

# Компьютерная графика

## Лекция 6: Ошибки OpenGL, расширения OpenGL, blending, освещение

2021

# Ошибки OpenGL

- ▶ Часто драйвер может понять, что в определённом OpenGL-вызове содержится ошибка

# Ошибки OpenGL

- ▶ Часто драйвер может понять, что в определённом OpenGL-вызове содержится ошибка
  - ▶ `glTexImage2D` с параметром `target`, не являющимся одним из типов текстур

# Ошибки OpenGL

- ▶ Часто драйвер может понять, что в определённом OpenGL-вызове содержится ошибка
  - ▶ `glTexImage2D` с параметром `target`, не являющимся одним из типов текстур
  - ▶ `glTexParameteri` устанавливающая `mag filter` в что-то отличное от `GL_LINEAR` или `GL_NEAREST`

# Ошибки OpenGL

- ▶ Часто драйвер может понять, что в определённом OpenGL-вызове содержится ошибка
  - ▶ `glTexImage2D` с параметром `target`, не являющимся одним из типов текстур
  - ▶ `glTexParameteri` устанавливающая `mag filter` в что-то отличное от `GL_LINEAR` или `GL_NEAREST`
  - ▶ `glBindBuffer` с ID буфера, который не был получен через `glGenBuffers`

# Ошибки OpenGL

- ▶ Часто драйвер может понять, что в определённом OpenGL-вызове содержится ошибка
  - ▶ `glTexImage2D` с параметром `target`, не являющимся одним из типов текстур
  - ▶ `glTexParameteri` устанавливающая `mag filter` в что-то отличное от `GL_LINEAR` или `GL_NEAREST`
  - ▶ `glBindBuffer` с ID буфера, который не был получен через `glGenBuffers`
  - ▶ etc.

# Ошибки OpenGL

- ▶ Часто драйвер может понять, что в определённом OpenGL-вызове содержится ошибка
  - ▶ `glTexImage2D` с параметром `target`, не являющимся одним из типов текстур
  - ▶ `glTexParameteri` устанавливающая `mag filter` в что-то отличное от `GL_LINEAR` или `GL_NEAREST`
  - ▶ `glBindBuffer` с ID буфера, который не был получен через `glGenBuffers`
  - ▶ etc.
- ▶ В таких случаях генерируется ошибка как особое значение типа `GLenum`

# Ошибки OpenGL

- ▶ Часто драйвер может понять, что в определённом OpenGL-вызове содержится ошибка
  - ▶ `glTexImage2D` с параметром `target`, не являющимся одним из типов текстур
  - ▶ `glTexParameteri` устанавливающая `mag filter` в что-то отличное от `GL_LINEAR` или `GL_NEAREST`
  - ▶ `glBindBuffer` с ID буфера, который не был получен через `glGenBuffers`
  - ▶ etc.
- ▶ В таких случаях генерируется ошибка как особое значение типа `GLenum`
- ▶ Как правило, вызвавшая ошибку операция не выполняется



# Ошибки OpenGL

- ▶ Часто драйвер может понять, что в определённом OpenGL-вызове содержится ошибка
  - ▶ `glTexImage2D` с параметром `target`, не являющимся одним из типов текстур
  - ▶ `glTexParameteri` устанавливающая `mag filter` в что-то отличное от `GL_LINEAR` или `GL_NEAREST`
  - ▶ `glBindBuffer` с ID буфера, который не был получен через `glGenBuffers`
  - ▶ etc.
- ▶ В таких случаях генерируется ошибка как особое значение типа `GLenum`
- ▶ Как правило, вызвавшая ошибку операция не выполняется
- ▶ `glGetError()` - получить значение ошибки, если она была

# Ошибки OpenGL

Возможные значения ошибок:

- ▶ `GL_NO_ERROR` - ошибки нет

# Ошибки OpenGL

Возможные значения ошибок:

- ▶ `GL_NO_ERROR` - ошибки нет
- ▶ `GL_INVALID_ENUM` - недопустимое значения перечисления (например, `GL_TEXTURE_2D` как первый параметр `glBindBuffer`)

# Ошибки OpenGL

Возможные значения ошибок:

- ▶ `GL_NO_ERROR` - ошибки нет
- ▶ `GL_INVALID_ENUM` - недопустимое значения перечисления (например, `GL_TEXTURE_2D` как первый параметр `glBindBuffer`)
- ▶ `GL_INVALID_VALUE` - недопустимое значение числового аргумента (например, отрицательная ширина текстуры в `glTexImage2D`)

# Ошибки OpenGL

Возможные значения ошибок:

- ▶ `GL_NO_ERROR` - ошибки нет
- ▶ `GL_INVALID_ENUM` - недопустимое значения перечисления (например, `GL_TEXTURE_2D` как первый параметр `glBindBuffer`)
- ▶ `GL_INVALID_VALUE` - недопустимое значение числового аргумента (например, отрицательная ширина текстуры в `glTexImage2D`)
- ▶ `GL_INVALID_OPERATION` - недопустимая операция (например, `glUniform1i` при отсутствии текущей шейдерной программы)

# Ошибки OpenGL

Возможные значения ошибок:

- ▶ `GL_NO_ERROR` - ошибки нет
- ▶ `GL_INVALID_ENUM` - недопустимое значения перечисления (например, `GL_TEXTURE_2D` как первый параметр `glBindBuffer`)
- ▶ `GL_INVALID_VALUE` - недопустимое значение числового аргумента (например, отрицательная ширина текстуры в `glTexImage2D`)
- ▶ `GL_INVALID_OPERATION` - недопустимая операция (например, `glUniform1i` при отсутствии текущей шейдерной программы)
- ▶ `GL_INVALID_FRAMEBUFFER_OPERATION` - операция рисования/чтения пикселей с невалидным framebuffer'ом (о них мы поговорим позже)

# Ошибки OpenGL

Возможные значения ошибок:

- ▶ `GL_NO_ERROR` - ошибки нет
- ▶ `GL_INVALID_ENUM` - недопустимое значения перечисления (например, `GL_TEXTURE_2D` как первый параметр `glBindBuffer`)
- ▶ `GL_INVALID_VALUE` - недопустимое значение числового аргумента (например, отрицательная ширина текстуры в `glTexImage2D`)
- ▶ `GL_INVALID_OPERATION` - недопустимая операция (например, `glUniform1i` при отсутствии текущей шейдерной программы)
- ▶ `GL_INVALID_FRAMEBUFFER_OPERATION` - операция рисования/чтения пикселей с невалидным framebuffer'ом (о них мы поговорим позже)
- ▶ `GL_OUT_OF_MEMORY` - закончилась память на GPU (может быть вызвана любой OpenGL-командой, обычно не обрабатывается)

# Ошибки OpenGL

- ▶ Документация к каждой функции описывает все возможные ошибки, которые эта функция может вызвать, и в каких случаях



# Ошибки OpenGL

- ▶ Документация к каждой функции описывает все возможные ошибки, которые эта функция может вызвать, и в каких случаях
  - ▶ N.B. `GL_OUT_OF_MEMORY` нигде не указан, потому что может появиться где угодно

# Ошибки OpenGL

- ▶ Документация к каждой функции описывает все возможные ошибки, которые эта функция может вызвать, и в каких случаях
  - ▶ N.B. `GL_OUT_OF_MEMORY` нигде не указан, потому что может появиться где угодно
- ▶ Не все ошибки покрываются этим механизмом (например: `glDrawArrays` с правильно настроенным VAO но с пустым VBO с вершинами)

# Ошибки OpenGL

- ▶ Документация к каждой функции описывает все возможные ошибки, которые эта функция может вызвать, и в каких случаях
  - ▶ N.B. `GL_OUT_OF_MEMORY` нигде не указан, потому что может появиться где угодно
- ▶ Не все ошибки покрываются этим механизмом (например: `glDrawArrays` с правильно настроенным VAO но с пустым VBO с вершинами)
- ▶ Нет информации о том, какая именно функция вызвала ошибку (любая из тех, что были вызваны до `glGetError`)

# Ошибки OpenGL

- ▶ Документация к каждой функции описывает все возможные ошибки, которые эта функция может вызвать, и в каких случаях
  - ▶ N.B. `GL_OUT_OF_MEMORY` нигде не указан, потому что может появиться где угодно
- ▶ Не все ошибки покрываются этим механизмом (например: `glDrawArrays` с правильно настроенным VAO но с пустым VBO с вершинами)
- ▶ Нет информации о том, какая именно функция вызвала ошибку (любая из тех, что были вызваны до `glGetError`)
  - ▶ ⇒ Придётся расставлять проверку после каждого OpenGL-вызова

# Ошибки OpenGL

- ▶ Драйвер может хранить один флаг ошибки, и пропускать все последующие ошибки до вызова `glGetError`

# Ошибки OpenGL

- ▶ Драйвер может хранить один флаг ошибки, и пропускать все последующие ошибки до вызова `glGetError`
- ▶ Драйвер может хранить несколько флагов, и `glGetError` возвращает и очищает любой из них

# Ошибки OpenGL

- ▶ Драйвер может хранить один флаг ошибки, и пропускать все последующие ошибки до вызова `glGetError`
- ▶ Драйвер может хранить несколько флагов, и `glGetError` возвращает и очищает любой из них
- ▶  $\Rightarrow$  Чтобы очистить ошибки, нужно вызывать `glGetError` в цикле:

```
while (glGetError() != GL_NO_ERROR);
```

# Ошибки OpenGL

- ▶ OpenGL 4.3+: более удобный механизм, `glDebugMessageCallback`



# Ошибки OpenGL

- ▶ OpenGL 4.3+: более удобный механизм, `glDebugMessageCallback`
- ▶ [khronos.org/opengl/wiki/OpenGL\\_Error](http://khronos.org/opengl/wiki/OpenGL_Error)
- ▶ [docs.gl/gl3/glGetError](http://docs.gl/gl3/glGetError)