



Univerza v Ljubljani
Fakulteta
za računalništvo
in informatiko

DOKUMENTACIJA

Navidezna fakulteta v WebGLu

Avtorji:

Miha ZIDAR

Anže PEČAR

Aleksandra BERSAN

1. junij 2011

1 Opis rešitve

1.1 Kaj rešitev sploh je

Vsi vemo, da ima naša fakulteta zelo nelogično razporejene prostore in pogosto se zgodi, da obiskovalec ne najde prave predavalnice. Da bi obiskovalcem in brucem olajšali življenje, smo se odločili narediti 3-D model fakultete.

Ker želimo, da bi bila naša aplikacija čim lažje dostopna, smo jo postavili na internet s pomočjo odprtokodne knjižnice WebGL. Potrudili smo se, da aplikacija teče tekoče na sodobnih brskalnikih z WebGL podporo.

1.2 Uporabljene metode



Jeziki

- JavaScript - Delo z gl knjižnico
- GLSL - za Fragment in Vertex shaderja
- Python - Backend, pretvarjanje .obj datotek v json

Ogrodja

- webgl-utils - Za funkcijo requestAnimationFrame
- glMatrix - Uporabne funkcije nad matrikami in vektorji
- jQuery - Izboljšanje uporabniške izkušnje
- Django - Backend za shranjevanje podatkov o mestih zanimanja v podatkovno bazo

Orodja

- Blender - Modeliranje faksa po gradbenih načrtih
- Gimp - Izdelava tekstur

Algoritmi

- Mipmap texture
- Phongov odbojni model
- Collision detection

2 Zabavni deli

2.1 Pretvarjanje iz .obj v json in parsanje

Za izrisovanja modelov iz Blenderja je potrebno prebrati datoteko v .obj formatu in iz zapisa izluščiti *vertexe*, *normale*, *face* in različne materiale. Za ta namen smo napisali preprosto python skripto, ki .obj datoteko prebere in prebrane podatke zapiše v json formatu. Na ta način smo pohitrili začetek izrisovanja, saj spletnemu brskalniku ni potrebno parsati .obj datoteke ob vsaki naložitvi strani.

2.2 Phongov odbojni model

2.3 Collision detection

3 Problemi

3.1 Blacklisted GPUs

Še preden smo začeli z razvojem naše seminarske naloge, smo naleteli na probleme. Tako Googlov Chrome, kot Mozillin Firefox sta imela na črni listi grafične kartice, ki so bile v naših računalnikih, kar je pomenilo, da je bil WebGL onemogočen. WebGLa ni bilo možno prisiliti k delovanju iz nastavitvev znotraj brskalnika pa čeprav smo šarili po nastavitvah za developerje (Chromov: `about:flags` in FFjev: `about:config`). Po nekaj dneh Googlanja smo le naleteli na sistemsko spremenljivko `MOZ_GLX_IGNORE_BLACKLIST=1`, s katero smo omogočili WebGL znotraj FireFoxa in Chromovo `--ignore-gpu-blacklist` zagonsko stikalo.

Z najnovejšo različico Google Chromea nastavljanje zagonskega stikala ni več potrebno, FireFox4 pa še vedno potrebuje nastavljeno sistemsko spremenljivko.

3.2 Lepljenje tekstur

3.3 Nesreca enega člana ekipe

4 Posnetki zaslonov