

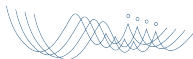
Render plugins

À. Vinacua, C. Andújar i professors de Gràfics

novembre de 2019

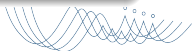
Tipus de plugins

(es tracta d'una distinció semàntica: tant sols hi ha una interfície, comuna a tots els "tipus")



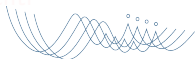
Tipus de plugins

- Effect Plugins
 - Canvien l'estat d'OpenGL abans i/o després de que es pinti l'escena.
 - Exemples: activar shaders, configurar textures, alpha blending. . .
- Draw Plugins (Sols un serà actiu)
 - Recorren els objectes per pintar les primitives de l'escena.
 - Exemples: dibuixar amb vertex arrays. . .
- Action Plugins
 - Executen accions arbitràries en resposta a events (mouse, teclat).
 - Exemples: selecció d'objectes, control de la càmera virtual. . .
- Render Plugins (Sols un serà actiu)
 - Dibuir un frame amb un o més passos de rendering.
 - Exemples: múltiples passos de rendering, shadow mapping. . .



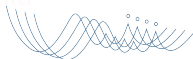
Fluxe de control

- Quan es carrega un nou plugin, es crida el seu `onPluginLoad()`
- Quan s'afegeix un nou model a l'escena es crida a `onObjectAdd()` de tots els plugins carregats
- Quan s'esborra l'escena, es crida a `onSceneClear()` de tots els plugins carregats
- Els events de ratolí i teclat (`keyPressEvent()`... `mouseMoveEvent()`...) es propaguen a tots els plugins carregats
- `GLWidget::paintGL()` crida:
 - `bind()` dels shaders per defecte
 - `setUniformValue()` pels uniforms que fan servir els shaders per defecte
 - `preFrame()` de tots els plugins
 - `paintGL()` del darrer plugin carregat que l'implementi
 - `postFrame()` de tots els plugins



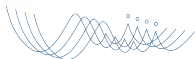
Fluxe de control

- Quan es carrega un nou plugin, es crida el seu `onPluginLoad()`
- Quan s'afegeix un nou model a l'escena es crida a `onObjectAdd()` de tots els plugins carregats
- Quan s'esborra l'escena, es crida a `onSceneClear()` de tots els plugins carregats
- Els events de ratolí i teclat (`keyPressEvent()`... `mouseMoveEvent()`...) es propaguen a tots els plugins carregats
- `GLWidget::paintGL()` crida:
 - `bind()` dels shaders per defecte
 - `setUniformValue()` pels uniforms que fan servir els shaders per defecte
 - `preFrame()` de tots els plugins
 - `paintGL()` del darrer plugin carregat que l'implementi
 - `postFrame()` de tots els plugins



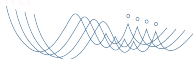
Fluxe de control

- Quan es carrega un nou plugin, es crida el seu `onPluginLoad()`
- Quan s'afegeix un nou model a l'escena es crida a `onObjectAdd()` de tots els plugins carregats
- Quan s'esborra l'escena, es crida a `onSceneClear()` de tots els plugins carregats
- Els events de ratolí i teclat (`keyPressEvent()`... `mouseMoveEvent()`...) es propaguen a tots els plugins carregats
- `GLWidget::paintGL()` crida:
 - `bind()` dels shaders per defecte
 - `setUniformValue()` pels uniforms que fan servir els shaders per defecte
 - `preFrame()` de tots els plugins
 - `paintGL()` del darrer plugin carregat que l'implementi
 - `postFrame()` de tots els plugins



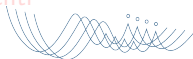
Fluxe de control

- Quan es carrega un nou plugin, es crida el seu `onPluginLoad()`
- Quan s'afegeix un nou model a l'escena es crida a `onObjectAdd()` de tots els plugins carregats
- Quan s'esborra l'escena, es crida a `onSceneClear()` de tots els plugins carregats
- Els events de ratolí i teclat (`keyPressEvent()`... `mouseMoveEvent()`...) es propaguen a tots els plugins carregats
- `GLWidget::paintGL()` crida:
 - `bind()` dels shaders per defecte
 - `setUniformValue()` pels uniforms que fan servir els shaders per defecte
 - `preFrame()` de tots els plugins
 - `paintGL()` del darrer plugin carregat que l'implementi
 - `postFrame()` de tots els plugins



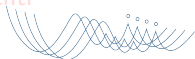
Fluxe de control

- Quan es carrega un nou plugin, es crida el seu `onPluginLoad()`
- Quan s'afegeix un nou model a l'escena es crida a `onObjectAdd()` de tots els plugins carregats
- Quan s'esborra l'escena, es crida a `onSceneClear()` de tots els plugins carregats
- Els events de ratolí i teclat (`keyPressEvent()`... `mouseMoveEvent()`...) es propaguen a tots els plugins carregats
- `GLWidget::paintGL()` crida:
 - `bind()` dels shaders per defecte
 - `setUniformValue()` pels uniforms que fan servir els shaders per defecte
 - `preFrame()` de tots els plugins
 - `paintGL()` del **darrer plugin carregat que l'implementi**
 - `postFrame()` de tots els plugins



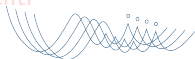
Fluxe de control

- Quan es carrega un nou plugin, es crida el seu `onPluginLoad()`
- Quan s'afegeix un nou model a l'escena es crida a `onObjectAdd()` de tots els plugins carregats
- Quan s'esborra l'escena, es crida a `onSceneClear()` de tots els plugins carregats
- Els events de ratolí i teclat (`keyPressEvent()`... `mouseMoveEvent()`...) es propaguen a tots els plugins carregats
- `GLWidget::paintGL()` crida:
 - `bind()` dels shaders per defecte
 - `setUniformValue()` pels uniforms que fan servir els shaders per defecte
 - `preFrame()` de tots els plugins
 - `paintGL()` del **darrer plugin carregat que l'implementi**
 - `postFrame()` de tots els plugins



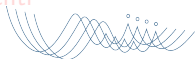
Fluxe de control

- Quan es carrega un nou plugin, es crida el seu `onPluginLoad()`
- Quan s'afegeix un nou model a l'escena es crida a `onObjectAdd()` de tots els plugins carregats
- Quan s'esborra l'escena, es crida a `onSceneClear()` de tots els plugins carregats
- Els events de ratolí i teclat (`keyPressEvent()`... `mouseMoveEvent()`...) es propaguen a tots els plugins carregats
- `GLWidget::paintGL()` crida:
 - `bind()` dels shaders per defecte
 - `setUniformValue()` pels uniforms que fan servir els shaders per defecte
 - `preFrame()` de tots els plugins
 - `paintGL()` del **darrer plugin carregat que l'implementi**
 - `postFrame()` de tots els plugins



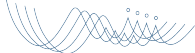
Fluxe de control

- Quan es carrega un nou plugin, es crida el seu `onPluginLoad()`
- Quan s'afegeix un nou model a l'escena es crida a `onObjectAdd()` de tots els plugins carregats
- Quan s'esborra l'escena, es crida a `onSceneClear()` de tots els plugins carregats
- Els events de ratolí i teclat (`keyPressEvent()`... `mouseMoveEvent()`...) es propaguen a tots els plugins carregats
- `GLWidget::paintGL()` crida:
 - `bind()` dels shaders per defecte
 - `setUniformValue()` pels uniforms que fan servir els shaders per defecte
 - `preFrame()` de tots els plugins
 - `paintGL()` del **darrer plugin carregat que l'implementi**
 - `postFrame()` de tots els plugins



Fluxe de control

- Quan es carrega un nou plugin, es crida el seu `onPluginLoad()`
- Quan s'afegeix un nou model a l'escena es crida a `onObjectAdd()` de tots els plugins carregats
- Quan s'esborra l'escena, es crida a `onSceneClear()` de tots els plugins carregats
- Els events de ratolí i teclat (`keyPressEvent()`... `mouseMoveEvent()`...) es propaguen a tots els plugins carregats
- `GLWidget::paintGL()` crida:
 - `bind()` dels shaders per defecte
 - `setUniformValue()` pels uniforms que fan servir els shaders per defecte
 - `preFrame()` de tots els plugins
 - `paintGL()` del **darrer plugin carregat que l'implementi**
 - `postFrame()` de tots els plugins



Fluxe de control

- Quan es carrega un nou plugin, es crida el seu `onPluginLoad()`
- Quan s'afegeix un nou model a l'escena es crida a `onObjectAdd()` de tots els plugins carregats
- Quan s'esborra l'escena, es crida a `onSceneClear()` de tots els plugins carregats
- Els events de ratolí i teclat (`keyPressEvent()`... `mouseMoveEvent()`...) es propaguen a tots els plugins carregats
- `GLWidget::paintGL()` crida:
 - `bind()` dels shaders per defecte
 - `setUniformValue()` pels uniforms que fan servir els shaders per defecte
 - `preFrame()` de tots els plugins
 - `paintGL()` del **darrer plugin carregat que l'implementi**
 - `postFrame()` de tots els plugins

