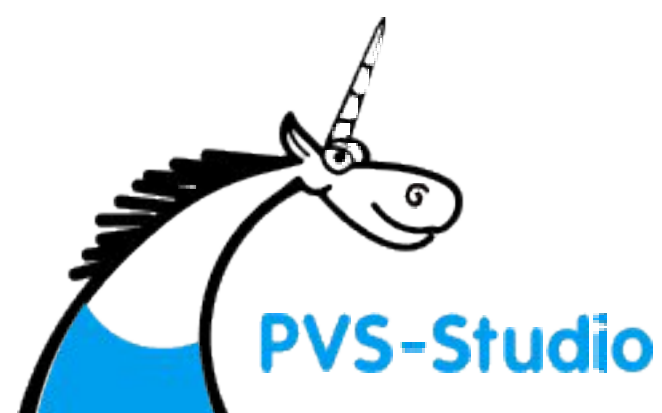


# Превратили PVS-Studio в город

Давайте вместе прогуляемся по нему и узнаем,  
какие тайны он в себе хранит :)



**Александра Уварова**  
Developer Advocate



# Александра Уварова

Developer Advocate

- Разработчик C++ части анализатора PVS-Studio.
- Рассказываю про качество кода и безопасную разработку на конференциях
- Пишу технические и научные статьи



@AleksandraUvarova

PVS-Studio

# Начнём с начала

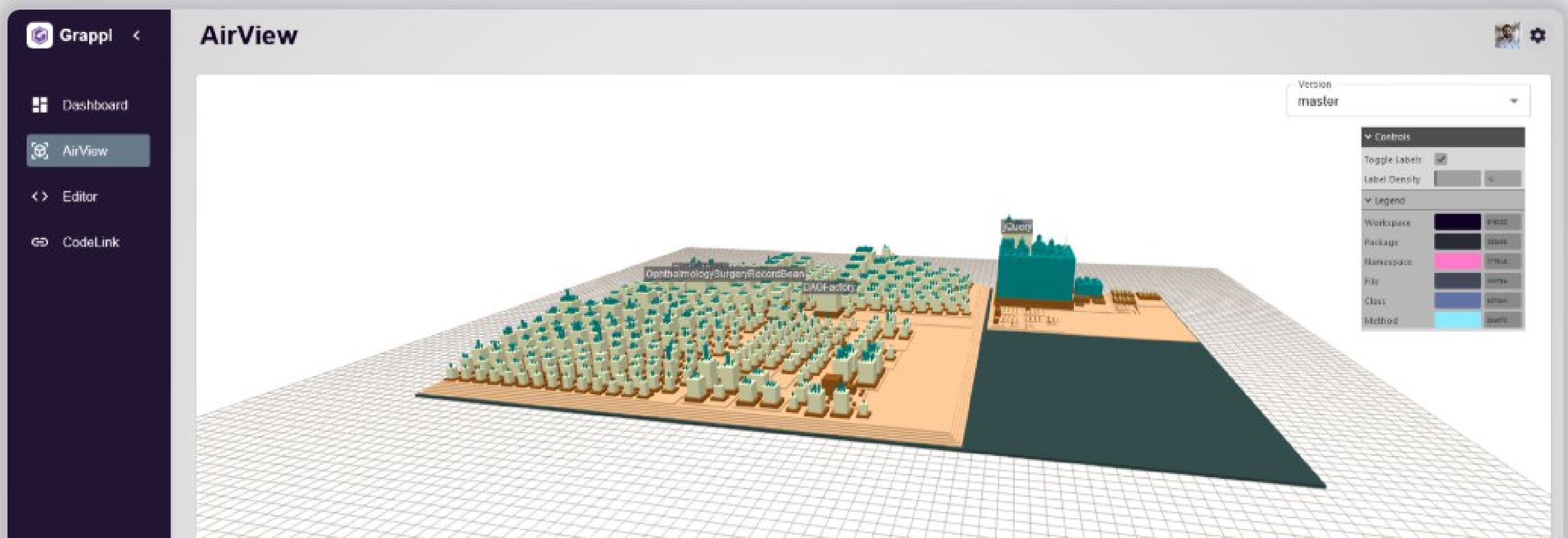


# Визуальная карта вашей кодовой базы

## Say hello to Grappl

Gen AI has transformed how teams write code; Grappl transforms how your team manages it.

Grappl is built for the next generation of software creators and decision makers. As AI generates vast amounts of code, the real challenge teams are facing now is maintaining visibility, control, and taking timely decisions.





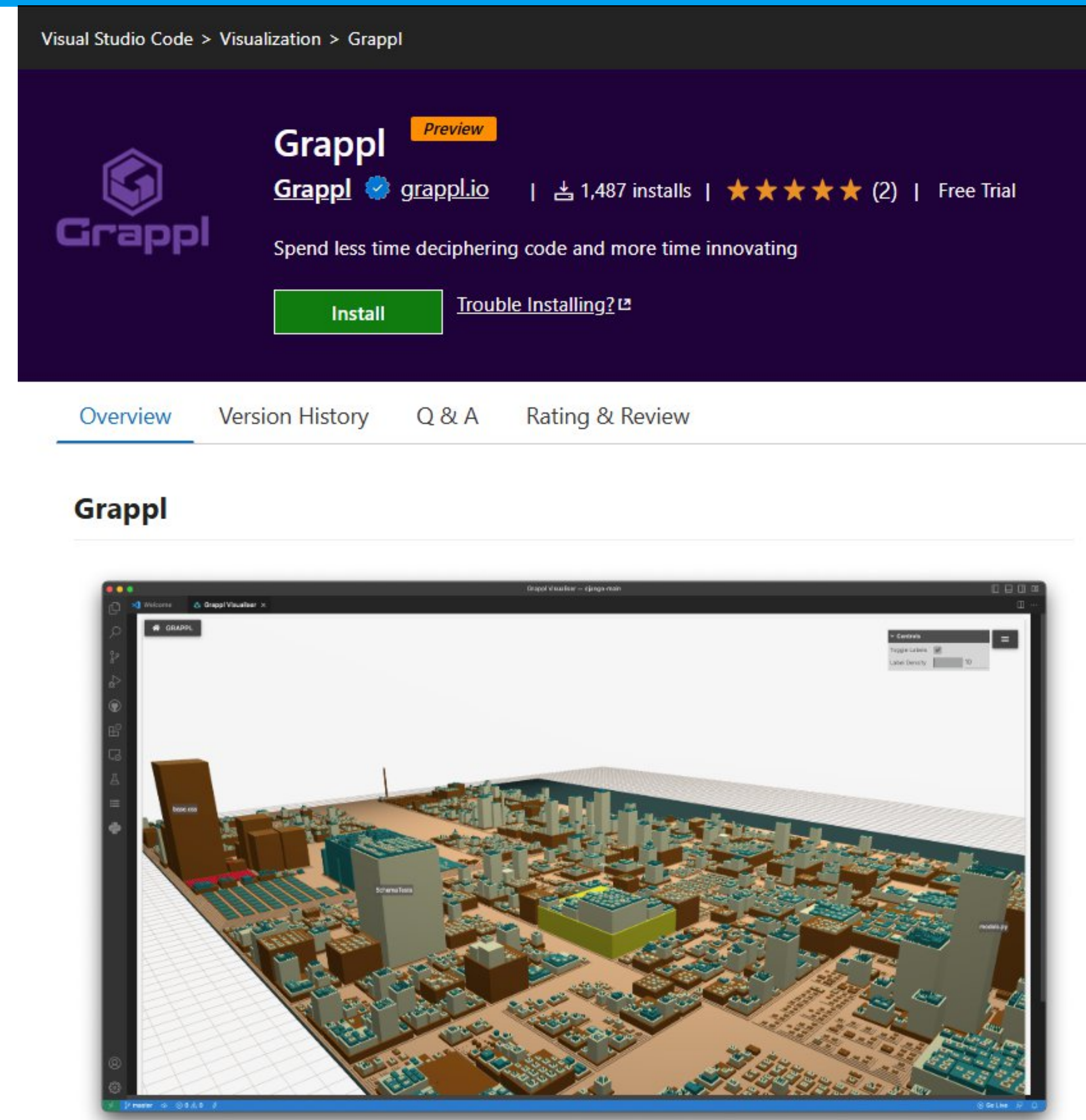
# Расширение Grappl

5

Бесплатно для Visual Studio Code

**Это решение для:**

- Улучшения взаимодействия между техническими и нетехническими членами команды
- Автоматическое связывание между собой кода, задач из трекеров и коммитов



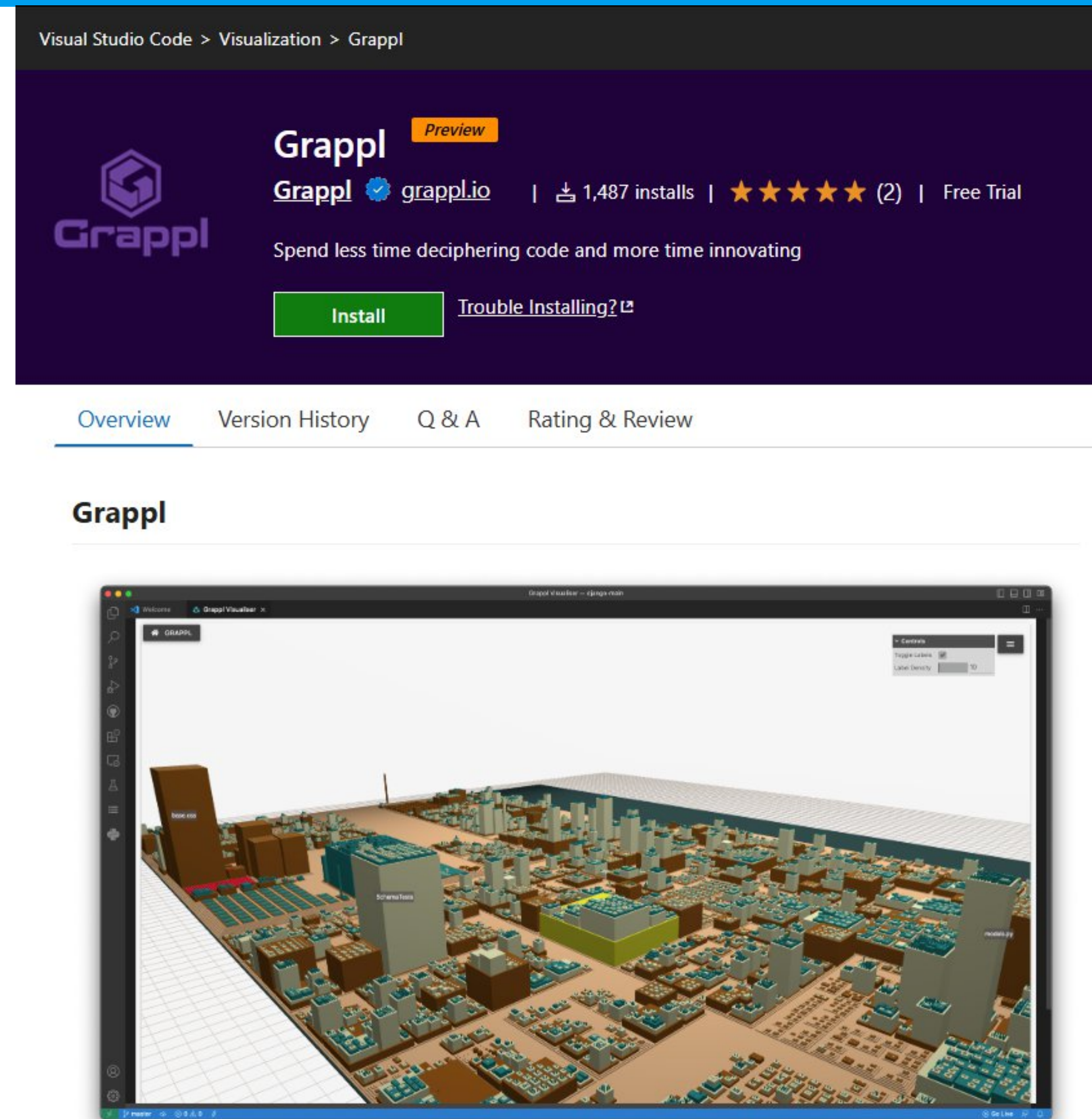


# Расширение Grappl

6



<https://marketplace.visualstudio.com/items?itemName=grappl.grappl>



1. Попробуем инструмент на разных проектах

2. Построим город из PVS-Studio

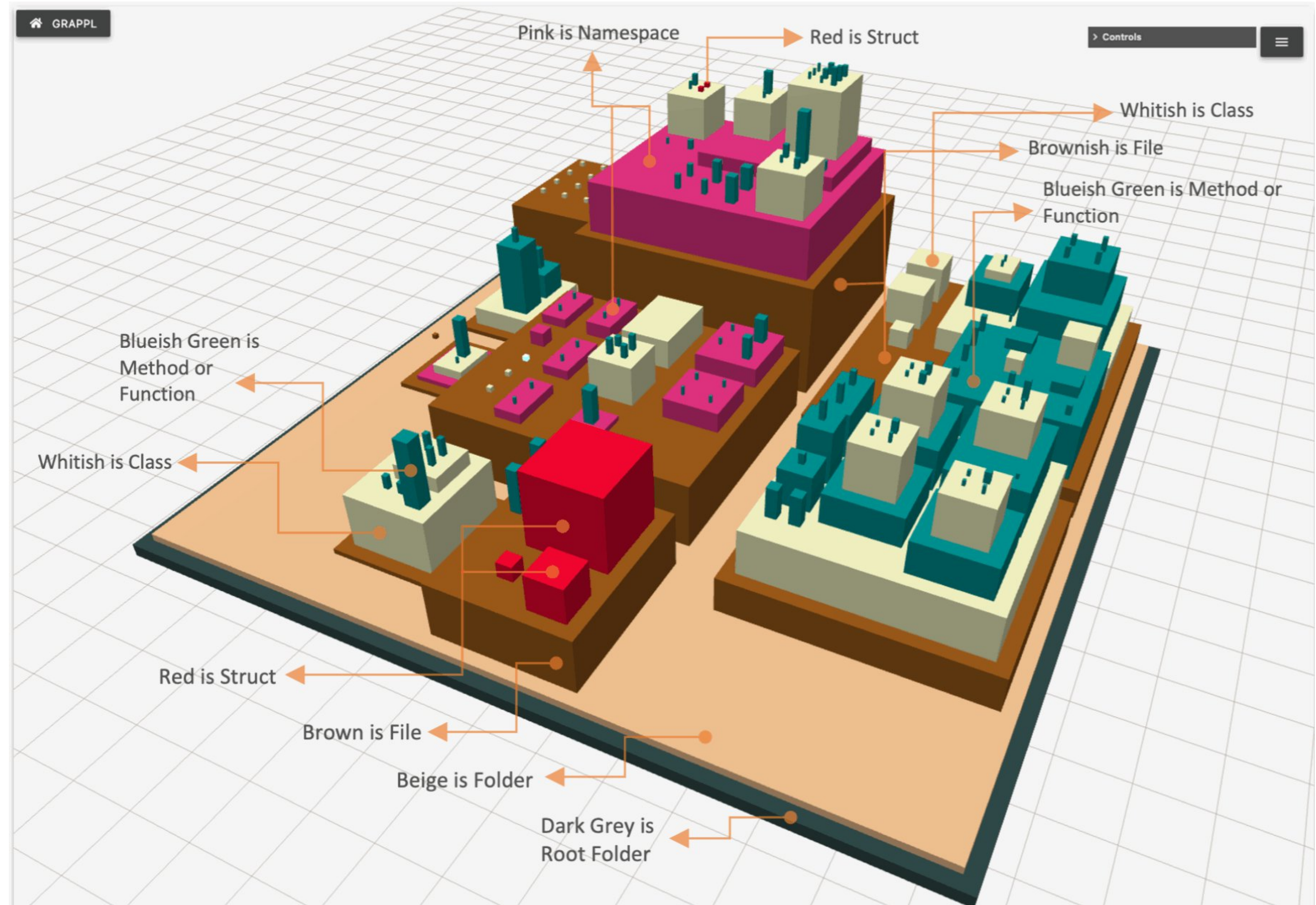
3. Прогуляемся по городу  
и изучим процесс анализа кода



# Условные обозначения

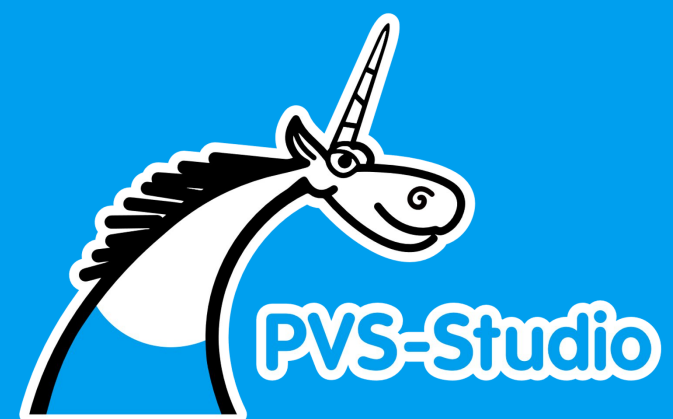
8

- Root Folder
- Folder
- File
- Namespace
- Struct
- Class
- Method or Function





# Кейсы





- Мы часто проверяем разные проекты с помощью нашего анализатора
- Рассмотрим некоторые из них

Проекты, в которых мы нашли ошибки  
с помощью PVS-Studio



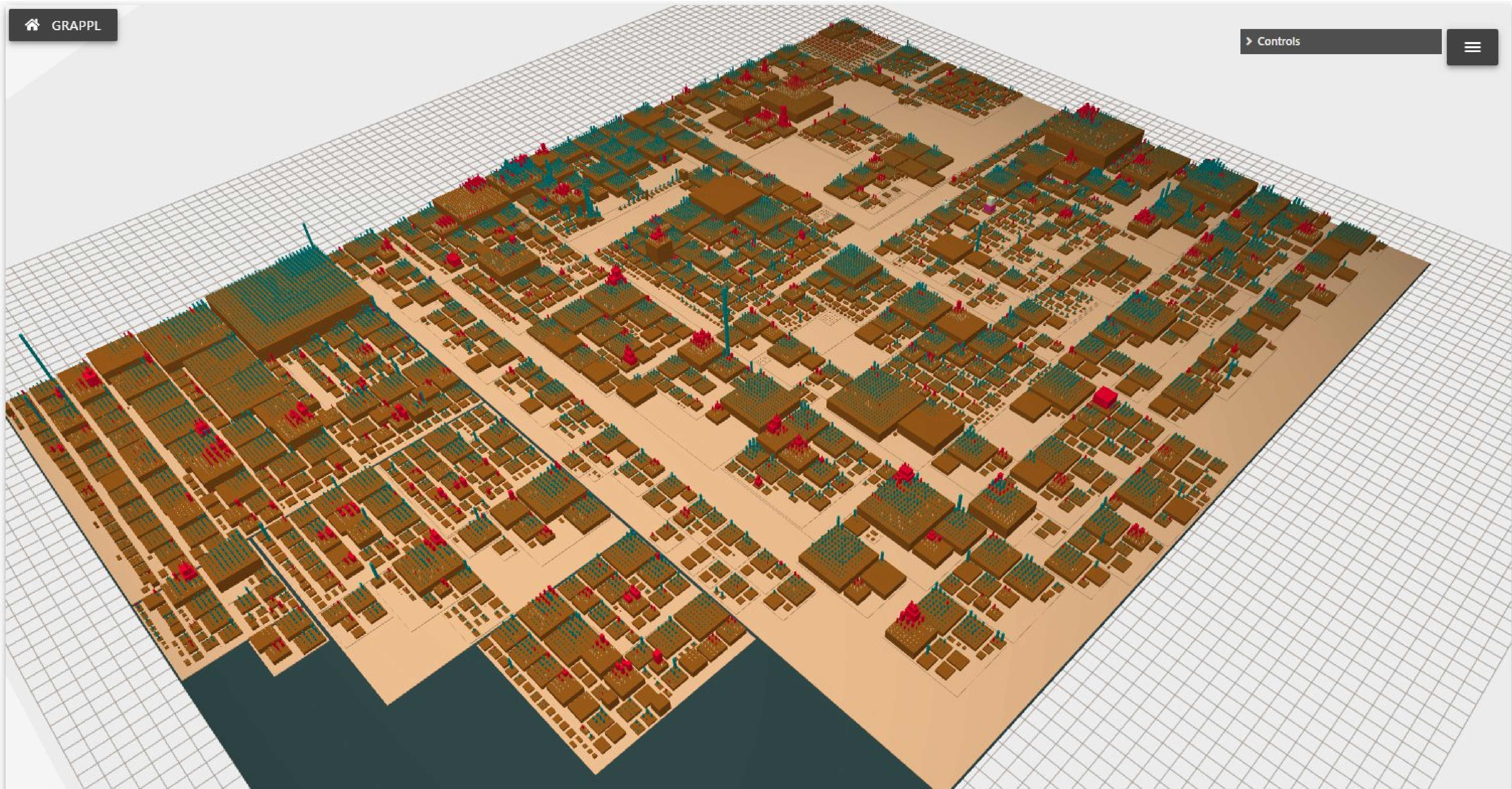
<https://pvs-studio.ru/ru/blog/inspections/>

Что делать, если ваш слон думает,  
что он баг?

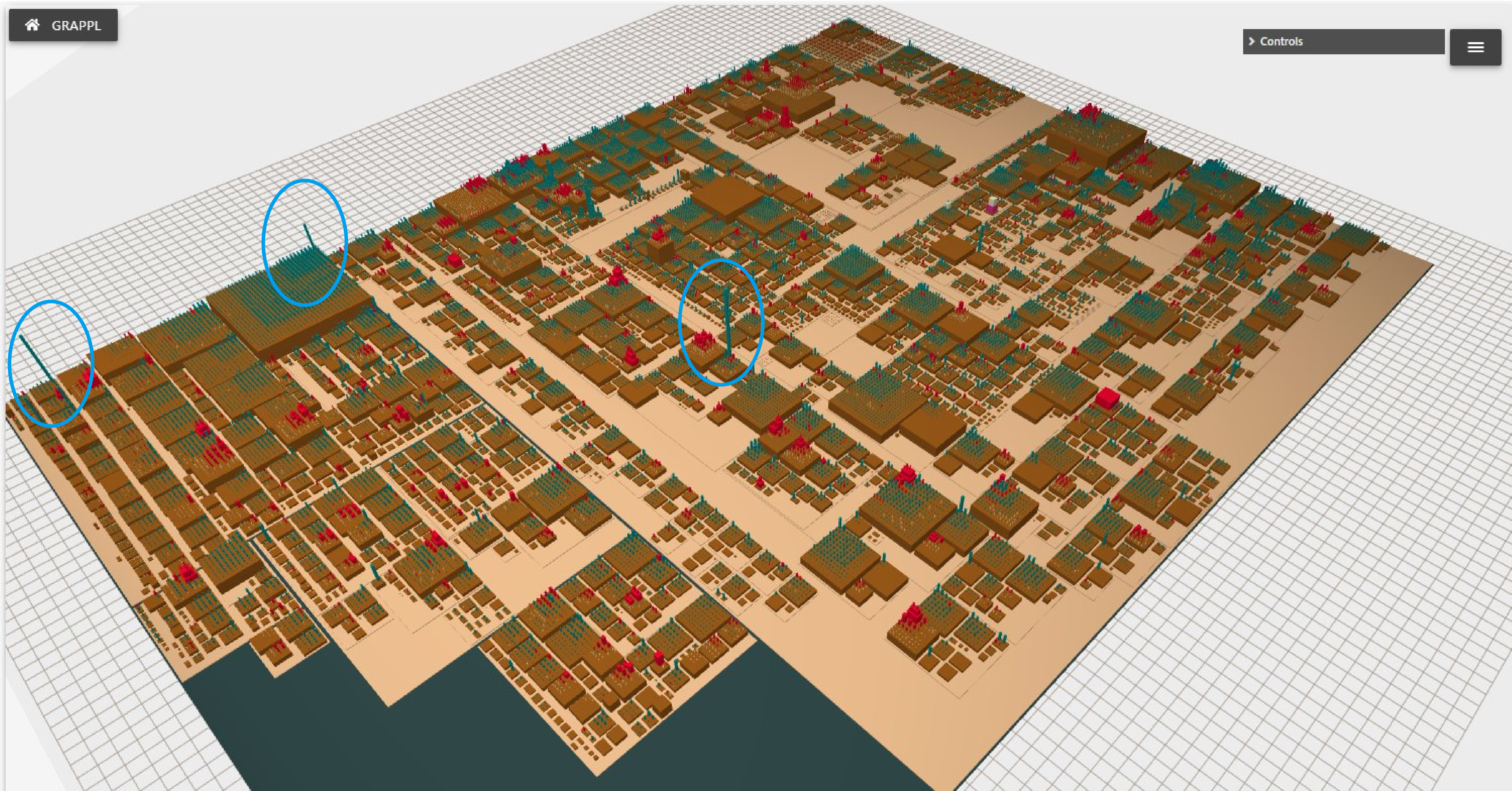


<https://pvs-studio.ru/ru/blog/posts/cpp/1280/>

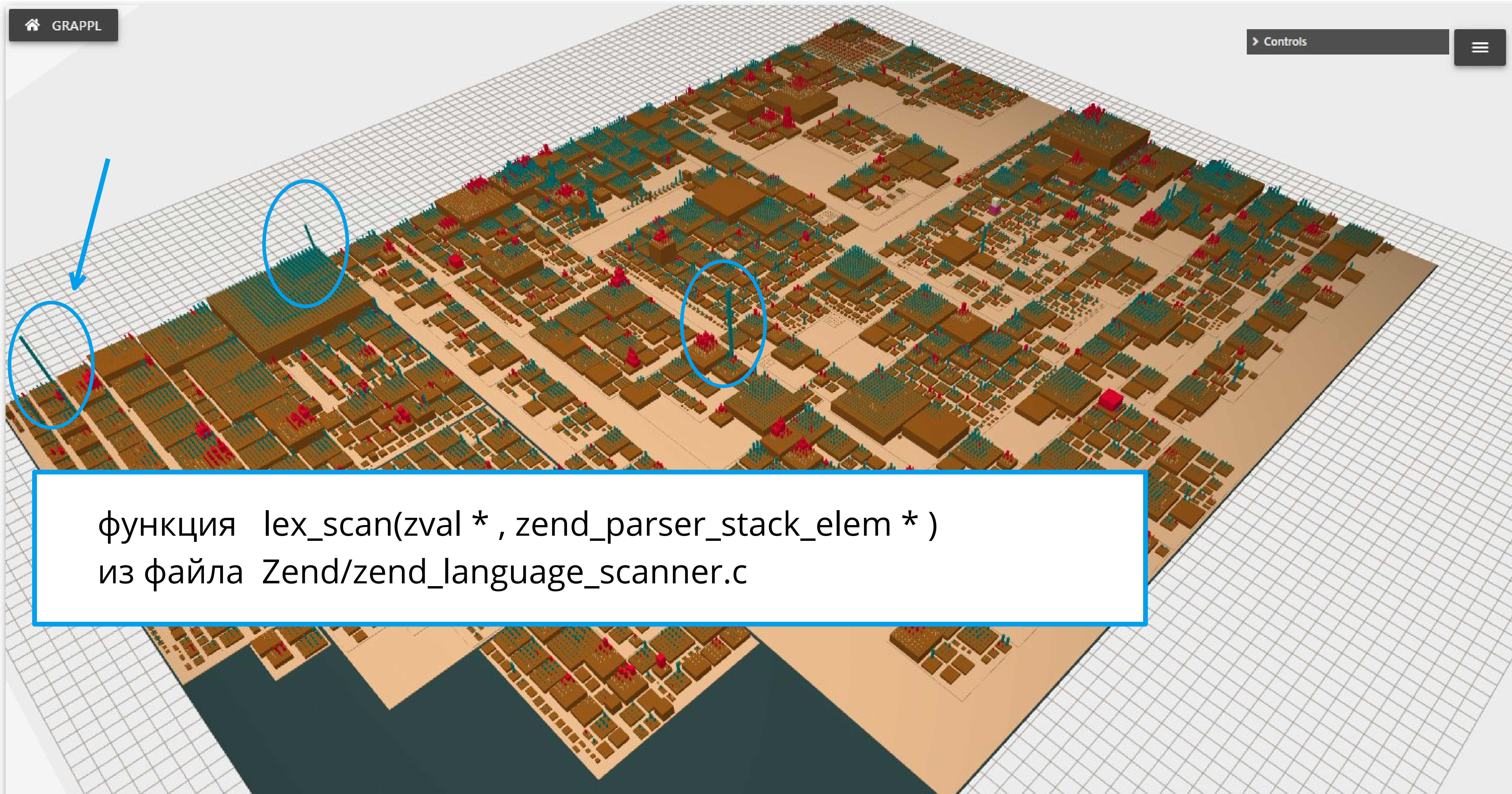
- PHP — один из популярных сценарных языков в области веб-программирования серверной части.
- Languages:  
C 69.9%    **PHP 27.7%**    C++ 0.9%











функция `lex_scan(zval *, zend_parser_stack_elem *)`  
из файла `Zend/zend_language_scanner.c`



Игровое поле экспериментов: какие ошибки  
могут подстергать программиста при  
создании эмулятора



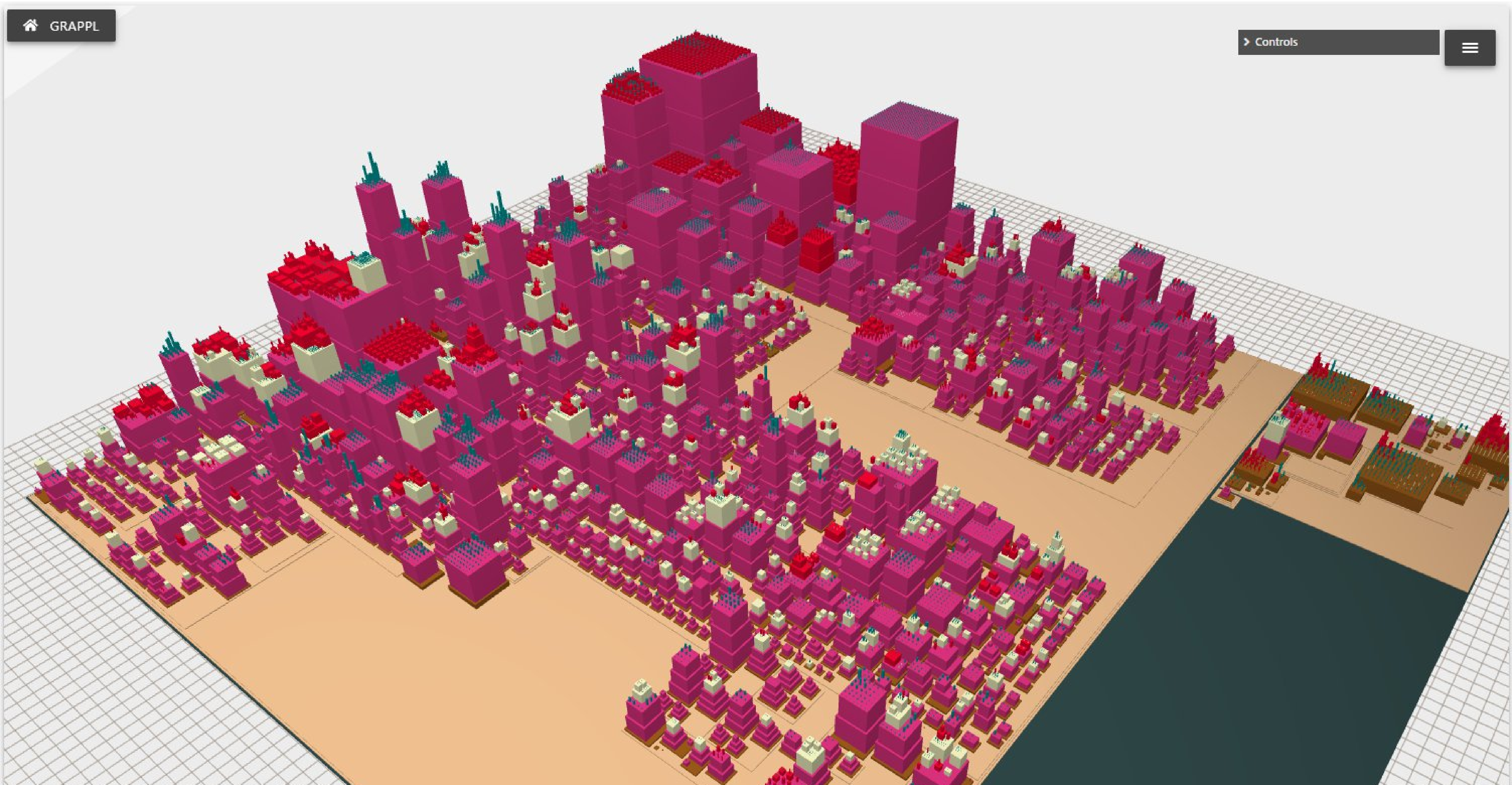
<https://pvs-studio.ru/ru/blog/posts/cpp/1177/>

- Xenia — экспериментальный эмулятор платформы Xbox 360.
- Languages:
  - C++ 94.6%**
  - Assembly 3.2%**



GRAPPL

&gt; Controls

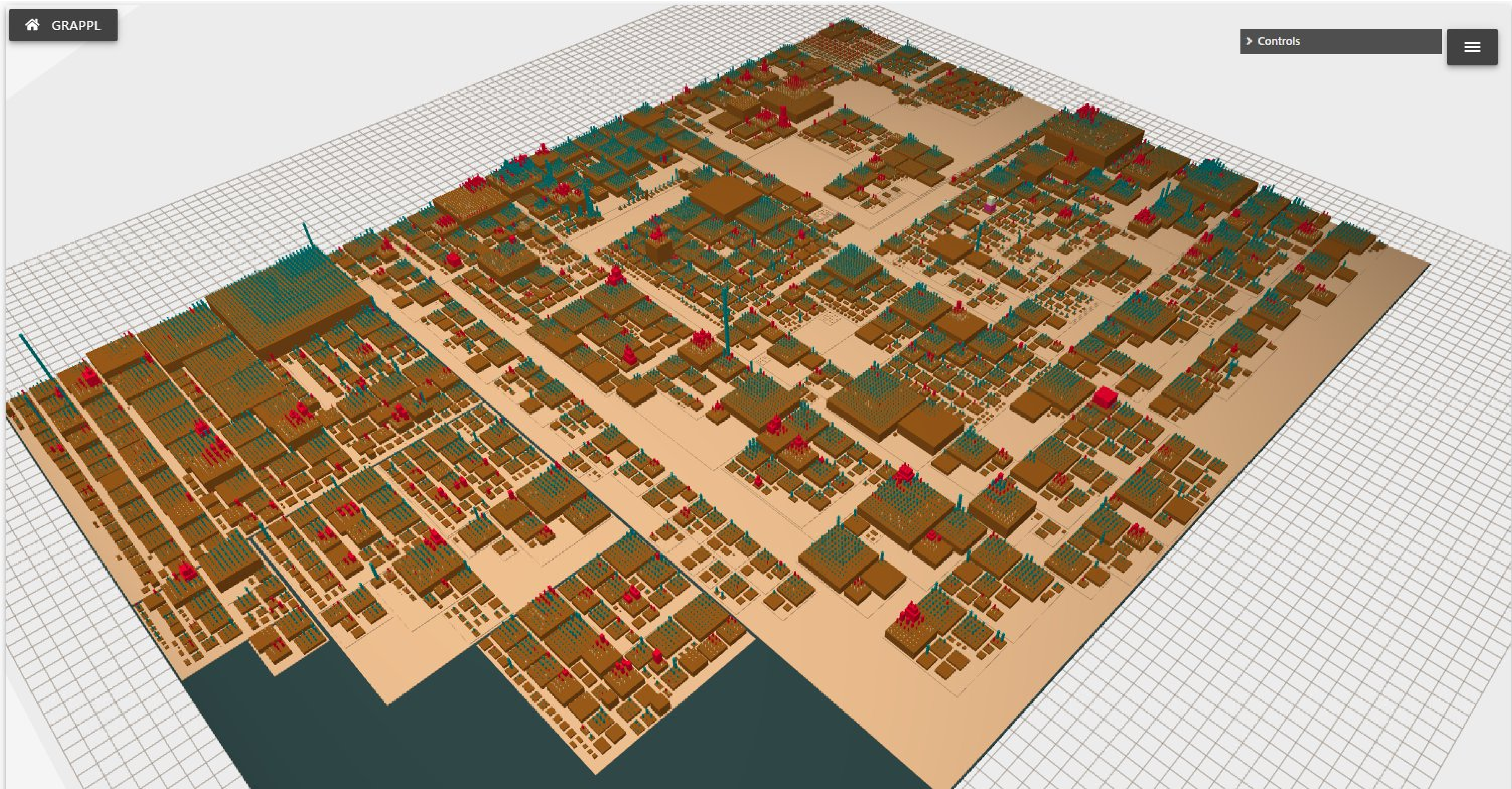




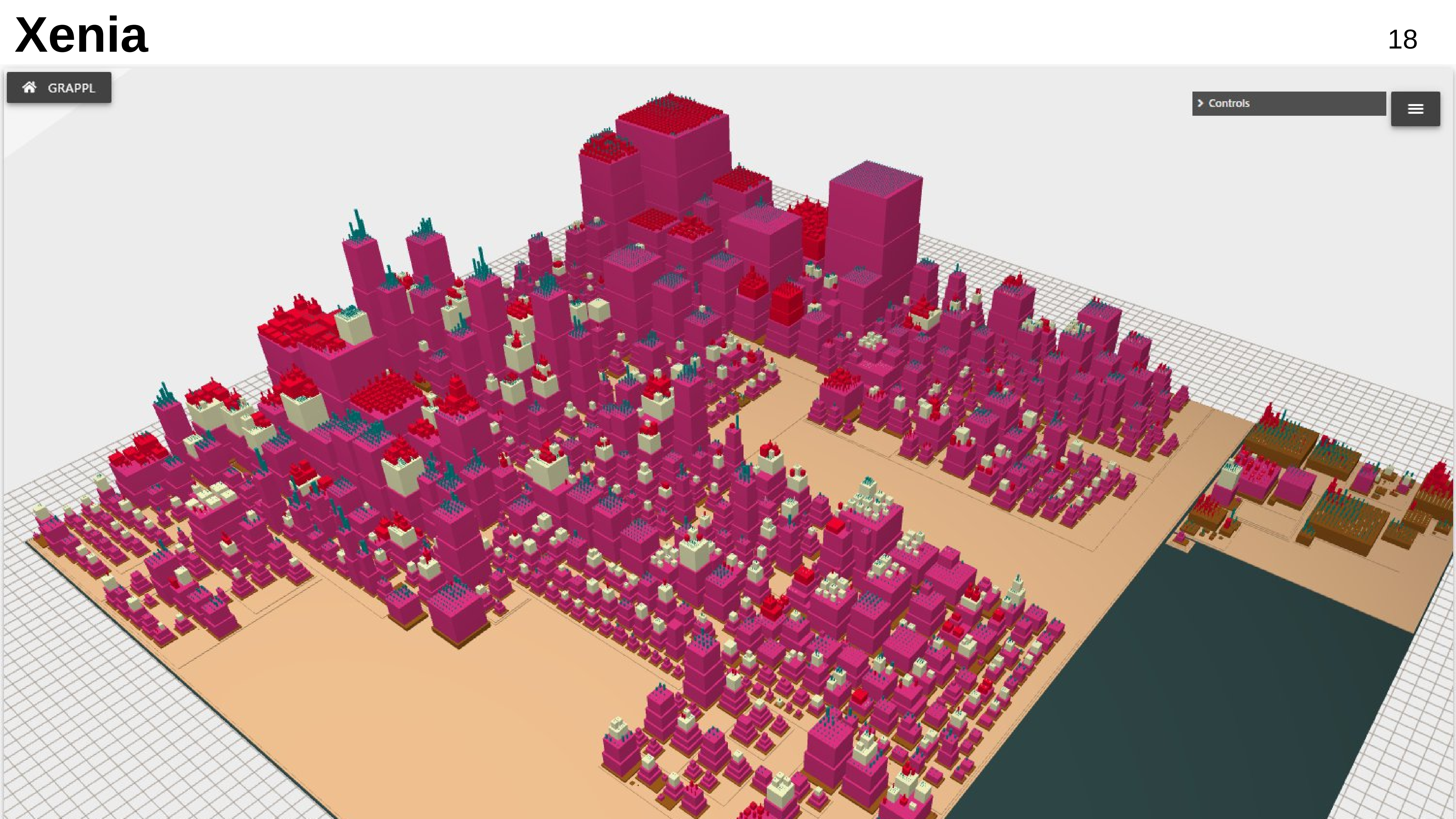


GRAPPL

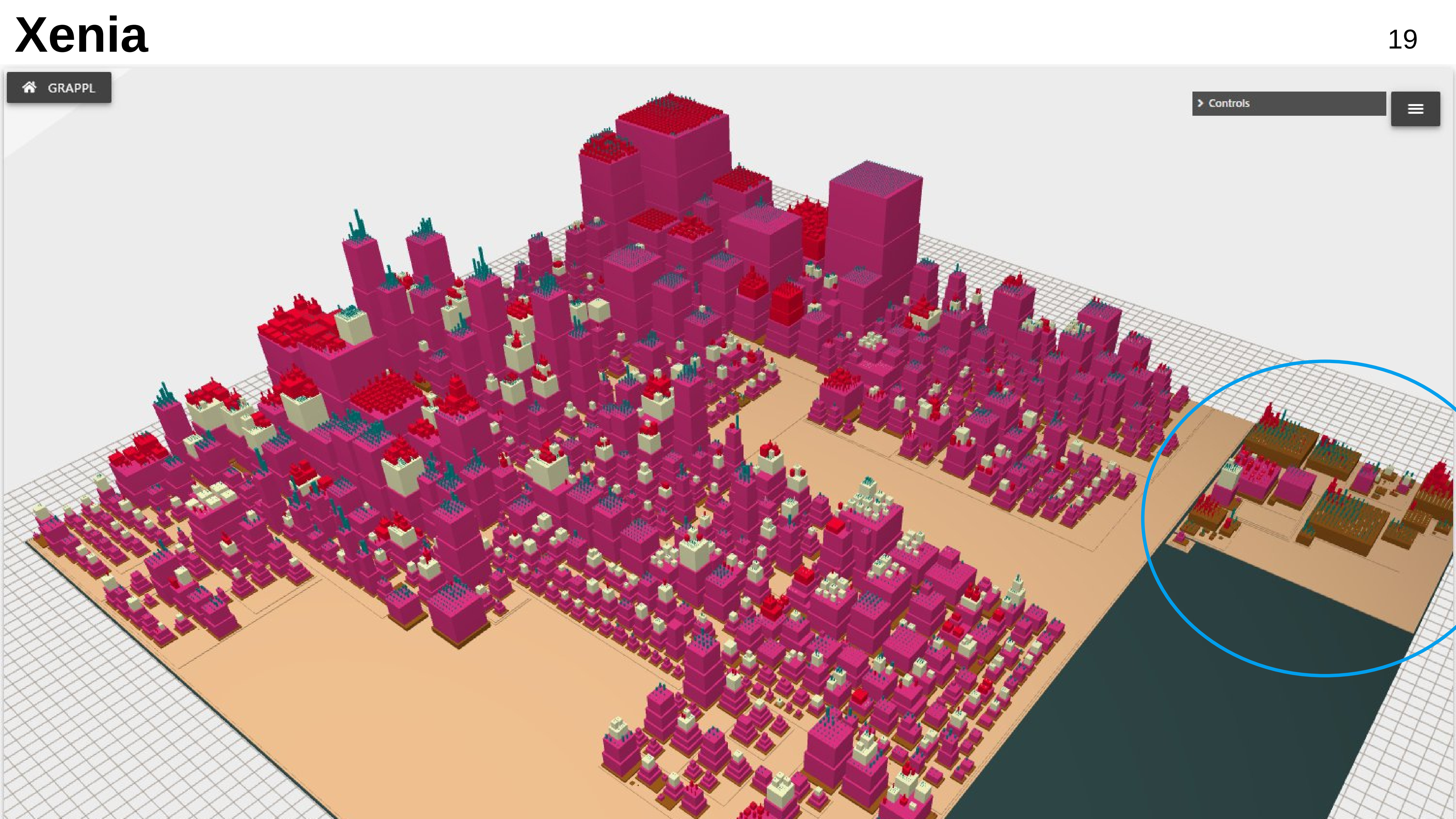
&gt; Controls













Ну и какой же город ЭТОТ ваш  
PVS-Studio?



PVS-Studio



<https://pvs-studio.ru/ru/>

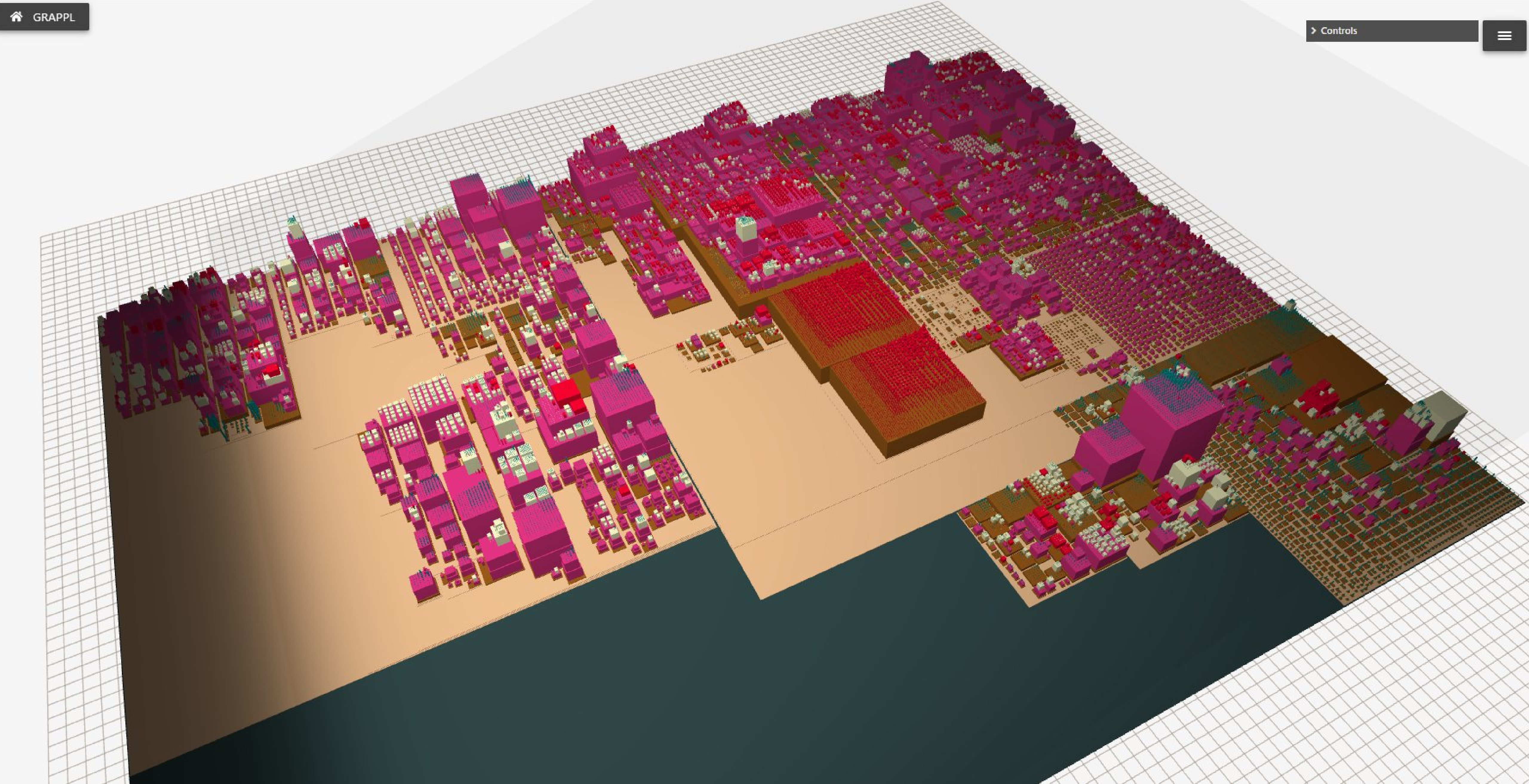
- PVS-Studio — статический анализатор кода, который развивается на рынке уже более 15 лет.
- Languages: C и C++, C#, Java

PVS-Studio

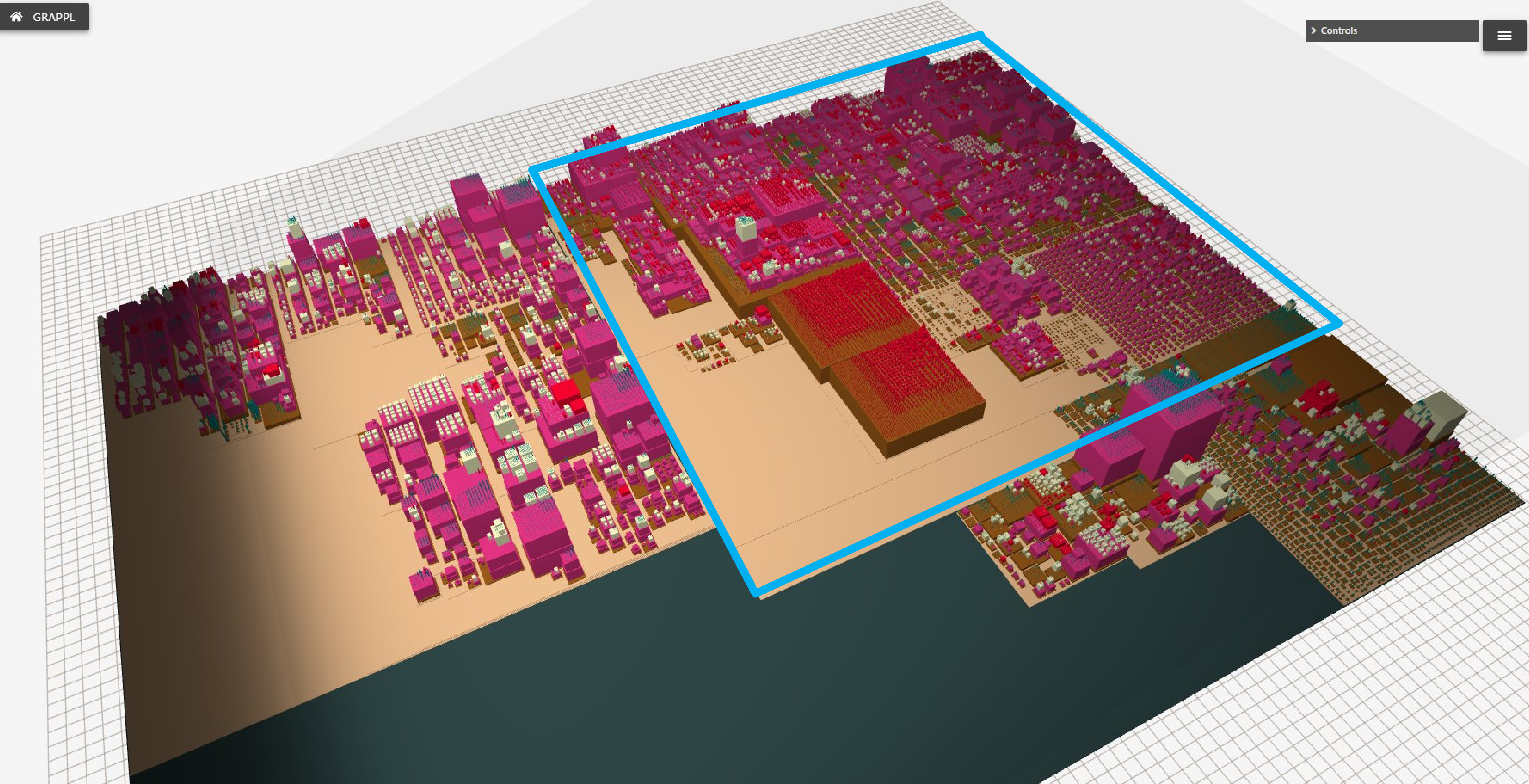


<https://pvs-studio.ru/ru/>

- PVS-Studio — статический анализатор кода, который развивается на рынке уже более 17 лет.
- Languages: C и C++, C#, Java



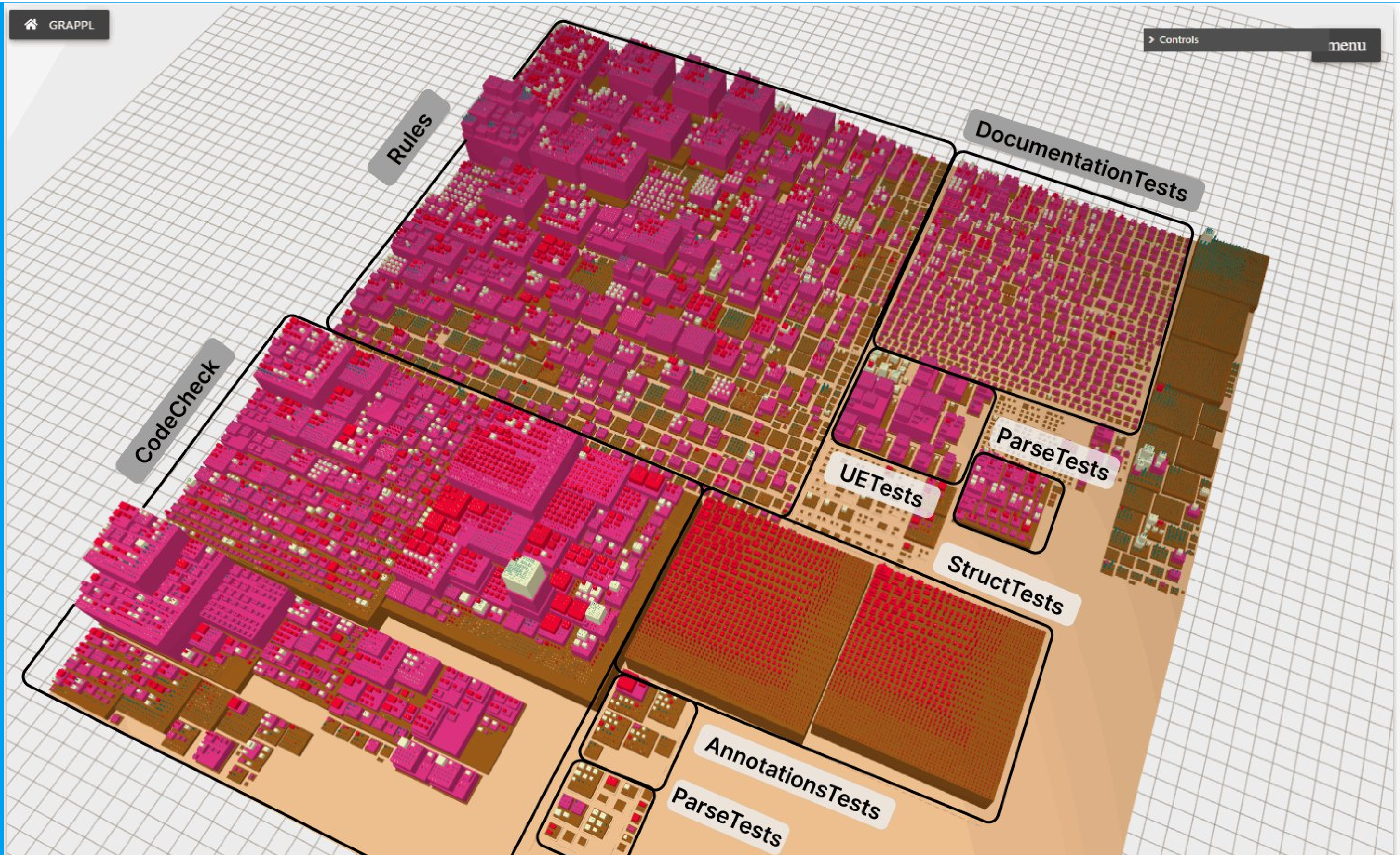




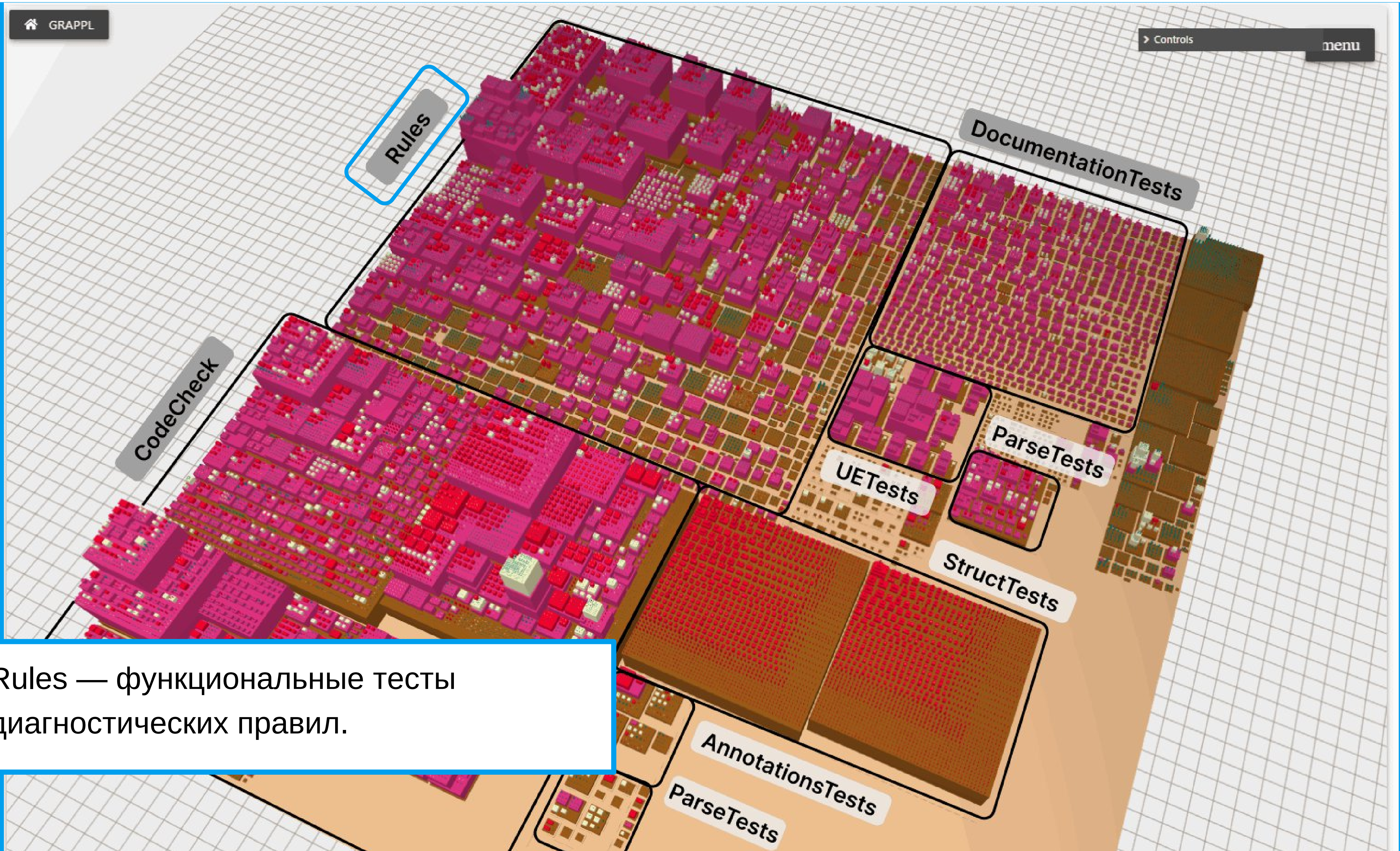


# Обширный жилой квартал: модули тестирования

25

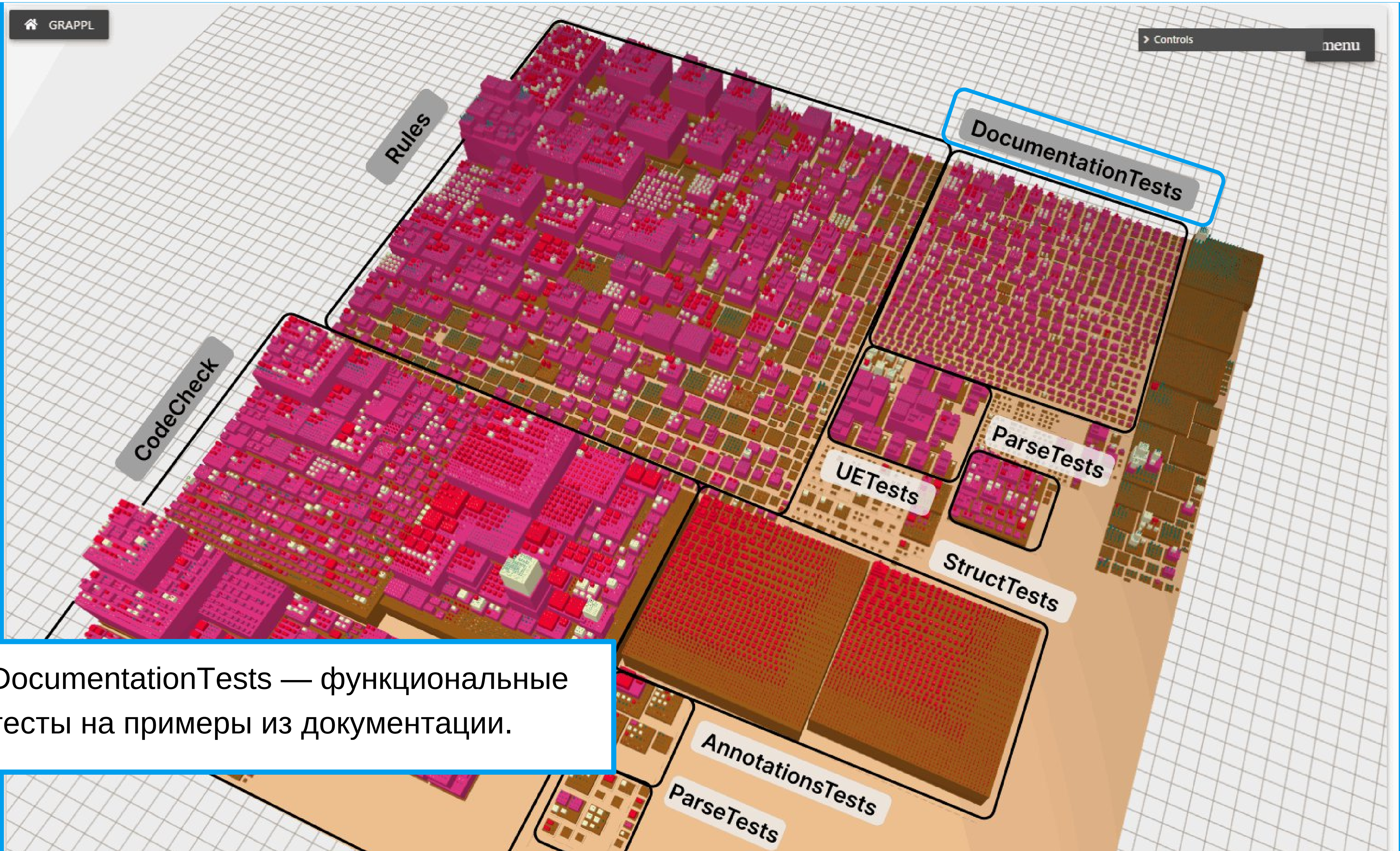






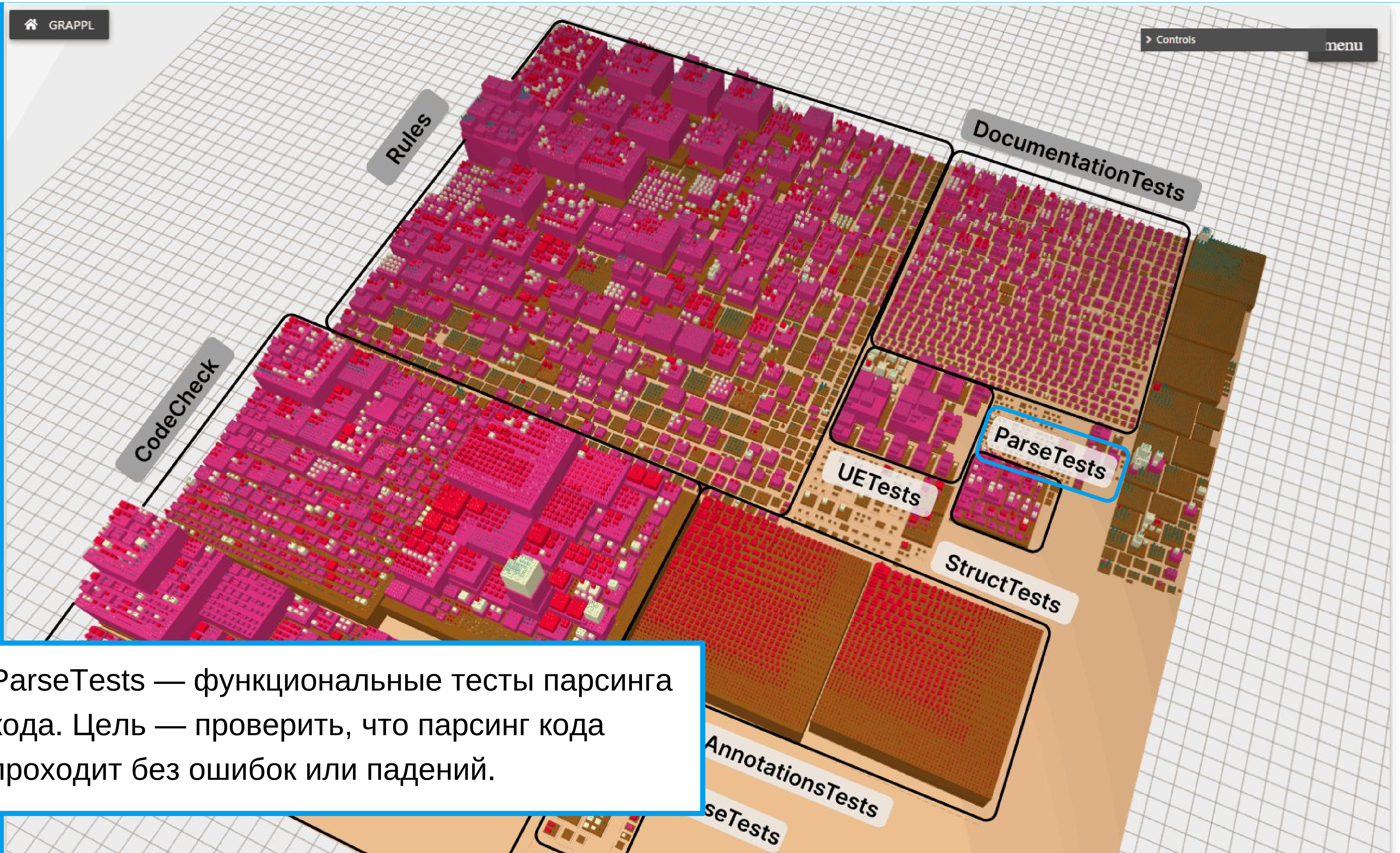
Rules — функциональные тесты  
диагностических правил.





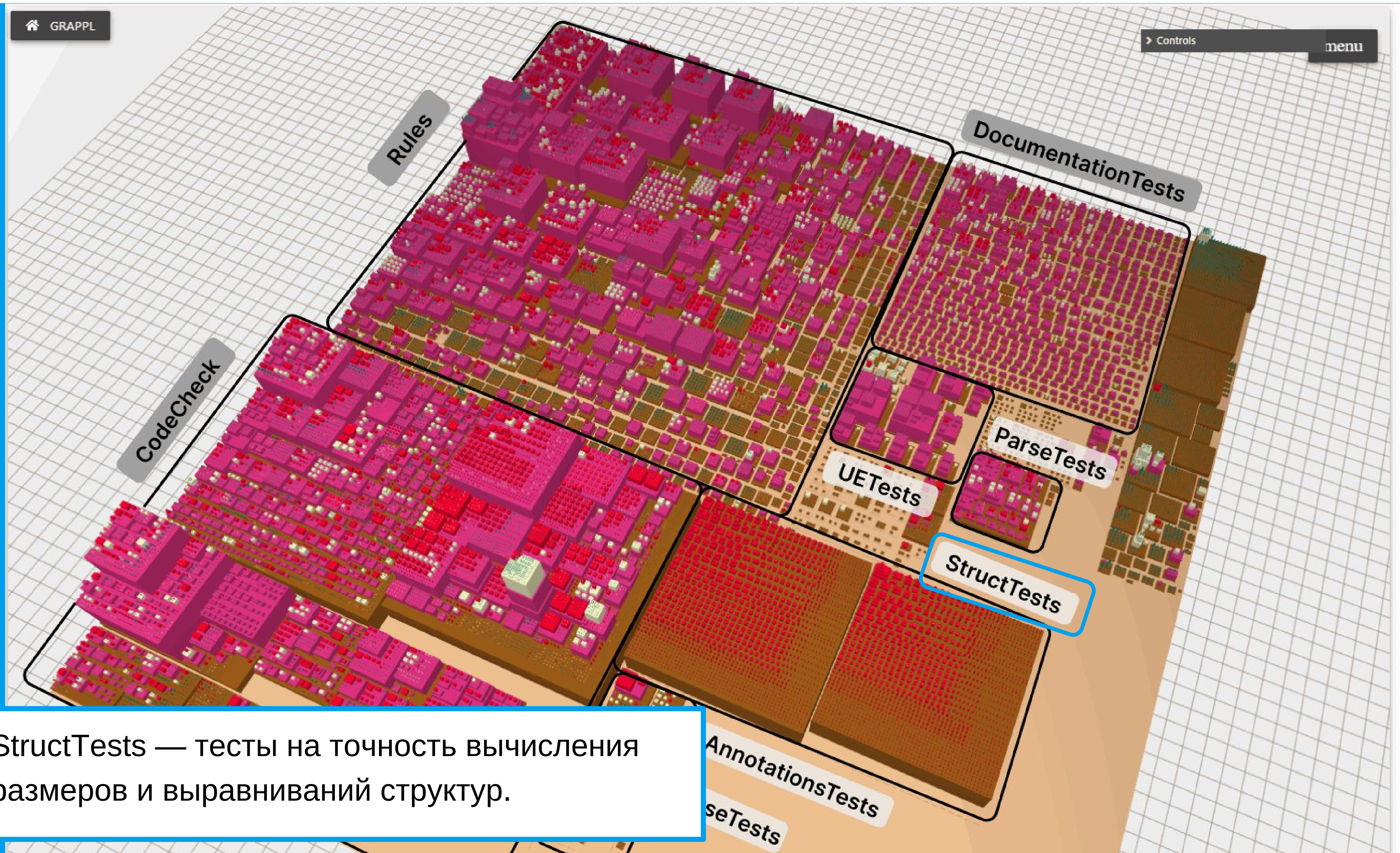
DocumentationTests — функциональные тесты на примеры из документации.





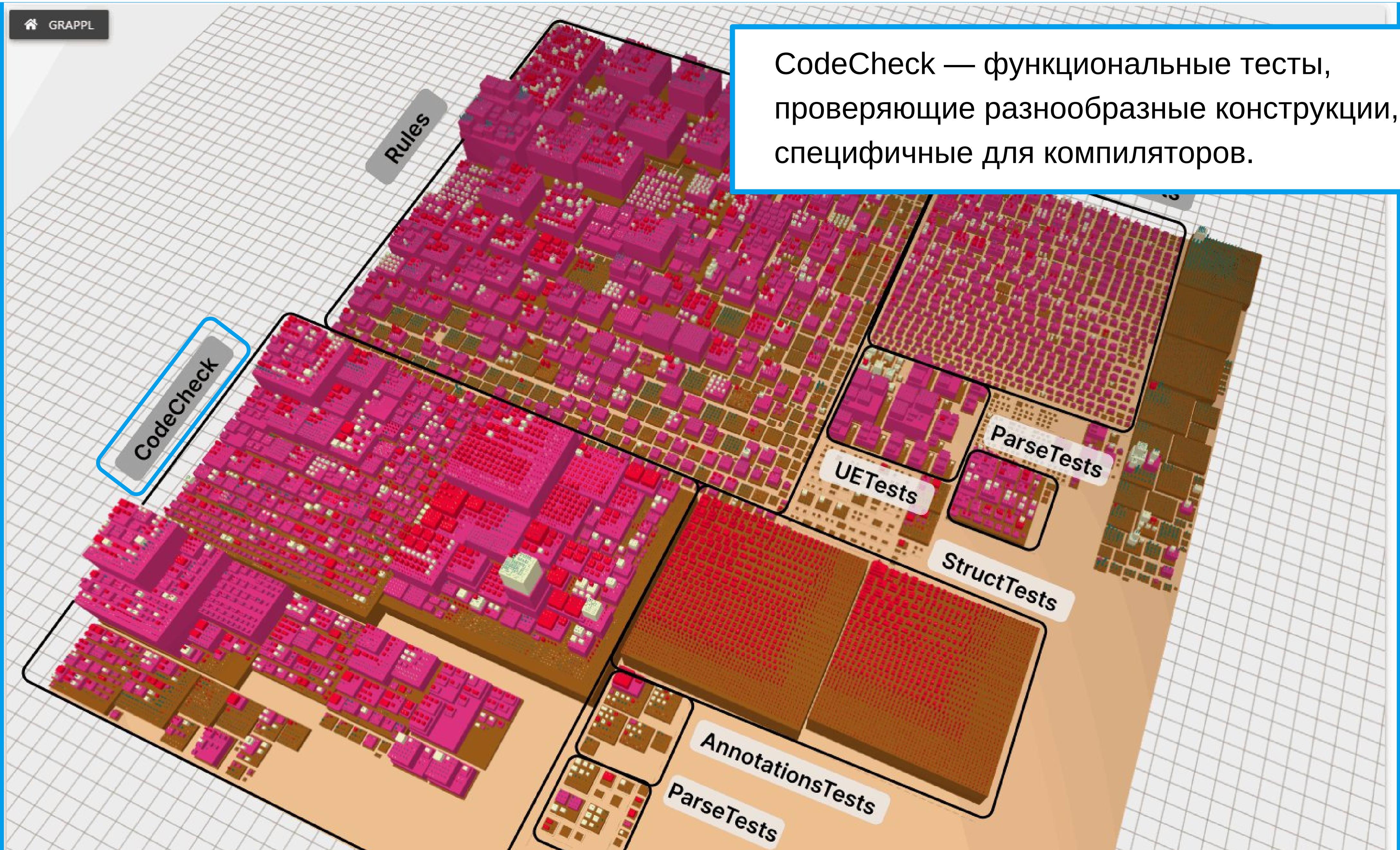
ParseTests — функциональные тесты парсинга кода. Цель — проверить, что парсинг кода проходит без ошибок или падений.





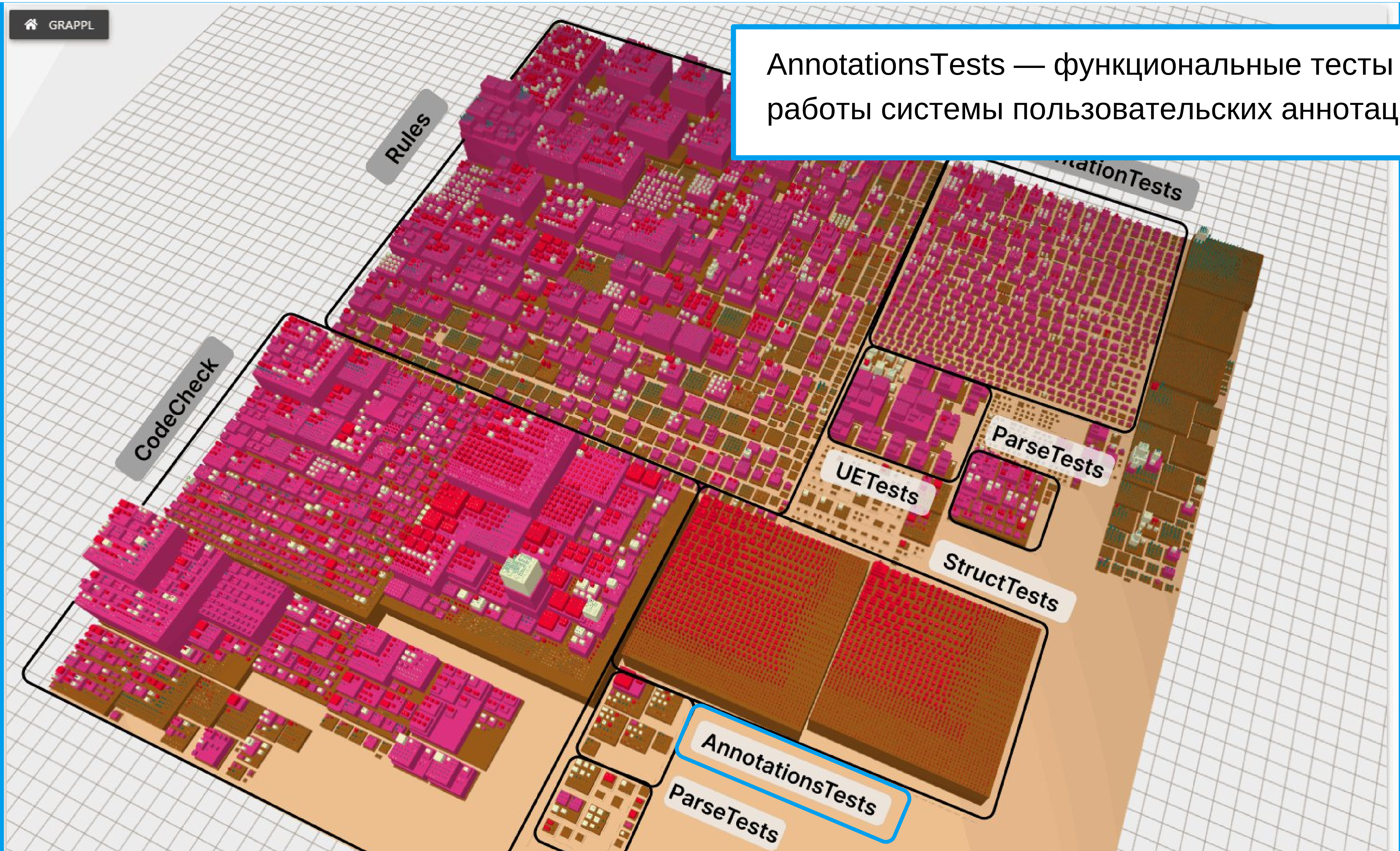
StructTests — тесты на точность вычисления размеров и выравниваний структур.





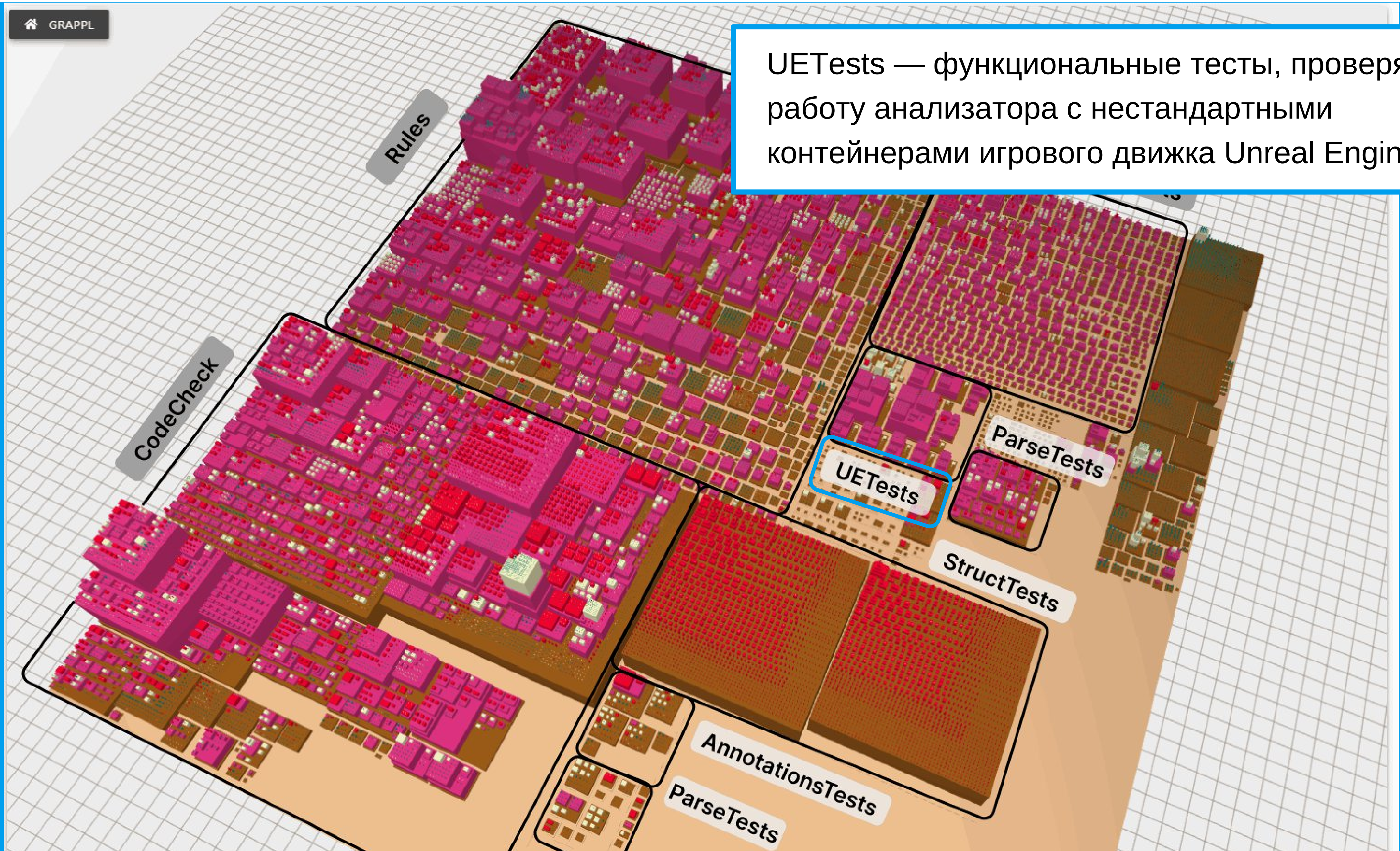
CodeCheck — функциональные тесты, проверяющие разнообразные конструкции, специфичные для компиляторов.





AnnotationsTests — функциональные тесты работы системы пользовательских аннотаций





UETests — функциональные тесты, проверяющие работу анализатора с нестандартными контейнерами игрового движка Unreal Engine

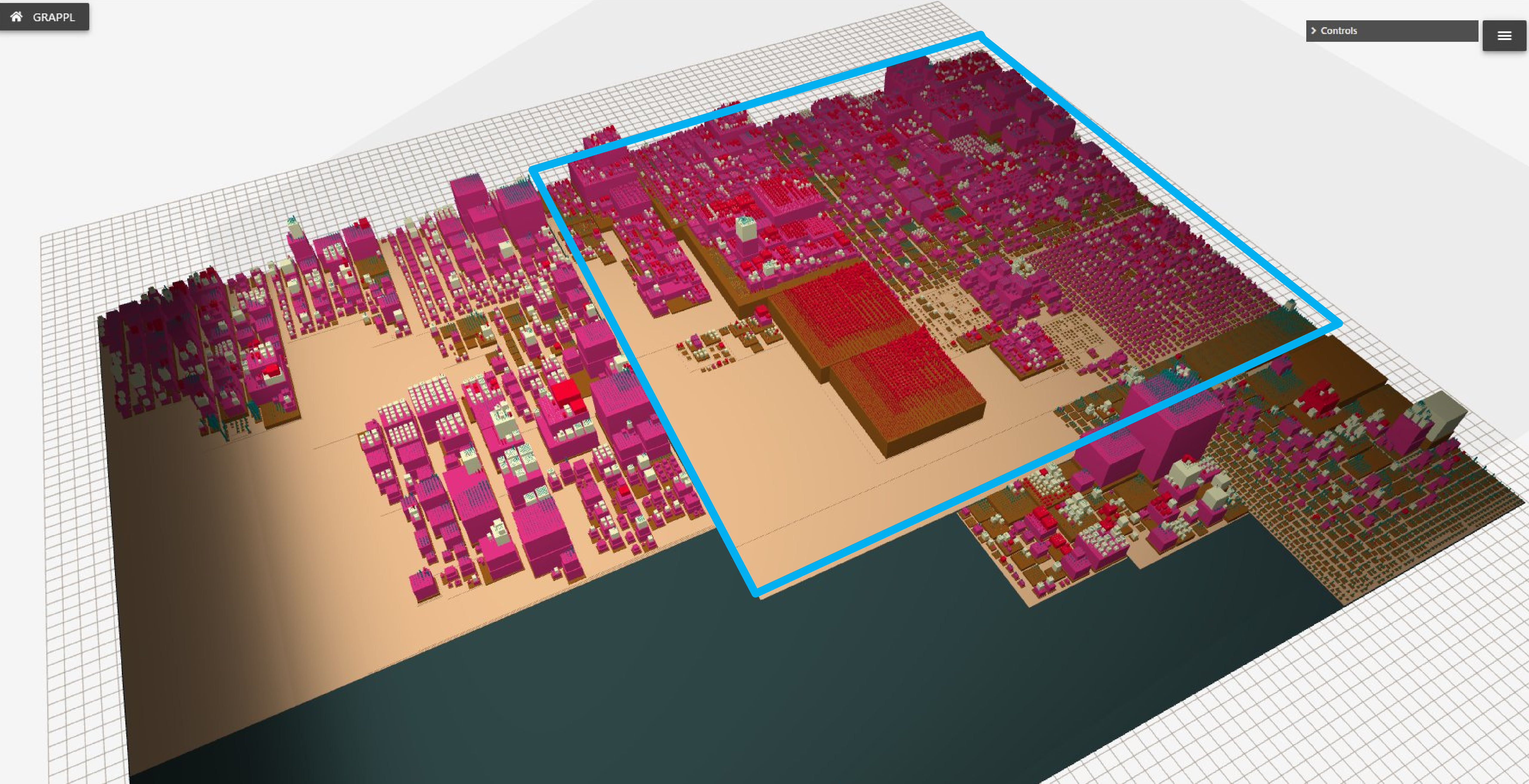


- В PVS-Studio есть **возможность анализировать Unreal Engine проекты.**
- Распределённая сборка и анализ через Horde + UBA
- Диагностические правила, специфичные для проектов на основе Unreal Engine

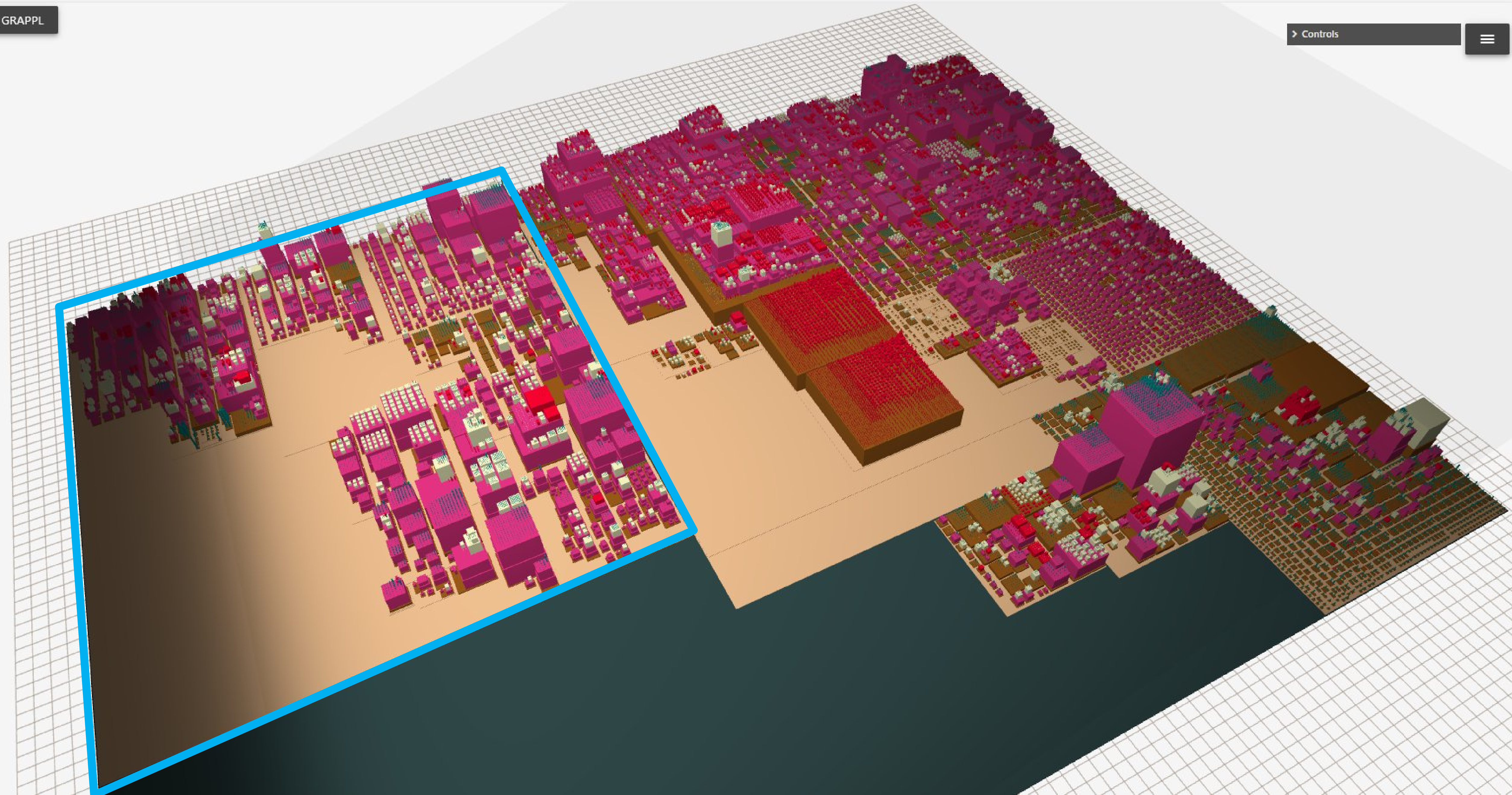
Как статический анализ помогает проектам на Unreal Engine



<https://pvs-studio.ru/ru/blog/posts/cpp/1273/>







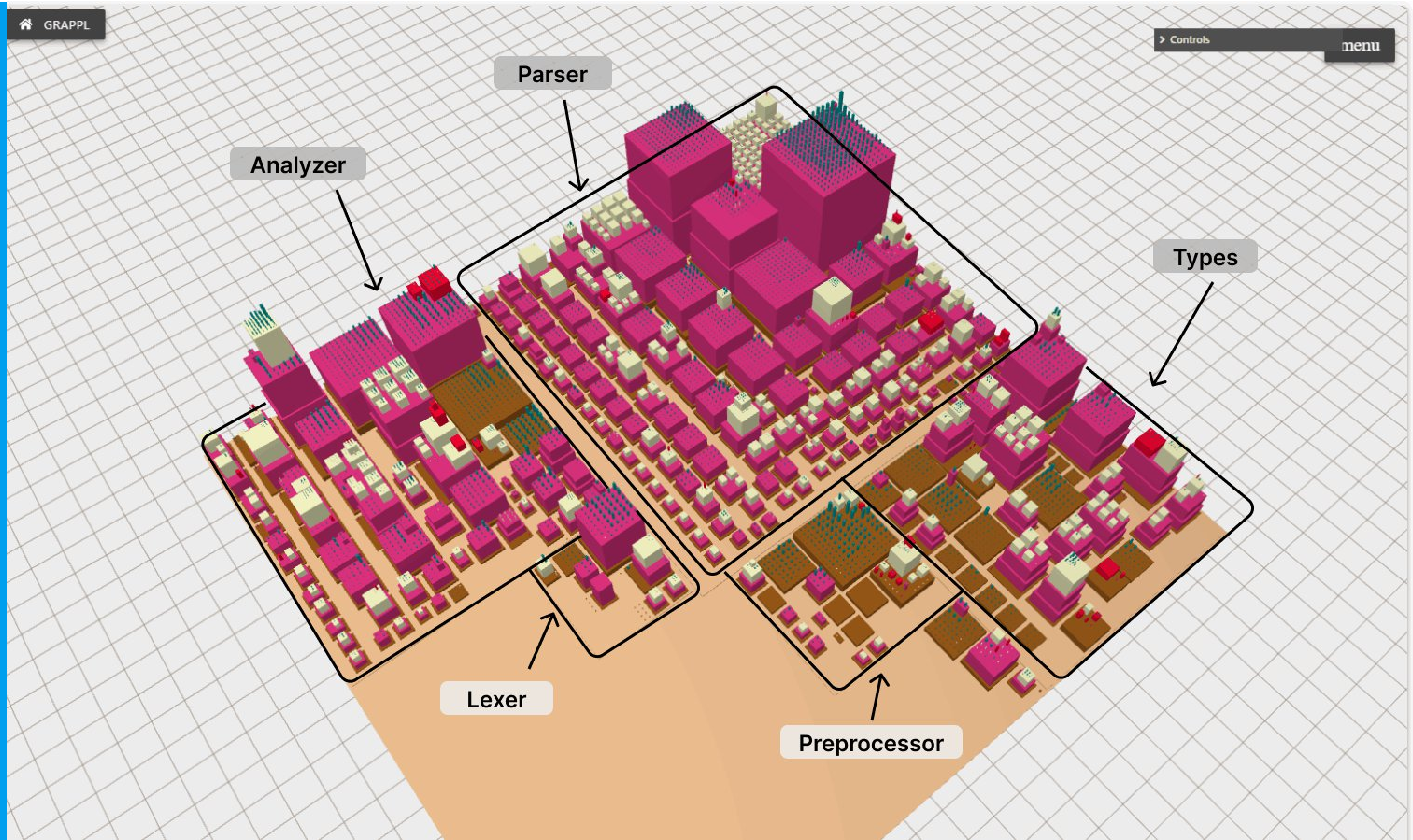


# Крупный деловой район: ядро анализатора

36









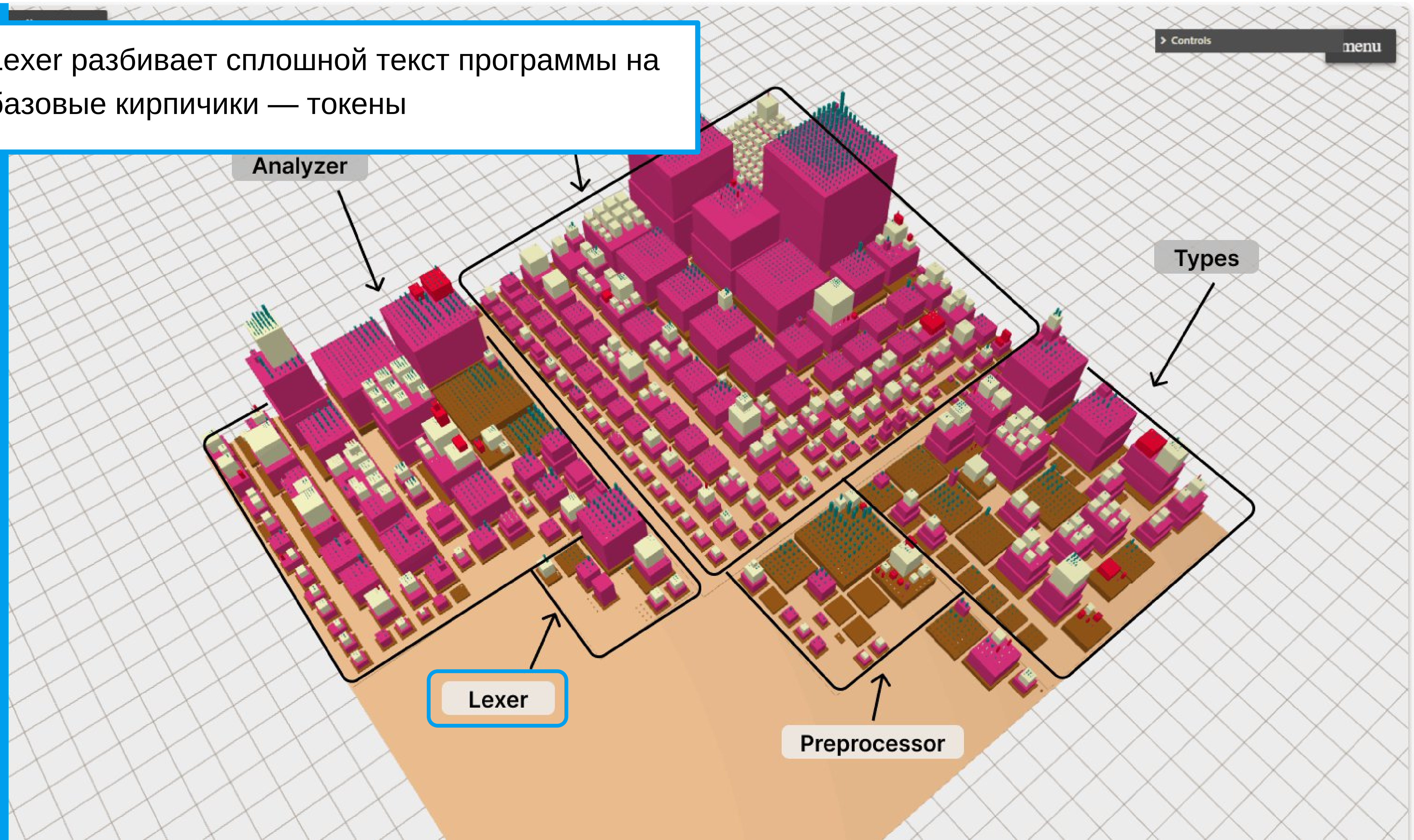
Препроцессор выполняет первоначальную подготовку файла:

- заменяет табы на пробелы;
- удаляет BOM;
- записывает builtin-функции;
- обрабатывает наши специальные комментарии;
- сопоставляет исходный и препроцессированный файлы через директивы `#line`;
- отслеживает расхождения в нумерации строк между файлами.

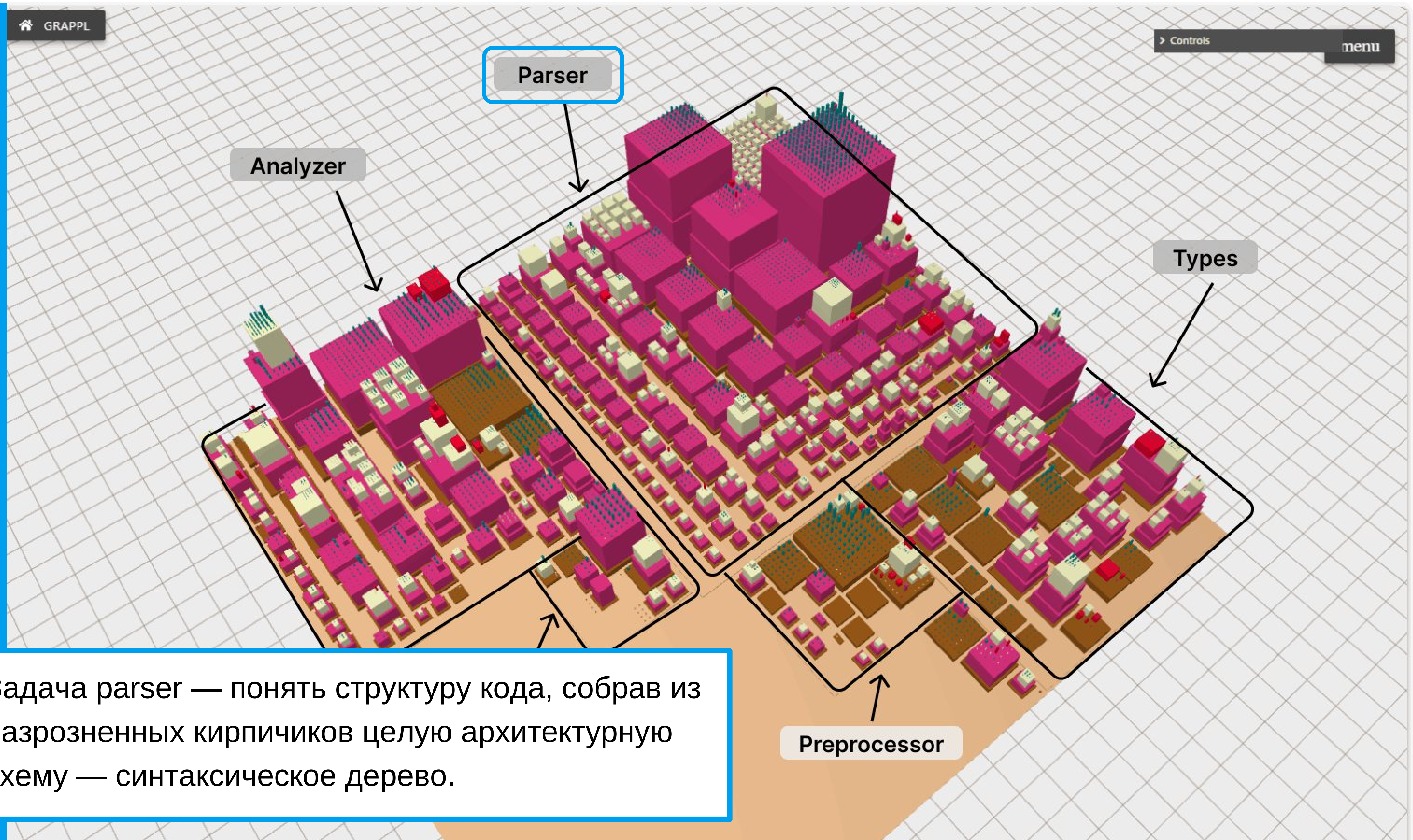




Lexер разбивает сплошной текст программы на базовые кирпичики — токены

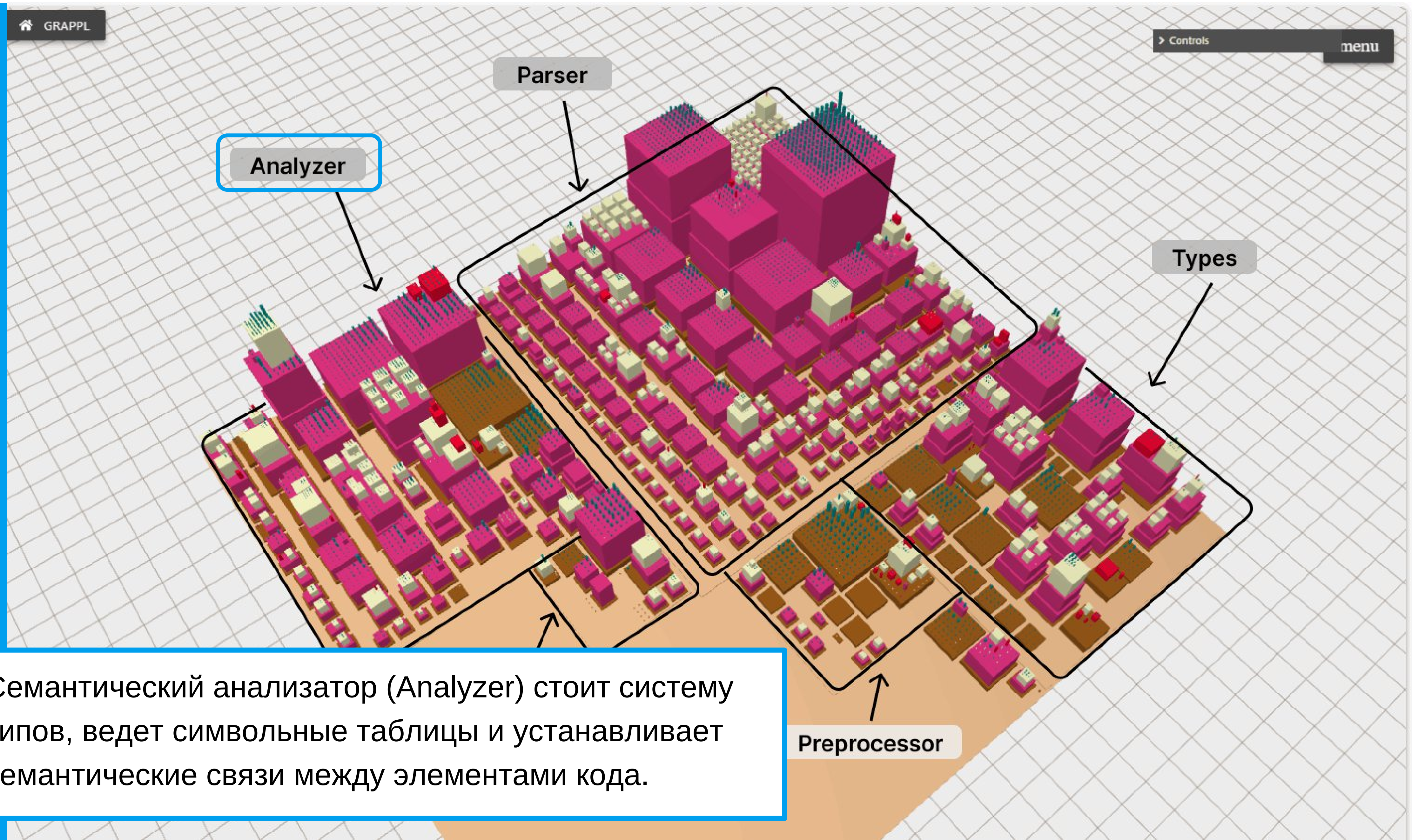






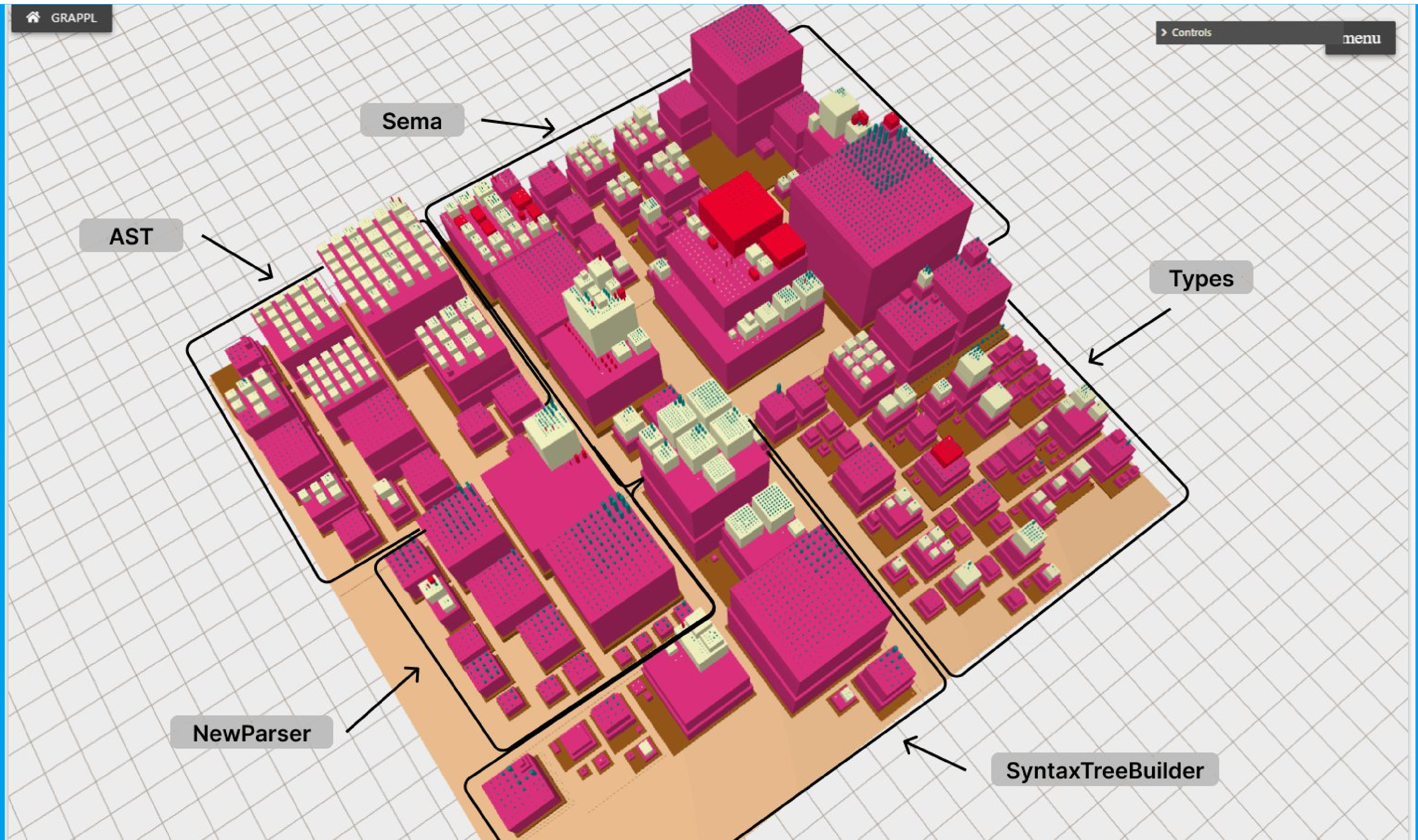
Задача parser — понять структуру кода, собрав из разрозненных кирпичиков целую архитектурную схему — синтаксическое дерево.





Семантический анализатор (Analyzer) стоит систему типов, ведет символные таблицы и устанавливает семантические связи между элементами кода.

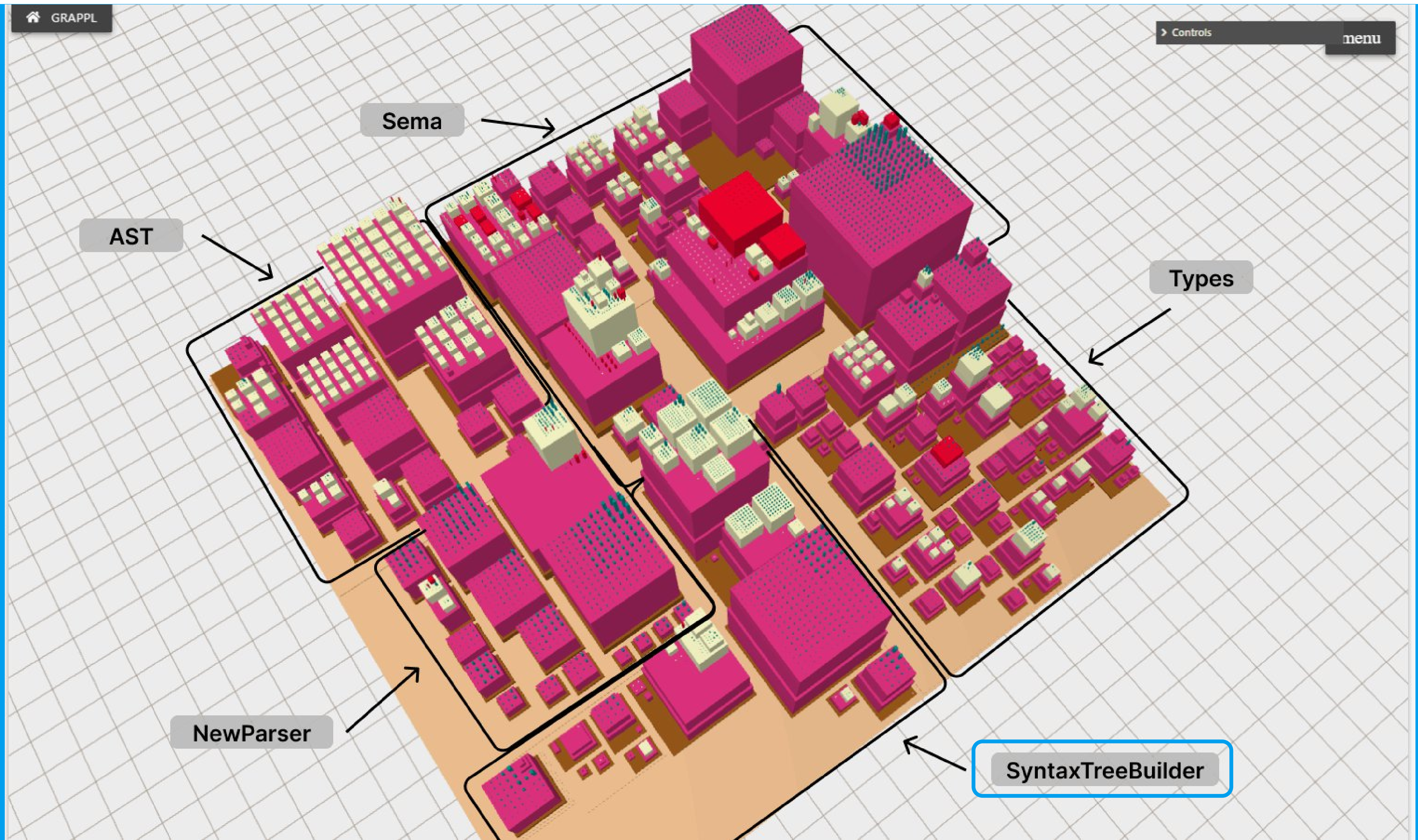




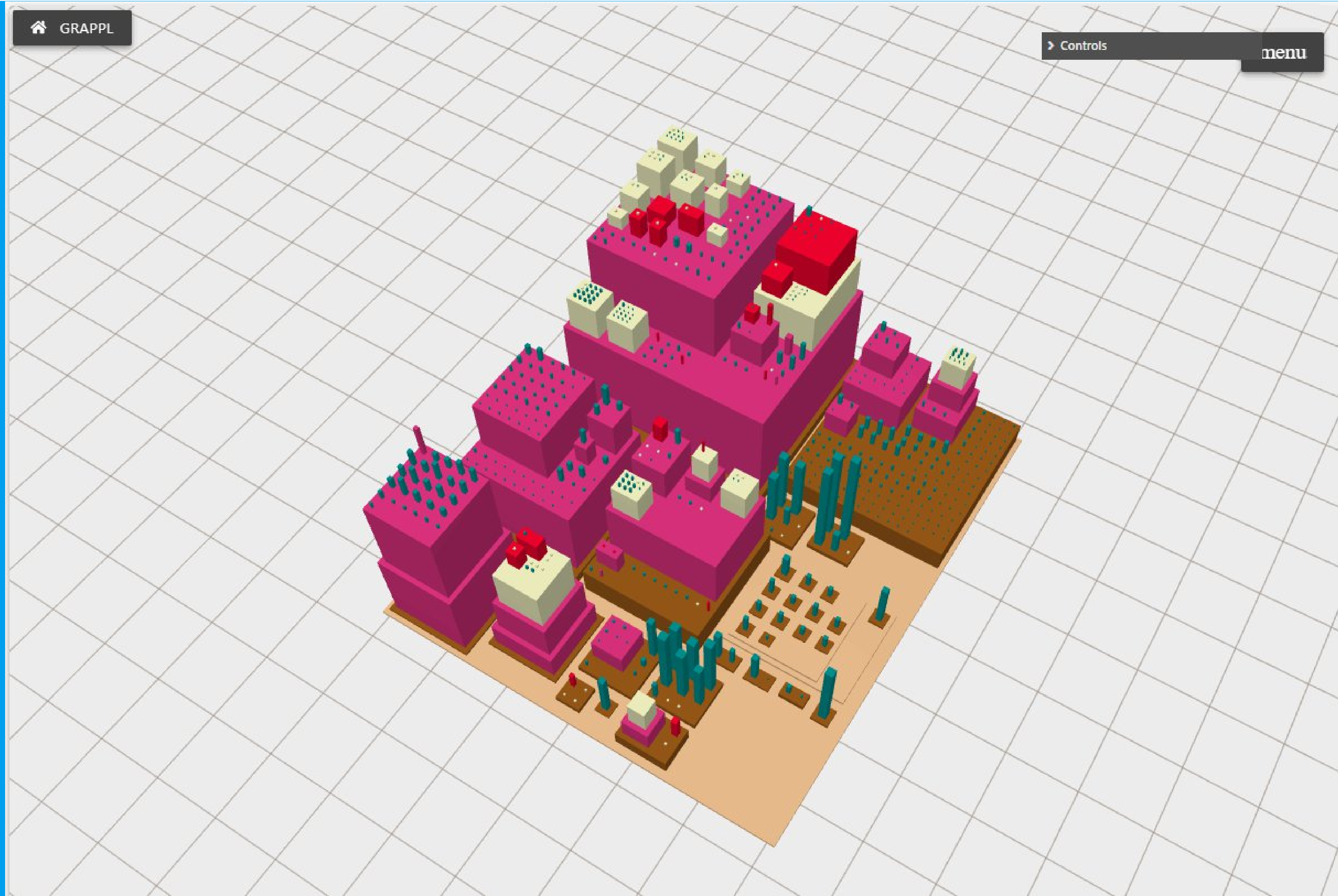














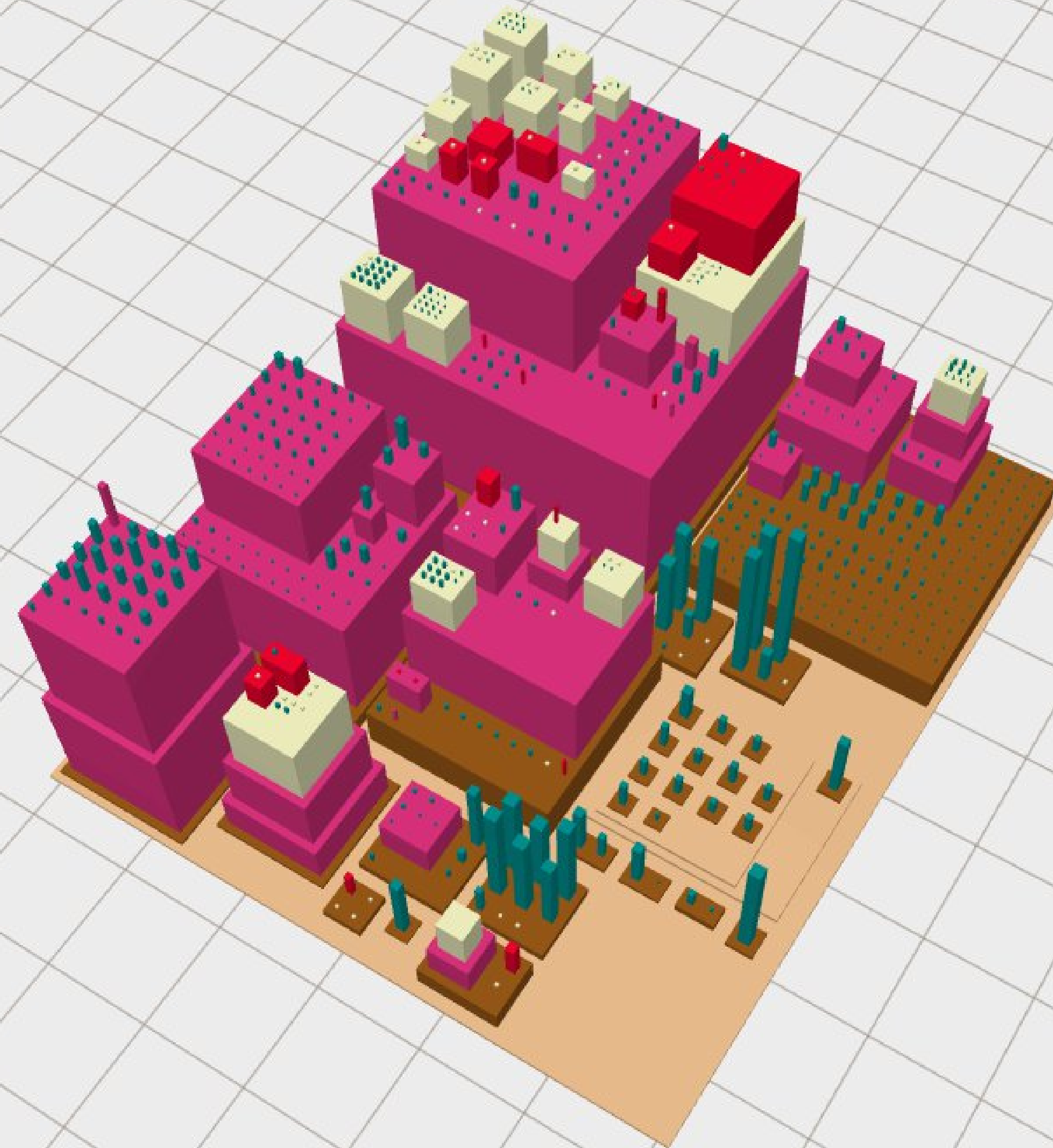
Блок аннотаций — внутренняя и внешняя система аннотирования сущностей.

Система содержит семантику классов стандартной библиотеки, информацию о поведении функций, их предусловиях и побочных эффектах.

Также в этой части находится механизм пользовательских аннотаций в формате JSON.

GRAPPL

Controls





Для типов:

- сходство со некоторыми классами из стандартной библиотеки
- семантику

Для функций:

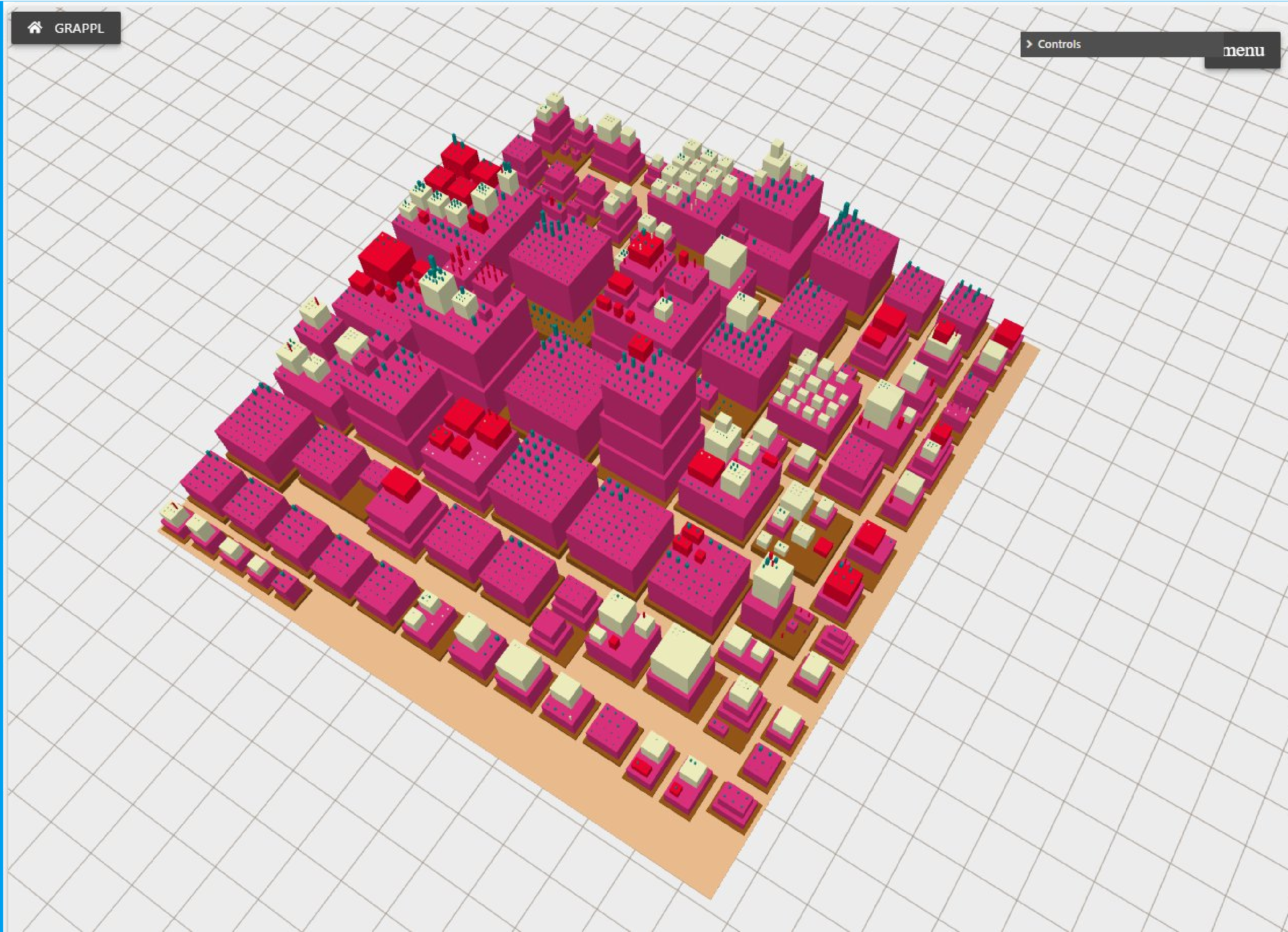
- свойства функции: не возвращает управление, объявлена как устаревшая и т.д.
- свойства каждого из параметров функции
- ограничения на параметры
- свойства возвращаемых значений

Механизм пользовательских аннотаций в формате JSON

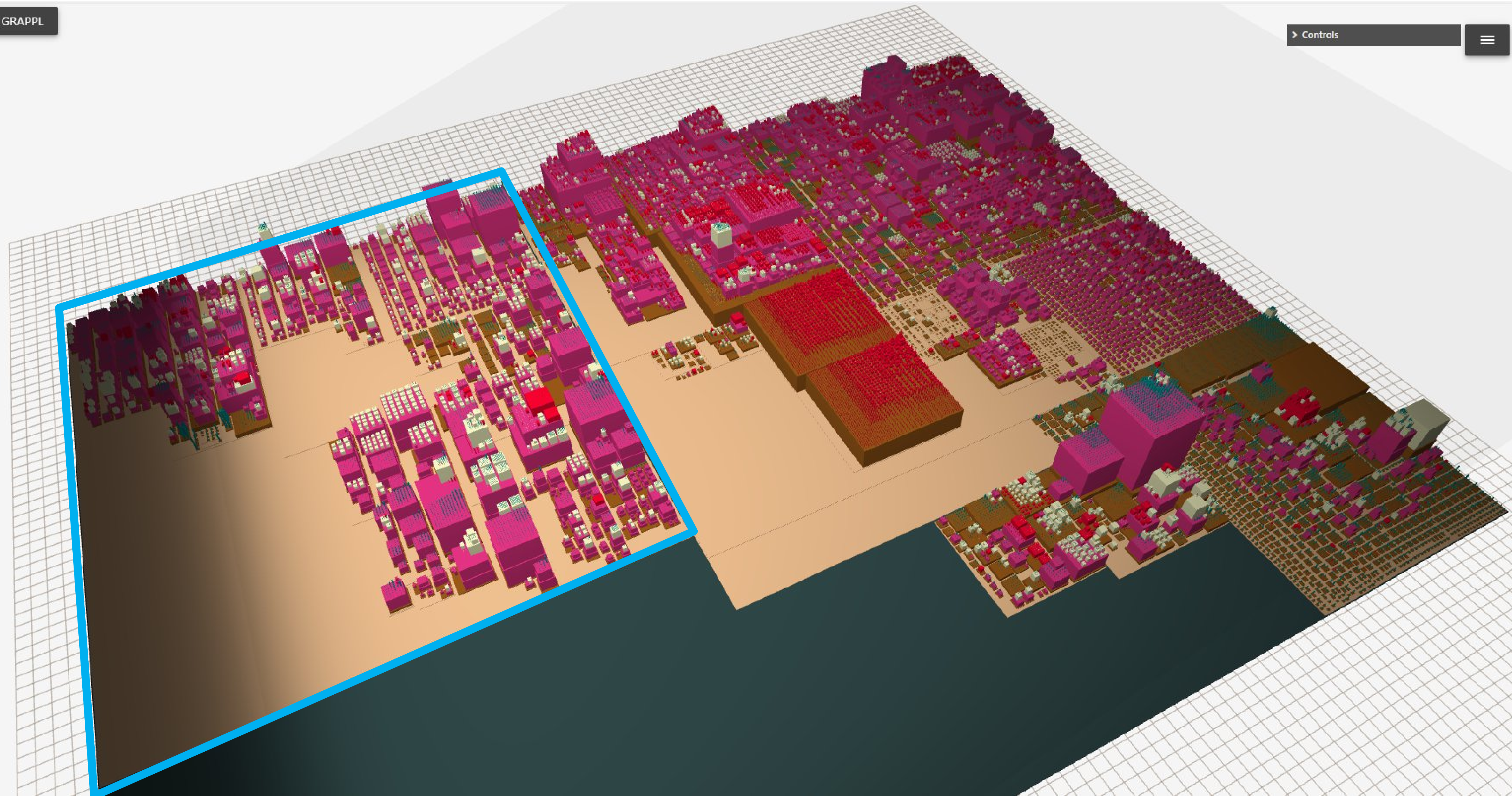


<https://pvs-studio.ru/ru/docs/manual/6810/>

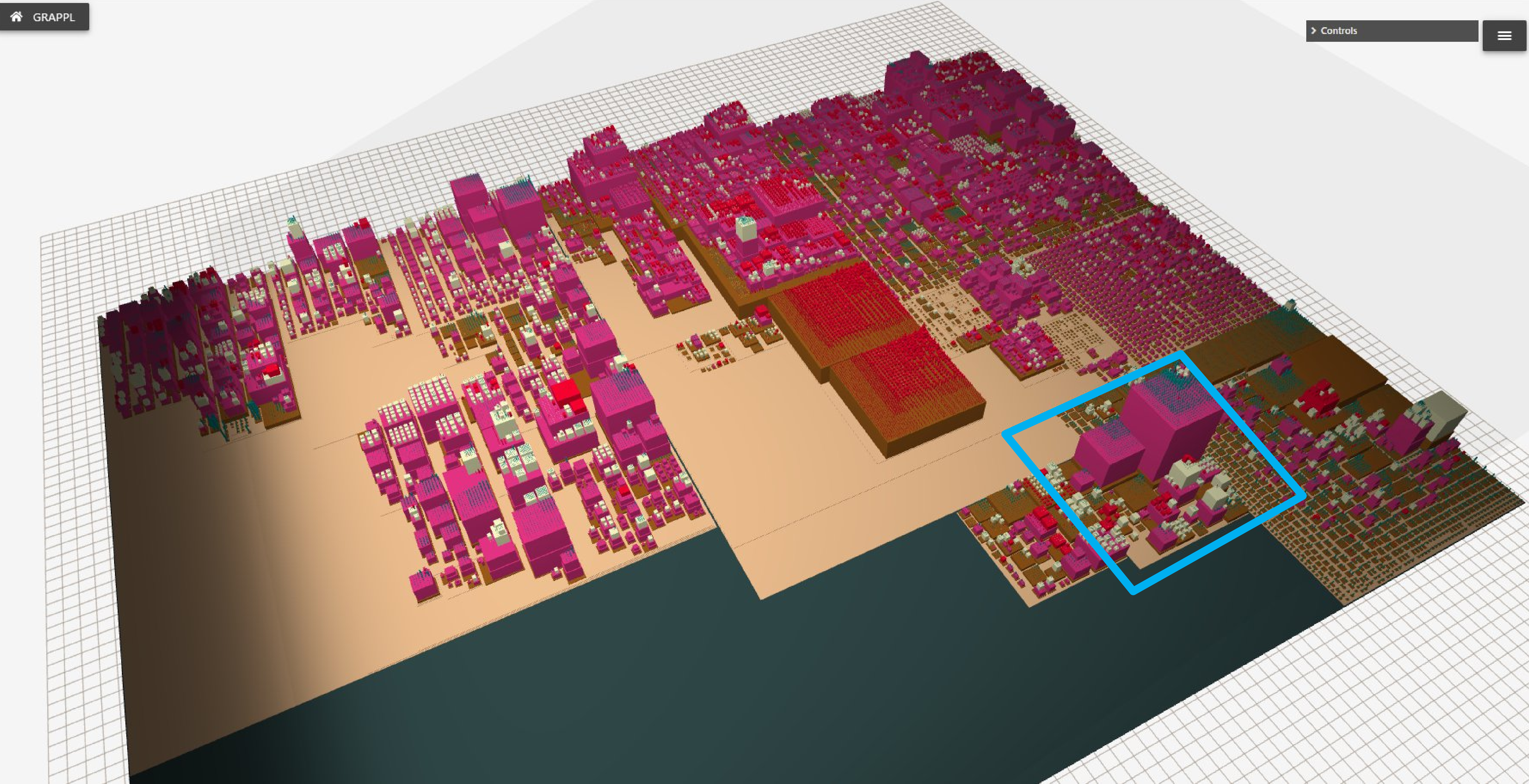




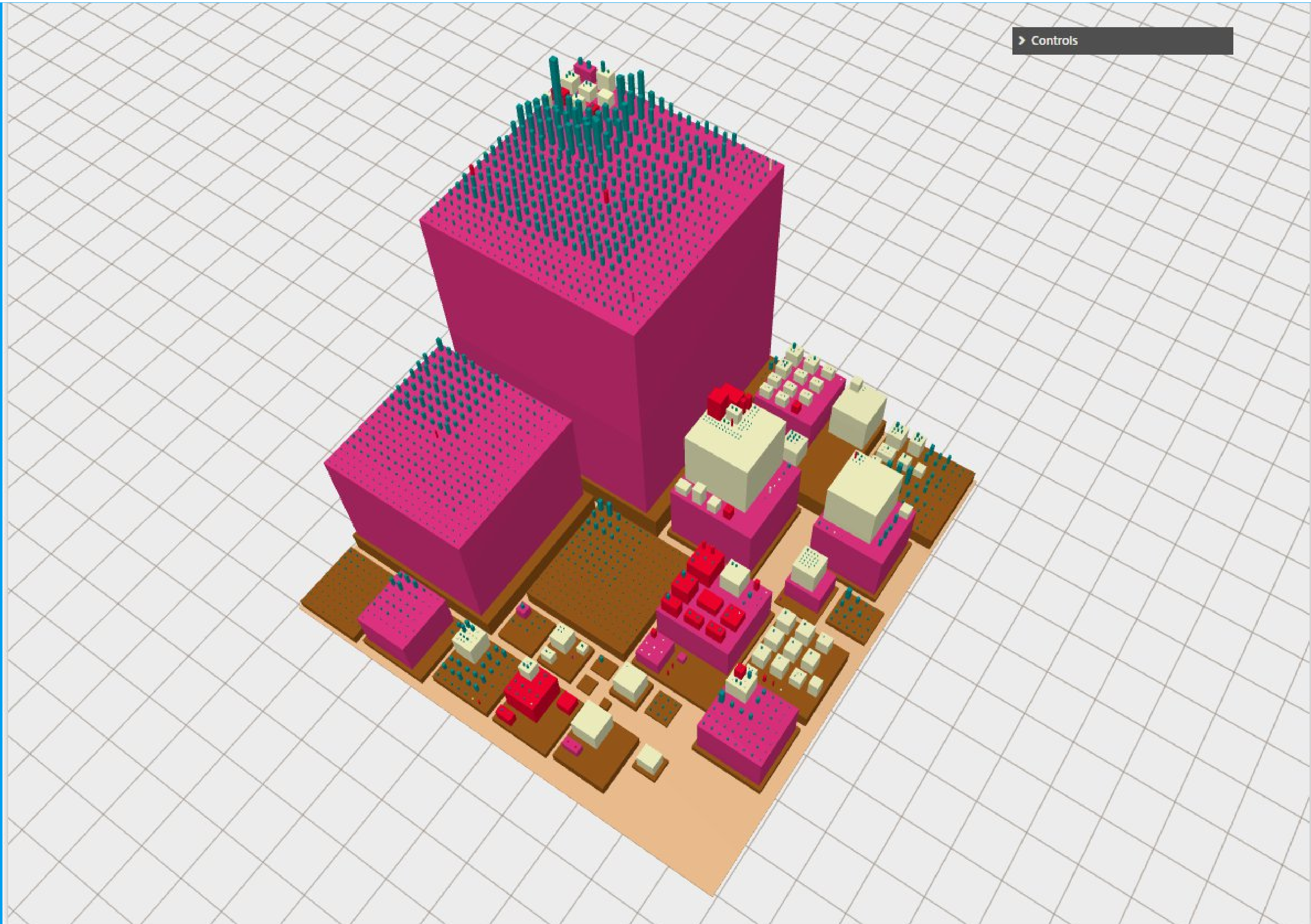








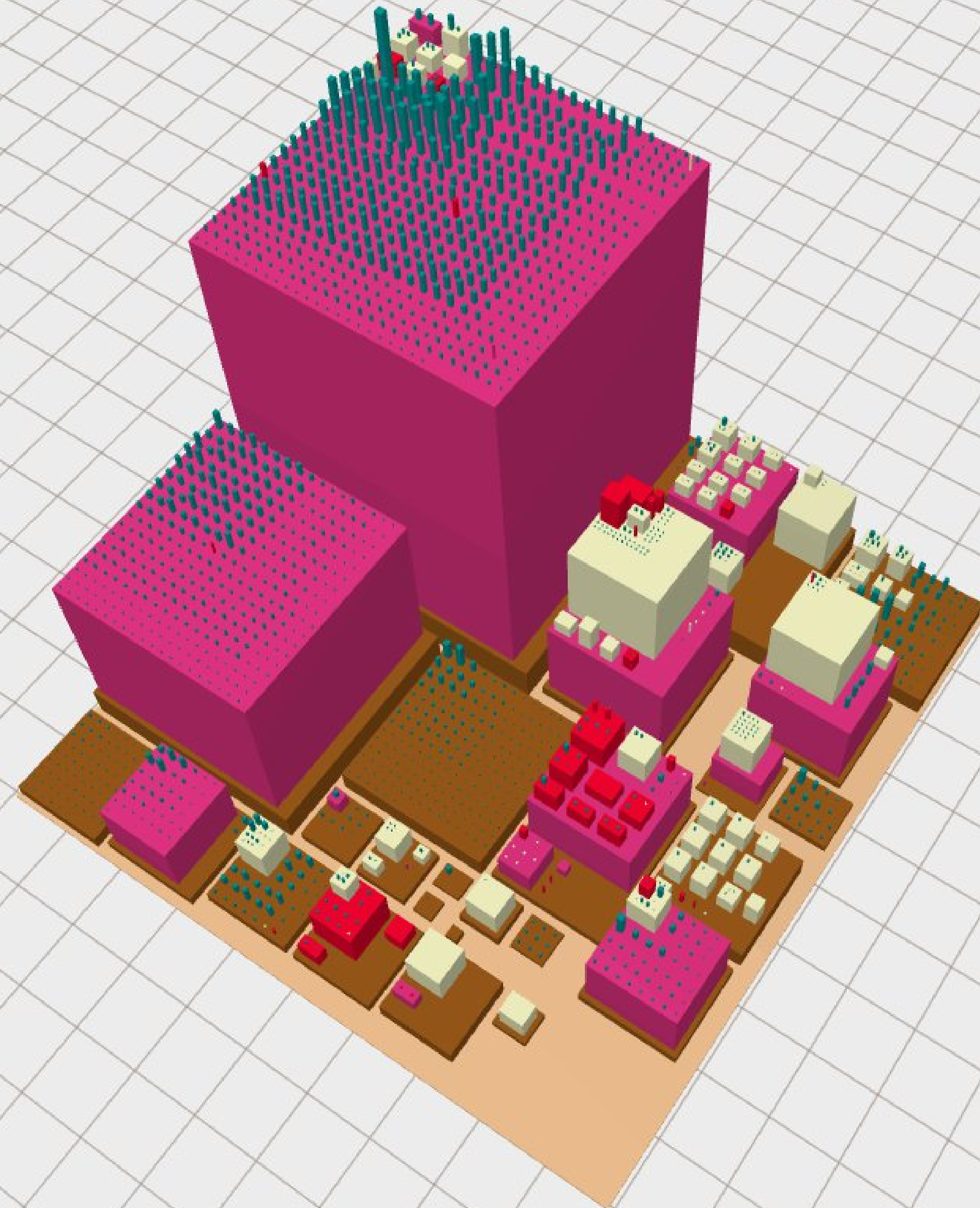




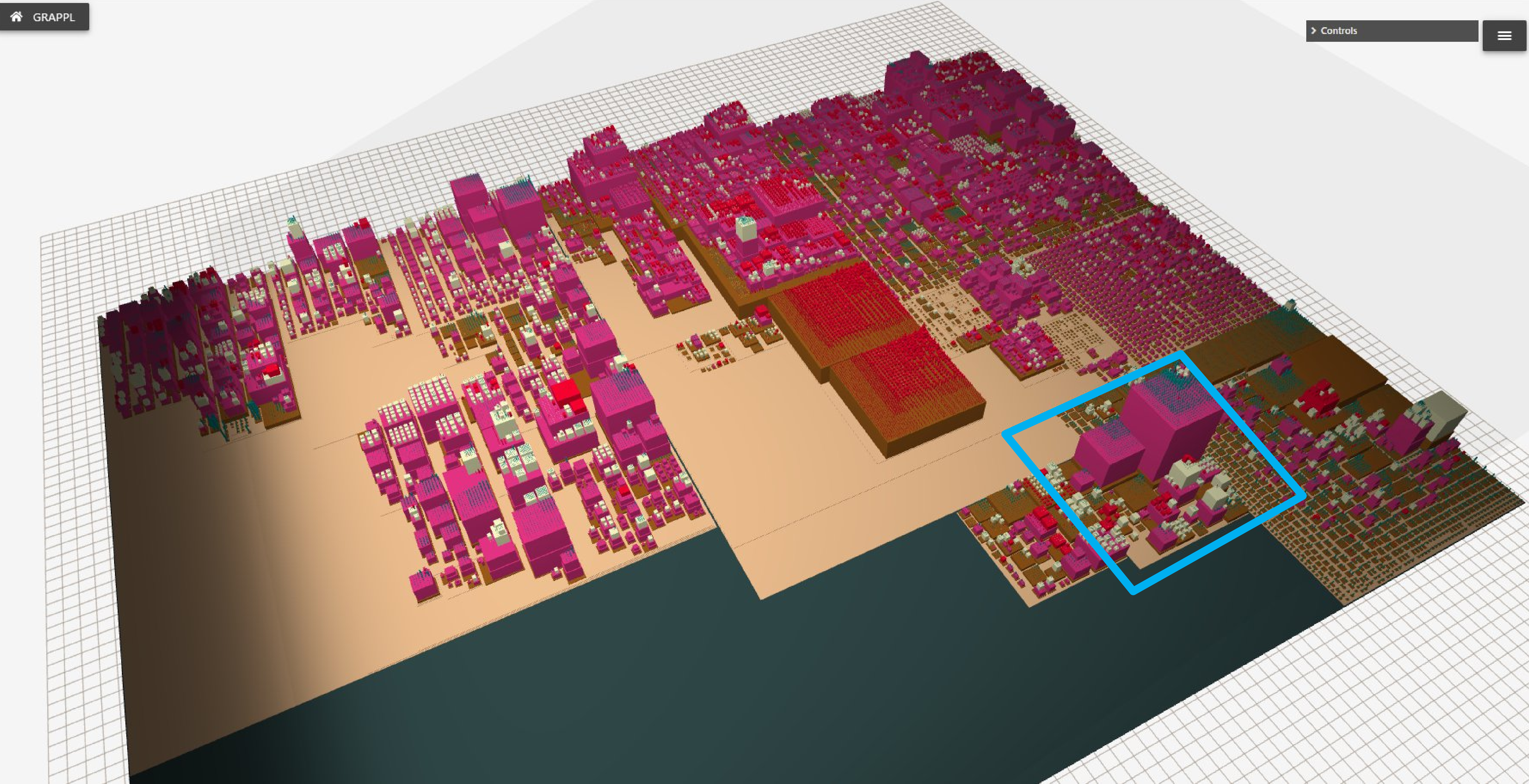


Здесь сосредоточены ключевые компоненты, отвечающие за организацию работы:

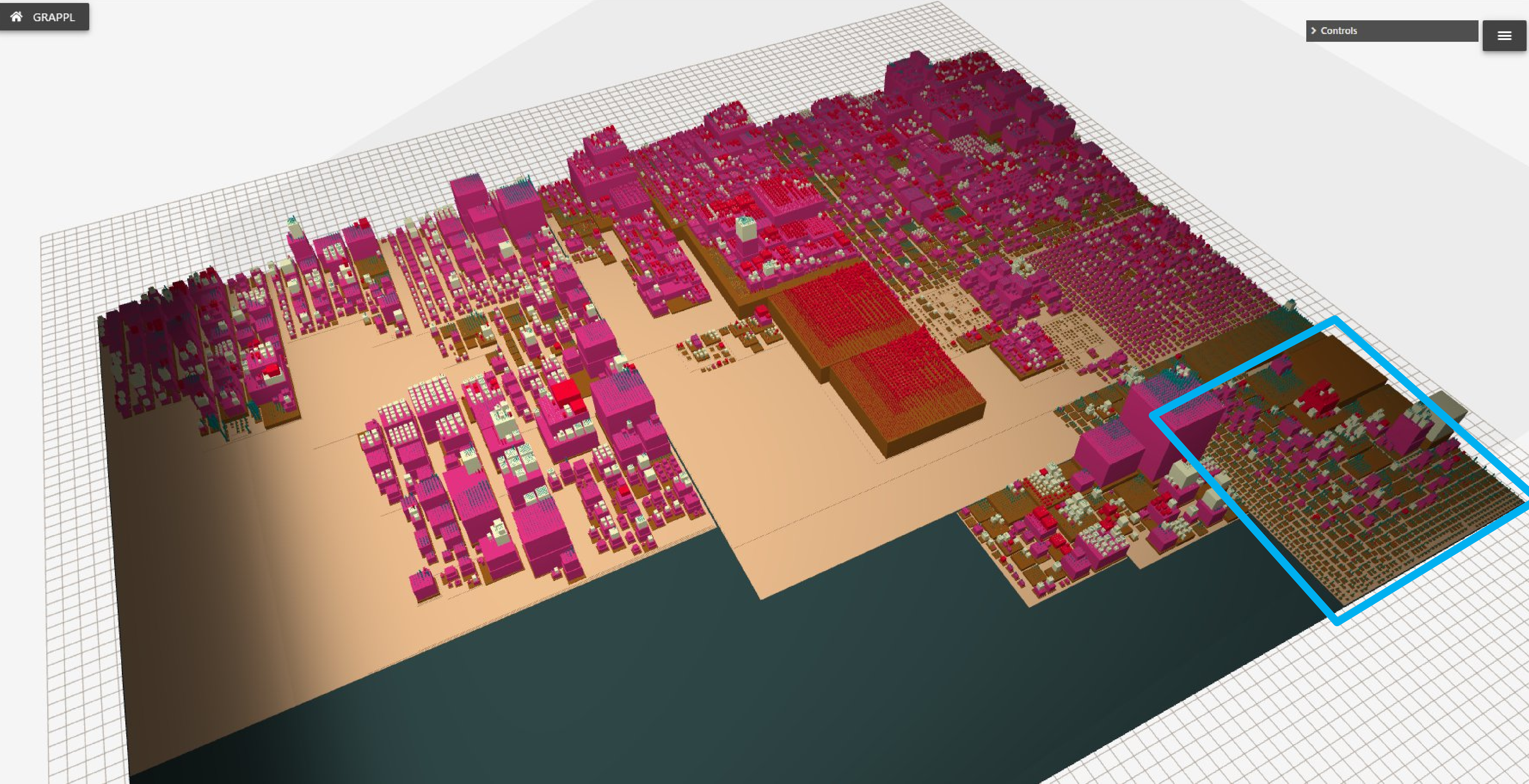
- обработка пользовательских настройки и определение параметров анализа
- система настроек анализа
- диспетчер правил
- обходчик дерева кода



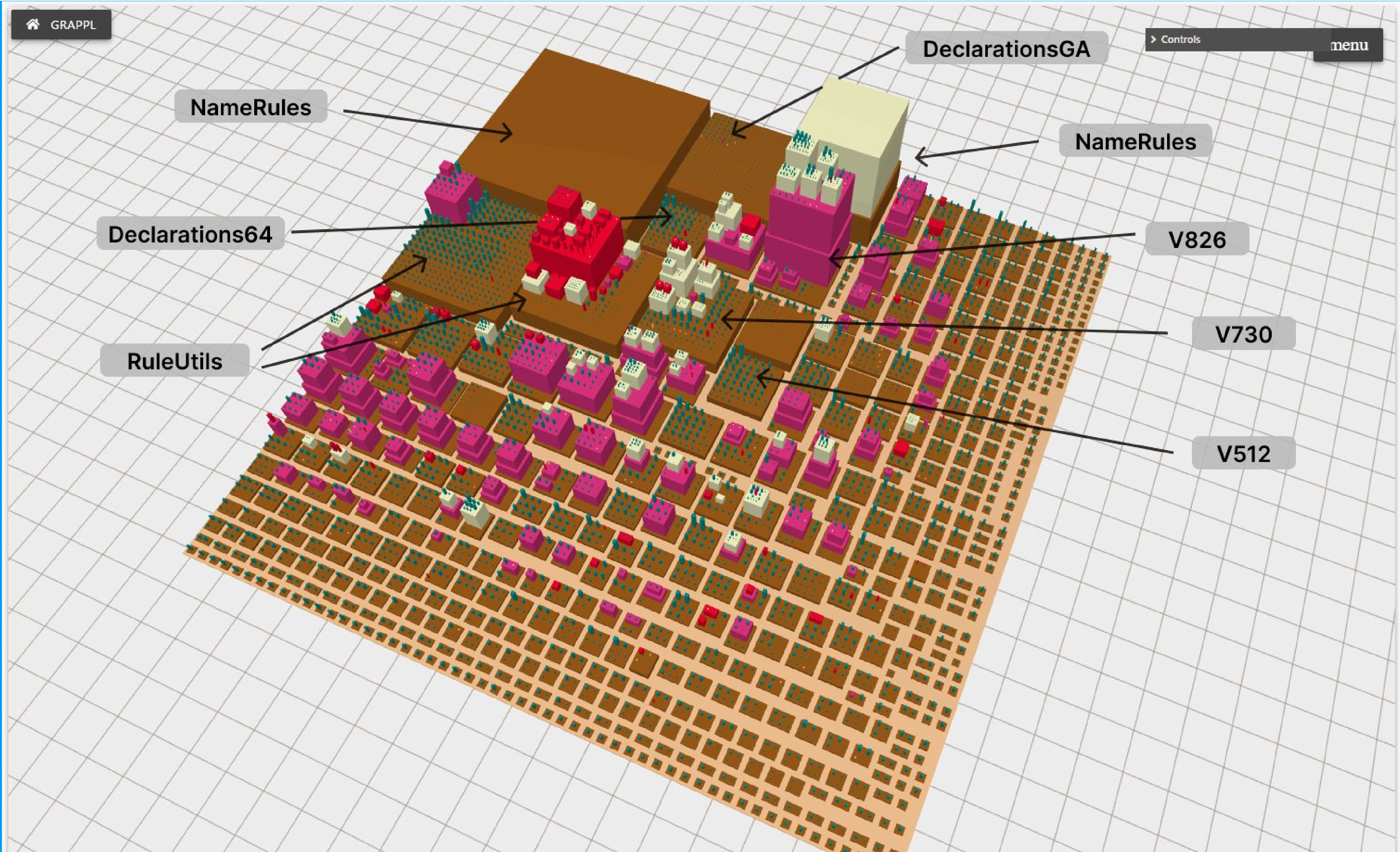




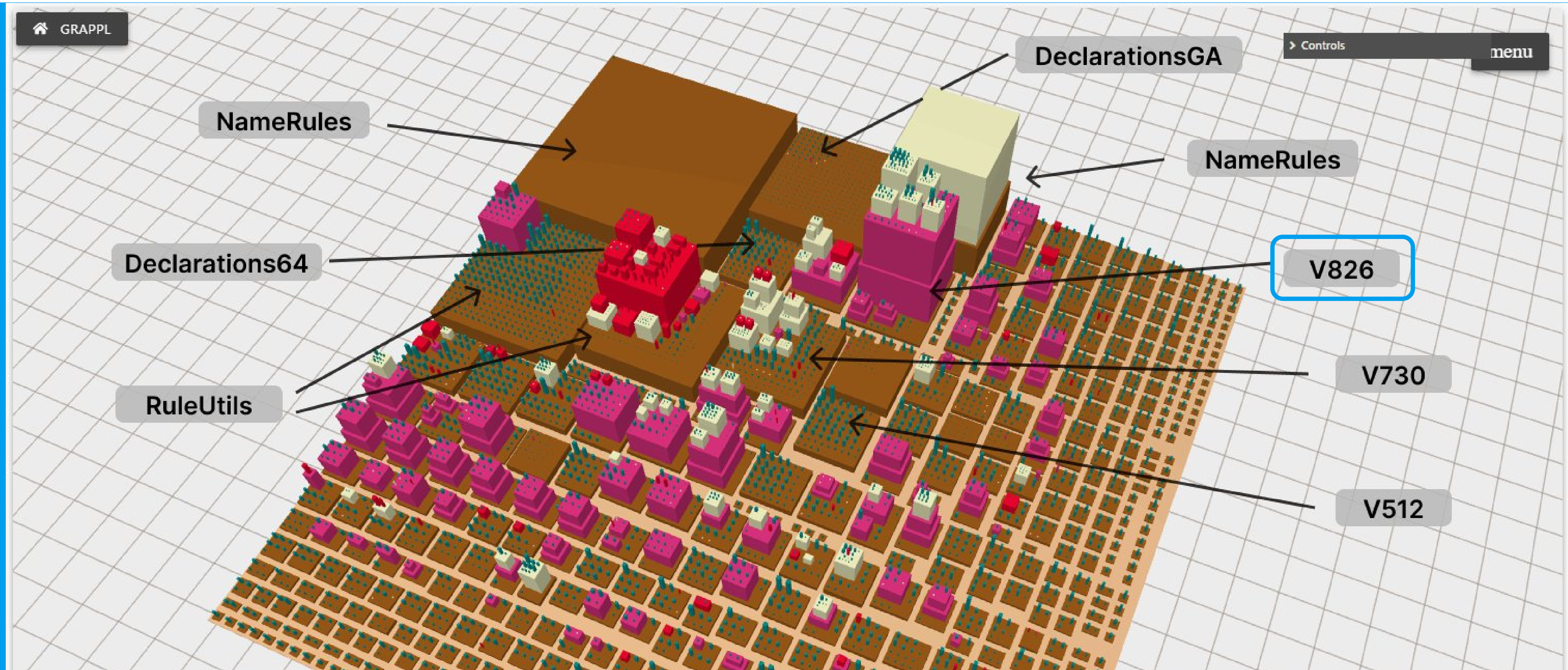












V826. Consider replacing standard container with a different one.

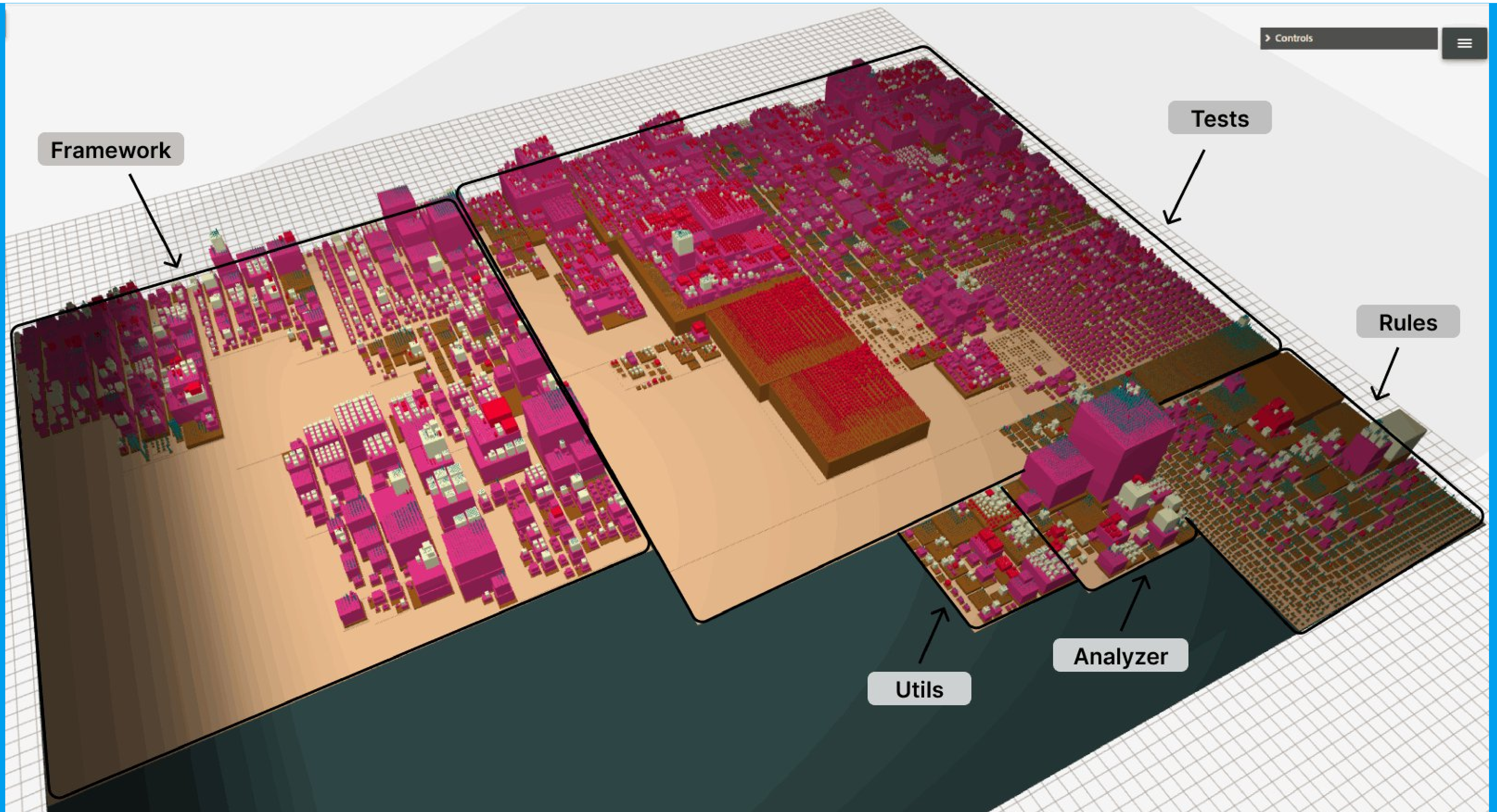
Анализатор обнаружил контейнер из стандартной библиотеки C++, который можно заменить на другой контейнер в целях оптимизации.



# Заканчиваем путешествие










# Grappl

59





<https://marketplace.visualstudio.com/items?itemName=grappl.grappl>

Visual Studio Code > Visualization > Grappl



## Grappl

**Preview**

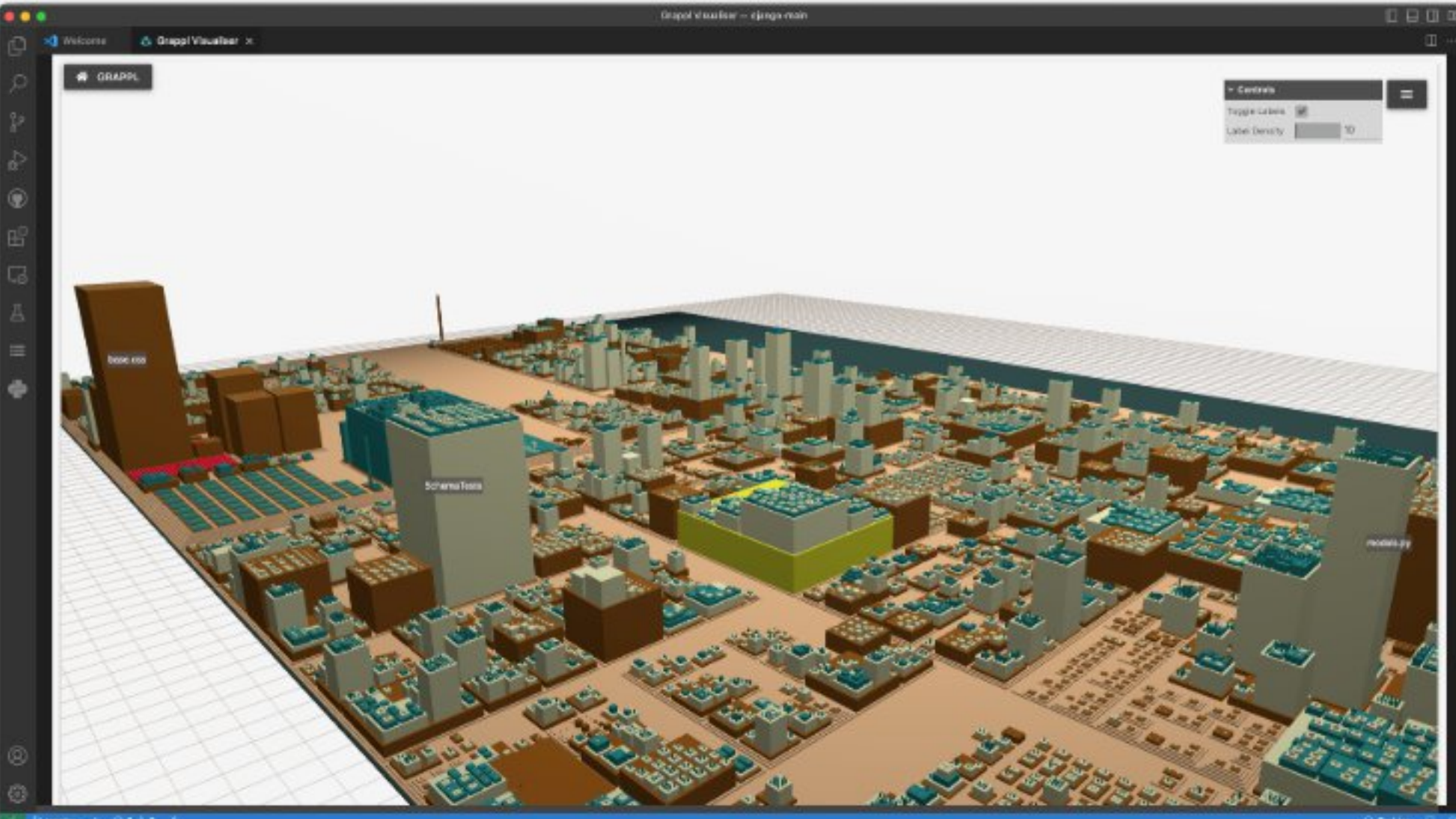
Grappl  [grappl.io](https://grappl.io) |  1,487 installs | ★★★★★ (2) | Free Trial

Spend less time deciphering code and more time innovating

[Install](#) [Trouble Installing?](#)

[Overview](#) [Version History](#) [Q & A](#) [Rating & Review](#)

### Grappl





# Остались вопросы

