

THREE.JS – SPRACOVÁVANIE VSTUPNÝCH PERIFÉRNÝCH ZARIADENÍ, OVLÁDANIE OBJEKTOV V ČASE

doc. Ing. Branislav Sobota, PhD.

Ing. Marián Hudák, Ing. Lenka Bubeňková

Katedra počítačov a informatiky, FEI TU v Košiciach

C 06

© 2024

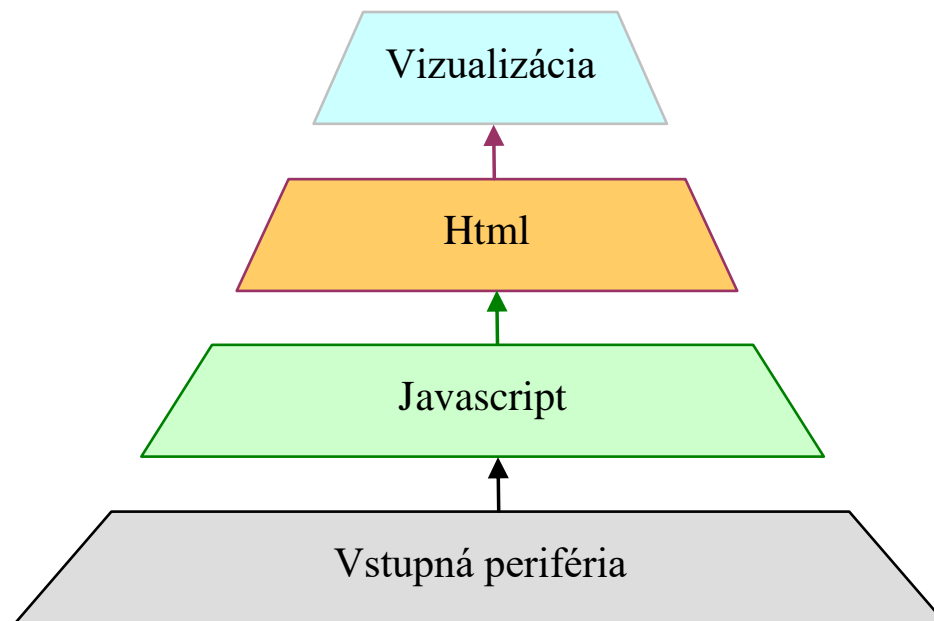
CIELE CVIČENIA

- Three.js - Nasadenie skriptu pre podporu vstupov z periférie klávesnice.
- Three.js - Deklarácia premenných a objektov pre ovládanie objektu v čase.
- Three.js - Implementácia metódy `update()` pre priradenie ovládania trojrozmerným objektom.
- Three.js - Používanie vektorov pri riadení pohybu 3D objektu.



1. *THREE.JS* –NASADENIE SKRIPTU PRE PODPORU VSTUPOV Z PERIFÉRIE KLÁVESNICE

- Vstupy z periférnych zariadení v *Three.js* vzhľadom na použité technológie sú spracovávané v hierarchickom usporiadaní tak, ako je zobrazené na nasledujúcom obrázku. Samotné spracovávanie vstupov je na úrovni JavaScript-u podľa kontextu aplikácie.



1. *THREE.JS* – PRÍPRAVA BALÍČKA

- **Úloha:** Stiahnite si balíček „**Threejs_Vstup_Ovladanie_objektov_v_case.zip**“ z portálu Moodle KPI a predmetu Počítačova grafika.
- Obsah balíčka skopírujte do vášho projektu aby štruktúra vyzerala nasledovne:
 - `WebGL_getStart`
 - `> css`
 - `> js`
 - `>> threejs`
 - `>>> threejs.KeyboardState.js`
 - `>> ThreeScene.js`
 - `models`
 - `> texture`
 - `index.html`

2. THREE.JS – DEKLARÁCIA PREMENNÝCH A OBJEKTOV PRE OVLÁDANIE OBJEKTU V ČASE

- **Úloha:** deklarujte potrebné premenné v skripte „ThreeScene.js“.
- 1. Otvorte si skript „**ThreeScene.js**“ a html súbor „**index.html**“ vo vašom vývojovom prostredí.
- 2. Html súbor „**index.html**“ **rozšírte o skript :**

```
<script src="js/threejs/threex.KeyboardState.js"></script>
```

Uvedený skript spracováva vstupy podávané periférnym zariadením - klávesnica.

- 3. V skripte „**ThreeScene.js**“ deklarujte nasledujúce **globálne premenné:**

```
var clock = new THREE.Clock();
var keyboard = new THREEEx.KeyboardState();
```

3. THREE.JS - IMPLEMENTÁCIA

METÓDY UPDATE() – MAPOVANIE VSTUPOV PERIFÉRIÍ

- *Táto metóda bude slúžiť pre získavanie a spracovávanie vstupov z periférií.*
- **Úloha:** V skripte „**ThreeScene.js**“ vytvorte na konci metódu **update()**, následne do metódy vložte kód:

```
var delta = clock.getDelta();
var moveDistance = 2 * delta;
var rotateAngle = Math.PI / 2 * delta;
```

3. THREE.JS - IMPLEMENTÁCIA

METÓDY UPDATE() – MAPOVANIE VSTUPOV PERIFÉRIÍ

1. Následne v skripte „**ThreeScene.js**“ v metóde **update()** **vložte kód:**

```
if ( keyboard.pressed("W") )
    cubel.translateZ( -moveDistance );
if ( keyboard.pressed("S") )
    cubel.translateZ( moveDistance );
if ( keyboard.pressed("Q") )
    cubel.translateX( -moveDistance );
if ( keyboard.pressed("E") )
    cubel.translateX( moveDistance );

if ( keyboard.pressed("up") )
    cubel.position.z -= moveDistance;
if ( keyboard.pressed("down") )
    cubel.position.z += moveDistance;
if ( keyboard.pressed("left") )
    cubel.position.x -= moveDistance;
if ( keyboard.pressed("right") )
    cubel.position.x += moveDistance;

controls.update();
```

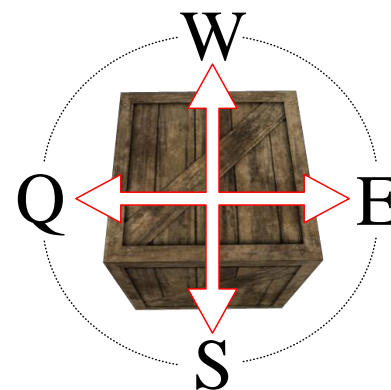
3. THREE.JS - ÚPRAVA IMPLEMENTÁCIE METÓDY RENDER()

- **Úloha:** V skripte „**ThreeScene.js**“ upravte implementáciu metódy **render()** nasledovne :

1. V metóde **render()** vymažte riadok s volaním *controls.update()*; nahradte ho **riadkom** *update()*; s volaním novo-implementovanej metódy.

controls.update(); → *update();*

2. Otestujte správnosť re-implementácie.
Teraz je možné **posun objektu kocky ovládať klávesami**: W,S,Q,E,
rovnako aj šípkami ←, →, ↑, ↓.



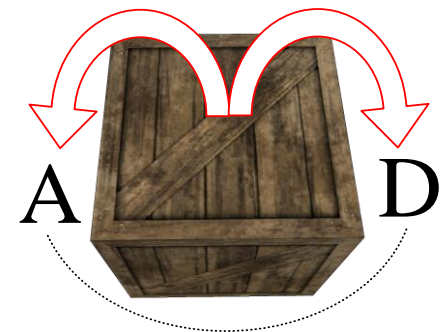
3. THREE.JS - ÚPRAVA IMPLEMENTÁCIE METÓDY UPDATE()

- **Úloha:** V skripte „**ThreeScene.js**“ upravte implementáciu metódy **update()** nasledovne :

3. V metóde **update()** vložte nasledujúci kód :

```
var rotation_matrix = new THREE.Matrix4().identity();
if ( keyboard.pressed("A") )
    cubel.rotateOnAxis( new THREE.Vector3(0,1,0),rotateAngle);
if ( keyboard.pressed("D") )
    cubel.rotateOnAxis( new THREE.Vector3(0,1,0),-rotateAngle);
```

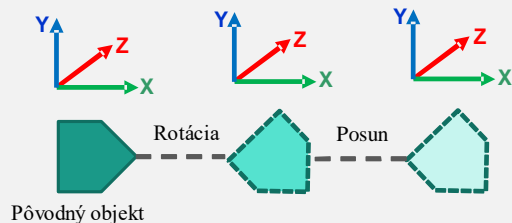
- **Pozor !** Kód musí byť vložený pred volaním metódy *controls.update()*;
4. Otestujte správnosť re-implementácie.
Teraz je možné **objekt otáčať klávesami: A, D.**



4. THREE.JS - POUŽÍVANIE VEKTOROV PRI RIADENÍ POHYBU 3D OBJEKTU

- Pre zachovanie prirodzeného správania a ovládania 3D objektu je použitie vektorov nevyhnutné. Rozdiel riadenia pohybu bez a s použitím vektora je zobrazený na nasledujúcom obrázku.

bez použitia vektora

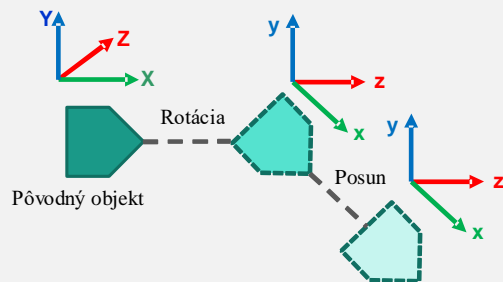


Demonštračný príklad implementácie

```
cube1.position.x = moveDistance;
cube1.rotateOnAxis( new THREE.Vector3(0,1,0), rotateAngle );
cube1.position.x = moveDistance;
```

Vyskúšajte kombinovať **pohyb** objektu pomocou **šípok** a **rotáciu** pomocou kláves **A,D**.

s použitím vektora



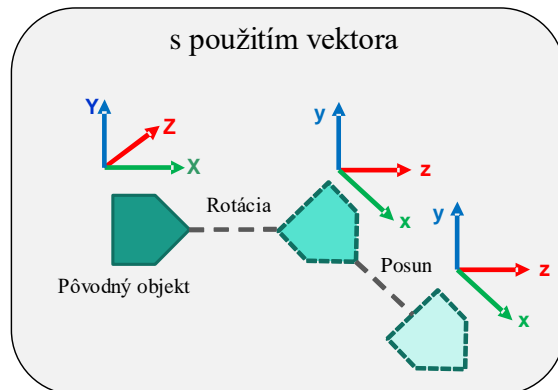
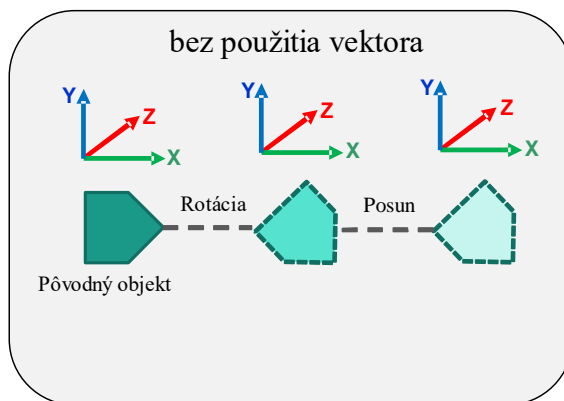
Demonštračný príklad implementácie

```
cube1.translateX( moveDistance );
cube1.rotateOnAxis( new THREE.Vector3(0,1,0), rotateAngle );
cube1.translateX( moveDistance );
```

Vyskúšajte kombinovať **pohyb** objektu pomocou kláves **W,A,S,D** a **rotáciu** pomocou kláves **A,D**.

4. THREE.JS - POUŽÍVANIE VEKTOROV PRI RIADENÍ POHYBU 3D OBJEKTU

- Pre zachovanie prirodzeného správania a ovládania 3D objektu je použitie vektorov nevyhnutné. Rozdiel riadenia pohybu bez a s použitím vektora je zobrazený na nasledujúcom obrázku.



Demonštračný príklad implementácie

```
cube1.position.x += 1;
cube1.rotation.y += 0.01;
cube1.rotation.z += 0.01;
```

Všimli ste si rozdiel ?

```
cube1.position.x += 1;
cube1.rotation.y += 0.01;
cube1.translateX( moveDistance );
```

Vyskúšajte kombinovať **pohyb** objektu pomocou kláves **W,A,S,D** a **rotáciu** pomocou kláves **A,D**.

DOPLŇUJÚCE ÚLOHY

- Vyskúšajte **implementovať ovládanie vlastného objektu** pomocou **využitia vstupov klávesov I,J,K,L** pre **pohyb vpred, vzad, doľava a doprava**.
Následne integrujte ovládanie pomocou šípok pre vami ďalší zvolený objekt, tak aby ste mohli separátne ovládať 2 objekty odlišnými vstupmi.
- Otestujte paralelné ovládanie na jednej klávesnici s inými študentmi na cvičení.



ÚLOHY NA SAMOSTATNÉ RIEŠENIE

- Pokúste sa pomocou vybraných klávesov ovládať rotáciu oblohy (Skybox). Sledujte vizuálny efekt tejto implementácie.
- Implementujte vami zvolené klávesy pre ovládanie všetkých základných transformácií vybraného objektu (posun, rotácia, zmena mierky)
- Pomocou *Three.js* 3D primitív a vzťahu *Rodič↔Potomok* vytvorte niektorému z ovládaných objektov kolesá (cylinder).



Q & A

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

Katedra počítačov a informatiky, FEI TU v Košiciach

© 2024