

# Лекция №6

Плоские объекты

Вывод текста

Оптимизация

Рябинин Константин Валентинович

e-mail: [icosaeder@ya.ru](mailto:icosaeder@ya.ru)

jabber: [icosaeder@jabber.ru](mailto:icosaeder@jabber.ru)

Пермь, 2011

- ➔ Большинство систем трёхмерной визуализации технически не разделяют 2D и 3D графику, всегда работая в трёхмерном пространстве, однако
- При помощи изменения проекции и положения объектов можно добиться эффекта двумерности
- «Двумерная» графика полезна в той же степени, что и трёхмерная
- Как правило на высоком уровне вводится разделение двумерной и трёхмерной сцен, которые, однако, могут быть выведены одновременно
- При этом некоторые объекты двумерной сцены могут попадать в трёхмерную и наоборот

**Оформление экрана** – неофициальное название для двумерной графики, располагающейся поверх трёхмерной сцены, как будто бы наклеенной на объектив камеры

- Используется для отображения контекстной информации (подсказки, индикаторы состояния и т.п.)
- Технически представляет собой набор изображений
- К примеру, в OpenGL изображения могут быть выведены как текстуры на простых полигональных структурах (прямоугольниках), которые рисуются
  - **после** отображения трёхмерной сцены
  - **с отключенным z-буфером**
  - **с иным способом проектирования**
- Проекция выбирается таким образом, чтобы система координат сцены была эквивалентна системе координат окна, тип проекции – ортографическая  
`gluOrtho2D(0, win_width, 0, win_height);`
- То, что вывод организуется в последнюю очередь, гарантирует корректное применение alpha-смешивания

**Спрайт** – это двумерное изображение, которое может свободно перемещаться по экрану

**Биллборд** (разновидность спрайта) – это двумерный объект на трёхмерной сцене

- Идея: фрагмент плоскости (как правило – прямоугольник, представленный двумя треугольниками) располагается на сцене и поворачивается всегда так, чтобы быть **перпендикулярным оси взгляда**
- На этот фрагмент накладывается текстура, изображающая некоторый объёмный объект
- Таким образом достигается эффект присутствия на сцене объёмного объекта с минимальными затратами на его вывод

- Биллборды, как правило, используются для
  - второстепенных объектов
  - объектов фона (облака, луна, солнце, ...)
  - массовых мелких объектов (трава, листья, ...)
  - световых эффектов (вспышки, искры, молнии, ореолы света, ...)  
=> часто формируют системы частиц
- удалённых объектов (приём оптимизации – по мере приближения объекты могут заменяться на объёмные модели). В этом случае биллборд носит название **импостер**
- Для спрайтов актуально alpha-смешивание, поэтому при их использовании необходимо осуществлять сортировку по удалённости и выводить сначала самые далёкие объекты, а затем – более близкие

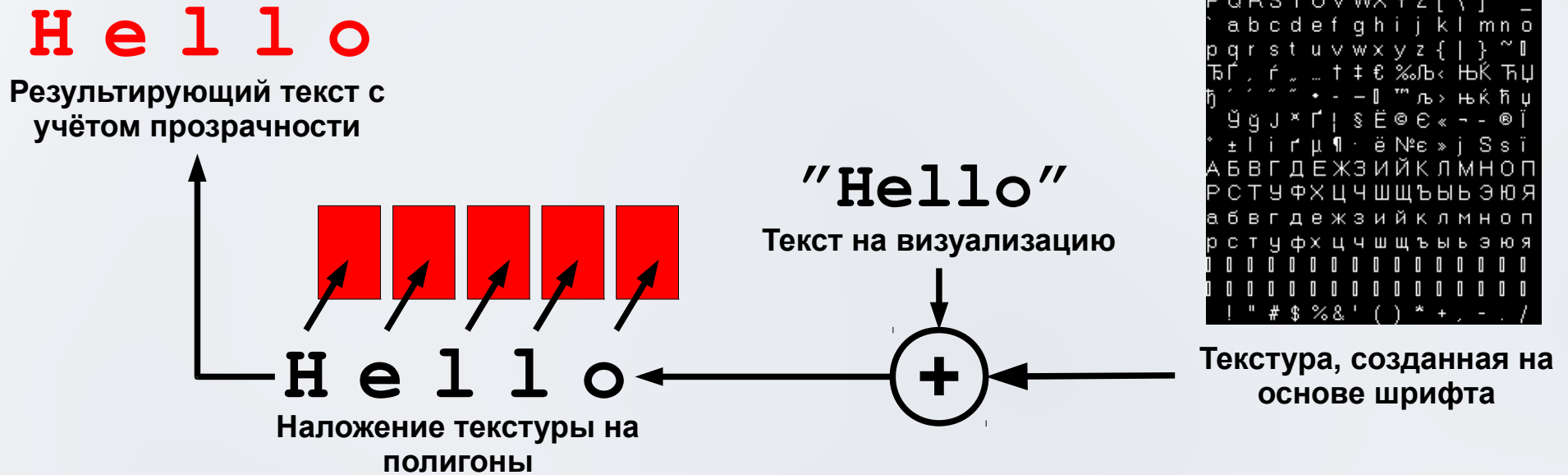
- Модификаци биллборда – «крестовина», трёхмерный объект, составленный из двух перпендикулярных друг другу прямоугольников, на которые наложена одинаковая текстура
  - В старых компьютерных играх очень часто таким образом создавались деревья
- «Крестовина» уже не изменяет своего поворота, либо изменяет так, чтобы медиана угла между прямоугольниками совпадала с осью взгляда
- Подобные объекты, хотя и не обеспечивают высокой реалистичности картинки, позволяют сэкономить огромное количество ресурсов
- Сейчас «крестовины» почти не используются – разве что в качестве импостеров

- Спрайты на трёхмерной сцене также могут быть ориентированы и не на камеру, а на какой-либо объект. Например, при моделировании кругов на воде спрайт располагается параллельно поверхности воды

- Текст – неотъемлемая составляющая мультимедийной системы
- Атомарная единица текста – **графема** (буква алфавита или иной символ)
- Атомарная графическая единица текста – **глиф** (конкретное графическое представление графемы)
- **Глиф != печатный знак**  
Глиф – атомарная структура, некоторые печатные знаки состоят из нескольких глифов, а некоторые глифы включают в себя несколько знаков, существующих по отдельности в других глифах
- Вывод текста представляет собой отрисовку соответствующих глифов, из которых складываются печатные знаки
- Множество конкретных глифов называется **шрифтом**



- В мультимедийных системах шрифт может быть как двумерным, так и трёхмерным (глифы имеют глубину)
- Трёхмерный шрифт:
  - набор трёхмерных моделей, соответствующих глифам
- Двумерный шрифт:
  - набор растровых изображений
  - набор векторных описаний
- Наиболее популярны векторные шрифты, так как допускают произвольное масштабирование
- Перед выводом глифы векторного шрифта должны быть растеризованы под выбранный размер
- Наиболее популярный формат хранения векторных шрифтов – TrueType (\*.ttf)



- В более сложном случае растровое изображение генерируется динамически по файлу векторного шрифта
- Для генерации используются специализированные библиотеки, например FreeType
- Следует организовывать «ассоциативный» массив, позволяющий быстро вывести нужный символ

## Оптимизация на уровне простых приёмов

- Разумная экономия на разрешении моделей и текстур
- Переиспользование данных моделей и текстур
- Соблюдение правила степени двойки для текстур
- Использование исключительно треугольников в качестве полигонов

Для OpenGL:

- При использовании `glBegin(GL_TRIANGLES) ... glEnd()` между этими вызовами структура кода должна быть как можно более простой, т. е.
  - Без вызовов процедур
  - Без циклов
  - С минимумом условных переходов
  - Лучше всего – только перечисление точек с их атрибутами

**Как правило библиотеки визуализации предоставляют различные средства для оптимизации хранения и вывода геометрии**

**В OpenGL:**

- **glBegin ... glEnd приемлем только для очень простых тестовых сцен. В новых версиях спецификации OpenGL эти функции удалены**
- **Списки отображения**
- **Массивы вершин**
- **Буферные объекты**

**Список отображения** – это средство объединения нескольких команд OpenGL в «пакет», который затем может быть отправлен на конвейер одной командой

- Упрощение отрисовки объекта
- Ускорение отрисовки объекта
- Невозможность изменения атрибутов объекта
- Поддерживается не всеми драйверами
- Удалено в новой версии OpenGL

```
GLuint listBase = glGenLists(1);  
glNewList(listBase, GL_COMPILE);  
constructObject(); // Вызов всех функций,  
                   // обеспечивающих отрисовку объекта  
glEndList();  
.  
.  
.  
glCallList(listBase);
```

**Массив вершин** – это средство для вывода последовательности вершин, атрибуты которых сохранены в одном или нескольких линейных массивах

- Эффективное хранение за счёт возможности нелинейной индексации
- Значительно более быстрая обработка по сравнению с `glBegin ... glEnd`
- Удобно использовать для вывода загруженных из файлов моделей

```
glEnableClientState(GL_VERTEX_ARRAY);  
glEnableClientState(GL_NORMAL_ARRAY);  
glEnableClientState(GL_TEXTURE_COORD_ARRAY);  
glVertexPointer(3, GL_FLOAT, 0, verices);  
glNormalPointer(GL_FLOAT, 0, normals);  
glTexCoordPointer(2, GL_FLOAT, 0, textureMap);  
glDrawElements(GL_TRIANGLES, polygonCount * 3,  
               GL_UNSIGNED_SHORT, indices);
```

**Буферные объекты** – это средство для хранения и быстрого вывода вершин, сгруппированных со всеми своими атрибутами

- Эффективное хранение на основе массивов
- Быстрая обработка (за счёт уменьшение числа вызовов функций)
- Сочетают в себе достоинства списков отображения и массивов вершин, устраняя недостатки
- В последней версии OpenGL буферные объекты – основное средство вывода геометрии
- В старых версиях доступны через расширение

→ [http://www.songho.ca/opengl/gl\\_vbo.html](http://www.songho.ca/opengl/gl_vbo.html)