

## СПЕЦИФИКАЦИЯ БАЗОВОЙ ГРАФИЧЕСКОЙ СИСТЕМЫ ДЛЯ МНОГООКОННОЙ ДИНАМИЧЕСКОЙ ГРАФИКИ

Эффективное программирование задач машинной графики требует наличия достаточно мощного языка. Некоторые возможности, имеющиеся в языке Модуля-2, делают его в высокой степени пригодным для создания графического программного обеспечения. К таким возможностям можно отнести: экспорт/импорт типов и констант, открытые массивы, параллельные процессы и машинно-зависимые возможности.

В работе представлен проект системы многооконной динамической графики, ориентированной на реализацию на мощных персональных ЭВМ с растровым графическим дисплеем.

### ПРЕДПОСЫЛКИ

Система должна быть многооконной, так как она будет использоваться для обеспечения графического взаимодействия оператора с целым рядом параллельно идущих процессов. При этом каждому процессу выделяется свое "окно" на экране, а оператор, делая "активным" то или иное окно процесса, может наблюдать за его ходом или вступить с ним в диалог. Окна могут как частично, так и полностью перекрываться на экране. Процессы, окна которых в текущий момент не активны, продолжают идти.

Поскольку система должна непрерывно отражать ход процессов, она должна быть динамической, то есть в ней должна иметься возможность выборочного изменения картинки без полной ее перерисовки.

### АРХИТЕКТУРА СИСТЕМЫ

Простой метод получения динамического изображения на растровом дисплее состоит в следующем. При выводе пикселя на экран будем производить сложение по модулю 2 его нового и старого цветов. Тогда для стирания любой подкартинки достаточно нарисовать ее повторно.

Этот метод динамизации, а также потребность куда-то записывать изображения, генерируемые процессами, окна которых в текущий момент не активны, приводит к необходимости иметь "буфер изображений". Этот буфер должен быть частью базовой графической системы.

Общая архитектура многооконной динамической графической системы представлена на рис. 1.



Рис. 1

Базовая графическая система содержит процедуры приблизительно того же уровня, что и система GKS [1]. Машинно-зависимый графический модуль ориентирован на работу с конкретным экраном и является, по существу, графическим драйвером, который, однако, расширен процедурами диалогового управления окнами [2]. Геометрический модуль содержит процедуры чисто геометрического характера, например отсечения и т. д.

Модули деловой и научной графики представляют собой проблемно-ориентированные графические пакеты того же уровня функциональности, что, например, система ГРАФОР для БЭСМ-6 [3].

## БАЗОВАЯ ГРАФИЧЕСКАЯ СИСТЕМА

В проекте базовой графической системы использованы терминология и некоторые идеи из системы GKS. Имеются три уровня структуризации графических данных — верхний (рабочие станции), средний (сегменты) и нижний (графические примитивы). Здесь, однако, рабочие станции мы будем называть картинками. Предполагается, что каждая картинка рисуется в своем окне.

Приняты и более существенные отклонения от идеологии GKS. Одновременная работа со многими картинками не допускается. Сегменты могут быть вложенными (допустима иерархия сегментов) и даже составлять более сложную структуру.

Текст модуля определенной базовой графической системы приведен на рис. 2. В этом тексте опущены процедуры установки систем координат и атрибутов.

```
DEFINITION MODULE MultiPic; (* А.Яковлев, ВЦ АН СССР, 1986 *)
```

```
TYPE Picture = CARDINAL;
```

```
PROCEDURE CreatePic (VAR p : Picture);
```

```
(* создание в буфере новой картинки и открытие ее *)
```

```
PROCEDURE OpenPic (p : Picture);
```

```
(* открытие существующей картинки p для модификации *)
```

```
PROCEDURE ClosePic;
```

```
(* закрытие текущей картинки *)
```

```
PROCEDURE ErasePic (VAR p : Picture);
```

```
(* удаление картинки p из буфера *)
```

```
PROCEDURE DrawPic (p : Picture);
```

```
(* выталкивание модификаций картинки p на экран *)
```

```
PROCEDURE Polyline (VAR x, y : ARRAY OF REAL; n : CARDINAL);
```

```
(* ломаная с n узлами *)
```

```
TYPE
```

```
Segment = CARDINAL;
```

```
Transform = ARRAY [1..2], [1..3] OF REAL;
```

```
PROCEDURE CreateSeg (VAR s : Segment; t : Transform);
```

```
(* создание нового сегмента *)
```

```
PROCEDURE CloseSeg;
```

```
(* закрытие текущего сегмента *)
```

```
PROCEDURE DeleteSeg (VAR s : Segment);
```

```
(* удаление сегмента s *)
```

```
PROCEDURE LinkSeg (father, son : Segment; t : Transform);
```

```
(* связывание сегментов father и son по преобразованию t *)
```

```
PROCEDURE UnlinkSeg (father, son : Segment);
```

```
(* обрыв связи между сегментами father и son *)
```

```
END MultiPic.
```

Картинки могут создаваться, открываться, модифицироваться, закрываться и удаляться. Все эти процедуры могут вызываться из прикладных модулей и оказывать влияние лишь на буфер, но не на экран. Выталкивание накопленных изменений на экран делает специальная процедура рисования, которая, вообще говоря, должна вызываться асинхронно только из машинно-зависимого графического модуля.

#### ГРАФИЧЕСКИЕ ПРИМИТИВЫ

В настоящее время определен единственный графический примитив – полилиния.

Сегменты могут создаваться, заполняться примитивами, закрываться и удаляться. Эти функции соответствуют имеющимся в GKS. Здесь, однако, в отличие от GKS, при создании сегмента может быть уже открыт сегмент: таким образом легко определяется дерево сегментов. Кроме этого, могут устанавливаться и рваться межсегментные связи, что обеспечивает возможность построения произвольного бесконтурного орграфа сегментов с источником.

Модификация содержимого сегментов, как и в системе GKS, не допускается.

#### ЛИТЕРАТУРА

1. Enderle G., Kansy K., Pfaff G. Computer Graphics Programming. GKS – the Graphics Standard. – Springer-Verlag, 1984.
  2. Herman I. Managing Multiple Context-Frames through GKS. – Computer Graphics Forum, 1984, v. 3, p. 79–82.
  3. Баяковский Ю. М., Галактионов В. А., Михайлова Т. Н. ГРАФОР. Графическое расширение Фортрана. – М.: Наука, 1985.
-