

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Физико-механический институт

Работа допущена к защите
Руководитель ОП
_____ К.Н. Козлов
« _____ » _____ 2023 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
РАБОТА БАКАЛАВРА**

**ОПТИМИЗАЦИЯ КОЛИЧЕСТВА ВЫЗОВОВ ОТРИСОВКИ В
СОВРЕМЕННЫХ ГРАФИЧЕСКИХ СИСТЕМАХ**

по направлению подготовки 01.03.02 "Прикладная математика и информатика"
Направленность (профиль) 01.03.02_02 "Системное программирование"

Выполнил
студент гр. 5030102/90201

В.А. Парусов

Руководитель
доцент ВШПМиВФ,
степень, звание

С.Ю. Беляев

Консультант
старший преподаватель, ВШПМиВФ

В.С. Чуканов

Санкт-Петербург
2023

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО
Физико-механический институт

УТВЕРЖДАЮ

Руководитель образовательной программы «Прикладная математика и информатика»

К.Н. Козлов

«__» _____ 202__ г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы
студенту Парусову Владимиру Алексеевичу, гр. 5030102/90201

1. Тема работы: «Оптимизация количества вызовов отрисовки в современных графических системах»
2. Срок сдачи студентом законченной работы: июнь 2023 г.
3. Исходные данные по работе:

Инструментальные средства:

- язык программирования C++
- среда разработки Visual Studio 2022
- программная библиотека для работы с видеокартой DirectX 12
- система контроля версий git


Ключевые источники литературы:

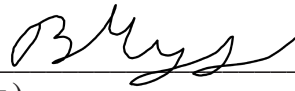
1. Riccio C., Lilley S. Introducing the programmable vertex pulling rendering pipeline //GPU Pro. – 2013. – Т. 4. – С. 21-37
2. Park H., Han J. H. Fast rendering of large crowds using GPU //International Conference on Entertainment Computing. – Springer, Berlin, Heidelberg, 2008. – С. 197-202.

4. Содержание работы (перечень подлежащих разработке вопросов):

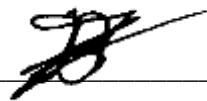
- 1) Введение. Обоснование актуальности
- 2) Постановка задачи
- 3) Обзор существующих решений
- 4) Предлагаемое решение
- 5) Результаты и их сравнительный анализ
- 6) Заключение

5. Дата выдачи задания 07.12.2022

Руководитель ВКР  _____ С.Ю. Беляев
(подпись)

Консультант ВКР  _____ В.С. Чуканов
(подпись)

Задание принял к исполнению

Студент  _____ В.А. Парусов
(подпись)

РЕФЕРАТ

На 23 с., 8 рисунков, 6 таблиц, 2 приложения

КЛЮЧЕВЫЕ СЛОВА: TBD: СТИЛЕВОЕ ОФОРМЛЕНИЕ САЙТА, УПРАВЛЕНИЕ КОНТЕНТОМ, PHP, MYSQL, АРХИТЕКТУРА СИСТЕМЫ.

Тема выпускной квалификационной работы: «Оптимизация количества вызовов отрисовки в современных графических системах»

Данная работа посвящена разработке и реализации архитектуры конвейера отрисовки трехмерной компьютерной графики, снижающей нагрузку на центральный процессор, тем самым увеличивая объем ресурсов компьютера, доступных для разработчиков ПО.

В качестве основной демонстрационной сцены используется совокупность различных трехмерных геометрических объектов, освещённых при помощи подхода, основанного на физической модели микрограней. Весь исходный код проекта написан на языке C++ с применением графической библиотеки DirectX 12. Средством для программирования шейдеров является HLSL.

Наиболее значимым результатом является система для вывода трёхмерных сцен, количество вызовов отрисовки в которой не зависит от числа выводимых объектов. В ходе работы был также разработан гибридный алгоритм отрисовки прозрачных объектов, который не требует их упорядочивания.

Предложенная архитектура пригодна для применения в современных графических приложениях и может быть развита в дальнейшем.

ABSTRACT

23 pages, 8 figures, 6 tables, 2 appendices

KEYWORDS: TBD: STYLE REGISTRATION, CONTENT MANAGEMENT, PHP, MYSQL, SYSTEM ARCHITECTURE.

The subject of the graduate qualification work is «Optimizing the number of draw calls on modern graphics systems».

The work aims to develop and implement graphics pipeline architecture, which will reduce usage of CPU and thereby increase amount of resources available to developers.

Main demonstration scene is a set of different geometric shapes, which uses physically-based lighting with microfacet model. All source code of project is written in C++ using the DirectX 12 graphics library. HLSL is used for shaders code.

The most significant result is a system for rendering three-dimensional scenes, the number of draw calls in which does not depend on the number of rendered objects. In the course of the work, a hybrid algorithm for rendering transparent objects was also developed, which does not require object sorting.

The proposed architecture is suitable for use in modern graphics applications and can be further developed to obtain more efficient results.

СОДЕРЖАНИЕ

РЕФЕРАТ	4
ABSTRACT	5
СОДЕРЖАНИЕ	6
ВВЕДЕНИЕ.....	7
ГЛАВА 1. Постановка задачи.....	9
1.1. Техническое задание	9
1.2. Ожидаемый результат	9
ГЛАВА 2. Обзор существующий решений	9
2.1. Название параграфа	9
2.2. Название параграфа	9
2.2.1. Название подпараграфа	10
2.3. Название параграфа	11
2.4. Выводы	18
ГЛАВА 3. Предлагаемое решение.....	19
3.1. Название параграфа	19
3.2. Название параграфа	19
3.3. Выводы	19
ГЛАВА 4. Результаты и их сравнительный анализ	19
4.1. Название параграфа	19
4.2. Название параграфа	19
4.3. Выводы	19
Заключение	20
Список сокращений и условных обозначений	21
Словарь терминов.....	22
Список использованных источников.....	23
Приложение 1. Краткие инструкции по настройке издательской системы L ^A T _E X.....	24
Приложение 2. Некоторые дополнительные примеры	28

ВВЕДЕНИЕ. ОБОСНОВАНИЕ АКТУАЛЬНОСТИ

В большинстве современных компьютеров установлено и одновременно используются два вида вычислительных процессоров: центральный процессор (CPU) и графический (GPU). Из-за того что они работают одновременно, а также имеют обособленную друг от друга физическую память, возникает необходимость в синхронизации и передаче данных. Эту необходимость решает программа-драйвер, которую реализуют производители графических вычислительных процессоров, предоставляя разработчикам графических приложений специальные API, такие как OpenGL, Vulkan, DirectX 11 и тд. Однако, из-за того что заранее предугадать архитектурные особенности итоговых графических приложений невозможно, программа-драйвер может добавлять дополнительные синхронизации, что приводит к ухудшению производительности, так как центральный и графический процессоры начинают работать последовательно, как представлено на рис.0.1.

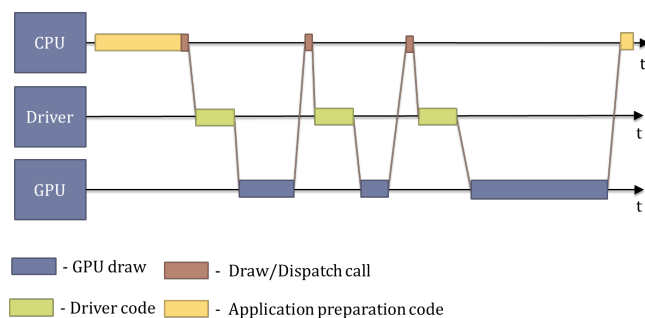


Рис.0.1. Схема работы приложения при неоптимальном драйвере DirectX 11

Во избежание подобных ситуаций, в более поздних и новых API (таких как DirectX 12 и Vulkan) задача синхронизации была снята с программы драйвера и была передана разработчику. Для этого было введено понятие “списка команд”, который разработчик мог заполнять и затем отправлять на исполнение, как показано на рис.0.2.

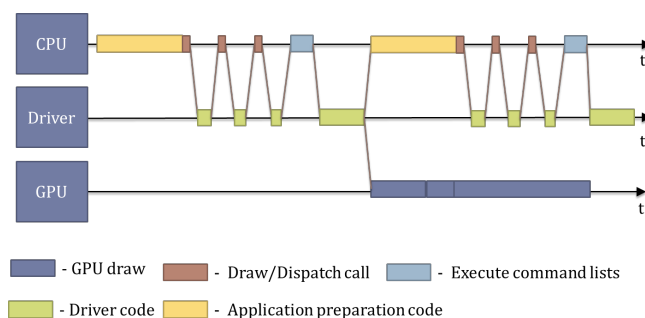


Рис.0.2. Схема работы приложения при DirectX 12

Однако несмотря на то, что данный подход даёт возможность явно управлять синхронизациями между процессорами, часть времени центрального процессора уходит на построение вышеупомянутого “списка”. Причём время построения и передачи будет зависеть от количества выводимых объектов (см. рис.0.3).

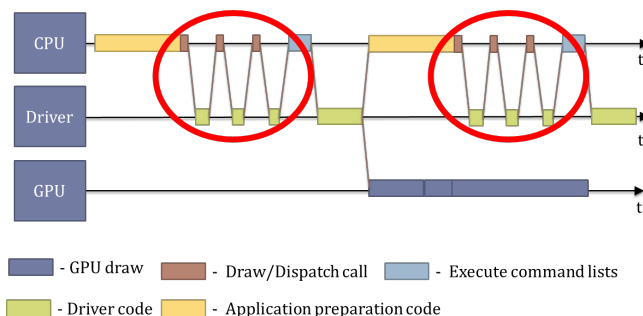


Рис.0.3. Схема работы приложения при DirectX 12, с выделенным временем на построение списка выводимых объектов

Целью работы является построение архитектуры графического приложения таким образом, чтобы время построения и передачи, а также размер “списка команд” не зависели от количества выводимых объектов (см. рис.0.4). Таким образом разработчику ПО будет предоставлено больше времени центрального процессора, а значит более трудоёмкие алгоритмы можно будет использовать для достижения желаемых результатов.

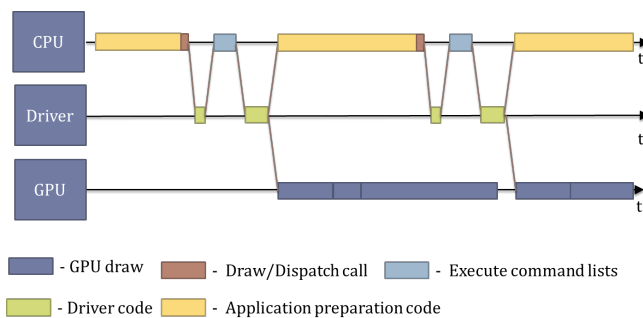


Рис.0.4. Схема работы приложения при DirectX 12 с использованием предлагаемого конвейера

ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ

1.1. Техническое задание

1.2. Ожидаемый результат

ГЛАВА 2. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Глава посвящена более подробным примерам оформления текстово-графических объектов.

В параграфе 2.1 приведены примеры оформления многострочной формулы и одиночного рисунка. Параграф 2.2 раскрывает правила оформления перечислений и псевдокода. В параграфе 2.3 приведены примеры оформления сложносоставных рисунков, длинных таблиц, а также теоремоподобных окружений.

2.1. Название параграфа

Все формулы, размещенные в отдельных строках, подлежат нумерации, например, как формулы (2.1) и (2.2) из [Ganter1999].

$$A^\uparrow = \{m \in M \mid gIm \ \forall g \in A\}; \quad (2.1)$$

$$B^\downarrow = \{g \in G \mid gIm \ \forall m \in B\}. \quad (2.2)$$

Обратим внимание, что формулы содержат знаки препинания и что они выровнены по левому краю (с помощью знака & окружения align).

На рис.2.1 приведёна фотография Нового научно-исследовательского корпуса СПбПУ.

2.2. Название параграфа

Название параграфа оформляется с помощью команды `\section{...}`, название главы — `\chapter{...}`.



Рис.2.1. Новый научно-исследовательский корпус СПбПУ [spbpu-gallery]

2.2.1. Название подпараграфа

Название подпараграфа оформляется с помощью команды `\subsection{...}`.

Использование подподпараграфов в основной части крайне не рекомендуется. В случае использования, необходимо вынести данный номер в содержание. Название подпараграфа оформляется с помощью команды `\subsubsection{...}`.

Вместо подподпараграфов рекомендовано использовать перечисления.

Перечисления могут быть с нумерационной частью и без неё и использоваться с иерархией и без иерархии. Нумерационная часть при этом формируется следующим способом:

1. в перечислениях *без иерархии* оформляется арабскими цифрами с точкой (или длинным тире).
2. В перечислениях *с иерархией* — в последовательности сначала прописных латинских букв с точкой, затем арабских цифр с точкой и далее — строчных латинских букв со скобкой.

Далее приведён пример перечислений с иерархией.

- A. Первый пункт.
- B. Второй пункт.
- C. Третий пункт.
- D. По ГОСТ 2.105–95 [**gost-russian-text-documents**] первый уровень нумерации идёт буквами русского или латинского алфавитов (*для определенности выбираем английский алфавит*), а второй — цифрами.

1. В данном пункте лежит следующий нумерованный список:

- a) первый пункт;
- b) третий уровень нумерации не нормирован ГОСТ 2.105–95 (*для определенности выбираем английский алфавит*);

- с) обращаем внимание на строчность букв в этом нумерованном и следующем маркированном списке:
- первый пункт маркированного списка.

Е. Пятый пункт верхнего уровня перечисления.

Маркированный список (без нумерационной части) используется, если нет необходимости ссылки на определенное положение в списке:

- первый пункт с *маленькой буквы* по правилам русского языка;
- второй пункт с *маленькой буквы* по правилам русского языка.

Оформление псевдокода необходимо осуществлять с помощью пакета `algorithm2e` в окружении `algorithm`. Данное окружение интерпретируется в шаблоне как рисунок. Пример оформления псевдокода алгоритма приведён на рис.2.2.

Обратим внимание, что можно сослаться на строку 1 псевдокода из рис.2.2.

2.3. Название параграфа

Одиночные формулы также, как и отдельные формулы в составе группы, могут быть размещены в несколько строк. Чтобы выставить номер формулы напротив средней строки, используйте окружение `multlined` из пакета `mathtools` следующим образом [Ganter1999]:

$$\begin{aligned}
 (A_1, B_1) &\leq (A_2, B_2) \Leftrightarrow \\
 &\Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow \\
 &\Leftrightarrow B_2 \subseteq B_1.
 \end{aligned}
 \tag{2.3}$$

Используя команду `\labelcref{...}` из пакета `cleveref`, допустимо оформить ссылку на несколько формул, например, (2.1–2.3).

Пример оформления четырёх иллюстраций в одном текстово-графическом объекте приведён на рис.2.3. Это возможно благодаря использованию пакета `subcaption`.

Далее можно сослаться на составные части данного рисунка как на самостоятельные объекты: рис.2.3а, рис.2.3b, рис.2.3с, рис.2.3d или на три из четырёх изображений одновременно: рис.2.3а–2.3с.

Приведём пример табличного представления данных с записью продолжения на следующей странице на табл.2.1.

Algorithm

Input: the many-valued context $\mathbb{M} \stackrel{\text{def}}{=} (G, M, W, J)$, the class membership $\varepsilon : G \rightarrow K$

Output: positive and negative binary contexts $\overline{\mathbb{K}}_+ \stackrel{\text{def}}{=} (\overline{G}_+, M, I_+)$, $\overline{\mathbb{K}}_- \stackrel{\text{def}}{=} (\overline{G}_-, M, I_-)$ such that i-tests found in $\overline{\mathbb{K}}_+$ are diagnostic tests in \mathbb{M} , and objects from $\overline{\mathbb{K}}_-$ are counter-examples

1. **for** $\forall g_i, g_j \in G$ **do**
2. **if** $i < j$ **then**
3. $\overline{G} \leftarrow (g_i, g_j);$
4. **for** $\forall (g_i, g_j) \in \overline{G}$ **do**
5. **if** $m(g_i) = m(g_j)$ **then**
6. $(g_i, g_j)Im;$
7. **if** $\varepsilon(g_i) = \varepsilon(g_j)$ **then**
8. $\overline{G}_+ \leftarrow (g_i, g_j);$
9. **else** $\overline{G}_- \leftarrow (g_i, g_j);$
10. $I_+ = I \cap (\overline{G}_+ \times M), I_- = I \cap (\overline{G}_- \times M);$
11. **for** $\forall \overline{g}_+ \in \overline{G}_+, \forall \overline{g}_- \in \overline{G}_-$ **do**
12. **if** $\overline{g}_+^\uparrow \subseteq \overline{g}_-^\uparrow$ **then**
13. $\overline{G}_+ \leftarrow \overline{G}_+ \setminus \overline{g}_+;$

Рис.2.2. Псевдокод алгоритма DiagnosticTestsScalingAndInferring
[Naidenova2017]

Таблица 2.1

Пример задания данных из [Peskov2004] (с повтором для переноса таблицы на новую страницу)

G	m_1	m_2	m_3	m_4	K
1	2	3	4	5	6
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1

Продолжение табл.2.1

1	2	3	4	5	6
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

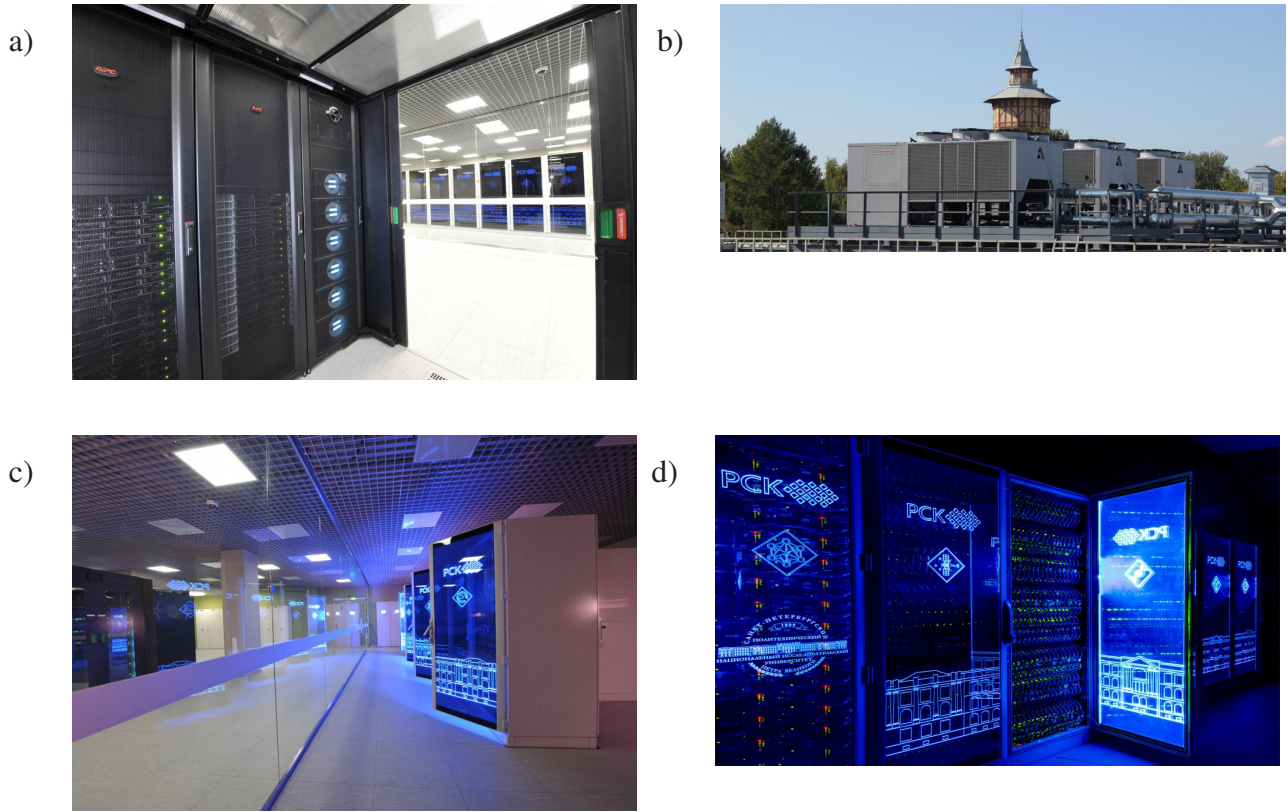


Рис.2.3. Фотографии суперкомпьютерного центра СПбПУ [spbpu-gallery]: *a* — система хранения данных и узлы NUMA-вычислителя; *b* — холодильные машины на крыше научно-исследовательского корпуса; *c* — машинный зал; *d* — элементы вычислительных устройств

Таблица 2.2

Пример представления данных для сквозного примера по ВКР [Peskov2004]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

Таблица 2.3

Пример задания данных в табличном виде из [Peskov2004] (с помощью окружения minipage)

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2



Рис.2.4. Новый научно-исследовательский корпус СПбПУ [spbpu-gallery] (с помощью окружения minipage)

Вопросы форматирования текстово-графических объектов (окружений) не регламентированы в известных нам ГОСТах, поэтому предлагаем придерживаться следующих правил:

- **полужирный текст** рекомендуем использовать только для названий стандартных окружений с нумерационной частью, например, для представления *впервые*: **определение 1.1, теорема 2.2, пример 2.3, лемма 4.5**;
- *курсив* рекомендуем использовать только для выделения переменных в формулах, служебной информации об авторах главы (статьи), важных терминов, представляемых по тексту, а также для всего тела окружений, связанных с получением *новых существенных результатов и их доказательством*: теорема, лемма, следствие, утверждение и другие.

По аналогии с нумерацией формул, рисунков и таблиц нумеруются и иные текстово-графические объекты, то есть включаем в нумерацию номер главы, например: теорема 3.1. для первой теоремы третьей главы монографии. Команды \LaTeX выставляют нумерацию и форматирование автоматически. Полный перечень команд для подготовки текстово-графических и иных объектов находится в подробных методических рекомендациях [spbpu-bci-template-author-guide].

Для удобства авторов названия стандартных окружений, рекомендованных к использованию, приведены в табл.2.4, а в табл.2.5 перечислены имена специально разработанных окружений для шаблонов SPbPU.

На базе пакета `tikz` разработано большое количество расширений [ctan-tikz], например, `tikzcd`, которые мы рекомендуем использовать для оформления иллюстраций.

Таблица 2.4

Стандартные окружения

Название окружения	Назначение
<code>center</code>	центрирование, аналог команды <code>\centering</code> , но с добавлением нежелательного пробела, поэтому лучше избегать применения <code>center</code>
<code>itemize</code>	перечисления, в которых нет необходимости нумеровать пункты (немаркированные списки)
<code>enumerate</code>	перечисления с нумерацией (немаркированные списки)
<code>refsection</code>	создание отдельных библиографических списков для глав
<code>tabular</code>	оформление таблиц
<code>table</code>	автоматическое перемещение по тексту таблиц, оформленных, например, с помощью <code>tabular</code> , для минимизации пустых пространств
<code>longtable</code>	оформление многостраничных таблиц
<code>tikzpicture</code>	создание иллюстраций с помощью пакета <code>tikz</code> [ctan-tikz]
<code>figure</code>	автоматическое перемещение по тексту рисунков, оформленных например, с помощью <code>tikz</code> или подключенных с помощью команды <code>\includegraphics</code> , для минимизации пустых пространств
<code>subfigure</code>	оформление вложенных рисунков в составе <code>figure</code>
<code>algorithm</code>	оформление псевдокода на основе пакета <code>algorithm2e</code> [ctan-algorithm2e]
<code>minipage</code>	оформление рисунков и таблиц без функций автоматического перемещения по тексту для минимизации пустых пространств
<code>equation</code>	оформление выключенных (не встроенных в текст с помощью <code>\$. . . \$</code>) одиночных формул на одной строке
<code>multilined</code>	оформление выключенных (не встроенных в текст с помощью <code>\$. . . \$</code>) одиночных формул в несколько строк
<code>aligned</code>	оформление нескольких формул с выравниванием по символу <code>&</code> .

В случае, если авторам потребовалось новое окружение, то создать его можно в файле `my_folder/my_settings.tex` согласно правилам, приведённым ниже.

- Для перехода в режим создания окружений следует указать:
 - `\theoremstyle{myplain}` — окружения с доказательствами или аксиомами
 - `\theoremstyle{mydefinition}` — окружения, не связанные с доказательствами или аксиомами.
- В команде создания окружения следует ввести краткий псевдоним (`m-new-env`) и отображаемое в pdf имя окружения (Название_окружения):
 - `\newtheorem{m-new-env-second}{Название_окружения}[chapter]`.

Таблица 2.5

Специальные окружения

Название окружения	Текстово-графический объект
abstr	реферат (abstract)
m-theorem	теорема
m-corollary	следствие
m-proposition	утверждение
m-lemma	лемма
m-axiom	аксиома
m-example	пример
m-definition	определение
m-condition	условие
m-problem	проблема
m-exercise	упражнение
m-question	вопрос
m-hypothesis	гипотеза

Теорема 2.1 (о чем-то конкретном). *Текст теоремы полностью выделен курсивом. Допустимо математические символы не выделять курсивом, если это искажает их значения. Используется абзацный отступ, так как “Абзацы в тексте начинают отступом” в соответствии с ГОСТ 2.105–95. Название теоремы допустимо убрать. Доказательство окончено.*

Доказательство теоремы 2.1, леммы, утверждений, следствий и других подобных окружений (в последнем абзаце) завершаем предложением в котором сказано, что доказательство окончено. Например, доказательство теоремы 2.1 окончено.

Тело доказательства не выделяется курсивом. Тело следующих окружений также не выделяется сплошным курсивом: определение, условие, проблема, пример, упражнение, вопрос, гипотеза и другие.

Определение 2.1 (термин). В тексте определения только *важные термины* выделяются курсивом. Если определение носит лишь вспомогательный характер, то допустимо не использовать окружение m-definition, представляя текст определения в обычном абзаце. Ключевые термины при этом обязательно выделяются курсивом.

Вместо теоремо-подобных окружений для вставки небольших текстово-графических объектов иногда используются команды. Типичным примером такого

подхода является команда `\footnote{text}`¹, где в аргументе `text` указывают текст *подстрочной ссылки (сноски)*. В них *нельзя добавлять веб-ссылки или цитировать литературу*. Для этих целей используется список литературы. Нумерация сносок сквозная по ВКР без точки на конце выставляется в шаблоне автоматически, однако в каждом приложении к ВКР нумерация, зависящая от номера приложения, выставляется префикс «П», например «П1.1» — первая сноска первого приложения.

2.4. Выводы

Текст заключения ко второй главе. Пример ссылок [**Article; Book; Booklet; Conference; Inbook; Incollection; Manual; Mastersthesis; Misc; Phdthesis; Proceedings; Techreport; Unpublished; badiou:briefings**], а также ссылок с указанием страниц, на котором отображены те или иные текстово-графические объекты [**Naidenova2017**] или в виде мультицитаты на несколько источников [**Naidenova2017; Ganter1999**]. Часть библиографических записей носит иллюстративный характер и не имеет отношения к реальной литературе.

Короткое имя каждого библиографического источника содержится в специальном файле `my_biblio.bib`, расположенном в папке `my_folder`. Там же находятся исходные данные, которые с помощью программы `Viber` и стилевого файла `Biblatex-GOST` [**ctan-biblatex-gost**] приведены в списке использованных источников согласно ГОСТ 7.0.5-2008. Многообразные реальные примеры исходных библиографических данных можно посмотреть по ссылке [**ctan-biblatex-gost-examples**].

Как правило, ВКР должна состоять из четырех глав. Оставшиеся главы можно создать по образцу первых двух и подключить с помощью команды `\input` к исходному коду ВКР. Далее в приложении 1 приведены краткие инструкции запуска исходного кода ВКР [**latex-miktex; latex-texstudio**].

В приложении 2 приведено подключение некоторых текстово-графических объектов. Они оформляются по приведенным ранее правилам. В качестве номера структурного элемента вместо номера главы используется «П» с номером главы. Текстово-графические объекты из приложений не учитываются в реферате.

¹Внимание! Команда вставляется непосредственно после слова, куда вставляется сноска (без пробела). Лишние пробелы также не указываются внутри команды перед и после фигурных скобок.

ГЛАВА 3. ПРЕДЛАГАЕМОЕ РЕШЕНИЕ

Хорошим стилем является наличие введения к главе. Во введении может быть описана цель написания главы, а также приведена краткая структура главы.

3.1. Название параграфа

3.2. Название параграфа

3.3. Выводы

Текст выводов по главе 3.

ГЛАВА 4. РЕЗУЛЬТАТЫ И ИХ СРАВНИТЕЛЬНЫЙ АНАЛИЗ

Хорошим стилем является наличие введения к главе. Во введении может быть описана цель написания главы, а также приведена краткая структура главы.

4.1. Название параграфа

4.2. Название параграфа

Пример ссылки на литературу [avtonomova:fya; Peskov2004-ru; Kotelnikov2004-ru; Kotelnikov2004].

4.3. Выводы

Текст выводов по главе 4.

ЗАКЛЮЧЕНИЕ

Заключение (2 – 5 страниц) обязательно содержит выводы по теме работы, *конкретные предложения и рекомендации* по исследуемым вопросам. Количество общих выводов должно вытекать из количества задач, сформулированных во введении выпускной квалификационной работы.

Предложения и рекомендации должны быть органически увязаны с выводами и направлены на улучшение функционирования исследуемого объекта. При разработке предложений и рекомендаций обращается внимание на их обоснованность, реальность и практическую приемлемость.

Заключение не должно содержать новой информации, положений, выводов и т. д., которые до этого не рассматривались в выпускной квалификационной работе. Рекомендуются писать заключение в виде тезисов.

Последним абзацем в заключении можно выразить благодарность всем людям, которые помогали автору в написании ВКР.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

DOI Digital Object Identifier.

WoS Web of Science.

ВКР Выпускная квалификационная работа.

ТГ-объект Текстово-графический объект.

СЛОВАРЬ ТЕРМИНОВ

TeX — язык вёрстки текста и издательская система, разработанные Дональдом Кнутом.

LaTeX — язык вёрстки текста и издательская система, разработанные Лэсли Лампортом как надстройка над TeX.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Приложение 1

Краткие инструкции по настройке издательской системы L^AT_EX

В SPbPU-BCI-template автоматически выставляются необходимые настройки и в исходном тексте шаблона приведены примеры оформления текстово-графических объектов, поэтому авторам достаточно заполнить имеющийся шаблон текстом главы (статьи), не вдаваясь в детали оформления, описанные далее. Возможный «быстрый старт» оформления главы (статьи) под Windows следующий^{П1.1}:

- A. Установка полной версии MikTeX [latex-miktex]. В процессе установки лучше выставить параметр доустановки пакетов «на лету».
- B. Установка TexStudio [latex-texstudio].
- C. Запуск TexStudio и компиляция my_chapter.tex с помощью команды «Build&View» (например, с помощью двойной зелёной стрелки в верхней панели). Иногда, для достижения нужного результата необходимо несколько раз скомпилировать документ.
- D. В случае, если не отобразилась библиография, можно
 - воспользоваться командой Tools → Commands → Biber, затем запустив Build&View;
 - настроить автоматическое включение библиографии в настройках Options → Configure TexStudio → Build → Build&View (оставить по умолчанию, если сборка происходит слишком долго): txs:///pdflatex | txs:///biber | txs:///pdflatex | txs:///pdflatex | txs:///view-pdf.

В случае возникновения ошибок, попробуйте скомпилировать документ до последних действий или внимательно ознакомьтесь с описанием проблемы в log-файле. Бывает полезным переход (по подсказке TexStudio) в нужную строку в pdf-файле или запрос с текстом ошибки в поисковиках. Наиболее вероятной проблемой при первой компиляции может быть отсутствие какого-либо установленного пакета L^AT_EX.

В случае корректной работы настройки «установка на лету» все дополнительные пакеты будут скачиваться и устанавливаться в автоматическом режиме. Если доустановка пакетов осуществляется медленно (несколько пакетов за один запуск

^{П1.1} Вниманию! Пример оформления подстрочной ссылки (сноски).

компилятора), то можно попробовать установить их в ручном режиме следующим образом:

1. Запустите программу: меню → все программы → MikTeX → Maintenance (Admin) → MiKTeX Package Manager (Admin).
2. Пользуясь поиском, убедитесь, что нужный пакет присутствует, но не установлен (если пакет отсутствует воспользуйтесь сначала MiKTeX Update (Admin)).
3. Выделив строку с пакетом (возможно выбрать несколько или вообще все неустановленные пакеты), выполните установку Tools → Install или с помощью контекстного меню.
4. После завершения установки запустите программу MiKTeX Settings (Admin).
5. Обновите базу данных имен файлов Refresh FNDB.

Для проверки текста статьи на русском языке полезно также воспользоваться настройками Options → Configure TexStudio → Language Checking → Default Language. Если русский язык «ru_RU» не будет доступен в меню выбора, то необходимо вначале выполнить Import Dictionary, скачав из интернета любой русскоязычный словарь.

Далее приведены формулы (П1.2), (П1.1), рис.П1.2, рис.П1.1, табл.П1.2, табл.П1.1.

$$\pi \approx 3,141. \quad (\text{П1.1})$$



Рис.П1.1. Вид на гидробашню СПбПУ [spbpu-gallery]

Представление данных для сквозного примера по ВКР [Peskov2004]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

П1.1. Параграф приложения

П1.1.1. Название подпараграфа

Название подпараграфа оформляется с помощью команды `\subsection{...}`.
Использование подподпараграфов в основной части крайне не рекомендуется.

П1.1.1.1. Название подподпараграфа

$$\pi \approx 3,141. \quad (\text{П1.2})$$



Рис.П1.2. Вид на гидробашню СПбПУ [spbpu-gallery]

Таблица П1.2

Представление данных для сквозного примера по ВКР [Peskov2004]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

Приложение 2

Некоторые дополнительные примеры

В приложении^{П2.1} приведены формулы (П2.2), (П2.1), рис.П2.2, рис.П2.1, табл.П2.2, табл.П2.1

$$\pi \approx 3,141.$$

(П2.1)



Рис.П2.1. Вид на гидробашню СПбПУ [spbpu-gallery]

Таблица П2.1

Представление данных для сквозного примера по ВКР [Peskov2004]

<i>G</i>	<i>m</i> ₁	<i>m</i> ₂	<i>m</i> ₃	<i>m</i> ₄	<i>K</i>
<i>g</i> ₁	0	1	1	0	1
<i>g</i> ₂	1	2	0	1	1
<i>g</i> ₃	0	1	0	1	1
<i>g</i> ₄	1	2	1	0	2
<i>g</i> ₅	1	1	0	1	2
<i>g</i> ₆	1	1	1	2	2

^{П2.1}Внимание! Пример оформления подстрочной ссылки (сноски).

П2.1. Подраздел приложения

$$\pi \approx 3,141. \quad (\text{П2.2})$$



Рис.П2.2. Вид на гидробашню СПбПУ [spbpu-gallery]

Таблица П2.2

Представление данных для сквозного примера по ВКР [Peskov2004]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2