ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

***«*САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»**

**ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ**

**ВЫСШАЯ ШКОЛА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ И СУПЕРКОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ**

**Отчет о прохождении   
стационарной производственной преддипломной практики**

**на тему:** **«Разработка модуля процедурной генерации трехмерных моделей зданий»**

Ларионова Алексея Сергеевича, гр. 3530203/60101

**Направление подготовки:** 02.03.03 Математическое обеспечение и администрирование информационных систем

|  |
| --- |
| **Место прохождения практики:** СПбПУ, ИКНТ, ВШИСиСТ |

*(указывается наименование профильной организации или наименование структурного подразделения*

|  |
| --- |
|  |

*ФГАОУ ВО «СПбПУ», фактический адрес)*

**Сроки практики:** с 21.04.2020 по 04.05.2020.

**Руководитель практики от ФГАОУ ВО «СПбПУ»:**

|  |
| --- |
| Туральчук К.А., ст. преп. ВШИСиСТ |

*(Ф.И.О., уч. степень, должность)*

**Оценка:**

Руководитель практики

от ФГАОУ ВО «СПбПУ» Туральчук К.А.

Обучающийся Ларионов А.С.

Дата: \_\_\_\_\_\_\_\_\_\_

Оглавление

[ВВЕДЕНИЕ 3](#_Toc39090035)

[РЕАЛИЗАЦИЯ МОДУЛЯ ГЕНЕРАЦИИ 4](#_Toc39090036)

[РЕАЛИЗАЦИЯ ВИЗУАЛИЗАЦИИ МОДЕЛЕЙ 5](#_Toc39090037)

[ЗАКЛЮЧЕНИЕ 8](#_Toc39090038)

ВВЕДЕНИЕ

В рамках выпускной работы бакалавра (ВКР), рассматривается и реализуется модуль для процедурной генерации трехмерных моделей зданий. Процесс процедурной генерации предполагает ввод нескольких конфигурационных параметров, а затем автоматическую генерацию модели. Модель должна быть осмотрена человеком на предмет удовлетворительного качества или структуры. После этого, модель может быть сгенерирована заново с новыми (или корректированными) параметрами, а подходящая модель может быть экспортирована в стандартный формат файла (OBJ или STL).

Целью данной работы является практическая реализация модуля в рамках ВКР, с помощью выбранного набора технологий по теоретическим выкладкам, рассмотренным и разработанным в ходе предшествующей работы над проектом (научно-исследовательские работы и пр.).

Для достижения поставленной цели, необходимо решить следующие задачи:

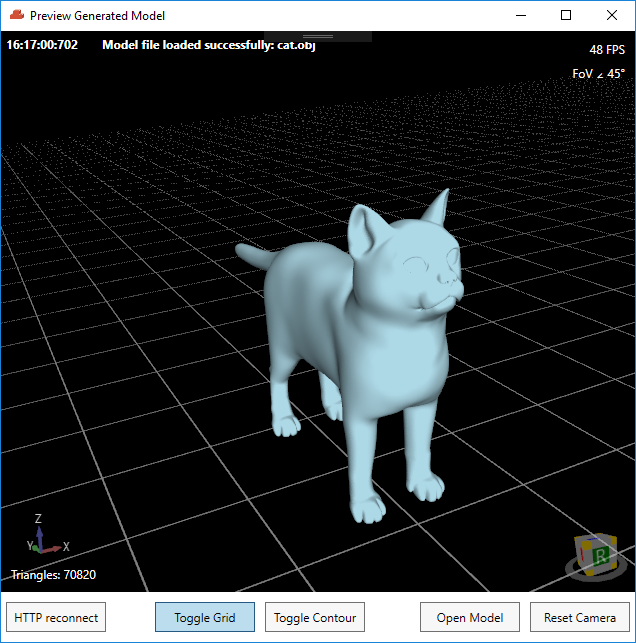
* реализовать общую программную архитектуру модуля и его функционал;
* реализовать графический интерфейс модуля процедурной генерации;
* реализовать дополнительную утилиту для визуализации сгенерированных моделей.

РЕАЛИЗАЦИЯ МОДУЛЯ ГЕНЕРАЦИИ

РЕАЛИЗАЦИЯ ВИЗУАЛИЗАЦИИ МОДЕЛЕЙ

Архитектура модуля предполагает реализацию (или использование существующих) программ, которые бы визуализировали сгенерированную модель. При этом созданы они могут быть на любом языке программирования и в том числе запущены на другом компьютере и такая гибкость обеспечена тем, что модуль процедурной генерации поддерживает отправку готовой модели по распространенному протоколу HTTP.

Для выпускной квалификационной работы был разработан графический интерфейс для операционной системы Windows, который реализован на языке C# (средствами библиотеки WPF) и который использует возможности библиотеки Helix Toolkit для визуализации трехмерных моделей. Данный интерфейс является дополнительным инструментом к разрабатываемому модулю, и в будущем может быть реализован любой другой визуализатор (в т.ч. интерфейс для другой операционной системы или плагин для некоторой системы автоматизированного проектирования).



Интерфейс при запуске открывает HTTP сервис (по адресу *http://localhost:64046/wpfVisualizerService*), используя средства встроенной библиотеки WCF языка C#. Удобство данной библиотеки заключается в возможности представлении команд и данных, передаваемых от HTTP сервиса, в виде класса, реализующего интерфейс сервиса. Таким образом с точки зрения программиста не нужно настраивать отдельные HTTP запросы и заниматься их приемом и пересылкой, а вместо этого просто использовать операции как часть обычного класса из языка программирования. Как только сервис визуализатора запущен, он оправляет запрос еще одному сервису – сервису генератора трехмерных моделей, который непосредственно реализует процедурную генерацию и оправку моделей визуализаторам. Данный сервис находится по постоянному адресу *http://localhost:64046/visualizationControllerService*. Сервис визуализатора регистрируется у сервиса генерации и с этого момента сгенерированные модели будут пересылаться для визуализации. Может быть подключено несколько разных визуализаторов (но они должны находиться на разных HTTP сервисах).

Реализованный графический интерфейс также позволяет открывать и визуализировать модели из файловой системы, перемещаться в трехмерном пространстве для осмотра модели, использовать дополнительные элементы визуализации: отображение каркаса модели, сетка X-Y плоскости.

С точки зрения программного кода, от процесса визуализации отделена DLL библиотека языка С#, определяющая общий функционал визуализатора - таким образом можно реализовать этот интерфейс в другом С# проекте и легко заменить существующий визуализатор, или создать проект на другом языке программирования, но с аналогичным функционалом.

public interface IVisualizer

{

string GetDescription();

void VisualizeModel(Stream model, ModelMetaBase modelMeta,

Stream materialLibrary, Stream[] materialFiles);

void Shutdown();

}

Далее, эксклюзивно для модуля WCF С# (для удобного HTTP взаимодействия) выделена DLL библиотека языка С#, определяющая интерфейс сервиса визуализации, который используется конкретно для WCF общения. По факту, сервис определяет такой же функционал, как и приведенный выше интерфейс, но структурирован он специально для WCF.

[ServiceContract]

public interface IVisualizerService

{

[OperationContract]

string GetDescription();

[OperationContract]

void PrepareForModel(ModelMetaBase modelMetadata);

[OperationContract]

void AcceptMaterialLib(Stream model);

[OperationContract]

void PrepareForMaterialFile(string materialFileId);

[OperationContract]

void AcceptMaterialFile(Stream materialFile);

[OperationContract]

void AcceptModel(Stream model);

[OperationContract]

void Visualize();

[OperationContract]

void Shutdown();

}

Сам визуализатор является WPF приложением, которое в XAML разметке содержит элемент <helix:HelixViewport3D> с конфигурационными параметрами, а также элементы управления (кнопки). Программа реагирует на события интерфейса, а также занимается созданием, регистрацией и закрытием HTTP сервисов. При этом необходимо обрабатывать возможность открытия нескольких экземпляров одного и того же визуализатора – принята схема, что только один экземпляр поддерживает открытие и закрытие сервиса (а соответственно и взаимодействие с процедурным генератором), а остальные в это время не делают никакой HTTP обработки (т.е. фактически могут быть использованы только для загрузки моделей из файловой системы). При закрытии этого управляющего экземпляра, можно будет вручную (нажатием кнопки) подключить один из оставшихся визуализаторов к сервису.

ЗАКЛЮЧЕНИЕ

В ходе преддипломной практики, была проведена работа по практической реализации основной части проекта для выпускной квалификационной работы бакалавра. Основой реализации послужили теоретические выкладки по схеме, архитектуре системы и выбора используемых технологий, которые были рассмотрены в прошлых работах в рамках ВКР.

Был реализованы функциональные подсистемы модуля процедурной генерации трехмерных моделей зданий (с архитектурной точки зрения, а также со стороны взаимодействия с пользователем): генерации модели, экспорта модели в файл и визуализации через внешние приложения. Дополнительно, реализована утилита для визуализации полученных трехмерных моделей. Соответственно обеспечены программные интерфейсы для реализации других визуализаторов для других операционных систем или с другим функционалом и реализован графический пользовательский интерфейс для визуализации в Windows.

По итогу практики, была достигнута цель программной реализации разрабатываемого в ВКР модуля. Были получены знания в разработке пользовательских интерфейсов на языке C# c библиотекой WPF, разработки HTTP сервисов c библиотекой WCF, а также применения библиотек Helix Toolkit и geometry3sharp языка C#.