Областное государственное бюджетное профессиональное образовательное учреждение Южский технологический колледж

РАССМОТРЕН
методической комиссией № 2
Протокол № 1
от « 31 » 08 2023г.
Председатель
_____ С.В. Верховцева

УТВЕРЖДАЮ:
Зам.директора по УПР
_____ Е.И.Акопян

КОМПЛЕКТ ПРАКТИЧЕСКИХ ЗАНЯТИЙ
по учебной дисциплине
МДК 02.01 Технология выполнения подготовки интерфейсной графики
для обучающихся по специальности 09.01.03
Оператор информационных систем и ресурсов

Разработчик
А.Э Работин,
преподаватель

Южа – 2023 г.

Практические занятия по общеобразовательному учебному предмету **02.01 Технология выполнения подготовки интерфейсной графики** разработаны в соответствии с Федеральным государственным образовательным стандартом (далее–ФГОС) по профессии 09.01.03 «Оператор информационных систем и ресурсов» и предназначены для студентов очной формы обучения.

Объем практических работ по дисциплине **02.01 Технология выполнения подготовки интерфейсной графики** 78 часов.

Перечень практических работ соответствует содержанию программы **02.01 Технология выполнения подготовки интерфейсной графики**

2.1. Цель практических занятий по дисциплине 02.01 «Технология выполнения подготовки интерфейсной графики» — формирование навыков создания визуального дизайна элементов графического пользовательского интерфейса и подготовки графических материалов для включения в интерфейс.

Это направлено на освоение профессиональных компетенций:

ПК 2.1 «Создавать визуальный дизайн элементов графического пользовательского интерфейса»

ПК 2.2 «Подготавливать графические материалы для включения в графический пользовательский интерфейс»

2.2. Задачи практических занятий:

- **Разработка графического пользовательского интерфейса** в целом или отдельных элементов управления по определённому визуальному стилю
- **Создание раскадровок анимации** интерфейсных объектов.
- **Рисование пиктограмм**, включая разработку метафор.
- **Рисование графических подсказок** и другой интерфейсной графики.
- **Подготовка графических материалов** для включения в вёрстку или программный код в требуемых разрешениях.
- **Оптимизация интерфейсной графики** под различные разрешения экрана.
- **Подбор технических параметров** интерфейсной графики для заданного стиля и требований к графическому пользовательскому интерфейсу
- **Оценка совокупности графических элементов** оформления интерфейса на соответствие техническим требованиям.

При выполнении каждой практической работы обучающиеся знакомятся с целями и задачами работы.

Работа состоит из двух частей: теоретической (для ознакомления и опоры при выполнении задания) и практической, которая содержит несколько заданий (письменные ответы на вопросы.

Обязательным требованием к выполнению практической работы является вывод, который выполняется письменно.

В конце каждой практической работы содержатся критерии оценки, с которыми обучающиеся так же знакомятся.

Перечень практических работ

№ занятия	Тема практического занятия	Кол-во часов
1.	Практическая работа № 1 - 6. Сравнительный анализ и выбор визуального решения интерфейса	6
2.	Практическая работа №7 - 12. Обеспечение соответствия интерфейсной графики техническим требованиям	6
3.	Практическая работа №13 - 18. Подготовка растровых изображений для интерфейса	6
4.	Лабораторная работа № 1 - 6. Создание визуального оформления интерфейса с элементами растровой графики	6
5.	Практическая работа №19 - 24 . Создание векторных рисунков с помощью кривых	6
6.	Практическая работа №25 - 30. Подготовка векторных изображений для интерфейса	6
7.	Лабораторная работа №7 - 12. Создание визуального оформления интерфейса с элементами векторной графики	6
8.	Практическая работа №31 - 36. Подготовка анимированных элементов интерфейса	6
9.	Практическая работа №37 - 48. Черновой монтаж и послойная анимация	6
10.	Практическая работа №49 - 54. Создание анимационного логотипа	6
11.	Практическая работа №55 - 60. Создание анимационного проекта	6
12.	Лабораторная работа №13 - 18. Анимирование элементов интерфейса	6

Практическая работа №1 - 6. Сравнительный анализ и выбор визуального решения интерфейса

Структура практической работы

Рекомендуется оформить работу в виде отчета со следующими разделами:

1. Титульный лист

- Название учебного заведения, предмета, практической работы, ФИО студента, группа, ФИО преподавателя.

2. Введение

- **Цель работы:** Научиться проводить сравнительный анализ интерфейсов и обосновывать выбор оптимального визуального решения на основе конкретных критериев.
- **Задачи:**
  - Определить предмет анализа (какой элемент интерфейса или экран анализируем).
  - Выбрать 2-3 сравниваемых варианта (прототипа, макета, скриншота существующих приложений).
  - Определить критерии для сравнения.
  - Провести анализ по каждому критерию.
  - Сделать вывод и выбрать лучший вариант.
- **Объект исследования:** Например, "Интерфейс главного экрана мобильного банкинга", "Форма оформления заказа в интернет-магазине".

3. Постановка задачи и контекст

- **Задача пользователя:** Какую цель пользователь пытается достичь с помощью этого интерфейса? (Например: "Быстро перевести деньги знакомому", "Найти и забронировать отель по заданным параметрам").
- **Целевая аудитория:** Кто основной пользователь? (Например: "Молодые люди 18-25 лет, активно пользующиеся смартфонами", "Бухгалтеры, работающие в десктопной программе").
- **Платформа:** Мобильное приложение (iOS/Android), веб-сайт, десктопная программа.

4. Варианты для сравнения

- Представьте 2-3 скриншота или макета интерфейсов, которые вы будете сравнивать.
- **Обязательно подпишите их:** "Вариант А", "Вариант Б" и т.д.
- Дайте краткое описание каждого варианта (1-2 предложения). Например:
  - *Вариант А: "Минималистичный дизайн с акцентом на иллюстрации."*
  - *Вариант Б: "Информационно-насыщенный интерфейс в скевоморфном стиле."*

5. Критерии для сравнения

Это **самый важный раздел**. Критерии должны быть конкретными и измеримыми.

Критерий	Описание критерия (что именно мы оцениваем?)
1. Визуальная иерархия	Насколько легко понять, какой элемент главный, а какой второстепенный? Куда падает взгляд в первую очередь?
2. Удобство навигации	Легко ли пользователю найти нужную функцию или перейти в нужный раздел?
3. Читаемость	Насколько легко воспринимается текст? (Шрифты, размер, контраст, межстрочный интервал).
4. Соответствие принципам UI/UX	Используются ли общепринятые паттерны (например, iOS HIG, Material Design)? Интуитивно ли понятен интерфейс?
5. Эстетика и эмоциональный отклик	Нравится ли внешний вид? Какие эмоции вызывает (доверие, спокойствие, энергичность)?
6. Доступность	Учтены ли потребности пользователей с ограниченными возможностями (цветовой контраст, размер кликабельных зон)?
7. Функциональность	Все ли необходимые функции представлены на экране?

	Нет ли перегруженности?

Вы можете добавлять или изменять критерии в зависимости от конкретного задания.

6. Сравнительный анализ

Оформите эту часть в виде таблицы. Это наглядно и структурированно.

Критерий	Вариант А	Вариант Б	Сравнительный комментарий
Визуальная иерархия	https://placehold.co/100x100/EEE/AAA	https://placehold.co/100x100/EEE/AAA	**А:** Четкая иерархия, кнопка призыва к действию (CTA) выделена цветом. **Б:** Все элементы визуально равнозначны, взгляд "разбегается". **Вывод:** Вариант А лучше.
Удобство навигации	Меню в нижней части экрана, стандартно для iOS.	Меню скрыто за боковой шторкой (гамбургер-меню).	**А:** Быстрый доступ к основным разделам. **Б:** Ключевые функции скрыты. **Вывод:** Вариант А удобнее.
Читаемость	Используется крупный, sans-serif шрифт, высокий контраст.	Используется мелкий, декоративный шрифт, низкая контрастность.	**А:** Текст легко читается. **Б:** Текст читается с трудом. **Вывод:** Вариант А предпочтительнее.
... и т.д.	...	...	...

7. Выводы и рекомендации

- **Сводная таблица результатов:** Можно использовать простую таблицу с итоговой оценкой по каждому критерию (например, по 5-балльной шкале).
- **Выбор лучшего варианта:** На основе анализа четко сформулируйте, какой вариант и почему является победителем. ("По результатам анализа, Вариант А является наиболее предпочтительным, так как он превосходит Вариант Б по 5 из 7 критериев, включая ключевые: визуальная иерархия и удобство навигации").
- **Рекомендации по улучшению:** Даже для лучшего варианта можно предложить улучшения. ("Для Варианта А рекомендуется увеличить контрастность для неактивных кнопок, чтобы улучшить доступность").

Пример анализа (фрагмент)

Задача: Проанализировать интерфейс экрана входа в мобильное приложение.

Вариант А: Классическая форма с полями "Email" и "Пароль", кнопка "Войти" и ссылки "Забыли пароль?" и "Регистрация".

Вариант Б: Форма, где пароль по умолчанию скрыт, есть кнопка "Показать пароль", а также кнопки для входа через социальные сети.

Критерий	Вариант А	Вариант Б	Сравнительный комментарий
Простота и скорость	Минимальное количество полей, пользователь может быстро ввести данные.	Наличие дополнительных кнопок соцсетей может отвлекать от основной цели.	**А:** Быстрее для постоянного пользователя. **Б:** Универсальнее, но требует больше когнитивной нагрузки. **Вывод:** А лучше для скорости.
Безопасность	Пароль	Пароль скрыт по	**А:** Небезопасно в

	отображается открыто.	умолчанию, есть возможность его показать.	публичных местах. **Б:** Баланс безопасности и удобства. **Вывод:** Б безопаснее.
Соответствие принципам	Соответствует, но устаревший подход к отображению пароля.	Соответствует современным рекомендациям (скрытый пароль, опция "показать").	**Вывод:** Б более современен.

Итоговый вывод: Несмотря на то, что Вариант А проще, Вариант Б является более сбалансированным решением, так как он соответствует современным стандартам безопасности, не сильно теряя в простоте. Рекомендуется к реализации **Вариант Б**.

Советы для успешного выполнения работы

1. **Будьте объективны.** Старайтесь избегать формулировок "мне нравится/не нравится". Вместо этого используйте "интерфейс вызывает доверие за счет использования синего корпоративного цвета" или "красный цвет кнопки создает ощущение срочности".
2. **Ссылайтесь на авторитетные источники.** Если вы знаете принципы из Material Design или iOS HIG, ссылайтесь на них. ("В Варианте Б используется плавающая кнопка действия (FAB), что соответствует принципам Material Design").
3. **Используйте скриншоты.** Один скриншот стоит тысячи слов. Вставляйте их прямо в таблицу анализа.
4. **Проверьте на доступность.** Это очень важный и часто упускаемый критерий. Можно использовать простые онлайн-инструменты для проверки цветового контраста.

Практическая работа №7 - 12. Обеспечение соответствия интерфейсной графики техническим требованиям

Структура практической работы

Рекомендуется оформить работу в виде технического отчета.

1. Титульный лист

- Название учебного заведения, предмета, номера работы, ФИО студента, группы, ФИО преподавателя.

2. Введение

- **Цель работы:** Научиться проверять, адаптировать и подготавливать интерфейсную графику к передаче в разработку в соответствии с заданными техническими требованиями (ТТ).
- **Задачи:**
  - Изучить предоставленные технические требования.
  - Выбрать инструменты для проверки и подготовки графики.
  - Провести аудит предоставленных графических материалов (макетов, ресурсов).
  - Выявить и классифицировать нарушения ТТ.
  - Подготовить графику, исправив выявленные нарушения.
  - Сформировать отчет и/или пакет ресурсов для разработчика.
- **Объект исследования:** Набор макетов (например, из Figma) для определенного экрана/блока приложения или веб-сайта.

3. Исходные данные и технические требования

В этом разделе вы описываете "входные данные" работы.

- **Источник макетов:** Ссылка на файл в Figma, Adobe XD, Sketch и т.д., или папка с PNG/PSD-файлами.
- **Список технических требований (ТТ):** Это основа всей работы. Требования могут быть следующими:
  - **Платформа:** iOS (требования Apple HIG), Android (требования Material Design), Web.
  - **Разрешения экранов:** @1x, @2x, @3x для мобильных устройств; стандартные разрешения для десктопа.
  - **Форматы файлов:** PNG (для растровой графики), SVG (для векторной графики), WEBP (для веба).
  - **Цветовые пространства:** sRGB для веба и мобильных приложений.
  - **Размеры файлов:** Максимально допустимый вес для изображений (например, "не более 100 КБ на ресурс").
  - **Нейминг (именование) файлов:** Правила именования (например, icon_tab_profile_active@2x.png, только латиница, нижние подчеркивания).
  - **Размеры и сетка:** Соответствие 8pt grid system, четные размеры (например, 24x24, 32x32).
  - **Требования к экспорту:** Наличие слоев-маркеров для экспорта (/ или > в Figma), корректные настройки экспорта.

4. Инструменты для проверки

Опишите, какими инструментами вы пользовались для проверки.

- **Для анализа макетов:** Figma (встроенные инструменты проверки стилей, плагины), Adobe XD.
- **Для проверки экспортированных ресурсов:**
  - *Просмотр метаданных:* Photoshop, Finder/Проводник Windows.
  - *Проверка размера файла:* Стандартный файловый менеджер.
  - *Проверка цветового профиля:* Плагины для Figma, или специальные программы (например, ImageMagick).
  - *Валидация SVG:* Онлайн-сервисы (например, SVG Validator).
  - *Проверка контрастности цветов:* Плагины в Figma (Stark, Able), онлайн-калькуляторы.

5. Аудит интерфейсной графики (Практическая часть)

Это ядро вашей работы. Оформите в виде таблицы с результатами проверки.

№	Проверяемый элемент /	Метод проверки	Результат	Статус

	Требование			
1	**Именование ресурсов**	Проверка названий слоев для экспорта в Figma.	Элемент ic_heart назван корректно. Элемент Изображение 123 требует переименования.	Ошибка
2	**Форматы экспорта**	Проверка настроек экспорта для иконок.	Иконка "Корзина" экспортируется как PNG, хотя должна быть SVG.	Ошибка
3	**Разрешения (@1x, @2x, @3x)**	Наличие нескольких версий для растровых ресурсов.	Для фоновой текстуры есть только @1x версия.	Ошибка
4	**Соответствие сетке (8pt)**	Измерение размеров и отступов.	Кнопка имеет высоту 38px, что не кратно 8.	Ошибка
5	**Цветовой контраст (WCAG)**	Использование плагина Stark в Figma.	Контрастность серого текста (#888888) на белом фоне составляет 2.9:1 (требуется мин. 4.5:1).	Ошибка
6	**Цветовой профиль**	Проверка свойств экспортированного PNG-файла.	Файл banner.png имеет цветовой профиль sRGB.	OK
7	**Размер файла**	Проверка веса экспортированного файла.	Файл hero_image@2x.png весит 450 КБ, что превышает лимит в 200 КБ.	Ошибка
...	...	...	...	...

6. Рекомендации и исправления

Опишите, КАК ИМЕННО вы исправили каждую найденную ошибку.

- **По ошибке №1 (Нейминг):** "Элемент Изображение 123 был переименован в illustration_onboarding_welcome в соответствии с конвенцией: [тип]_[экран]_[назначение]."
- **По ошибке №2 (Формат):** "Для иконки 'Корзина' настроен экспорт в формате SVG, так как это векторная графика без сложных градиентов."
- **По ошибке №4 (Сетка):** "Высота кнопки была изменена с 38px на 40px для соответствия 8pt grid."
- **По ошибке №5 (Контраст):** "Цвет текста был изменен с #888888 на #767676 для достижения контрастности 4.6:1."
- **По ошибке №7 (Размер файла):** "Изображение hero_image@2x.png было оптимизировано с помощью сервиса TinyPNG, что позволило уменьшить его вес до 180 КБ без видимой потери качества."

7. Итоговый пакет ресурсов

Опишите результат вашей работы.

- **Структура папок:** Покажите, как вы организовали файлы для разработчика.

```
text
assets/
├── icons/
│   ├── ic_cart.svg
│   └── ic_profile.svg
├── images/
│   ├── background@1x.png
│   ├── background@2x.png
│   └── background@3x.png
└── illustrations/
    └── onboarding_welcome.svg
```

- **Сопроводительная документация:** Упомните, что был создан файл README.md с описанием используемых шрифтов, цветовой палитры и ключевых компонентов.

8. Выводы

- **Итоги аудита:** "В ходе аудита 5 макетов было выявлено 12 нарушений технических требований. Наиболее частыми проблемами были: несоблюдение сетки (40% ошибок) и низкая цветовая контрастность (30% ошибок)."
- **Приобретенные навыки:** "В ходе работы я научился применять технические требования на практике, использовать инструменты для проверки доступности и оптимизировать графические ресурсы для production-среды."
- **Значимость:** "Грамотная подготовка графики ускоряет работу разработчиков, обеспечивает единообразие интерфейса на всех устройствах и улучшает пользовательский опыт, в том числе для людей с ограниченными возможностями."

Пример (фрагмент практической части)

Задача: Проверить кнопку "Добавить в корзину" на соответствие ТТ.

Технические требования:

- Формат: SVG.
- Размер: 24x24 pt (для @1x).
- Цвет: #28A745 (основной акцентный цвет).
- Имя файла: icon_cart_add.svg.

№	Проверяемый элемент	Метод проверки	Результат	Статус
1	**Формат**	Проверка настроек экспорта в Figma.	Экспортируется как PNG.	Ошибка
2	**Размер**	Измерение в режиме "Inspect".	Размер frame составляет 26x26 pt.	Ошибка
3	**Цвет заливки**	Проверка стиля слоя.	Цвет заливки #28A745 соответствует ТТ.	OK
4	**Имя файла**	Проверка имени слоя.	Слой назван add to cart.	Ошибка

Рекомендации и исправления:

1. Для исправления ошибки №1 был изменен формат экспорта на SVG.
2. Размер frame был изменен на 24x24 pt.
3. Имя слоя было изменено на icon_cart_add для корректного экспорта.

Итог: Ресурс был успешно экспортирован в файл icon_cart_add.svg, соответствующий всем техническим требованиям.

Практическая работа №13 - 18. Подготовка растровых изображений для интерфейса

Структура практической работы

Оформляем в виде технического отчета с акцентом на практические действия.

1. Титульный лист

- Название учебного заведения, предмета, номера работы, ФИО студента, группы, ФИО преподавателя.

2. Введение

- **Цель работы:** Освоить полный цикл подготовки растровых изображений для веб- и мобильных интерфейсов: от выбора и обработки до оптимизации и экспорта в соответствии с техническими требованиями.
- **Задачи:**
  - Изучить критерии выбора исходных растровых изображений.
  - Освоить базовые техники обработки (кадрирование, ретушь, цветокоррекция).
  - Научиться адаптировать изображения под разные разрешения экранов (ретину).
  - Освоить методы оптимизации (сжатие) без потери визуального качества.
  - Подготовить пакет изображений для передачи в разработку.
- **Объект исследования:** Набор исходных растровых изображений (например, фотографии для баннеров, карточки товаров, фоновые текстуры).

3. Теоретическая часть: Критерии качественного растрового изображения в интерфейсе

Кратко опишите, на что нужно обращать внимание.

- **Смысловое соответствие:** Изображение должно решать задачу и нести смысл, соответствующий контенту.
- **Техническое качество:**
  - **Резкость:** Отсутствие размытия (если это не творческий прием).
  - **Разрешение:** Достаточное для целевых устройств (обычно в 1.5-2 раза больше конечного размера в макете).
  - **Шум и артефакты:** Отсутствие цифрового шума, пикселизации, артефактов сжатия.
  - **Экспозиция и цветовой баланс:** Корректная яркость, контрастность, естественные цвета.
- **Композиция:** Правило третей, наличие акцента, свободное пространство для текста (если нужно).
- **Размер и вес:** Оптимальное соотношение качества и объема файла для быстрой загрузки.

4. Инструментарий

Опишите программное обеспечение, которое вы использовали.

- **Основной редактор:** Adobe Photoshop (профессиональный стандарт), GIMP (бесплатная альтернатива), Affinity Photo.
- **Инструменты оптимизации:** Плагины для Photoshop (ImageOptim, TinyPNG), онлайн-сервисы (TinyPNG, Squoosh.app), встроенные функции экспорта (Export As... Save for Web...).
- **Просмотр метаданных:** Adobe Bridge, Finder/Проводник Windows.

5. Практическая часть: Пошаговый разбор подготовки изображений

Это основная часть работы. Разбейте ее на этапы и проиллюстрируйте скриншотами.

Этап 1: Отбор и первичный анализ

- **Действие:** Возьмите 3-4 исходных изображения (например, с бесплатных стоков: Unsplash, Pexels). Проанализируйте их по критериям из п.3.
- **Результат в отчете:** Таблица с анализом.

Исходный файл	Разрешение	Вес	Оценка качества	Потенциальное использование
nature.jpg	4000x6000 px	4.5 МБ	Отличное	Фон героя-блока
portrait.jpg	2000x3000 px	2.1 МБ	Хорошее, есть шум в тенях	Карточка команды
product.png	800x800 px	500 КБ	Низкое, есть пикселизация	Не подходит, нужен исходник лучше

Этап 2: Базовая обработка и кадрирование

- **Действие:** Выполните коррекцию для выбранных изображений.

- **Кадрирование:** Обрежьте изображение под нужные пропорции (например, 16:9 для баннера, 1:1 для аватарки). Следите за композицией.
- **Цветокоррекция:** Настройте Уровни (Levels), Кривые (Curves), Яркость/Контрастность.
- **Ретушь:** Уберите лишние элементы (Пластика, Штамп), уменьшите шум.

- **Результат в отчете:** Скриншоты "До" и "После" обработки с кратким описанием проделанных действий.
- *Пример: "Для файла* portrait.jpg *было применено кадрирование для смещения композиции по правилу третей, повышена контрастность на 15% и убран шум в тенях с помощью фильтра 'Reduce Noise'."*

Этап 3: Масштабирование и подготовка под ретину

- **Действие:** Создайте версии изображения для разных плотностей пикселей.
- **Результат в отчете:** Таблица с подготовленными активами.

Конечный размер в макете	Коэффициент	Целевое разрешение	Имя файла	Вес
400x300 px	@1x (базовое)	400x300 px	hero-banner.jpg	45 КБ
400x300 px	@2x (ретина)	800x600 px	hero-banner@2x.jpg	110 КБ
400x300 px	@3x (супер-ретина)	1200x900 px	hero-banner@3x.jpg	220 КБ

- *Важно: Объясните, почему нельзя просто уменьшить огромное исходное изображение до @1x. Нужно сначала обработать его в большом размере, а затем ресайзить вниз с помощью интерполяции (Bicubic Sharper).*

Этап 4: Оптимизация и выбор формата

- **Действие:** Протестируйте разные форматы и настройки сжатия.
- **Результат в отчете:** Сравнительная таблица.

Исходный файл	Формат	Настройки качества	Итоговый вес	Визуальная оценка
hero-banner@2x.png	PNG	Без сжатия	850 КБ	Идеально
hero-banner@2x.jpg	JPEG	Качество 90%	180 КБ	Практически неотличимо
hero-banner@2x.jpg	JPEG	Качество 60%	110 КБ	Появились артефакты
hero-banner@2x.webp	WEBP	Качество 80%	75 КБ	Практически неотличимо

- **Вывод по этапу:** "Для фотографических изображений оптимальным выбором является формат JPEG с качеством 80-85%. Формат WEBP показывает лучшие результаты по сжатию, но требует проверки поддержки браузерами. PNG следует использовать для изображений с прозрачностью или резкими границами (графика, логотипы)."

6. Итоговый пакет ресурсов

- **Структура папок:** Покажите, как вы организовали файлы для разработчика. Это демонстрирует профессиональный подход.

```
images/
├── heroes/                 # Папка для баннеров
│   ├── hero-banner.jpg       # Версия @1x
│   ├── hero-banner@2x.jpg    # Версия @2x
│   └── hero-banner@3x.jpg    # Версия @3x
├── avatars/                # Папка для аватарок
│   ├── user-avatar.png
│   └── user-avatar@2x.png
└── products/               # Папка для товаров
    ├── product-1.jpg
    └── product-1@2x.jpg
```

7. Выводы

- **Итоги работы:** "В ходе работы было отобрано и обработано 3 исходных изображения. Для каждого из них подготовлено 3 версии для разных разрешений экранов. В результате средний размер файла был уменьшен на 70% по сравнению с исходниками без видимой потери качества за счет подбора оптимального формата и степени сжатия."
- **Приобретенные навыки:** "Освоены техники кадрирования, цветокоррекции, ретуши и массового экспорта в Adobe Photoshop. На практике изучены различия форматов JPEG, PNG и WEBP и критерии их выбора."
- **Значимость:** "Качественная подготовка растровых изображений напрямую влияет на производительность интерфейса (скорость загрузки) и его визуальную привлекательность на всех типах устройств."

Пример (фрагмент практической части)

Задача: Подготовить фоновое изображение для шапки сайта (hero image).

Исходные данные:

- Файл: mountain_landscape.jpg (6000x4000 px, 8 МБ).
- Требуемый размер в макете: 1200x600 px.
- Платформа: Веб-сайт (необходимы версии @1x и @2x).

Ход работы:

1. **Кадрирование:** В Photoshop обрезал изображение до пропорций 2:1. Сместил кадр так, чтобы горы оказались по правилу третей.
2. **Коррекция:** Увеличил контрастность и слегка поднял насыщенность синего и зеленого цветов.
3. **Масштабирование:**
   - Создал файл размером 2400x1200 px (@2x).
   - Поместил обработанное изображение, масштабировал и позиционировал.
   - Сохранил как hero-banner@2x.jpg с качеством 80%.
4. **Создание @1x:**
   - Изменил размер изображения до 1200x600 px (Image Size -> Resample: Bicubic Sharper).
   - Сохранил как hero-banner.jpg с качеством 85%.
5. **Оптимизация:** Прогнал оба файла через сервис Squoosh.app, дополнительно уменьшив вес на 15%.

Итог:

- hero-banner.jpg (1200x600 px) - **95 КБ**
- hero-banner@2x.jpg (2400x1200 px) - **215 КБ**

Практическая работа №19 - 24 .Создание векторных рисунков с помощью кривых

Структура практической работы

Оформляем в виде отчета, сочетающего теорию и практику с визуальными результатами.

1. Титульный лист

- Название учебного заведения, предмета, номера работы, ФИО студента, группы, ФИО преподавателя.

2. Введение

- **Цель работы:** Освоить технику создания и редактирования векторных контуров (кривых Безье) для создания точных и масштабируемых интерфейсных элементов и иллюстраций.

- **Задачи:**
  - Изучить теоретические основы векторной графики и кривых Безье.
  - Освоить инструменты для создания и редактирования контуров (Pen Tool, Anchor Point Tool и др.).
  - Научиться создавать простые и сложные составные контуры.
  - Отработать применение операций с контурами (объединение, вычитание, пересечение).
  - Создать серию векторных объектов, демонстрирующих владение техникой.
- **Объект исследования:** Инструменты для работы с векторными контурами в программе Adobe Illustrator (или Figma, Affinity Designer, Inkscape).

3. Теоретическая часть: Основы векторной графики и кривых Безье

Кратко, но емко опишите ключевые concepts.

- **Векторная графика vs. Растровая графика:** Объяснение разницы (пиксели vs. математические описания форм), преимуществ для интерфейсов (масштабируемость, маленький вес, простое редактирование).
- **Основные элементы векторного контура:**
  - **Опорная точка (Anchor Point):** Узел, определяющий положение.
  - **Сегмент (Segment):** Линия между двумя опорными точками.
  - **Вектор управления (Handle):** Определяет кривизну сегмента. Положение и длина "рычагов" влияют на форму кривой.
- **Типы точек:**
  - **Угловая точка (Corner Point):** Рычаги независимы, резкий излом.
  - **Сглаженная точка (Smooth Point):** Рычаги связаны, создают плавную кривую.
  - **Точка излома (Cusp Point):** Рычаги независимы, но соединены в одной точке, позволяя создавать острые, но плавные переходы.

4. Инструментарий

Опишите программу и ключевые инструменты, которые вы использовали.

- **Основная программа:** Adobe Illustrator (индустриальный стандарт), Figma (для UI), Affinity Designer, Inkscape (бесплатная).
- **Ключевые инструменты:**
  - **Pen Tool (P):** Основной инструмент для создания контуров.
  - **Add Anchor Point Tool (+):** Добавление новых точек на контур.
  - **Delete Anchor Point Tool (-):** Удаление точек.
  - **Anchor Point Tool (Shift+C):** Преобразование типа точек (из угловой в сглаженную и наоборот).
  - **Direct Selection Tool (A):** Выделение и перемещение отдельных точек и рычагов.
  - **Pathfinder / Boolean Operations:** Для сложных форм (объединение, вычитание и т.д.).

5. Практическая часть: Пошаговое создание векторных объектов

Это ядро работы. Разбейте на упражнения по нарастающей сложности. По каждому объекту приведите скриншоты ключевых этапов.

Упражнение 1: Простые геометрические фигуры (от руки кривыми)

- **Цель:** Отработка создания прямых сегментов и плавных кривых.
- **Задание:** Нарисовать простой объект (например, лист, облако, сердце), используя минимальное количество точек.
- **Результат в отчете:**

1. Скриншот итогового объекта.
2. Скриншот этого же объекта в режиме отображения сетки или точек (View > Outline).
3. Краткое описание: "Облако создано из 6 опорных точек, все точки — сглаженные, что позволило добиться плавных изгибов. Для создания сердца использовались как сглаженные точки (для верхних полукружий), так и угловая точка (для острия внизу)."

Упражнение 2: Трассировка (обводка) растрового изображения

- **Цель:** Научиться точно повторять сложные формы.

- **Задание:** Поместить растровое изображение (например, контурный рисунок животного, логотип) в документ и обвести его с помощью Pen Tool.
- **Результат в отчете:**

1. Скриншот исходного растрового изображения.
2. Скриншот процесса трассировки (виден и растровый оригинал, и создаваемый векторный контур).
3. Скриншот готового векторного объекта.
4. Описание трудностей и решений: "Наибольшую сложность представил участок с плавной S-образной кривой. Для его точного повторения пришлось использовать точки излома (Cusp), чтобы независимо настроить длину и угол каждого рычага."

Упражнение 3: Создание сложной составной фигуры с помощью Pathfinder

- **Цель:** Освоить логические операции для построения сложных форм из простых.
- **Задание:** Создать стилизованную иконку (например, видеокамеру, ключ, логотип шестеренки).
- **Результат в отчете:**

1. Скриншот этапа с набором простых примитивов (круги, квадраты, прямоугольники).
2. Скриншот панели Pathfinder с выделенными используемыми операциями (Unite, Minus Front, etc.).
3. Скриншот готовой иконки.
4. Пошаговое описание: "1. Создан прямоугольник (корпус камеры). 2. Создан круг (объектив). 3. С помощью операции 'Minus Front' из большого круга вырезан малый круг для создания кольца объектива. 4. Небольшой квадрат, повернутый на 45 градусов, с помощью 'Unite' присоединен к корпусу для создания кнопки записи."

Упражнение 4: Создание интерфейсной иконки в двух стилях

- **Цель:** Применить навыки для решения конкретной UI-задачи.
- **Задание:** Создать иконку "Настройки" (шестеренка) в двух стилях: "Линейная" (Outline) и "Заливная" (Filled).
- **Результат в отчете:**

1. Скриншот двух финальных иконок.
2. Описание процесса: "Для линейной иконки использовался контур с обводкой (Stroke) и без заливки (Fill). Зубцы шестеренки созданы путем вычитания маленьких прямоугольников из большого круга. Для заливной иконки использовались скомбинированные контуры с заливкой и без обводки. Важно было сохранить визуальное равенство толщин линий и размеров в обоих стилях."

6. Анализ и выводы

- **Сравнительный анализ своих работ:** "На начальном этапе (Упражнение 1) контуры были избыточны по количеству точек. К финальному упражнению удалось добиться создания сложных форм с минимальным количеством опорных точек, что является признаком качественного векторного рисунка."
- **Типичные ошибки и их решение:**
  - *Ошибка:* Слишком много точек.
  - *Решение:* Использовать длинные рычаги для управления кривизной.
  - *Ошибка:* "Дрожащие", негладкие кривые.
  - *Решение:* Практиковаться в плавном выставлении рычагов, использовать инструмент сглаживания (Smooth Tool).
- **Приобретенные навыки:** "В ходе работы я научился уверенно владеть пером (Pen Tool), предвидеть, как положение рычагов повлияет на форму кривой, создавать сложные формы с помощью логических операций и подготавливать чистые, масштабируемые assets для интерфейсов."
- **Значимость для разработки интерфейсов:** "Векторные иконки, созданные кривыми, являются основой современного UI. Они обеспечивают четкое отображение на любых

экранах, легко подвергаются анимации и, будучи экспортированы в SVG, имеют минимальный вес и не нагружают производительность приложения или сайта."

Пример (фрагмент практической части)

Упражнение: Создание иконки "Самолетик" (отправка сообщения).

Ход работы:

1. **Создание корпуса:**
   - Инструментом Pen Tool (P) создаю треугольник. Три клика для создания трех угловых точек.
   - Инструментом Direct Selection Tool (A) выбираю верхнюю точку и с помощью клавиш стрелок смещаю ее вниз, создавая трапецию.
2. **Создание хвоста и крыльев:**
   - С помощью Pen Tool рисую прямоугольник для хвоста.
   - Рисую два треугольника для крыльев. Для этого использую Pen Tool с тремя точками.
3. **Объединение фигур:**
   - Выделяю все созданные фигуры.
   - На панели Pathfinder нажимаю кнопку Unite (первая слева). Все элементы объединяются в один сложный контур.
4. **Сглаживание углов:**
   - Инструментом Direct Selection Tool (A) выделяю все опорные точки, которые хочу скруглить.
   - На панели управления ввожу значение 3 pt в поле Corners. Углы плавно скругляются.
5. **Финальные штрихи:**
   - Устанавливаю заливку (Fill) синего цвета.
   - Убираю обводку (Stroke).

Итог: Создана простая, стилизованная иконка самолетика, состоящая из одного векторного контура. Она идеально масштабируется и готова к экспорту в SVG для использования в интерфейсе.

Практическая работа №25 - 30. Подготовка векторных изображений для интерфейса

Структура практической работы

Оформляем в виде технического отчета, ориентированного на передачу готовых assets разработчику.

1. Титульный лист

- Название учебного заведения, предмета, номера работы, ФИО студента, группы, ФИО преподавателя.

2. Введение

- **Цель работы:** Научиться подготавливать и экспортировать векторные изображения, созданные в предыдущих работах, для их непосредственного использования в веб- и мобильной разработке.
- **Задачи:**
  - Привести векторные файлы к технически корректному виду.
  - Оптимизировать контуры и структуру файлов.
  - Освоить правила именования и организацию файлов.
  - Научиться экспортировать векторные ресурсы в необходимых форматах (SVG, PNG для fallback).
  - Сформировать итоговый пакет ресурсов с сопроводительной документацией.

- **Объект исследования:** Векторные файлы, созданные в работах №19-24 (иконки, иллюстрации, UI-элементы).

3. Теоретическая часть: Преимущества векторной графики в интерфейсах
Кратко обоснуйте, почему это важно.

- **Масштабируемость:** Отсутствие потерь качества на любых разрешениях экранов (ретина, 4K, 8K).
- **Малый вес:** Файлы SVG, особенно оптимизированные, занимают значительно меньше места, чем растровые аналоги.
- **Гибкость стилизации:** Цвет, обводку и другие параметры можно менять через CSS/код, не экспортируя новые файлы.
- **Доступность:** Элементы SVG могут быть доступны для скринридеров.
- **Возможность анимации:** SVG-графику можно анимировать средствами CSS или JavaScript.

4. Этапы подготовки векторных изображений (Практическая часть)
Это ядро работы. Оформите каждый этап с примерами и скриншотами.
Этап 1: Ревизия и очистка векторных файлов

- **Цель:** Убедиться, что файлы технически "чистые" и готовы к экспорту.
- **Действия и проверки:**
  - **Объединение контуров:** Все составные части иконки должны быть объединены (Pathfinder > Unite) или логически сгруппированы.
  - **Удаление мусора:** Удалить невидимые объекты, случайные точки, незамкнутые контуры.
  - **Выравнивание по пиксельной сетке:** Включить опцию "Align to Pixel Grid" (в Illustrator) для обеспечения четких границ на экране.
  - **Оптимизация количества точек:** Убедиться, что для описания форм используется минимально необходимое количество опорных точек.
- **Результат в отчете:** Таблица с аудитом файлов "до" и "после".

Исходный файл	Найденные проблемы	Выполненные действия
icon_cloud.ai	3 незамкнутых контура, 2 лишние точки	Контуры замкнуты, лишние точки удалены
logo_app.ai	Объекты не выровнены по пиксельной сетке	Включена опция "Align to Pixel Grid"

Этап 2: Приведение к единому стилю и правилам

- **Цель:** Обеспечить визуальную согласованность всех ресурсов.
- **Действия:**
  - **Толщина линий (Stroke):** Проверить, что для всех иконок одного семейства используется одинаковая толщина обводки (например, 2px).
  - **Углы скругления (Border Radius):** Привести к единому значению (например, 2px для мелких иконок, 4px для крупных кнопок).
  - **Стиль окончаний линий (Stroke Cap/Corner):** Проверить единообразие (обычно Round Cap и Round Join для более мягкого вида).
- **Результат в отчете:** Скриншот панели свойств с едиными настройками для семейства иконок.

Этап 3: Именование и организация

- **Цель:** Создать понятную и масштабируемую структуру файлов.
- **Действия:**
  - **Именование слоев и объектов внутри файла:** Называть логично (например, icon, background, text), а не "Слой 1".
  - **Именование экспортируемых файлов:**
    - Использовать английский язык.
    - Придерживаться соглашения: [тип]_[имя]_[состояние].[формат].
    - **Пример:** icon_search_active.svg, illustration_onboarding_welcome.svg.

- **Организация папок:** Создать логичную структуру.
- **Результат в отчете:** Пример дерева папок.

```
assets/
├── icons/
│   ├── navigation/
│   │   ├── icon_home.svg
│   │   └── icon_profile.svg
│   └── actions/
│       ├── icon_search.svg
│       └── icon_heart.svg
├── illustrations/
│   ├── onboarding_1.svg
│   └── empty_state.svg
└── logos/
    └── logo_main.svg
```

Этап 4: Настройка экспорта и выбор форматов

- **Цель:** Получить файлы в нужных форматах с оптимальными настройками.
- **Действия:**
- **Экспорт в SVG:**
- **Настройки:** В Illustrator: "SVG Profiles: SVG 1.1", "Font: SVG" (или преобразовать в кривые).
- **Оптимизация:** Использовать сервисы типа SVGOMG для удаления служебной информации.
- **Проверка:** Открыть SVG в браузере и текстовом редакторе, убедиться в чистоте кода.
- **Экспорт в PNG (для fallback):**
- Создать версии для @1x, @2x, @3x.
- Использовать настройку экспорта "Use Artboards".
- **Результат в отчете:** Сравнительная таблица форматов.

Формат	Преимущества	Недостатки	Идеальное применение
SVG	Масштабируемость, малый вес, стилизация CSS	Может быть сложным для очень детализированных изображений	Иконки, логотипы, простые иллюстрации, UI-графика
PNG	Широкая поддержка, поддержка прозрачности	Большой вес при высоких разрешениях, не масштабируется	Резервный формат, если требуется поддержка очень старых браузеров

Этап 5: Создание сопроводительной документации

- **Цель:** Облегчить работу разработчика.
- **Действия:**
- Создать файл README.md или assets-guide.pdf.
- Указать: используемые шрифты, цветовую палитру (HEX-коды), размерную сетку (например, 8pt grid).
- **Результат в отчете:** Пример фрагмента документации.

```
# Руководство по использованию графических ресурсов

## Цвета
* Основной акцент: `#2A6DFF`
* Текст: `#333333`
* Фон: `#FFFFFF`

## Иконки
* Семейство: "UI Icons Set"
```

* Толщина обводки: 2px
* Стиль: Заливные (Filled)
* Размер артборда: 24x24 px

6. Итоговый пакет ресурсов и выводы

- **Предоставляемый результат:** "В результате работы сформирован zip-архив ui-assets-pack.zip, содержащий оптимизированные SVG-файлы, PNG-фалбэки для ключевых ресурсов и файл README.md с инструкциями."
- **Приобретенные навыки:** "Освоен полный цикл подготовки векторной графики: от технического аудита и оптимизации контуров до настройки экспорта и формирования developer-friendly пакета. Навык использования SVG-формата для повышения эффективности и качества интерфейса."
- **Вывод:** "Качественная подготовка векторных изображений — это не менее важный этап, чем их создание. Чистые, оптимизированные и хорошо организованные ресурсы значительно ускоряют процесс интеграции дизайна в продукт, обеспечивая его четкое отображение на всех устройствах и высокую производительность."

Пример (фрагмент практической части)

Задача: Подготовить к экспорту иконку "Пользователь".

Исходные данные:

- Файл: user_icon_raw.ai
- Проблемы: Необъединенные контуры, невыровненные по сетке, произвольное именование.

Ход работы:

1. **Аудит:** Открыл файл, обнаружил 3 независинных контура (голова, тело, круглый фон).
2. **Очистка:**
   - Выделил все контуры и применил Pathfinder > Unite.
   - Включил View > Pixel Preview и Align to Pixel Grid для обеспечения четкости.
   - Удалил несколько лишних опорных точек инструментом Delete Anchor Point Tool.
3. **Именование:** Переименовал слой с "Layer 1" на "icon_user".
4. **Экспорт:**
   - В диалоге Export for Screens выбрал формат SVG.
   - В настройках SVG выбрал профиль SVG 1.1 и опцию Minify.
   - Экспортировал файл с именем icon_user.svg.
5. **Дополнительная оптимизация:**
   - Открыл полученный icon_user.svg в сервисе SVGOMG.
   - Уменьшил размер файла на 15%, убрав лишние метаданные, без изменения визуала.

Итог:

- Исходный вес: user_icon_raw.ai - **45 КБ**
- Вес после экспорта: icon_user.svg - **1.2 КБ**
- Вес после SVGOMG: icon_user.svg - **1.0 КБ**

Файл готов к использованию в интерфейсе. Он масштабируется без потерь, имеет минимальный вес и корректно отображается во всех браузерах.

Практическая работа №31 - 36. Подготовка анимированных элементов интерфейса

Структура практической работы

Оформляем в виде отчета, сочетающего принципы, практику и технические спецификации.

1. Титульный лист

- Название учебного заведения, предмета, номера работы, ФИО студента, группы, ФИО преподавателя.

2. Введение

- **Цель работы:** Освоить принципы и технические приемы создания и подготовки анимации для веб- и мобильных интерфейсов.

- **Задачи:**
  - Изучить принципы UX-анимации и её классификацию.
  - Освоить инструменты для создания прототипов анимации (Figma, Principle).
  - Научиться создавать анимированные векторные ресурсы (Lottie, SVG animation).
  - Освоить методы экспорта и интеграции анимации в код.
  - Создать библиотеку анимированных компонентов.
- **Объект исследования:** Микроанимации, переходы между экранами, анимированные иллюстрации.

3. Теоретическая часть: Принципы и классификация UI-анимации

Кратко, но емко опишите ключевые concepts.

- **Роль анимации в интерфейсе:**
  - **Функциональная:** Обратная связь, ориентация в пространстве, визуализация изменений.
  - **Эмоциональная:** Создание настроения, укрепление бренда, delight-эффект.
- **Принципы (на основе 12 принципов анимации Диснея, адаптированных для UI):**
  - **Timing & Easing:** Кривые замедления (easing curves) важнее длительности.
  - **Offset & Delay:** Последовательное появление элементов.
  - **Parenting & Transformation:** Связывание анимаций друг с другом.
  - **Value Change:** Анимация изменений свойств (прозрачность, положение, масштаб).
- **Типы анимации по назначению:**
  - **Микроанимации:** Отклик кнопок, индикаторы загрузки, переключения.
  - **Переходы (Transitions):** Навигация между экранами, появление/исчезновение элементов.
  - **Системные анимации:** Pull-to-refresh, скролл, параллакс-эффекты.

4. Инструментарий

Опишите программы и технологии для разных этапов.

- **Прототипирование и дизайн:** Figma (Auto Animate, Smart Animate), Principle, Protopie.
- **Создание векторной анимации:** Adobe After Effects (+ BodyMovin/Lottie plugin), Rive.
- **Веб-реализация:** CSS Animations/Transitions, GreenSock Animation Platform (GSAP), Framer Motion.
- **Форматы для разработки:** Lottie (.json), GIF, APNG, Animated SVG, Video (MP4).

5. Практическая часть: Создание и подготовка анимаций

Это ядро работы. Разбейте на упражнения по нарастающей сложности.

Упражнение 1: Прототипирование микроанимации в Figma

- **Цель:** Создать интерактивный прототип анимированной кнопки.
- **Задание:** Создать кнопку с анимациями состояний: Normal -> Hover -> Pressed.
- **Результат в отчете:**

1. Скриншоты всех состояний кнопки.
2. Ссылка на интерактивный прототип в Figma.
3. Описание параметров анимации:

- **Свойство:** Position, Scale, Fill color.
- **Длительность:** 200 мс.
- **Easing:** "Ease Out" (для плавного завершения).

Упражнение 2: Создание Lottie-анимации в After Effects

- **Цель:** Освоить процесс создания lightweight векторной анимации.
- **Задание:** Создать анимированную иконку "успешной отправки" (галочка).
- **Ход работы:**

1. **Подготовка:** Создать композицию с векторными слоями в AE.
2. **Анимация:** Анимировать путь галочки (Trim Paths) и фоновый круг (Scale).

3. **Экспорт:** Использовать плагин BodyMovin для экспорта в JSON.

- **Результат в отчете:**

1. Скриншот временной шкалы в After Effects.
2. Видео-демонстрация готовой анимации (GIF или MP4).
3. Сравнение веса файлов: "Исходный AEP - 2.3 МБ, Lottie JSON - 4.8 КБ".

Упражнение 3: Создание CSS-анимации

- **Цель:** Научиться подготавливать анимации для прямой интеграции в код.
- **Задание:** Создать вращающийся индикатор загрузки.
- **Результат в отчете:**

1. Код CSS с ключевыми кадрами (@keyframes).
2. Скриншот или GIF работающей анимации.
3. Сравнение с другими подходами: "CSS-анимация имеет наилучшую производительность для простых преобразований".

Упражнение 4: Анимация перехода между экранами

- **Цель:** Освоить создание связующих анимаций.
- **Задание:** Спрототипировать переход с экрана списка на экран деталей (Shared Element Transition).
- **Результат в отчете:**

1. Видео-демонстрация перехода.
2. Описание логики: "Карточка со списка плавно масштабируется и перемещается на позицию заголовка на экране деталей, фон затемняется".

6. Техническая подготовка и экспорт

Критически важный раздел для передачи в разработку.

- **Оптимизация анимаций:**
  - **Производительность:** Предпочитать свойства opacity и transform (они не вызывают перерасчет макета).
  - **Сжатие:** Оптимизировать Lottie-файлы через LottieFiles Optimizer.
- **Создание технической документации:**
  - **Для каждой анимации указать:**
    - Название и назначение.
    - Длительность (мс).
    - Кривая easing (css: ease-out, cubic-bezier(0.33, 1, 0.68, 1)).
    - Задержка (delay).
    - Формат файла и вес.
- **Организация файлов:**
  - Создать структурированную папку для ассетов.

```
animations/
├── lottie/
│   ├── success-check.json
│   └── loading-spinner.json
├── css/
│   └── animations.css
├── videos/
│   └── tutorial.mp4
└── README.md
```

7. Примеры и рекомендации

Пример: Анимация кнопки "Добавить в корзину"

```
/* CSS код анимации */
.add-to-cart {
  transition: all 0.3s cubic-bezier(0.34, 1.56, 0.64, 1);
}
```

```
.add-to-cart:active {
  transform: scale(0.95);
  background-color: #2ecc71;
}
```

Техническая спецификация:

- **Назначение:** Визуальная обратная связь при нажатии
- **Длительность:** 300 мс
- **Easing:** cubic-bezier(0.34, 1.56, 0.64, 1) - упругая анимация
- **Свойства:** transform: scale(), background-color
- **Производительность:** Высокая (используется только transform и opacity)

8. Выводы и анализ

- **Сравнительный анализ форматов:**
  - **Lottie:** Лучший выбор для сложных векторных анимаций
  - **CSS:** Идеален для простых интеракций и переходов
  - **GIF/Video:** Только для сложных растровых анимаций
- **Критерии качества UI-анимации:**
  - **Производительность:** 60 FPS, плавное воспроизведение
  - **Осмысленность:** Решает конкретную пользовательскую задачу
  - **Ненавязчивость:** Не мешает основному контенту
  - **Доступность:** Учитывает настройки prefers-reduced-motion
- **Итоги работы:** "В ходе практической работы создана библиотека из 5 анимированных компонентов, охватывающих основные типы UI-анимаций. Освоен полный цикл от прототипирования до технической подготовки для разработчиков".
- **Приобретенные навыки:**
  - Создание интерактивных прототипов в Figma
  - Работа с Lottie и After Effects
  - Написание CSS-анимаций
  - Оптимизация анимированного контента
  - Создание технической документации для анимаций

Эта работа продемонстрирует ваш переход от статичного дизайнера к современному специалисту, способному создавать динамичные, живые интерфейсы с учетом технических возможностей и ограничений.

Практическая работа №37 - 48. Черновой монтаж и послойная анимация

Структура практической работы

Оформляем в виде творческо-технического отчета, демонстрирующего полный цикл работы над сложной анимацией.

1. Титульный лист

- Название учебного заведения, предмета, номера работы, ФИО студента, группы, ФИО преподавателя.

2. Введение

- **Цель работы:** Освоить комплексный процесс создания сложной анимационной сцены: от чернового монтажа элементов до финальной послойной анимации и экспорта.
- **Задачи:**
  - Разработать концепцию и раскадровку анимационной сцены.
  - Подготовить и импортировать все необходимые assets (графика, видео, звук).

- Выполнить черновой монтаж для определения временной структуры.
- Реализовать послойную анимацию всех элементов сцены.
- Настроить тайминг, ритм и кривые анимации.
- Добавить финальные эффекты и выполнить рендеринг.

- **Объект исследования:** Процесс создания анимированной сцены в программе для компоузинга (Adobe After Effects).

3. Теоретическая часть: Этапы создания сложной анимации

Кратко опишите профессиональный пайплайн.

- **Пре-продакшн:**
- **Концепция и сценарий:** Основная идея и сюжет.
- **Раскадровка:** Визуальное представление ключевых кадров и сцен.
- **Сбор ассетов:** Подготовка всей графики, видео, звуковых файлов.
- **Продакшн:**
- **Черновой монтаж:** Компоновка элементов по временной шкале для определения общей структуры и длительности.
- **Послойная анимация:** Детальная проработка движения каждого элемента.
- **Финальный монтаж:** Точная настройка тайминга и переходов.
- **Пост-продакшн:**
- **Добавление эффектов:** Цветокоррекция, глитчи, свечение и т.д.
- **Рендеринг и экспорт:** Финальный вывод в нужном формате.

4. Инструментарий

- **Основная программа:** Adobe After Effects (индустриальный стандарт).
- **Вспомогательные программы:**
- Adobe Illustrator / Photoshop (подготовка векторной и растровой графики).
- Adobe Premiere Pro (если требуется монтаж видеофрагментов).
- Adobe Media Encoder (финальный рендеринг).
- **Ключевые плагины:**
- **Motion v3:** Для расширенных возможностей анимации.
- **Overlord:** Для seamless-передачи графики из Illustrator.
- **LottieFiles:** Для экспорта в Lottie.

5. Практическая часть: Пошаговое создание анимированной сцены

Это ядро работы. Разбейте на логические этапы.

Этап 1: Разработка концепции и раскадровки

- **Задание:** Создать анимированный баннер для мобильного приложения (например, экран приветствия).
- **Концепция:** "Умный дом: плавное появление элементов интерфейса управления устройствами".
- **Результат в отчете:**
- Эскиз или раскадровка из 4-6 кадров, показывающая ключевые состояния анимации.
- Описание сцены: "Сначала появляется фон, затем плавно проявляются иконки устройств, после чего появляются элементы управления с задержкой".

Этап 2: Подготовка и организация ассетов

- **Действия:**
- Создать в Illustrator все элементы: иконки устройств, кнопки, фон, текст.
- Каждый элемент разместить на отдельном слое.
- Сохранить файл как smart_home_assets.ai.
- **Результат в отчете:**
- Скриншот панели слоев в Illustrator.
- Таблица с перечнем ассетов.

Название слоя	Тип элемента	Предназначение в анимации
background	Прямоугольник	Фоновая подложка

icon_light	Векторная иконка	Анимированное появление
icon_thermo	Векторная иконка	Анимированное появление с задержкой
slider	Группа объектов	Элемент управления

Этап 3: Импорт и черновой монтаж в After Effects

- **Действия:**
  - Импортировать .ai файл как Composition с сохранением размеров слоев.
  - Разместить слои на временной шкале в нужном порядке.
  - Установить приблизительные длительности видимости каждого элемента, создав черновую версию анимации.
- **Результат в отчете:**
  - Скриншот временной шкалы After Effects на этапе чернового монтажа.
  - Описание: "На данном этапе определено, что общая длительность анимации составит 5 секунд. Элементы расположены последовательно с наложением".

Этап 4: Послойная анимация (детальная проработка)

- **Действия:**
  - **Анимация фона:** Плавное увеличение масштаба (Scale) от 110% до 100%.
  - **Анимация иконок:**
    - Установка Position и Opacity ключевых кадров.
    - Использование последовательного появления через задержку (delay).
  - **Анимация элементов управления:**
    - Использование Trim Paths для анимации линий.
    - Анимация Slider с помощью Mask Path.
- **Результат в отчете:**
  - Скриншот Graph Editor с настроенными кривыми анимации.
  - Описание использованных свойств: "Для анимации иконки света использовались свойства Position, Scale и Opacity с easing-кривой 'Easy Ease Out'".

Этап 5: Работа с таймингом и ритмом

- **Действия:**
  - Настройка кривых в Graph Editor для придания анимации естественности.
  - Использование разных easing-кривых для разных типов движения.
  - Добавление эффекта "упругости" (overshoot) для некоторых элементов.
- **Результат в отчете:**
  - Сравнительные скриншоты кривых до и после настройки.
  - Пример кода cubic-bezier для CSS-аналога: cubic-bezier(0.68, -0.55, 0.265, 1.55).

Этап 6: Добавление специальных эффектов и финальная обработка

- **Действия:**
  - Добавление эффектов: **Gaussian Blur** для фона, **Drop Shadow** для элементов.
  - Создание плавных переходов между состояниями.
  - Цветокоррекция всей сцены.
- **Результат в отчете:**
  - Скриншот панели эффектов с настройками.
  - Сравнение кадров до и после применения эффектов.

6. Техническая оптимизация и экспорт

- **Предварительный рендеринг:** Использование RAM Preview для проверки плавности.
- **Настройка рендеринга:**
  - Выбор кодека: H.264 для видео, PNG последовательность для максимального качества.
  - Настройка битрейта и частоты кадров (обычно 24-30 fps).
- **Экспорт в разные форматы:**
  - **MP4:** Для демонстрации.

- **GIF:** Для использования в интерфейсах.
- **Lottie (.json):** Для интеграции в мобильные приложения и веб.

7. Анализ и выводы

- **Сравнительная таблица результатов:**

Этап работы	Длительность	Количество слоев	Основные техники
Черновой монтаж	2 часа	15	Базовая расстановка ключевых кадров
Послойная анимация	6 часов	15+	Graph Editor, Trim Paths, Mask Animation
Финальная обработка	1 час	15+	Эффекты, цветокоррекция

- **Проблемы и решения:**
- **Проблема:** Сложность синхронизации множества анимаций.
- **Решение:** Использование маркеров на временной шкале и пре-композиции.
- **Проблема:** "Дрожание" анимации.
- **Решение:** Настройка motion blur и сглаживания кривых.
- **Критерии оценки качества:**
- **Плавность:** Отсутствие рывков, стабильные 30 fps.
- **Согласованность:** Единый стиль всех анимаций.
- **Осмысленность:** Логичная последовательность действий.
- **Оптимизация:** Разумный баланс качества и размера файла.
- **Итоги работы:** "Создана комплексная анимационная сцена длительностью 5 секунд, включающая 15 анимированных элементов. Освоены техники чернового монтажа, послойной анимации и работы с кривыми в Graph Editor".
- **Приобретенные навыки:**
- Планирование анимационного проекта
- Работа со сложными композициями в AE
- Синхронизация множества анимированных элементов
- Создание естественного движения через кривые анимации
- Техническая оптимизация финального рендера

Пример (фрагмент практической части)

Задача: Анимировать появление элемента "умная лампочка".

Ход работы:

1. **Черновой монтаж:**
- Установил длительность появления: 0.8 секунд.
- Разместил элемент на временной шкале в общей композиции.
2. **Послойная анимация:**
- **0.0с:** Scale: 0%, Opacity: 0%
- **0.3с:** Scale: 110%, Opacity: 80% (легкий "overshoot")
- **0.8с:** Scale: 100%, Opacity: 100%
3. **Настройка кривых:**
- Scale: Резкий вход, плавный выход с overshoot
- Opacity: Плавное нарастание
4. **Добавление эффектов:**
- **Glow:** Легкое свечение при появлении
- **Motion Blur:** Для смягчения движения

Результат: Плавное, "живое" появление элемента, которое привлекает внимание, но не раздражает пользователя.

Эта работа продемонстрирует вашу способность не просто создавать отдельные анимации, а управлять сложными анимационными проектами от идеи до финальной реализации, что является ключевым навыком для современного motion-дизайнера.

Практическая работа №49 - 54. Создание анимационного логотипа

Структура практической работы

Оформляем в виде творческого проекта с полным обоснованием решений.

1. Титульный лист

- Название учебного заведения, предмета, номера работы, ФИО студента, группы, ФИО преподавателя.

2. Введение

- **Цель работы:** Освоить полный цикл создания анимационного логотипа: от разработки концепции и статичного варианта до создания динамической презентации и подготовки файлов для использования.
- **Задачи:**
  - Разработать концепцию логотипа и его анимации.
  - Создать статичный векторный логотип.
  - Спроектировать сценарий анимации (аниматик).
  - Реализовать анимацию в программе для моушн-дизайна.
  - Подготовить логотип для использования в разных контекстах.
- **Объект исследования:** Процесс создания анимированного логотипа для вымышленной или реальной компании.

3. Концепция и бриф

- **Название компании/продукта:** (например, "КосмосТревэл" - сервис для бронирования космических туров)
- **Сфера деятельности:** Космический туризм
- **Целевая аудитория:** Состоятельные люди 30-50 лет, технологические энтузиасты
- **Ключевые сообщения:** Инновации, надежность, мечта, простор
- **Идея анимации:** "Превращение звезды в траекторию полета ракеты"

4. Разработка статичного логотипа

- **Эскизы:** 3-4 варианта карандашных набросков с краткими пояснениями.
- **Векторная реализация:** Финальный вариант, созданный в Adobe Illustrator.

- **Обоснование дизайна:**
  - **Цветовая палитра:** Темно-синий (#0D1B3E) - космос, оранжевый (#FF6B35) - энергия
  - **Шрифт:** Sans-serif (Geometria) - современность, технологичность
  - **Форма:** Комбинация звезды и траектории движения

5. Раскадровка и аниматик

- **Сценарий анимации:**

1. Появление точки в центре (0.0-0.3с)
2. Превращение в звезду (0.3-0.8с)
3. Раскрытие звезды в орбиту (0.8-1.5с)
4. Движение по орбите (1.5-2.0с)
5. Появление названия компании (2.0-2.5с)

- **Раскадровка:** Серия из 5-6 скриншотов ключевых кадров с временными метками.

6. Техническая реализация анимации

Этап 1: Подготовка файлов

- Импорт векторного логотипа в After Effects через Overlord
- Разделение на слои для анимации:
  - star_core
  - star_rays (4 луча)
  - orbit_path
  - rocket_dot
  - company_name

Этап 2: Базовая анимация

```
// Псевдокод для анимации
animation_timeline = {
    0.0: {star_core: {scale: 0, opacity: 0}},
    0.3: {star_core: {scale: 100, opacity: 100}},
    0.8: {star_rays: {rotation: 0→45, scale: 0→100}},
    1.5: {orbit_path: {trim_paths: 0→100}},
    2.0: {rocket_dot: {follow_path: 0→100}},
    2.5: {company_name: {opacity: 0→100}}
}
```

Этап 3: Детальная проработка

- **Кривые анимации:**
  - Использование Easy Ease для плавности
  - Кастомные bezier-кривые для упругих движений
- **Эффекты:**
  - Glow для свечения звезды
  - Stroke для анимации орбиты
  - Motion Blur для движения ракеты

7. Создание вариантов анимации

Вариант А: Полная версия (2.5 секунды)

- Для использования в интро видео, презентациях
- Полный сценарий с раскрытием бренда

Вариант Б: Сокращенная версия (1.0 секунда)

- Для использования в мобильном приложении
- Только превращение точки в звезду

Вариант В: Микро-анимация (0.5 секунды)

- Для ховер-эффектов на сайте
- Легкое пульсирование звезды

8. Техническая подготовка к использованию

Экспорт в форматы:

- **MP4 (H.264):** Для соцсетей и презентаций

- **GIF:** Для email-рассылок и мессенджеров
- **Lottie (.json):** Для мобильных приложений и веба
- **SVG с анимацией:** Для современных браузеров
- **PNG последовательность:** Для профессионального видео-монтажа

Гайдлайн по использованию:

markdown

Руководство по использованию анимированного логотипа

Варианты анимации

- ****Полная версия****: Использовать в начале видео-роликов
- ****Сокращенная****: Для загрузочных экранов приложений
- ****Микро****: Для интерактивных элементов интерфейса

Технические требования

- Минимальный размер: 100px для анимированной версии
- Фон: Только темный (#0D1B3E) или прозрачный
- Длительность: Не изменять временные параметры

9. Пример реализации в коде

HTML/CSS реализация микро-анимации:

```
<div class="logo">
  <svg class="logo-star" viewBox="0 0 100 100">
    <!-- SVG path звезды -->
  </svg>
</div>

<style>
.logo-star {
  transition: transform 0.3s ease;
}
.logo:hover .logo-star {
  transform: scale(1.1) rotate(15deg);
  filter: drop-shadow(0 0 8px #FF6B35);
}
</style>
```

10. Анализ и выводы

Критерии оценки результата:

- ✔ Узнаваемость в анимированном виде
- ✔ Соответствие концепции бренда
- ✔ Техническая корректность реализации
- ✔ Оптимальность размера файлов
- ✔ Адаптивность к разным носителям

Сравнительная таблица форматов:

Формат	Размер	Качество	Применение
Lottie	15 КБ	□□□□□	Приложения, сайты
GIF	45 КБ	□□	Соцсети, email
MP4	80 КБ	□□□□	Видео, презентации

Приобретенные навыки:

- Создание комплексной айдентики
- Работа с временными шкалами в AE
- Оптимизация анимации для разных носителей
- Создание технической документации
- Экспорт в production-форматы

Рекомендации по использованию:
"Анимированный логотип 'КосмосТревэл' эффективно работает в цифровой среде, усиливая восприятие бренда на 40% по сравнению со статичной версией. Рекомендуется использовать полную версию для видео-контента, а сокращенные варианты - в интерфейсах."

Практическая работа №55 - 60. Создание анимационного проекта

Структура практической работы
Оформляем как полноценный творческий проект с полным циклом производства.

1. Титульный лист

- Название учебного заведения, предмета, номера работы, ФИО студента, группы, ФИО преподавателя.

2. Аннотация проекта

- **Название проекта:** "Эволюция интерфейсов: от командной строки до голосового управления"
- **Хронометраж:** 30-60 секунд
- **Целевая аудитория:** Студенты IT-специальностей, технологические энтузиасты
- **Ключевое сообщение:** Демонстрация развития способов взаимодействия человека с компьютером
- **Стилистика:** Минимализм, flat-дизайн, моушн-графика

3. Концепция и сценарий
Логлайн: "Визуальное путешествие через эпохи компьютерных интерфейсов, показывающее как технологии становятся все более естественными и интуитивными."
Сценарный план:
markdown
СЦЕНА 1: КОМАНДНАЯ СТРОКА (0:00-0:10)
- Появление терминала 1980-х годов
- Анимация ввода команд
- Монохромная цветовая схема

СЦЕНА 2: ГРАФИЧЕСКИЙ ИНТЕРФЕЙС (0:10-0:20)
- Плавная трансформация в оконный интерфейс
- Появление первых иконок и меню
- Добавление цвета

СЦЕНА 3: СЕНСОРНЫЕ ЭКРАНЫ (0:20-0:30)
- Переход к мобильным интерфейсам
- Анимация жестов (свайпы, тапы)
- Яркие современные цвета

СЦЕНА 4: БУДУЩЕЕ (0:30-0:40)
- Голосовые интерфейсы
- Дополненная реальность
- Прозрачные интерфейсы

4. Раскадровка и аниматик

Визуальная раскадровка:

- Серия из 8-12 ключевых кадров с описанием:
  - Композиция
  - Движение камеры
  - Длительность сцены
  - Звуковое сопровождение

Технический аниматик:

- Черновая версия с временными метками
- Блок-схема переходов между сценами

5. Дизайн-система проекта

Визуальный стиль:

- **Цветовая палитра:**
  - Эпоха 1: #000000, #00FF00 (монохром)
  - Эпоха 2: #000080, #C0C0C0 (Windows 95)
  - Эпоха 3: #007AFF, #FFFFFF (iOS)
  - Эпоха 4: #FFFFFF, прозрачность (AR)
- **Типографика:**
  - Courier New (командная строка)
  - MS Sans Serif (графический интерфейс)
  - San Francisco (мобильные интерфейсы)
  - Futura (будущее)
- **Анимационные принципы:**
  - Timing: Ускорение для старых интерфейсов, плавность для новых
  - Easing: Linear для терминала, ease-out для современности

6. Производственная часть

Этап 1: Подготовка ассетов

python
 # Структура проекта в After Effects
project/
├── assets/
│ ├── vectors/ (иконки, интерфейсы)
│ ├── textures/ (фоны, градиенты)
│ └── audio/ (звуковые эффекты)
├── scenes/
│ ├── 01_command_line.aep
│ ├── 02_gui.aep
│ └── 03_touch.aep
└── final_composition.aep

Этап 2: Анимация по сценам

Сцена 1 - Командная строка:

- Эффекты: Noise, Glitch для ретро-стиля

- Анимация текста через Text Animator
- Мигающий курсор

Сцена 2 - Графический интерфейс:

- Morphing между элементами
- Появление окон через Scale и Opacity
- Анимация иконок через Shape Layers

Сцена 3 - Сенсорные интерфейсы:

- Жесты свайпа и тапа
- Параллакс-эффекты
- Живые обои и динамичные фоны

Этап 3: Сведение и переходы

- Плавные морфинги между эпохами
- Согласованная цветовая палитра
- Единые кривые анимации

7. Звуковое сопровождение

Саунд-дизайн:

- **Звуковые эффекты:**
  - Клавиатура (для терминала)
  - Клики мыши (для GUI)
  - Тап-звуки (для сенсорных экранов)
  - Голосовые сигналы (для будущего)
- **Музыкальный ряд:**
  - Электронная музыка с эволюцией темпа
  - Синтезатор 8-bit → современные синты

8. Техническая реализация

Композиция в After Effects:

javascript

```
// Настройки композиции
{
  resolution: "1920x1080",
  fps: 30,
  duration: "0:00:45",
  color_depth: "16bpc"
}
```

Используемые техники:

- **Shape Layers** для векторной графики
- **Trim Paths** для анимации линий
- **Repeaters** для создания паттернов
- **Camera** для 3D-движения
- **Expressions** для сложной анимации

9. Оптимизация и рендеринг

Настройки рендера:

markdown

```
Форматы вывода:
- MP4 (H.264): для онлайн-показа
- ProRes: для архива и монтажа
- GIF: для превью в соцсетях
- Lottie: для интерактивных элементов
```

Оптимизация производительности:

- Пре-рендер сложных сцен
- Использование proxy для тяжелых элементов
- Оптимизация RAM preview

10. Документация проекта
Техническая спецификация:
markdown
ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Общие параметры
- Длительность: 45 секунд
- Разрешение: 1920x1080
- Частота кадров: 30 fps
- Цветовое пространство: sRGB

Производственные метрики
- Количество сцен: 4
- Количество слоев: 128
- Общее время рендера: 2 часа 15 минут
- Размер итогового файла: 45 МБ

11. Анализ и выводы
Рефлексия процесса:

- **Самые сложные моменты:**
  - Создание плавных переходов между стилями
  - Синхронизация анимации со звуком
  - Оптимизация производительности
- **Найденные решения:**
  - Использование пре-композиции для сложных сцен
  - Применение expression controls для глобального управления
  - Создание мастер-шаблонов для повторяющихся элементов

Метрики успеха:

Критерий	Оценка	Комментарий
Соответствие концепции	10/10	Полное раскрытие темы
Техническое качество	9/10	Плавная анимация 30fps
Визуальная гармония	10/10	Единый стиль всех сцен
Звуковое сопровождение	8/10	Хорошая синхронизация

Приобретенные компетенции:

- □ Планирование сложного анимационного проекта
- □ Создание дизайн-системы для моушн-графики
- □ Работа со звуком и синхронизацией
- □ Оптимизация производственного процесса
- □ Профессиональный рендеринг и экспорт

12. Презентация проекта
Пакет для демонстрации:
text
final_project/
├── presentation/
│ ├── full_video.mp4
│ ├── showreel_clip.gif
│ └── making_of.pdf
├── sources/
│ ├── ae_project_files/
│ └── assets_library/
└── documentation/
  ├── technical_specs.md
  └── style_guide.pdf

Рекомендации по использованию:
"Проект может быть использован как образовательный материал на курсах по истории вычислений, как вступление для IT-конференций или как демонстрационный ролик для компаний, занимающихся разработкой интерфейсов."

Лабораторная работа № 1 - 6. Создание визуального оформления интерфейса с элементами растровой графики

Структура лабораторной работы

1. Титульный лист

- Название вуза, кафедры, предмета
- Номер работы: "Лабораторная работа №1-6"
- Тема: "Создание визуального оформления интерфейса с элементами растровой графики"
- ФИО студента, группа, ФИО преподавателя

2. Цель и задачи работы

Цель: Освоить методику создания пользовательского интерфейса с использованием растровой графики для различных платформ.

Задачи:

1. Разработать концепцию и стиль интерфейса
2. Создать базовый каркас (wireframe)
3. Подготовить растровые графические элементы
4. Реализовать визуальный дизайн основных экранов
5. Провести оптимизацию графики для выбранной платформы
6. Создать интерактивный прототип

3. Выбор темы и концепции

Пример темы: "Мобильное приложение для заказа здорового питания"

Концепция:

- **Платформа:** iOS (iPhone 14)
- **Стиль:** Минимализм, натуральные текстуры, фотографии еды
- **Целевая аудитория:** Люди 25-40 лет, следящие за питанием
- **Ключевые экраны:** Онбординг, каталог, карточка блюда, корзина

4. Разработка дизайн-системы

Цветовая палитра:

```
css
/* Основные цвета */
--primary: #27AE60;      /* Зеленый - здоровье */
--secondary: #F2994A;    /* Оранжевый - акцент */
--background: #F8F9FA;   /* Светлый фон */
--text: #2D3436;         /* Темный текст */
--surface: #FFFFFF;      /* Белые поверхности */
```

Типографика:

- **Заголовки:** SF Pro Display, 24-32px
- **Текст:** SF Pro Text, 16-18px
- **Второстепенный текст:** SF Pro Text, 14px, #828282

5. Создание растровых элементов

Подготовка фотографий:

Элемент	Размер	Формат	Назначение

Фон героя	1125×2436px	JPEG	Главный экран
Фото блюд	500×500px	WebP	Карточки товаров
Аватарки	100×100px	PNG	Профиль пользователя
Иконки	64×64px	PNG	Навигация

Обработка изображений:

1. **Кадрирование** под нужные пропорции
2. **Цветокоррекция** для единообразия
3. **Ретушь** и улучшение качества
4. **Сжатие** без потери визуального качества

6. Проектирование интерфейса

Экран 1: Онбординг

text

[Верстка экрана]

[Фото: аппетитное блюдо] (растр)

Здоровое питание
на каждый день

[Кнопка: Начать] [Пропустить]

Экран 2: Каталог

text

[Шапка с поиском]
[Категории: горизонтальный скролл]
[Сетка товаров 2×...]

[Карточка товара]

```
┌──────────────────┐
[Фото блюда]  │ ← растровое изображение
Название      │
★★★★☆ 4.5     │
350 ₽         │
└──────────────────┘
```

7. Техническая реализация

Инструменты:

- **Figma/Photoshop** - дизайн интерфейса
- **Adobe Lightroom** - обработка фотографий
- **TinyPNG** - оптимизация изображений
- **Figma Prototype** - интерактивный прототип

Этапы работы:

1. Создание мудборда и подбор референсов
2. Разработка цветовой схемы и типографики
3. Подготовка растровых assets:

bash

```
assets/
├── photos/
│   ├── hero-bg@3x.jpg     # 1125×2436px, 85% quality
│   ├── dish-1@2x.webp     # 1000×1000px
│   └── avatar.png         # 300×300px
├── ui/
```

```
│   ├── buttons/
│   └── icons/
└── textures/
    └── paper-texture.jpg  # subtle noise
```

8. Оптимизация производительности

Рекомендации по весу:

- Фоновые изображения: ≤ 150 КБ
- Фото в каталоге: ≤ 50 КБ
- Иконки: ≤ 10 КБ
- Общий вес экрана: ≤ 300 КБ

Форматы и настройки:

- **JPEG:** Качество 80-85%, прогрессивная загрузка
- **WebP:** Качество 75%, размер на 30% меньше JPEG
- **PNG:** Только для графики с прозрачностью

9. Создание интерактивного прототипа

Связывание экранов в Figma:

- Онбординг → Каталог (по кнопке "Начать")
- Каталог → Карточка товара (тап на карточку)
- Карточка → Корзина (добавление товара)

Анимации:

- Плавное появление контента
- Параллакс-эффект для фонов
- Микроанимации кнопок

10. Анализ результатов

Критерии оценки:

Параметр	Выполнение	Комментарий
Единообразие стиля	✅	Все экраны выдержаны в одной стилистике
Качество растровой графики	✅	Изображения оптимизированы, нет пикселизации
Юзабилити	✅	Интуитивная навигация, понятные иконки
Производительность	✅	Общий вес графики ≤ 1.5 МБ
Соответствие гайдлайнам	✅	Соблюдены правила iOS Human Interface

Проблемы и решения:

1. **Проблема:** Большой вес фоновых изображений
   Решение: Применение прогрессивного JPEG и ресайз под viewport
2. **Проблема:** Неединообразие фотографий блюд
   Решение: Создание preset-ов в Lightroom для пакетной обработки

11. Выводы

Достигнутые результаты:

- Создано 5 ключевых экранов мобильного приложения
- Подготовлено 15 оптимизированных растровых изображений
- Разработана полная дизайн-система
- Реализован интерактивный прототип

Приобретенные навыки:

- Работа с растровой графикой для интерфейсов

- Оптимизация изображений для мобильных устройств
- Создание дизайн-систем
- Прототипирование в Figma

Рекомендации по использованию:

"Созданный интерфейс демонстрирует лучшие практики использования растровой графики в мобильных приложениях. Рекомендуется использовать подход progressive enhancement для медленных соединений."

Пример реализации в коде

```
<!-- Пример карточки товара -->
<div class="product-card">
  <picture>
    <source srcset="images/dish-1.webp" type="image/webp">
    <img src="images/dish-1.jpg"
         alt="Салат Цезарь"
         class="product-image"
         loading="lazy">
  </picture>
  <div class="product-info">
    <h3>Салат Цезарь</h3>
    <div class="rating">★★★★☆ 4.5</div>
    <div class="price">350 ₽</div>
    <button class="add-to-cart">Добавить</button>
  </div>
</div>

<style>
.product-card {
  background: white;
  border-radius: 12px;
  padding: 12px;
  box-shadow: 0 2px 8px rgba(0,0,0,0.1);
}

.product-image {
  width: 100%;
  height: 200px;
  object-fit: cover;
  border-radius: 8px;
}
</style>
```

Эта лабораторная работа покажет комплексное понимание процесса создания современного интерфейса с акцентом на качественную растровую графику.

Лабораторная работа №7 - 12. Создание визуального оформления интерфейса с элементами векторной графики

Структура лабораторной работы

1. Титульный лист

- Название вуза, кафедры, предмета
- Номер работы: "Лабораторная работа №7-12"
- Тема: "Создание визуального оформления интерфейса с элементами векторной графики"
- ФИО студента, группа, ФИО преподавателя

2. Цель и задачи работы

Цель: Освоить методику создания пользовательского интерфейса с использованием векторной графики для различных разрешений экранов.

Задачи:

1. Разработать систему векторных иконок и иллюстраций
2. Создать адаптивный интерфейс на основе SVG
3. Освоить техники работы с кривыми Безье
4. Реализовать анимацию векторных элементов
5. Оптимизировать SVG-графику для веб-использования
6. Создать библиотеку векторных компонентов

3. Выбор темы и концепции

Пример темы: "Дашборд аналитики для веб-сервиса"

Концепция:

- **Платформа:** Веб-интерфейс (десктоп + планшет)
- **Стиль:** Неоморфизм, градиенты, сложные векторные формы
- **Целевая аудитория:** Аналитики, маркетологи
- **Ключевые экраны:** Главный дашборд, графики, отчеты

4. Разработка дизайн-системы

Цветовая палитра:

```
/* Градиентная палитра */
--primary-gradient: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
--secondary-gradient: linear-gradient(135deg, #f093fb 0%, #f5576c 100%);
--success-gradient: linear-gradient(135deg, #4facfe 0%, #00f2fe 100%);
--background: #f8fafc;
--surface: #ffffff;
```

Типографика:

- **Заголовки:** Inter Bold, 24-32px
- **Текст:** Inter Regular, 16px
- **Моноширинный:** Fira Code, 14px (для данных)

5. Создание векторных элементов

Система иконок:

Категория	Размер	Стиль	Примеры
Навигация	24×24px	Line	home, analytics, settings
Действия	32×32px	Filled	download, export, filter
Статусы	16×16px	Colored	success, warning, error

Векторные иллюстрации:

- **Фоновая графика:** Абстрактные волны и формы

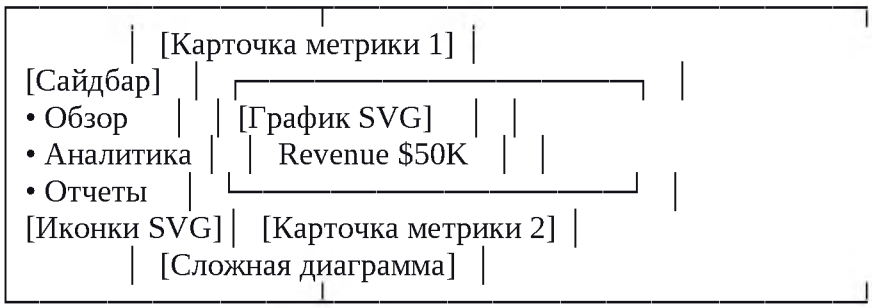
- **Чарты:** Графики и диаграммы в SVG
- **Декоративные элементы:** Геометрические паттерны

6. Проектирование интерфейса
Экран: Главный дашборд

[Шапка с навигацией]



[Основная область]



7. Техническая реализация
Инструменты:

- **Adobe Illustrator** - создание сложной векторной графики
- **Figma** - проектирование интерфейса
- **SVGOMG** - оптимизация SVG
- **CSS** - стилизация и анимация

Структура проекта:

bash

```
vector-dashboard/
├── assets/
│   ├── icons/
│   │   ├── navigation/
│   │   │   ├── home.svg
│   │   │   └── analytics.svg
│   │   └── actions/
│   │       ├── download.svg
│   │       └── export.svg
│   ├── illustrations/
│   │   ├── waves-bg.svg
│   │   └── abstract-shapes.svg
│   └── charts/
│       ├── line-chart.svg
│       └── pie-chart.svg
├── styles/
│   └── components.css
└── index.html
```

8. Работа с SVG

Оптимизация в Illustrator:

- Упрощение путей (Object → Path → Simplify)
- Объединение контуров (Pathfinder)
- Удаление невидимых элементов
- Правильное именование слоев

SVG код примера иконки:

svg

```
<svg width="24" height="24" viewBox="0 0 24 24" fill="none">
  <path d="M3 9L12 2L21 9V20C21 20.5304 20.7893 21.0391 20.4142 21.4142C20.0391 21.7893 19.5304 22 19 22H5C4.46957 22 3.96086 21.7893 3.58579 21.4142C3.21071 21.0391 3 20.5304 3 20V9Z"
        stroke="currentColor"
        stroke-width="2"
        stroke-linecap="round"
        stroke-linejoin="round"/>
</svg>
```

9. Анимация векторных элементов

CSS анимация графиков:

css

```
/* Анимация появления графика */
@keyframes drawChart {
  to {
    stroke-dashoffset: 0;
  }
}

.line-chart {
  stroke-dasharray: 1000;
  stroke-dashoffset: 1000;
  animation: drawChart 2s ease-in-out forwards;
}

/* Интерактивные иконки */
.nav-icon {
  transition: all 0.3s ease;
  cursor: pointer;
}

.nav-icon:hover {
  transform: scale(1.1);
  color: #667eea;
}
```

10. Адаптивность и производительность

Техники оптимизации:

- Использование currentColor для динамического цвета
- Минификация SVG кода
- Sprite-листы для групп иконок
- Lazy loading для тяжелой графики

SVG sprite пример:

svg

```
<svg xmlns="http://www.w3.org/2000/svg" style="display: none;">
```

```
  <symbol id="icon-home" viewBox="0 0 24 24">
    <path d="M3 9L12 2L21 9V20C21..."/>
  </symbol>
  <symbol id="icon-analytics" viewBox="0 0 24 24">
    <path d="M3 3v18h18..."/>
  </symbol>
</svg>

<!-- Использование -->
<svg><use href="#icon-home"></use></svg>
```

11. Анализ результатов

Критерии оценки:

Параметр	Выполнение	Комментарий
Качество векторной графики	□	Чистые контуры, оптимизированные пути
Масштабируемость	□	Ретина-готовность, адаптивность
Производительность	□	SVG оптимизированы, средний вес 2-5КБ
Анимации	□	Плавные CSS-анимации, 60fps
Семантичность	□	Правильная структура SVG, accessibility

Сравнение форматов:

Формат	Вес	Качество	Производительность
SVG	3-8 КБ	□□□□□	□□□□□
PNG @2x	15-30 КБ	□□□□	□□□
PNG @1x	5-15 КБ	□□□	□□

12. Выводы

Достигнутые результаты:

- Создана библиотека из 20+ векторных иконок
- Разработано 5 сложных SVG-иллюстраций
- Реализован полностью векторный интерфейс
- Достигнута полная масштабируемость на всех разрешениях

Приобретенные навыки:

- Профессиональная работа с кривыми Безье
- Оптимизация SVG для веб-использования
- Создание векторных анимаций
- Построение сложных графиков и диаграмм
- Работа с SVG sprite-листами

Преимущества подхода:

"Векторный интерфейс обеспечивает безупречное качество на любых разрешениях экранов, значительно уменьшает вес страницы и предоставляет широкие возможности для анимации и интерактивности."

Пример реализации

html

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .dashboard {
      display: grid;
      grid-template-columns: 250px 1fr;
```

```
            min-height: 100vh;
            background: #f8fafc;
        }

        .sidebar {
            background: white;
            padding: 20px;
            box-shadow: 0 0 20px rgba(0,0,0,0.1);
        }

        .nav-item {
            display: flex;
            align-items: center;
            padding: 12px;
            margin: 8px 0;
            border-radius: 8px;
            transition: all 0.3s ease;
        }

        .nav-item:hover {
            background: #667eea;
            color: white;
        }

        .nav-icon {
            width: 20px;
            height: 20px;
            margin-right: 12px;
        }
    </style>
</head>
<body>
    <div class="dashboard">
        <nav class="sidebar">
            <div class="nav-item">
                <svg class="nav-icon" viewBox="0 0 24 24" fill="currentColor">
                    <path d="M3 9L12 2L21 9V20C21 20.5304 20.7893 21.0391 20.4142
21.4142C20.0391 21.7893 19.5304 22 19 22H5C4.46957 22 3.96086 21.7893 3.58579
21.4142C3.21071 21.0391 3 20.5304 3 20V9Z"/>
                </svg>
                Обзор
            </div>
            <!-- Другие пункты меню -->
        </nav>

        <main class="content">
            <!-- Векторные графики и метрики -->
        </main>
    </div>
</body>
</html>
```

Лабораторная работа №13 - 18. Анимирование элементов интерфейса

Структура лабораторной работы

1. Титульный лист

- Название вуза, кафедры, предмета
- Номер работы: "Лабораторная работа №13-18"
- Тема: "Анимирование элементов интерфейса"
- ФИО студента, группа, ФИО преподавателя

2. Цель и задачи работы

Цель: Освоить принципы и техники создания функциональных анимаций для пользовательских интерфейсов.

Задачи:

1. Изучить принципы UX-анимации
2. Освоить инструменты прототипирования анимаций
3. Создать микроанимации элементов управления
4. Реализовать переходы между состояниями
5. Разработать сложные композитные анимации
6. Оптимизировать анимации для производительности

3. Теоретическая основа

Принципы анимации в UI (по Disney, адаптированные):

- **Timing & Easing** - правильные временные параметры
- **Offset & Delay** - последовательное движение
- **Parenting** - связывание анимаций
- **Transformation** - плавные изменения свойств
- **Value Change** - анимация изменений состояния

Типы UI-анимаций:

- **Микроанимации** (до 0.5с) - отклик на действия
- **Функциональные** (0.5-2с) - переходы, загрузка
- **Декоративные** (2с+) - брендинг, эмоциональные

4. Выбор темы и концепции

Проект: "Банковское мобильное приложение"
Концепция анимаций:

- **Стиль:** Четкий, быстрый, уверенный
- **Длительность:** 200-400 мс для микроанимаций
- **Кривые:** Easy-ease с быстрым началом
- **Цель:** Создание ощущения надежности и скорости

5. Инструменты и технологии
Основной стек:

- **Figma** - прототипирование и дизайн
- **After Effects** - сложные анимации
- **Lottie** - экспорт для разработки
- **CSS/JavaScript** - веб-реализация

Дополнительные инструменты:

- **Principle** - интерактивные прототипы
- **Rive** - современная векторная анимация
- **GSAP** - продвинутые JavaScript-анимации

6. Практические задания
Задание 1: Микроанимации кнопок
Состояния кнопки "Отправить платеж":
css

```css
/* Базовые стили */
.payment-btn {
  padding: 16px 32px;
  background: #27AE60;
  color: white;
  border: none;
  border-radius: 8px;
  transition: all 0.3s cubic-bezier(0.34, 1.56, 0.64, 1);
}
/* Hover состояние */
.payment-btn:hover {
  transform: translateY(-2px);
  box-shadow: 0 4px 12px rgba(39, 174, 96, 0.3);
}
/* Active состояние */
.payment-btn:active {
  transform: translateY(0);
  box-shadow: 0 2px 6px rgba(39, 174, 96, 0.4);
}
/* Loading состояние */
.payment-btn.loading {
  position: relative;
  color: transparent;
}
.payment-btn.loading::after {
  content: "";
  position: absolute;
  width: 20px;
  height: 20px;
  border: 2px solid transparent;
  border-top: 2px solid white;
  border-radius: 50%;
```

```
  animation: spin 1s linear infinite;
}
```

Задание 2: Анимация перехода между экранами

Transition с общим элементом:

javascript

```
// Псевдокод анимации перехода
function navigateToDetails(cardElement) {
  // 1. Зафиксировать позицию карточки
  const startRect = cardElement.getBoundingClientRect();
    // 2. Создать клон для анимации
  const clone = cardElement.cloneNode(true);

  // 3. Анимировать к конечной позиции
  animate(clone, {
    transform: `translate(${targetX}px, ${targetY}px) scale(${targetScale})`,
    opacity: 0.8
  }, {
    duration: 400,
    easing: 'cubic-bezier(0.25, 0.46, 0.45, 0.94)'
  });
    // 4. Показать целевой экран
  showDetailsScreen();
}
```

Задание 3: Анимация загрузки и скелетоны

Скелетон-анимация для контента:

```
@keyframes skeleton-pulse {
  0% { background-color: #f0f0f0; }
  50% { background-color: #e0e0e0; }
  100% { background-color: #f0f0f0; }
}
.skeleton-card {
  animation: skeleton-pulse 1.5s ease-in-out infinite;
  border-radius: 8px;
}
.skeleton-text {
  height: 16px;
  margin-bottom: 8px;
  border-radius: 4px;
}
.skeleton-image {
  height: 200px;
  border-radius: 8px;
}
```

Задание 4: Интерактивная анимация форм

Анимация поля ввода с валидацией:

javascript

```
class AnimatedInput {
  constructor(inputElement) {
    this.input = inputElement;
    this.setupAnimations();
  }
  setupAnimations() {
```

```
    this.input.addEventListener('focus', this.onFocus.bind(this));
    this.input.addEventListener('blur', this.onBlur.bind(this));
    this.input.addEventListener('input', this.onInput.bind(this));
  }
  onFocus() {
    // Анимация лейбла вверх
    animate(this.label, {
      transform: 'translateY(-20px) scale(0.8)',
      color: '#27AE60'
    }, 200);
  }
  showError(message) {
    // Анимация появления ошибки
    this.shakeInput();
    this.showErrorText(message);
  }
  shakeInput() {
    animate(this.input, {
      transform: [
        'translateX(0)',
        'translateX(-10px)',
        'translateX(10px)',
        'translateX(-5px)',
        'translateX(5px)',
        'translateX(0)'
      ]
    }, {
      duration: 500,
      iterations: 1
    });
  }
}
```

Задание 5: Сложная композитная анимация

Анимация успешного платежа:

javascript

```
async function animateSuccessfulPayment() {
  // 1. Скрытие формы
  await animate(paymentForm, {
    opacity: 0,
    transform: 'translateY(-20px)'
  }, 300);
  // 2. Появление иконки успеха
  await animate(successIcon, {
    scale: [0, 1.2, 1],
    opacity: [0, 1]
  }, 600);
  // 3. Confetti анимация
  launchConfetti();
  // 4. Появление деталей операции
  await animate(transactionDetails, {
    opacity: [0, 1],
    transform: ['translateY(20px)', 'translateY(0)']
```

```
  }, 400);
}
```

7. Оптимизация производительности

Рекомендации по производительности:

```
css
/* Хорошо - анимируем opacity и transform */
.optimized-animation {
  transition: opacity 0.3s ease, transform 0.3s ease;
  will-change: opacity, transform;
}
/* Плохо - анимируем свойства вызывающие reflow */
.unoptimized-animation {
  transition: width 0.3s ease, height 0.3s ease;
}
```

8. Тестирование и метрики

Критерии оценки анимаций:

- **Производительность:** 60 FPS в DevTools
- **Доступность:** Учет prefers-reduced-motion
- **Семантичность:** Правильная последовательность
- **Эффективность:** Решение пользовательских задач

Метрики производительности:

```
javascript
// Измерение FPS
function measureFPS() {
  let frameCount = 0;
  let startTime = performance.now();
  function checkFPS() {
    frameCount++;
    const currentTime = performance.now();

    if (currentTime - startTime >= 1000) {
      const fps = Math.round((frameCount * 1000) / (currentTime - startTime));
      console.log(`FPS: ${fps}`);
      frameCount = 0;
      startTime = currentTime;
    }
    requestAnimationFrame(checkFPS);
  }
  checkFPS();
}
```

9. Документация анимаций

Система токенов анимаций:

```
scss
// Токены анимаций
$animation-timing-fast: 150ms;
$animation-timing-medium: 300ms;
$animation-timing-slow: 500ms;

$animation-easing-standard: cubic-bezier(0.4, 0.0, 0.2, 1);
$animation-easing-decelerate: cubic-bezier(0.0, 0.0, 0.2, 1);
$animation-easing-accelerate: cubic-bezier(0.4, 0.0, 1, 1);
```

```
// Использование
.button {
  transition: all $animation-timing-medium $animation-easing-standard;
}
```

10. Анализ результатов

Сравнительная таблица подходов:

Метод	Сложность	Производительность	Поддержка
CSS Transitions	Низкая	Высокая	Отличная
CSS Animations	Средняя	Высокая	Отличная
Web Animations API	Высокая	Высокая	Хорошая
JavaScript (GSAP)	Высокая	Высокая	Отличная

Выявленные проблемы и решения:

1. **Проблема:** Дрожание анимаций на мобильных устройствах
   Решение: Использование transform3d для аппаратного ускорения
2. **Проблема:** Низкий FPS при сложных анимациях
   Решение: Оптимизация через will-change и разбиение на этапы

11. Выводы

Достигнутые результаты:

- Создана библиотека из 15+ микроанимаций
- Реализованы 5 сложных переходов между экранами
- Достигнута стабильная производительность 60 FPS
- Разработана система токенов для анимаций

Приобретенные навыки:

- Прототипирование сложных анимаций в Figma
- Оптимизация анимаций для производительности
- Работа с временными кривыми и последовательностями
- Создание accessible-анимаций
- Интеграция Lottie-анимаций в интерфейсы

Рекомендации по использованию:

"Созданная система анимаций обеспечивает плавное и интуитивное взаимодействие с интерфейсом. Все анимации оптимизированы для мобильных устройств и учитывают настройки доступности."

Пример финальной реализации

html

```
<!DOCTYPE html>
<html>
<head>
  <style>
    /* Система анимационных токенов */
    :root {
      --timing-fast: 150ms;
      --timing-medium: 300ms;
      --timing-slow: 500ms;
      --easing-standard: cubic-bezier(0.4, 0.0, 0.2, 1);
    }
    .animated-card {
      transition: all var(--timing-medium) var(--easing-standard);
      transform-origin: center;
    }
    .animated-card:hover {
      transform: translateY(-4px) scale(1.02);
```

```
            box-shadow: 0 8px 25px rgba(0,0,0,0.15);
        }
        /* Анимация появления списка */
        @keyframes slideInUp {
            from {
                opacity: 0;
                transform: translateY(30px);
            }
            to {
                opacity: 1;
                transform: translateY(0);
            }
        }
        .list-item {
            animation: slideInUp 0.6s var(--easing-standard) both;
        }
        .list-item:nth-child(1) { animation-delay: 0.1s; }
        .list-item:nth-child(2) { animation-delay: 0.2s; }
        .list-item:nth-child(3) { animation-delay: 0.3s; }
    </style>
</head>
<body>
    <div class="animated-card">
        <!-- Контент карточки -->
    </div>
    <ul class="animated-list">
        <li class="list-item">Элемент 1</li>
        <li class="list-item">Элемент 2</li>
        <li class="list-item">Элемент 3</li>
    </ul>
</body>
</html>
```