

Наконец, помимо группового переключения окон, актуальной задачей представляется также их групповой запуск (что позволит перекрыть функционал ярлыков запуска программ в док-панели). Панель мини-окон может содержать для этого специальный доклет – псевдогруппу с пиктограммой, отвечающей за историю работы оконного интерфейса. Выбор ее в списке мини-окон приводит к затемнению экрана и отображению на нем групп окон, сформировавшихся пользователем ранее (рис. 4).

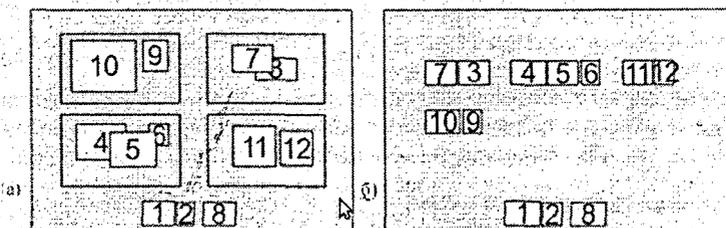


Рисунок 7 – История группированных окон

При незначительном числе групп можно отображать уменьшенные изображения виртуальных рабочих столов, как это делает пейджер (рис.7-а), а в случае значительного количества, когда экран не позволяет отображать рабочие столы с удовлетворительным коэффициентом масштабирования, – в виде сгруппированных миниатюр (рис. 7-б).

Выбор одной из групп на экране приводит к запуску входящих в нее приложений и размещению их окон в запомненных позициях экрана.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Раскин, Дж. Интерфейс: новые направления в проектировании компьютерных систем. – СПб.: Символ-Плюс, 2003. – 272 стр.
2. Гульятёв, А.К. Проектирование и дизайн пользовательского интерфейса. – А.К. Гульятёв, В.А. Машин / СПб.: "КОРОНА принт", 2000. – 352 стр.
3. Дёмин, В.В. Реализация модели динамических мини-окон в аппаратно-ускоренном графическом интерфейсе // Современные проблемы математики и вычислительной техники: материалы VI Республиканской научной конференции молодых ученых и студентов, Брест, 26-28 ноября 2009 г. – Брест: БрГТУ, 2009. – Ч. II. – С. 21-24
4. Костюк, Д.А. Модель мини-окон с динамическим отображением в аппаратно-ускоренном графическом интерфейсе / Д.А. Костюк, В.В. Дёмин // Вестник БрГТУ. – 2009, №5 (59): Физика, математика, информатика. – С. 71-74.

УДК 004.514.62

Дёмин В.В

Научный руководитель: к.т.н, доцент Костюк Д.А.

РЕАЛИЗАЦИЯ МОДУЛЯ РАСШИРЕНИЯ ДЛЯ МИНИАТЮРИЗАЦИИ ОКОН ПРИЛОЖЕНИЙ В ОКОННОМ МЕНЕДЖЕРЕ COMPIZ

Введение

Задача манипулирования окнами в ситуации, когда их совокупная площадь существенно превышает разрешение устройства вывода, возникла почти одновременно с графическим интерфейсом пользователя. Один из ранних способов облегчить навигацию в таких условиях получил название иконификации – окно «сворачивалось» в пиктограмму

соответствующего приложения, снабженную поясняющей подписью. Впоследствии концепция получила альтернативное развитие. С ростом вычислительных возможностей процессоров и разрешающей способности экранов появились графические оболочки, отображавшие вместо пиктограммы скрытого окна его уменьшенное изображение. Хотя такой подход сопряжен с рядом технических трудностей, возможность видеть одновременно, хотя бы с уменьшенной детализацией, изображение всего рабочего пространства дает пользователю ощутимые преимущества [1, 2]. Особенно это актуально в среде динамически изменяющихся объектов, к которым относятся все многозадачные графические среды современных операционных систем.

Нами был разработан реализующий этот подход модуль расширения оконного менеджера Comriz [3], являющегося на сегодняшний день наиболее популярным компонентом аппаратно-ускоренных графических оболочек ОС GNU/Linux.

Специфика аппаратно-ускоренного манипулирования окнами

Все современные аппаратно-ускоренные менеджеры окон (включая Comriz) используют возможности библиотеки OpenGL для передачи вычислительной нагрузки графическому акселератору. Одно из свойств OpenGL – объекты фреймбуфера – дает оконному менеджеру эффективный доступ к окнам неактивных приложений. Для приложения объекты фреймбуфера выглядят как обычные окна, а для оконного менеджера – как текстуры, которыми можно управлять с помощью обычных команд отрисовки мультитекстур (рис. 1).

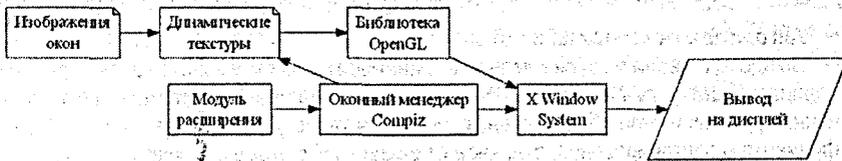


Рисунок 1 – Схема взаимодействия разработанного модуля с графической оболочкой и средствами поддержки аппаратно-ускоренной графики

Разработанный модуль расширения Comriz отрисовывает миниатюры окон на основе этой технологии [4].

Модуль состоит из трех функциональных частей: кода инициализации, обработчиков событий и сервисных функций. Код инициализации вызывается при загрузке модуля в память. При этом системе передаются указатели на конструкторы и деструкторы, вызываемые при создании и удалении контекста дисплея, каждого из соответствующих дисплею виртуальных экранов, и каждого из соответствующих виртуальному экрану окон. Доступ к контексту дисплея позволяет перехватывать события, экран дает возможность вклиниться в цепочку процедур обновления изображения, включающую подготовительную стадию, собственно отрисовку экрана (или экранов, если их несколько) и завершение отрисовки (рис. 2).



Рисунок 2 – Стадии отрисовки виртуальных экранов

Подготовительная стадия используется для выполнения различных подготовительных расчетов, например, для вычисления новых координат. В частности, в разработан-

ном модуле на этой стадии при ее первом прохождении выполняется загрузка в объекты текстур в памяти изображений, для их последующего вывода в качестве подложки миниатюризированных окон и формирования, таким образом, панели, на которой размещены миниатюры (рис. 3).

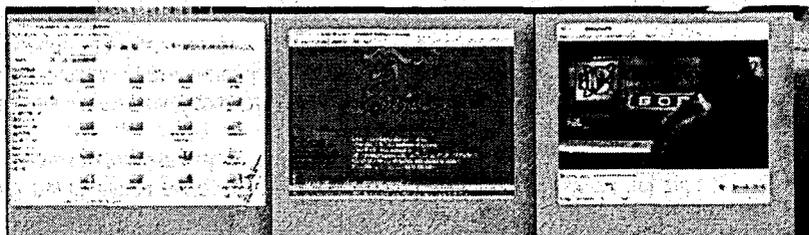


Рисунок 3 – Внешний вид реализованной панели с мини-окнами

Любой метод отрисовки в первую очередь должен вызывать базовую реализацию, обновляющую изображение экрана, а затем выполняет с полученным изображением любые действия, доступные средствами OpenGL; при этом он имеет доступ к устройству вывода, физическим координатам и всем открытым окнам, для каждого из которых существует соответствующая текстура. Стадия завершения отрисовки при необходимости используется для принудительной генерации события перерисовки экрана, благодаря чему выполняется отображение анимационных эффектов в реальном масштабе времени.

Алгоритм миниатюризации окон

В ходе активации модуля командой миниатюризации окна запоминаются прежние координаты окна, устанавливается новое значение координат в соответствии с предназначенной для мини-окон областью внизу экрана, а также устанавливается значение коэффициента масштабирования, отличное от единицы. По этому коэффициенту код модуля, обрабатывающий события отрисовки, отличает минимизированные окна от стандартных.

При инициализации экрана модуль закрепляет собственные обработчики за необходимыми событиями. Задействованы следующие основные обработчики событий:

PreparePaintScreen, вызывается непосредственно перед выполнением PaintWindow и производит необходимые подготовительные расчеты;

- PaintWindow, срабатывающий каждый раз при обновлении содержимого окна, проверяющий масштаб окна и, если оно должно быть миниатюризировано, вносящий изменения в его матрицу трансформации;
- DamageWindowRect, определяющий конкретный фрагмент экрана как требующий перерисовки (таким образом достигается дополнительное снижение аппаратных затрат по сравнению с полным обновлением экрана).

Также служебная функция, встроенная в цикл обработки событий, отслеживает щелчок мышью по мини-окну для возвращения его к единичному масштабу и прежним координатам на экране.

Изменение масштаба окна затрагивает только его изображение. Очевидно, что для самого окна при этом не происходит никаких изменений в размерах, иначе была бы нарушена обратная совместимость с существующими приложениями. Однако аналогичная ситуация с размерами окна сохраняется и для X-сервера. Поэтому с точки зрения системы все элементы управления окна сохраняют свои прежние координаты, и для взаи-

действия с этими элементами необходимо помещать указатель мыши в ту область экрана, где они находились бы, сохраняя окно единичный масштаб. Что еще более нежелательно, манипуляции мышью за пределами мини-окна должны приводить к активации его различных элементов.

Способ избежать рассогласования в координатах, реализованный в разработке, предполагает перехват событий мини-окна, связанных с указателем мыши. Это не позволяет взаимодействовать с элементами окна, но такая возможность для мини-окон и не требуется. События мыши, приходящиеся на область масштабированного изображения, обрабатываются разработанным модулем расширения Compriz (для подсветки мини-окна при движении указателя или перевода в немасштабированное состояние по щелчку), а остальные – блокируются. Данный механизм поддерживается непосредственно через расширение XShare, изначально предназначенное для приложений, использующих канал прозрачности для отображения окон непрямоугольной формы (события мыши, попадающие на внешние прозрачные участки окна, блокируются). Механизм известен как формирование входных сигналов и задействован в Compriz при реализации ряда стандартных модулей.

При этом для мини-окон остаются по-прежнему доступными события клавиатуры и смена фокуса стандартными средствами переключения окон, что позволяет в ряде случаев взаимодействовать с мини-окном без его возвращения к стандартному масштабу.

Представление структур данных

Для хранения и доступа к информации о рабочих столах и окнах, расположенных на нем, используется ряд специализированных структур. Дисплею системы X Window соответствует структура mWinDisplay, хранящая в себе указатель на функцию – обработчик событий, а также массив индексов, адресующих имеющиеся экраны (рис. 4).

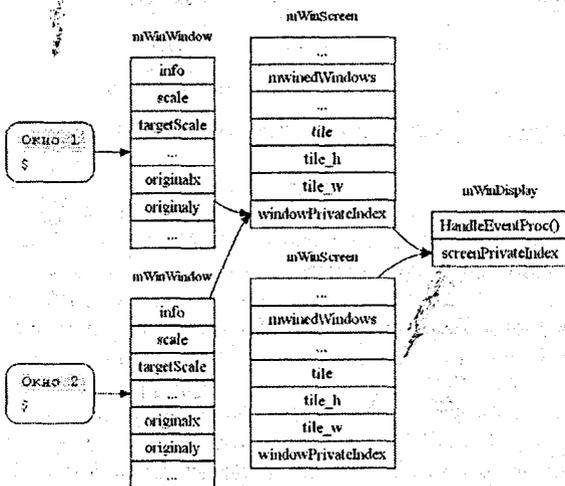


Рисунок 4 – Схема закрепления окон за каждым из рабочих столов

Каждому экрану соответствует структура mWinScreen. В данной структуре в числе другой служебной информации хранятся сведения о текстуре, отображающей подложку

панели мини-окон, а также индексный массив, адресующий каждое из имеющихся окон. Представлением окна для разработанного программного модуля является структура `mWinWindow`. Она содержит в себе исходные координаты окна и коэффициент масштабирования, отличный от единицы для мини-окон (а также промежуточный масштабный коэффициент для анимированного отображения процесса миниатюризации). Структура `mWinWindow` связана со стандартным типом, представляющим окно в `Compoz`, и может быть легко для него получена.

Панель мини-окон представлена в виде динамического односвязного списка, состоящего из структур `mWinedWindowInfo` (рис. 4). Элемент этого типа создается при миниатюризации окна и добавляется в конец списка. Он хранит координаты подложки мини-окна и ссылку на объект окна `Compoz`. Соответственно, каждый объект `mWinWindow` содержит в себе ссылку `info`, адресующую соответствующий ему элемент списка и, таким образом, позицию мини-окна в панели (см. окна 1-3 на рис. 4). Для неминиатюризованных окон данное поле является нулевым (см. окно 4).

После нажатия комбинации клавиш, активирующей миниатюризацию, размеры окна изменяются и измененная информация заносится в структуру `mWinedWindowInfo`, а начальные положение окна, его размеры, масштаб относительно миниатюры – в структуру `mWinWindow`. Эти данные далее будут использованы для восстановления окна из миниатюризованного состояния.

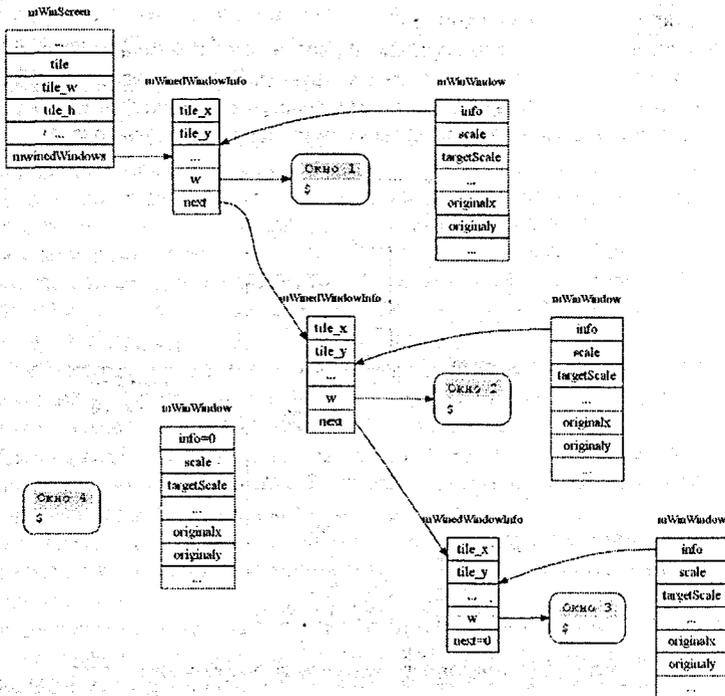


Рисунок 5 – Схема связи структур модуля

Как можно заметить на рис. 5, для отрисовки окна достаточно передать ссылку на структуру `mWinScreen`, и далее по ней можно найти все необходимые данные. Благодаря этому становится возможным независимое хранение окон на рабочих столах. Приведенная на рис. 1 схема связи программных частей модуля позволяет использовать мультиэкранный режим работы.

Несмотря на большое количество вычислений, постоянно производимых модулем при активном использовании панели миниатюр, это не нагружает центральный процессор, т.к. все функции аппаратно ускорены через библиотеку `OpenGL`. Для того чтобы еще понизить загрузку системы, при модификации панели мини-окон (например, разминиатюризации одного из окон) выполняется не полная, а частичная перерисовка экрана, соответствующая координатам изменившейся области.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Раскин Дж. Интерфейс: новые направления в проектировании компьютерных систем. – СПб.: Символ-Плюс, 2003. – 272 с.
2. Борушко, И.Н. Применение модели периферического зрения в графическом интерфейсе пользователя. Современные информационные компьютерные технологии: сб. науч. ст. / И.Н. Борушко, Е.В. Гоманова, Д.А. Костюк. – Гродно: ГрГУ, 2006. – С. 22-27.
3. Дёмин, В.В. Реализация модели динамических мини-окон в аппаратно-ускоренном графическом интерфейсе // Современные проблемы математики и вычислительной техники: материалы VI Республиканской научной конференции молодых ученых и студентов, Брест, 26-28 ноября 2009 г. – Брест: БрГТУ, 2009. – Ч. II. – С. 21-24.
4. Костюк, Д.А. Модель мини-окон с динамическим отображением в аппаратно-ускоренном графическом интерфейсе / Д.А. Костюк, В.В. Дёмин // Вестник БрГТУ. – 2009, №5 (59): Физика, математика, информатика. – С. 71-74.

УДК 004.514.62

Дёмин В.В

Научный руководитель: доцент Дунец А.П.

РЕАЛИЗАЦИЯ МОДЕЛИ РОБОТА ИГРОКА ДЛЯ ROBOSUP SOCCER SIMULATION SERVER

Введение

RoboSup – международные соревнования среди роботов, основанные в 1993 г. Официальная цель проекта – к середине XXI-го века команда полностью автономных человекоподобных роботов-футболистов должна выиграть футбольный матч, соблюдая правила FIFA, у победителя Чемпионата мира.

Simulation league – это часть проекта Robosup, которая направлена на моделирование игры в простейшем 2D-варианте, без относительно реальных реализаций роботов-футболистов.

В лиге симуляции принимают участие две команды с одиннадцатью автономными программами (которые называют агенты), играющие в футбол на двухмерном виртуальном стадионе, который предоставляется центральным сервером под названием SoccerServer. На сервере содержится вся необходимая информация об игре, такая как позиции игроков и мяча, физика, направления игроков и другие параметры и свойства виртуального мира. Игра основана на сообщениях между сервером и агентами. С одной стороны, каждый игрок получает данные с его виртуальных сенсоров (визуальные, аку-