

Методы оптимизации производительности мобильных интерфейсов

Николаев К. В.

Научный руководитель: Филиппов В. Б.

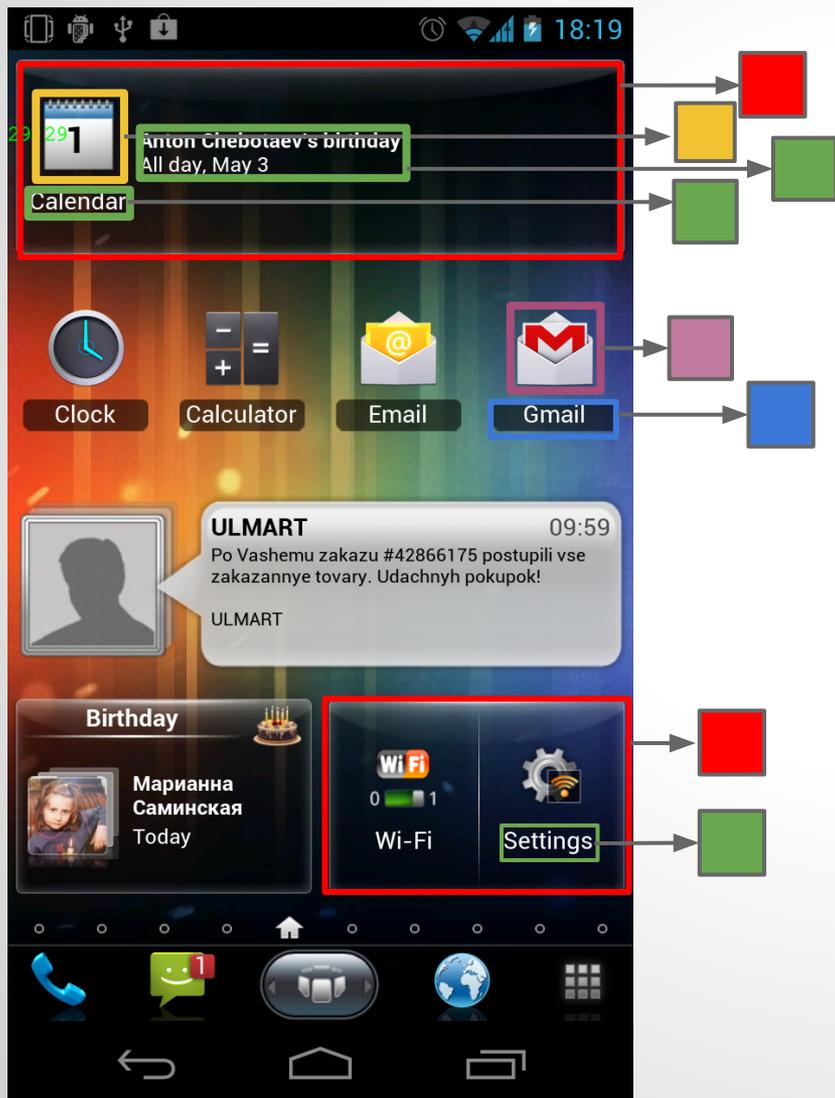
1 Мобильные интерфейсы

- OpenGL.
- Цель: 60 кадров в секунду.
- 2d с элементами 3d.

Каждый объект:

- Вершины,
- Материал:
 - текстура,
 - шейдер,
- `glDrawElements`, `glDrawArrays`.

2 Идея



- Много простых объектов.
- Используют общие материалы.

Число объектов: N .

Время на отрисовку: $O(N)$.

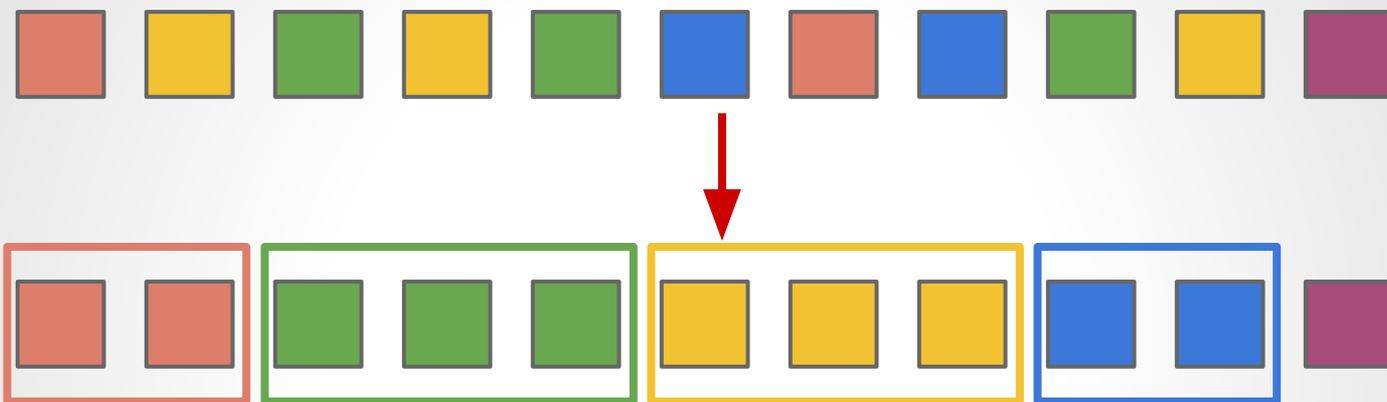
Время на отрисовку одного объекта: $O(\log N)$

(Слабо зависит от числа вершин).

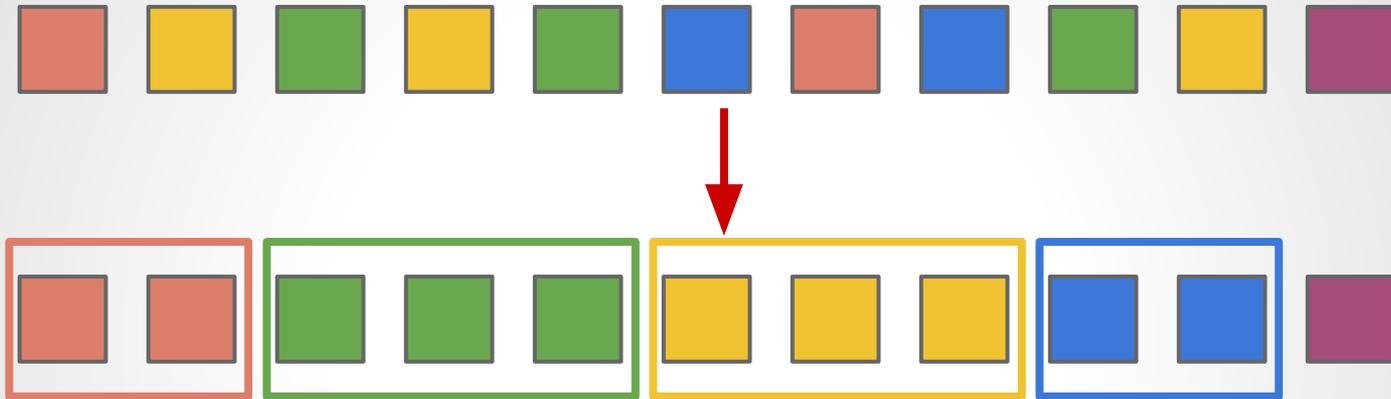
Изменение состояния (текстуры, шейдера) — дорого.

Идея: группировать и объединять похожие объекты.

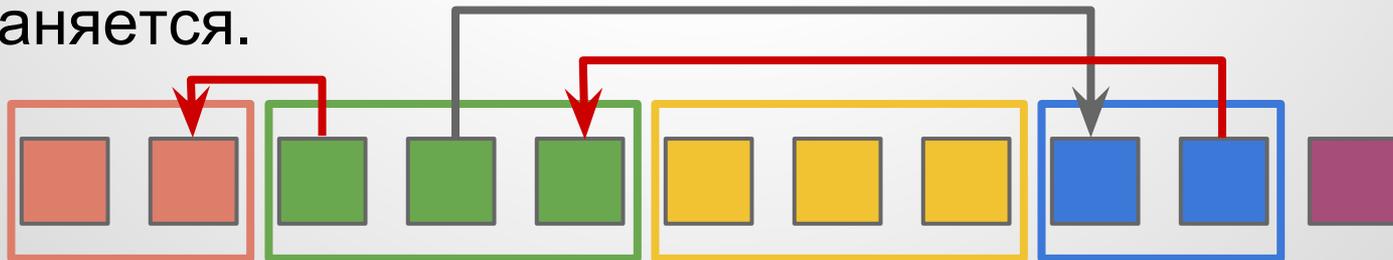
3 Наивный метод



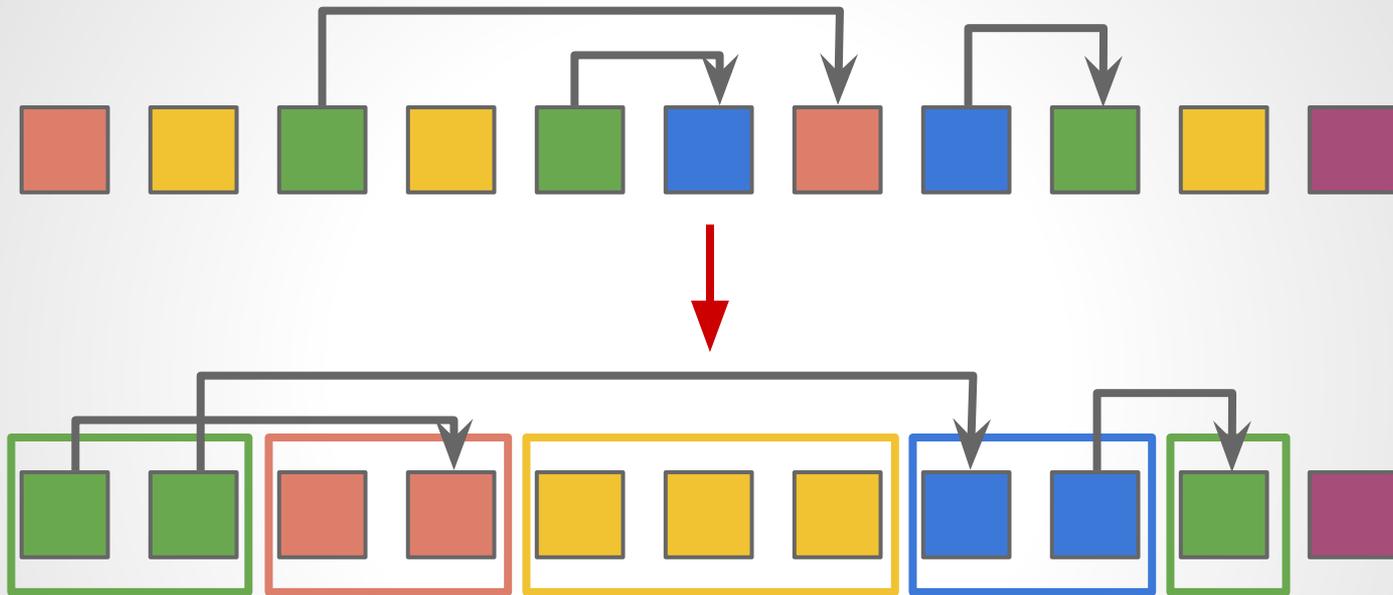
4 Наивный метод



Проблема: объекты зависят друг от друга.
Смешивание полупрозрачных объектов.
После трансформации порядок отрисовки не сохраняется.



5 Учет порядка



Зависимости соблюдены.

Зеленая группа разорвана, но это нормально.

Однако такие случаи необходимо минимизировать.

6 Формулировка задачи

$A = \{ a_1, a_2, \dots, a_n \}$ — объекты.

$M = \{ \mu_1, \mu_2, \dots, \mu_m \}$ — классы (материалы).

$\varphi: A \rightarrow M$ — соответствие объектов классам.

$\leq = \{ (a_i, \beta_i) \mid a_i \in A, \beta_i \in A, i=1..k \}^+$ — отношение зависимости (частичный порядок).

Необходимо найти такой порядок объектов a_1, a_2, \dots, a_n , такой что из зависимости $a_i \leq a_j$ следует $i < j$, и одновременно мощность множества точек смен материала $|\{ i \mid \varphi(a_i) \neq \varphi(a_{i+1}) \}|$ минимальна.

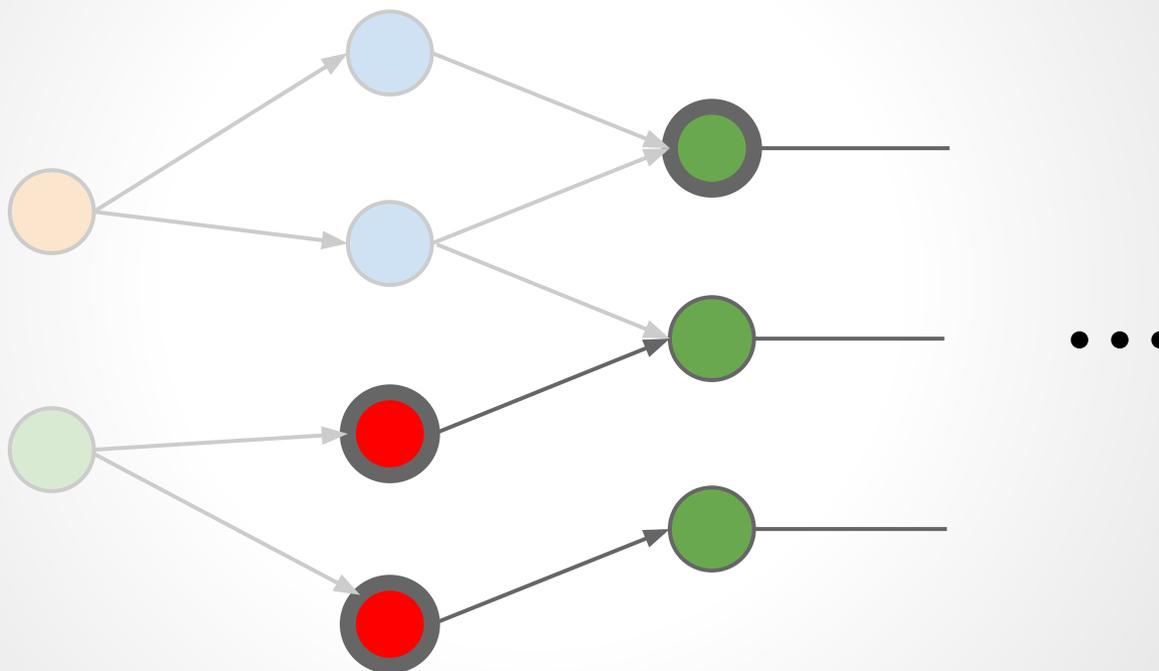
7 Возможные подходы к решению

- *NPO*-трудная.
 - Shortest Common Supersequence.
 - Множество-носитель — материалы.
 - Элементы — элементы рисования с соответствующим материалом.
- Ограничения:
 - $T < 1/60$ с,
 - $N \sim 100\text{--}200$.
- Подходы:
 - построение графа частичных решений (EХР),
 - неполный граф частичных решений,
 - жадный алгоритм.

8 Жадный алгоритм

1. Поддерживается список групп независимых необработанных объектов по материалам.
2. На каждом шаге выбрать:
 - a. группу, содержащую все объекты соответствующего материала;
 - b. иначе, группу максимального размера.
3. Выбранную группу отправить на вывод.
4. Обновить группы независимых объектов.

9 Шаг работы алгоритма



10 Полученные результаты

- Подходит для структур интерфейсов.
- Преобразование структуры автоматическое, прозрачное для пользователя движка.
- В этом главное отличие от других решений, например, Cocos2d, Ogre3d, Irrlicht.
- Позволило увеличить частоту кадров до 50–60 в секунду на слабых устройствах.

12 Результаты тестирования

На сложных сценах:

- было 20 fps (кадров в секунду);
- стало 60 fps.

На обычных сценах:

- запас по производительности;
- возможность использовать более сложную и богатую графику.

Спасибо за внимание

Буду рад ответить на ваши вопросы.