УДК 004.5

ПРОИЗВОДИТЕЛЬНОСТЬ ИНТЕРФЕЙСА В ЗАВИСИМОСТИ ОТ ГРАФИЧЕСКОГО ОФОРМЛЕНИЯ

♦♦♦♦♦

INTERFACE PERFORMANCE DEPENDING ON GRAPHIC DESIGN

Питкин Виктор Александрович
старший преподаватель
кафедры физического воспитания и спорта,
Кубанский государственный технологический университет
irvik25@mail.ru

Бритвина Александра Михайловна
студентка группы 25-КБ-ПИ2,
факультет информационных технологий
и кибербезопасности,
Кубанский государственный технологический университет
sashabritvina489@gmail.com

Аннотация. В статье рассматривается влияние графического оформления на производительность интерфейсов. Описан принцип рендеринга, включающий три этапа: обработку пользовательского ввода, пересчёт расположения элементов и финальную отрисовку. Показано, что наибольшую нагрузку создаёт пересчёт расположения (layout recalculation), тогда как трансформации на видеокарте обрабатываются быстрее. Выделены основные факторы, снижающие производительность: смена стилей, изменение геометрии элементов, а также смена текстур и режимов рисования. Описаны методы оптимизации – групповая отрисовка (батчинг), использование текстурных атласов и замена анимаций пересчёта на анимации трансформаций. Статья предназначена для разработчиков интерфейсов, веб-дизайнеров и специалистов по оптимизации графики.

Ключевые слова: производительность интерфейса, рендеринг, частота кадров, пересчёт расположения элементов, трансформации, батчинг, групповая отрисовка, оптимизация графики, UI-поток, команды отрисовки, графика.

Pitkin Viktor Alexandrovich
Senior Lecturer of the Department
of Physical Education and Sports,
Kuban State Technological University
irvik25@mail.ru

Britvina Alexandra Mikhailovna
Student of group 25-KB-PI2,
Faculty of Information Technology
and Cybersecurity,
Kuban State Technological University
sashabritvina489@gmail.com

Annotation. The article examines the influence of graphic design on interface performance. The rendering principle is described, including three stages: processing user input, recalculating the layout of elements, and final rendering. It is shown that the greatest load is created by layout recalculation, while transformations on the graphics card are processed faster. The main factors that reduce performance are identified: changing styles, changing element geometry, and changing textures and drawing modes. Optimization methods are described – batching, the use of texture atlases, and replacing layout animations with transformation animations. The article is intended for interface developers, web designers, and graphics optimization specialists.

Keywords: interface performance, rendering, frame rate, layout recalculation, transformations, batching, batch rendering, graphics optimization, UI thread, draw calls, graphics.

Производительность интерфейса – это способность графического пользовательского интерфейса (GUI) обеспечивать плавное отображение и мгновенную реакцию на действия пользователя при заданных параметрах графического оформления. Основным количественным показателем производительности является частота кадров (frame rate) – количество обновлений экрана в секунду. Снижение частоты кадров воспринимается пользователем как «тормоза», «зависания» и «дерганность» интерфейса [1].

Речь идёт о влиянии параметров графического оформления – количества теней, градиентов, скруглений, анимаций, текстур и режимов наложения – на вычислительную нагрузку центрального процессора (CPU) и графического процессора (GPU). Первые исследования зависимости производительности от графического оформления появились ещё в эпоху становления графических операционных систем (Windows 3.x, Mac OS Classic). Значительный прорыв произошёл в середине 2000-х годов с появлением аппаратно-ускоренных интерфейсов (Windows Aero, macOS Quartz Compositor). На сегодняшний день проблема производительности интерфейсов стала особенно актуальной в связи с распространением мобильных устройств с ограниченными ресурсами и веб-приложений со сложной анимацией [2].

Принцип работы

Рендеринг интерфейса включает три последовательных этапа, выполняемых в одном потоке – UI-потоке. Первый этап – обработка событий пользовательского ввода (клики, нажатия клавиш, движения мыши). Второй этап – пересчёт расположения элементов интерфейса (layout recalculation), в ходе которого система заново вычисляет позиции и размеры всех визуальных компонентов. Третий этап – отрисовка (rendering), при которой сформированное изображение отправляется на экран. Если суммарное время всех трёх этапов превышает временной бюджет одного кадра, происходит пропуск кадров, и интерфейс начинает «тормозить» [3].

Наибольшую вычислительную нагрузку создаёт второй этап – пересчёт расположения элементов. Изменение таких свойств, как ширина, высота, положение элемента (например, при выдвижении бокового меню или анимации появления), запускает полный перерасчёт геометрии всей иерархии элементов, включая те, которые визуально не изменились. Трансформации (сдвиг, поворот, масштабирование) обрабатываются непосредственно на GPU и не вызывают перерасчёта расположения, поэтому они значительно быстрее.

Влияние различных графических деталей на производительность можно описать следующими факторами:

1. Смена стиля. Каждый раз, когда у элемента меняется цвет, шрифт, тень или другое оформление, система заново просчитывает, как этот элемент должен выглядеть. Если внутри одного блока вложено много других элементов (например, меню с подпунктами), нагрузка становится выше.

2. Изменение геометрии. Когда у элемента меняется форма (скругление углов, толщина рамки или контур), системе приходится перестраивать его графическую основу с нуля.

3. Смена текстур и режимов рисования. Использование масок, плавных переходов цвета (градиентов), теней или полупрозрачности заставляет видеокарту каждый раз переключаться между разными способами рисования.

Преимущества и недостатки

К достоинствам минималистичного графического оформления относится высокая производительность даже на слабых устройствах, низкое энергопотребление и предсказуемое время отклика. Интерфейсы, использующие трансформации вместо пересчёта layout, способны поддерживать плавную анимацию при частоте 60 кадров в секунду без пропусков.

Главным недостатком сложного графического оформления является очень высокая вычислительная нагрузка. Один анимированный элемент с изменяемой шириной может вызывать пересчёт расположения всех элементов на странице, что при глубокой вложенности приводит к заметным задержкам. Частая смена текстур и режимов наложения увеличивает количество draw calls (команд отрисовки), а каждое переключение состояния GPU стоит времени. При анимации скруглений или границ происходит постоянная перестройка вершинных буферов, что также создаёт избыточную нагрузку. Для решения этих проблем применяют групповую отрисовку (батчинг), использование текстурных атласов и замену анимаций layout на анимации трансформаций [3].

Современные возможности

Современные реализации интерфейсов включают гибридный подход: для статичных элементов используется минимальный пересчёт, а для анимированных трансформации на GPU. Групповая отрисовка позволяет объединять визуальные элементы с одинаковыми текстурами и шейдерами в одну команду отрисовки, что значительно снижает нагрузку на CPU. Текстурные атласы – объединение множества мелких иконок в одно большое изображение, сохраняют эффективность батчинга даже при использовании разных исходных картинок. Эти технологии широко применяются в мобильных приложениях (Android Compose, SwiftUI) и игровых движках (Unity, Unreal Engine) [4].

Одними из самых ярких примеров влияния графического оформления на производительность являются современные веб-порталы с тяжёлой анимацией, например лендинги студий веб-дизайна, где множество анимированных элементов с тенями и

градиентами вызывают заметные тормоза на устройствах среднего уровня. Обратным примером можно считать интерфейс текстового редактора iA Writer, где минималистичное оформление и полное отсутствие лишних анимаций обеспечивают мгновенный отклик даже на самых слабых компьютерах. Технология активного мониторинга производительности (Realtime Performance Monitoring) позволяет в реальном времени отслеживать проблемные места интерфейса и выдавать рекомендации по упрощению графического оформления [1].

Производительность интерфейса остаётся одной из ключевых характеристик качества программного продукта. С развитием аппаратного обеспечения (более быстрые GPU, высокая частота обновления экранов) и алгоритмов оптимизации (батчинг, текстурные атласы, асинхронный рендеринг) метод продолжает активно развиваться. Комплексный подход, включающий разумное ограничение теней и градиентов, использование трансформаций вместо пересчёта layout и применение групповой отрисовки, позволяет значительно повысить производительность интерфейса и сделать его плавным даже на устройствах с ограниченными ресурсами.

Литература

1. Готская И.Б. Универсальный подход к мониторингу производительности веб-графики в информационно-технологической инфраструктуре / И.Б. Готская, В.С. Абрамов // Современные наукоемкие технологии. – 2025. – № 4. – С. 24–29.
2. Ловец В.В. Оптимизация графики мобильных устройств / В.В. Ловец, А.С. Волчек // Радиотехника и электроника: Сборник материалов 61-й научной конференции. – Мн. : БГУИР, 2025. – С. 88–89.
3. Курта П.А. Система статистического измерения атомарной эффективности графических элементов интерфейсов // Труды учебных заведений связи. – 2024. – Т. 10. – № 6. – С. 99–110.
4. Морунов А.Д. Алгоритмы сглаживания изображения для систем с несколькими графическими адаптерами / А.Д. Морунов, И.А. Карасева, А.М. Суворов // Сборник тезисов докладов конгресса молодых ученых. – СПб. : Университет ИТМО, 2026.

References

1. Gotskaya I.B. A universal approach to monitoring the performance of web graphics in information technology infrastructure / I.B. Gotskaya, V.S. Abramov // Modern High Technologies. – 2025. – № 4. – P. 24–29.
2. Lovets V.V. Optimization of graphics for mobile devices / V.V. Lovets, A.S. Volchek // Radio Engineering and Electronics: Proceedings of the 61st Scientific Conference. – Mn. : BSUIR, 2025. – P. 88–89.
3. Kurta P.A. A system for statistical measurement of the atomic efficiency of graphical interface elements // Proceedings of Telecommunication Universities. – 2024. – Vol. 10. – № 6. – P. 99–110.
4. Morunov A.D. Image smoothing algorithms for systems with multiple graphics adapters / A.D. Morunov, I.A. Karaseva, A.M. Suvorov // Collection of abstracts of the Congress of Young Scientists. – SPb. : ITMO University, 2026.