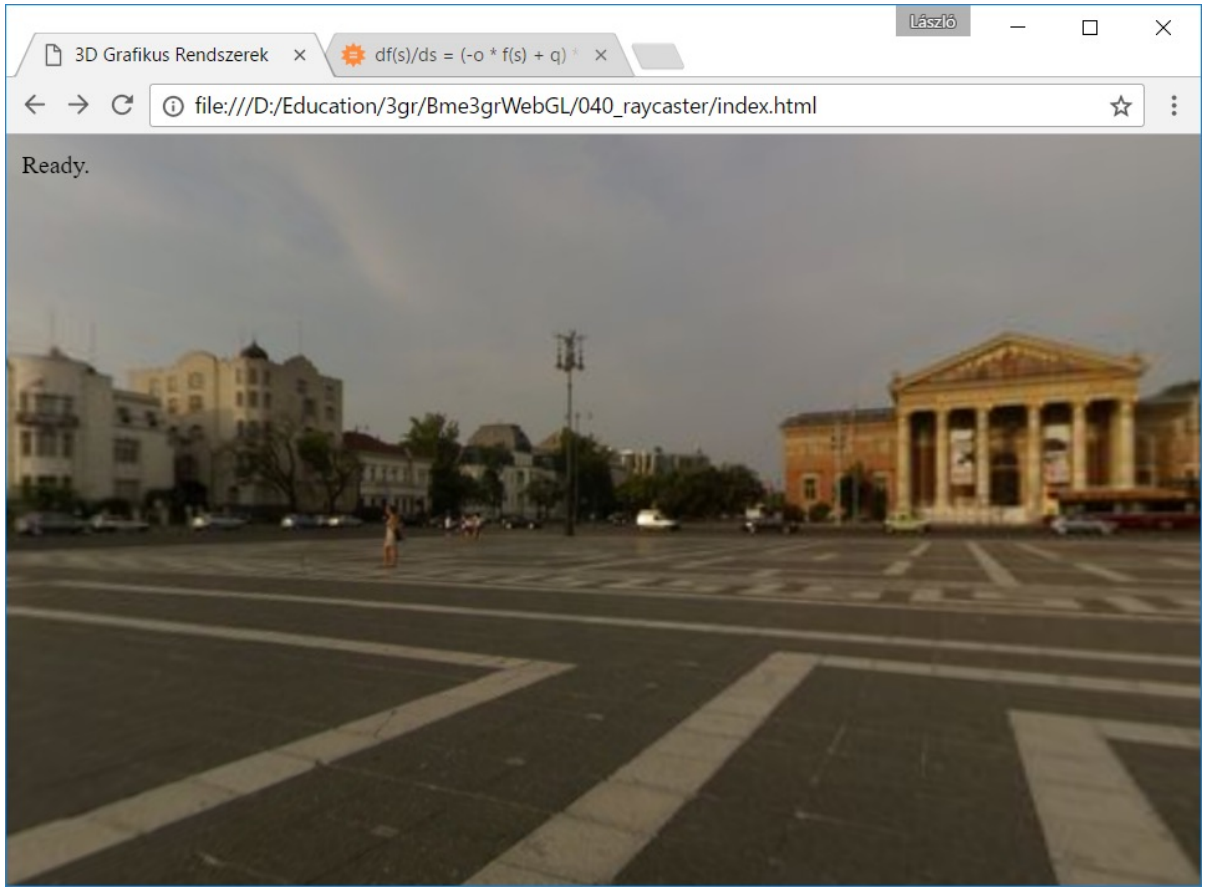
Szintfelület sugárkövetése

Szécsi László

3D Grafikus Rendszerek

6. labor

Indulás: environment háttér



Terep textúrából

```
uniform struct {
    samplerCube envTexture;
    sampler2D noiseTexture;
} material;

float f(vec3 p){
   return p.y - texture(material.noiseTexture,
                        p.xz * 0.01).r;
}
```

terep implicit függvénye

y = 0 sík implicit egyenlete, ehhez keverjük a 2D zajt

freki

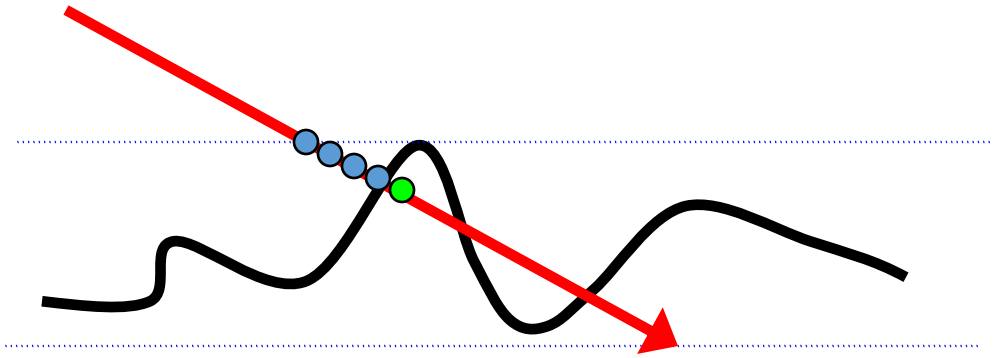
Kilépési, belépési pont

```
uniform struct {
  mat4 rayDirMatrix;
  vec3 position;
} camera;
in vec4 rayDir;

void main(void) {
  vec3 d = normalize(rayDir.xyz);
  float t1 = ?sugár hol metszi az y=1 síkot?;
  float t2 = ?sugár hol metszi az y=0 síkot?;
  float tstart = max(min(t1, t2), 0.0);
  float tend = max(t1, t2);
```

Lineáris keresés (ray marching)

- Első pont megtalálás legyen biztos



Ray marching

```
bool found = false;
vec3 p, color;

if(tstart < tend) {
    p = camera.position + d * tstart;
    vec3 step = d * min((tend - tstart)/128.0, 0.05);
// feladat:
// ciklus fut 128-szor
// p léptetése step-pel
// ha f(p)<0 (átléptük a 0 szintfelületet):
//         found=true; break;
}
```

Magasság szerinti árnyalás

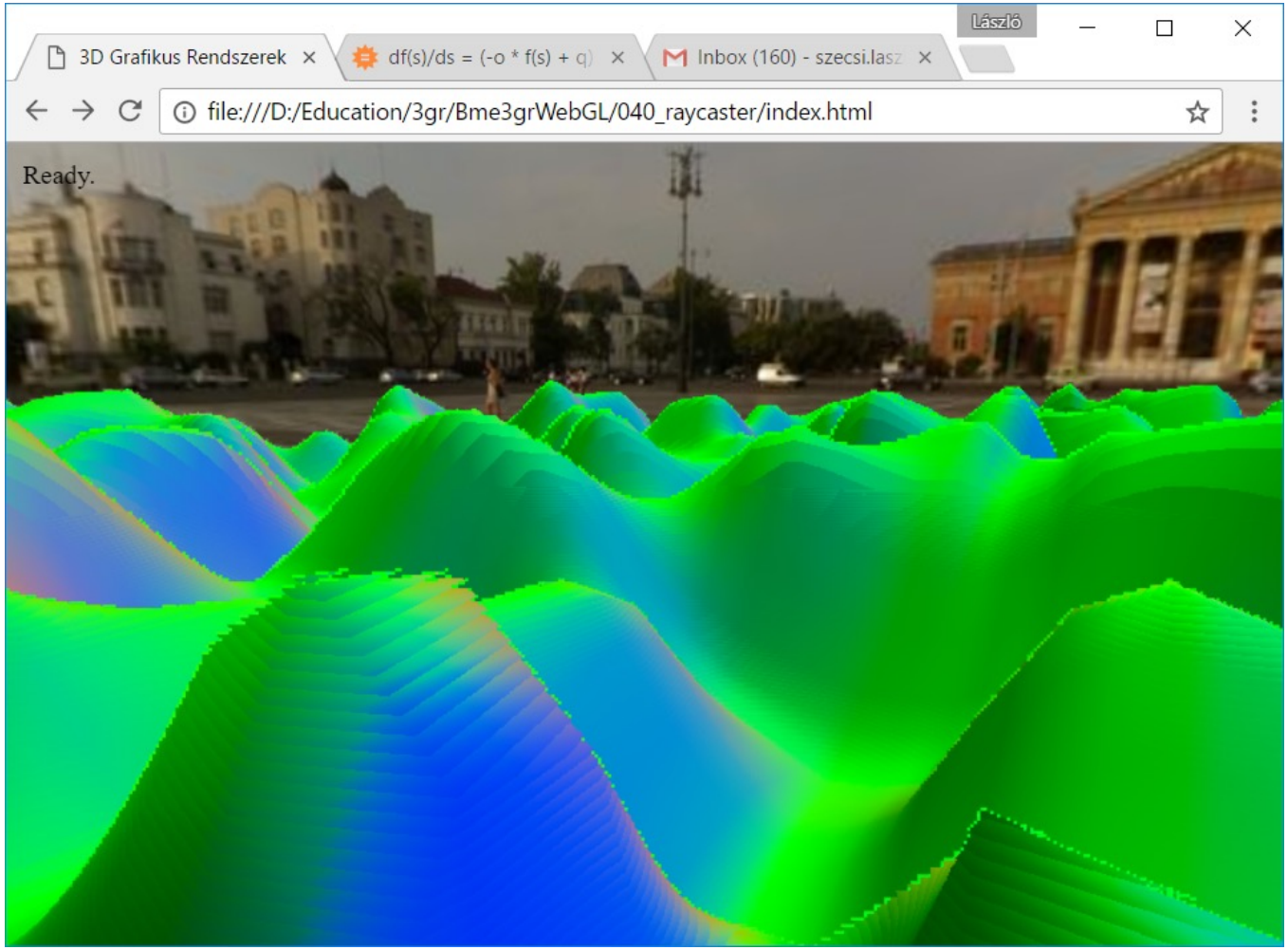
- ha nem értük el a felületet, akkor adjuk vissza a háttérszínt
- különben **`p.y`**-t

Várt eredmény

Normálvektor megjelenítése

```
vec3 fGrad(vec3 p){
  return vec3(
    f(p + vec3(+0.05, 0.0, 0.0) ) -
    f(p + vec3(-0.05, 0.0, 0.0) ) ,
    f(p + vec3(0.0, +0.05, 0.0) ) -
    f(p + vec3(0.0, -0.05, 0.0) ) ,
    f(p + vec3(0.0, 0.0, +0.05) ) -
    f(p + vec3(0.0, 0.0, -0.05) )
    );
}
// számoljuk a gradienst a metszéspontban,
normalizáljuk, adjuk vissza, mint színt
```

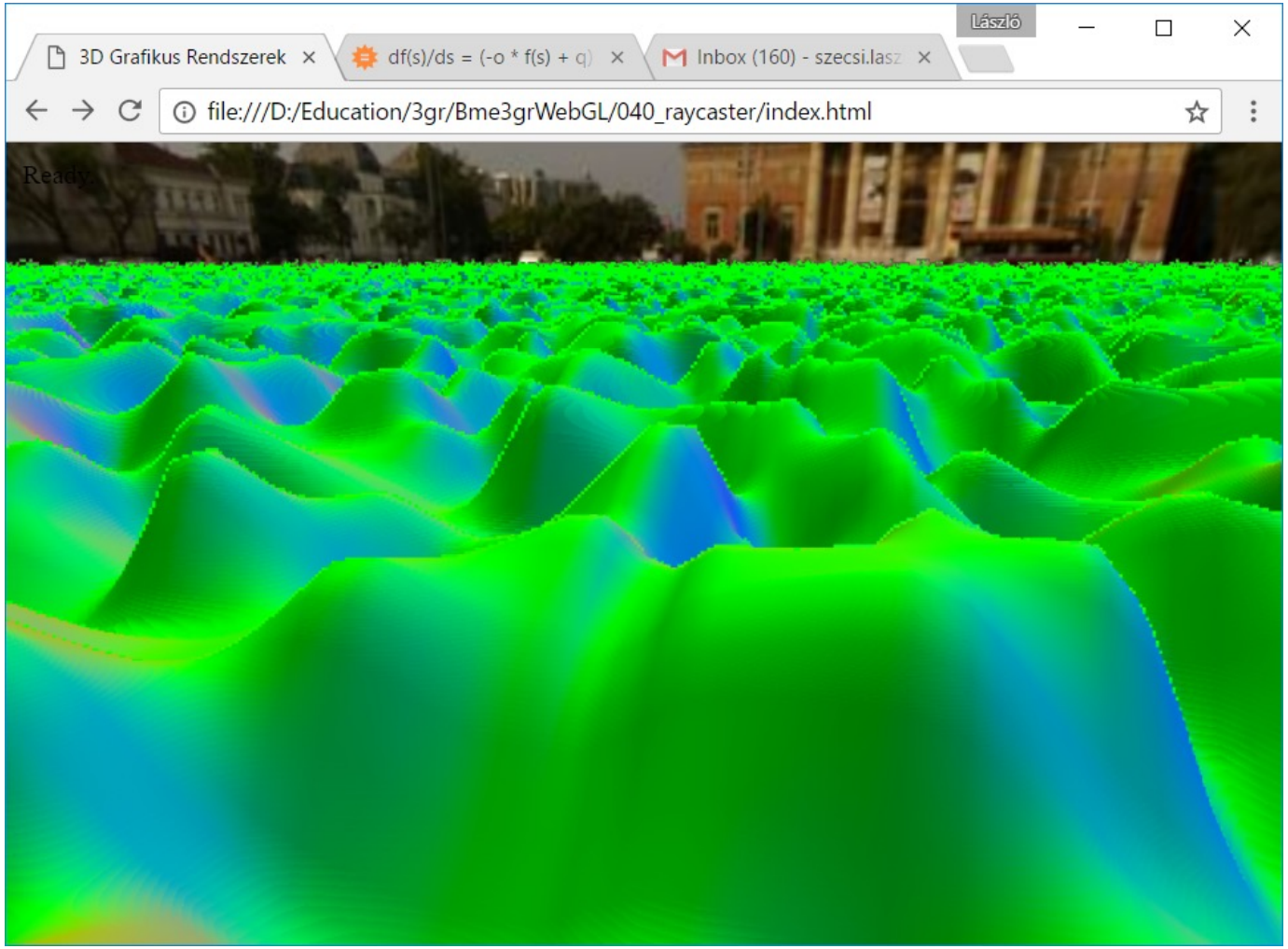
Várt eredmény



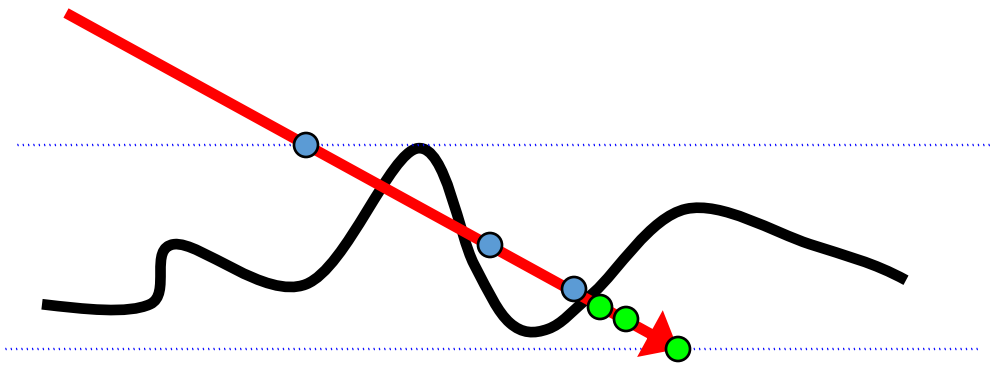
Változó lépéshossz

- ray marchingban minden ciklusban növeljük a lépést 2%-kal
- kezdeti lépés nem a teljes táv 128-ada, hanem 580-ada (geometriai sor)
- minimumlépés csökkenthető pl. 0.01-re

Várt eredmény



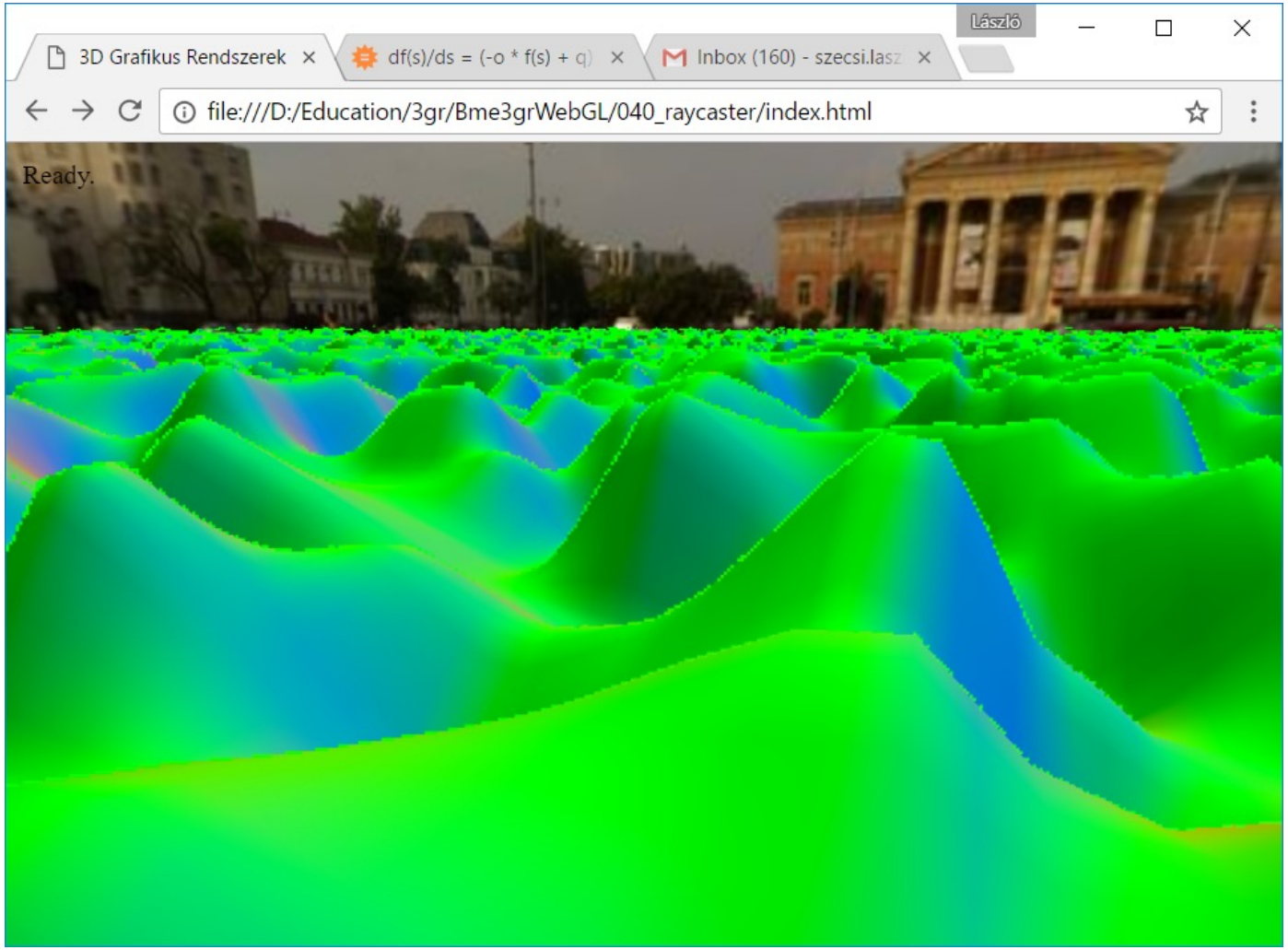
Bináris keresés



Eredmény finomítása bináris kereséssel

- `step` legyen a legutolsó lépés fele
- kezdőpozíció a két utolsó érték között félúton
  - tehát lépjünk egyet visszafelé új `step`-pel
- ciklusban (kb. 16x)
  - felezzük a lépést
  - ha kint vagyunk, előre lépünk
  - ha bent vagyunk, visszább lépünk

Várt eredmény



Cseréljük le a textúrából vett magasságértékeket 3D implicit zajfüggvényre

- nem magasságmező, lesznek hidak is
- normálvektor analitikusan számolható

Egyszerű zajfüggvény és gradiense

```
float noise(vec3 r) {
  uvec3 s = uvec3(
    0x1D4E1D4E,
    0x58F958F9,
    0x129F129F);
  float f = 0.0;
  for(int i=0; i<16; i++) {
    vec3 sf =
    vec3(s & uvec3(0xFFFF))
  / 65536.0 - vec3(0.5, 0.5, 0.5);

    f += sin(dot(sf, r));
    s = s >> 1;
  }
  return f / 32.0 + 0.5;
}
```

```
vec3 noiseGrad(vec3 r) {
  uvec3 s = uvec3(
    0x1D4E1D4E,
    0x58F958F9,
    0x129F129F);
  vec3 f = vec3(0, 0, 0);
  for(int i=0; i<16; i++) {
    vec3 sf =
    vec3(s & uvec3(0xFFFF))
  / 65536.0 - vec3(0.5, 0.5, 0.5);

    f += cos(dot(sf, r)) * sf;
    s = s >> 1;
  }
  return f;
}
```

Sík + egyszerű zaj

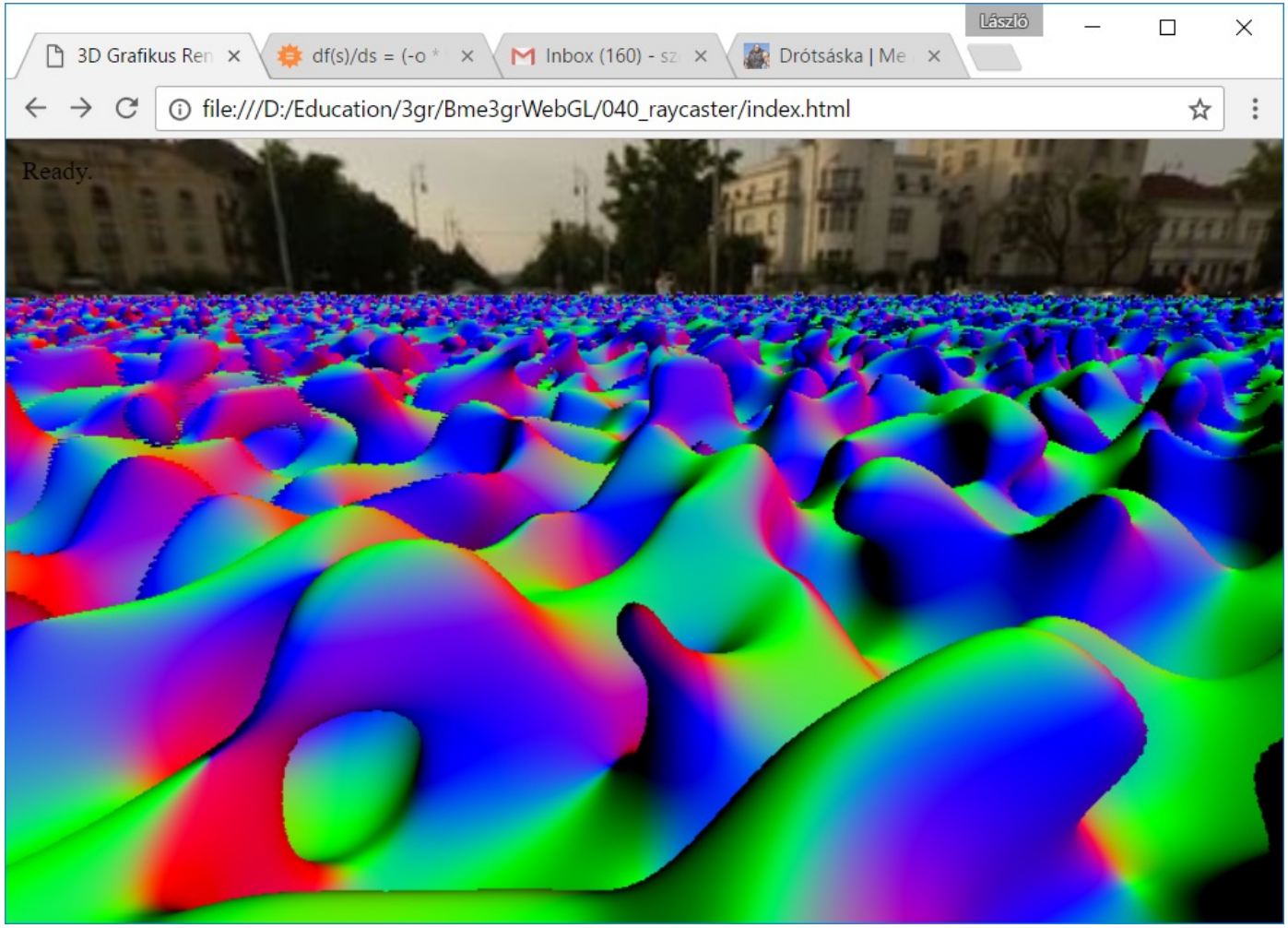
freki

```
float f(vec3 p){
  return p.y - noise(p * 50.0);
}
```

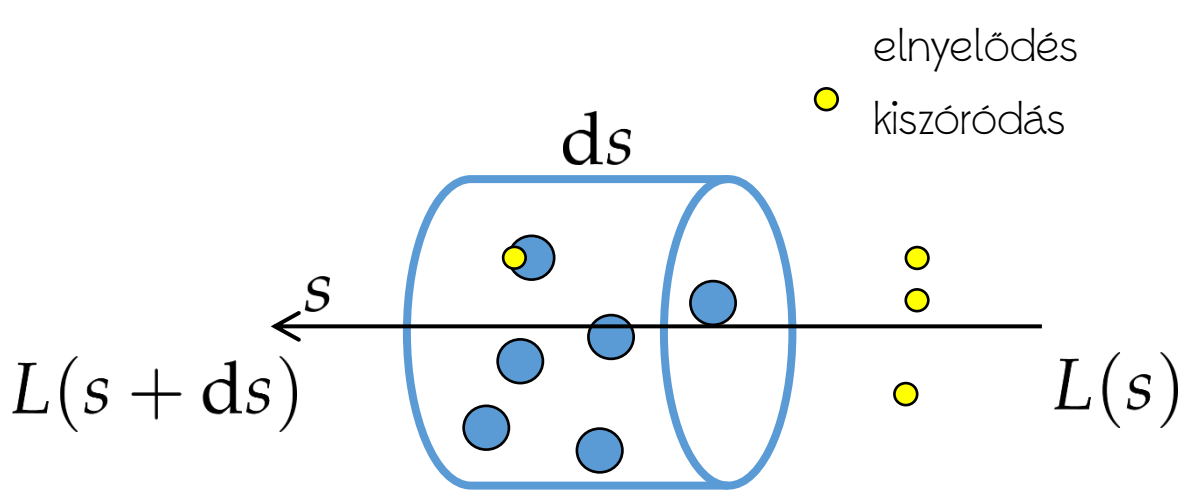
y = 0 sík implicit egyenlete, ehhez keverjük a 3D zajt

```
vec3 fGrad(vec3 p){
  return vec3(0, 1, 0) - noiseGrad(p * 50.0);
}
```

Várt eredmény



Köd



radiancia változása

csillapítási tényező

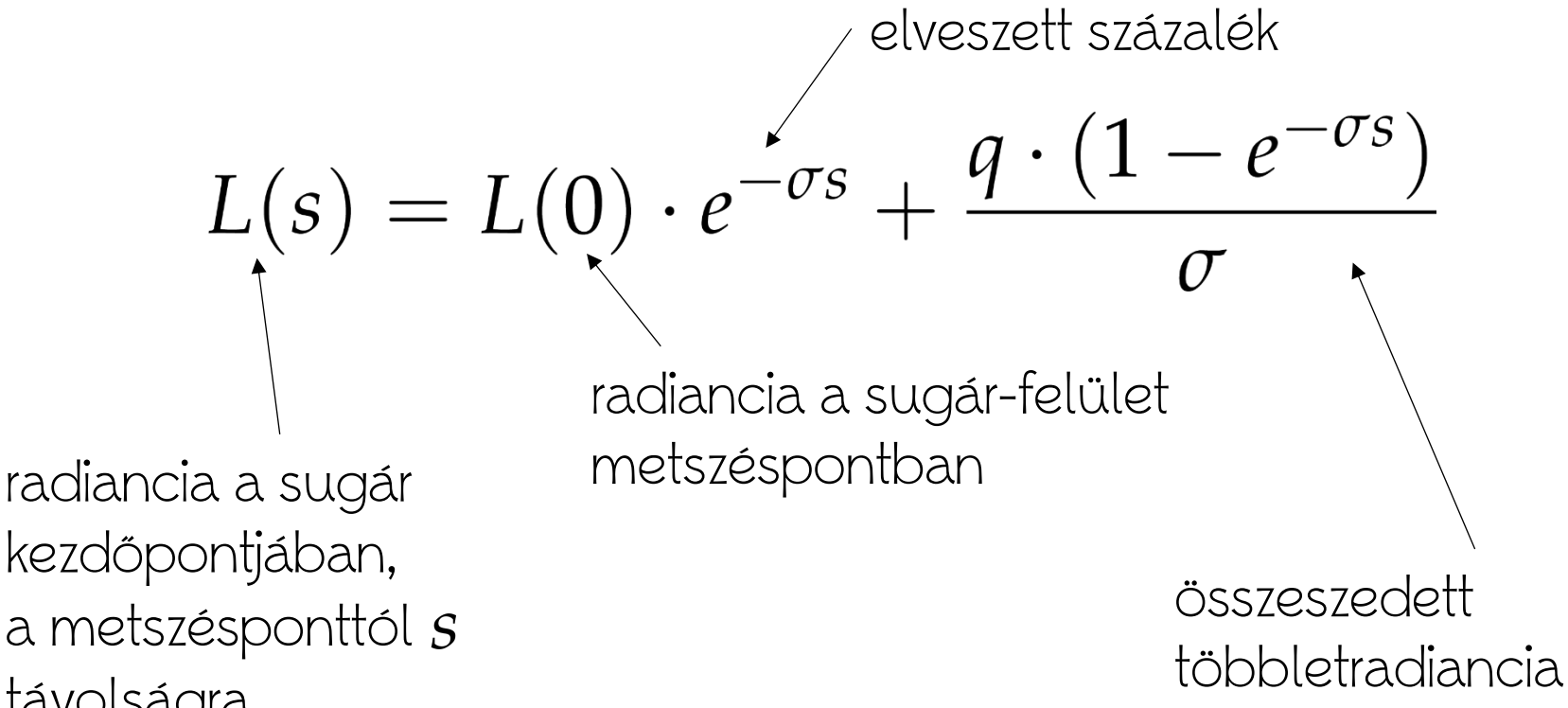
forrástag

$$\frac{\mathrm{d}L}{\mathrm{d}s} = -\sigma \cdot L + q$$

egység hosszon a sugár mentén

radiancia épp itt

Diffegyenlet megoldása

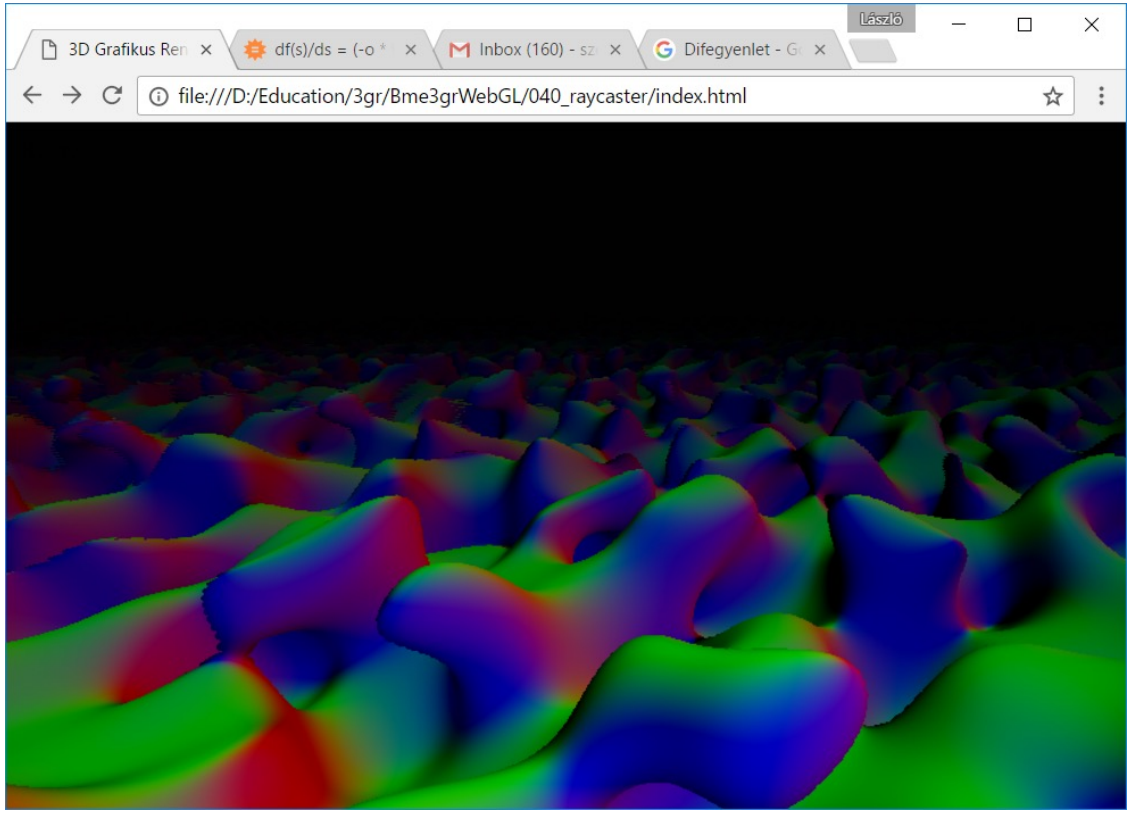


Feladat: köd

- válasszunk csillapítást (kb. 200) és forrástagot (kb. 200), tudjuk ezeket könnyen változtatni
  - ergo érdemes ezekre változót gyártani
  - lehet skalár, vagy RGB, ha színes a köd
- a fenti képletet értékeljük ki a FS végefelé
- ha nem volt metszéspont, akkor végtelen (vagy csak nagyon nagy) a távolság (a háttér végtelen messze van)

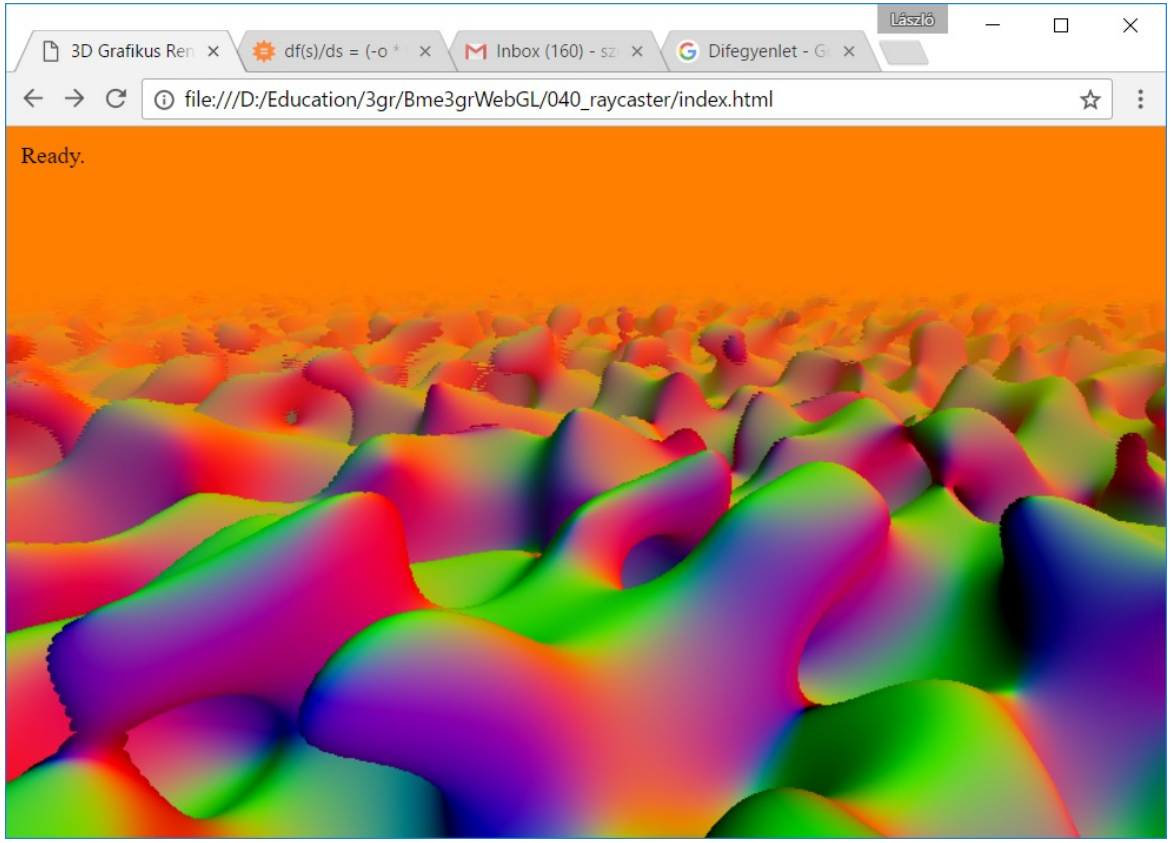
Várt eredmény

- nulla forráshatás, fekete köd



Várt eredmény

- színes köd, nem látszik a háttér (nyilván)



Nem-konstans köd

- lejjebb sűrűbb, feljebb ritkább, exponenciálisan
  - válasszunk alkalmas b értéket (kb. -7)

$$\sigma(y) = \sigma_0 e^{b \cdot y}$$

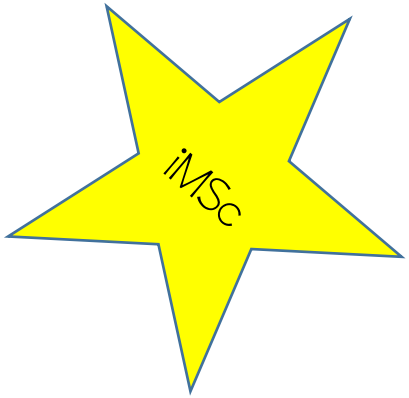
$$q(y) = q_0 e^{b \cdot y}$$

- y a sugáron

$$y(s) = y_{\text{ray\_origin}} - y_{\text{ray\_dir}} \cdot s$$

$$\frac{\mathrm{d}L}{\mathrm{d}s} = (-\sigma \cdot L + q) e^{b \cdot \left(y_{\text{ray\_origin}} + y_{\text{ray\_dir}} \cdot s\right)}$$

Diffegyenlet megoldás



$$L(s) = \frac{L(0)\cdot\sigma - q}{\sigma} e^{-\frac{\sigma}{b\cdot y_{\text{ray\_dir}}}\left(e^{b\cdot\left(y_{\text{ray\_origin}}+s y_{\text{ray\_dir}}\right)} - e^{b\cdot y_{\text{ray\_origin}}}\right)} + \frac{q}{\sigma}$$

Várt eredmény