Animated UI technologies in Maemo 5 Fremantle

Kate Alhola 6.4.2009

Forum NOKIA

Introduction

Kate Alhola

- Maemo Chief Engineer at Forum Nokia
- Participating Qt maemo port project, several maemo projects in garage.maemo.org
- Long term Open Source developer, first contributions 8-bit microprocessor in early 80's
- Linux kernel driver from early 1.x kernels
- Katix RTOS with IP stack for PC, 68K and PPC
- Multiple GUI applications with Qt and GTK (and X11/Athena, Motif ...)
- Before Nokia, long career embedded Linux and RTOS related development in small subcontractor companies
- Numerous embedded HW designs

Forum Nokia

- Nokia 3rd party developers organization
- Consultancy, marketing, business development, university support, technical support, developer events, prototype loaning¸etc
-

Forum NOKIA

Maemo 5 Fremantle

- New generation of Maemo
- New UI style
- Animated Ui technoligies
- Qt4.5 from Forum Nokia
- Compositeing window manager
- Qt4.5 from Forum Nokia
- Some new API's
- Accelerated graphics with OpenGL-ES2
- High resolution camera
- Omap3 architecture
- HSPA (High Speed Packet Access / Cellular 3G connectivity)
-

Forum NOKIA

Animated UI technologies

- UI animation technolgies introcuced in maemo 5
    - Hildon desktop compositing window manager
    - OpenGL-ES2.0
    - Clutter
    - Qt QGraphicsView
    - Qt QGlWidget
- Some experimental stuff
    - Qt Kinetic
    -

compositing window manager

- Desktop Effects and applications switching
- Effects works with all existing applications, applications are not affected at all
- Application window is rendered into off screen window
- Window manager renders and transforms windows with OpenGL transformations
- Can be rotated, blurred, mirrored, dimmed only imagination is limit
- Common technology in power desktops like Macintosh Quartz, Linux Compiz ....

Qt4.5 for Fremantle

- Qt4.5 Fremantle port released 2.3
  - Home qt4.garage.maemo.org
  - Based on Qt4.5 rc 1
  - Hildon input method (auto completion not implemented)
  - Hildon menus
  - QGTK/Hildon Style (some things needs improvement)
  - OpenGL-ES2.0 support
  - Lot of Fremantle related fixes and workarounds
  - Installable from maemo.org extras-devel repository
- Version based on Qt 4.5 final release soon

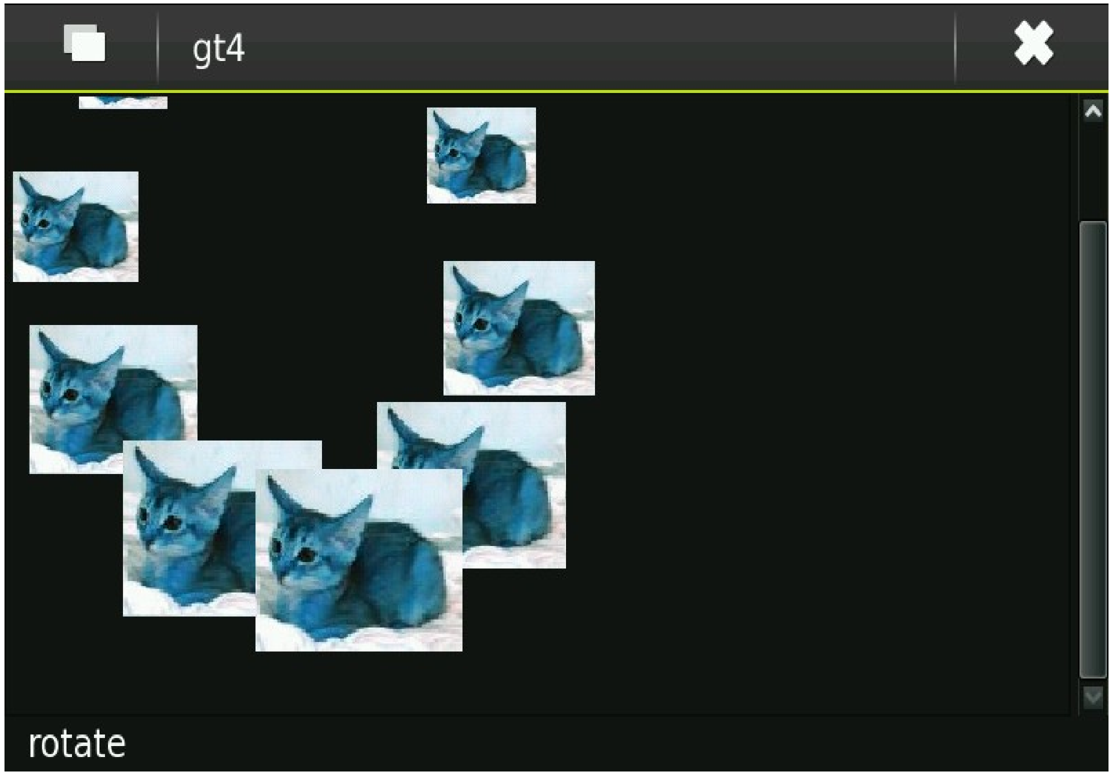
Qt GraphicsView

- Qt way to implement animated UI
- Similar functionality than Clutter
- GraphicsView is a low level toolkit which does not contain any high level widgets people are used to. You can make QGraphicsItems to behave like animated buttons for example or you can embed 2D QT Widgets but they lack lack animated behavior
- You have to manage the layout manually with coordinates, it is not automated
-

Forum NOKIA

GraphicsView for animatd UI

- QGraphicsScene, QGraphicsItem and QtimeLine
- Supports both OpenGL for desktop and OpenGL-ES in mobile
- GraphicsItems are objects that behave in stage
  - Move, rotate, scale, change opacity .....
  - Textures, SVG, vectors
- Timelines and events and events control behavior
- Normal 2D Qt widgets can be embedded and transformed as GraphicsItem



Forum NOKIA

Simple Qt graphicsview example

```
#include <Qapplication>
#include <QgraphicsScene>
#include <QgraphicsView>
#include <QgraphicsPixmapItem>

int main( int argc, char **argv )
{
    QApplication app(argc, argv);
    QGraphicsScene *scene = new QgraphicsScene(0,0,800,480);
    QGraphicsView  *view  = new QgraphicsView();
    view->setScene(scene);
    QGraphicsPixmapItem *pixitem =
        new QgraphicsPixmapItem(QPixmap("aureo100.jpg"));
    pixitem->setPos(100,100);
    scene->addItem(pixitem);
    view->show();
    return app.exec();
}
```

Forum NOKIA

QGraphicsWiew Example

```
RPixmap::RPixmap(const QPixmap &pixmap): QGraphicsPixmapItem(pixmap)
{
    setAcceptsHoverEvents(true);

    timeLine.setDuration(180*4);
    timeLine.setFrameRange(0, 180*4);
    connect(&timeLine, SIGNAL(frameChanged(int)), this, SLOT(setFrame(int)));
}

void RPixmap::setFrame(int frame)
{
    QPointF center = boundingRect().center();
    resetMatrix();
    int rof=50;

    setTransform(QTransform().translate(rof,rof).rotate(frame /
4.0,Qt::XAxis).translate(-rof,-rof).scale(0.1 , 0
.1 ));
}

void RPixmap::mousePressEvent ( QGraphicsSceneMouseEvent * event )
{
  if (timeLine.state() == QTimeLine::NotRunning)
    timeLine.start();

}
```

Making them go around

```
gtMainWindow::gtMainWindow()
   {
     QToolBar *ctrl=addToolBar("ctrl");
     QAction *rotQtAct = new QAction("rotate", this);
     ctrl->addAction(rotQtAct);
     connect(rotQtAct, SIGNAL(triggered()), this, SLOT(rotateEvent()));
    scene = new QgraphicsScene(0,0,800,480);
     view  = new QgraphicsView();
     view->setScene(scene);
     int i;
     for(i=0;i<10;i++) {
       pixitems[i] = new QGraphicsPixmapItem(QPixmap("aureo100.jpg"));
       scene->addItem(pixitems[i]);
     };
     setFrame(0);
     setCentralWidget(view);
     timeLine.setDuration(1800*4);
     timeLine.setFrameRange(0, 180*4);
     timeLine.start();
     connect(&timeLine, SIGNAL(frameChanged(int)), this, SLOT(setFrame(int)));
   }
```

Forum NOKIA

Timeline

```
• QPoint gtMainWindow::cposition(double n)
  {
     return Qpoint(80.0*sin(n/100.0*3.141592653),100);
  };
    double gtMainWindow::csize(double n)
  {
    return(1.0+0.5*cos(n/100*3.141592653));
  }
  void gtMainWindow::setFrame(int frame)
  {
    int i;
    for(i=0;i<10;i++) {
    pixitems[i]->resetTransform();
    pixitems[i]->scale(csize(i*20+frame/5),csize(i*20+frame/5));
    pixitems[i]->setZValue(csize(i*20+frame/5));
    pixitems[i]->setPos(cposition(i*20+frame/5)*2+QPoint(150,150));
  }
  void gtMainWindow::rotateEvent()
  {
    if (timeLine.state() == QtimeLine::NotRunning)    timeLine.start();
  }
```

Demo videos

-
- Graphicsview widget demo
- http://www.youtube.com/watch?v=MXS3xKV-UM0&feature=channel_pag
- http://www.youtube.com/watch?v=eJcTBJaPRZg&feature=channel_page
-

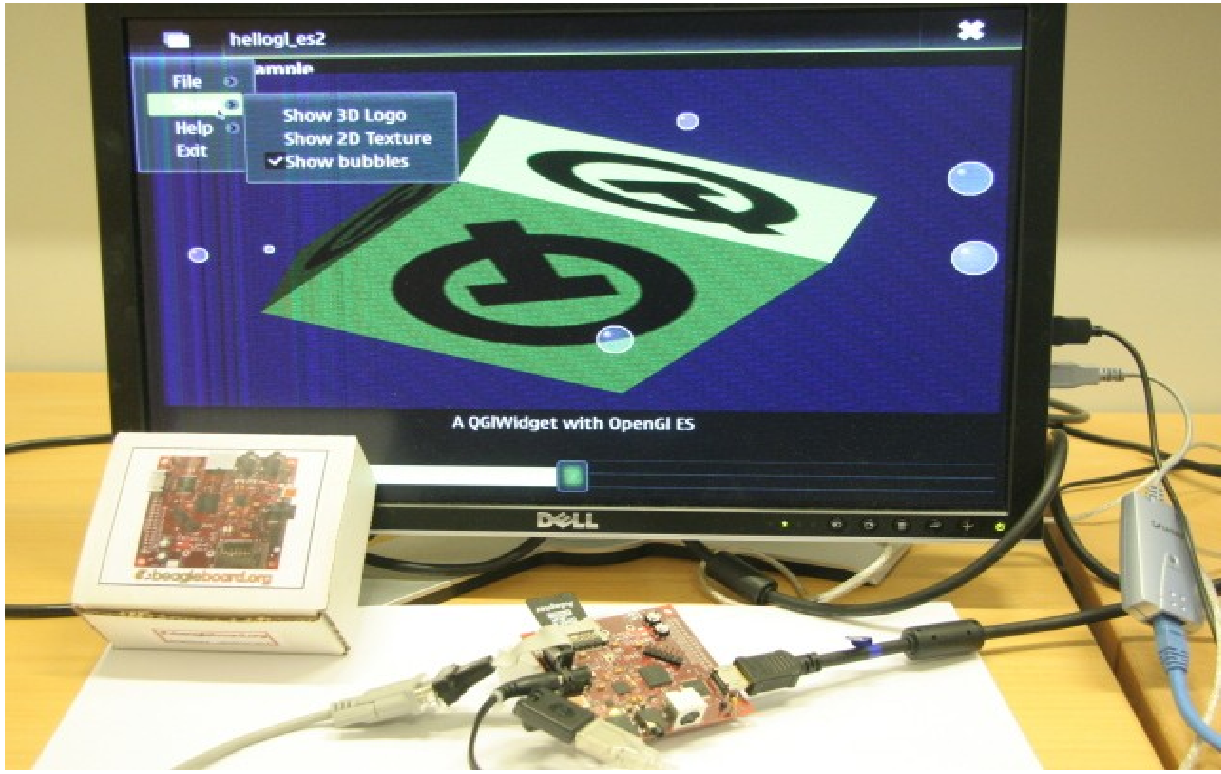
Forum NOKIA

Qt Kinetic

- At the moment, under development
- http://labs.trolltech.com/page/Projects/Graphics/Kinetic
- Some key points
    - Extended animation API
    - EasingCurve,
    - Paralel animations
    - State Machine Framework
- Declarative UI

Qt OpenGL-ES 2.0

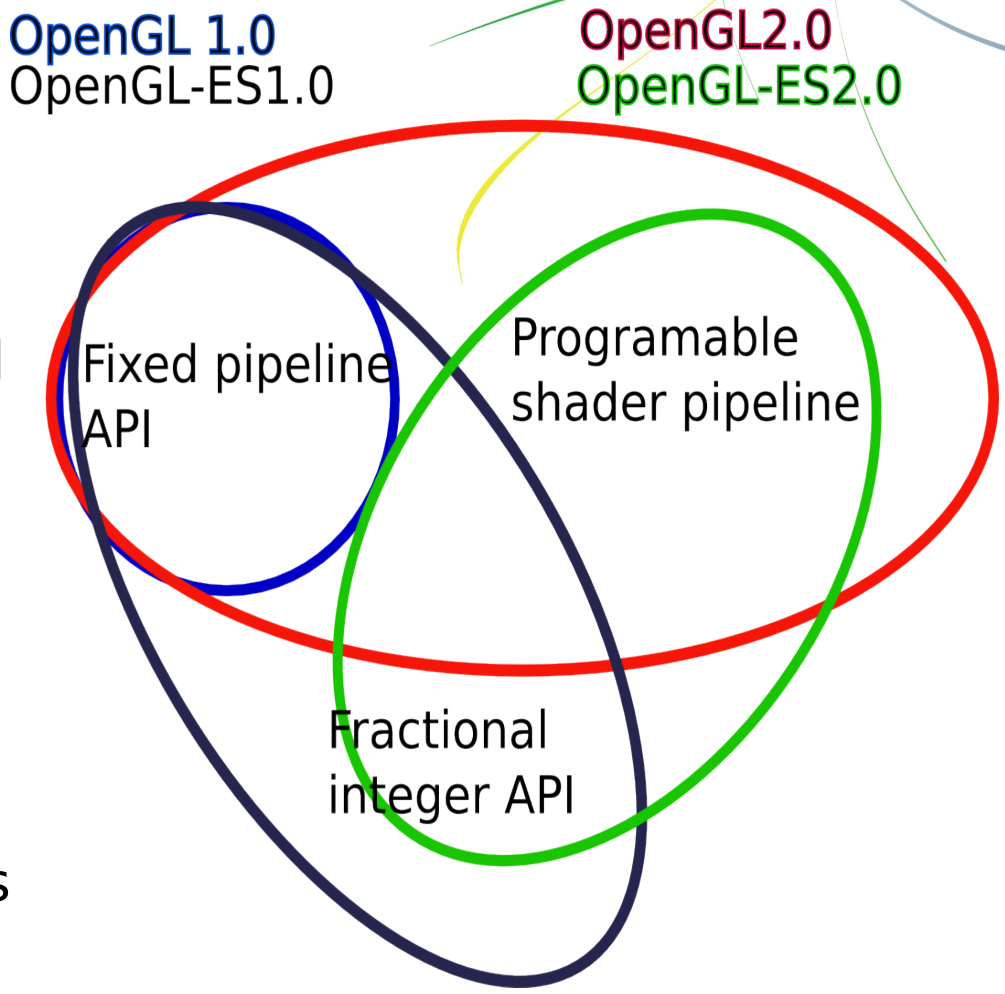
- As new feature, Qt 4.5 has native OpenGL-ES2.0 support
- GL render for QGraphicsView
  - Transparent to applications
  - Speeds up rendering
- QGLWidget
  - Exposes all features of OpenGL to application
- There are several compatibility issues with OpenGL-ES2.0 driver that needed to fix in maemo port



Forum NOKIA

OpenGL versions

- OpenGL 1.0 has fixed shaders and fixed function API to using them
- OpenGL 2.0 adds programmable shaders, but fixed function pipeline API is still there for backward compatibility
- OpenGL 3.1 removes support for old fixed function pipeline, so it is very similar to OpenGL ES 2.0
- OpenGL ES 1.0 is based on OpenGL 1.0 with extra redundant API's removed and fixed point (fractional integer) API added
- OpenGL ES 2.0 is based on OpenGL 2.0, so using programmable shaders is mandatory, since all old fixed function API's have been removed



Forum NOKIA

Porting between OpenGL versions

- OpenGL 1.0 application works with OpenGL 2.0 but not vice versa
- OpenGL 1.0 application is possible to port to OpenGL ES 1.0, but needs work if it is using some of the removed API's
- OpenGL 2.0 application that only uses programmable shaders is possible to port OpenGL ES 2.0, but may still need some work
- Porting OpenGL 1.0 or OpenGL ES 1.0 applications to OpenGL ES 2.0 needs a rewrite to replace fixed function API usage with programmable shaders
- It is possible to emulate OpenGL-ES1.0 with OpenGL2.0 compatible hardware by loading shader code that emulates OpenGL-ES1.0 fixed function pipeline

OpenGL-ES2.0

- Hardware accelerated OpenGL-ES2.0 in Omap3 based devices
- Not supported in Omap2 devices like N800 and N810
- There is OpenGL-ES1.0 driver for Omap2 but due legal reasons, we have not been able to distribute it yet.
- No support by default in scratchbox x86 mode
- Imagination Technologies SDK x86 Linux OpenGL-ES2.0 emulation library can be packet and installed to under scratchbox
- Imagination technologies SDK
    - http://www.imgtec.com/powervr/insider/sdk/KhronosOpenGLES2xSGX.asp
- Books
    - "Mobile 3D graphics with OpenGL-ES and M3G"
    - "OpenGL ES 2.0 programming guide"

OpenGL-ES2.0

- It is important to know that all OpenGL variants are not compatible.
- In Desktop newer OpenGL variants are upward compatible with older variants.
- OpenGL 2 introduced programmable pipeline, programmable shaders but it still had also traditional OpenGL 1.x API.
- OpenGL ES 1.0 was made to be compact, light weight optimized to Mobile devices. Many API's that had duplicated functionality or did not provide optimum performance were removed
- For example direct API's glBegin/glVertex/glEnd were removed and parsing vertex arrays with glVertexAttribPointer is the only method. OpenGL-ES2 went even further, now using the programmable pipeline and then write appropriate shader program implementing wanted function
- Fixed point fractional datatype and single precision float, no double

Forum NOKIA

Programable function pipeline

- In OpenGL-(ES)2.0 shader pipeline is programable
- Vertex transformations, texturing and lightning are programable
- In OpenGL-ES2.0 There is no other choice, you must program them
- Shader language is C-like
- Shader compiler is included in OpenGL-ES2.0 runtime libraries
- Shader source code is included as source format, as example as constant string in application code and they are compiled in runtime.
- Vertex shader computes vertex parameters, as example transformations or per vertex lighting
- Fragment shader computes per pixel value as example texture or per pixel lighting

OpenGL, the old way

- OpenGL-1.0 with glBegin and glEnd
-
```
glColor3f(0.0f, 0.0f, 0.90f); // Blue
glBegin(GL_QUADS);
  glVertex3f(-100.0f, -25.3f, -100.0f);
  glVertex3f(-100.0f, -25.3f, 100.0f);
  glVertex3f(100.0f,  -25.3f, 100.0f);
  glVertex3f(100.0f,  -25.3f, -100.0f);
glEnd();
```

OpenGL-ES1.0

- Only supporting passing vertex as arrays

```
VERTTYPE afVertices[] = {
 f2vt(-0.4f),f2vt(-0.4f),f2vt(0.0f),// Position
 f2vt(1.0f),f2vt(1.0f),f2vt(0.66f),f2vt(1.0f), // Color
 f2vt(+0.4f),f2vt(-0.4f),f2vt(0.0f),
 f2vt(1.0f),f2vt(1.0f),f2vt(0.66f),f2vt(1.0f),
 f2vt(0.0f),f2vt(0.4f),f2vt(0.0f),
 f2vt(1.0f),f2vt(1.0f),f2vt(0.66f),f2vt(1.0f)};
 glBufferData(GL_ARRAY_BUFFER,uiSize,afVertices,GL_STATIC_DRAW);
 glVertexPointer(3, VERTTYPEENUM, sizeof(VERTTYPE) * 7, 0);
 glEnableClientState(GL_COLOR_ARRAY);
 glColorPointer(4,VERTTYPEENUM,sizeof(VERTTYPE) * 7, (GLvoid*)
 (sizeof(VERTTYPE) * 3)/*The color starts after the 3 position values
 (x,y,z)*/);
 glDrawArrays(GL_TRIANGLES, 0, 3);
```

- **Fixed point fractional integers**

```
#define FLOAT2X(f)       ((int) ( (f) * (65536)))
 #define f2vt(f)      FLOAT2X(f)
```

Forum NOKIA

Specfying triengle in OpenGL-ES2

```
int i32Location = glGetUniformLocation(uiProgramObject,
 "myPMVMatrix");
 glUniformMatrix4fv( i32Location, 1, GL_FALSE, pfIdentity);

 GLfloat pfVertices[] =
 {-0.4f,-0.4f,0.0f, +0.4f,-0.4f,0.0f, 0.0f,0.4f,0.0f};

 glBindAttribLocation(uiProgramObject, VERTEX_ARRAY,
 "myVertex");

 glEnableVertexAttribArray(VERTEX_ARRAY);
 glVertexAttribPointer(VERTEX_ARRAY, 3, GL_FLOAT, GL_FALSE,
 0,pfVertices);
 glDrawArrays(GL_TRIANGLES, 0, 3);
```

The simple shader program

- Fragment shader, just return fixed color

```
const char* pszFragShader = "\
void main (void)\
{\
 gl_FragColor = vec4(1.0, 1.0, 0.66 ,1.0);\
}";
```

- Vertex shader, just transform vertex with myPMVMatrix

```
const char* pszVertShader = "\
attribute highp vec4   myVertex;\
uniform mediump mat4   myPMVMatrix;\
void main(void)\
{\
 gl_Position = myPMVMatrix * myVertex;\
}";
```

- Uniform variable remains constant for all vertexes

Forum NOKIA

Compiling the shaders

-
  ```
  m_uiFragShader = glCreateShader(GL_FRAGMENT_SHADER);

  // Load the source code into it
  glShaderSource(m_uiFragShader,1,(const char**)&pszFragShader, NULL);

  // Compile the source code
  glCompileShader(m_uiFragShader);

  // Check if compilation succeeded
  GLint bShaderCompiled;
      glGetShaderiv(m_uiFragShader, GL_COMPILE_STATUS, &bShaderCompiled);
  if (!bShaderCompiled)
  {
  }
  m_uiProgramObject = glCreateProgram();

  // Attach the fragment and vertex shaders to it
      glAttachShader(m_uiProgramObject, m_uiFragShader);
      glAttachShader(m_uiProgramObject, m_uiVertexShader);
  ```

Forum NOKIA

Applying texture

-
  ```
  char* pszFragShader = "\
  uniform sampler2D sampler2d;\
  varying mediump vec2   myTexCoord;\
  void main (void)\
  {\
      gl_FragColor = texture2D(sampler2d,myTexCoord);\
  }";
  char* pszVertShader = "\
  attribute highp vec4   myVertex;\
  attribute mediump vec4 myUV;\
  uniform mediump mat4   myPMVMatrix;\
  varying mediump vec2   myTexCoord;\
  void main(void)\
  {\
    gl_Position = myPMVMatrix * myVertex;\
    myTexCoord = myUV.st;\
  }";
  ```
- **Varying is computed for every vertex**

Forum NOKIA

QGLWidget

- Allows to use full power of OpenGL inside of the Qt
- You need to use system OpenGL API you have available, OpenGL, OpenGL-ES or OpenGL-ES2.0
- Takes care of OpenGL initialization, OpenGL-Context etc
- You must provide
- Shaders and compile them
- Draw actual content with OpenGL
-

Clutter

- OpenGL based animated GUI toolkit
- Very simple, low level, but still easy. Building blocks which are straightforward to use.
- C / Glib based
- API and conventions resemble that of Gtk+
- Bindings for Python, C++, Vala ....
- Mobile optimized as design principle
- Supports both OpenGL for desktop and OpenGL-ES in mobile

Clutter continued

- Clutter is a low level toolkit which does not contain any high level widgets people are used to (e.g. with Gtk+).
- You have got the entity called actor
- There is for example no button, text box, list box etc.
- You can make actors to behave like buttons for example.
- You have to manage the layout manually with coordinates and animations, it is not automated
- Because of the advantage the timelines and fluid animations clutter provides, it is possible to make your UI feel more alive – e.g. if you make a button, you can animate the state change from unpressed to pressed and vice versa rather than just replacing the picture.
-

Forum NOKIA

Clutter Continued

- There is no theming framework in Clutter.
- You can do theming to clutter applications by creating graphics for each different used object size separately. It has some overhead and consumes some space, but it is not that big problem.
- You can also use clutter-cairo to draw textures that are then used as clutter actors. E.g. you can create button graphics with cairo and then make it a texture which is then used to represent a button.
- You can connect signals to actors, no matter if they are moving, rotated, etc. you can press them.
- Clutter is very optimized and efficient with detecting the user input (faster than e.g. Qt GraphicsView at the moment)

Clutter model

- Model: stage (=the main window & canvas) and actor (item you can press, animate etc. and use as building block widgets you write by yourself)
- Actors are objects that behave in stage
  - Move, rotate, scale, change opacity .....
  - Textures
  - Actor is a GObject
- Timelines and events and events control behavior
- Gstreamer media rendering as Actor
- Offers powerful and accurate timelines for animations

Forum NOKIA

Understanding clutter

- So instead of using layout managers to manage your widgets, you place your actors where you want and animate them to move plces you want
- One actor can contain more than textures
- You can stack textures, and each texture can have alpha channel transparency
- There is no button, but you can do one by yourself.
- There is no list, but you can do one by yourself.
- And there is nothing else either but you can do everything by yourself, and quite quickly.
-

Clutter libraries

- Clutter contains multiple related libraries
- Clutter – The basic rendering library
    - Only this library is present in alpha SDK
- Clutter-cairo cairo rendering to clutter texture
- Clutter-gtk
- Clutter-gst gstreameg module
- Clutter-box2d 2d physics engine

Doing button with clutter

- For example: If you have two states on a button, pressed and unpressed state, you can have two textures on top of each other so that the pressed state texture is stacked on top of the button background texture, instead of replacing the button background texture.
  - This way you can create a different style button, it can be for example have highlight when pressed
  - And you can fade the highlight nicely away through alpha channel on button release if you like
  -

Forum NOKIA

Clutter examples continued

- Another example:
  - You can create a toggle button so that you have a toggle button background image and the toggle slider highlight that you stack on top of the background.
  - Then you stack on top of the textures your label.
  - When the user presses the toggle button, you can for example animate the transition from the other side to the another, it can for example slide, fade, etc. instead of just jumping to the another place like would happen on Gtk+.

Forum NOKIA

Clutter is easy continued

- Simplest form of clutter actor
  - Is a texture that is loaded from a file (e.g. a png or jpg)
  - Does not need to be even reactive to user input – e.g. you can do a fake dialog background (which simulates a dialog even if it is not), and you can place e.g. your button actors (that you have to write by yourself, because clutter provides none by default as said before), on top of that.
  - Simplest reactive actor is a texture that behaves like a button, so you get clicks from the actor and your callback function gets executed like you would have on a gtk button a callback function connected to it.
  - And most importantly: CLUTTER IS VERY MUCH FUN TO CODE!

Forum NOKIA