

**Правительство Российской Федерации**  
**Федеральное государственное автономное образовательное учреждение**  
**высшего профессионального образования**  
**"Национальный исследовательский университет "Высшая школа**  
**экономики"**

*Отделение программной инженерии*  
*Кафедра Управления разработкой программного обеспечения*

УТВЕРЖДАЮ  
Зав. кафедрой УРПО

\_\_\_\_\_ С.М. Авдошин  
«\_\_» \_\_\_\_\_ 2014 г.

***ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА***  
***по направлению 231000.62 Программная инженерия***  
***подготовки бакалавра***

На тему: Плагин для Autodesk Maya для генерации трехмерной модели  
города на основе данных из открытых источников

Студента группы № 472ПИ \_\_\_\_\_ Николенко Артема Павловича  
Подпись (Ф. И. О.)

\_\_\_\_\_  
(Дата)

Научный руководитель

Доцент кафедры УРПО, к.т.н. \_\_\_\_\_ Брейман Александр Давидович  
(должность, звание) Подпись (Ф.И.О.)

\_\_\_\_\_  
(Дата)

Консультант

к.ф.м.н. \_\_\_\_\_ Соломенцев Дмитрий Валентинович  
(должность, звание) Подпись (Ф.И.О.)

\_\_\_\_\_  
(Дата)

Москва, 2014 г.

# Аннотация

В данной работе рассматривается разработка плагина, предназначенного для генерации трехмерных моделей городов в среде Autodesk Maya. Программа встраивается напрямую в трехмерный редактор и предоставляет простой доступ как к функциям генерации моделей городов, так и к вспомогательным функциям, таким как получение картографических данных посредством сети интернет, и генерация и настройка пути облета камерой модели и управление процессом Batch Render.

Данный плагин разрабатывался как часть проекта AeroState, целью которого является расчет распространения загрязнений в атмосфере городов. С помощью этого плагина был сгенерирован ряд моделей, которые использовались для создания визуальных материалов для презентаций проекта AeroState. В дальнейшем планируется дальнейшее развитие данного плагина, интеграция с данными о распространение загрязнений получаемыми с помощью системы OpenFOAM и использование данного плагина для визуализации реалистичных данных о распространении загрязнений.

В качестве среды трехмерного моделирования была выбрана Autodesk Maya, как мощный продукт легко поддающийся модификации. Разработка велась на языках Python, MEL и MATLAB. В качестве источников данных использовались карты OSM, высотная информация DEM и информация о состоянии атмосферы WRF. Для работы с данными использовались библиотеки GDAL и NetCDF4.

В результате работы над ВКР был разработан продукт полностью реализующий поставленные перед ним задачи. Данный плагин был успешно внедрен в рамках проекта AeroState, о чем есть соответствующий акт.

## Оглавление

Аннотация.....	1
Введение.....	4
Глава 1. Обзор имеющихся аналогов.....	6
Глава 2. Общий подход к генерации трехмерной модели .....	11
Глава 3. Особенности реализации .....	16
Заключение .....	21
Источники .....	23

# Введение

Потребность в продукте, разработка которого является целью данной ВКР, возникла в рамках проекта «AeroState». Целью данного проекта является создание программного комплекса способного производить высокоточные расчеты распространения загрязнений в условиях городской среды.

Во время проекта возникла необходимость в визуализации получаемых данных для демонстрации как потенциальным инвесторам, так и потенциальным клиентам. Было принято решение визуализировать данные путем наложения информации о загрязнениях на трехмерную модель города, что позволило бы легко получать высококачественные изображения и видео материалы. Для ускорения хода разработки было принято решение о разделении проекта на 2 части, разрабатываемые параллельно: на программу визуализации и на программу физических расчетов.

В качестве среды трехмерного моделирования была выбрана Autodesk Maya, за счет активного комьюнити, легкости скриптования на языках Python и Mel, и мощного физического движка, который позволил бы производить примитивные расчеты для рендеринга демонстрационных материалов до того, как будет завершена работа над программой произведения высокоточных расчетов распространения загрязнений на базе OpenFOAM.

Были проанализированы имеющиеся на рынке решения для генерации трехмерных моделей городов совместимые с Autodesk Maya. Выяснилось, что ни одно из имеющихся на рынке решений не подходит для целей проекта и было принято решение о разработки собственной программы генерации трехмерных моделей городов, на основе данных из открытых источников в виде плагина для Autodesk Maya, что и является целью данной ВКР.

В рамках данной ВКР были:

- проанализированы имеющиеся открытые источники геоданных
- изучена работа с форматами OSM XML, DEM, NetCDF

- реализован вспомогательный компонент для получения данных о пробках путем анализа данных сервиса Яндекс.Карты(в целях получения данных о загрязнении для демонстрационных визуализаций)
- изучен функционал Autodesk Maya, Maya Python API, Maya Embedded Language
- реализован функционал генерации трехмерных моделей городов
- реализован ряд вспомогательных функций, призванных облегчить пользование продуктом.

# Глава 1. Обзор имеющихся аналогов.

Данная глава посвящена определению 2 основных подходов используемых при генерации трехмерных моделей городов, обзору ряда существующих решений и объяснению, по какой причине было принято решение о разработке собственного продукта, вместо использования одного из уже присутствующих на рынке решений.

Необходимость в генерации трехмерных моделей городов присутствует во многих сферах. Навигационные программы с помощью подобных моделей увеличивают наглядность предоставления информации, различные телевизионные передачи используют модели городов в новостных и погодный выпусках, индустрия кино и индустрия видеоигр используют подобные модели для воссоздания событий, происходящих в городской среде. В проекте «AeroState», частью которого является разработанный в рамках данной ВКР плагин, мы столкнулись с необходимостью получения визуального отображения распространения загрязнений в городской среде.

Были проанализированы подходы к написанию подобных программ и имеющиеся на рынке решения. Как оказалось, существуют 2 основных подхода:

## **1. Процедурная генерация.**

Данный подход заключается в том, что структура города и зданий задается не вручную, и не на основе каких-либо данных, а генерируется посредством некоего алгоритма и его входных параметров. Процедурная генерация широко распространена в компьютерной графике.

Преимуществами данного подхода является легкая масштабируемость получаемой модели, и при использовании достаточно продвинутого алгоритма генерации возможно задания множества параметров получаемой модели.

Недостатками являются долгое время работы при использовании достаточно сложных алгоритмов, сложность в реализации подобных продвинутых алгоритмов, недостаточная реалистичность получаемой городской среды при использовании простых алгоритмов и ограниченная область применения. Очевидно, что полученные подобным образом модели могут использоваться только в случае если нужен просто произвольный город, а виртуальная версия реального города.

## **2. Генерация на основе геоданных.**

Данный подход использует данные описывающие городскую среду из различных географических информационных систем, и производит генерацию модели города на основе полученных данных.

В преимущества к данному подходу можно записать высокую скорость работы и реалистичность получаемой городской среды за счет того, что она является отображением реально существующего города.

Недостатком можно считать необходимость получения достоверных входных данных для генерации и невозможность достаточно гибко настроить получаемый результат из-за специфики самого подхода.

Подобные модели востребованы там, где требуется модель реальной городской среды, например, в новостных передачах, научных расчетах, медиа проектах, делающих реалистичность одной из своих особенностей.

Как видно из вышесказанного каждый подход обладает своими плюсами и минусами и занимает свою нишу.

Был произведен анализ имеющихся на рынке решений. Так как среди трехмерного моделирования множество, а большинство подобных продуктов являются плагинами к подобным системам, было принято решение рассматривать исключительно standalone решения и продукты совместимые с Autodesk Maya [1], так как данная система удовлетворяла всем потребностям проекта.

### **1. Esri CityEngine**

CityEngine [2] представляет из себя самостоятельное приложение для трехмерного моделирования, основной специализацией которого является генерация трехмерных моделей городской среды. Данное приложение было разработано компанией Procedural Inc, которая в дальнейшем была приобретена компанией Esri специализирующейся на производстве геоинформационных систем.

Данная система реализует оба подхода. Возможно проведение как процедурной генерации, так и генерации на основе геоданных.

Несмотря на очевидные преимущества специализированного проприетарного ПО, такие как широкий функционал с рамках области специализации и наличие

квалифицированной службы поддержки данное решение было отклонено по 2 причинам.

Во-первых, функция генерации по геоданным, которая и требуется в рамках проекта «AeroState» в данном решении ориентирована на собственную геоинформационную систему Esri – ArcGIS, в то время как изначально было принято решение об ориентации на открытые источники данных, такие как Open Street Map, с которыми данная система работает с трудом и множеством ошибок.

Во-вторых, функционал рендеринга в рамках данной системы представлен в минимальном объеме, а экспортированные модели при импорте в Autodesk Maya содержали множество ошибок.

## **2. KludgeCity**

KludgeCity [3] является плагином для Autodesk Maya для генерации трехмерных городов разработанным Ed Whetstone. Несмотря на данный продукт способен генерировать весьма детализированные и красивые модели зданий в нашем случае он не применим по причине того, что нам необходимо воссоздание реальной модели города, в то время как KludgeCity способен только на процедурную генерацию. Помимо этого, здания создаваемые плагином имеют весьма футуристичный вид, что не подходит к нашему случаю. И хотя плагин имеет открытый код, было принято решение что модификация кода для задач проекта «AeroState» будет не оправдана.

## **3. QTown**

QTown [4] представляет из себя плагин для Autodesk Maya.

Данный продукт умеет производить исключительно процедурную генерацию, что делает его неприменимым для наших целей.

Также данный продукт ориентирован на генерацию крупномасштабных моделей городов, из-за чего возникают трудности при необходимости генерации меньшего масштаба (район, улица и т.п.).

Кроме этого похоже, что разработка данного продукта была прекращена в 2011 году, из-за чего могут быть обнаружены проблемы в совместимости данного плагина с новыми версиями Autodesk Maya.



#### 4. Ninja city

Данный продукт также является плагином для Autodesk Maya [5].

Как и предыдущие он не ориентирован генерацию города по геоданным, и реализует процедурный подход к генерации.

В плагине реализована возможность генерации зданий по заданным пользователем “отпечаткам” – двумерным контурам оснований зданий. Хотя в теории подобный функционал мог бы позволить использовать данный продукт в сочетании с отдельным скриптом, предназначенным для генерации отпечатков по геоданным, для генерации реалистичной городской среды, этот подход был отклонён из-за больших трудозатрат необходимых для приведения полученной модели к необходимым нам критериям.

*Таблица 1*

Сравнение имеющихся решений

	<b>Esri CityEngine</b>	<b>KludgeCity</b>	<b>QTown</b>	<b>Ninja City</b>
<b>Совместимость с Autodesk Maya</b>	Частичная	Полная	Полная	Полная
<b>Цена</b>	Высокая	Бесплатно	Бесплатно	Низкая
<b>Подход к генерации</b>	Процедурный/на основе геоданных	Процедурный	Процедурный	Процедурный
<b>Работа с открытыми источниками</b>	Ограниченная	Отсутствует	Отсутствует	Отсутствует
<b>Возможность модификации</b>	Отсутствует	Ограниченная	Ограниченная	Ограниченная

Как видно из приведенного выше обзора имеющихся на рынке продуктов большинство из них не подходят по причине процедурного подхода к генерации.

Единственным продуктом способным генерировать трехмерную модель города на основе геоданных оказался Esri CityEngine, но он был отклонён по причине плохой

совместимости с Autodesk Maya и выбранной геоинформационной системой Open Street Map.

В связи во всем вышесказанным было принято решение о необходимости разработки собственного решения, в формате плагина для Autodesk Maya, способного производить генерацию трехмерной модели города на основе геоданных.

## Глава 2. Общий подход к генерации трехмерной модели

В данной главе будет рассказано о используемых входных данных и принципе работы алгоритма генерации трехмерной модели города.

Данному плагину для работы требуются 2 категории данных. Это данные описывающие структуру самого города и данные о состоянии атмосферы в нем.

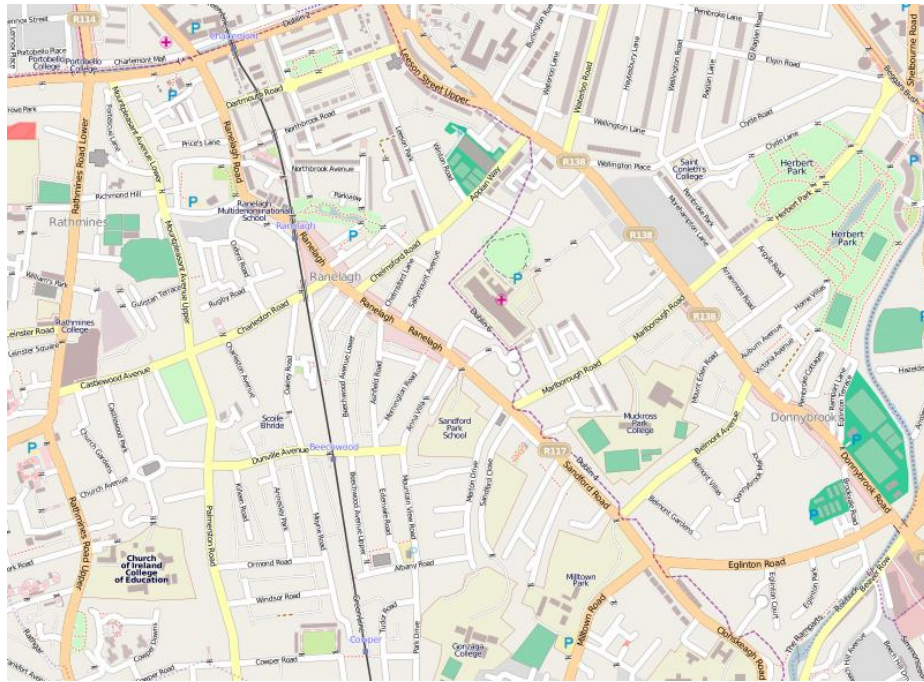
Структуру города можно разделить на 2 вида данных: двумерные картографические данные, описывающие взаимное расположение зданий и дорог, и данные о высотах. Если картографические данные получить сравнительно просто (в рамках данного проекта эти данные будут получаться от системы Open Street Map), то с высотами все сложнее. Данные об этажности зданий считаются закрытыми, поэтому чтобы хотя бы приблизительно задавать высоты зданий были использованы данные DEM.

Данные об атмосфере города также разделяются на 2 типа: состояние самой атмосферы и данные о загрязнениях. Состояние атмосферы мы получаем из WRF данных, а в качестве информации об источниках загрязнений в данный момент в демонстрационных целях выступают данные о пробках от сервиса Яндекс.Карты.

Немного подробнее про каждый формат:

### **1. Open Street Map XML**

Open Street Map – это некоммерческий проект по созданию подробной и открытой карты мира (Рис. 1) силами Интернет сообщества. Проект был создан в 2004 году и активно развивается по сей день.



*Рис. 1 Пример карты проекта OSM*

Open Street Map позволяет экспортировать фрагменты карты в XML формат [6].

Файл OSM XML содержит:

- Координаты границ данного участка карты
- Список тегов Node
- Список тегов Way
- Список тегов Relation.

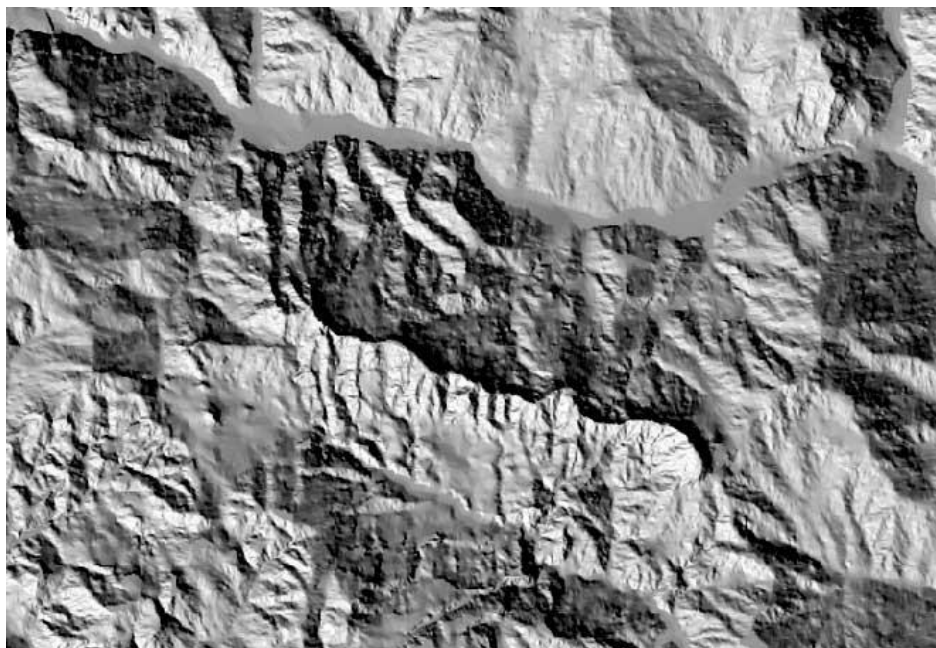
Для наших задач нам интересны границы участка, а также информация из тегов Node и Way.

Теги Node представляют из себя точку на карте. Каждый node имеет уникальный id, координаты и ряд служебных данных которые нам в рамках данного проекта не интересны.

Теги Way описывают двумерный объект на карте. Каждый way имеет уникальный id, содержит список id точек node задающих собой данный объект и ряд дополнительной информации, из которой нам будет интересен тип данного объекта.

## 2. Digital elevation model

Digital elevation model [7] представляет из себя двумерную матрицу содержащую высокоточную информацию о рельефе поверхности планеты, включая здания (Рис. 2).



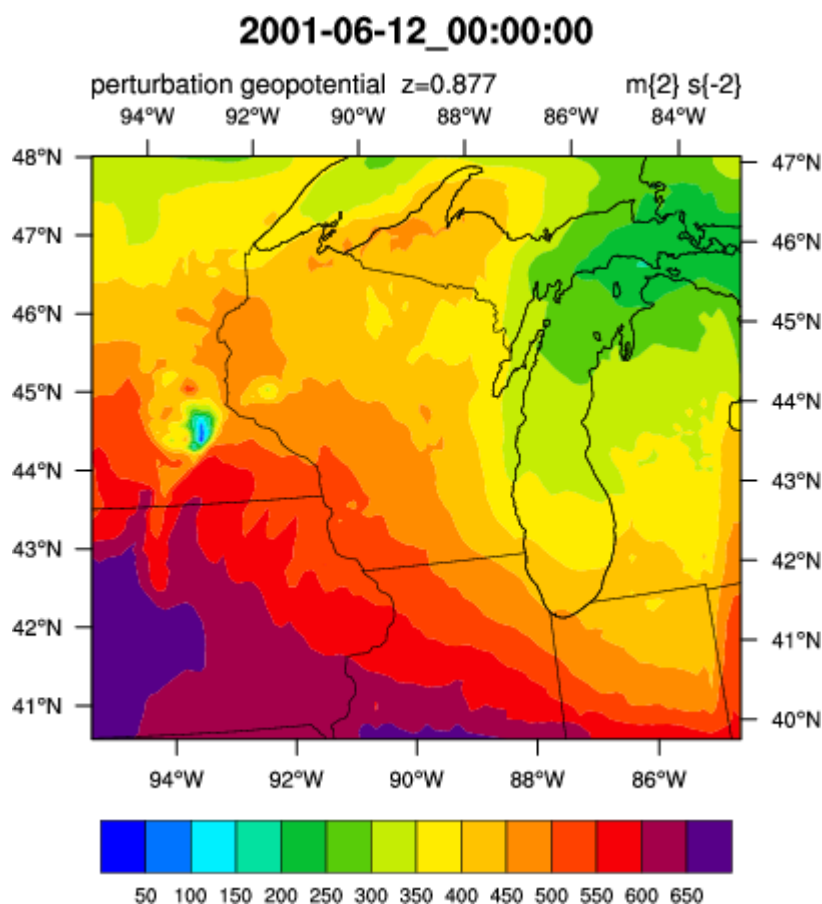
*Рис. 2 Графическое представление файла DEM*

В рамках данного проекта данные DEM используются для расчета высот зданий. В текущей версии проекта высота местности над уровнем моря задается пользователем. В будущих версиях возможно использование двух подвидов DEM, Digital Terrain Model (DTM), содержащей информацию исключительно о рельефе местности и Digital Surface Model (DSM), содержащей информацию о рельефе поверхности включая здания, для более точных расчетов высот зданий.

### **3. Weather Research and Forecasting Model**

Weather Research and Forecasting Model [8] является системой численного прогноза погоды созданной для применения как в исследовательских целях, так и для нужд прогнозирования погоды (Рис. 3).

Данные WRF хранятся в файлах NetCDF (Network Common Data Form) [9] являющимся стандартом для обмена научными данными.



*Рис. 3 Графическое представление данных WRF*

Каждый файл содержит ряд многомерных таблиц, описывающих различные аспекты состояния атмосферы, для различных координат в различное время.

В рамках данного проекта применяются данные о направлении и скорости ветров в зоне генерации трехмерной модели города.

#### **4. Данные о пробках.**

В целях получения демонстрационных материалов до момента готовности модуля точных расчетов распространения загрязнений в городской среде в данном проекте используются данные о пробках сервиса Яндекс.Карты. Так как общедоступного API по которому можно было бы получить данную информацию не существует, то был написан скрипт на MATLAB [10], который скачивает слой карт содержащий информацию о точках, и конвертирует полученные графические файлы в описание уровня интенсивности пробок по координатам.

Вышеперечисленные входные данные обеспечивают все необходимое для генерации трехмерной модели города.

Алгоритм генерации трехмерной модели города имеет следующий вид:

- Вычисляются коэффициенты перевода географических координат в координаты Autodesk Maya.
- По границам области из файла OSM XML создается базовая плоскость
- Считываются данные о высотах
- У каждого node координаты конвертируются в систему координат Autodesk Maya и в соответствие каждому node ставится значение высоты данной точки.
- Для каждого way выбираются принадлежащие ему точки.
- Если way является дорогой, то происходит генерация дороги по соответствующим точкам, после чего на основании данных о пробках для данной дороги вычисляется коэффициент излучения частиц и ей присваивается эмиттер с соответствующим значением.
- Если way является зданием, то мы вычисляем его высоту как среднее арифметическое высот образующих его точек, и производим генерацию соответствующего здания.
- Создаются поля атмосферы для данной области, согласно параметрам, полученным из WRF модели.

## Глава 3. Особенности реализации

Данная глава посвящена подробному разбору процесса разработки плагина, являющегося темой данной работы, проблемам которые были выявлены во время разработки и особенностям реализации.

Как уже было сказано в предыдущих главах, в качестве платформы для реализации генерации трехмерных моделей городов была выбрана система Autodesk Maya. Разработка плагинов для данной системы возможна с помощью четырех основных языков: MEL, Python, C++, C#. Был проведен анализ всех вариантов:

### 1. MEL

Maya Embedded Language (MEL) [11]– это скриптовый язык который был разработан компанией Autodesk с целью упростить выполнение сложных либо рутинных операций в среде Maya и позволить пользователям создавать и распространять расширяющие функционал скрипты.

MEL синтаксически схож с Perl и в особенности с Tcl (который использовался разработчиками Autodesk Maya на ранних этапах). Он позволяет не просто создавать пользовательские скрипты, но полностью кастомизировать среду Maya, так как большая часть окружения и инструментов Maya написаны на MEL. Код написанный на MEL является платформонезависимым и может исполняться на любой операционной системе для которой доступно исполнение самой Maya.

В плюсы написания скриптов на MEL стоит отнести наиболее полный доступ к функционалу Maya и возможность практически неограниченной настройки окружения.

Минусом является недостаток функционала языка за пределами задач, связанных непосредственно с взаимодействием с Autodesk Maya.

### 2. Python

Maya Python API [12]– это интерфейс прикладного программирования Maya разработанный для языка Python.

Python – это высокоуровневый интерпретируемый язык в первую очередь ориентированный на читабельность исходного кода и повышения



производительности разработчика. За счет простоты в изучении и огромного количества различных сторонних библиотек в последнее время он стал одним из стандартов в программировании систем трехмерного моделирования.

Maya Python API предоставляет доступ к большинству функций доступных MEL, но в то же время за счет богатого функционала самого языка и большого количества расширений предоставляет гораздо лучшие возможности в областях не связанных с взаимодействием с Autodesk Maya. В случае если необходимые разработчику функции Maya недоступны из Python API имеется возможность вызовов MEL команд непосредственно из Python кода.

### **3. C++**

Maya C++ API [13] был разработан чтобы удовлетворить потребности крупных студий трехмерного моделирования в написании тяжелых инструментов для собственного пользования.

Основным преимуществом плагинов, написанных на C++ по сравнению с MEL и Python является быстрое действие. В среднем код написанный с использованием Maya C++ API выполняется в 10 раз быстрее чем аналогичный скрипт реализованный с помощью MEL или Python. Аналогично с Python API в C++ API имеется возможность вызова команд MEL из C++ кода.

### **4. C#**

Maya C# API [14] – это последнее добавление в семейство API Maya. Разработанный в конце 2012 года данный интерфейс прикладного программирования призван позволить вести разработку модулей Maya на языке C# и в более лаконичном стиле по сравнению с Python и C++ API.

К сожалению, так как данный API сравнительно новый информации о нем и о деталях разработки с его помощью в интернете очень мало, но один недостаток виден сразу. Так как платформа .Net доступна только на операционных системах Windows, то и плагины, написанные на Maya C# API могут выполняться только на данной системе.

Результаты анализа доступных языков были сведены в таблицу 2.

Сравнение языков программирования совместимых с Autodesk Maya

	<b>MEL</b>	<b>Python API</b>	<b>C++ API</b>	<b>C# API</b>
<b>Доступ к функциям Maya</b>	Полный	Практически полный	Практически полный	Практически полный
<b>Быстродействие</b>	Медленное	Медленное	Высокое	Среднее
<b>Кроссплатформенность</b>	Полная	Полная	Полная	Отсутствует
<b>Количество сторонних библиотек</b>	Мало/отсутствуют	Множество	Множество	Среднее
<b>Сложность освоения</b>	Средняя	Низкая	Высокая	Средняя

Так как в рамках данной задачи генерации трехмерных моделей городов не ставилось требования высокого быстродействия, но, зато, требовалось взаимодействовать со специфическими форматами данных и, по возможности, код плагина должен был быть доступен для понимания членам команды не являющимися профессиональными разработчиками ПО, то в качестве языка разработки был выбран Python, обладающий всеми необходимыми сторонними библиотеками и простым синтаксисом.

Следующим этапом разработки приложения было изучение форматов входных данных и методов работы с ними. В качестве библиотек для работы с данными DEM и WRF были выбраны библиотеки GDAL и NetCDF4-Python:

### 1. GDAL

GDAL (Geospatial Data Abstraction Library) [15] – Open Source библиотека для работы с растровыми гео-пространственными форматами данных. Библиотека совместима с множеством языков, в том числе и с Python. В рамках данного проекта библиотека использовалась для получения данных о высотах из .tif файлов DEM.

### 2. NetCDF4-Python

Данная библиотека [16] предназначена для работы с данными в формате NetCDF из языка Python. В рамках данного проекта библиотека использовалась для получения данных о скорости ветра по трем координатным осям из .nc файлов WRF.

На данном этапе были выявлены следующие проблемы:

- Отсутствие открытого API позволяющего получать информацию о пробках с сервиса Яндекс.Пробки

- Несовместимость библиотек необходимых для работы с данными DEM и WRF со встроенным в Autodesk Maya интерпретатором Python

Для решения первой проблемы было решено получать информацию о пробках путем анализа графического представления слоя пробок сервиса Яндекс.Карты. Данный подход не обеспечивает высокую точность получаемых данных, но так как эти данные используются исключительно для создания демонстрационных визуализаций, а не для реальных расчетов, то низкая точность не является проблемой.

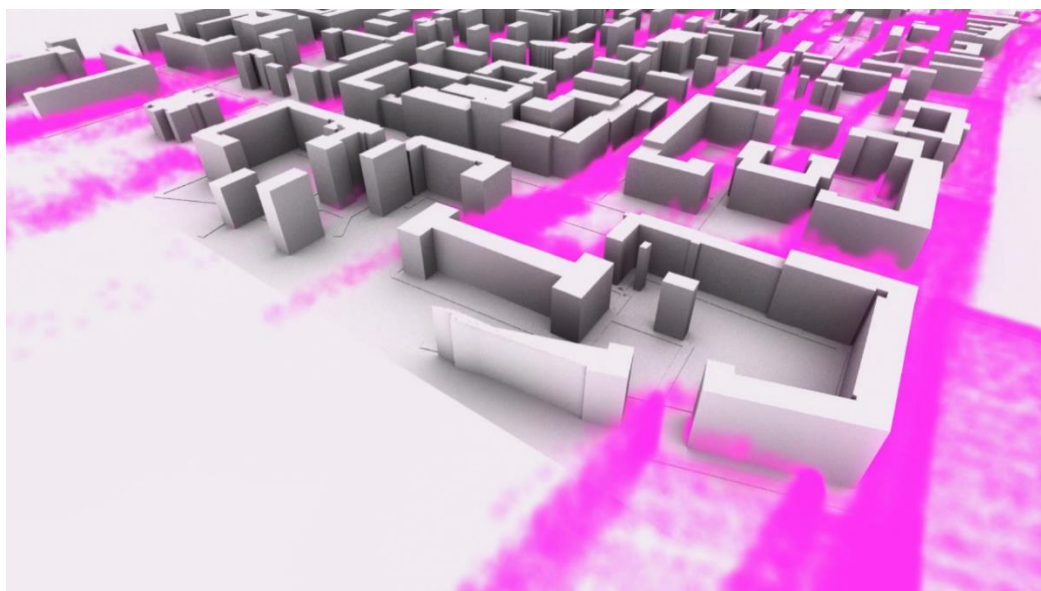
Реализация модуля получения данных о пробках была проведена на языке MATLAB. Данный выбор был обусловлен тем что часть проекта AeroState реализована именно на этом языке, и написание модуля получения данных о пробках на этом языке было хорошей тренировкой, призванной помочь в будущем в работе с модулями, написанными на MATLAB. Также MATLAB хорошо приспособлен к обработке больших массивов данных, таких как слой пробок для целого города.

В качестве решения проблемы несовместимости библиотек GDAL и NetCDF4-Python со встроенным в Autodesk Maya интерпретатором Python обработка файлов DEM и WRF была вынесена в отдельные модули. Данные модули предназначены для запуска на системном интерпретаторе Python с установленными на него соответствующими библиотеками и производят выделение из общего массива данных информации относящейся непосредственно к рассматриваемой географической области, извлечения ее и сохранения в простой текстовый формат, который может быть обработан основным программным модулем без участия сторонних библиотек. Эти модули могут быть вызваны как непосредственно из интерфейса плагина, так и посредством командной строки.

После решения всех проблем связанных с обработкой входных данных стало возможно приступить к реализации непосредственно механизма генерации трехмерной модели города. Так как данный модуль является основным и должен подключаться непосредственно к Autodesk Maya, то базовая структура модуля была задана стандартами написания плагинов для Maya.

Первым делом данный модуль считывает данные о заданной географической области и вычисляет коэффициенты преобразования координат на основе заданного пользователем масштаба. Создается базовая плоскость модели. Следующим этапом является создание и конфигурация используемых в процессе визуализации объектов частиц. После этого производится считывание данных о высотах на рассматриваемой

области, за которым следует считывание информации о точках карты области. С помощью информации о картографических объектах на основе полученных данных о высотах производится генерация зданий и дорог и конфигурация их параметров. В случае если пользователем был задан файл с информацией о пробках, то дополнительно создаются эмиттеры частиц в зоне дорог. В конце задаются поля ветров и гравитации (Рис. 4).



*Рис. 4 Пример рендера на основе 3D модели города*

Во время разработки данного проекта было также выявлено что для рендеринга видео с облетом камерой области необходимо задания маршрута облета и параметров камеры, что является довольно длительной и однообразной операцией. Для облегчения данной задачи была реализована возможность автоматической генерации путей облета и задания параметров камеры. Также была реализована возможность постановки на паузу и возобновления процесса Batch Render отсутствующая в Maya.

# Заключение

В рамках данной работы был решен следующий ряд задач:

- Проведен анализ конкурентов имеющих источники геоданных
- Изучена программа трехмерного моделирования Autodesk Maya и методы взаимодействия с ней.
- Изучены форматы данных OSM, DEM, WRF.
- Реализовано получение информации о пробках путем анализа данных сервиса Яндекс.Карты
- Реализована генерация трехмерных моделей города с наложением данных о источниках загрязнений
- Реализован дополнительный функционал плагина для облегчения процесса рендеринга полученных моделей.

Программа была реализована в виде плагина, встраивающего новый функционал непосредственно в графическую оболочку Autodesk Maya и обеспечивающего легкую работу даже для неопытного пользователя.

*Таблица 3*

## Финальное сравнение с аналогами

	<b>Esri CityEngine</b>	<b>KludgeCity</b>	<b>QTown</b>	<b>Ninja City</b>	<b>ВКР</b>
<b>Совместимость с Autodesk Maya</b>	Частичная	Полная	Полная	Полная	Полная
<b>Цена</b>	Высокая	Бесплатно	Бесплатно	Низкая	Бесплатно
<b>Подход к генерации</b>	Процедурный/ на основе геоданных	Процедурный	Процедурный	Процедурный	На основе геоданных
<b>Работа с открытыми источниками</b>	Ограниченная	Отсутствует	Отсутствует	Отсутствует	Полная
<b>Возможность модификации</b>	Отсутствует	Ограниченная	Ограниченная	Ограниченная	Полная

Как видно из таблицы 3 созданный продукт занимает свою нишу среди конкурентов и соответствует критериям, выдвинутым в начале проекта.

Данный продукт был успешно внедрен в рамках проекта AeroState и в данный момент используется для создания демонстрационных визуализаций в рамках переговоров с инвесторами. Имеется соответствующий акт о внедрении.

## ИСТОЧНИКИ

1. Maya // Autodesk. — 2014. — <http://www.autodesk.com/products/autodesk-maya/overview>.
2. City Engine // Esri.com. — 2014. — <http://www.esri.com/software/cityengine/getting-started>.
3. Kludge City // Kludgeworks. — 2013. — <http://kludgeworks.com/kludgocity/>.
4. QTown 2.0 Release // Brave Rabbit. — 2011. — <http://www.braverabbit.de/playground/?p=538>.
5. Ninja City 4.1.0 // Creative Crash. — 2013. — <http://www.creativecrash.com/maya/script/ninja-city>.
6. OSM XML // Open Street Map Wiki. — 2013. — [http://wiki.openstreetmap.org/wiki/OSM\\_XML](http://wiki.openstreetmap.org/wiki/OSM_XML).
7. Digital elevation model // Wikipedia. — 2014. — [http://en.wikipedia.org/wiki/Digital\\_elevation\\_model](http://en.wikipedia.org/wiki/Digital_elevation_model).
8. The Weather Research & Forecasting Model // wrf-model.org. — 2004. — <http://www.wrf-model.org/index.php>.
9. NetCDF // Wikipedia. — 2014. — <http://en.wikipedia.org/wiki/NetCDF>.
10. MATLAB Code // MathWorks. — 2014. — <http://www.mathworks.com/discovery/matlab-code.html>.
11. MEL Overview // Maya SDK Documentation. — 2014. — [http://download.autodesk.com/global/docs/maya2014/en\\_us/index.html?url=files/Background\\_MEL\\_Overview.htm,topicNumber=d30e788699](http://download.autodesk.com/global/docs/maya2014/en_us/index.html?url=files/Background_MEL_Overview.htm,topicNumber=d30e788699).
12. Maya Python API 2.0 // Maya SDK Documentation. — 2014. — <http://docs.autodesk.com/MAYAUL/2014/ENU/Maya-API-Documentation/index.html?url=files/GUID-AC9B3C66-90FE-493F-A7B2-44ABBA77CED2.htm,topicNumber=d30e35486>.
13. Maya C++ API Reference // Maya SDK Documentation. — 2014. — [http://docs.autodesk.com/MAYAUL/2014/ENU/Maya-API-Documentation/index.html?url=c++\\_ref/index.html,topicNumber=c++\\_ref\\_index\\_html71540f82-cdbd-4011-8ae1-3b5e2a4f502b](http://docs.autodesk.com/MAYAUL/2014/ENU/Maya-API-Documentation/index.html?url=c++_ref/index.html,topicNumber=c++_ref_index_html71540f82-cdbd-4011-8ae1-3b5e2a4f502b).
14. Getting started with .NET API // Maya SDK Documentation. — 2014. — <http://docs.autodesk.com/MAYAUL/2014/ENU/Maya-API-Documentation/index.html?url=files/GUID-C050065C-A7DE-4A46-B035-4B526449DE68.htm,topicNumber=d30e35593>.
15. Geospatial Data Abstraction Library // gdal.org. — 2014. — <http://www.gdal.org/>.
16. netcdf4-python // GitHub. — 2014. — <https://github.com/Unidata/netcdf4-python>.