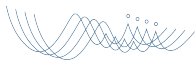


Laboratori de Gràfics, Action plugins.

À. Vinacua, C. Andújar i professors de Gràfics

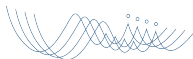
3 de desembre de 2013

Tipus de plugins



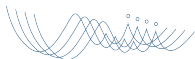
Tipus de plugins

- 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
 - Recórren els objectes per pintar les primitives de l'escena.
 - Exemples: dibuixar amb glBegin/glEnd, 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
 - Dibuixar un frame amb un o més passos de rendering.
 - Exemples: múltiples passos de rendering, shadow mapping...



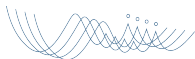
Tipus de plugins

- 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
 - Recórren els objectes per pintar les primitives de l'escena.
 - Exemples: dibuixar amb glBegin/glEnd, 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
 - Dibuixar un frame amb un o més passos de rendering.
 - Exemples: múltiples passos de rendering, shadow mapping...



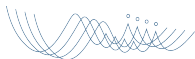
Tipus de plugins

- 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
 - Recórren els objectes per pintar les primitives de l'escena.
 - Exemples: dibuixar amb glBegin/glEnd, 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
 - Dibuir un frame amb un o més passos de rendering.
 - Exemples: múltiples passos de rendering, shadow mapping...

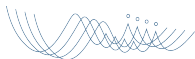


Tipus de plugins

- 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
 - Recórren els objectes per pintar les primitives de l'escena.
 - Exemples: dibuixar amb glBegin/glEnd, 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
 - Dibuixar un frame amb un o més passos de rendering.
 - Exemples: múltiples passos de rendering, shadow mapping...



Aquesta sessió: Action plugins



Action plugins

Mètodes propis

- virtual void keyPressEvent (QKeyEvent *)
- virtual void keyReleaseEvent (QKeyEvent *)
- virtual void mouseMoveEvent (QMouseEvent *)
- virtual void mousePressEvent (QMouseEvent *)
- virtual void mouseReleaseEvent (QMouseEvent *)
- virtual void wheelEvent (QWheelEvent *)

Mètodes/Atributs heredats

- virtual void onPluginLoad()
- virtual void onObjectAdd()
- GLWidget* pglwidget; // dóna accés a l'escena i la càmera
- pglwidget->scene()->objects().size() // num objectes
- pglwidget->camera()->getObs() // pos de l'observador

Action plugins

Mètodes propis

- virtual void keyPressEvent (QKeyEvent *)
- virtual void keyReleaseEvent (QKeyEvent *)
- virtual void mouseMoveEvent (QMouseEvent *)
- virtual void mousePressEvent (QMouseEvent *)
- virtual void mouseReleaseEvent (QMouseEvent *)
- virtual void wheelEvent (QWheelEvent *)

Mètodes/Atributs heredats

- virtual void onPluginLoad()
- virtual void onObjectAdd()
- GLWidget* pglwidget; // dóna accés a l'escena i la càmera
- pglwidget->scene()->objects().size() // num objectes
- pglwidget->camera()->getObs() // pos de l'observador

Action plugins

Mètodes propis

- virtual void keyPressEvent (QKeyEvent *)
- virtual void keyReleaseEvent (QKeyEvent *)
- virtual void mouseMoveEvent (QMouseEvent *)
- virtual void mousePressEvent (QMouseEvent *)
- virtual void mouseReleaseEvent (QMouseEvent *)
- virtual void wheelEvent (QWheelEvent *)

Mètodes/Atributs heredats

- virtual void onPluginLoad()
- virtual void onObjectAdd()
- GLWidget* pglwidget; // dóna accés a l'escena i la càmera
- pglwidget->scene()->objects().size() // num objectes
- pglwidget->camera()->getObs() // pos de l'observador

Action plugins

Mètodes propis

- virtual void keyPressEvent (QKeyEvent *)
- virtual void keyReleaseEvent (QKeyEvent *)
- virtual void mouseMoveEvent (QMouseEvent *)
- virtual void mousePressEvent (QMouseEvent *)
- virtual void mouseReleaseEvent (QMouseEvent *)
- virtual void wheelEvent (QWheelEvent *)

Mètodes/Atributs heredats

- virtual void onPluginLoad()
- virtual void onObjectAdd()
- GLWidget* pglwidget; // dóna accés a l'escena i la càmera
- pglwidget->scene()->objects().size() // num objectes
- pglwidget->camera()->getObs() // pos de l'observador

Action plugins

Mètodes propis

- virtual void keyPressEvent (QKeyEvent *)
- virtual void keyReleaseEvent (QKeyEvent *)
- virtual void mouseMoveEvent (QMouseEvent *)
- virtual void mousePressEvent (QMouseEvent *)
- virtual void mouseReleaseEvent (QMouseEvent *)
- virtual void wheelEvent (QWheelEvent *)

Mètodes/Atributs heredats

- virtual void onPluginLoad()
- virtual void onObjectAdd()
- GLWidget* pglwidget; // dóna accés a l'escena i la càmera
- pglwidget->scene()->objects().size() // num objectes
- pglwidget->camera()->getObs() // pos de l'observador

Action plugins

Mètodes propis

- virtual void keyPressEvent (QKeyEvent *)
- virtual void keyReleaseEvent (QKeyEvent *)
- virtual void mouseMoveEvent (QMouseEvent *)
- virtual void mousePressEvent (QMouseEvent *)
- virtual void mouseReleaseEvent (QMouseEvent *)
- virtual void wheelEvent (QWheelEvent *)

Mètodes/Atributs heredats

- virtual void onPluginLoad()
- virtual void onObjectAdd()
- GLWidget* pglwidget; // dóna accés a l'escena i la càmera
- pglwidget->scene()->objects().size() // num objectes
- pglwidget->camera()->getObs() // pos de l'observador

Action plugins

Mètodes propis

- virtual void keyPressEvent (QKeyEvent *)
- virtual void keyReleaseEvent (QKeyEvent *)
- virtual void mouseMoveEvent (QMouseEvent *)
- virtual void mousePressEvent (QMouseEvent *)
- virtual void mouseReleaseEvent (QMouseEvent *)
- virtual void wheelEvent (QWheelEvent *)

Mètodes/Atributs heredats

- virtual void onPluginLoad()
- virtual void onObjectAdd()
- GLWidget* pglwidget; // dóna accés a l'escena i la càmera
- pglwidget->scene()->objects().size() // num objectes
- pglwidget->camera()->getObs() // pos de l'observador

Action plugins

Mètodes propis

- virtual void keyPressEvent (QKeyEvent *)
- virtual void keyReleaseEvent (QKeyEvent *)
- virtual void mouseMoveEvent (QMouseEvent *)
- virtual void mousePressEvent (QMouseEvent *)
- virtual void mouseReleaseEvent (QMouseEvent *)
- virtual void wheelEvent (QWheelEvent *)

Mètodes/Atributs heredats

- virtual void onPluginLoad()
- virtual void onObjectAdd()
- GLWidget* pglwidget; // dóna accés a l'escena i la càmera
- pglwidget->scene()->objects().size() // num objectes
- pglwidget->camera()->getObs() // pos de l'observador

Action plugins

Mètodes propis

- virtual void keyPressEvent (QKeyEvent *)
- virtual void keyReleaseEvent (QKeyEvent *)
- virtual void mouseMoveEvent (QMouseEvent *)
- virtual void mousePressEvent (QMouseEvent *)
- virtual void mouseReleaseEvent (QMouseEvent *)
- virtual void wheelEvent (QWheelEvent *)

Mètodes/Atributs heredats

- virtual void onPluginLoad()
- virtual void onObjectAdd()
- GLWidget* pglwidget; // **dóna accés a l'escena i la càmera**
 - pglwidget->scene()->objects().size() // num objectes
 - pglwidget->camera()->getObs() // pos de l'observador

Action plugins

Mètodes propis

- virtual void keyPressEvent (QKeyEvent *)
- virtual void keyReleaseEvent (QKeyEvent *)
- virtual void mouseMoveEvent (QMouseEvent *)
- virtual void mousePressEvent (QMouseEvent *)
- virtual void mouseReleaseEvent (QMouseEvent *)
- virtual void wheelEvent (QWheelEvent *)

Mètodes/Atributs heredats

- virtual void onPluginLoad()
- virtual void onObjectAdd()
- GLWidget* pglwidget; // dóna accés a l'escena i la càmera
- pglwidget->scene()->objects().size() // num objectes
- pglwidget->camera()->getObs() // pos de l'observador

Action plugins

Mètodes propis

- virtual void keyPressEvent (QKeyEvent *)
- virtual void keyReleaseEvent (QKeyEvent *)
- virtual void mouseMoveEvent (QMouseEvent *)
- virtual void mousePressEvent (QMouseEvent *)
- virtual void mouseReleaseEvent (QMouseEvent *)
- virtual void wheelEvent (QWheelEvent *)

Mètodes/Atributs heredats

- virtual void onPluginLoad()
- virtual void onObjectAdd()
- GLWidget* pglwidget; // dóna accés a l'escena i la càmera
- pglwidget->scene()->objects().size() // num objectes
- pglwidget->camera()->getObs() // pos de l'observador

Fluxe de control

Per cada refresc:

- Si hi han registrats EffectPlugins, es crida el mètode preFrame() de cadascun.
- Si n'hi ha, es crida (l'únic) RenderPlugin
- Si hi han registrats EffectPlugins, es crida el mètode postFrame() de cadascun.

Tractament d'esdeveniments

Es propaguen als ActionPlugins que hi hagi registrats:

```
1 ...  
2 void GLWidget::mousePressEvent( QMouseEvent *e)  
3 {  
4     for (unsigned int i=0; i<actionPlugins.size(); ++i)  
5         qobject_cast<ActionInterface*>  
6             (actionPlugins[i]->instance())->mousePressEvent(e);  
7 }  
8 ...
```

