WebGL – Pojmy, kontext a farby

doc. Ing. Branislav Sobota, PhD.
Ing. Marián Hudák, Ing. Lenka Bubenková
Katedra počítačov a informatiky, FEI TU v Košiciach

C 02

Ciele cvičenia

1. Spustenie projektu v prostredí WebStorm
2. WebGL – obrazový bod - pixel.
3. WebGL - grafický kontext.
4. WebGl - grafické primitíva (entity)
5. WebGL - 3D štruktúra modelu
6. WebGL - farebný model

1. *Spustenie projektu v prostredí WebStorm*

Nainštalujte balíček Webgl_Start.zip a spustite ho vo vývojárskom prostredí.

1. Stiahnite z portálu Moodle KPI a z predmetu Počítačová grafika balíček **Webgl_getStart.zip**
2. Balíček rozbaľte a uložte na disku počítača.
3. Balíček otvorte vo vývojárskom prostredí.
4. Otvorenie balíčka v prostredí ***Webstorm*** je sprevádzané nasledujúcimi krokmi :
    1. Po extrahovaní balíčka na disk počítača je viditeľný koreňový adresár „**WebGL_getStart**“
    2. V prostredí ***Webstorm*** kliknite na „***Otvoriť projekt***“ a zvoľte cestu k adresáru „**WebGL_getStart**“
    3. Potvrďte otvorenie projektu

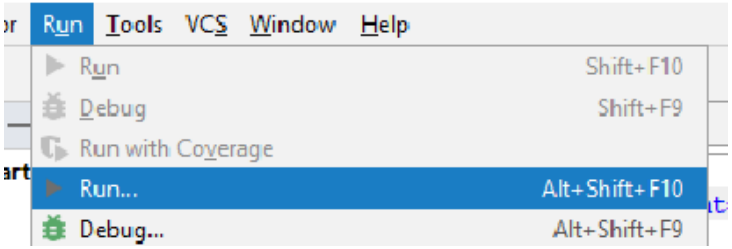
1. Štruktúra balíčka

- *Samotný balíček je pripravený rovnako pre tvorbu zadaní, ktorých štruktúra bude musieť byť rovnako dodržaná.*
- Balíček obsahuje hlavný koreňový adresár „**WebGL_getStart**“ a podadresáre napr. pre vkladanie textúr, skriptov (javascript) a 3D modelov, ktoré budú počas priebehu semestra použité.

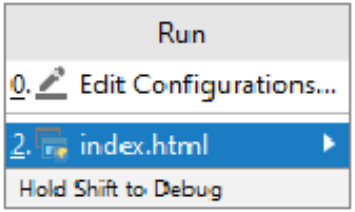
  - `WebGL_getStart`
    - `> js`
    - `  models`
    - `  texture`
    - `  Index.html`

1. SPUSTENIE PROJEKTU V PROSTREDÍ *WEBSTORM*

1. Po otvorení projektu prejdite kurzorom myši na ponuku Run a kliknite na možnosť *Run*... (spustenie projektu)

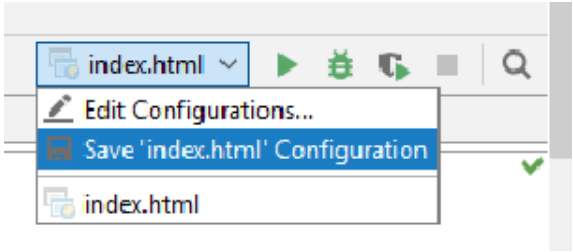


2. Po otvorení projektu prejdite kurzorom myši na ponuku Run a kliknite na možnosť *Run*... (spustenie projektu)



1. SPUSTENIE PROJEKTU V PROSTREDÍ *WEBSTORM*

3. Teraz je potrebné konfiguráciu uložiť. V pravom hornom rohu kliknite na výber :



   a zvoľte možnosť „***Save index.html Configuration***“

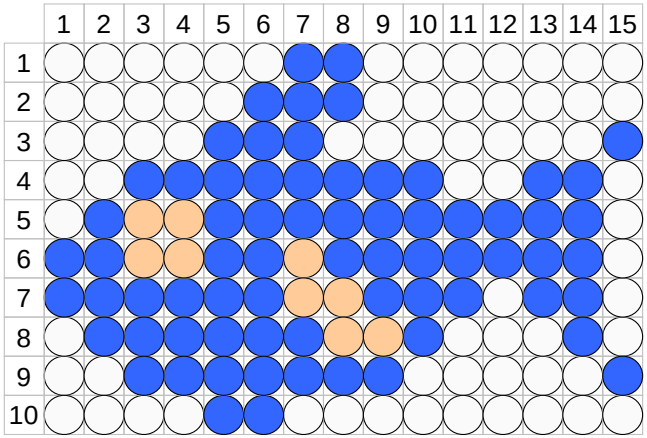
4. V tomto prípade ste uložili konfiguráciu a môžete spustiť projekt kliknutím na zelenú šípku „***Run***“. 

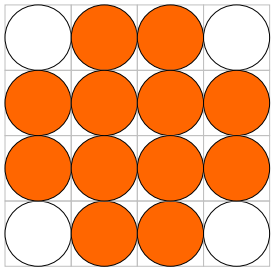
5. Automaticky sa otvorí súbor „**index.html**“ s nalinkovaným skriptom kontextu scény. (biela plocha)

2. WebGL – obrazový bod - pixel

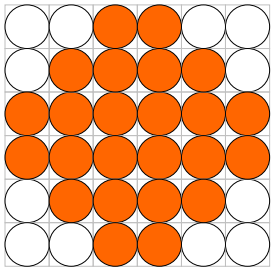
Základným používaným bodom je *pixel* [px]. Jedná sa o obrazový bod, ktorý je charakterizovaný dvomi súradnicami polohy a svojou farbou.

Z pohľadu WebGL je *pixel* chápaný ako najmenšia jednotka digitálnej rastrovej grafiky.

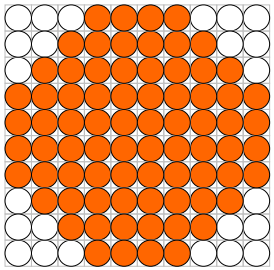




4PPI



6PPI



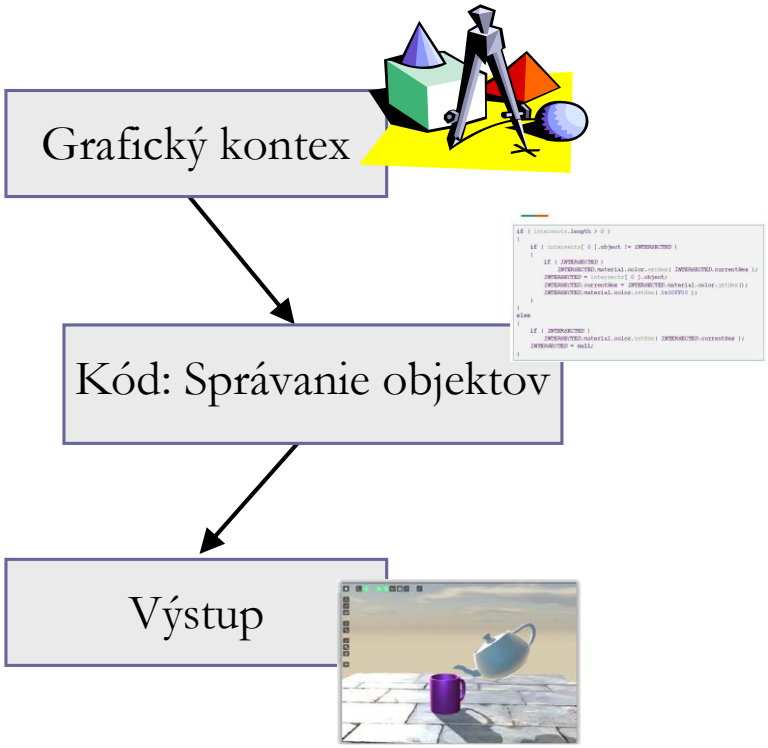
10PPI

3. WebGL – grafický kontext

ZÁKLADNÁ DEFINÍCIA

Z hľadiska WebGL grafický kontext obsahuje:

- *Plochu pre grafický výstup* (okno, plocha, celá obrazovka),
- *Množinu štandardných grafických entít* (obsiahnuté v každej grafickej knižnici, frameworku),
- *Množinu doplnkových grafických entít* (závislé od typu grafickej knižnice, frameworku),
- *Palety a nástroje pre prácu s kontextom* (typ kresliaceho štetca, farba, hrúbka, typ vyplnenia čiary).



3. WebGL – grafický kontext

Príprava Plátna

- Otvorte projekt ***Webgl_getStart*** vo vývojárskom prostredí a pripravte otvorený súbor „**index.html**“.
- Súbor „**index.html**“ obsahuje elementy tela, do ktorých je potrebné vložiť element plátna.
- Vložte nižšie uvedený element plátna do tela „index.html“ aby štruktúra kódu vyzerala nasledovne:

```
<body>
<canvas id="glCanvas" width="800" height="600">
</canvas>
</body>
```

3. WebGL – grafický kontext

Príprava kontextu JavaScript

- V rozbalenom balíčku na disku editujte súbor „**main.js**“ nachádzajúci sa v priečinku **/js**.
- V súbore „**main.js**“ je **prvom kroku je potrebné inicializovať kontext, ktorý je viazaný na element plátna.**
- Nasledujúce fragmenty vkladajte do funkcie **main()**

```
const canvas = document.querySelector("#glCanvas")
  const gl = canvas.getContext("webgl");
```

- V druhom kroku vložte kód pre kontrolu inštancie WebGL:

```
if (gl === null) {
    alert("Nemožno inicializovať WebGL.");
    return;
  }
```

3. WebGL – grafický kontext

Príprava kontextu JavaScript

- V treťom kroku doplňte implementáciu pre nastavenie farby pozadia kontextu:

```
    gl.clearColor(0.0, 0.0, 0.0, 1.0);
```

- V poslednom kroku do implementácie vložte fragment pre vyprázdnenie grafického buffera :

```
    gl.clear(gl.COLOR_BUFFER_BIT);
```

3. WEBGL – GRAFICKÝ KONTEXT

VLOŽENIE SKRIPTU DO SÚBORU INDEX.HTML

- Do súboru „**index.html**“ vložte medzi elementy plátna „**canvas**“ elementy skriptu :

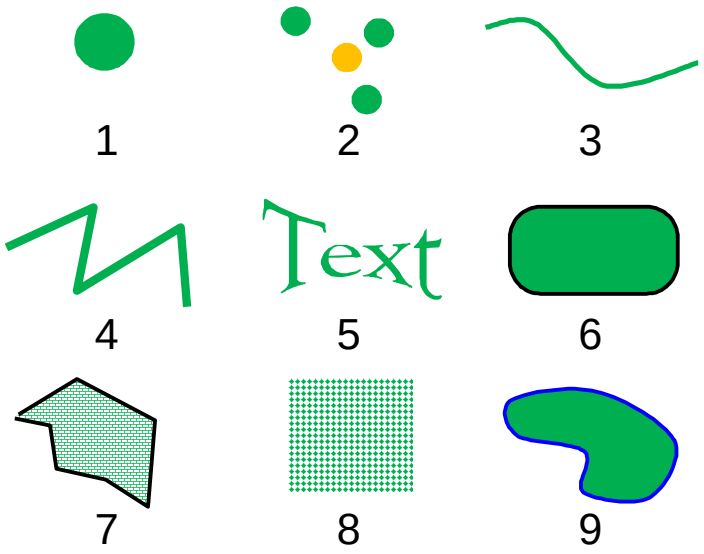
```
<script src="js/main.js"></script>
```

- **KONTROLA IMPLEMENTÁCIE :**
- Spustite vo webovom prehliadači súbor „index.html“. Pokiaľ ste postupovali správne, v prehliadači sa vykreslí čierne okno.

4. WebGL – grafické primitíva (entity)

Reprezentujú množinu geometrickch tvarov a nástrojov použiteľných pre tvorbu grafického prostredia :

- body, čiary, úsečky,
- kruhy, kružnice, polkruhy, polobúky,
- obdĺžniky, štvorce, trojuholníky,
- plochy, texty.



4. *WebGL* – MÓD KRESLENIA PRIMITÍV

- Pri vykresľovaní bodov, čiar, primitív sú vo WebGL používané metódy :

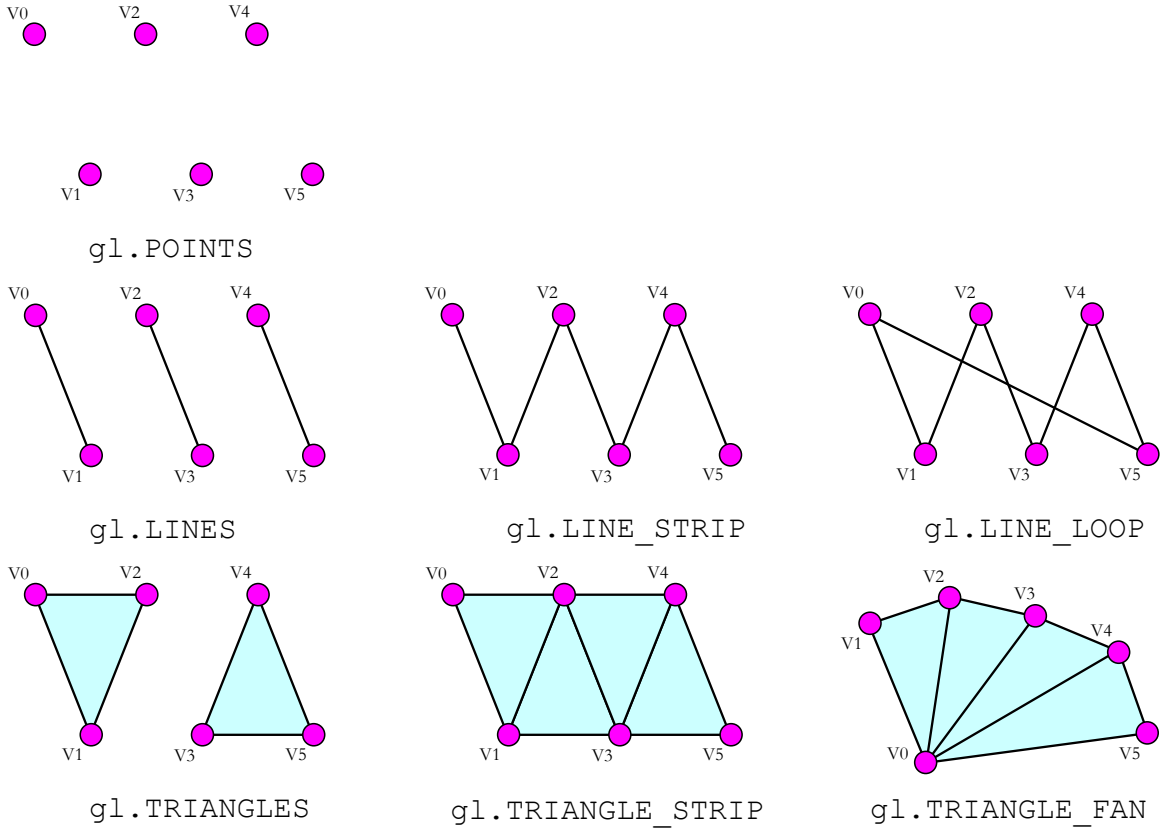
  - **Pre 2D :**
    - void **gl.drawArrays**(mode, first, count);
      - **mode** – enumeračný typ primitíva
      - **first** – počiatočný index poľa súradníc
      - **count** – počet renderovaných prvkov poľa súradníc

  - **Pre 3D :**
    - void **gl.drawElements**(mode, count, type, offset);
      - **mode** - enumeračný typ primitíva
      - **count** – počet renderovaných prvkov poľa súradníc
      - **type** – typ hodnôt renderovaných prvkov poľa
      - **offset** – počiatočný prvok renderovania

  Rozlišujú sa počtom vstupných parametrov.

4. WebGL – Enumeračné typy

- **Enumeračné typy primitív (mode) :**
  - gl.POINTS,
  - gl.LINE_STRIP,
  - gl.LINE_LOOP,
  - gl.LINES,
  - gl.TRIANGLE_STRIP,
  - gl.TRIANGLE_FAN,
  - gl.TRIANGLES.



4. WebGL – MÓD KRESLENIA PRIMITÍV

PRÁCA S PRIMITÍVOM BOD GL.POINTS

1. Prepíšte cestu skriptu v súbore „index.html“ na

```
<script src="js/points.js"></script>
```

2. Spustite projekt.
3. Prezrite si štruktúru skriptu.
4. Pozrite si rozdiely v implementácii jednotlivých bodov :
   Zápis súradníc bodov v premennej `var vertices`

4. WebGL – Mód kreslenia primitív

Práca s primitívom Čiara GL.LINES

1. Prepíšte cestu skriptu v súbore „index.html“ na

```
<script src="js/lines.js"></script>
```

2. Spustite projekt.
3. Prezrite si štruktúru skriptu.
4. Pozrite si rozdiely v implementácii počiatočných a koncových bodov každej z čiar:
   Zápis súradníc bodov sa nachádza v premennej `var vertices`

4. WebGL – MÓD KRESLENIA PRIMITÍV

PRÁCA S PRIMITÍVOM TROJUHOLNÍK GL.TRIANGLES

1. Prepíšte cestu skriptu v súbore „index.html“ na

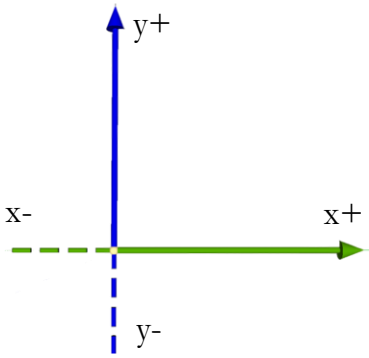
```
<script src="js/triangle.js"></script>
```

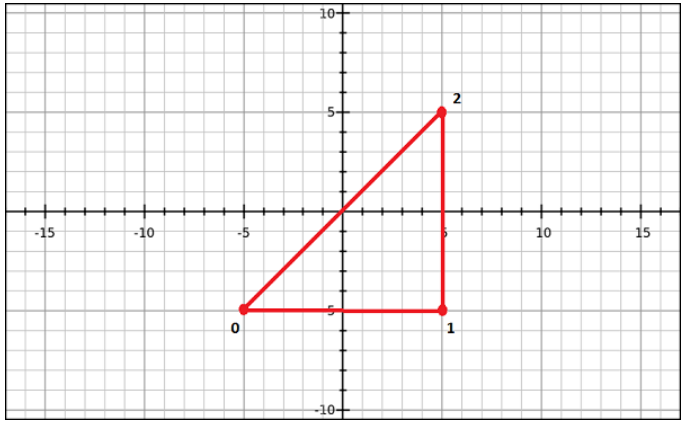
2. Spustite projekt.
3. Prezrite si štruktúru skriptu.
4. Pozrite si rozdiely v implementácii jednotlivých bodov poľa **var vertices** a **var indices**.

4. WebGL – MÓD KRESLENIA PRIMITÍV

POLE INDEXOV (IDENTIFIKÁTOROV) BODOV – VRCHOLOV PRIMITÍVA

- 2D súradnicový systém vo WebGL je definovaný nasledovne :





- **var indices** = [0,1,2];
  - je pole identifikátorov bodov podľa predchádzajúceho obrázku. Pole identifikátorov sa používa len pri vykresľovaní metódou drawElements(), pri vykresľovaní metódou drawArrays() nie je potrebné.
- **var vertices** = [-0.5, -0.5, //Vertex 1 = indice 0
  0.5, -0.5, //Vertex 2 = indice 1
  0.5, 0.5, //Vertex 3 = indice 2
  ];
  - je pole súradníc vrcholov primitíva.

4. WebGL – MÓD KRESLENIA PRIMITÍV

PRÁCA S PRIMITÍVOM TROJUHOLNÍK A ŠTVOREC GL.TRIANGLES

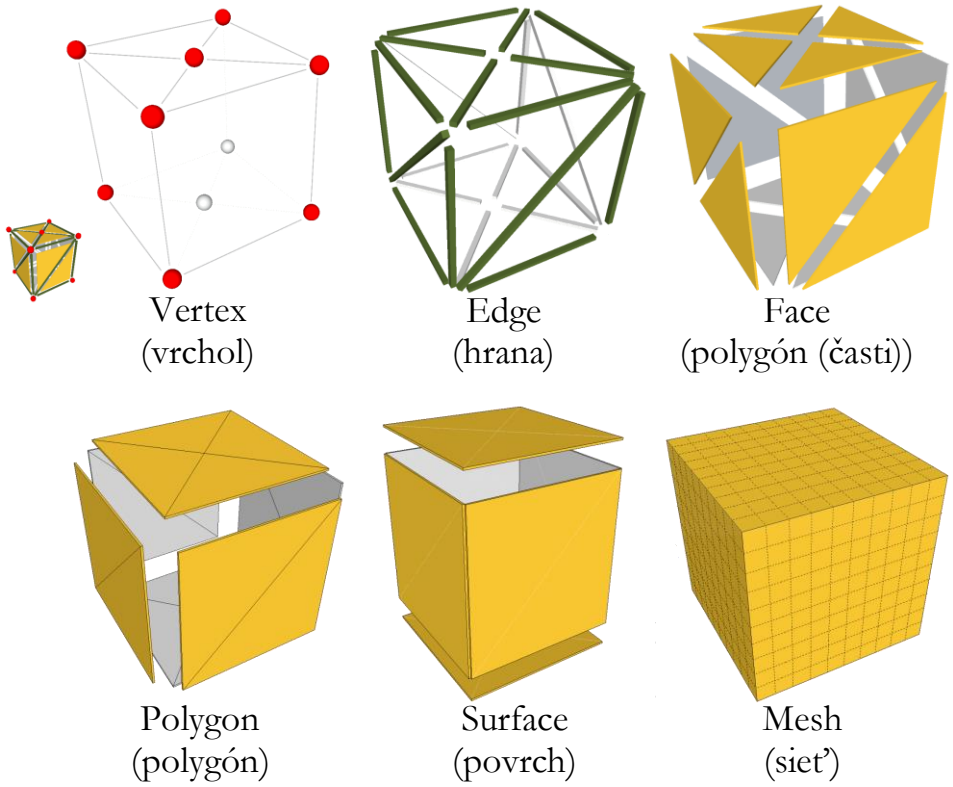
Úloha : ***Vykreslite primitívum ŠTVORCA skladaním trojuholníkov***

- Vytvorte skript **quad.js**, ktorý použije upravenú implementáciu skriptu **triangle.js**
- Nezabudnite pre úpravu metódy **gl.drawArrays**();
- **Aký najmenší počet trojuholníkov je potrebný k vykresleniu primitíva štvorec ?**

5. WebGL – Štruktúra 3D modelov

Kocka

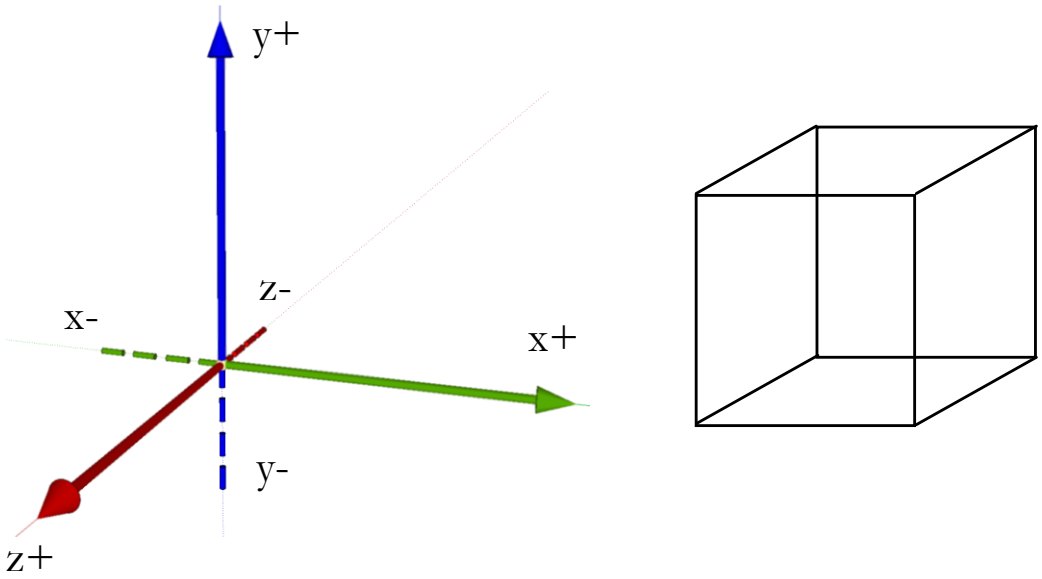
- 3D modely/objekty je možné definovať vo WebGL niektorým z nasledujúcich spôsobov:



5. WebGL – Štruktúra 3D modelov

Kocka

- 3D súradnicový systém vo WebGL je definovaný nasledovne :



5. WebGL – Štruktúra 3D modelov

Práca s primitívom KOCKA GL.TRIANGLES

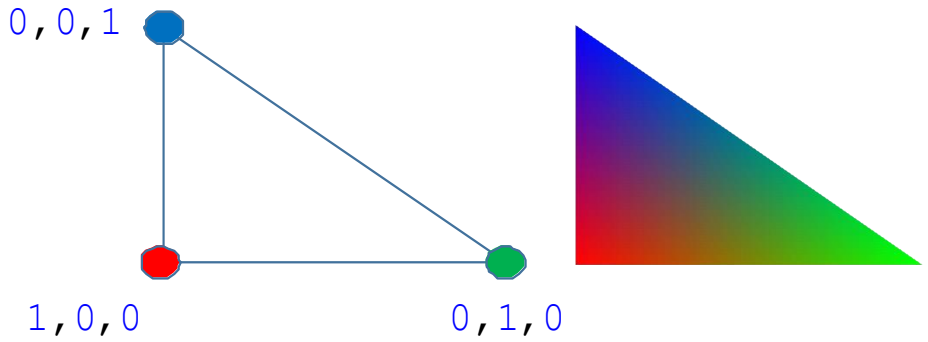
1. Prepíšte cestu skriptu v súbore „index.html“ na

```
<script src="js/cube.js"></script>
```

2. Spustite projekt.
3. Pozrite si spôsob vykreslenia kocky metódou **drawElements**().
   - void **gl.drawElements**(mode, count, type, offset);
   - **mode** - enumeračný typ primitíva
   - **count** – počet renderovaných prvkov poľa súradníc
   - **type** – typ hodnôt renderovaných prvkov poľa
   - **offset** – počiatočný prvok renderovania

6. WebGL – farebný model RGB

- **Farba pozadia**
  - gl.clearColor(0.5, 0.5, 0.5, 0.9);
  - R G B A (alfa, priesvitnosť)
- **Farba primitív**
  - *Pole súradníc vrcholov :*
    - **var vertices** = [-0.5,0.5,0.0,
      -0.5,-0.5,0.0,
      0.5,-0.5,0.0];
  - Pole farieb vrcholov – v hodnotách RGB :
    - **var colors** = [0,0,1, 1,0,0, 0,1,0];



5. WebGL – farebný model RGB

Práca s primitívom TROJUHILNÍKA GL.TRIANGLES s priradením farieb

1. Prepíšte cestu skriptu v súbore „index.html“ na

```
<script src="js/coloredTriangle.js"></script>
```

2. Spustite projekt.
3. Pozrite si súvislosti zápisu farieb a pridelenia farieb pre jednotlivé vrcholy primitíva.

DOPLŇUJÚCE ÚLOHY

- Diskutujte o spôsobe, ktorým je vytvorený kontext prostredníctvom vloženého skriptu „context.js“
- Vytvorte 2 trojuholníky vedľa seba, každému priraďte odlišnú farbu.

ÚLOHY NA SAMOSTATNÉ RIEŠENIE

- Použite zmenu farieb na primitívum trojuholníka tak aby :
- bol celý červenej farby.
- bol celý modrej farby.
- bol celý zelenej farby.
- bol celý žltej farby.
- Vytvorte štvorec a každému rohu priraďte inú farbu.
- Vytvorte štvorec a implementuje prechod medzi dvomi farbami:
- Vodorovne alebo zvisle zľava doprava (zhora nadol)
- Medzi vrcholmi na hlavnej diagonále (uhlopriečke) štvorca.

Q & A

branislav.sobota@tuke.sk
lenka.bubenkova@tuke.sk

Katedra počítačov a informatiky, FEI TU v Košiciach