

情報可視化論最終レポート

2018/06/15

システム情報学研究科 CS25

183x120x 吉見 脩平

<概要>

今回の課題では、元々の **Isovalue** の変更機能、ロブスターの色の変更機能を追加した。初めに、ボタン機能を用いての上記の機能実装を行った。左上の部分に表示させている。次に、w11 で紹介された **User Interface** の **dat.UI** を用いて同様の機能を実装させた。右上に表示させている。機能としては、重複しているが感覚的な使用は異なりそれぞれに使いやすさがあるので、今回はあえて二通りの機能を採用した。

なお、“**Lambertian Reflection**”、“**Phong Reflection**”、“**Blinn-Phong Reflection**” の 3 種類の **Reflection** の変更機能、“**Gouraud Shading**”、“**Phong Shading**” の 2 種類の **Shading** の変更機能を試みたがそれぞれ単独の使用のみしか実装できず変更機能としてはうまく実装することができなかった。よって、**Reflection** には“**Phong Reflection**”を、**Shading** には“**Phong Shading**”を設定している。

初期状態の実行画面を図 1 に示す。**Isovalue=28** に設定、**color** は以下のように赤 (**#CC0000**) に設定した。なおブラウザは **Firefox** を用いた。

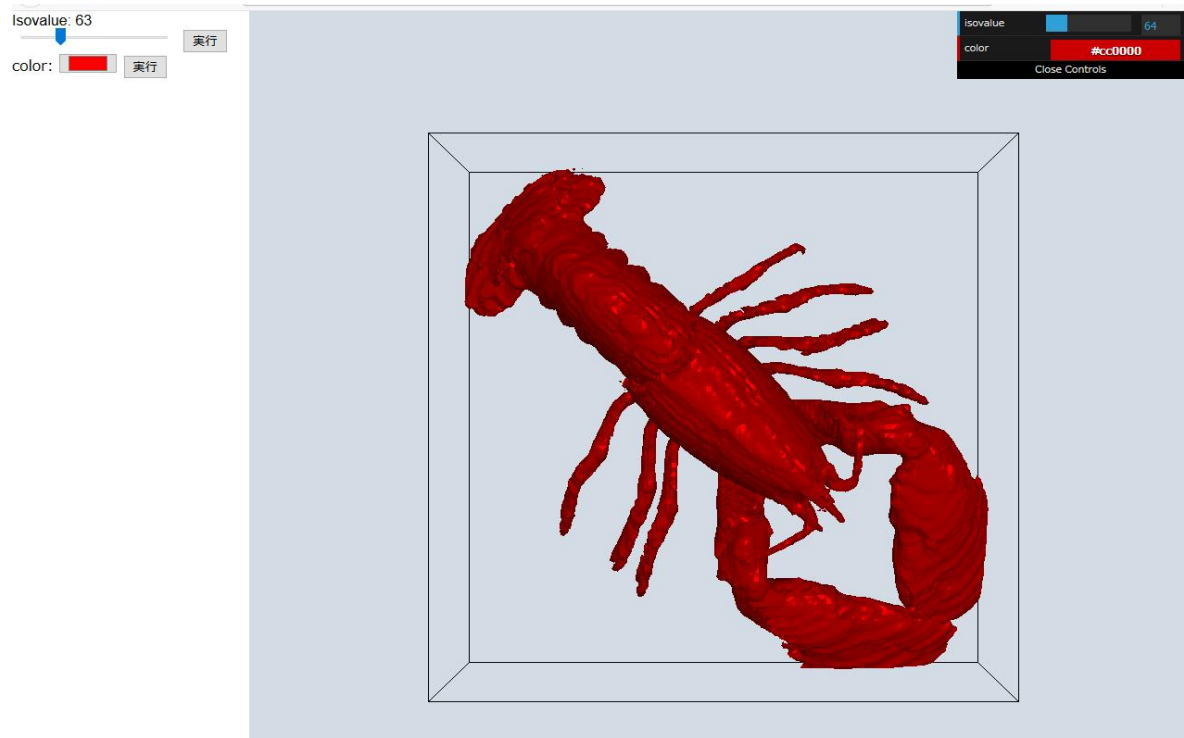


図 1. 初期画面

<詳細>

簡単ではあるが、js ファイルの実装部分の詳細を示す。

- ・ ボタン機能を用いての場合

Html ファイルでボタン id 等を実装したのちに、document.getElementById('change-iso-value-button').addEventListener('click', function(){

のように実行ボタンが押されると以下の内側で記したものが実行される。

前のものをいったん消す⇒受け取った値（選択された iso-value や color の値）を代入⇒表面の設定を計算⇒適応させる
といった内容になっている。

```
document.getElementById('label').innerHTML = "Isovalue: " + isovalue ;

document.getElementById('iso-value').addEventListener('mousemove', function() {
    var isovalue = +document.getElementById('iso-value').value;
    document.getElementById('label').innerHTML = "Isovalue: " + isovalue ;
});

document.getElementById('change-iso-value-button').addEventListener('click',
function() {
    screen.scene.remove( surfaces );
    isovalue = +document.getElementById('iso-value').value;
    surfaces = Isosurfaces( volume, isovalue, screen, d_color);
    screen.scene.add( surfaces );
});

document.getElementById('change-color-button').addEventListener('click',
function() {
    screen.scene.remove( surfaces );
    d_color = document.getElementById('d_color').value;
    surfaces = Isosurfaces( volume, isovalue, screen, d_color);
    screen.scene.add( surfaces );
});
```

- dat.UI を用いての場合

`controller.onFinishChange` の部分でマウスの動作が終わったとき (`isovalue` や `color` の値が選択された後) その内側で記したものが実行される。前のものをいったん消す⇒受け取った値 (選択された `isovalue` や `color` の値) を代入⇒表面の設定を計算⇒適応させる
といったボタン機能を用いた場合と同様な内容になっている。

```
var FizzyText = function() {
    this.isovalue = 64;
    this.color = "#CC0000";
};

var text = new FizzyText();
var gui = new dat.GUI();
var controller = gui.add(text, 'isovalue', 0, 255);
var controller2 = gui.addColor(text, 'color');

controller.onFinishChange(function(value) {
    screen.scene.remove( surfaces );
    isovalue = value;
    surfaces = Isosurfaces( volume, isovalue, screen, d_color);
    screen.scene.add( surfaces );
});

controller2.onFinishChange(function(d_value) {
    screen.scene.remove( surfaces );
    d_color = d_value;
    surfaces = Isosurfaces( volume, isovalue, screen, d_color);
    screen.scene.add( surfaces );
});
```

<実行結果>

いくつか実行例を記載する。

- ・ ボタン機能を用いての場合

Isovalue=28 に設定、color は以下のように緑に設定した場合の実行例 1 を図 2 に示す。

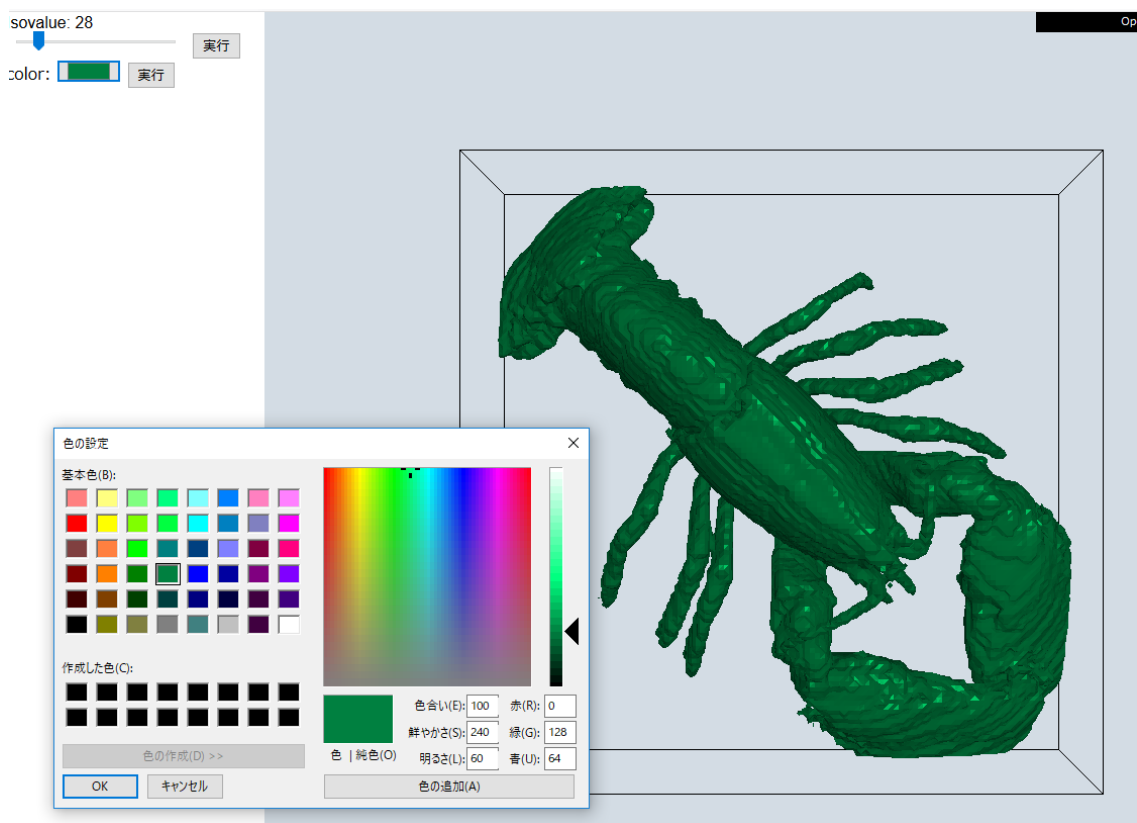


図 2. 実行例 1

- dat.UI を用いての場合

Isovalue=157 に設定、color は以下のように青（#2f56b3）に設定した。

