

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Кузбасский государственный технический университет  
имени Т. Ф. Горбачева»

Кафедра информационных и автоматизированных  
производственных систем

Составители  
В. В. Зиновьев  
И. С. Кузнецов  
А. Н. Стародубов

## **СОЗДАНИЕ АНИМАЦИОННОЙ МОДЕЛИ ИНФОРМАЦИОННОЙ СИСТЕМЫ В PROOF ANIMATION**

Методические указания к практической работе  
по дисциплине «Математическое моделирование»

для обучающихся специальности СПО  
09.02.07 Информационные системы и программирование

Рекомендовано цикловой методической комиссией  
математических и естественнонаучных дисциплин  
в качестве электронного издания  
для использования в образовательном процессе

Кемерово 2018

Рецензенты:

Чичерин И. В. – кандидат технических наук, доцент кафедры информационных и автоматизированных производственных систем

Трусов А. Н. – кандидат технических наук, доцент кафедры информационных и автоматизированных производственных систем

**Зиновьев Василий Валентинович**

**Кузнецов Илья Витальевич**

**Стародубов Алексей Николаевич**

**Создание анимационной модели информационной системы в Proof Animation:** методические указания к практической работе [Электронный ресурс] по дисциплине «Математическое моделирование» для обучающихся специальности СПО 09.02.07 Информационные системы и программирование очной формы обучения / сост. В. В. Зиновьев, И. С. Кузнецов, А. Н. Стародубов; КузГТУ. – Электрон. издан. – Кемерово, 2018.

Приведено содержание практического занятия, материал, необходимый для успешного изучения дисциплины.

Назначение издания – помощь обучающимся в получении знаний по дисциплине «Математическое моделирование» и организация практической работы.

© КузГТУ, 2018

© Зиновьев В. В.,  
Кузнецов И. С.,  
Стародубов А. Н.,  
составление, 2018

**Цель работы** – изучить методы отображения движения элементов систем и процессов с помощью средств компьютерной анимации Proof Animation.

В ходе выполнения практической работы, обучающиеся должны изучить принципы анимационного отображения динамики систем и процессов при помощи языка Proof Animation и научиться применять различные способы отображения движения объектов.

## 1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

### 1.1. Язык компьютерной анимации Proof Animation

Анимация – это отображение динамики работы оборудования на компьютерной мнемосхеме технологического процесса, формируемое имитационной моделью процесса.

Язык компьютерной анимации Proof Animation может работать в комплексе с универсальными языками BASIC, C++, FORTRAN, PASCAL или специализированными языками GPSS, SIMAN, SIMPLE, SIMSCRIPT, SLAM, создающими стандартные ASCII-файлы. Proof Animation является универсальным программным обеспечением системы анимации для персональных компьютеров, позволяющим проектировщику создавать любые цветных двухмерные и изометрические рисунки, формировать набор команд для движения элементов рисунка, редактировать последовательность движений.

Применение компьютерной анимации дает возможности:

- проверить адекватность модели объекту в деталях и в целом и выявить трудноуловимые ошибки имитационного моделирования;
- доказать заказчику правильность работы имитационной модели;
- проигрывать различные ситуации и анализировать результаты моделирования непрограммирующему пользователю;
- вести обучение на модели.

Простые инструкции языка Proof Animation сводятся к заданию времени и конечных точек перемещения нарисованных объектов. Изображение процесса строят из статических и динамических элементов. Статическими элементами могут являться сервер, линии передачи данных, контуры участка цеха, станки, накопители и т.д. Их рисуют с помощью линий, дуг, прямоугольников, окружностей, имеющихся в меню программы. Возможны выбор цвета, ширины и типа линии; просмотр размеров и ориентации объекта до его ввода в рисунок; изображение кривой любой формы путем передвижения «мыши». Статическими элементами могут быть пояснительные надписи и таблицы для отображения показателей процесса. Динамические элементы накладываются на статический фон, меняя свои

размеры, форму, цвет или положение в процессе моделирования. При анимации различных систем к динамическим элементам можно отнести передаваемые данные, заготовки, транспортные средства, исполнительные органы.

Анимационная оболочка Student Proof Animation обрабатывает пять типов входных файлов и шесть выходных. Для анимации простых систем достаточно использовать два типа входных и три типа выходных файлов (рис. 1-1).

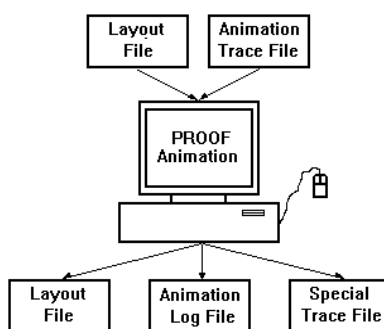


Рисунок 1-1. Входные и выходные файлы, в Student Proof Animation

*Входные файлы:*

- LayoutFile – файл разметки. Содержит описание всех статических и динамических объектов: фоновый текст, таблицы, контуры помещений, накопители, оборудование, людей и т.д.
- TraceFile – файл управления анимацией или файл трассировки. Включает команды Student Proof Animation, посредством которых перемещаются динамические объекты.

*Выходные файлы:*

- LayoutFile – файл разметки создается для сохранения нарисованных графических объектов. Выходной файл разметки подобен входному. После редактирования графических элементов текущее изображение сохраняется как файл с расширением .lay.
- LogFile – файл регистрации ошибок. Язык Proof Animation отображает на экране предупреждения или сообщения об ошибках. Информация об ошибках автоматически записывается в файле регистрации ошибок (LogFile) с расширением .log.
- SpecialTraceFile – файл управления анимацией. Управляет движением динамических объектов (рабочих органов, транспортных средств, деталей). Этот файл может генерироваться системой имитационного моделирования GPSS.

## Опции меню Proof Animation

После запуска программы анимации Proof Animation, путем загрузки файла `sp4.exe`, на экране появится оформление анимации (рис. 1-2) с активными спускающимися меню: File (файл), Mode (режим), View (вид), Run (запуск).



Рисунок 1-2. Видоболочки Proof Animation

Опция File предназначена для следующих операций с файлами:

- **OpenLayout&Trace** – открыть файлы графических элементов и управления анимацией;
- **OpenLayoutonly** – открыть только файл графических элементов;
- **SaveLayout** – сохранить файл графических элементов;
- **ChangeDirectory** – изменить текущий каталог;
- **ChangeDisk** – изменить текущий диск.

Опция Mode предназначена для выбора одного из следующих режимов анимации:

- **RunMode** – управление скоростью и текущим временем анимации;
- **DebugMode** – выполнение анимации по шагам файла управления (E-Step) или системного времени (T-Step);
- **DrawMode** – рисование и редактирование статических элементов анимации;
- **PathMode** – определение путей, по которым будут двигаться динамические объекты;
- **ClassMode** – рисование и редактирование динамических объектов анимации;
- **PresentationMode** – управление заданным сценарием презентации;
- **SetupMode** – изменение конфигурации (типа и цвета палитры, скорости мыши и т.д).

Опция View предназначена для управления размером, расположением и ориентацией окна анимации с помощью команд:

- **SelectView** – выбор окна анимации (анимация может содержать несколько окон со своим расположением, размером и ориентацией);
- **DefineView** – изменяет вид окна анимации;
- **Pan** – сдвигает окно просмотра на 25% влево (Left 25%), вправо (Right 25%), вверх (Up 25%) или вниз (Down 25%) без изменения масштаба просмотра;
- **Zoom** – изменяет масштаб окна просмотра;
- **ZoomBox** – изменяет размер выбранной части окна (по этой команде появляется прямоугольник, размеры которого регулируются мышью);

- **Zoom-to-fit** – автоматический выбор масштаба для отображения всех графических элементов анимации;
- **Out+Back** – перемещает центр вращения окна просмотра анимации;
- **Rotate** – вращает окно просмотра вокруг центра на заданный опцией «Degrees» угол;
- **Grid** – отображает масштабную сетку;
- **Refresh** – регенерирует изображение.

**Опция Time** предназначена для отображения текущего времени анимации и перемещения во времени по процессу (для изменения текущего времени необходимо щелкнуть левой клавишей мыши по текущему времени, при этом в нижней части экрана появится надпись Pleaseentertimetojump to (введите новое время), а затем ввести новое текущее время);

**Опция Speed** предназначена для отображения и изменения скорости анимации, которая измеряется отношением к реальному времени (для изменения скорости щелкните мышью по текущей скорости. На запрос в нижней части экрана: Pleaseentersimulion: viewingtimeratio введите новый коэффициент скорости);

**Опция +Faster/- Slower** предназначена для увеличения или уменьшения текущей скорости анимации шагами по 10%;

**Опция Pause** предназначена для временной остановки анимации;

**Опция Go** предназначена для начала или продолжения выполнения анимации.

## 1.2. Построение статических и динамических объектов в Proof Animation

Для создания статических объектов в Proof Animation используют режим **DrawMode**. Панель инструментов в DrawMode имеет следующий вид (рис.1-3.):

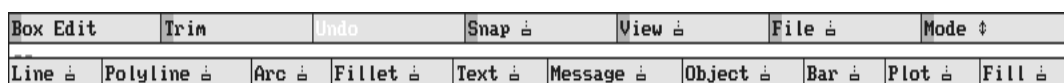


Рисунок 1-3. Панель инструментов DrawMode

Для создания динамических объектов – классов используют режим Class, в котором имеется точно такая же панель инструментальных средств, как и в режиме Draw. Отличие режима Class от режима Draw в том, что масштаб экрана в режиме Class уменьшен в 10 раз, а точка с координатами (0;0) находится в центре экрана.

### - Line – Добавление линии

Команду Line используют с соответствующим комплектом инструментальных средств (рис.1-4). Построение линии производят указаниями щелчками мыши ее начальной и конечной точки.

Line		
Start	X→	25.00
Start	Y→	45.00
End	X→	25.00
End	Y→	45.00
Length	→	0.00
Angle	→	0.00

Рисунок 1-4. Комплект инструментальных средств Line

- Start X – x – значение отметки начала Line.
- Start Y – y – значение отметки начала Line.
- End X – x – значение конечной отметки Line.
- End Y – y – значение конечной отметки Line.
- Length – общая длина Line.
- Angle – угол, на который Line выводится.

Редактирование линии осуществляют также с помощью команды Line. При редактировании конечные точки линии высвечиваются красными прямоугольниками, а появившееся меню комплекта инструментальных средств Line отразит текущие свойства линии (координаты x и y, длину и т.д.). Чтобы изменить длину или угол наклона линии, необходимо щелкнуть мышью по одной из высвеченных конечных точек и затем переместить ее в новое положение. Для изменения положения линии на экране указатель мыши устанавливают посередине линии и перетаскивают ее в другое место.

Для изменения цвета линии сначала выбирают в цветовой палитре (ColorPalette) нужный цвет, а затем щелчком мыши указывают линию.

**- Polyline – добавление ломаной линии**

Опция Polyline дает возможность вывести на экран соединенные сегменты Line любой формы. Polyline имеет комплект инструментальных средств, идентичный тому, который был описан в разделе Line. Этот комплект отражает свойства только той линии, которая в настоящее время выводится как часть операции Polyline.

**- Arc – добавление дуги**

При использовании Arc, комплект инструментальных средств задает следующие параметры дуги (рис. 1-5):

Arc		
Center	X→	26.00
Center	Y→	43.00
Start	X →	27.00
Start	Y →	43.00
End	X →	26.00
End	Y →	44.00
Radius	→	1.00
Start	∠ →	0.00
End	∠ →	90.00
Full Circle		
Direction	▶	CCW
Rotate	▶	+90
Track	▶	Location

Рисунок 1-5. Комплект инструментальных средств Arc

- Center X – значение по x центра Arc.
- Center Y – значение по y центра Arc.
- Start X – значение по x начальной отметки.
- Start Y – значение по y начальной отметки.
- End X – значение по x конечной отметки.
- End Y – значение по y конечной отметки.
- Radius – длина радиуса Arc.
- Start – угол от центра до начала отметки.
- End – угол от центра до конечной отметки.
- FullCircle – опция для преобразования текущей Arc в полный круг.
- Direction – направление угла Arc (по часовой стрелке или против часовой стрелки).
- Rotate + 90 – инструмент, который вращает Arc на 90 градусов – по часовой стрелке.
- Track – атрибут Arc (Location (Расположение), Radius (Радиус), или Angles (Углы) – могут управляться мышью).

- **Fill** – *заполнение контура цветом*

Fill позволяет заполнить ограниченную область текущим перьевым цветом. Ограниченная область может быть задана любой комбинацией Lines и Arcs. Область должна быть замкнута, иначе заполнится все неограниченное пространство. Для изменения текущего цвета области выбирают необходимый цвет в ColorPalette и указывают щелчком мыши область заполнения. Для отмены указанных действий используют Undo или Unfill.

- **BoxEdit** – *редактирование нескольких элементов*

BoxEdit – это мощный инструмент для редактирования группы элементов. Чтобы выбрать элементы, которые нужно отредактировать, обведите эти элементы рамкой блока.

При использовании BoxEdit (редактирование блока) появится следующее меню комплекта инструментальных средств (рис. 1-6):

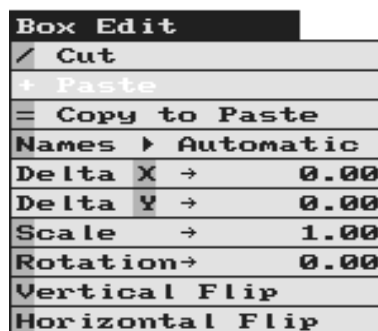


Рисунок 1-6. Комплект инструментальных средств BoxEdit



- Cut – удаляет выбранные элементы, после этого они доступны для вставки.
- Paste – вставка вырезанных или скопированных элементов.
- Name – определит, как обработать именованные элементы при вставке многократных копий.
- Delta X – относительное смещение по оси x.
- Delta Y – относительное смещение по оси y.
- Scale – масштаб.
- Rotation – относительный угол вращения.
- VerticalRip – инвертирует элементы “верхняя часть к нижней части”.
- HorizontalFlip – инвертирует элементы “слева направо”.

Блок задают щелчками мыши сначала одного, а затем другого противоположного по диагонали углов. Следует обратить внимание на то, что BoxEdit можно использовать для перемещения геометрических элементов в DrawMode и ClassMode, из режима в режим, из файла в файл (изменяя файлы размещения без выхода из Proof Animation).

### Создание классов в Proof Animation

При создании объекта в файле анимации необходимо определить его *ObjectClass* (класс объекта), который устанавливает форму и другие начальные свойства объекта.

Если в компьютерной анимации применяется множество подобных объектов, то можно задать один или несколько классов объектов. Свойства объекта копируют с «родительского» ObjectClass, когда объект создан, но любая принадлежность к классу может быть отменена для любого данного объекта.

Определение класса подобно рисунку фонового режима со следующими исключениями:

- Классы меньше, чем фоновый рисунок;
- Классы имеют «горячую» и «тыльную» точки;
- Классы не могут содержать графики и диаграммы.

### - ClassMode

Панель инструментов в ClassMode имеет следующий вид (рис. 1-7):

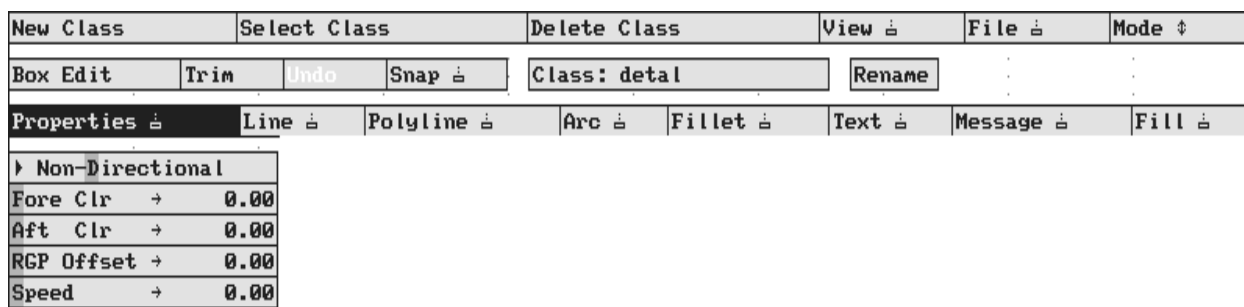


Рисунок 1-7. Панель инструментов в ClassMode

- **NewClass** (Новый класс)

NewClass используют для определения нового класса. После задания NewClass Proof Animation выдает подсказку для ввода имени нового класса. Имена классов чувствительны к регистру и могут быть длиной до 16 символов. Например, имя «FULLAGV» отличается от имени «FullAGV».

Если форма и свойства нового класса определены, то можно продолжать действия, т.к. класс автоматически включается в схему. Однако новые классы не сохраняются на диске, пока не будет задана команда **SaveLayout** из меню **File**.

- **SelectClass** (Выбор класса)

Опция SelectClass позволяет редактировать геометрические элементы или связи существующего класса. При выборе SelectClass в центре экрана появится список всех существующих классов для текущего размещения. Выбранный класс становится текущим.

- **DeleteClass** (Удаление класса)

Команда DeleteClass удаляет с подтверждением действий текущий класс из схемы. Чтобы отменить удаление класса надо вновь открыть предварительно сохраненную копию вашего файла, не сохраняя при этом последние изменения.

**Определение свойств класса**

После задания имени класса в верхнем левом углу экрана появляется меню комплекта инструментальных средств **Properties** (свойства). С помощью этого меню могут быть определены следующие четыре свойства класса: *directionality* (направление), *clearance* (диспетчерское разрешение) *RGP offset* (смещение) и *speed* (скорость). Указанные свойства могут быть определены как до, так и после создания рисунка.

- **HotPoint** (Горячая отметка)

В ClassMode Proof Animation рассматривает, перемещает и вращает объекты, созданные из класса, относительно горячей отметки. Горячая отметка ObjectClass имеет координаты (0,0). Для изменения координат горячей отметки используют BoxEdit (редактирование блока).

- **Directionality and Orientation** (Направлениеиориентация)

ObjectClass может быть задан как направленный или как ненаправленный. Первая опция в меню **Properties** задает режимы **Non-Directional** (ненаправленный) и **Directional** (направленный). По умолчанию устанавливается **Non-Directional**. Если установлен класс с направленными объектами, то указывают направления перемещения объектов. Если класс с ненаправленными объектами, то объекты перемещают по экрану в одном направлении. Для указания направления можно использовать команду **BoxEdit**, которая позволяет вращать геометрию класса.

- **Clearance** (Диспетчерское разрешение)

Для того чтобы объекты, выстроенные в линию, не сталкивались на пути, необходимо определить значение «диспетчерского разрешения». Диспетчерское разрешение наследуется объектом из класса.

Диспетчерское разрешение определяется в линейных единицах (по умолчанию – ноль). Имеются два значения диспетчерского разрешения: «fore» (перед) и «aft» (после). Значения диспетчерского разрешения созданных объектов могут быть изменены командой Object...Clearance (объект набора ... Clearance).

- **RGP Offset (RGP смещение)**

RGP смещение позволяет определять факультативную тыльную отметку для текущего класса. RGP действует как вторая отметка присоединения, когда направленные объекты двигаются по пути. RGP смещения должен быть установлен как отрицательное число, поскольку отметка RGP всегда находится позади горячей отметки.

- **Speed (Скорость)**

Все объекты перемещаются с определенной скоростью по пути. Скорость может быть задана для объекта или класса объектов. Скорость класса будет полезна только в тех случаях, когда скорости несходных объектов на пути зависят от классов объектов, например, для транспортных средств, перемещающихся по скоростной дороге.

- **Color (Цвет)**

Цвет – собственность класса. Каждый объект класса может иметь любую комбинацию цветов. Для изменения цвета элемента класса сначала щелчком мыши указывают элемент, а затем цвет. Для замены цвета класса сначала выделяют блок вокруг этого класса командой VoxEdit, а затем устанавливают цвет из цветовой палитры.

### **1.3. Движение в Proof Animation**

Большинство видов движения можно представить, как перемещение по заданному пути. Задание движения по определенному пути - одна из наиболее важных особенностей Proof Animation. Путь задается по линиям (Lines) и дугам (Arcs) и является невидимым элементом.

- **PathMode (Режим пути)**

Proof Animation обеспечивает специальный режим определения пути, который можно выбирать из меню **Mode**(режим). Пути показываются в цвете. Другие элементы отображаются черным цветом на сером фоне. В большинстве случаев пути показываются синим цветом. Путь, который необходимо исправить (Repairs), показывается белым цветом. У текущего выбранного пути цвет желтый, а у выбранного сегмента пути цвет красный. Меню **PathMode** имеет следующий вид (рис. 1-8):

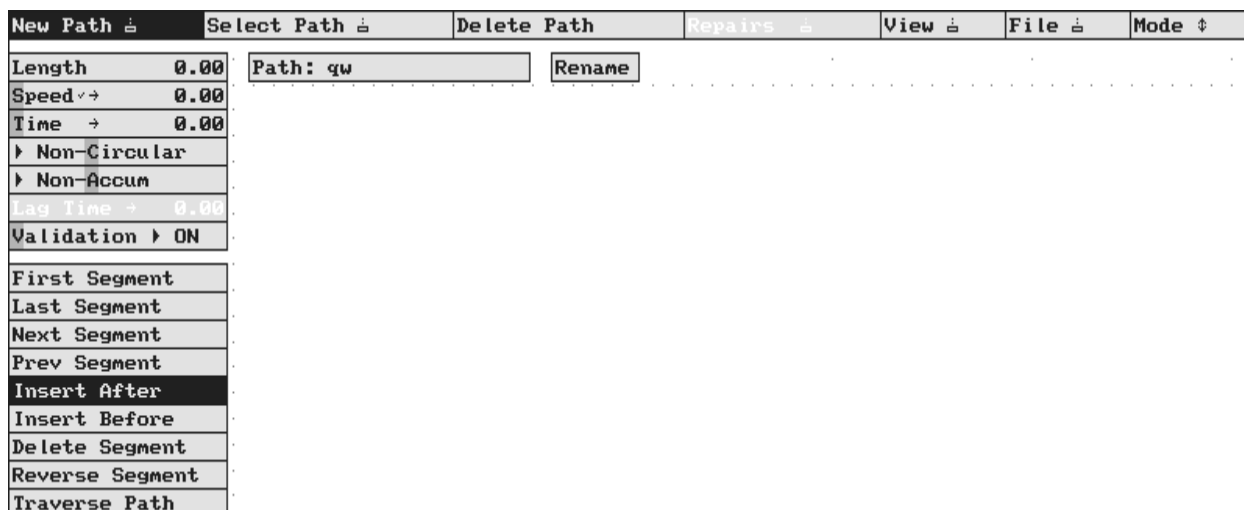


Рисунок 1-8. Панель инструментов в PathMode

Полоса меню PathMode (режим пути) расположена в верхней части экрана. Она содержит следующие опции: **NewPath** (новый путь), **SelectPath** (выбрать путь), **DeletePath** (удалить путь), **Repairs** (исправить), и уже знакомые **View** (вид), **File**(файл) и опции **Mode**(режим).

Команда **DeletePath** доступна только после того, как путь был задан или выбран для редактирования. Команда **Repairs** доступна, если изменения, сделанные в DrawMode (вывести режим), испортили сегменты одного или более путей.

После выбора команды **NewPath**(новый путь) **Proof Animation** просит ввести уникальное имя пути. Имя пути может содержать символы верхнего и нижнего регистров, и(или) цифры. Имя должно начинаться с символа и может включать до 16 символов.

После ввода имени пути можно начинать строить схему пути путем добавления сегментов.

- ***Path Properties (Свойства пути)***

Верхнее меню комплекта инструментальных средств отображает несколько свойств пути: **Length** (длину), **Speed** (скорость), **Time** (время), **Circularity** (циркуляцию), **Accumulation** (накопление), **LagTime** (запаздывание) и **Validation** (проверку).

- ***Length (Длина)***

**Proof Animation** автоматически определяет длину пути в своих единицах измерения. Длина пути рассматривают как расстояние по пути от начала первого сегмента до конца последнего.

- ***Speed (Скорость)***

**Speed** задает скорость двигающихся по пути объектов. После определения скорости на комплекте инструментальных средств появится метка, следующая за **Speed**. В этом случае время вычисляется и отображается автоматически, и основывается на определенной скорости и расчетной длине. Например, если длина пути и скорость равны соответственно 75,0 и

5, то время равно 15,0. Заданная скорость остается такой же для каждого последующего пути, пока скорость или время не будут изменены.

- **Time** (Время)

Командой Time задается время прохождения объекта от начала первого сегмента до конца последнего. Если время определено, то на комплекте инструментальных средств появится маленькая метка, следующая за словом Time. В этом случае скорость вычисляется и отображается автоматически, и основывается на определенном значении времени и расчетной длине. Например, если длина равна 75,0, а время равно 150, то скорость будет равна 0,5.

- **Accumulation** (Накопление)

Текущая установка Accumulation переносится на каждый последующий определяемый путь. Значение по умолчанию в PathMode – «Non – Accum» (ненакопляемый). Накопляемый путь отличается от ненакопляемого обработкой объектов, которые достигли конца пути.

Если путь накапливаемый, то объект движется к концу этого пути, чтобы «накопиться» позади других объектов, уже достигших конца этого пути. Когда первый объект удален из накапливаемого пути, другие объекты будут двигаться вперед автоматически. Накопляемый путь не может быть круговым.

- **Circularity** (Циркуляция)

У кругового пути нет конца. Объекты, которые достигают конца пути, переходят непосредственно к началу, даже если конец не соединен с началом. Круговой путь не может быть накапливаемым. Для круговых путей установка значения по умолчанию – “Off” (выключен). Текущая установка переносится к каждому последующему определяемому пути.

- **Validation** (Проверка правильности)

Validation выдает предупреждение Leapfrog, если объект удален из накапливаемого пути, или предупреждение Encroachment, если объекты на накапливаемом пути сталкиваются друг с другом. Опцию Validation можно переключить на **Validation \* ON** (включено) или **Validation \* OFF** (выключено).

**PathSegments** (Сегменты пути)

Пути в Proof Animation могут состоять из одного или более сегментов. Сегмент может относиться ко всей Line (линии) или Arc (дуге) либо к их частям. Сегменты пути упорядочены. В каждом пути имеется первый и последний сегменты. Каждый сегмент имеет направление. Направления сегментов обозначены на экране стрелкой в конце каждого сегмента. Эти стрелки не видны в режимах DrawMode и RunMode. Конец сегмента не маркирован.

**Комплект инструментальных средств создания и редактирования сегментов пути**

В нижнем меню комплекта инструментальных средств есть опции добавления и удаления сегментов. Для добавления сегментов к пути ис-

пользуют опцию **InsertAfter** (вставка после). Каждый новый сегмент начинается с конца предшествующего сегмента, соединенного с началом стрелки, и указывается щелчком мыши. При задании начала нового пути опция InsertAfter вызывается автоматически. Для исправления ошибок построения пути из сегментов используют клавиши <Backspace (возврат на один символ) > или <Delete (удалить)>.

- **InsertBefore** (*Вставка перед*), **InsertAfter** (*Вставка после*)

Если сегмент выбран, то в путь можно вставлять один или более сегментов перед ним, нажимая InsertBefore (вставка перед), или после него, нажимая InsertAfter (вставка после). Для нового сегмента пути InsertAfter выбирается автоматически.

- **DeleteSegment** (*Удалить сегмент*)

DeleteSegment позволяет удалить выбранный сегмент. После того как сегмент удален, выбирается предыдущий сегмент. Удалять сегменты можно неоднократно, вплоть до достижения начала пути. Если предыдущий сегмент отсутствует, то выбирается следующий сегмент. Можно удалять выбранные сегменты, используя клавиши <Backspace> или <Delete>.

- **ReverseSegment** (*Направить сегмент*)

ReverseSegment позволяет задавать направление выбранного сегмента. ReverseSegment полезен в том случае, если сегмент, особенно первый сегмент пути, направлен неправильно, и необходимо повторно выбрать сегмент. Для добавления сегментов после их изменения необходимо сначала нажать InsertAfter.

- **TraversePath** (*Двигаться по пути*)

TraversePath позволяет просмотреть последовательно весь путь по сегментам. Для выхода из TraversePath используют <Esc>.

#### **1.4. Файл управления анимацией (файл трассировки)**

Управление динамическими объектами в Proof Animation осуществляется посредством файла трассировки (TraceFile). Файл трассировки – содержит упорядоченную последовательность команд, в соответствии с которыми Proof Animation выполняет определенные действие в анимационной модели. Для написания файла управления вручную можно использовать текстовый редактор, создающий не форматированные текстовые файлы (например, встроенный редактор NC). Далее рассмотрим основные команды файла трассировки.

- Время компьютерной анимации

Команда TIME является основной командой файла трассировки компьютерной анимации в Proof Animation. Синтаксис команды TIME:

**TIME** <значение времени>

Команду TIME используют для того, чтобы задать часам компьютерной анимации новое значение. Proof Animation продолжает отображать

анимацию до тех пор, пока время, определенное в команде TIME, не будет достигнуто. После этого Proof Animation читает и обрабатывает следующую строку в файле следа компьютерной анимации.

- Создание объекта

Основным элементом в компьютерной анимации является Object (объект). Объект – это динамический элемент, который может перемещаться на экране. Над объектом выполняют следующие действия: создание, уничтожение, помещение в координатную точку, изменение цвета, изменение формы, перемещение по пути, перемещение от одной отметки до другой, изменение быстродействия, вращение.

При создании объекта вводится его спецификация – класс объектов. Это удобно, поскольку объект может наследовать свойства класса.

Для имен ObjectClass устанавливают некоторые простые правила. Верхние и строчные символы букв обрабатывают как уникальные символы в именах идентификатора. Каждое имя должно начинаться с буквенного символа (A-Z или a-z). Имена могут быть длиной не более шестнадцати символов.

Перед тем как объект появится на экране, его сначала нужно создать в файле следа анимации, для этого используют команду Create.

Синтаксис команды Create:

**CREATE** <имя класса><номер объекта>

Например: *CREATE Car 20*

Имя класса должно содержать имя класса объектов, который сохранен в файле размещения (\*.lay файл). Номер объекта является номером или именем Object (объекта). Номер и имя должны быть уникальными.

- Размещение объекта на экране

Proof Animation имеет несколько команд, которые заставляют Object появляться на экране. Основной является команда типа - PLACE ... AT .

Синтаксис PLACE ... AT:

**PLACE** <номер объекта> AT x y

Например: *PLACE 52 AT 12.5 25*

PLACE ... AT заставляет Object появляться в системе координат Proof Animation, но не заставляет его двигаться.

- Удаление объекта

Удаление объектов с экрана осуществляется командой DESTROY. При этом объект удаляется только с экрана, а на класс объектов данная команда не воздействует.

Синтаксис для DESTROY:

**DESTROY** <имя объекта>

- Установка и изменение цвета объекта

Если объект уже видим на экране, можно изменить его цвет при помощи команды SET COLOR в файле следа.

Синтаксис команды SET COLOR:

**SET** <имя объекта> COLOR

В Proof Animation используются цвета двух типов:

- **Цветафона:**

BACKDROP (Black)

LAYOUT (Lay)

- **Цвета активного слоя:**

BLUE или F1

RED – F2

WHITE – F3

YELLOW – F4

PINK – F5

TAN – F6

GREEN – F7

Для определения цвета, используется соответствующий номер с буквой F или английское название цвета. Например,

*SET 3 COLOR F2*

*SET BOX COLOR RED*

Объекты фоновых цветов двигаются без интерференции через другие объекты. Если два объекта цвета активного слоя соприкоснутся друг с другом, то область перекрытия будет показана в третьем отличном цвете. Если произойдет наложение двух одинаковых объектов одного активного цвета, то появится цвет самого нижнего слоя BACKDROP (объекты станут невидимы).

- Перемещение объектов между двумя точками

Движение по прямой между двумя точками. Применяется для перемещения объектов в точку, координаты которой известны или могут быть вычислены в ходе выполнения имитационной модели. Для такого перемещения используется команда MOVE.

Синтаксис команды MOVE:

**MOVE <имя объекта> (время движения) (x) (y) [RELATIVE]**

**MOVE <имя объекта>SPEED (скорость движения) (x) (y) [RELATIVE]**

При выполнении этой команды объекты двигаются по прямой линии между двумя точками. Объект сразу начинает движение из текущих координат к точке с координатами (x, y) с заданными временем или скоростью. Достигнув заданных координат, он останется в этой точке на время, соответствующее модулю времени, определенному в команде TIME. Если используется ключевое слово RELATIVE, то объект начнет перемещаться из текущих координат на заданное количество модулей в направлении x и y. Например, по команде: *MOVE 21 10 4 6 RELATIVE* объект 21 начнет перемещаться из текущих координат на 4 единицы в направлении x и на 6 единиц в направлении y за время 10.

Рассмотрим еще один пример. Допустим, что объект робокар «rob» создан и помещен в координатное пространство Proof Animation. Тогда в результате выполнения подпрограммы:

TIME 50



MOVE rob 10 40 50  
TIME 60

По достижении текущего времени TIME 50 объект «rob» начнет перемещаться за время 10 в точку с координатами  $x=40$ ,  $y=50$ . Его перемещение будет продолжаться  $60-50=10$  единиц времени.

- Перемещение объектов по пути

Если Object помещен на Path и быстроедействие Path-и или Object-а больше чем ноль, Object автоматически будет двигаться по Path-и, пока не достигнет конца, если некоторая другая команда файла следа компьютерной анимации (например, DESTROY или END) не прервет движение.

Команда для использования Paths – PLACE ... ON. Синтаксис для PLACE ... ON:

**PLACE <номер объекта> ON <имя пути>**

Например: *PLACE Box ON Conveyor*

- Вращение Объектов

Если объект помещен на экран, его можно вращать. Причем это вращение, не зависит от действий над объектами, т.е. во время вращения можно изменять цвет объекта, перемещать его командой MOVE или помещать на путь. Для вращения объектов используется команда ROTATE.

Синтаксис команды:

**ROTATE <имя объекта> [TO] <угол> TIME <время> <шаг>**

**ROTATE <имя объекта> TO <угол> SPEED <скорость> <шаг>**

Угол поворота и шаг измеряются в градусах, продолжительность поворота в единицах времени Proof Animation, скорость поворота в градусах на единицу времени. Положительный угол – против часовой стрелки, а отрицательный – по часовой. Если задать угол, но не задавать продолжительность и скорость поворота, то вращение произойдет мгновенно. Если определить скорость поворота, но не определять угол, то объект будет вращаться с этой скоростью, до выполнения другой команды ROTATE или до отмены этой команды. Использование ROTATE со скоростью 0, остановит вращение. Плавность вращательного движения управляется величиной шага. По умолчанию Proof Animation вращает объект с приращением 30 градусов.

Вращение может быть абсолютным и относительным. При абсолютном вращении в команде ROTATE перед углом ставится опция «TO». Если опустить опцию «TO» в команде ROTATE, то вращение будет относительным.

**Пример абсолютного вращения:**

*ROTATE 1 TO 90*

... мгновенно вращает объект 1 на угол 90 градусов против часовой стрелки.

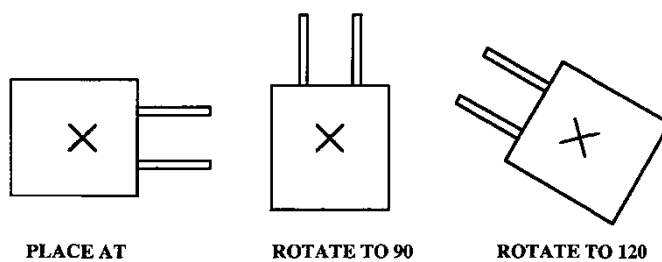


Рисунок 1-9. Пример абсолютного вращения

### *ROTATE 1 120*

... мгновенно вращает объект 1 на 120 градусов против часовой стрелки относительно положения, в которое его поместили командой `PLACE ... ON` (рис. 1-9).

### **Пример относительного вращения:**

### *ROTATE 1 90*

... мгновенно вращает объект 1 на угол 90 градусов против часовой стрелки (рис. 1-10).

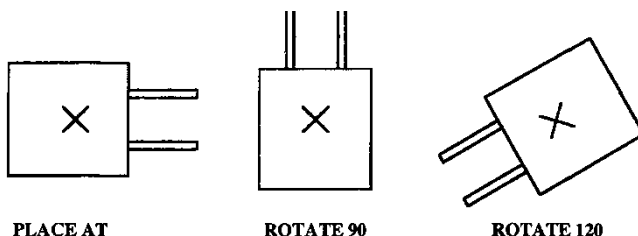


Рисунок 1-10. Пример относительного вращения

### *ROTATE 1 120*

... мгновенно вращает объект 1 на 120 градусов против часовой стрелки относительно текущей позиции.

Другие примеры команды `ROTATE`:

### *ROTATE 1 SPEED -90*

... непрерывно вращает объект 1 по часовой стрелке относительно горячей точки на 360 градусов (четыре единицы времени `Proof Animation`). Другая команда `ROTATE` с определенным углом остановит непрерывное вращение.

### *ROTATE 1 90 TIME 10*

### *ROTATE 1 90 SPEED 9*

... эквивалентные команды вращают объект 1 в три шага (*т.к. по умолчанию 30 градусов*) за 10 единиц времени `Proof Animation`. Вращение заканчивается, когда объект повернется на 90 градусов против часовой стрелки относительно текущей ориентации.

### *ROTATE 1 90 TIME 10 STEP 5*

... то же, что и предыдущем примере, но объект будет делать 18 шагов вместо трех, для поворота на 90 градусов (здесь отменяется заданный по умолчанию размер шага – 30 градусов).

- Изменение содержания сообщения

Прежде чем вывести динамическое сообщение, необходимо его создать и определить его прототип в цвете Backdrop режиме DrawMode по команде Message. Изменить сообщение (его цвет и содержание) можно по команде WRITE.

Синтаксис команды WRITE:

**WRITE**<имя сообщения><текст сообщения>

Например, покоманде:

*WRITE line1 Control of robocar*

- в линию line1 (должна быть определена в разметке) помещается надпись «Control of robocar» (Управление робокаром).

- Завершение анимации

Файл следа обязательно должен заканчиваться командой END.

Синтаксис команды END:

**END.**

## 2. ПРИМЕР АНИМАЦИОННОГО МОДЕЛИРОВАНИЯ

Процедуру создания анимационной модели покажем на примере изготовления деталей разными типами станков.

Заготовка последовательно обрабатывается двумя типами станков (типом А и В). Группа станков А включает два станка. В начале процесса заготовка в течение 15 единиц времени транспортируется до станков типа А откуда поступает на обработку на первый станок группы. Здесь происходит ее обработка в течение 10 единиц времени. После этого, полуфабрикат, в течение 5 единиц времени, поступает на обработку на станок типа В. Окончательная обработка детали занимает 20 единиц времени. Готовая деталь транспортируется, в течении 5 единиц в накопитель. После этого к станкам группы А поступает вторая заготовка и процесс повторяется за исключением того, что в группе А заготовка обрабатывается вторым станком. Необходимо создать анимационную модель технологического процесса.

### Метод построения модели

Войдем в Student Proof Animation путем загрузки файла spa.exe. Выберем в верхнем меню пункт Mode, а в нем режим DrawMode. С помощью набора команд Student Proof Animation нарисуем статические элементы технологии станки, траектории движения деталей (в невидимом цвете Backdrop) и необходимый текст (рис.2-1).

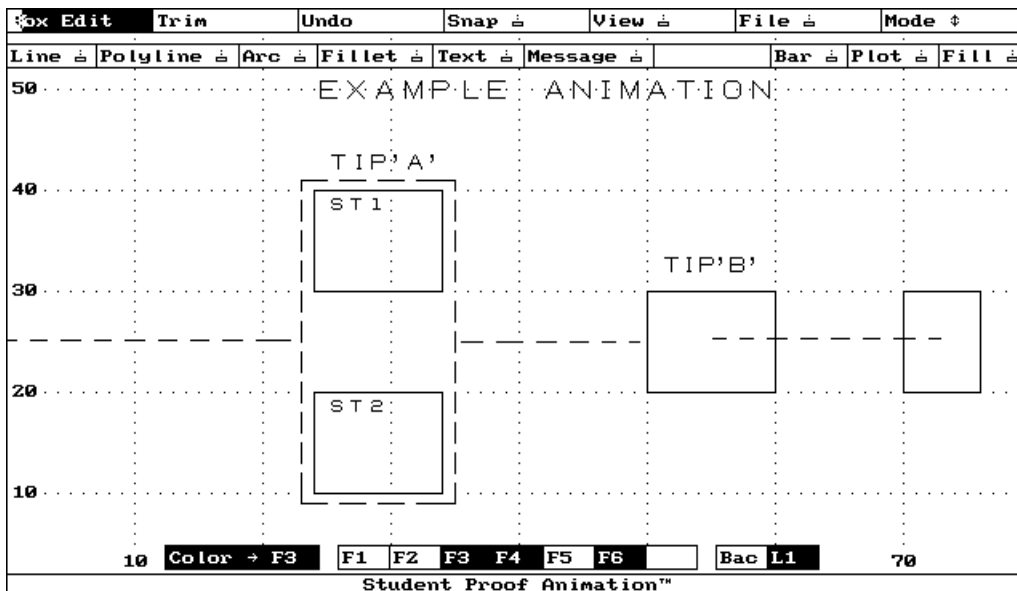


Рисунок 2-1. Статические элементы технологии  
(пунктиром показаны траектории движения динамических элементов,  
в режиме Draw они не видимы)

В режиме ClassMode выберем опцию «Newclass» и нарисуем динамический объект (класс) – деталь (под именем det) (рис. 2-2).

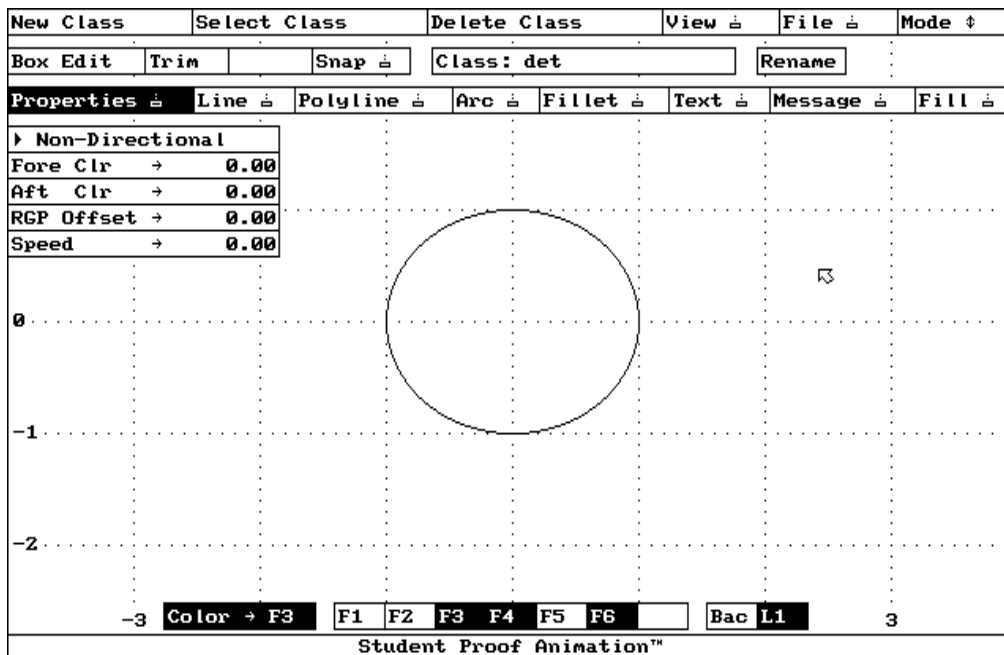


Рисунок 2-2. Динамический объект технологии

В режиме PathMode выберем опцию «NewPath» и создадим пути на основе уже нарисованных (в режиме Draw) траекторий движения деталей и назовем их p1, p2 и p3. Выберем в меню опцию «Time» и впишем время движения по пути, равное 15 для p1 и 5 для p2 и p3 (рис. 2-3).

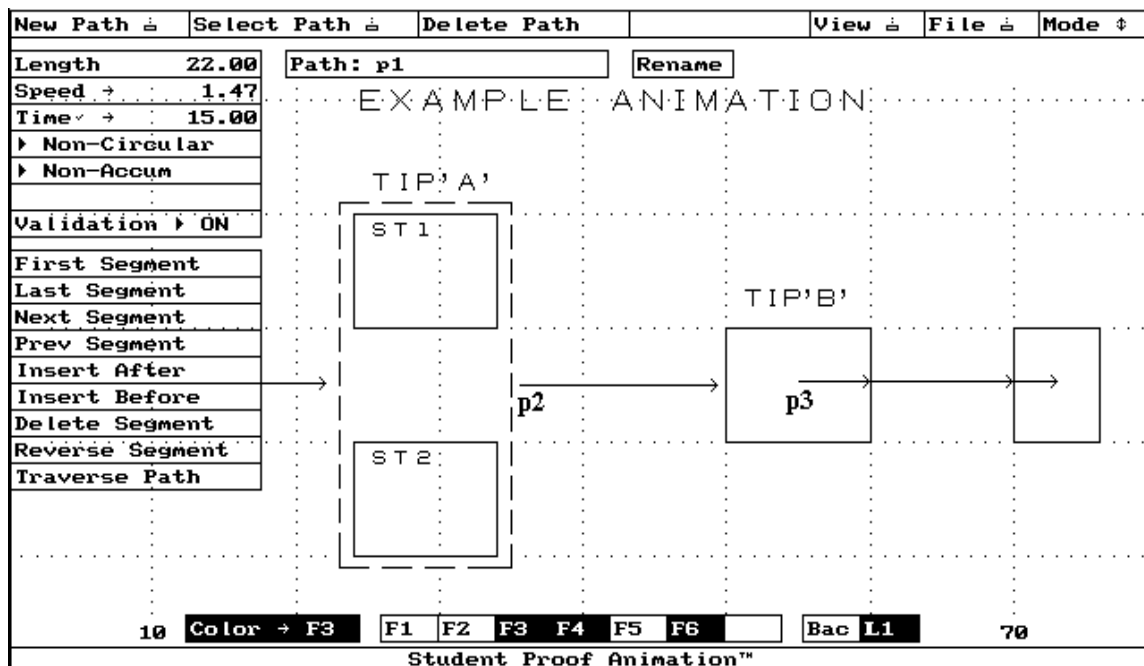


Рисунок 2-3. Пути движения динамических объектов

Сохраним файл разметки как `primera.lay` и выйдем из Proof Animation.

При помощи текстового редактора (Far, NC, VC, Блокнот, WordPad и т.п.), формирующего ASCII файлы, используя команды Student Proof Animation для управления динамическими объектами, создадим файл трассировки с тем же названием, что и файл разметки (\*.lay файл), но с расширением \*.atf – `primera.atf`. Листинг файла трассировки приведен ниже:

```

TIME 0
CREATE det 1
PLACE 1 ON p1
TIME 15
PLACE 1 29 15
TIME 25
SET 1 COLOR YELLOW
PLACE 1 ON p2
TIME 30
PLACE 1 55 25
TIME 50
SET 1 COLOR RED
PLACE 1 ON p3
TIME 55
CREATE det 2
PLACE 2 ON p1
TIME 70
PLACE 2 29 35
TIME 80

```

```

SET 2 COLOR YELLOW
PLACE 2 ON p2
TIME 85
PLACE 2 55 25
TIME 105
SET 2 COLOR RED
PLACE 2 ONp3
TIME 110
END

```

Сохраним этот файл в той же директории, что и файл разметки.

После сохранения файла управления (как `primera.atf`) запустим анимацию, выбрав файл `sra.exe` и нажав клавишу `<Enter>`. Выведем на экран анимационную модель технологии путем входа с помощью мыши в пункт `"File"` верхнего горизонтального меню, подпункт `"OpenLayout&Trace"`. В появившемся вертикальном меню выберем пункт `primera` путем нажатия на нем клавиши мыши. В режиме `«RUN»` командой `«Go»` запустим программу на выполнение.

На рис. 2-4 представлен фрагмент анимации технологии изготовления деталей.

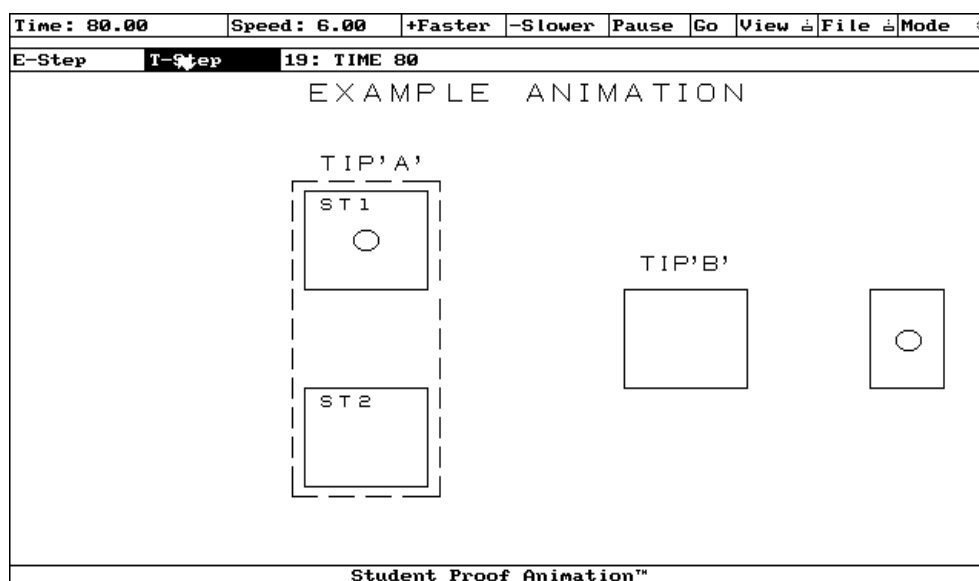


Рисунок 2-4. Фрагмент анимации технологии изготовления деталей

### 3. ЗАДАНИЕ И ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Сервер в локальной сети проводит опрос пяти компьютеров на поддержание подключения. Сигнал, посланный сервером на компьютер №1, отправляется на остальные компьютеры по локальной сети до тех пор, пока от последнего компьютера в опросе (компьютер №5) не вернется к серверу с результатами опроса (рис. 3-1).

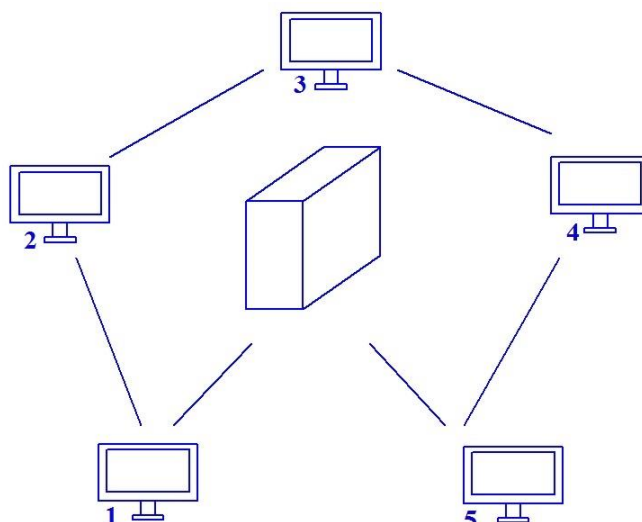


Рисунок 3-1. Схема подключения компьютеров в локальной сети.

1. Войдите в Student Proof Animation путем загрузки файла sp4.exe.
2. Выберите в верхнем меню пункт Mode, а в нем режим DrawMode.
3. С помощью набора команд Student Proof Animation нарисуйте в масштабе маршрут опроса в соответствии с рис. 3-1.
4. Отобразите перемещение объектов, по заданному маршруту, используя команду MOVE файла управления анимацией (без использования режима Path).
5. В том же меню выберите режим ClassMode и создайте объект – прямоугольник (заявка) с вершинами (-1,-1), (2,-1), (2,1) и (-1,1) в координатах Student Proof Animation.
6. При помощи любого редактора, формирующего ASCII файлы, используя команды Student Proof Animation для управления динамическими объектами, создайте файл управления анимацией (файл .atf), в котором, через каждые 5 единиц времени создается объект, в отметке «Старт», помещается на путь, движется до отметки «Финиш» и уничтожается. Анимация должна завершиться в момент, когда будут уничтожены 5 объектов.
7. Запустите анимацию.

#### 4. ТРЕБОВАНИЯ К ОТЧЕТУ

Отчет о работе должен содержать:

1. Задание и исходные данные по лабораторной работе.
2. Два файла статических и динамических элементов. lay.
3. Два файла управления динамическими объектами. atf.

## 5. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего используется компьютерная анимация и язык Proof Animation?
2. Каково назначение меню режимов (Mode) в Proof Animation?
3. Как изменить масштаб и скорость анимации?
4. Какие виды движения объектов используются в Proof Animation?
5. Что необходимо сделать для создания динамического объекта в Proof Animation?
6. Какое минимальное количество файлов необходимо для запуска анимации в Proof Animation?