių stadijos.





- Viršūnėlė (vertex) yra duomenų struktūra kuri nusako tašką 2D ar 3D erdvėje. Ekranu objektai sudaryti ir eilės plokščių figūrų (pvz., trikampių) masyvų, o tos viršūnėlės nurodo tų figūrų kampų vietą ir kažkokius požymius.
- Viršūnėlių šeideris turi būti įvykdomas kiekvienai iš viršūnėlių paduotų jūsų GPU. Esmė transformuoti kiekvienos viršūnėlės poziciją iš 3D virtualios erdvės į 2D ekrano koordinatę. Viršūnėlių šeiderioje negalima pridėti naujų viršūnėlių.

- Objekto į poziciją ekrane transformacija

```
struct VertexShaderInput
{
    float4 Position : POSITION0;
};

struct VertexShaderOutput
{
    float4 Position : POSITION0;
};

VertexShaderOutput VertexShaderFunction(VertexShaderInput input)
{
    VertexShaderOutput output;

    float4 worldPosition = mul(input.Position, World);
    float4 viewPosition = mul(worldPosition, View);
    output.Position = mul(viewPosition, Projection);

    return output;
}
```

- Pikselių (fragmentinis) šeideris iš esmės skaičiuoja su spalva susijusią informaciją kiekvienam iš pikselių. Šiame šeideryje pavyzdžiui, galima koreguoti pikselio „gylį“ (Z-buffering) ar apjungti keletą spalvų.
- Manipuliacija per raudonos spalvos kanalą (RED):

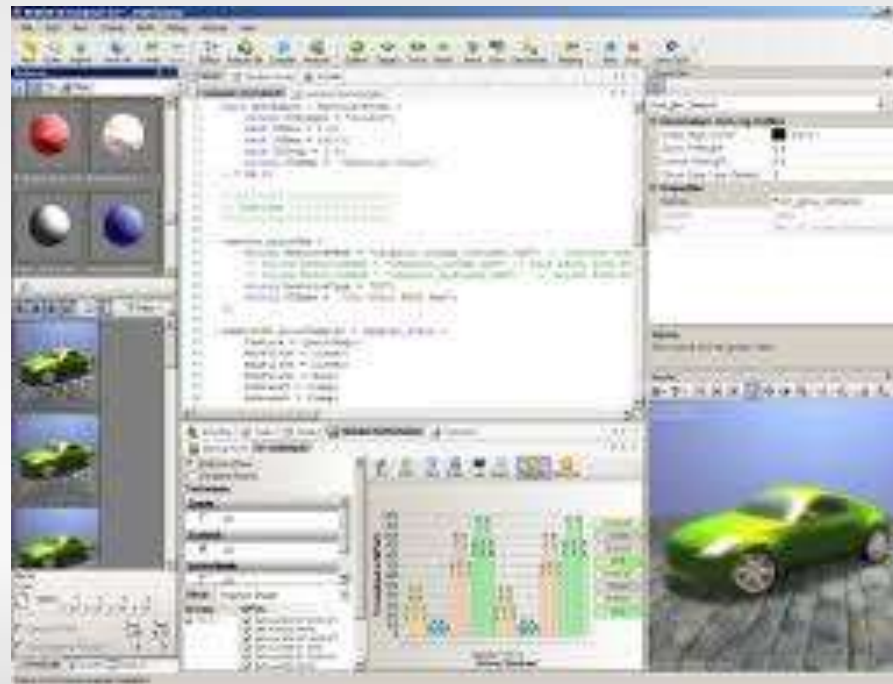
```
float4 PixelShaderFunction(VertexShaderOutput input) : COLOR0  
{  
    return float4(1,0,0,1);  
}
```

# Efektų failai (fx)



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA

- Juose saugoma šneiderio funkcionalumas (įskaitant vertex, geometrijos ir pixelių šneiderius).
- Šeideryje gali būti naudojamos efektų apdorojimo instrukcijos (angl. techniques). Kiekviena jų turi unikalų vardą, o renderinant vaizdą galima pasirinkti kurią iš šneiderio instrukcijų naudoti.



# Unity3D Spalvos pakeitimas



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA

```
Shader "Example/Diffuse Simple" {  
    SubShader {  
        Tags { "RenderType" = "Opaque" }  
        CGPROGRAM  
        #pragma surface surf Lambert  
        struct Input {  
            float4 color : COLOR;  
        };  
        void surf (Input IN, inout SurfaceOutput o) {  
            o.Albedo = 1;  
        }  
        ENDCG  
    }  
    Fallback "Diffuse"  
}
```

# Unity3D Spalvos pakeitimas



MULTIMEDIJOS  
INŽINERIJOS  
KATEDRA





- **KLAUSIMAI ? / PASIŪLYMAI ?**
- **KITA SAVAITĖ: KAIP VEIKIA APŠVIETIMAS?**