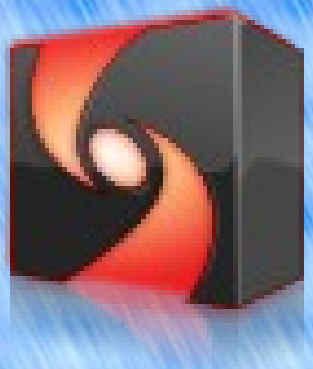


Разработка модулей расширения для аппаратно-ускоренных графических оболочек на базе Compiz

Основные отличия разработки плагинов compiz 0.8.x и compiz++ 0.9.x

Дёмин Владимир
E-mail: spas.work@gmail.com

Для разработки

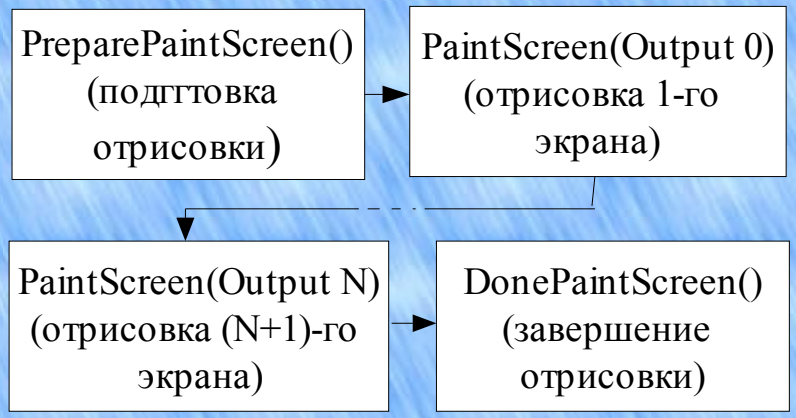
- Композитный менеджер окон Compiz
- Библиотеки OpenGL
- Фреймбуффер

Модуль

Функциональные части:

- Код инициализации
- Обработчик событий
- Сервисные функции

Схема отрисовки:



Результаты анализа средств разработки

Что было до?

Версия 0.8.x

- make файл для сборки проекта;
- xml-файл для описания графического интерфейса управления плагином;
- plugin.info файл содержащий список зависимостей, необходимых для работы модуля;
- файлы исходного кода плагинов на Си.

Версия 0.8.x

- код плагинов слабо документирован
- множество вставок объектно-ориентированного подхода, которые затрудняют понимание кода;
- отсутствие документации API и мануалов по созданию плагинов.

Что будет?

Версия 0.9.x

- Новый интерфейс для создания плагинов, несовместимый с разработанными для Compiz плагинами, но отличающийся повышенной производительностью;
- Разделение композитного (XComposite) и OpenGL уровней (реализованы в виде раздельных плагинов), что позволяет использовать Compiz++ как обычный оконный менеджер, в случае когда использовать композитный режим невозможно;

Версия 0.9.x

- Переработка системы обработки текстур, позволяющих создавать мозаичные текстуры, в которых возможно интегрировать несколько текстур на один pixmap;
- Миграция на систему сборки CMake.

Си реализация

```
#define GET_MWIN_DISPLAY(d) \
    ((mWinDisplay *) (d)->base.privates[displayPrivateIndex].ptr)
#define MWIN_DISPLAY(d) \
    mWinDisplay *sd = GET_MWIN_DISPLAY (d)
#define GET_MWIN_SCREEN(s, sd) \
    ((mWinScreen *) (s)->base.privates[(sd)->screenPrivateIndex].ptr)
#define MWIN_SCREEN(s) \
    mWinScreen *ss = GET_MWIN_SCREEN (s, GET_MWIN_DISPLAY (s-
>display))
#define GET_MWIN_WINDOW(w, ss) \
    ((mWinWindow *) (w)->base.privates[(ss)->windowPrivateIndex].ptr)
#define MWIN_WINDOW(w) \
    mWinWindow *sw = GET_MWIN_WINDOW  (w,                \
                GET_MWIN_SCREEN  (w->screen,         \
                GET_MWIN_DISPLAY (w->screen->display)))
```

C++ реализация

```
static Bool
pluginActionCallback
(CompDisplay     *d,
CompAction      *action,
CompActionState state,
CompOption      *option,
int             nOption)
{}

CompWindow *w = findWindowAtDisplay (d, getIntOptionNamed (option, nOption,
"window", -1));
CompScreen *s = findScreenAtDisplay (s, getIntOptionNamed (option, nOption,
"screen", -1));
```

Си реализация

Функции обработки событий:

```
static void
tdPreparePaintScreen (CompScreen *s,
                      int        msSinceLastPaint)
{
    CompWindow *w;
    Bool       active;

    TD_SCREEN (s);
    CUBE_SCREEN (s);
      .....
      .....
      .....
    UNWRAP (tds, s, preparePaintScreen);
    (*s->preparePaintScreen) (s, msSinceLastPaint);
    WRAP (tds, s, preparePaintScreen, tdPreparePaintScreen);

    cs->paintAllViewports |= tds->active;
}
```

C++ реализация

Функции обработки событий:

```
void
TdScreen::preparePaint (int msSinceLastPaint)
{
    bool       active;

    CUBE_SCREEN (screen);
.....
.....
.....

    cScreen->preparePaint (msSinceLastPaint);

    setFunctions (mActive);
    cs->paintAllViewports (mActive);
}
```

Различия в коде

Compiz

```
static void
tdDonePainScreen (CompizScreen *s)
{
	TD_SCREEN(s);
	if (tds->active && tds->damage)
	{
		tds->damage = FALSE;
		damageScreen(s);
	}
	UNWRAP(tds, s, donePainScreen);
	(*s->donePaintScreen,
		TdDonePaintScreen);
}
```

Compiz++

```
void
TdScreen::donePaint ()
{
	if (mActive && mDamage)
	{
		mdamage = false;
		cScreen->damageScreen;
	}
		cScreen->donePaint();
}
```

Какие проблемы ожидают?

Недостатки

- необходимость портирования всего кода на другой язык программирования, а значит появления новых ошибок, потеря большого количества времени разработки;
- неизвестно, насколько может упасть производительность работы системы в целом;
- отток чистых Си программистов.

Спасибо за внимание!

Дёмин Владимир
E-mail: spas.work@gmail.com