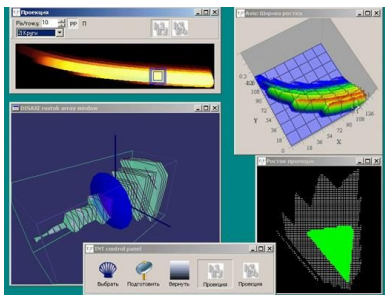




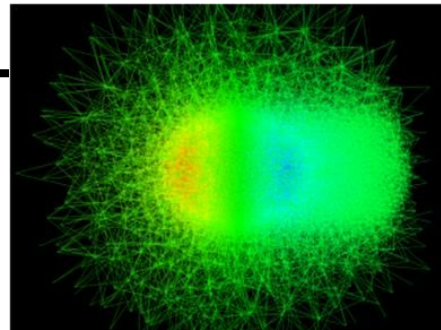
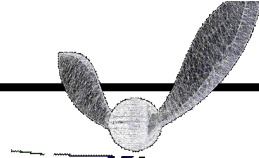
Новый язык описания сцен научной визуализации

Михаил Бахтерев, Павел Васёв, Дмитрий Манаков

Сектор компьютерной визуализации
ИММ УРО РАН им. Н.Н. Красовского

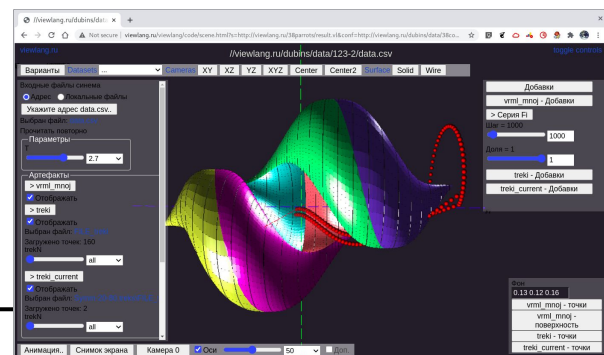
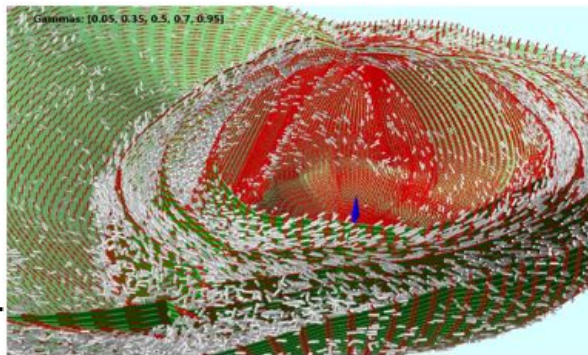
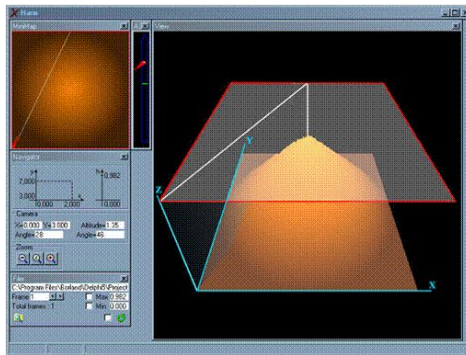
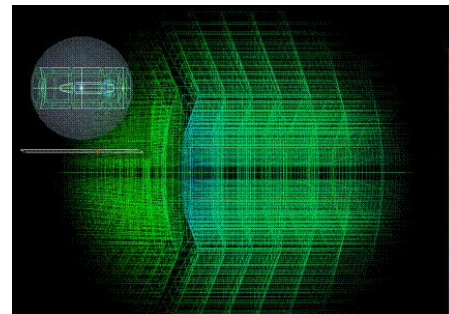
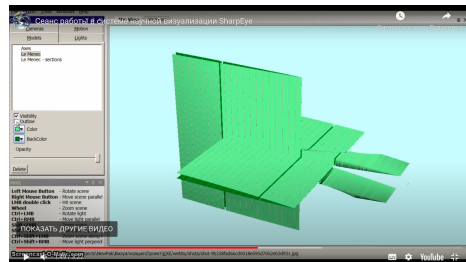


Сектор компьютерной визуализации



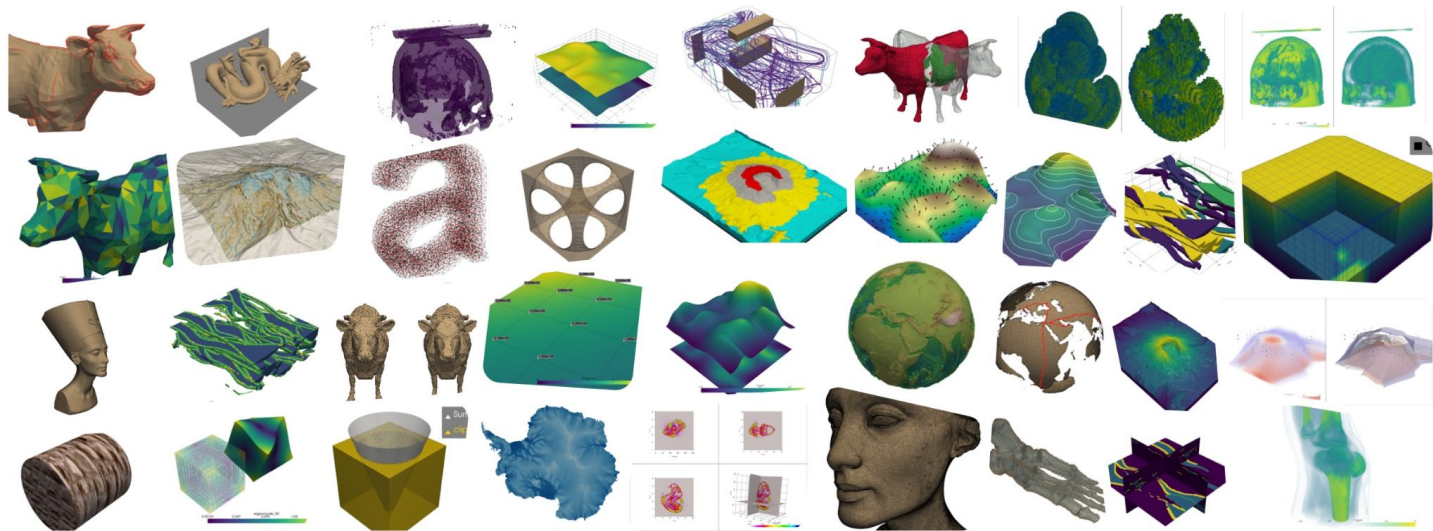
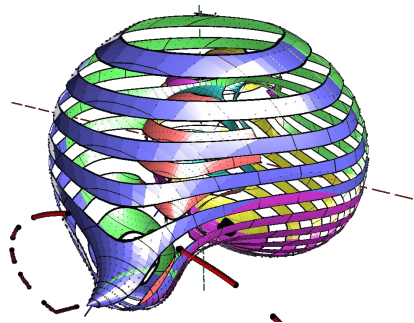
Научный руководитель до 2020
Владимир Лазаревич Авербух

www.cv.imm.uran.ru



Виды систем научной визуализации

- Универсальные
- Специализированные



Постановка задачи

Требуется язык программирования специализированных систем визуализации. Потребности, которые данный язык должен уметь программировать:

- Загрузка данных.
 - Содержимое и динамика 3D сцены.
 - Преобразование данных в элементы сцены.
 - 2D пользовательский интерфейс.
 - Взаимодействие с пользователем.
-

Существующие языки

Визуальные языки (среды) ~ универсальные системы

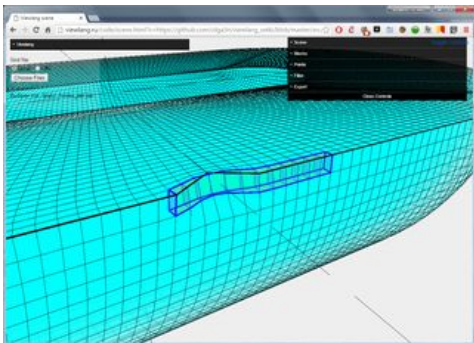
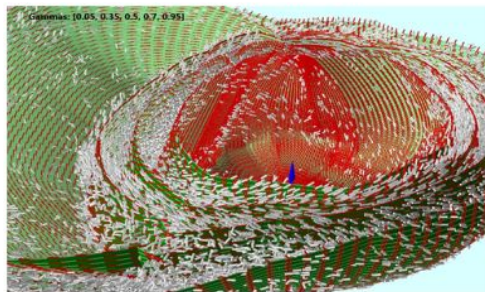
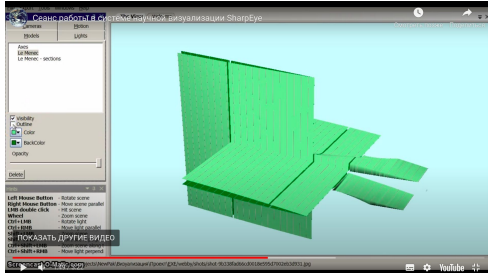
Excel, ParaView, Visit

Текстовые языки и среды

Python + Matplotlib, Streamlit, Vega grammar, A-Frame
+ встроенные в пакеты, например Matlab

Инновационные / очень визуальные

Enso.org, SciVi.tools, KNIME.com



История нашего исследования

2011 SharpEye

2012 JUJ

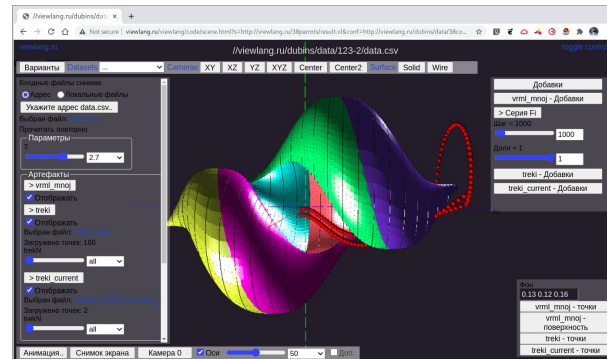
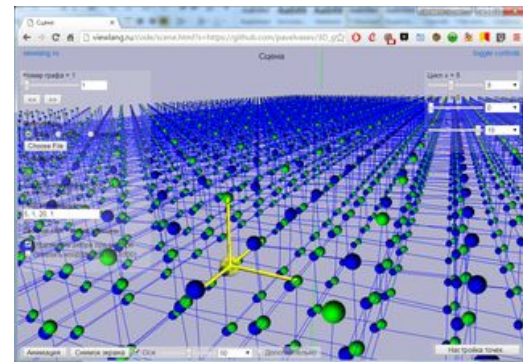
2014 Viewlang (3D QML)

2019 CodeMusic

2020 CinemaScience 3D

2021 Viewzavr

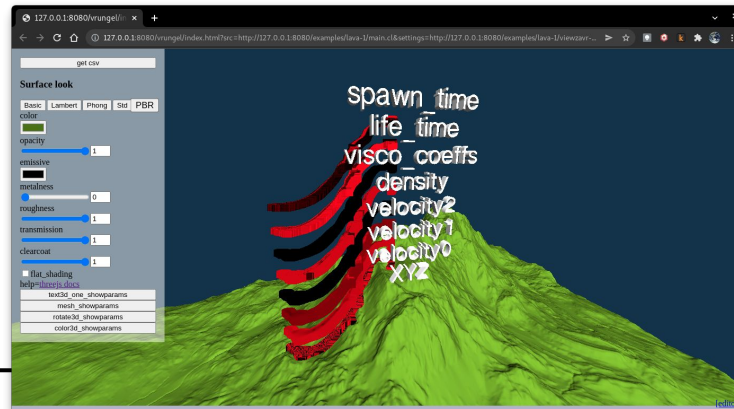
2022 Compolang / Vrungel



Программа описывается текстом (пока так).

Работает в браузере.

Может работать с данными из компьютера пользователя, из сети,
с суперкомпьютера...



Модель языка

Программа это набор **живых** объектов. Объекты связываются между собой через параметры. Синтаксис описания объекта:

```
имя: класс параметры.. улучшения.. {  
    вложенные объекты  
};
```

Пример (объявлено
2 объекта)

имя класс параметр и его значение

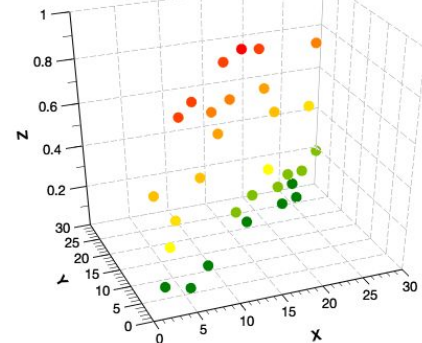
```
fdata: load_file file="a.csv" show_progress;
```

улучшение

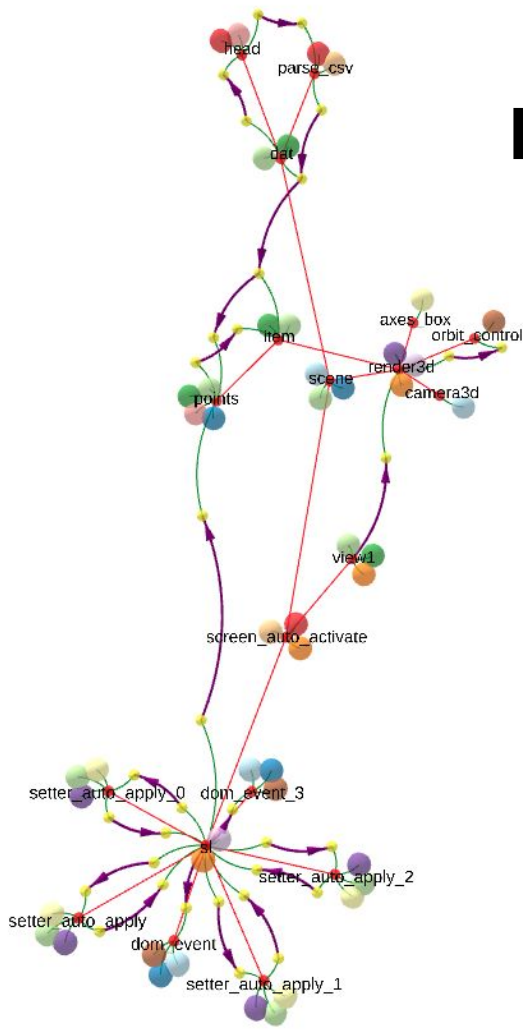
```
console_log text=@fdata->output;
```

ссылка

Пример программы



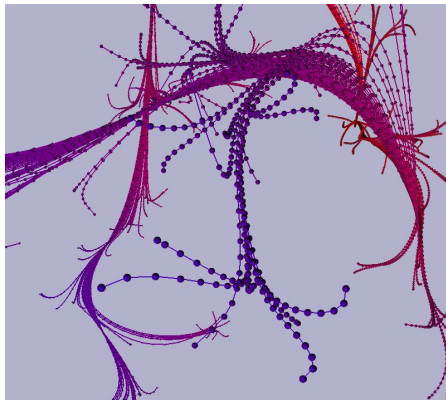
```
1.  dat : load-file file="a.csv" | parse_csv;
2.
3.  render3d target=@view1 {
4.    axes_box; orbit_control;
5.    @dat | points radius=@sl->radius;
6.  };
7.
8.  screen auto_activate {
9.    sl: slider min=10 max=100 step=5;
10.   view1: view3d fill_parent;
11.  };
```



Что сделано

- Разработана первая версия языка.
 - Сделан интерпретатор.
 - Разработано несколько библиотек
(2d графика, 3d графика, преобразования данных)
 - Решено несколько задач визуализации.
-

Решение задачи вывода траекторий мутаций вирусов. Автор Маджид Али Форгани

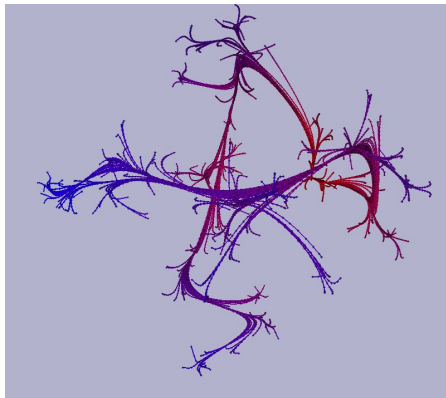


Дано: текстовый CSV файл с координатами наборов ломаных с колонками N,X,Y,Z,R,G,B,TEXT

Требуется: нарисовать набор ломаных, где каждая ломаная обозначается своим N, с цветами отрезков R,G,B и подписать текстом там, где поле TEXT не пустое.

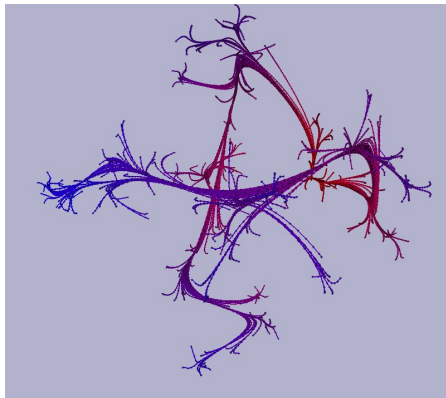
N,X,Y,Z,R,G,B,TEXT

1,-8.16343488150047,-0.027063223645974424,2.4203186454490675,71,0,184,НК/1/68
1,-8.232329989501082,-0.1678621188384646,1.9047405962436552,67,0,188,
1,-8.237565208277069,-0.39702100891016984,1.5645622351010444,63,0,192,
1,-8.20806786963832,-0.641349241463584,1.335140975282562,61,0,194,



Код программы

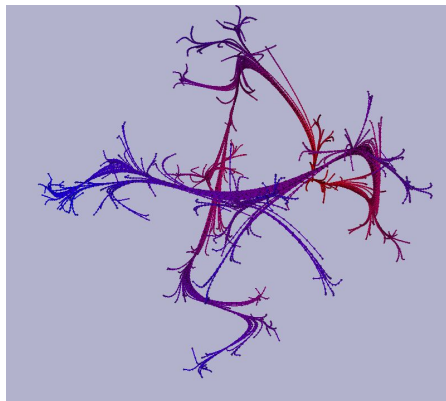
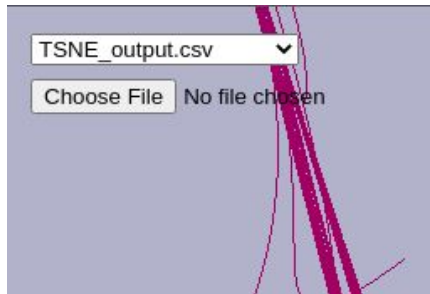
```
1.  load files="lib3d csv";
2.
3.  dat: load-file file="data.csv" | parse_csv;
4.
5.  render3d target=@view {
6.    orbit_controls;
7.
8.    @dat | linestrips;
9.  };
10.
11.  screen auto_activate {
12.    view: view3d fill_parent;
13.  };
```



Добавка пересчета данных

```
1.  load files="lib3d csv";
2.
3.  dat: load-file file="data.csv" | parse_csv | rescale_rgb;
4.
5.  render3d target=@view {
6.    orbit_controls;
7.
8.    @dat | linestrips;
9.  };
10.
11. screen auto activate {
12.   view: view3d fill_parent;
13. };
14.
15. register feature name="rescale rgb" code=`
16.   env.onvalue("input", (df) => {
17.     if (!df || !df.isDataFrame || !df['R']) return;
18.     df = df.clone();
19.     df['R'] = df['R'].map( v => v/255.0 );
20.     df['G'] = df['G'].map( v => v/255.0 );
21.     df['B'] = df['B'].map( v => v/255.0 );
22.     env.setParam("output", df);
23.   })
24. `;
```

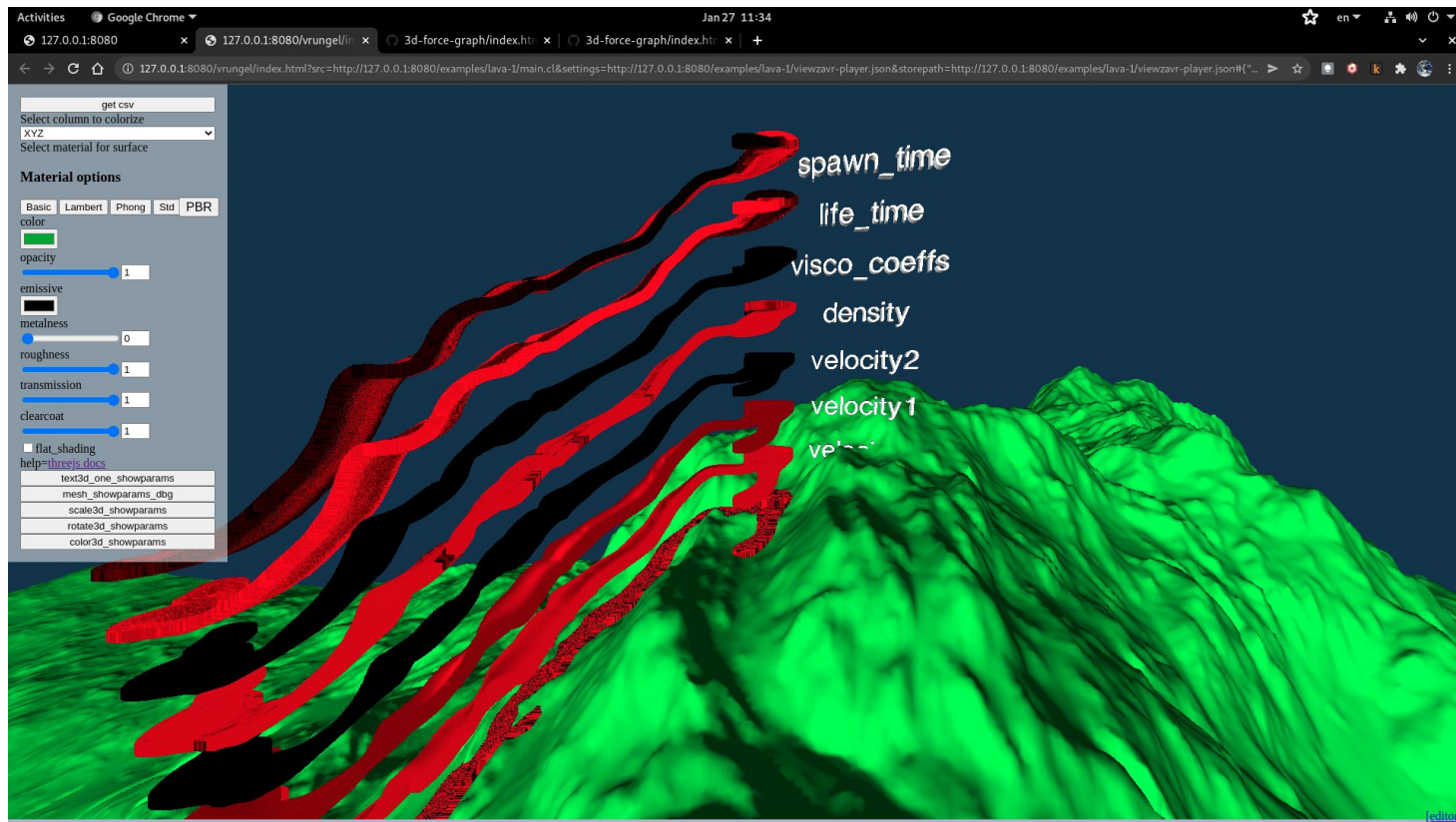
Добавка выбора файла



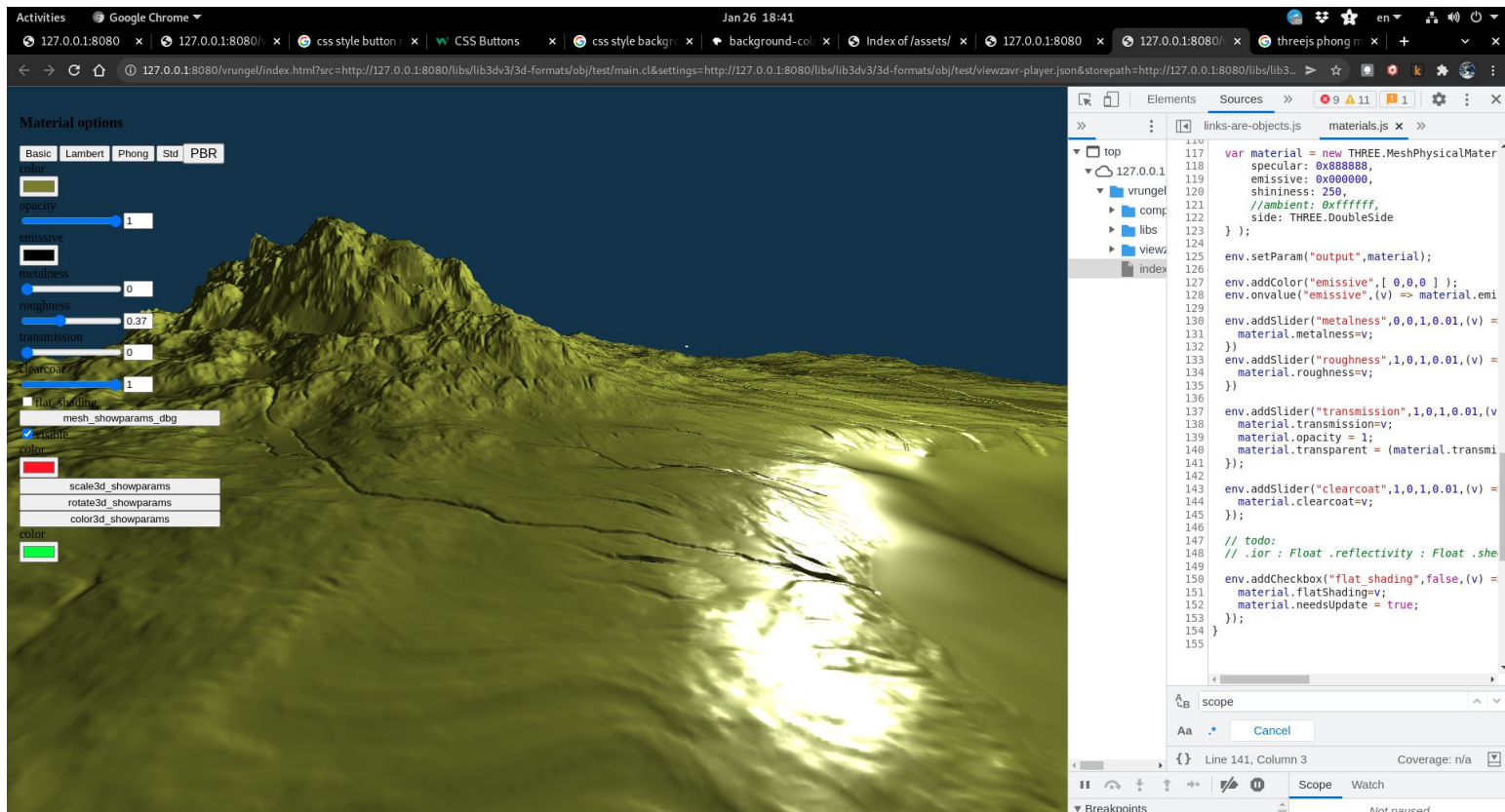
```
1.   load files="lib3d csv";
2.
3.   dat: load-file file=@user_file->value | parse_csv | rescale_rgb
4.
5.   render3d target=@view {
6.     orbit_controls;
7.
8.     @dat | linestrips;
9.   }
10.
11.  screen auto activate {
12.    view: view3d fill parent below others;
13.    column gap="0.5em" padding="1em" {
14.      cb_fn: combobox values=["TSNE_output.csv", "MDS_output.csv"];
15.      user_file: input_file value=@cb_fn->value;
16.    }
17.  }
18.
19.  register feature name="rescale_rgb" code=`
20.    env.onvalue("input", (df) => {
21.      if (!df || !df.isDataFrame || !df['R']) return;
22.      df = df.clone();
23.      df['R'] = df['R'].map( v => v/255.0 );
24.      df['G'] = df['G'].map( v => v/255.0 );
25.      df['B'] = df['B'].map( v => v/255.0 );
26.      env.setParam("output", df);
27.    })
28.  `;
```


Илья Сергеевич Стародубцев

Визуализация расчетов по лаве

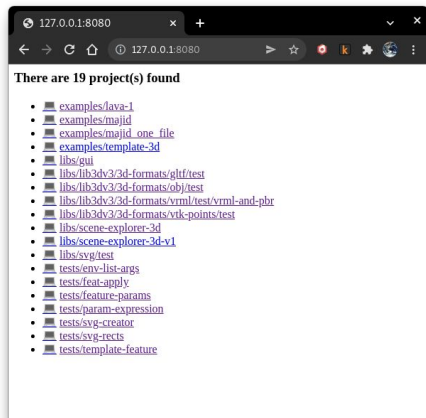


Особенность: отладка в браузере





Использование

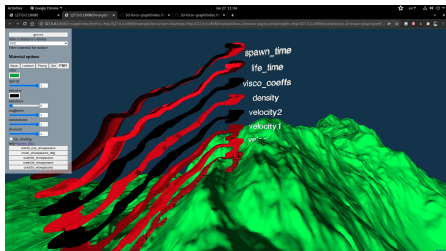


1. Установить NodeJS
2. Для запуска интерпретатора выполнить команду:

npx vrungel

Также возможен запуск из Windows Explorer.

Сайт проекта: <https://github.com/viewzavr/vrungel>



Благодарим за внимание!

Михаил Бахтерев, Павел Васёв, Дмитрий Манаков

Сектор компьютерной визуализации
ИММ УРО РАН им. Н.Н. Красовского

<https://github.com/viewzavr/vrungel>

