

# OMWeather and Clutter

Vlad Vasilyeu 28.04.2009



# Introduction



## **Vlad Vasilyeu**

- Chief of IT department
- Using Unix system more than 10 years
- Designed and implemented migration the enterprise from Windows to Linux (over 40 servers and 500 stations, 60 program components)
- Designed and developed in C/C++/Perl/Python
- Managed several Web 2.0 projects using LAMP and AJAX technology

## **Maemo**

- Omweather
- Patches for Mplayer, Evince
- Pomni



# OMWeather

- Hildon-home applet for Maemo Platform
- Use GTK and Hildon
- Full name - Other Maemo Weather. Why 'Other' ?
- Main **goals** of the project
  - Weather forecast display on Maemo devices
  - Creation of a convenient program for users
  - Usage of all features of the Maemo platform in the program



# OMWeather



- Main features
  - Current weather and forecast for 10 days
  - Temperature, humidity, pressure, wind, gust and other parameters
  - Possibility to choose various units for parameters: Imperial units, Metric units.
  - Flexible parameters for weather forecast update
  - Use of GPS for automatic selection of weather station
  - And many other configurable options



# OMWeather



- Other features
  - Flexibility of applet's external outlook
    - 5 layouts
    - 6 iconsets
    - 5 variants of icon size
    - Changeable number of displayed icons
    - 5 possible positions of the text around icons
    - Wide range of transparency for applet background
    - Choice of the font size and the font colour

# OMWeather

- Since 2006 year
- Supports all versions of Maemo OS
  - OS2006 (Greagle)
  - OS2007 (Bora) s
  - OS2008 (Chinook, Diablo)
  - OS2009 (Fremantle)





# OMWeather

- Most popular 3-rd party application for Maemo up to date

**Popular**

	Other Maemo Weather (135376 downloads)
	Canola2 Media Player (132281 downloads)
	Maemo Mapper (123449 downloads)
	MPlayer (113125 downloads)
	OpenSSH client (75130 downloads)

Nokia Corporation | [Terms of Use](#) | [Privacy Policy](#)

sponsored by **NOKIA**



# OMWeather

- International Team

- Belarus 
- Russia 
- Finland 
- Canada 
- Germany 
- Mexico 
- France 





# OMWeather

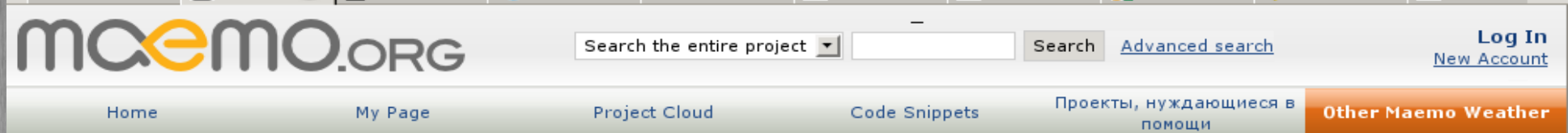
- Classical team for opensource project
  - Maintainer
  - Developers
  - Art Designer
  - Translators
- Powerusers are our beta-testers
- Communication between team members with e-mail, IM and VoIP



# OMWeather

- Site of the project

<http://garage.maemo.org/omweather>



- We use
  - SVN
  - Various trackers
  - Forums
  - “File” section
  - Mail List





# OMWeather

- For enduser feedback we use email, forums in project's Garage and popular community site <http://www.internettalk.com/forums/>



SMOKE

# OMWeather

- In January 2009 OMWeather had been chosen by Maemo Team as Fremantle Star project



- Thank you very much



# OMWeather

- Now we are developing a new version - 0.22
  - Adapting for Fremantle
  - Adding new sources of weather forecast
  - Adding hour weather forecast for today
  - Adding new mode – Application
  - Adding Clutter animation



# Clutter

- Clutter is an open source software library for creating fast, visually rich and animated graphical user interfaces.

<http://www.clutter-project.org/>



# Clutter

- Important features of Clutter for Omweather
  - Scene-graph of layered 2D interface elements manipulated in 3D space via position, grouping, transparency, scaling, clipping and rotation.
  - Frame based animation engine providing path interpolation, transitions and other custom effects via Behaviours and Timelines.
  - Scriptable JSON based layout and animation file support.

# Clutter

- Clutter model
  - Stage and Actors (Items you can animate and press)
  - Actors are objects that can
    - Scale, Move, Rotate,.....
  - Behaviours
  - Timelines



# Clutter

- Clutter Actors
  - ClutterRectangle
  - ClutterTexture
  - ClutterLabel
  - ClutterEntry
- ClutterGroup — Actor class containing multiple children. Actors.

# Clutter

- Behaviours
  - ClutterBehaviourB spline
  - ClutterBehaviourDepth
  - ClutterBehaviourEllipse
  - ClutterBehaviourOpacity
  - ClutterBehaviourPath
  - ClutterBehaviourRotate
  - ClutterBehaviourScale



# Clutter

- Timeline
  - Three main parameters:
    - n\_frames - the number of frames
    - fps - the number of frames per second
    - Loop - whether the timeline should automatically rewind and restart.

# Clutter - Simple Example

```
#include <clutter/clutter.h>
void
on_new_frame (ClutterTimeline *timeline,
              gint             frame_num,
              gpointer         data){
    ClutterActor *actor = CLUTTER_ACTOR(data);
    clutter_actor_set_rotation (actor, CLUTTER_Z_AXIS,
                                (gdouble) frame_num,
                                clutter_actor_get_width (actor) / 2,
                                clutter_actor_get_height (actor) / 2,
                                0);
}

int
main (int argc, char *argv[])
{
    ClutterTimeline *timeline;
    ClutterActor *stage, *actor;
    GdkPixbuf *pixbuf;

    clutter_init (&argc, &argv);
    stage = clutter_stage_get_default ();
```



# Clutter - Simple Example

```
error = NULL;
pixbuf = gdk_pixbuf_new_from_file ("./redhand.png", &error);
actor = gtk_clutter_texture_new_from_pixbuf(pixbuf);

clutter_container_add_actor (CLUTTER_CONTAINER (stage), actor);

clutter_actor_set_position (actor, 100, 100);

timeline = clutter_timeline_new_for (360, 60); /* a degree per frame */
clutter_timeline_set_loop (timeline, TRUE);

g_signal_connect (timeline, "new-frame", G_CALLBACK (on_new_frame),
actor);

clutter_actor_show_all (stage);

clutter_timeline_start (timeline);

clutter_main();

return 0;
}
```

# Clutter

- JSON
  - Object of ClutterScript
  - The UI definition format is JSON, the JavaScript Object Notation as described by RFC 4627
  - ClutterScript can load a JSON data stream, parse it and build all the objects defined into it.



# Clutter

- JSON example

```
{
  "id"    : "red-button",
  "type"  : "ClutterRectangle",
  "width" : 100,
  "height": 100,
  "color" : "#ff0000ff"
  "behaviours" : [ "rotate-behaviour" ]
}
```

## In source code:

```
ClutterActor *red_button;
```

```
red_button = CLUTTER_ACTOR
(clutter_script_get_object (script, "red-button"));
```

```
{
  "id"      : "rotate-behaviour",
  "type"    : "ClutterBehaviourRotate",
  "angle-start" : 0.0,
  "angle-end" : 360.0,
  "axis"    : "z-axis",
  "alpha"   : {
    "timeline" : { "duration" : 4000, "fps" : 60, "loop" : true },
    "function" : "sine"
  }
}
```

# OMWeather & Clutter

- We need rich animation
- We will make animation for icon of weather forecast





# OMWeather & Clutter

- We have 6 iconsets



- More than 240 icons
- Source code for each icon?
- No! We will use JSON-scripts

# OMWeather & Clutter

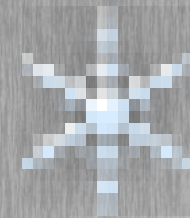
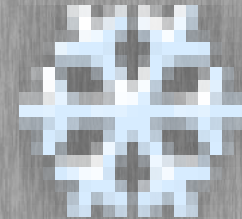
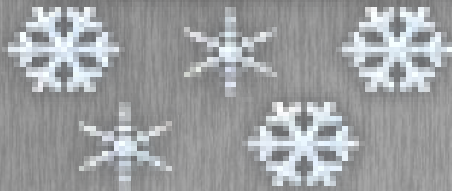
- We have a static icon





# OMWeather & Clutter

- Let's divide this icon into elements



# OMWeather & Clutter

- \* Let's make description in JSON script for a cloud

```
{
  "id" : "gray_cloud",
  "type" : "ClutterTexture",
  "filename" :
"gray_cloud.png",
  "visible" : true,
  "x": 10,
  "y": 10,
  "behaviours" : [
    "fade-behaviour"
  ]
},
```



# OMWeather & Clutter

Let's make description in JSON script for snowflake2 and snowflake3

```
{
  "id" : "snowflake2",
  "type" : "ClutterTexture",
  "filename" : "snowflake.png",
  "visible" : true,
  "behaviours" : [
    "snow2-behaviour",
    "rotate-behaviour",
  ]
},
```

```
{
  "id" : "snowflake3",
  "type" : "ClutterTexture",
  "filename" : "snowflake.png",
  "visible" : true,
  "behaviours" : [
    "snow3-behaviour",
    "rotate-behaviour",
  ]
},
```

# OMWeather & Clutter

Let's make the description in JSON script for snowflakelight1 and snowflakelight2

```
{
  "id" : "snowflakelight1",
  "type" : "ClutterTexture",
  "filename" : "snowflake_light.png",
  "visible" : true,
  "behaviours" : [
    "snowflakelight1-behaviour",
    "rotate2-behaviour"
  ]
},
```

```
{
  "id" : "snowflakelight2",
  "type" : "ClutterTexture",
  "filename" : "snowflake_light.png",
  "visible" : true,
  "behaviours" : [
    "snowflakeligh2-behaviour",
    "rotate2-behaviour"
  ]
},
```



# OMWeather & Clutter

Let's make behaviours for a snowflake

```
{
  "id"      : "snow1-behaviour",
  "type"    : "ClutterBehaviourPath",
  "knots"   : [[25,30], [30,80]]
  "alpha"   : {
    "timeline" : "main-timeline",
    "function" : "ramp-inc"
  }
},
```

```
{
  "id"      : "rotate2-
behaviour",
  "type"    :
"ClutterBehaviourRotate",
  "angle-start" : 0.0,
  "angle-end"   : 360.0,
  "axis"       : "z-axis",
  "alpha"     : {
    "timeline" : "main-timeline",
    "function" : "ramp-inc"
  }
},
```

# OMWeather & Clutter

- Union all actors to ClutterGroup

```
{
  "id" : "icon_name_16",
  "type" : "ClutterGroup",
  "x" : 0,
  "y" : 0,
  "width" : 128,
  "height" : 128,
  "children" : [
    .....Actor1.....
    .....
    .....Actor2.....
  ]
}
```



# OMWeather & Clutter

- Let's show how it works

Video on Youtube

<http://www.youtube.com/watch?v=KKdpBKUXdCQ>

# OMWeather & Clutter

Thank you for your time