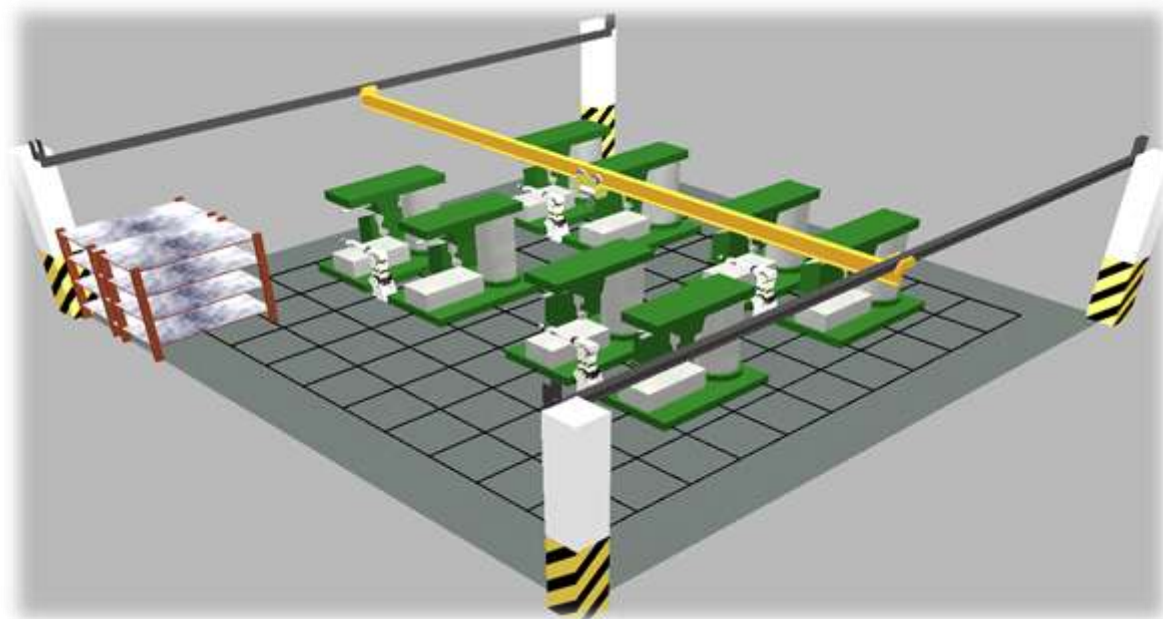




А. Н. Стародубов В. В. Зиновьев И. С. Кузнецов

КОМПЬЮТЕРНАЯ ИМИТАЦИЯ И АНИМАЦИЯ ПРОИЗВОДСТВЕННЫХ СИСТЕМ

Учебное пособие



Кемерово 2018

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

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

КОМПЬЮТЕРНАЯ ИМИТАЦИЯ И АНИМАЦИЯ ПРОИЗВОДСТВЕННЫХ СИСТЕМ

Учебное пособие

Кемерово 2018

УДК 004.94:658.51(075.8)

Рецензенты:

Заведующий лабораторией моделирования горнотехнических систем ФГБНУ «Федеральный исследовательский центр угля и углехимии Сибирского отделения Российской академии наук» доктор технических наук, профессор РАН А. Е. Майоров.

Заведующий кафедрой «Автоматизация производственных процессов и автоматизированные системы управления» ФГБОУ ВО «Кемеровский государственный университет» кандидат технических наук Р. В. Котляров.

Стародубов, А. Н. **Компьютерная имитация и анимация производственных систем:** учеб, пособие / А. Н. Стародубов, В. В. Зиновьев, И. С. Кузнецов; КузГТУ. – Кемерово, 2018. – 94 с.
ISBN 978-5-906969-75-0

В учебном пособии представлены основные понятия, необходимые для освоения технологии создания компьютерной анимации при исследовании производственных систем и процессов.

Изложенный материал позволит студентам овладеть подходами, методами и приёмами описания и исследования производственных систем и процессов средствами разработки анимации, приобрести опыт решения практических задач на анимационных моделях производственных систем и процессов с использованием программных продуктов Proof Animation и GPSS Studio, усвоить особенности отображения производственных систем и процессов с различной структурой и параметрами.

Предназначено для студентов направления подготовки 15.03.04 «Автоматизация технологических процессов и производств».

Печатается по решению редакционно-издательского совета КузГТУ.

УДК 004.94:658.51(075.8)

© КузГТУ, 2018

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

ISBN 978-5-906969-75-0

ОГЛАВЛЕНИЕ

Предисловие	4
Введение	5
1 Язык компьютерной анимации Proof Animation.....	17
Контрольные вопросы к разделу 1.....	39
2 Связь анимации с имитационной моделью	40
Контрольные вопросы к разделу 2.....	61
3 Анимационное представление систем в GPSS Studio.....	62
Контрольные вопросы к разделу 3.....	78
Заключение	79
Приложение 1	82
Приложение 2.....	86
Список рекомендуемой литературы	91

ПРЕДИСЛОВИЕ

Учебное пособие является составной частью учебно-методического комплекса по дисциплине «Компьютерная имитация производственных систем» и предназначено для студентов направления подготовки 15.03.04 «Автоматизация технологических процессов и производств».

Учебное пособие знакомит читателей с основными понятиями анимационного представления при исследовании, проектировании и эксплуатации технологических процессов и производственных систем.

Одним из наиболее распространенных специализированных языков имитационного моделирования является GPSS, предложенный фирмой IBM в 1962 г. Он развивался во многих версиях. В настоящем учебном пособии рассмотрены различные версии языка: GPSS/H, GPSS World и GPSS Studio – с точки зрения разработки анимационных моделей.

Недостатком имитационного моделирования является то, что необходимо интерпретировать результаты моделирования. Сам процесс имитации скрыт от экспериментатора. Поэтому специализированные языки имитационного моделирования соединяются с программами компьютерной анимации (CIMAN – CINEMA, GPSS/H – Proof Animation), позволяющими отображать динамику процесса на экране компьютера в соответствии с работой имитационной модели и выводить результаты на монитор. Более поздние, современные версии (GPSS World и GPSS Studio) уже включают возможность анимационного представления моделей систем.

Авторы приносят извинения у читателей за возможные допущенные ошибки и предлагают направлять свои пожелания и отзывы по адресу: a.n.starodubov@gmail.com

ВВЕДЕНИЕ

На протяжении всего своего существования человек пытался отразить движение в своем искусстве. Первые попытки передачи движения в рисунке относятся примерно к 2000 году до нашей эры (Египет). Еще один пример движения найден в пещерах Северной Испании: это рисунок кабана с восемью ногами. Сегодня передача движения может быть реализована средствами анимации.

Анимация (animation) – производное от латинского «anima» – душа, следовательно, анимация означает одушевление или оживление. В нашем кино анимацию чаще называют мультипликацией (дословно – «размножение»). Искусство анимации, как это ни парадоксально, старше самого кино, которое во многом именно ей обязано своим рождением.

Чтобы понять, как действует анимация, создадим на бумаге последовательность рисунков, отличающихся элементами, которые должны меняться или двигаться. После создания рисунков их последовательность фотографируется в правильном порядке по одному снимку за раз. При воспроизведении фильма данная последовательность неподвижных изображений будет восприниматься как непрерывное движущееся изображение.

Анимация используется в разных сферах нашей жизни:

- рекламе;
- обучении;
- исследованиях;
- искусстве;
- фильмах;
- видео;
- интернете;
- мультимедийных презентациях.

На данный момент существует множество видов анимации и способов её создания, рассмотрим их подробнее.

1. Классическая анимация

Классическая (традиционная) анимация представляет собой поочередную смену рисунков, каждый из которых нарисован отдельно (рис. 1). Это очень длительный процесс, так как аниматорам

приходится отдельно создавать каждый кадр, что очень трудоёмко и отнимает много времени даже у большого коллектива художников.

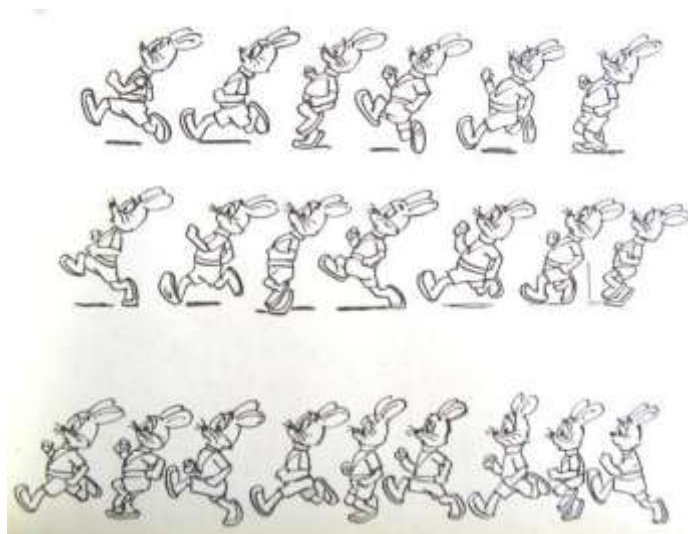


Рисунок 1 – Пример традиционной анимации

Затем была придумана послойная техника рисования объектов и фонов на прозрачных плёнках, накладываемых друг на друга. На одном слое можно было разместить задний фон, на другом – неподвижные части тел персонажей, на третьем – подвижные и т. д. Это значительно уменьшило трудоёмкость работ, так как не нужно было рисовать каждый кадр с нуля. Впервые послойную технику применил Уолт Дисней.

Достоинством рисованной мультипликации является её техническая простота (в пределах достаточен показ самих рисунков, не требуется даже кинооборудование). Также рисованная мультипликация легко поддаётся разделению труда мультипликаторов и созданию «конвейера», наиболее типичным представителем которого была мультипликация Уолта Диснея.

В настоящий момент большинство рисованных фильмов производится в Японии. Студии Уолта Диснея либо закрыты, либо переведены на 3D-технологии. Но большое количество мультсериалов продолжают делать по классической технологии.

2. Стоп-кадровая (кукольная) анимация

При создании такой анимации, размещенные в пространстве объекты фиксируются кадром, после чего их положение изменяется и вновь фиксируется.

Кукольная анимация – метод объёмной мультипликации. При создании используется сцена-макет и куклы-актёры. Сцена фотографируется покадрово, после каждого кадра в сцену вносятся минимальные изменения (например изменяется поза куклы). При воспроизведении полученной последовательности кадров возникает иллюзия движения объектов.



Рисунок 2 – Пример кукольной анимации

Пластилиновая анимация – вид анимации. Фильмы делаются путём покадровой съёмки пластилиновых объектов с модификацией (этих объектов) в промежутках между кадрами.

В пластилиновой анимации существует несколько техник.

– **Перекладка:** композиция состоит из нескольких слоёв персонажей и декораций, которые располагаются на нескольких стёклах, расположенных друг над другом, камера находится вертикально над стёклами. Персонажи и декорации для этого вида анимации делаются специальной, плоской формы. В настоящее время слои снимаются по отдельности и совмещаются при компьютерном монтаже. Этот вид анимации используется для удобства анимирования персонажей.



Рисунок 3 – Пример пластилиновой анимации по технике «Перекидка»

– **Объёмная анимация:** объёмные, «настоящие» персонажи располагаются в объёмной декорации. Работать в этой технике гораздо сложнее, поскольку анимировать персонажей приходится в пространстве; их необходимо специально укреплять в декорации, иногда используя дополнительные опоры и подвески.



Рисунок 4 – Пример пластилиновой анимации по технике «Объёмной анимации»

– **Комбинированная анимация:** персонажи анимируются по отдельности и снимаются на фоне синего экрана, после чего «вживляются» в снятые отдельно пластилиновые декорации. В данном виде пластилиновой анимации основной объём работы приходится не на работу с пластилином, а на работу с компьютером.

3. Песочная анимация

Песочная анимация, рисование песком, песочная фантазия, сыпучая анимация или техника порошка – это стиль изобразительного искусства, а также технология создания анимационных фильмов, появилась в 70-е годы XX века. Метод позволяет делать не только мультипликационные фильмы, но и шоу-номера для «живого» зрительного зала.



Рисунок 5 – Пример создания песочной анимации

Лёгкий порошок – обычно очищенный и просеянный песок, но также соль, кофе, или что-то другое – тонкими слоями наносится на стекло; с помощью диапроектора или световой доски получающееся изображение можно передавать на экран. Обычно все действия выполняются руками, но в качестве приспособлений могут использоваться кисточки.

В песочной анимации художник не просто рисует эффектные картинки, он создаёт целый сюжет, в котором каждое новое изображение как бы вырастает из предыдущего. А еще во время живого шоу песочная анимация сопровождается музыкой, подобранной

настолько удачно и точно, что, кажется, каждая нота соответствует сюжету работы.



Рисунок 6 – Пример готовой песочной анимации

4. Спрайтовая анимация

Спрайтовая анимация реализуется при помощи языка программирования. Широкое применение такой вид анимации получил в компьютерных играх, в особенности для прорисовок персонажей.

Простейший способ создания анимации заключается в использовании спрайтов. При поочередном отображении этих кадров в окне приложения с достаточно высокой скоростью возникает эффект анимации.

Спрайт (sprite) – небольшое изображение, свободно перемещающееся по монитору. В первоначальном смысле слова этот термин применялся только для аппаратно выводимых изображений. Собственно, лишь один настоящий спрайт можно встретить на IBM PC – аппаратный курсор мыши. При архитектуре x86 под спрайтом принято понимать программно выводимое изображение, которое может иметь сложную форму и передвигаться поверх фона, не затирая его.



Рисунок 7 – Пример спрайт анимации

5. Морфинг

Морфинг – преобразование одного объекта в другой за счёт генерации заданного количества промежуточных кадров (технология в компьютерной анимации, визуальный эффект, создающий впечатление плавной трансформации одного объекта в другой).

Для создания эффекта используются как минимум два изображения, на которых художник задаёт в зависимости от используемого программного обеспечения опорные фигуры или ключевые точки (т. н. маркеры, или метки), которые помогают компьютеру выполнить правильный морфинг, то есть создать изображения промежуточных состояний.



Рисунок 8 – Пример морфинг анимации

6. 2D-анимация

2D-анимация – создание подвижных объектов или сцен с использованием компьютерных технологий. Область применения такой анимации чрезвычайно широка: от рекламных роликов до компьютерных игр.

В зависимости от типа используемых в ролике изображений выделяют векторную анимацию и растровую. Соответственно различаются и технические приёмы, и программное обеспечение.

Процесс «оживления» пиксельных картинок не слишком отличается от классической анимации. Точно так же сначала прорабатывается объект или сцена, тщательно отрисовываются жесты, мими-

ка, позы. Затем получаемые кадры используются для покадрового монтажа в специальных программах.

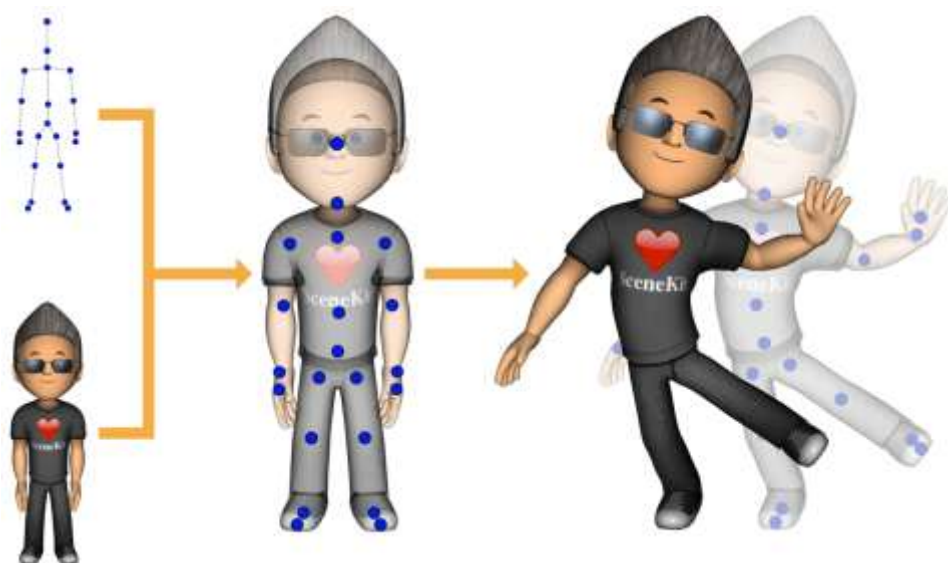


Рисунок 9 – Пример растровой 2D-анимации

Несколько проще работать с векторной анимацией, так как она позволяет осуществлять «перетекание» объекта из одного кадра в другой за счёт математических построений. Процедура кейфрейминга проста. Различные положения объекта фиксируются в специальных ключевых или опорных кадрах (от англ. keyframe – ключевой кадр), а промежуточные картинки генерируются системой автоматически.

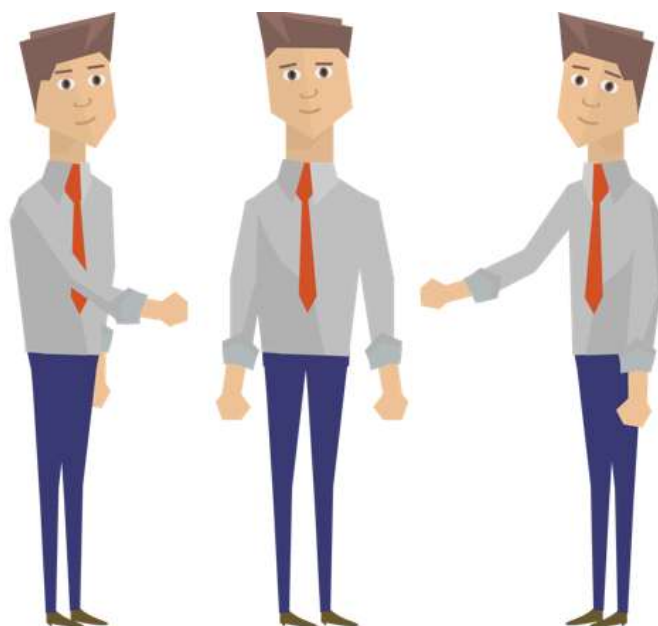


Рисунок 10 – Пример векторной 2D-анимации

Самыми распространенными форматами для 2D-анимации являются AVI, SWF, GIF и д. р.

Существует множество программ для создания 2D-анимации. В Приложении 1 представлена сравнительная таблица программных продуктов для создания 2D-анимации и моделирования и анимирования систем массового обслуживания (СМО).

7. 3D-анимация

3D-анимация создаётся при помощи специальных программ (например, 3D MAX). Картинки получаются путём визуализации сцены, а каждая сцена представляет собой набор объектов, источников света, текстур.

С развитием высоких технологий большую популярность приобретает трёхмерная графика (3D). Именно она позволяет создавать различные объёмные объекты: от макетов зданий до проекций новых миров. Технология 3D также существенно упрощает процесс анимации. Достаточно создать сцену, персонажей и задать параметры перемещения.

Таким образом, среди множества областей применения 3D-графики основными являются следующие.

– 3D-моделирование

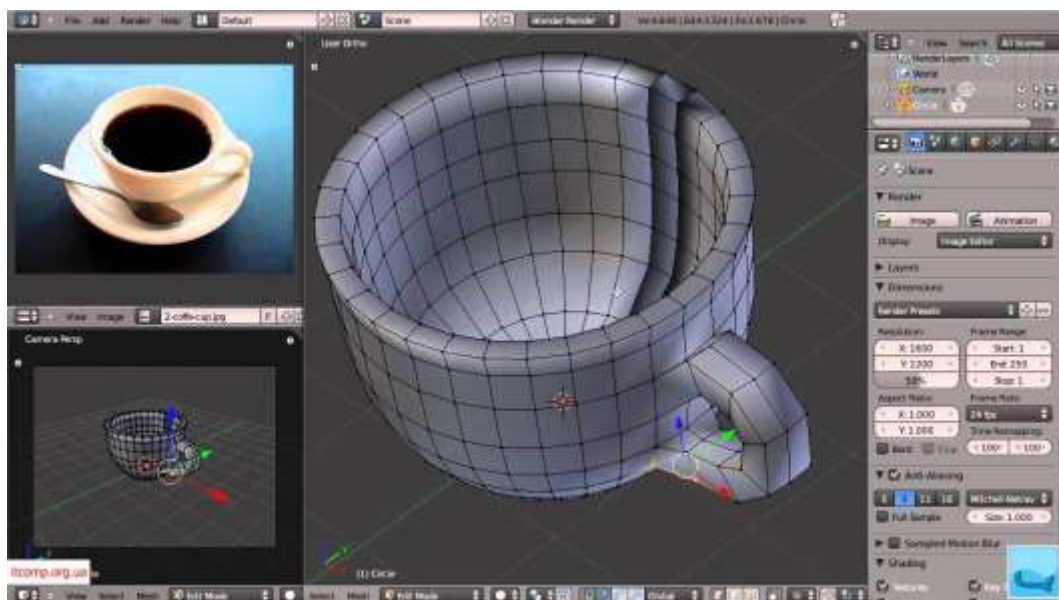


Рисунок 11 – Пример 3D-моделирования

Посредством специальных программ разрабатываются игровые или фотореалистичные модели существующих или планируемых объектов. Создаваемые модели могут использоваться в играх,

рекламных роликах, презентациях и так далее. К 3D-моделированию прибегают и при подготовке архитектурных проектов. Создание объёмных моделей в данном случае позволяет не только оценить привлекательность проектируемых зданий, но и изучить их параметры в целом, обнаружить возможные ошибки и исправить их на стадии проекта.

– **3D-анимация**

Разработка трёхмерных персонажей для создания анимационных роликов, а также анимация архитектурных и иных объектов для мультимедийных презентаций.



Рисунок 12 – Пример 3D-анимации

– **Визуализация**



Рисунок 13 – Пример Визуализации

Технология 3D позволяет с высокой точностью визуализировать архитектурные и иные объекты. Данный метод используется в архитектуре, сфере интерьерного дизайна, а также для оформления различных выставочных стендов и мест продаж.

Данный вид технологии позволяет осуществлять дизайн объектов на совершенно ином уровне. При этом требуются значительные ресурсы, навыки и знания. Для создания трёхмерной реальности понадобится довольно мощный компьютер.

Конфигурация компьютера и выбор оптимального софта зависят от целей применения трёхмерной графики. Создание интерьеров занимает гораздо меньше ресурсов и оказывает меньшую нагрузку на процессор, чем 3D-анимация.

8. Захват движения

Захват движения – первое направление анимации, которое даёт возможность передавать естественные, реалистичные движения в реальном времени.

Датчики прикрепляются на живого актёра в тех местах, которые будут приведены в соответствие с контрольными точками компьютерной модели для ввода и оцифровки движения. Координаты актёра и его ориентация в пространстве передаются графической станции, и анимационные модели оживают.



Рисунок 14 – Пример анимации «Захват движения»

9. Анимация технологических процессов

Отдельно выделим пользующуюся сегодня всё большей популярностью анимацию технологических процессов.

В данном случае анимация – это отображение динамики работы оборудования на компьютерной мнемосхеме технологического процесса, формируемое имитационной моделью процесса.

Зачастую визуальное отображение технологического процесса необходимо не только самим конструкторам как помощь в разработке, но и для «внешних» нужд предприятия: клиенты и партнёры должны легко уяснить суть и разобраться в этапах процесса, даже не имея специальных знаний в области деятельности предприятия.

Создание двух- и трёхмерных моделей оказывается наиболее эффективным и экономичным способом наглядного иллюстрирования машин, механизмов и всего процесса в целом. Анимация позволяет избежать чрезмерных финансовых затрат, сэкономить массу человеко-часов и тем самым способствовать повышению экономической эффективности технологического процесса при его реализации.

Наиболее популярными методами визуализации процессов в наши дни являются 3D-анимация и 3D-модели. Часто объединяют видеоматериалы и компьютерную графику. Визуализация стала наиболее эффективным и выгодным методом доступного демонстрирования действующих на предприятии машин, установок и всего производственного процесса в целом.

Визуализация незаменима в том случае, когда требуется внедрение или разработка инновационной и ранее не применявшейся технологии. При помощи компьютерной графики и 3D-анимации можно наглядно показать все преимущества и перспективы технологии доступно и эффективно.

Возможности 3D-графики на порядок превосходят возможности схем, чертежей и рисунков, используемых для графического отображения технологического процесса.

Несмотря на огромные дизайнерские возможности визуализации, при применении её для технологических процессов задача дизайнера очень практична: облегчить понимание технологического процесса. Максимальная детальная проработка обычно не является целесообразной.

Разработка и создание динамических моделей, наглядно представляющих технологический процесс, преследует несколько целей:

- привлечение инвесторов за счёт наглядной демонстрации принципа действия объекта;
- знакомство с новыми технологиями, которые планируется вводить в целях оптимизации производственного процесса, а также возможность сравнить старые способы производства с новыми, наглядно отслеживая изменения;
- проверка адекватности модели объекту в деталях и в целом;
- выявление трудноуловимых ошибок имитационного моделирования;
- «проигрывание» различных ситуаций для непрограммирующего пользователя.

При анимации технологических процессов недостаточно представить только визуализацию операций и оборудования. Современные средства позволяют связывать компьютерные имитационные модели процессов с их анимационной моделью, что даёт более широкие возможности для исследователя, проектировщика или заказчика. Такая «связка» позволяет не только понять, как работает система, но и проводить имитационные эксперименты на модели объекта, отслеживая его параметры, а также визуально оценивать динамику при различных условиях и входных воздействиях.

1 ЯЗЫК КОМПЬЮТЕРНОЙ АНИМАЦИИ PROOF ANIMATION

1.1 Общие сведения о языке компьютерной анимации Proof Animation

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

Применение компьютерной анимации даёт возможности:

- проверить адекватность модели объекту в деталях и в целом;
- выявить трудноуловимые ошибки имитационного моделирования;
- доказать заказчику правильность работы имитационной модели;
- проиграть различные ситуации для непрограммирующего пользователя;
- 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. Для процессов машиностроительных производств статическими элементами являются 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 путём 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 технологических процессов в машиностроении к динамическим элементам относятся заготовки, транспортные средства, исполнительные органы.

Анимационная оболочка Student Proof Animation обрабатывает пять типов входных файлов и шесть выходных. Для анимации простых технологических процессов достаточно использовать два типа входных и три типа выходных файлов (рис. 15).

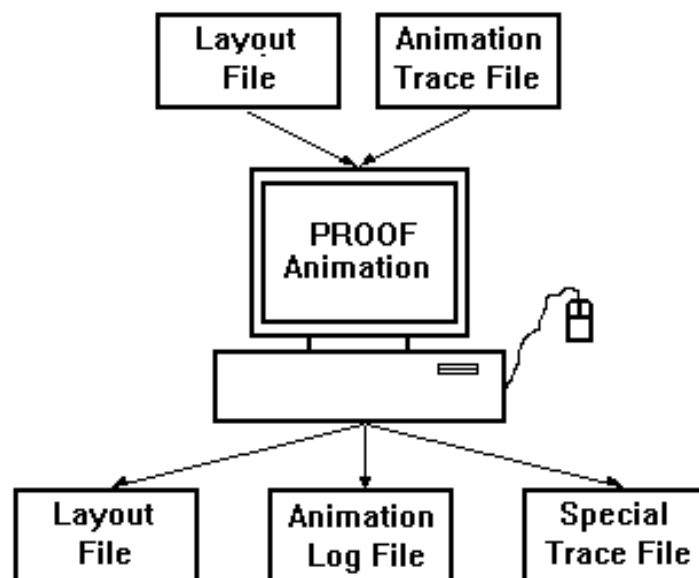


Рисунок 15 – Входные и выходные файлы, обрабатываемые Student Proof Animation

Входные файлы:

- Layout File – файл разметки. Содержит описание всех статических и динамических объектов: фоновый текст, таблицы, контуры участка цеха, накопители, технологическое оборудование, рабочих и т. д.
- Trace File – файл управления анимацией, или файл трассировки. Включает команды Student Proof Animation, посредством которых перемещаются динамические объекты.

Выходные файлы:

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

1.2 Построение статических и динамических объектов в Proof Animation

Для создания статических объектов в Proof Animation используют режим Draw Mode. Панель инструментов в Draw Mode имеет следующий вид (рис. 16).

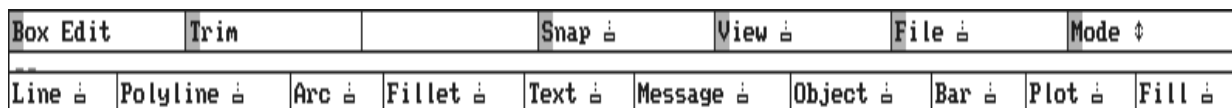


Рисунок 16 – Панель инструментов Draw Mode

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. 17). Построение линии производят указаниями её начальной и конечной точки щелчками мыши.

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

Рисунок 17 – Комплект инструментальных средств Line

Комплект инструментальных средств Line:

- Start X – x-значение начальной отметки Line;
- Start Y – y-значение начальной отметки 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 высвечи-

ваются 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 щёлкнуть мышью по одной из высвеченных конечных точек и затем переместить её в новое положение. Для изменения положения линии на экране указатель мыши устанавливают посередине линии и перетаскивают её в другое место.

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

Polyline – добавление ломаной линии

Опция Polyline даёт возможность вывести на экран соединённые 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 задаёт следующие параметры дуги (рис. 18).

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 18 – Комплект инструментальных средств Arc

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;
- Full Circle – 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° 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 цветом. Ограниченная область может быть задана любой комбинацией «Lines» и «Arcs». Область должна быть замкнута, иначе заполнится всё неограниченное пространство. Для изменения текущего цвета области выбирают необходимый цвет в «Color Palette» и указывают щелчком мыши область заполнения. Для отмены указанных действий используют «Undo» или «Unfill».

«**Box Edit**» – редактирование нескольких элементов

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

При использовании «Box Edit» (редактирование блока) появится меню комплекта инструментальных средств (рис. 19).

Box Edit	
/ Cut	
= Copy to Paste	
Names ▶ Automatic	
Delta X →	0.00
Delta Y →	0.00
Scale →	1.00
Rotation→	0.00
Vertical Flip	
Horizontal Flip	

Рисунок 19 – Комплект инструментальных средств Box Edit

Комплект инструментальных средств Box Edit:

- 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;
- Vertical Rip – инвертирует элементы «верхняя часть к нижней части».
- Horizontal Flip – инвертирует элементы «слева направо»;

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

1.3 Создание классов в Proof Animation

При создании объекта в файле анимации необходимо определить его Object Class (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» Object Class, 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.

Class Mode

Панель инструментов в Class Mode имеет следующий вид (рис. 20).

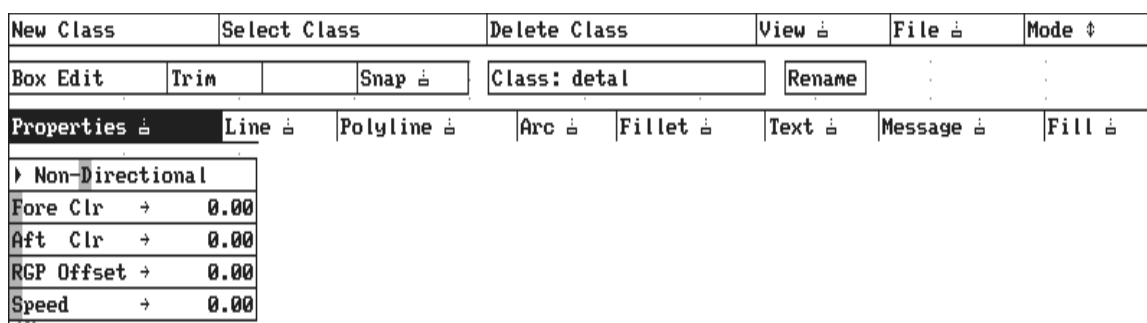


Рисунок 20 – Панель инструментов в «Class Mode»

New Class (Новый класс)

New Class используют для определения нового класса. После задания **New Class** Proof Animation выдаёт 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 **Save Layout** из меню **File**.

Select Class (Выбор класса)

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

Delete Class (Удаление класса)

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

1.4 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 создания рисунка, так и после.

Hot Point (Горячая отметка)

В **Class Mode Proof Animation** отображает, перемещает и вращает объекты, созданные из класса, относительно горячей отметки. Горячая отметка **Object Class** имеет координаты (0, 0). Для изменения координат горячей отметки используют **Box Edit** (редактирование блока).

Directionality and Orientation (Направление и ориентация)

Object Class может быть задан как направленный или как ненаправленный. Первая опция в меню **Properties** задаёт 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 **Box Edit**, которая позволяет вращать изображение 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 раз-

решения 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 определённой 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 «**Box Edit**», а затем устанавливают цвет из цветовой палитры.

1.5 Движение в Proof Animation

Большинство видов движения можно представить, как перемещение по заданному пути. Задание движения по определённому пути – одна из наиболее важных возможностей Proof Animation. Путь задаётся по линиям (Lines) и дугам (Arcs) и является невидимым элементом.

Path Mode (Режим пути)

Proof Animation обеспечивает специальный режим определения пути, который можно выбирать из меню **Mode** (режим). Пути показываются в цвете. Другие элементы отображаются чёрным цветом на сером фоне. В большинстве случаев пути показываются синим цветом. Путь, который необходимо исправить (Repairs), показывается белым цветом. У текущего выбранного пути цвет жёлтый, а у выбранного сегмента пути цвет красный. Меню **Path Mode** имеет вид, указанный на рисунке 21.

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

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

После выбора команды **New Path** (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 путём добавления сегментов.

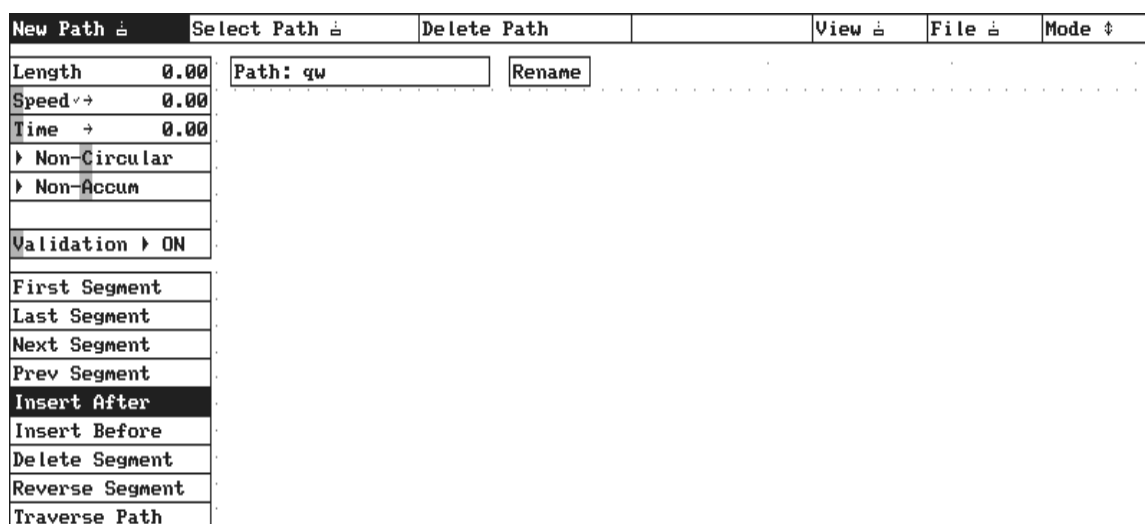


Рисунок 21 – Панель инструментов в **Path Mode**

Path Properties (Свойства пути)

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

Length (Длина)

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

Speed (Скорость)

Speed задаёт 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 определённой скорости и расчётной 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** задаётся 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 определённом значении времени и расчётной 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 **Path Mode** – **Non-Accum** (не накапливаемый). Накапливаемый путь отличается от не накапливаемого 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 удалён из накапливаемого пути, другие объекты будут двигаться вперед автоматически. Накапливаемый 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 соединён 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 выдаёт предупреждение **Leapfrog**, если объект удалён из накапливаемого пути, или предупреждение **Encroachment**, ес-

ли объекты на накопляемом пути сталкиваются друг с другом. Опцию **Validation** можно переключить на **Validation*ON** (включено) или **Validation*OFF** (выключено).

Path Segments (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 **Draw Mode** и **Run Mode**. Конец сегмента не маркирован.

1.6 Комплект инструментальных средств создания и редактирования сегментов

В нижнем меню комплекта инструментальных средств есть опции добавления и удаления сегментов. Для добавления сегментов к пути используют опцию **Insert After** (вставка после). Каждый новый сегмент начинается с конца предшествующего сегмента, соединённого с началом стрелки, и указывается щелчком мыши. При задании начала нового пути опция **Insert After** вызывается автоматически. Для исправления ошибок построения пути из сегментов используют клавиши **<Backspace>** (возврат на один символ) или **<Delete>** (удалить).

Insert Befor» (Вставка перед), **Insert After** (Вставка после)

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

Delete Segment (Удалить сегмент)

Delete Segment позволяет удалить выбранный сегмент. После того как сегмент удалён, 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>**.

Reverse Segment (Направить сегмент)

Reverse Segment позволяет задавать направление выбранного сегмента. **Reverse Segment** полезен в том случае, если сегмент, осо-

бенно первый сегмент пути, направлен неправильно и необходимо повторно выбрать сегмент. Для добавления сегментов после их изменения необходимо сначала нажать **Insert After**.

Traverse Path (Двигаться по пути)

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

1.7 Файл трассировки

Управление динамическими объектами в Proof Animation осуществляется посредством файла трассировки (Trace File). Файл трассировки содержит упорядоченную последовательность команд, в соответствии с которыми Proof Animation выполняет определённые действия в анимационной модели. Такой список команд может быть написан имитационной моделью (GPSS/Н-программой). Для написания файла управления вручную можно использовать текстовый редактор, создающий неформатируемые 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, определённое 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 **Object Class** устанавливаются некоторые простые правила. Верхние и строчные символы букв обрабатываются 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 сохранён 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:

- **Цвета фона:**
 - 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> (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 заданной скоростью. Достигнув заданных координат, он останется в этой точке на время, соответствующее модулю времени, определённого 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 со скоростью 10.

Рассмотрим еще один пример. Допустим, что объект робокар «rob» создан. Тогда в результате выполнения подпрограммы:

TIME 50

MOVE rob 10 40 50

TIME 60

по достижении текущего времени TIME 50 объект «rob» начнет перемещаться со скоростью 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 быстродействие 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°.

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° против часовой стрелки (рис. 22).

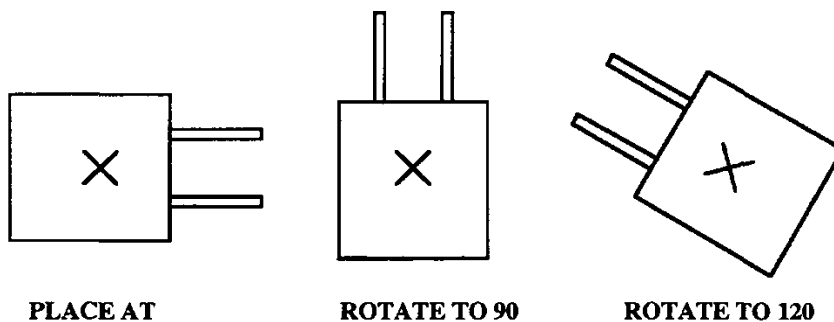


Рисунок 22 – Пример абсолютного вращения

ROTATE 1 120

... мгновенно вращает объект 1 на 120° против часовой стрелки относительно положения, в которое его поместили командой **PLACE ... ON**.

Пример относительного вращения:

ROTATE 1 90

... мгновенно вращает объект 1 на угол 90° против часовой стрелки (рис. 23).

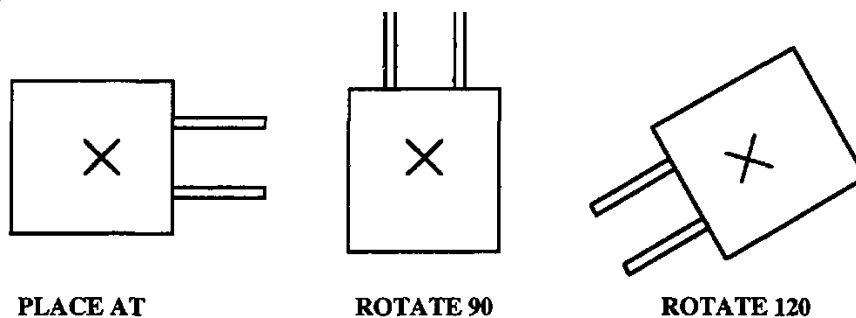


Рисунок 23 – Пример относительного вращения

ROTATE 1 120

... мгновенно вращает объект 1 на 120° 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° (четыре единицы времени Proof Animation). Другая команда **ROTATE** с определённым 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°*) за 10 единиц времени Proof Animation. Вращение заканчивается, когда объект повернется на 90° против часовой стрелки относительно текущей ориентации.

ROTATE 1 90 TIME 10 STEP 5

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

– **Изменение содержания сообщения**

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

Синтаксис команды **WRITE**:

WRITE <имя сообщения> <текст сообщения>

Например, по команде:

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

1.8 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 её 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 течение 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 путём загрузки файла `sra.exe`. Выберем в верхнем меню пункт **Mode**, а в нём режим **Draw Mode**. С помощью набора команд Student Proof Animation нарисуем статические элементы технологии: станки, траектории движения деталей (в невидимом цвете **Backdrop**) и необходимый текст (рис. 24).

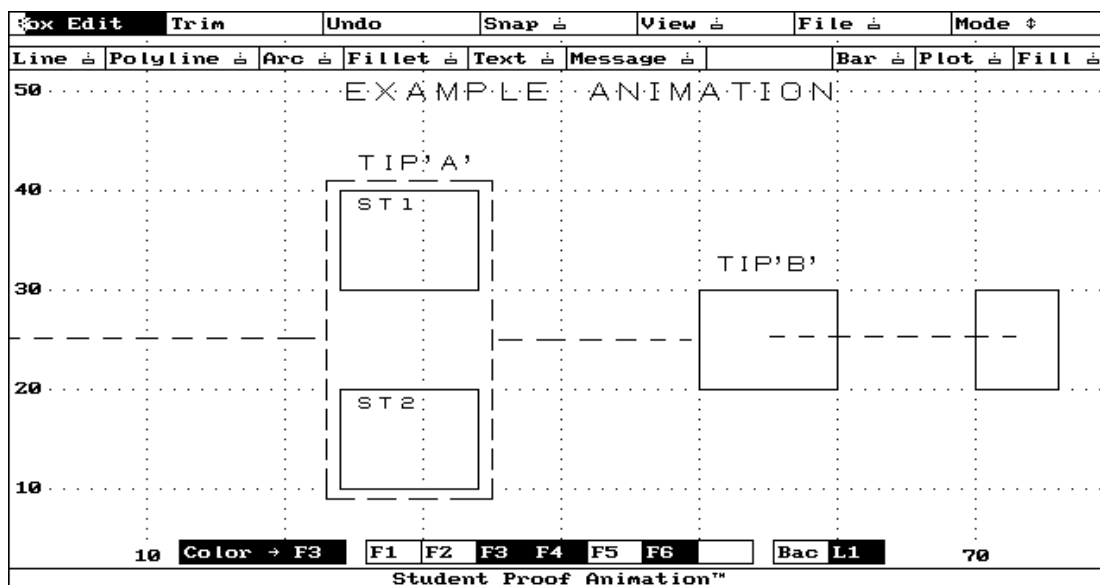


Рисунок 24 – Статические элементы технологии
(Пунктиром показаны траектории движения динамических элементов, в режиме Draw они невидимы)

В режиме **Class Mode** выберем опцию **New class** и нарисуем динамический объект (класс) – деталь (под именем det) (рис. 25).

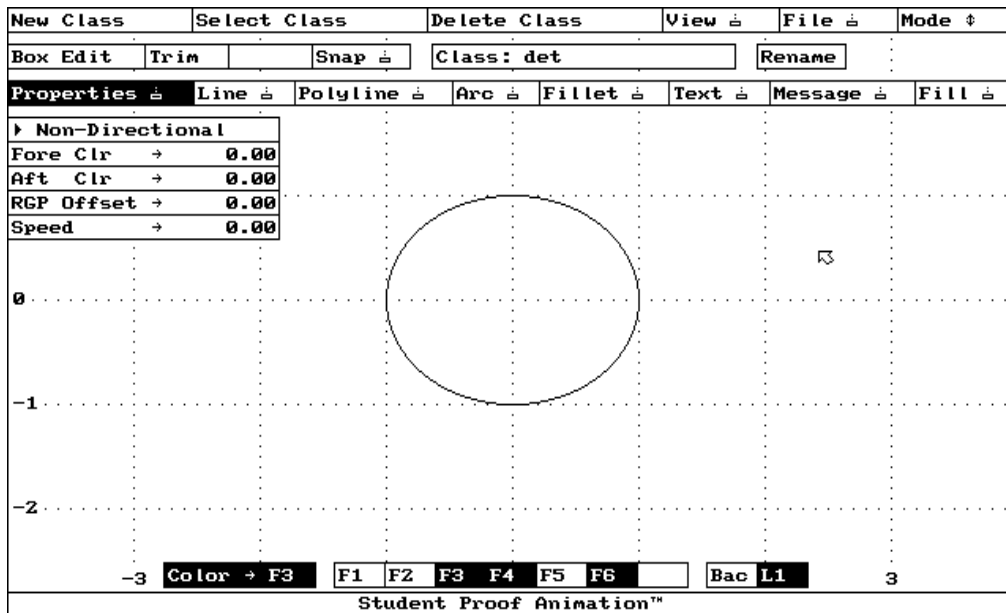


Рисунок 25 – Динамический объект технологии

В режиме **Path Mode** выберем опцию **New Path**, 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. 26).

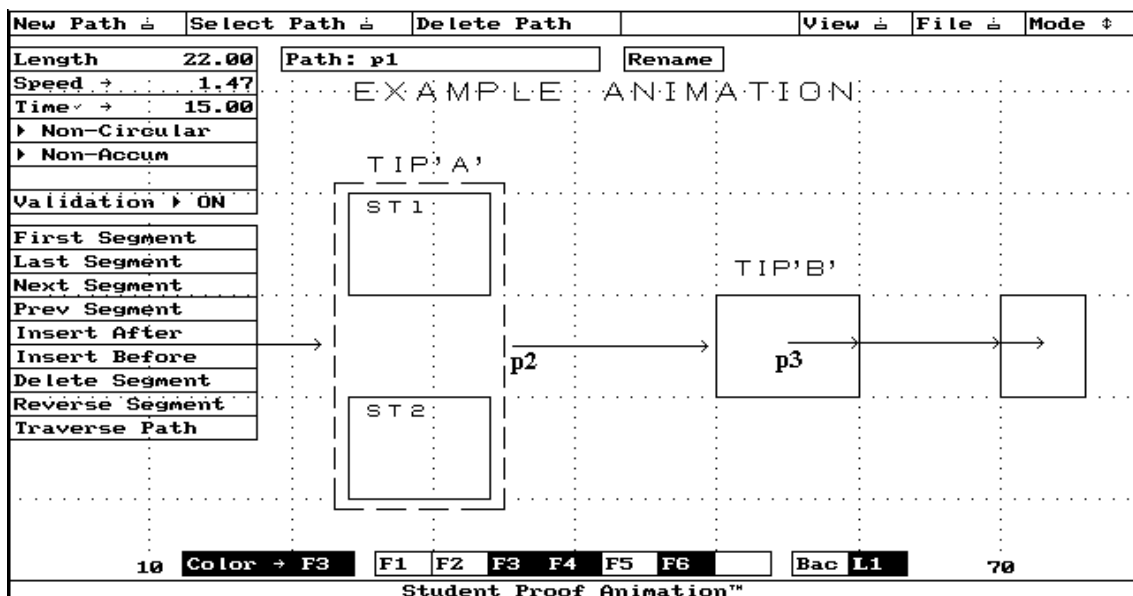


Рисунок 26 – Пути движения динамических объектов

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 ON p3
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 путём входа с помощью мыши в пункт **File** верхнего горизонтального меню, подпункт **Open Layout&Trace**. В появившемся вертикальном меню выберем пункт primera путём нажатия на нём клавиши мыши. В режиме **RUN** командой **Go** запустим программу на выполнение.

На рисунке 27 представлен фрагмент анимации технологии изготовления деталей.

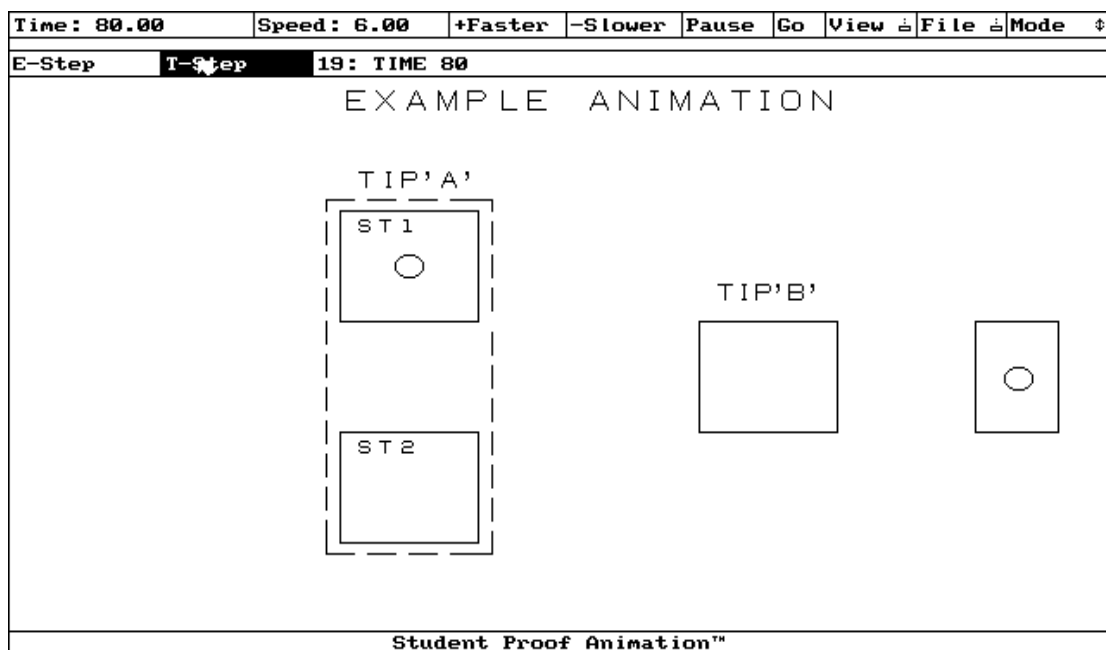


Рисунок 27 – Фрагмент анимации технологии изготовления деталей

КОНТРОЛЬНЫЕ ВОПРОСЫ К РАЗДЕЛУ 1

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?

2 СВЯЗЬ АНИМАЦИИ С ИМИТАЦИОННОЙ МОДЕЛЬЮ

2.1 Генерирование файла трассировки (.atf) имитационной моделью

Управляющим оператором и блоком, создающим линии файла трассировки, являются оператор PUTPIC и блок BPUTPIC.

Формат:

PUTPIC **opt,..., (list)**

BPUTPIC **opt,..., (list)**

где opt – опция, а list – список чисел, числовых выражений, переменных, стандартных числовых атрибутов, которые GPSS/H записывает во внешний файл (файл.atf).

Опции, связанные с оператором PUTPIC и блоком BPUTPIC:

FILE=log

LINES=unt

где log – логическое имя внешнего файла, в который будут записываться данные (по умолчанию данные будут выводиться на экран), а unt – число строк, отображаемых после блока BPUTPIC.

Для создания GPSS/H-модели файла управления анимацией, сначала его необходимо связать с логическим именем, которое будет использоваться в GPSS/H-программе. Для такой связи используется специальный оператор FILEDEF.

Формат оператора FILEDEF:

LOG **FILEDEF 'NAME'**

где NAME – имя .atf-файла, а LOG – логическое имя файла.

Например,

...

ATF FILEDEF «TIME.ATF»

...

 BPUTPIC FILE=ATF,LINES=3,AC1

TIME *.*

CREATE rob rob

PLACE rob at 12 18

В приведенном примере файл «TIME.ATF» будет автоматически создан в текущей директории, связан с логическим именем ATF и в него будут записаны три строки, расположенные ниже блока BPUTPIC. При этом вместо звездочек (*.**) запишется значение те-

кущего времени моделирования (значение стандартного числового атрибута AC1).

Для записи в файл управления команды END используется управляющий оператор PUTPIC:

```
PUTPIC      FILE=ATF  
END
```

Такая запись обязательна, она используется для завершения анимации и обычно записывается после оператора START.

2.2 Переменные в GPSS/H-моделях

При связи GPSS/H-модели с внешними файлами используют переменные. Переменные позволяют читать из внешнего файла входные параметры и записывать во внешний файл (.atf) результаты моделирования.

Для определения переменных в GPSS/H-моделях используются утверждения INTEGER (целая) и REAL (действительная).

Формат INTEGER и REAL:

```
INTEGER    &v,...  
REAL      &v,...
```

где v – имя переменной.

При объявлении переменных их начальные значения равны 0. Для присвоения численных значений, отличных от нуля, используются оператор LET и блок BLET.

Формат блока BLET и оператора LET:

```
LET    &v=var  
BLET  &v=var
```

где v – имя переменной, а var – числовое значение или математическое выражение.

Предположим, что переменные &TIME, &V и &S определены в модели как действительные

```
REAL    &TIME,&V,&S
```

и обозначают соответственно время движения транспортного средства от склада до рабочего места, скорость движения и расстояние доставки соответственно. Тогда для присвоения переменной &TIME числового значения можно использовать строку

```
LET    &TIME=&V/&S
```

Примечание

Для получения нецелого результата, например в выражении $3/2$, необходимо значения числителя и знаменателя записывать через десятичную точку, т. е. $3.0/2.0$. В противном случае из результата выражения будет отброшена дробная часть.

2.3 Чтение данных из внешнего файла

Оператор GETLIST – читает данные из внешнего файла или с клавиатуры (данные во внешнем файле должны быть в свободном формате, с входными значениями, отделяемыми одним или более пробелами, или написанными на разных строках).

Формат оператора GETLIST:

GETLIST FILE=(логическое имя файла),(list),...

где FILE – опция, определяющая логическое имя файла, из которого должны читаться входные данные, а list – список переменных, которым будут присвоены числовые значения.

Например, в GPSS/H-модели значение переменной &I, обозначающей количество станков в цехе, должно читаться из внешнего файла «IN». Этого можно добиться, записав в модели строку:

GETLIST FILE=IN,(&I)

2.4 Этапы создания модели компьютерной имитации и анимации

Создание модели компьютерной имитации и анимации сводится к четырём этапам.

Этап 1. Представление заданного технологического процесса в виде системы массового обслуживания (СМО).

На этом этапе реализуется переход от словесного описания технологического процесса к его математической модели. Здесь требуется описать заданный объект моделирования в абстрактных терминах и понятиях с использованием теории систем массового обслуживания.

Для этого необходимо:

– определить потоки событий (входящие потоки заявок и потоки обслуживаний для каждой очереди и прибора обслуживания);

- определить структуру системы массового обслуживания (число фаз, число каналов обслуживания, число очередей для каждой из фаз обслуживания заявок и связи источников заявок, приборов и очередей);

- определить алгоритмы функционирования системы массового обслуживания (дисциплины ожидания заявок в очередях и выбора на обслуживание каналов, правила ухода заявок из очередей и приборов).

Этап 2. Разработка в соответствии с СМО имитационной модели на специализированном языке GPSS/H.

Здесь математическая модель, сформированная на первом этапе, воплощается в конкретную машинную модель, ориентированную на использование специализированного языка компьютерной имитации GPSS/H. Вначале требуется построить модель по блочно-му принципу, т. е. в виде совокупности стандартных блоков языка GPSS/H. Для этого необходимо:

- определить, какие объекты технологии будут отображаться транзактами в модели;

- определить количество сегментов, из которых будет состоять модель;

- подобрать блоки (цепь блоков), которые будут отображать события в заданной технологии.

После построения блок-схемы необходимо перейти к программированию модели. Переход от блок-схемы к программе является формальным шагом, так как заключается в записи пространственной структуры в линейном виде с добавлением необходимых управляющих операторов, что не требует специальных навыков.

На данном этапе также рекомендуется создать файл-меню, куда будут вводиться исходные данные. Файл-меню создаётся специально для данного процесса в виде неформатируемого текстового файла типа ASCII.

Этап 3. Отображение динамики технологического процесса при помощи языка компьютерной анимации Proof Animation.

На этом этапе требуется отобразить динамику заданного технологического процесса на компьютере при помощи специализированного языка компьютерной анимации Proof Animation. Для этого при помощи опций режима Draw Mode необходимо нарисовать статические элементы анимации (контуры цеха, станки, траектории

движения транспортных средств и т. п.), а в режиме Class Mode динамические объекты (детали, транспортные средства, рабочих и т. п.).

Для связи анимации с имитационной моделью необходимо добавить в те места GPSS/H-модели, где требуется динамическое отображение операций, специальные команды управления Proof Animation.

На этом этапе также требуется вывести результаты моделирования на анимацию технологического процесса.

Этап 4. Оценка характеристик технологического процесса на разработанной модели компьютерной имитации и анимации.

На этом этапе компьютер используется для проведения имитационных экспериментов на составленной программе. Результаты этих экспериментов надо использовать для анализа и формулирования выводов о характеристиках заданного технологического процесса. Необходимо решить вопрос о форме представления результатов моделирования (графики, диаграммы, гистограммы, схемы и т. п.). В каждом конкретном случае целесообразно выбрать наиболее подходящую форму представления результатов моделирования. В большинстве случаев результаты удобнее сводить в таблицы, хотя графики позволяют более наглядно иллюстрировать полученные результаты.

При проведении имитационных экспериментов сначала надо вызвать файл-меню и ввести в него исходные данные. Затем запустить составленную программу GPSS/H. После прогона модели сформируется файл управления анимацией «.atf»-файл. При запуске анимации результаты моделирования должны быть выведены на экран в виде движения оборудования, рабочих, надписей и количественных показателей процесса.

2.5 Пример создания модели компьютерной имитации и анимации

Для создания имитационной и анимационной модели выбран технологический процесс изготовления вала-червяка (рис. 28). Червяк входит в состав червячного редуктора, который в свою очередь входит в кран манипулятор КМ-41. Червяк служит для вращения поворотной части крана. Червячная передача предназначена для

уменьшения количества поворотов вала на выходе редуктора по сравнению с числом оборотов червяка.

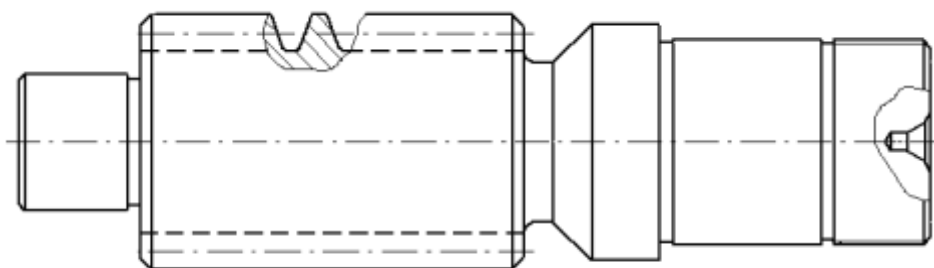


Рисунок 28 – Вал-червяк

Производство характеризуется, как крупносерийное: объём выпуска – 200000 штук и масса детали – 2,4 кг.

Исходя из того, что математические процедуры сложны и трудоёмки; желательно осуществить наблюдение за ходом процесса в течение определённого периода; может понадобиться сжатие временной шкалы, наиболее целесообразным является применение имитационного моделирования.

Описание работы участка, обеспечивающего процесс изготовления вала-червяка.

Сначала заготовка, представляющая собой поковку, горячештампованную в закрытых штампах, поступает со склада заготовок на фрезерно-центровальный станок 1 (рис. 29), на котором подрезаются и центрируются торцы 5 и 10 (рис. 30). Затем на токарно-винторезных станках происходит формирование поверхностей: на 2-м рабочем месте обтачиваются поверхности 11, 12, 13, 14, 15, 16, 17, 18 и нарезается резьба на поверхности 16; на 3-м рабочем месте – поверхности 1, 2, 3, 4, 5, 6, 7, 9; на 4-м рабочем месте нарезается червяк по поверхности 7. Далее на 8-м рабочем месте – контрольно-измерительной машине КИМ -141В – производится контроль полученных поверхностей. На 5-ом рабочем месте фрезеруется сбег витка червяка по поверхностям 18 и 19. После этого червяк шлифуется по внешнему диаметру и поверхностям 3 и 18 на 6-м рабочем месте и по среднему диаметру – на 7-м. В итоге готовый червяк поступает на контроль – на 8-е рабочее место. Если червяк не проходит контроль, то он бракуется. Остальные перемещаются на склад готовой продукции для дальнейшей сборки в составе червячного редуктора.

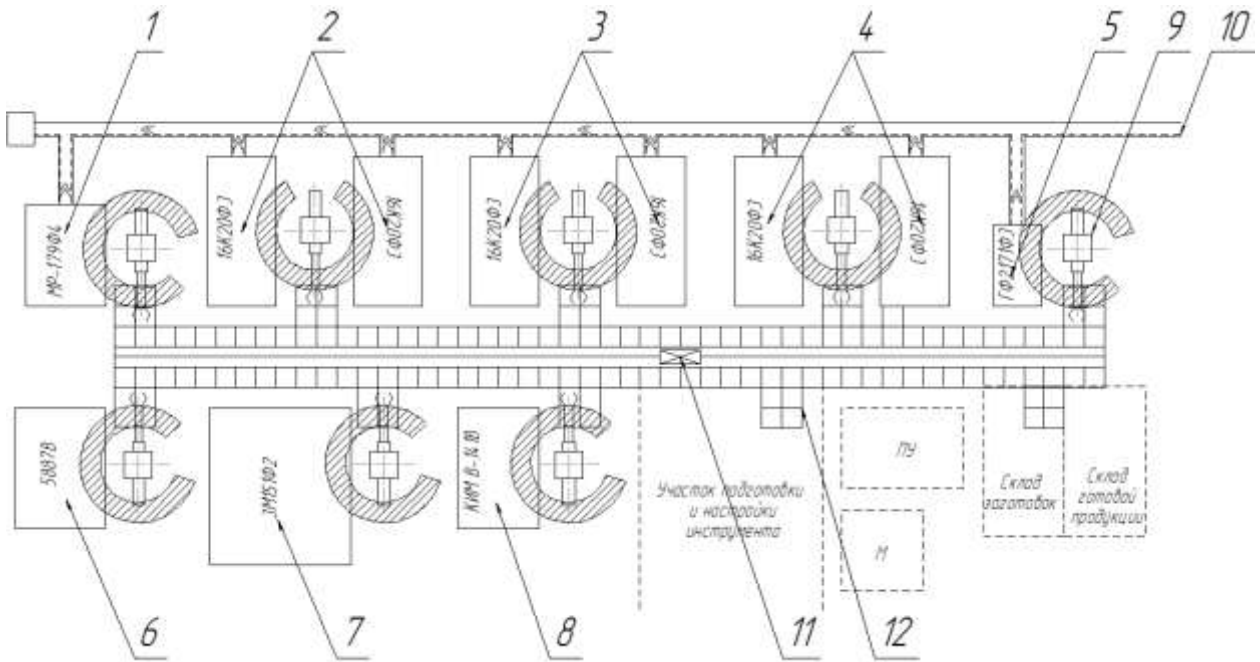


Рисунок 29 – Планировка автоматизированного участка изготовления деталей типа вал-червяк

- На планировке числами обозначены:
- 9 – промышленный робот М20П40.01;
 - 10 – конвейер для удаления стружки;
 - 11 – кран-штабелёр СА-ТСС-0,25;
 - 12 – рольганг КР-0,25.

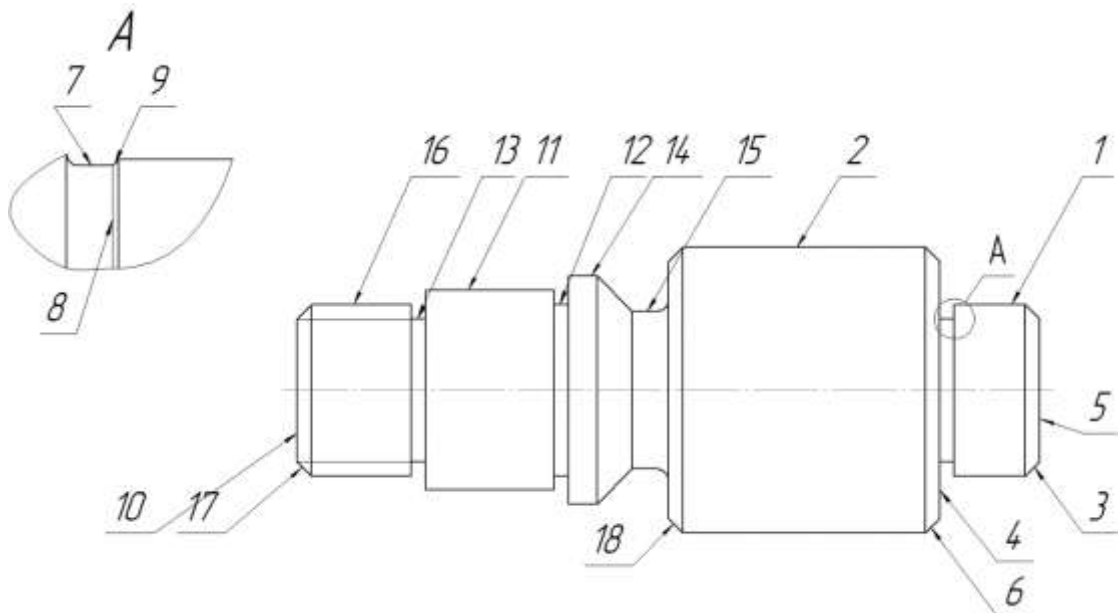


Рисунок 30 – Обозначение поверхностей червяка

Этап 1.

Представим производство червяка в виде системы массового обслуживания. В производстве заготовка червяка движется от одного станка к другому, на них последовательно обрабатываются поверхности. Каждую такую заготовку можно представить как заявку на перемещение к конечной точке. Конечная точка – это готовое изделие. Подробнее о разработке моделей систем с использованием теории массового обслуживания можно узнать из учебного пособия [1].

Производство можно представить, как многофазную многоканальную СМО (рис. 31).

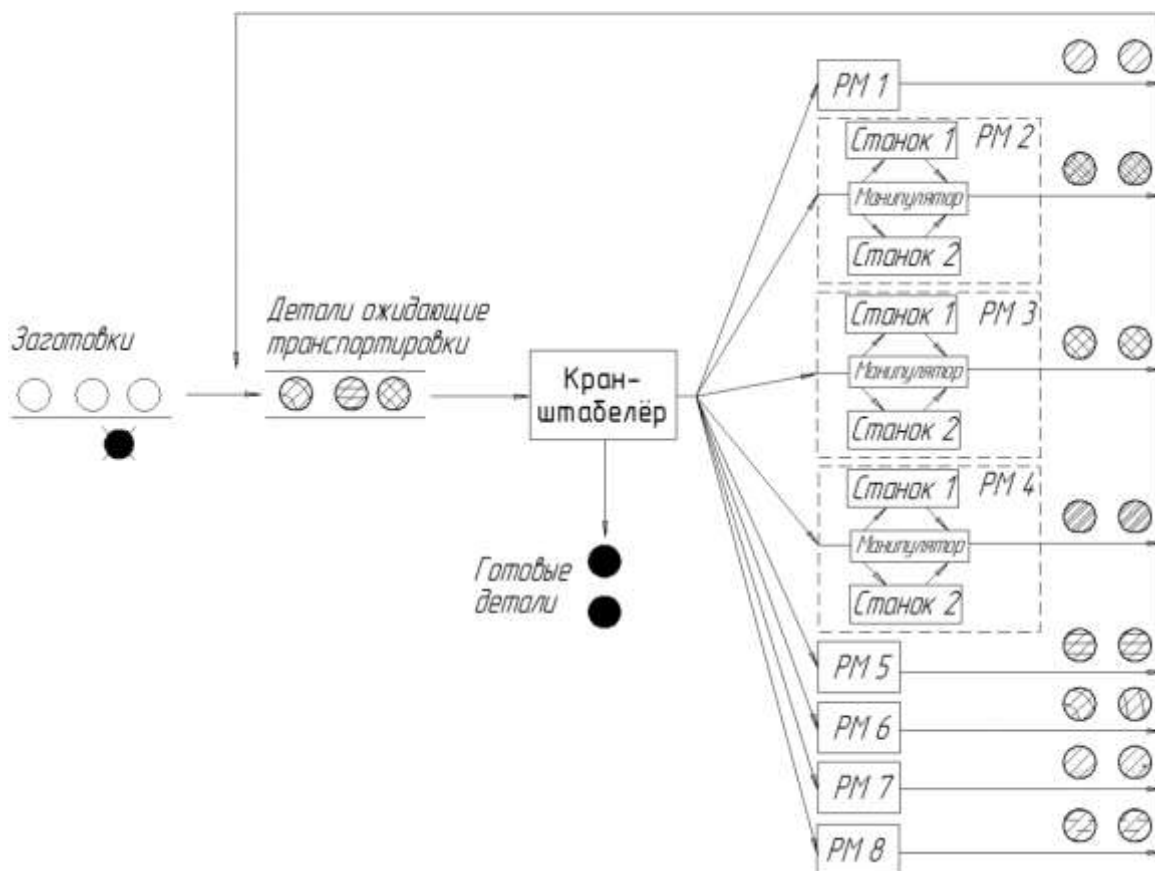


Рисунок 31 – Производство червяка в виде сети СМО

Разработанная модель производства червяка в виде сети однофазных одноканальных, однофазных многоканальных и многофазных одноканальных СМО является разомкнутой с ограниченным входным потоком заявок. Этот поток формируется в соответствии с производственным планом.

Этап 2.

Переходя от системы массового обслуживания к языку GPSS/H, определим, что заявки в СМО соответствуют транзактам в имитационной модели. Для их ввода в модель используем блок GENERATE.

Приборами в модели являются станки, выполняющие обработку, манипуляторы и кран-штабелёр. Их можно описать цепью блоков SEIZE-RELEASE или ENTER-LEAVE.

Для моделирования продолжительности операций используем блоки ADVANCE с нормальным распределением временных интервалов.

Блок TERMINATE применен для удаления транзактов из модели по окончании моделирования.

После того, как технологические операции описаны в виде операторов языка GPSS/H, их объединяют в блок-схему.

На рисунке 32 показана блок-схема имитационной модели технологии производства червяка на языке GPSS/H.

Подробнее о разработке имитационных моделей систем можно узнать из учебного пособия [1].

На основе блок-схемы построим имитационную модель автоматизированной производственной системы (АПС). Фрагмент GPSS/H-модели, моделирующей появление заготовок и перемещение их краном-штабелёром на 1-е рабочее место, имеет вид:

SIMULATE	начало моделирования
GENE ,,200000,,1PB	создание требований, соответствующих заготовкам червяка
SEIZE MESTO	
SEIZE SHTABEL	занятие крана-штабелёра
ADVA RVNORM(1,0.22,0.04)	перемещение крана-штабелёра с деталью
RELE SHTABEL	освобождение крана-штабелёра

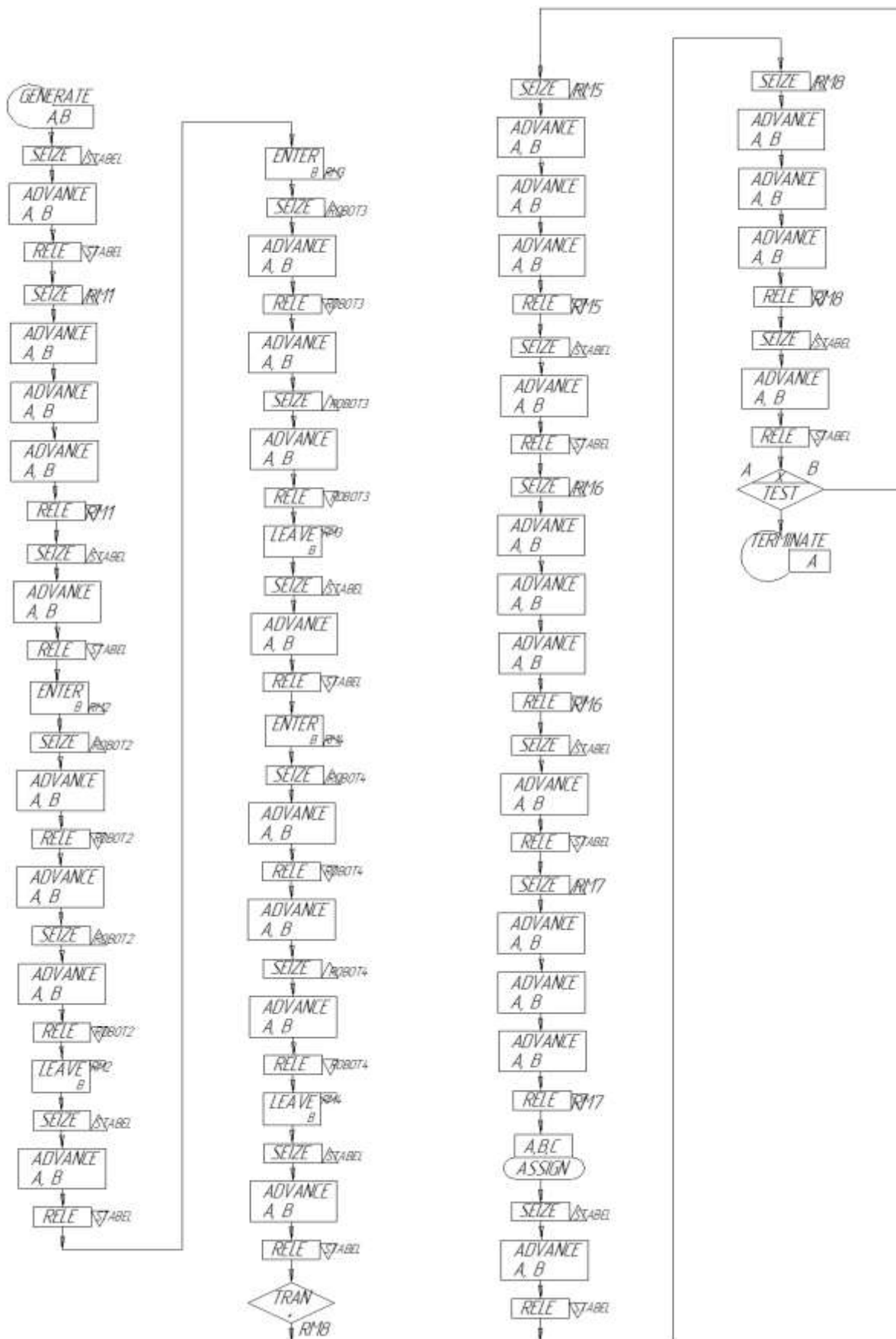


Рисунок 32 – Блок-схема GPSS/H-модели производства червяка

Этап 3.

Для построения анимационной модели войдем в Student Proof Animation путём загрузки файла sra.exe. Выберем в верхнем меню пункт Mode, а в нём режим Draw Mode. С помощью набора команд Student Proof Animation нарисуем статические элементы (рис. 33):

- станки;
- основания манипуляторов;
- склады заготовок и готовой продукции;
- склад для крана-штабелёра;
- рольганги;
- необходимый текст.

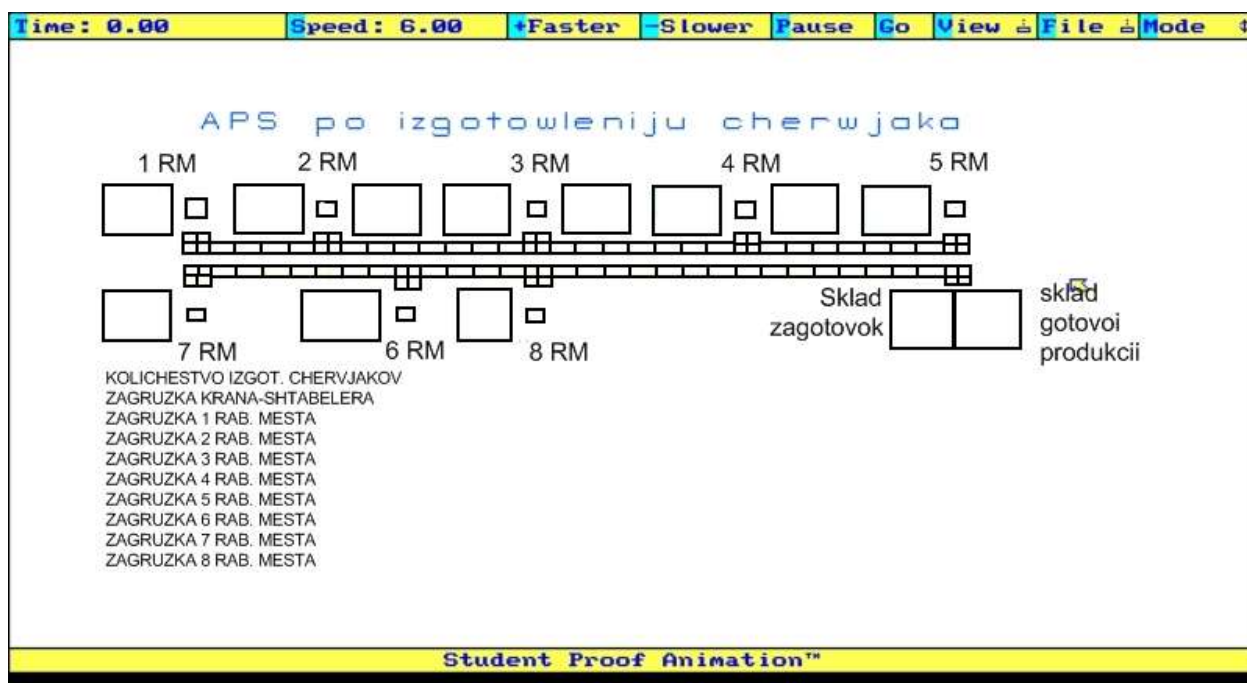


Рисунок 33 – Статические элементы

В режиме Class Mode выберем опцию «New class» и нарисуем динамические объекты – заготовку (под именем SAG), манипулятор (MAN) и кран-штабелёр (STABEL) (рис. 34-36).

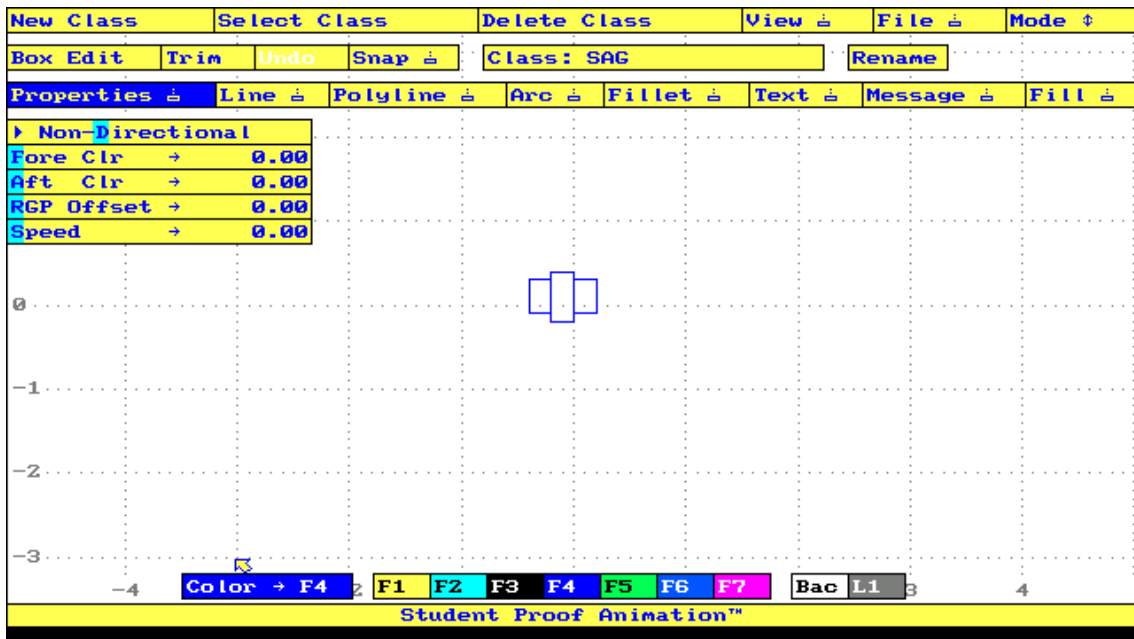


Рисунок 34 – Динамический объект – заготовка

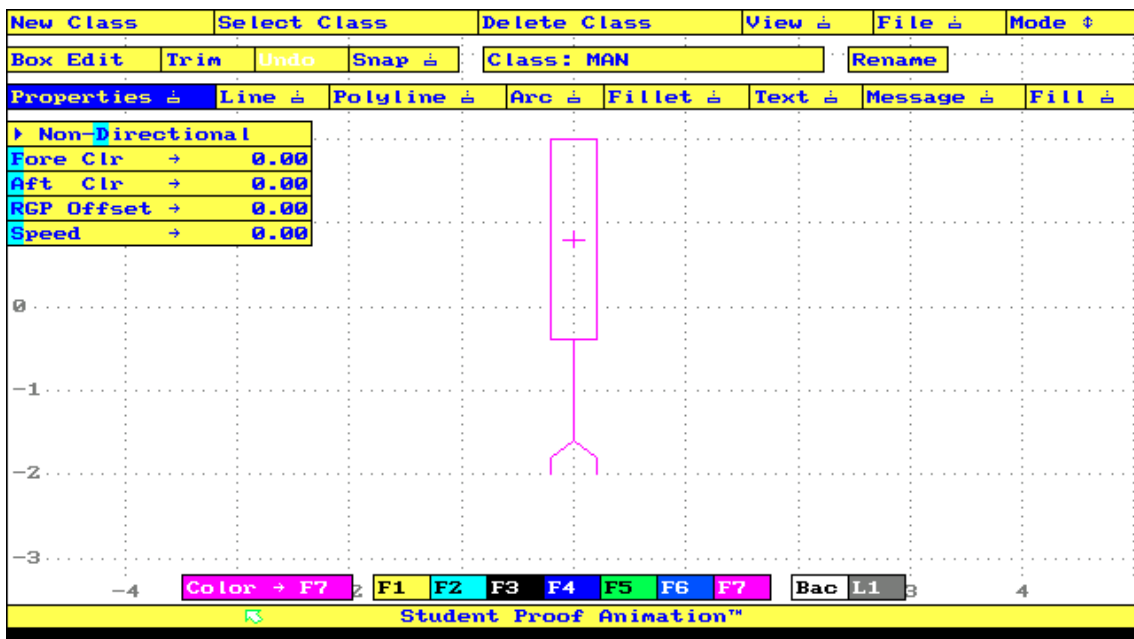


Рисунок 35 – Динамический объект – манипулятор

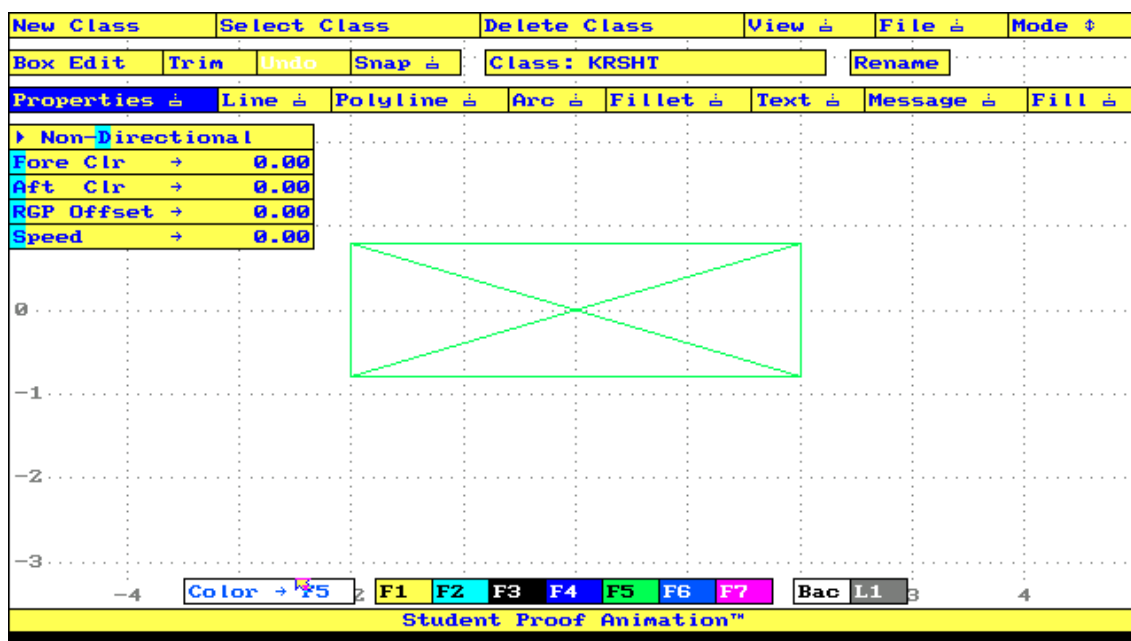


Рисунок 36 – Динамический объект – кран-штабелёр

Сохраним файл разметки как MODAPS.lay и выйдем из Proof Animation.

Свяжем имитационную и анимационную модели технологического процесса изготовления деталей.

Изменим GPSS/Н-модель в соответствии с заданными требованиями:

- для ввода входных параметров модели из внешнего файла используем GPSS/Н-переменные (с утверждениями INTEGER и REAL) и оператор GETLIST;
- для формирования файла управления анимацией воспользуемся блоком WPUTPIC совместно с управляющим оператором FILEDEF;
- для вывода результатов моделирования на монитор компьютера используем стандартные числовые атрибуты;
- в места, где требуется анимация операций, введём команды файла управления Student Proof Animation.

После ввода изменений сохраним файл как MODAPS.gps. Измененный фрагмент GPSS/Н-модели с командами управления анимацией, моделирующий появление заготовок и перемещение их краном-штабелёром на 1-е рабочее место, представлен ниже:

SIMULATE		начало моделирования
DANNIE	FILEDEF 'DANNIE.txt'	
RES	FILEDEF 'RES.TXT'	
REAL	&TOBR1,&TOBR2,&TOBR3,&TOBR4,&TOBR5,&TOBR6,&TOBR7	
REAL	&TPOW1,&TPOW2,&TPOW3,&TPOW4,&TPOW5,&TPOW6,&TPOW7	
REAL	&TEK,&V,&TCIKL,&X1,&X2,&X3,&X4,&X5,&X6,&X7,&X8,&XSZ,&A,&B	
REAL	&XST1,&XST2,&XST3,&XST4,&XST5,&XST6,&XST7,&XST8	
INTEGER	&NST1,&NST2,&NST3,&NST4,&NST5,&NST6,&NST7,&ST8	
INTEGER	&ST1,&ST2,&ST3,&ST4,&ST5,&ST6,&ST7	
GETLIST	FILE=DANNIE,(&ST1)	
GETLIST	FILE=DANNIE,(&XST1)	
GETLIST	FILE=DANNIE,(&ST2)	
GETLIST	FILE=DANNIE,(&XST2)	
GENE	,,,200000,,1PB	создание требований, соответствующих заготовкам червяка
BPUTPICFILE=ATF,LINES=4,AC1,XID1,XID1		
TIME *.*		
CREATE SAG *		создать заготовку
PLACE * AT 85 29		поместить её в точку с координатами 85,29
SEIZE MESTO		
SEIZE SHTABEL		занятие крана-штабелёра
BPUTPICFILE=ATF,LINES=2,AC1		
TIME *.*		
PLACE R AT 85 31		поместить кран-штабелёр в точку с координатами 85,31
BPUTPICFILE=ATF,LINES=2,AC1		
TIME *.*		
MOVE R 0.22 27 31		переместить кран-штабелёр в точку 27,31 за 0,22 мин.
ADVA RVNORM(1,0.22,0.04)		перемещение крана-штабелёра с деталью
RELE SHTABEL		освобождение крана-штабелёра

Запустим интерпретатор GPSS/H. В строке запроса имени файла указываем имя *.gps-файла нашей модели (MODAPS.gps) и нажмем клавишу <Enter>. В результате сформируется файл управ-

ления анимацией (файл MODAPS.atf) с результатами моделирования.

Фрагмент этого файла приведен ниже:

```
TIME 0.0
CREATE SAG 1
PLACE 1 AT 85 29
MOVE R 3 85 31
TIME 13.1
MOVE R 3 27 31
TIME 13.1
PLACE 1 AT 24 36.6
ROTATE M1 TO -90
TIME 42.4
DESTROY 1
ROTATE M1 TO 0
TIME 42.4
MOVE R 3 27 31
TIME 48.6
MOVE R 3 37 31
TIME 48.6
CREATE SAG 1
PLACE 1 AT 35 36.6
ROTATE M2 TO -90
TIME 48.6
CREATE SAG 2
PLACE 2 AT 85 29
MOVE R 3 85 31
TIME 67.3
MOVE R 3 27 31
...
END
```

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

Запустим анимацию, выбрав файл sra.exe и нажав клавишу <Enter>, при этом файл управления графическими объектами и сам файл графических объектов должны быть в одной траектории. Выведем на экран анимационную модель технологии путём входа с помощью мыши в пункт **File** верхнего горизонтального меню, подпункт **Open Layout&Trace**. В появившемся вертикальном меню выберем пункт MODAPS путём нажатия на нём клавиши мыши. В режиме **RUN** командой **Go** запустим программу на выполнение.

На рисунке 37 представлен фрагмент анимации изготовления червяка.

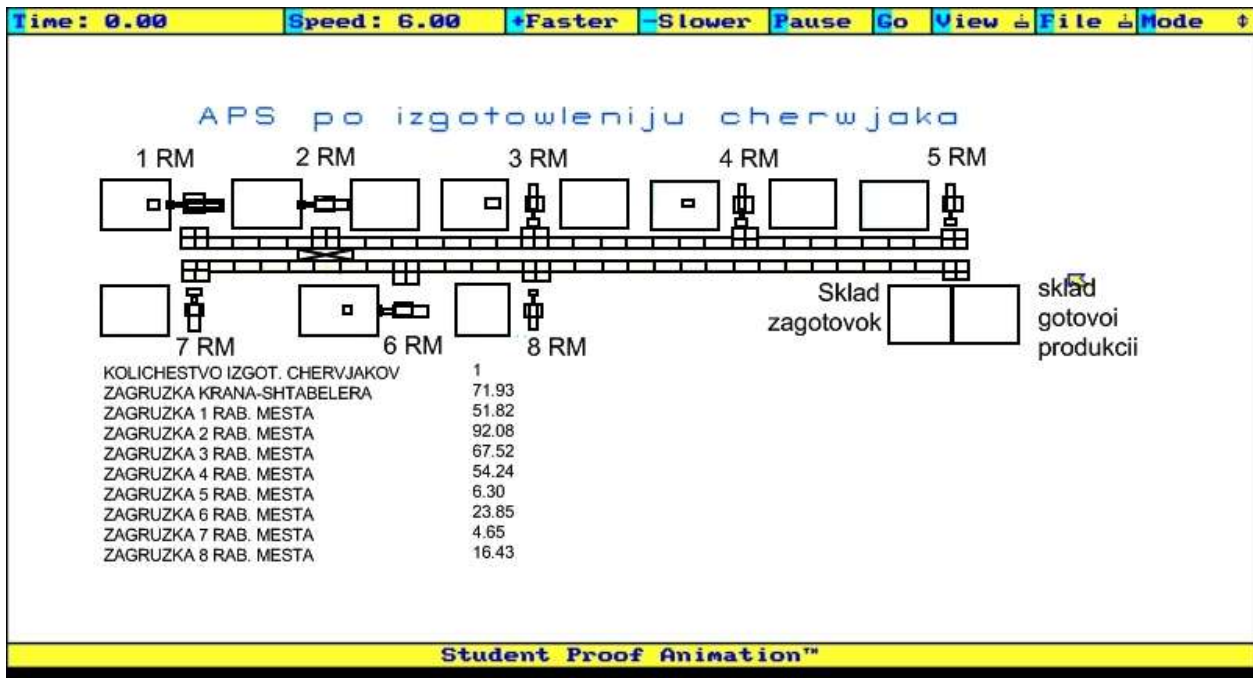


Рисунок 37 – Фрагмент анимации изготовления червяка

На анимационной модели движущимися объектами являются:

- 1) заготовка червяка;
- 2) манипуляторы;
- 3) кран-штабелёр.

Они появляются с началом моделирования. Кран-штабелёр перемещает заготовку к очередному рабочему месту, перекладывает её на рольганг. Далее манипулятор устанавливает её на станок, а после обработки снимает с него и кладёт обратно на рольганг. Затем кран-штабелёр перемещает заготовку к следующему рабочему месту. После обработки заготовки складываются на складе готовой продукции, а количество сделанных червяков увеличивается на единицу.

Этап 4.

При проведении имитационных экспериментов в модель вводятся начальные параметры (число рабочих мест, мест контроля, число станков на рабочем месте, время имитации, среднее время операций обработки и контроля, время перемещений крана-штабелёра между рабочими местами, время на установку манипуляторами заготовок на станок) (рис. 38).

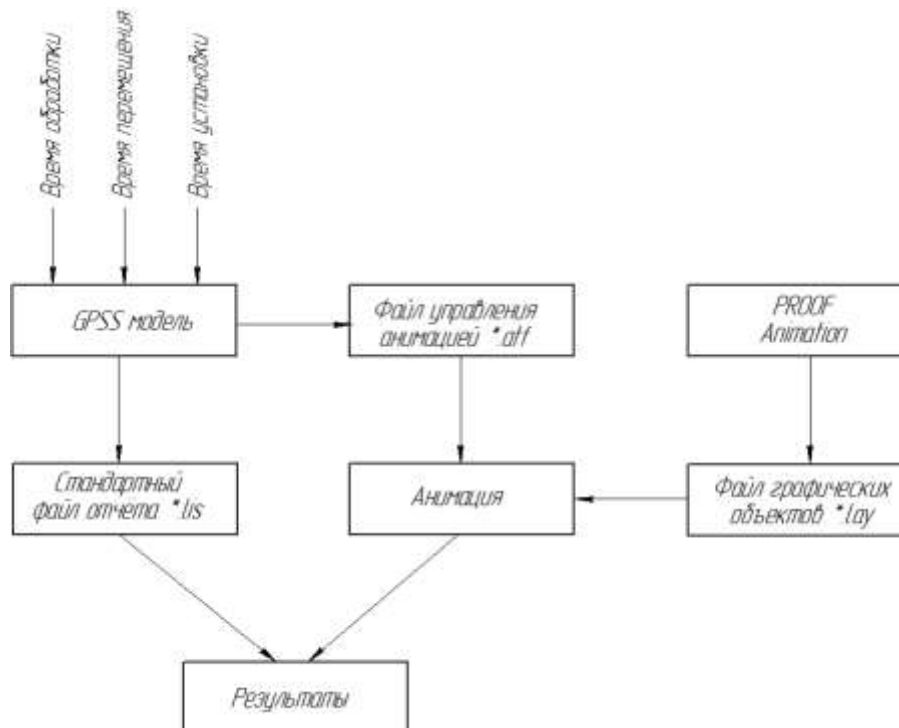


Рисунок 38 – Схема проведения имитационных экспериментов

После запуска интерпретатора GPSS/H получаем анимацию с результатами моделирования (загрузка рабочих мест, крана-штабелёра, манипуляторов, число готовых червяков).

Этапы работы с анимационной моделью:

- запустить файл «Sra.EXE»;
- загрузить файл анимации «MODAPS» с помощью команды **Open Layout&Trace** пункта меню **File** (рис. 39);

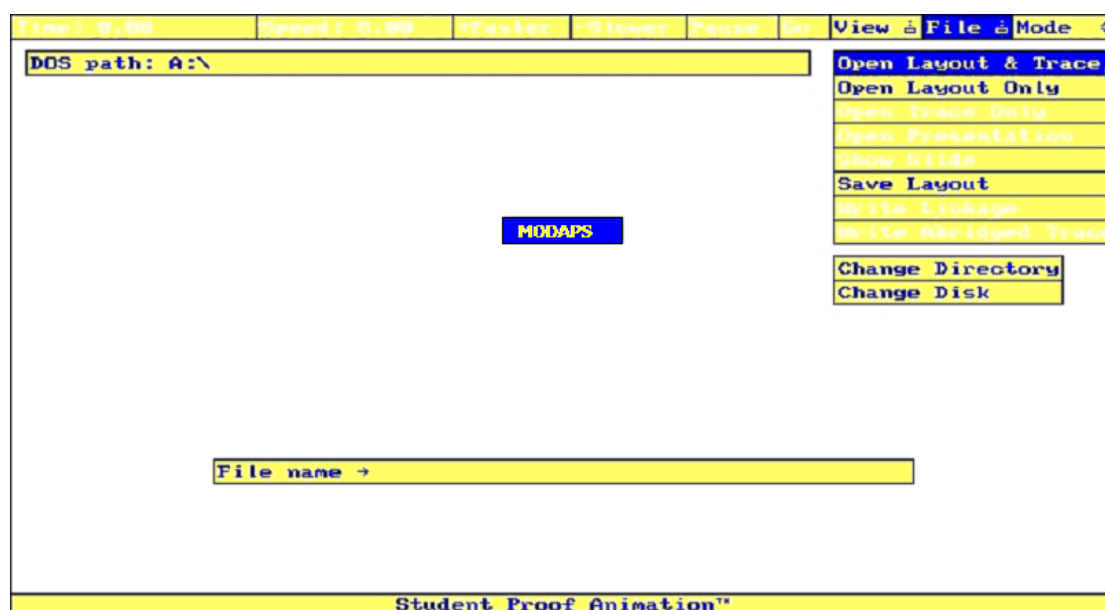


Рисунок 39 – Загрузка файла в SPA

- нажать команду **GO** и посмотреть анимацию (рис. 40). Команда **GO** предназначена для пуска или продолжения выполнения анимации после паузы.

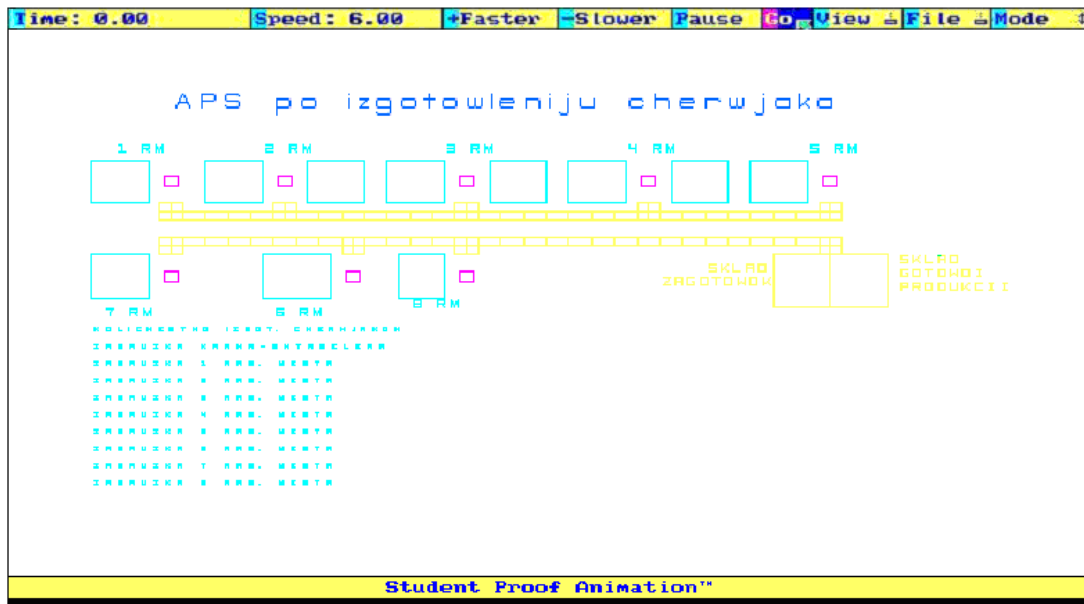


Рисунок 40 – Команда пуска анимации **GO**

В случае необходимости увеличения изображения можно воспользоваться командой **Zoom Box** (рис. 41). Для этого в верхнем меню выделить опцию **View**, а затем нажать команду **Zoom Box**.

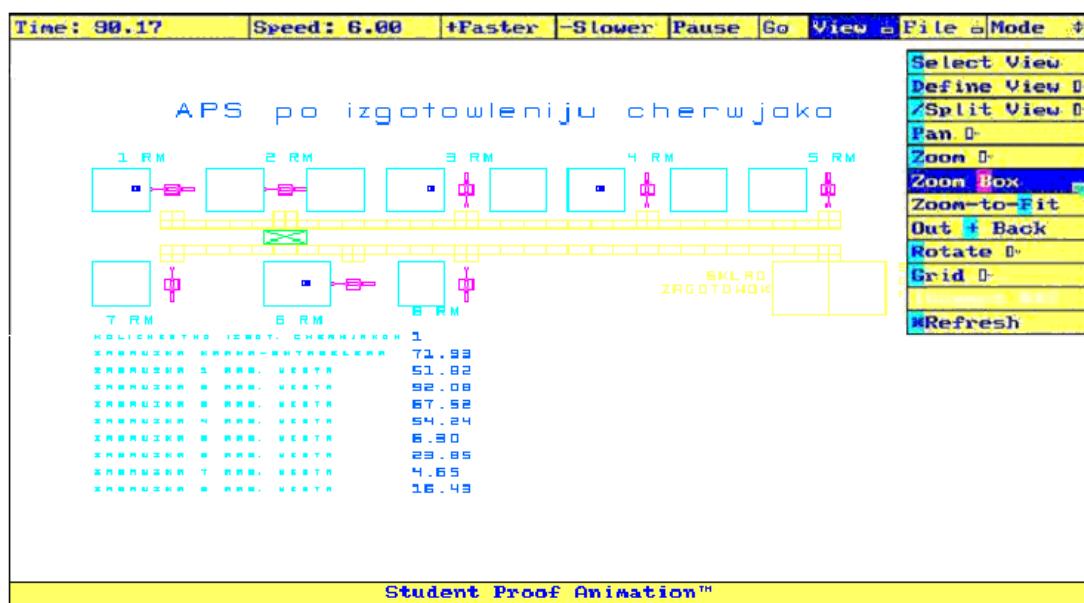


Рисунок 41 – Команда увеличения изображения **Zoom Box**

Затем появляется прямоугольник, размеры которого регулируются движением мыши (рис. 42). С его помощью выбираем участок, который необходимо увеличить.

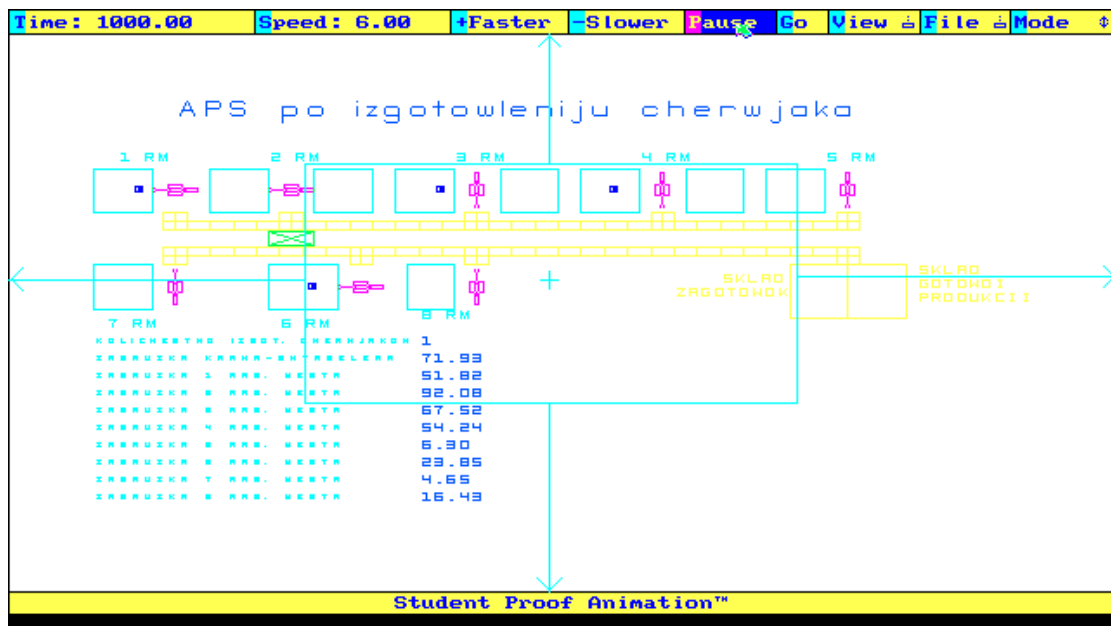


Рисунок 42 – Прямоугольник увеличения **Zoom Box**

Также изображение можно увеличить или уменьшить с помощью команды **Zoom** из меню **View** (рис. 43).

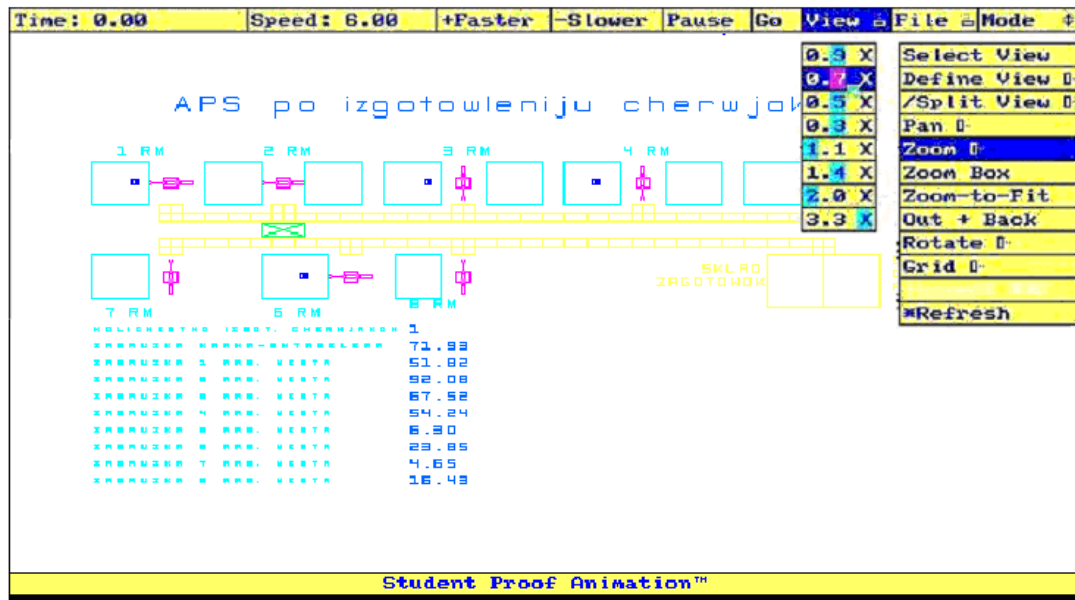


Рисунок 43 – Команда увеличения изображения **Zoom**

Для перемещения окна просмотра используется команда **Pan** из меню **View** (рис. 44). Она сдвигает окно просмотра на 25% влево

(Left 25%), вправо (Right 25%), вверх (Up 25%) или вниз (Down 25%) без изменения масштаба просмотра.

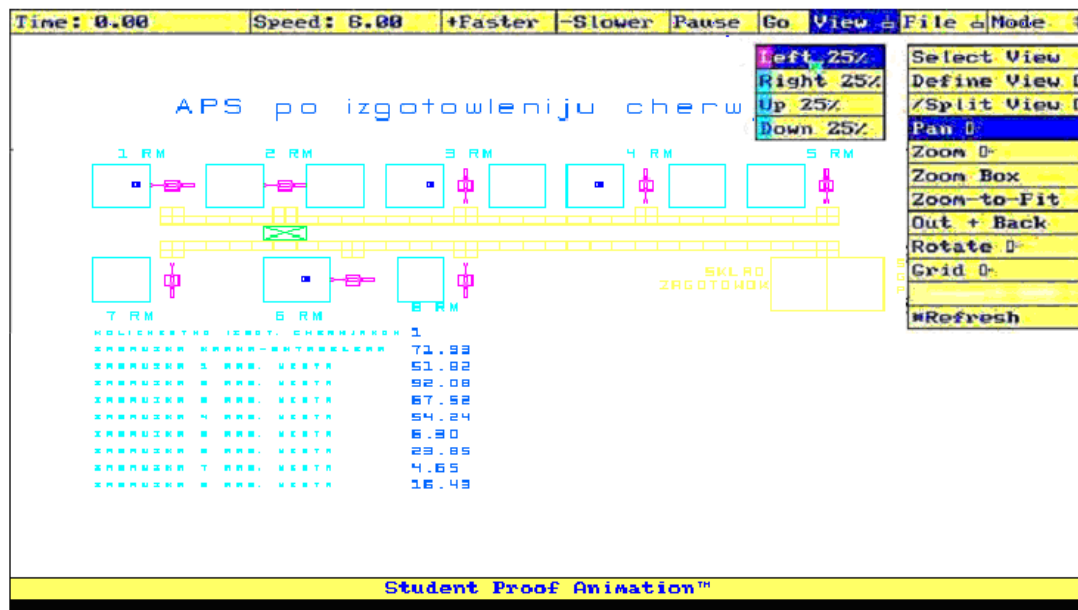


Рисунок 44 – Команда перемещения окна просмотра Pan

В Proof Animation имеется возможность изменения скорости анимации с помощью команд **Faster**, **Slower** и **Speed** (рис. 45).

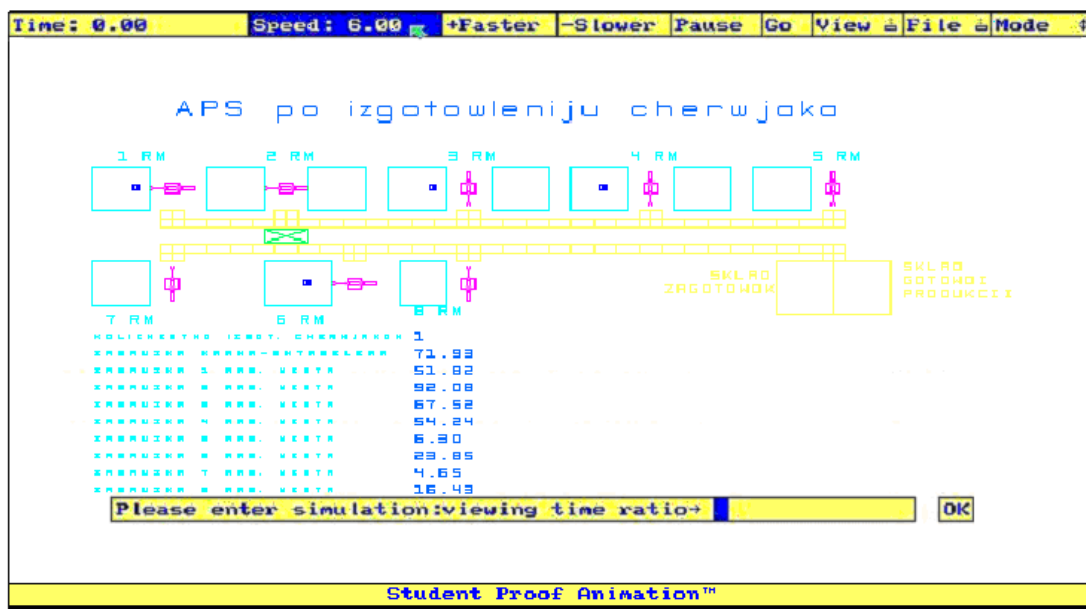


Рисунок 45 – Задание командой Speed скорости анимации

Faster и **Slower** – увеличение или уменьшение текущей скорости шагами по 10% используется для замедления или ускорения времени процесса. **Speed** (скорость) – отображение скорости анима-

ции, которая измеряется отношением к реальному времени. Для изменения скорости щёлкните мышью по текущей скорости. При этом в нижней части экрана появится надпись: **Please enter simulion: viewing time ratio** (введите новый коэффициент скорости). Необходимо ввести новый коэффициент скорости. Так можно прогнозировать ход процесса.

2.6 Проведение имитационных экспериментов

После создания системы, подтверждения правильности её работы и достоверности результатов можно перейти к проведению имитационных экспериментов с целью определения показателей системы при различных входных данных, таких как: количество изготовленных червяков за определённый период времени, коэффициенты загрузки станков, манипуляторов и крана-штабелёра в случае стандартного расчёта времени обработки и в случае учёта внецикловых потерь. Также проведены эксперименты, показывающие зависимость производительности системы от скорости горизонтального перемещения крана-штабелёра (табл. 1).

Таблица 1
Зависимость производительности системы и загрузки крана-штабелёра от скорости его перемещения.

№ п/п	Скорость, м/мин	Загрузка крана-штабелёра, %	Производительность, шт./год
1	1	100	118937
2	1,8	96	166966
3	3	72	172070
4	4	53	173117
5	6	43	173762

Ниже представлены графики (рис. 46), построенные на основе данных табл. 1.

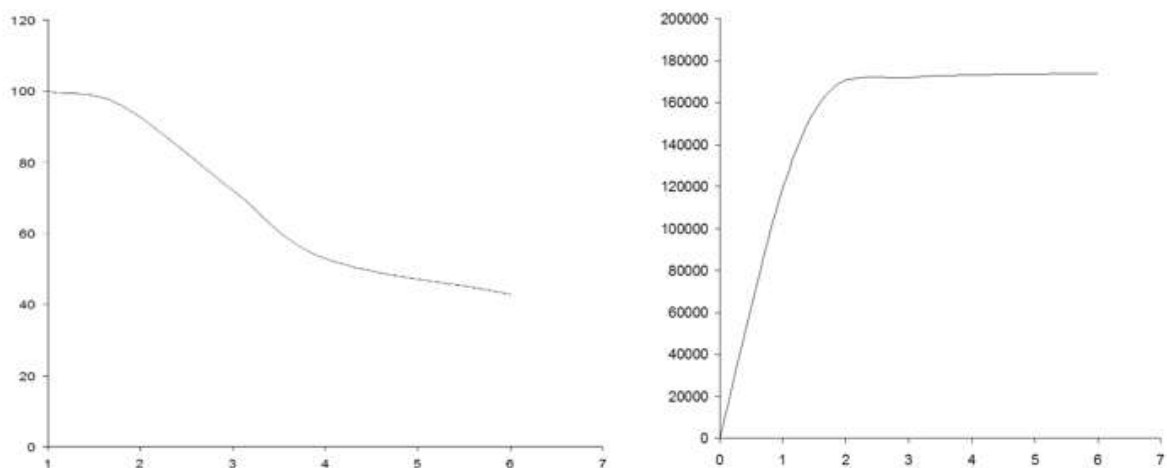


Рисунок 46 – Зависимости производительности АПС и загрузки крана-штабелёра от скорости его горизонтального перемещения

Как видно из графиков, загрузка крана-штабелёра при увеличении скорости его перемещения уменьшается, а производительность системы возрастает. Этот рост ограничен пределом скорости. На графике также видно, что рост производительности сначала пропорционален росту скорости, а затем вообще прекращается. Это объясняется тем, что время на загрузку/разгрузку крана начинает существенно превышать время перемещения между станками.

КОНТРОЛЬНЫЕ ВОПРОСЫ К РАЗДЕЛУ 2

1. Какие переменные и оператор используются для ввода входных параметров модели из внешнего файла в GPSS/H?
2. Какой блок и управляющий оператор используются для формирования файла управления анимацией?
3. Какие стандартные числовые атрибуты используются для вывода результатов моделирования на монитор компьютера?
4. В какие части GPSS-кода необходимо вводить команды файла управления Student Proof Animation?
5. В какой директории нужно сохранять файла управления анимацией?

3 АНИМАЦИОННОЕ ПРЕДСТАВЛЕНИЕ СИСТЕМ В GPSS STUDIO

3.1 Создание нового файла проекта

Процесс построения анимации в GPSS Studio представляет собой процесс построения «скелета» на основе кода модели.

Анимация строится на основе ранее созданной имитационной модели на GPSS World или Studio. Сначала необходимо создать проект (рис. 47).

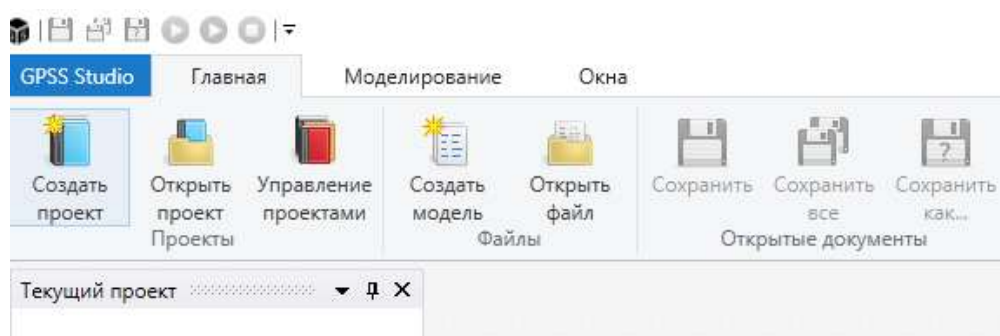


Рисунок 47 – Создание проекта

Затем проекту присваивается имя, при этом есть возможность добавить описание в соответствующем поле (рис. 48).

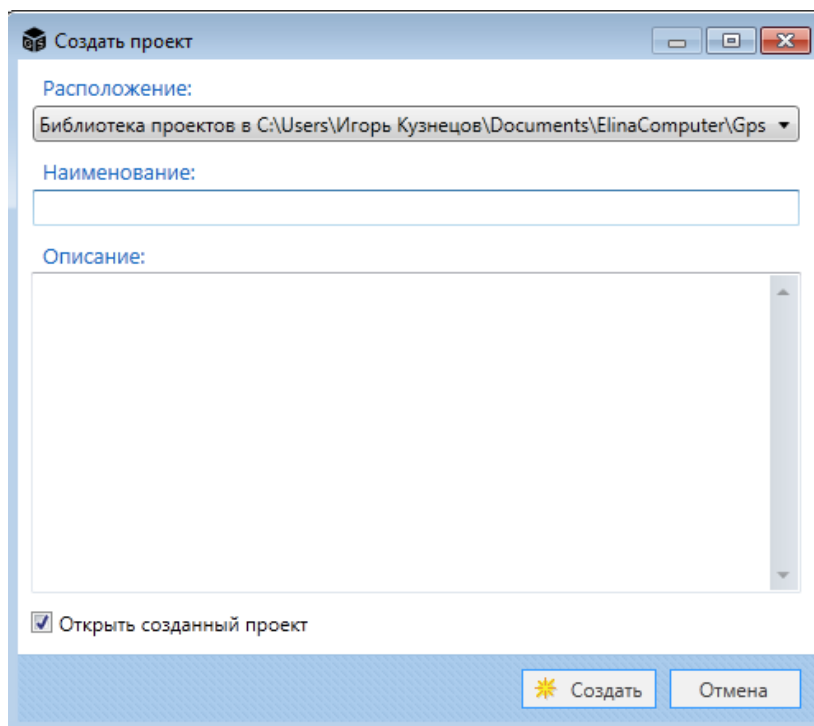


Рисунок 48 – Установка имени нового проекта и его описания

После создания проекта откроется новое окно, в котором будет отображено всё древо проекта (рис. 49). Анимационная форма модели создаётся путём перехода к элементу «Форма» (рис. 50), которой аналогично проекту даётся имя (рис. 51).

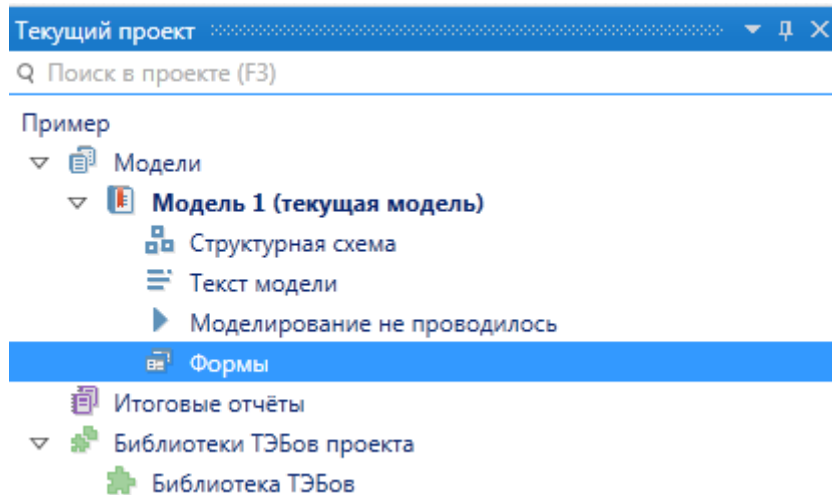


Рисунок 49 – Древо проекта

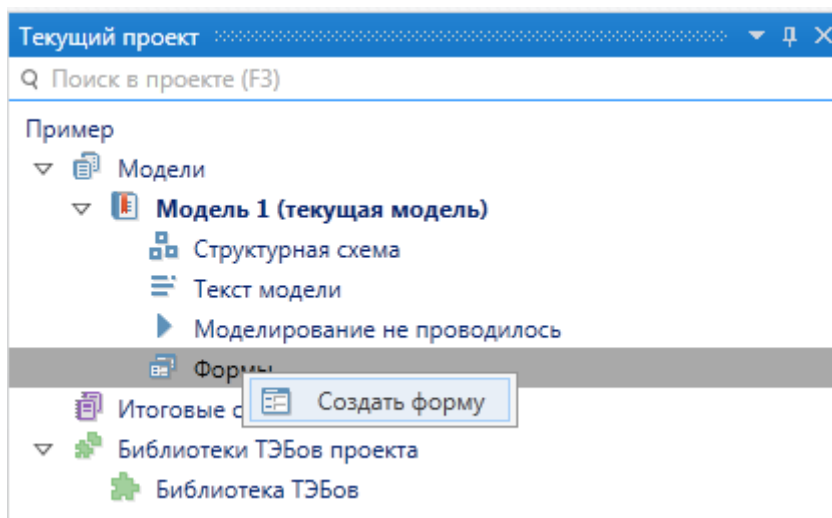


Рисунок 50 – Создание формы

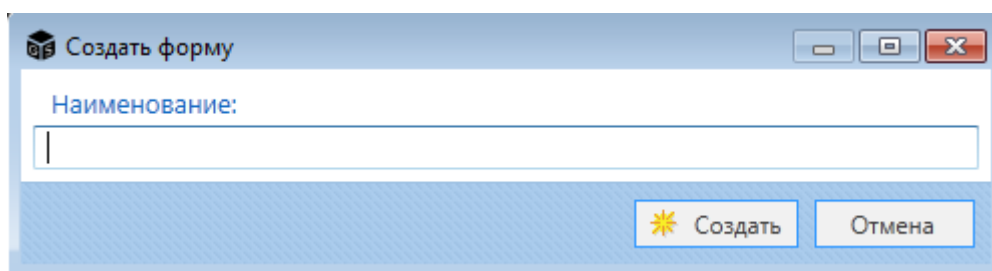


Рисунок 51 – Название новой формы

Для дальнейшей работы с проектом открываем созданную форму (рис. 52).

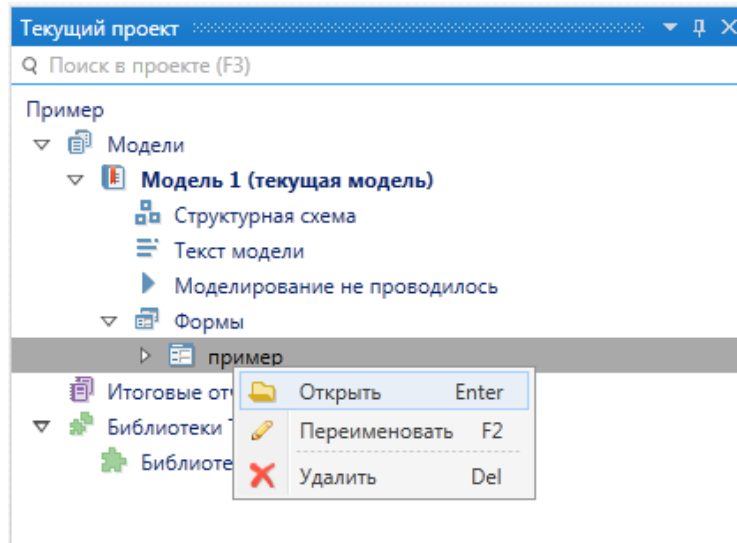


Рисунок 52 – Открытие созданной формы проекта

3.2 Наполнение формы

После создания и открытия новой «Формы» откроется рабочее поле и необходимые для работы вкладки.

Рабочее пространство формы содержит следующие элементы.

Вверху располагаются вкладки:

- *Указать модель* – указание имитационной модели, на основе которой будет создаваться форма;
- *Открыть файл результатов* – открытие ранее полученных результатов экспериментов;
- *Ввод данных* – ввод/редактирование исходных данных при проведении имитационных экспериментов;
- *Планирование экспериментов* – задание зависимых и независимых факторов;
- *Динамика хода эксперимента* – визуальное отображения хода экспериментов в виде графиков, чисел и др. элементов;
- *Анимационная форма* – создание анимационного представления созданной модели;
- *Настройка модели* – настройка модели, и подготовка формы к запуску, этом разделе указывают:
 - Период моделирования: либо по временному показателю, либо по количеству уничтоженных транзактов;

- Единицу модельного времени (секунды, минуты, часы и т. д.);
- Дату и время начала моделирования в модели.
- *Настройка мониторинга* – фиксация динамики зависимых факторов при проведении экспериментов, включает в себя следующие элементы:
 - сбор динамики с указанием периода сбора этих данных;
 - сбор данных для анимации – является обязательным условием для запуска анимации, после проведения экспериментов;
 - дерево объектов мониторинга, которые будут отражаться в результатах моделирования.
- *Проверка формы* – проверка правильности построения всех элементов формы и включает в себя следующие основные элементы:
 - *ввод данных* – указывают входные данные;
 - *планирование* – указывают зависимые и независимые факторы, если ранее они не были указаны;
 - *моделирование* – запуск модели, с указанием типа эксперимента: одиночный или серия экспериментов, настройки модели, настройки мониторинга и настройки самого приложения GPSS Studio;
- *Результаты* – отображение результатов моделирования и включает элементы:
 - *общая информация* – отображаются основные показатели моделирования: начало и конец моделирования, интервал сбора данных, единицу модельного времени и др.;
 - *текст модели* – отображение кода модели;
 - *исходные данные* – отображение входных данных, которые применялись для проведения экспериментов;
 - *ролик динамики* – демонстрация динамики хода экспериментов;
 - *анимационный ролик* – демонстрация и запуска анимационной формы модели;
 - *динамика показателей* – отображение динамики зависимых факторов, в виде графиков;
 - *стандартный отчёт* – просмотр результатов моделирования.

Справа располагается «Тип дерева» – отображение всей модели, но в разных условиях:

- *Дерево текста модели* – указываются все блоки и команды модели;
- *Дерево объектов для мониторинга* – указываются все элементы для фиксации динамики зависимых факторов;
- *Дерево блоков для анимации* – указываются все необходимые блоки для построения анимации.

Внизу располагается код имитационной модели, для которой создаётся форма.

3.3 Элементы построения анимации в GPSS Studio

Для создания анимации используют вкладку «**Анимационная форма**», включающую полотно для работы с анимацией (рис. 53), фон которого по предпочтению разработчика или требованию заказчика может редактироваться (рис. 54).

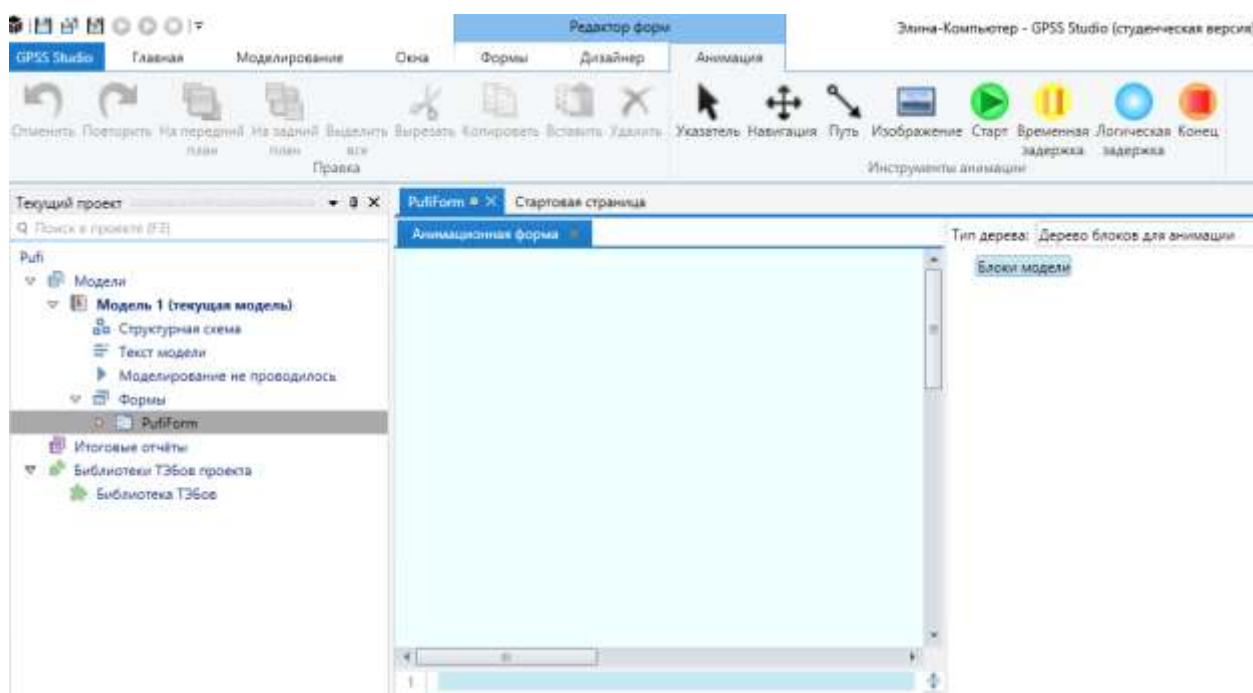


Рисунок 53 – Полотно для создания анимационной формы модели

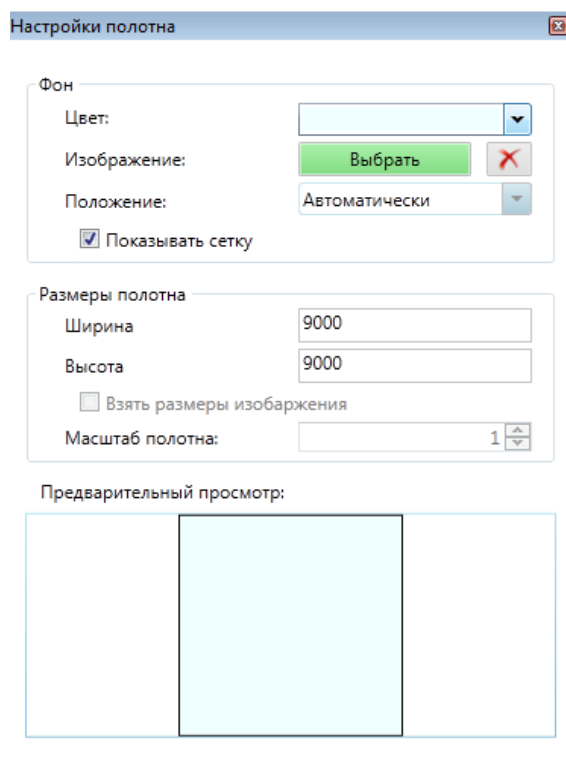


Рисунок 54 – Редактирование фона рабочего полотна

Вкладка «Анимация» используется для добавления элементов анимации:

- *Указатель* – обращение к конкретному элементу анимации;
- *Навигация* – перемещение по рабочему полю;
- *Путь* – соединение основных элементов между собой и в необходимом направлении;
- *Изображение* – размещение изображения всех форматов (.jpg, .png и др., кроме .gif файлов, которые будут отображаться в виде статической картинки);
- *Старт* – указание начала анимации, соответствует блокам GENERATE, SPLIT;
- *Временная задержка* – отображение выполнения какого-либо процесса – блок ADVANCE;
- *Логическая задержка* – отображение устройств, условий направления примитивов по блокам, свойства блоков (например, позволять создание очереди, и др.), соответствует блокам: ENTER, SEIZE, LEAVE, RELEASE, ALTER, CLOSE DISPLACE, EXAMINE, FUNAVAIL, EXECUTE, GATE, GATHER, LINK, LOOP, MATCH, PREEMPT, REMOVE, SCAN, SELECT, TEST, TRANSFER, UNLINK, WRITE;

- *Конец* – отображение окончания работы анимации, соответствует блокам TERMINATE, ASSEMBLE.

3.4 Динамические элементы анимации в GPSS Studio

В анимационном представлении моделей, помимо статических объектов, присутствуют и динамические элементы – примитивы.

Примитивы, соответствующие транзактам имитационной модели, – элементы, движущиеся в анимации, которые могут отображать детали, автосамосвалы, заготовки и т. п. Внешний вид примитива выбирается в зависимости от предпочтений разработчика или требований заказчика. Устанавливаются и редактируются примитивы в любом элементе анимационной формы в разделе «**Установить примитивы**» (рис. 55, 56).

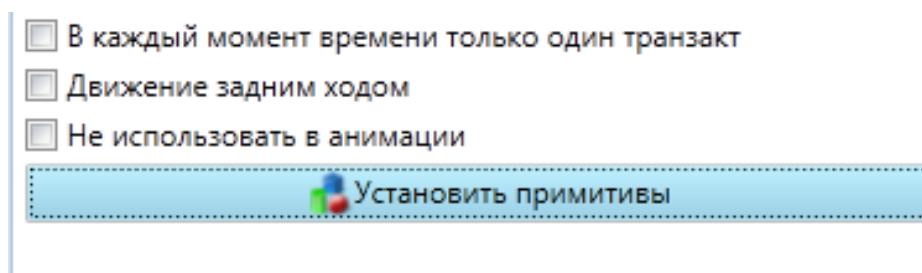


Рисунок 55 – Установка примитивов

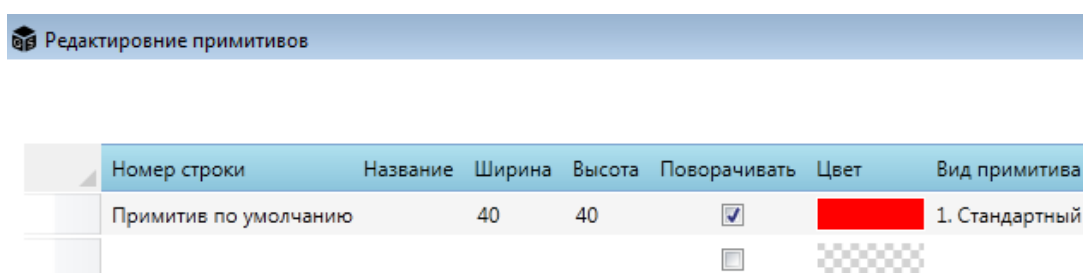


Рисунок 56 – Выбор вида примитива

Для добавления нового примитива существует возможность выбора его из списка стандартных примитивов на вкладке «**Список всех примитивов**».

Помимо стандартных примитивов имеется возможность создания нового, используя пиктограмму «Плюс» и указывая путь к объекту; чаще всего это графические файлы популярных форматов – PNG, JPG, BMP (рис. 57).

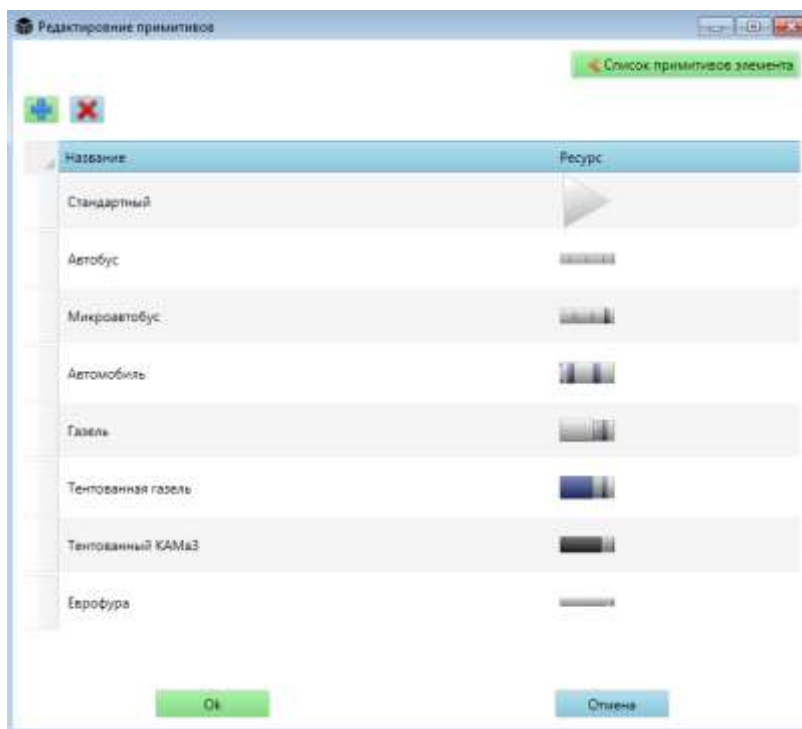


Рисунок 57 – Список примитивов

3.5 Пример построения анимационной формы в GPSS Studio

Разработаем анимационную модель АПС, представленную на рисунке 29. Имитационная модель данной АПС создана в разделе 2, опустив все этапы, связанные со сбором данных, разработкой концептуальной модели и переноса её в компьютер, переходим к этапам построения анимационной формы и запуску анимации.

Построение статических элементов по своей сути и принципу представляет собой расположение блоков GPSS модели на полотне в той последовательности, в которой они написаны в коде.

Построить статические элементы анимационной формы (рис. 58) можно двумя способами:

- 1) выбрав из списка элементов статических объектов, располагающийся сверху, разместить их в любом месте полотна, но при этом необходимо указать, какой блок соответствует этому элементу;
- 2) используя код GPSS-модели, располагающийся справа в древе модели, разместить элементы на полотне так, как требует решаемая задача.

При этом важно в обоих случаях указывать связь блоков между собой, используя варианты блока ADVANCE (рис. 59).

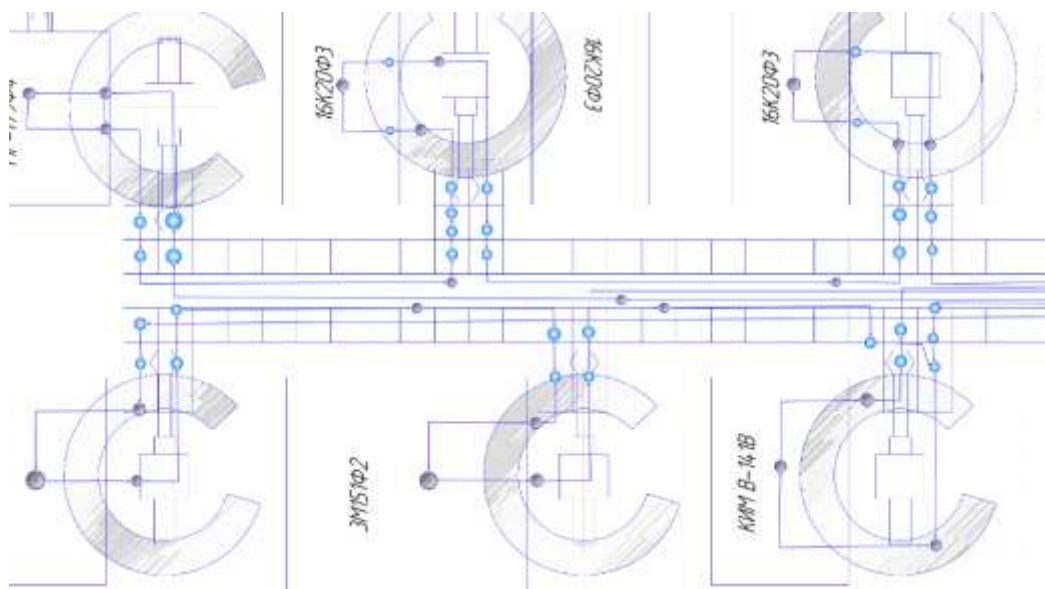


Рисунок 58 – Фрагмент статистического элемента анимации

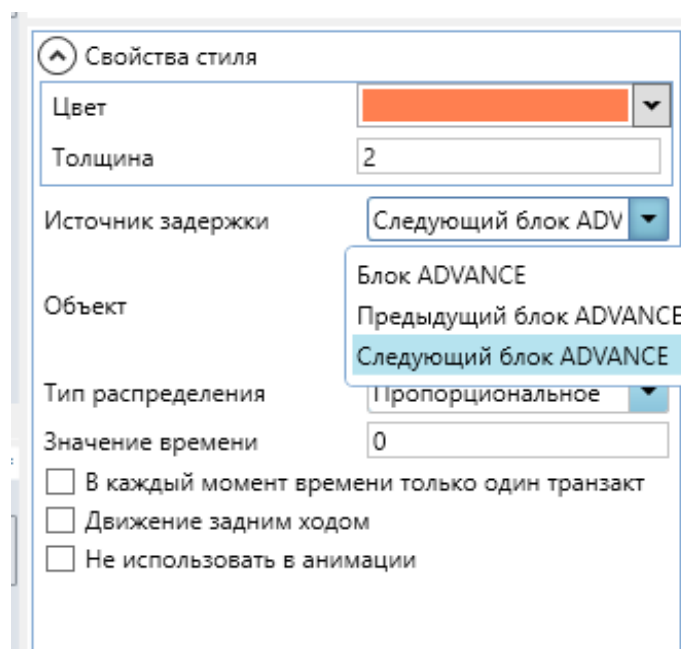


Рисунок 59 – Связь статистических элементов при задании движения

Расположение статических объектов и указание связей между ними завершает этап создания статических элементов анимационной формы.

После создания статичных элементов переходим к созданию динамических элементов. Динамическими элементами в модели являются заготовки в виде прутков (рис. 60) и изготовленные детали (рис. 28). Эти динамические элементы добавляются в список примитивов (рис. 61). После указываются блоки, где необходимо разместить выбранные примитивы (рис. 62, 63).

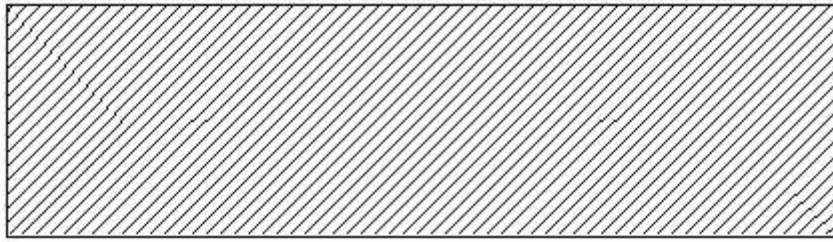


Рисунок 60 – Динамический элемент – заготовка

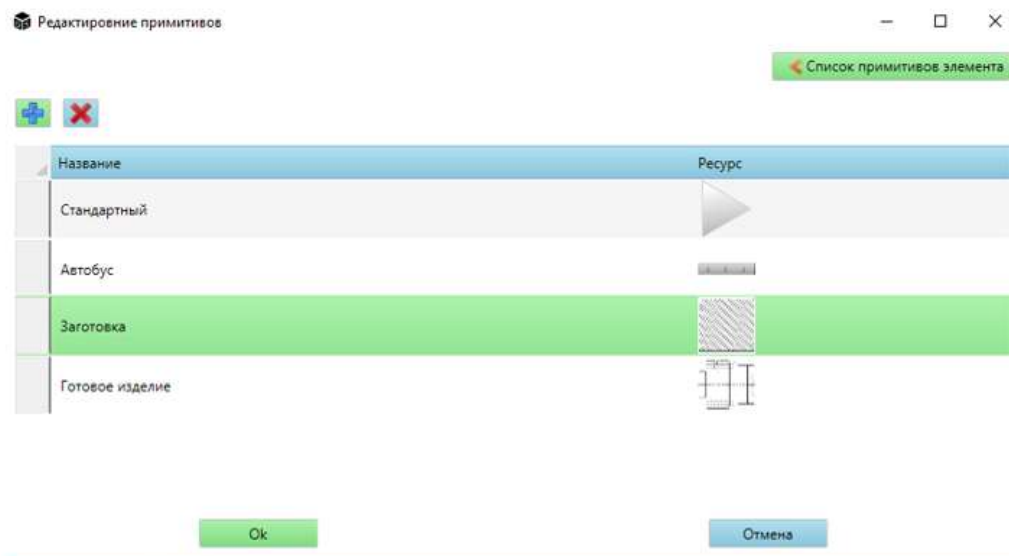


Рисунок 61 – Добавление необходимых примитивов

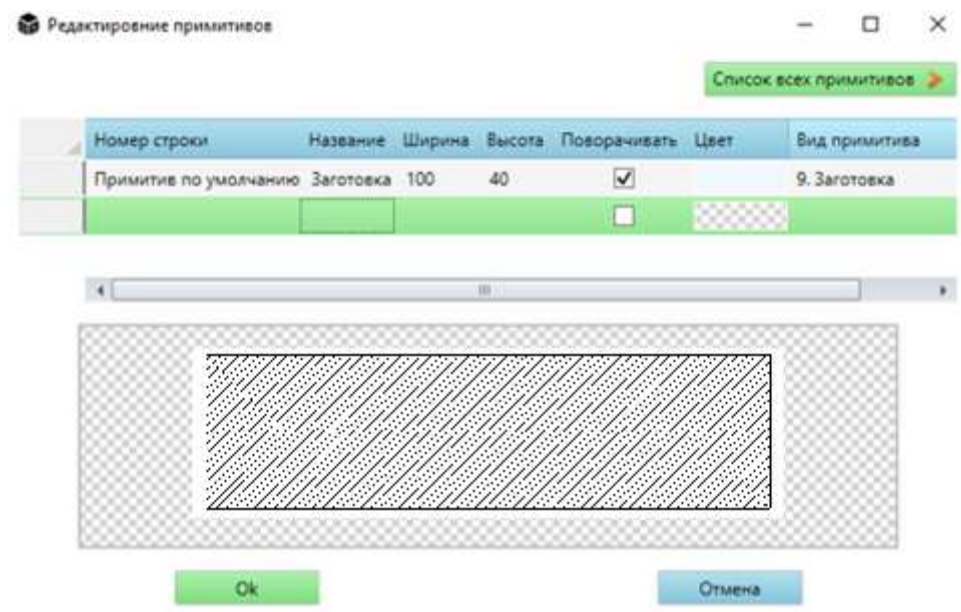


Рисунок 62 – Установка примитива «Заготовка» в блоке GENERATE

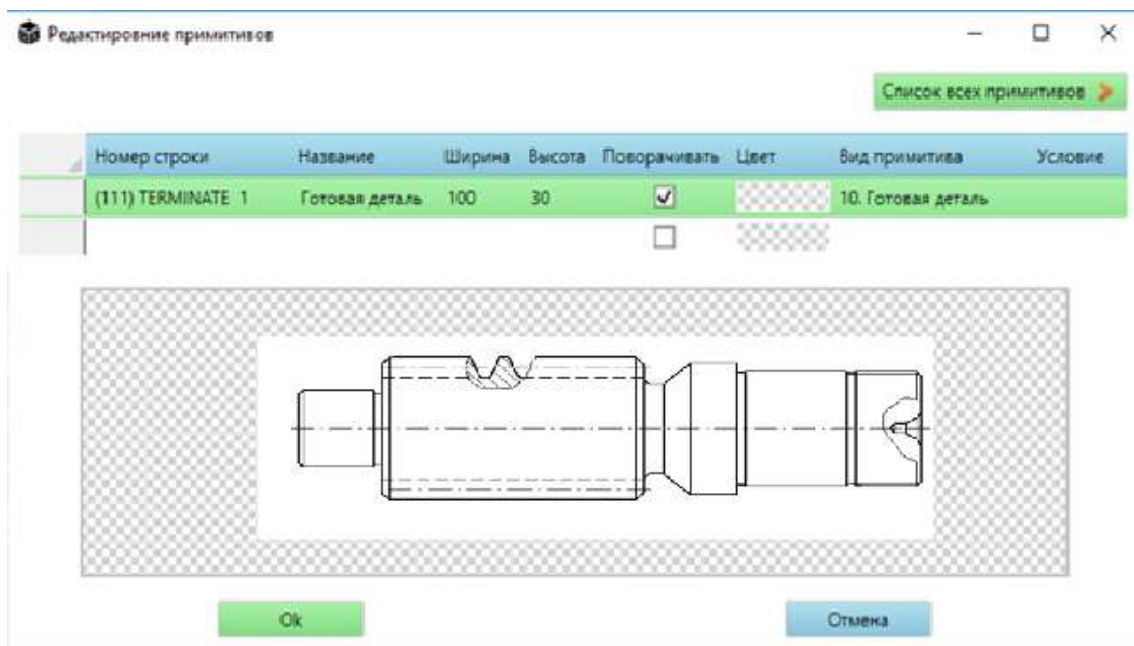


Рисунок 63 – Добавление примитива «Готовая деталь» в блоке TERMINATE

По желанию пользователя добавлять примитивы можно в любые позиции модели.

На этом построение анимации завершено. Для проверки правильности построения анимации необходимо провести пробный эксперимент; если никаких ошибок не возникло при проигрывании анимации, то анимация построена верно; в противном случае выветится окно с сообщением об ошибке. Самой распространенной ошибкой является неправильное указание путей (рис. 64).

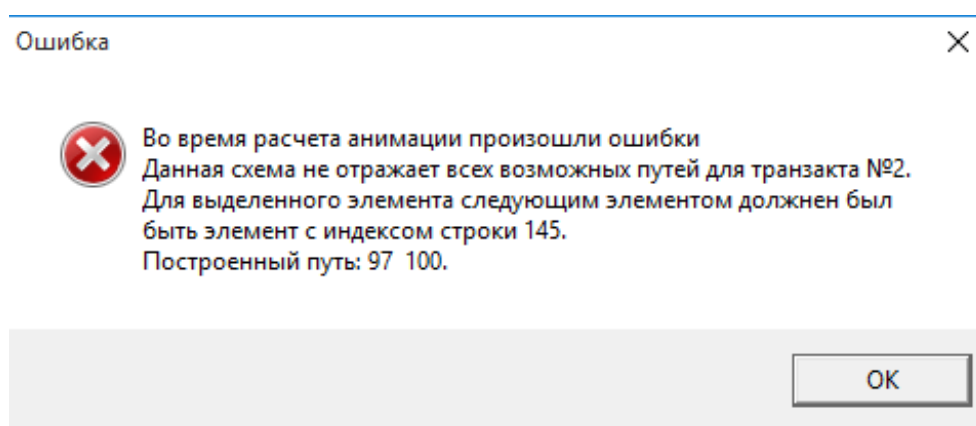


Рисунок 64 – Ошибка при создании анимации

На этапе проведения экспериментов в качестве факторов (рис. 65) выберем:

- количество заготовок;

– число станков на позициях 2, 3, 4;

Планирование экспериментов

Автоматическое построение плана с использованием шага

Факторы Целевые показатели План серии экспериментов

Название	Псевдоним	Операнд	Шаг	Минимальное значение	Максимальное значение	Участует в эксперименте
Количество станков в позиции 2, шт.	RM2	Операнд А	1	1	2	<input checked="" type="checkbox"/>
Количество станков в позиции 3, шт.	RM3	Операнд А	1	1	2	<input checked="" type="checkbox"/>
Количество станков в позиции 4, шт.	RM4	Операнд А	1	1	2	<input checked="" type="checkbox"/>
Количество заготовок, шт.	(5) GENERATE	Операнд D	100	100	300	<input checked="" type="checkbox"/>

Рисунок 65 – Факторы

В качестве целевых показателей (рис. 66) и их динамической фиксации выберем коэффициенты использования станков, роботоманипуляторов и крана-штабелёра. Шаг модельного времени можно выбрать любой; в нашем примере период сбора информации выбран равный 1 минуте. Обязательно необходимо установить «галочку» «Собирать данные для анимации» (рис. 67).

Планирование экспериментов

Автоматическое построение плана с использованием шага

Факторы **Целевые показатели** План серии экспериментов

Название	Псевдоним объекта	Тип объекта	Название СЧА
Коэффициент использования станков в позиции 2	RM2	Память	SR
Коэффициент использования станков в позиции 3	RM3	Память	SR
Коэффициент использования станков в позиции 4	RM4	Память	SR
Коэффициент использования крана - штабелера	SHTABEL	Устройство	FR
Коэффициент использования станка в позиции 1	RM1	Устройство	FR
Коэффициент использования станка в позиции 5	RM5	Устройство	FR
Коэффициент использования станка в позиции 6	RM6	Устройство	FR
Коэффициент использования станка в позиции 7	RM7	Устройство	FR
Коэффициент использования станка в позиции 8	RM8	Устройство	FR
Коэффициент использования робота - манипулятора в позиции 2	ROBOT2	Устройство	FR
Коэффициент использования робота - манипулятора в позиции 3	ROBOT3	Устройство	FR
Коэффициент использования робота - манипулятора в позиции 4	ROBOT4	Устройство	FR

Рисунок 66 – Фрагмент «Целевые показатели»

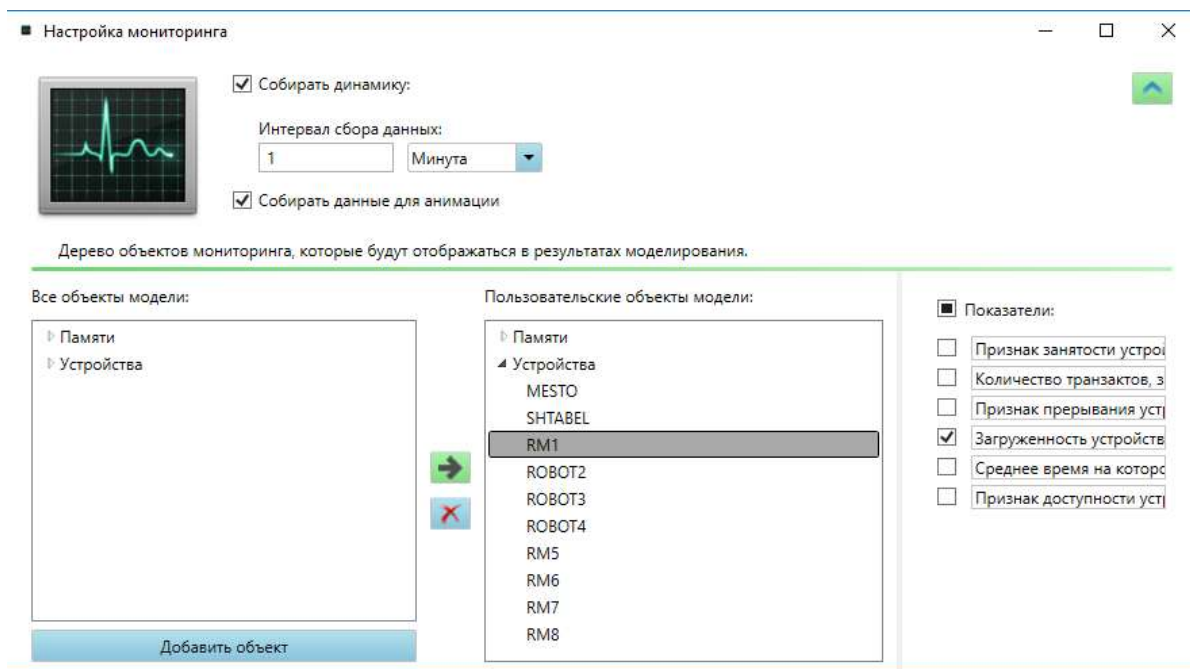


Рисунок 67 – Мониторинг динамики целевых показателей

Существует два способа создания плана имитационных экспериментов.

1. Вручную. При выборе такого способа разработчик сам вручную составляет план серии экспериментов и указывает все необходимые значения фактора(ов) и их всевозможных комбинаций.

2. Построение плана автоматически. В данном случае необходимо указать минимальное и максимальное значения и шаг изменения фактора. План серии экспериментов будет создан автоматически.

В нашем примере выберем автоматический способ создания плана серии экспериментов с указанием шага для манипуляторов – 1 шт., для количества заготовок – 100 шт., для времени появления заготовок – 10 с. Фрагмент плана серии экспериментов представлен на рисунке 68.

Планирование экспериментов

Автоматическое построение плана с использованием шага

Факторы Целевые показатели **План серии экспериментов**

	Количество станков в позиции 2, шт.	Количество станков в позиции 3, шт.	Количество станков в позиции 4, шт.
1	1	1	1
2	2	1	1
3	1	2	1
4	2	2	1
5	1	1	2
6	2	1	2
7	1	2	2

Рисунок 68 – Фрагмент «Плана серии экспериментов»

Укажем период моделирования по времени – 8 часов (рис. 69).

Настройка модели

Название модели:
АПС

Единица модельного времени: 1 Минута

Дата и время начала моделирования в модели: 21.02.2018 14:31:48

Завершение моделирования по времени

Период моделирования: 8 Час

Завершение моделирования по количеству уничтоженных транзактов

Количество транзактов: 20

Ok Отмена

Рисунок 69 – Настройка модели с указанием периода моделирования

Далее нажатием кнопки запускаем модель и ждём, когда закончится серия экспериментов (рис. 70).

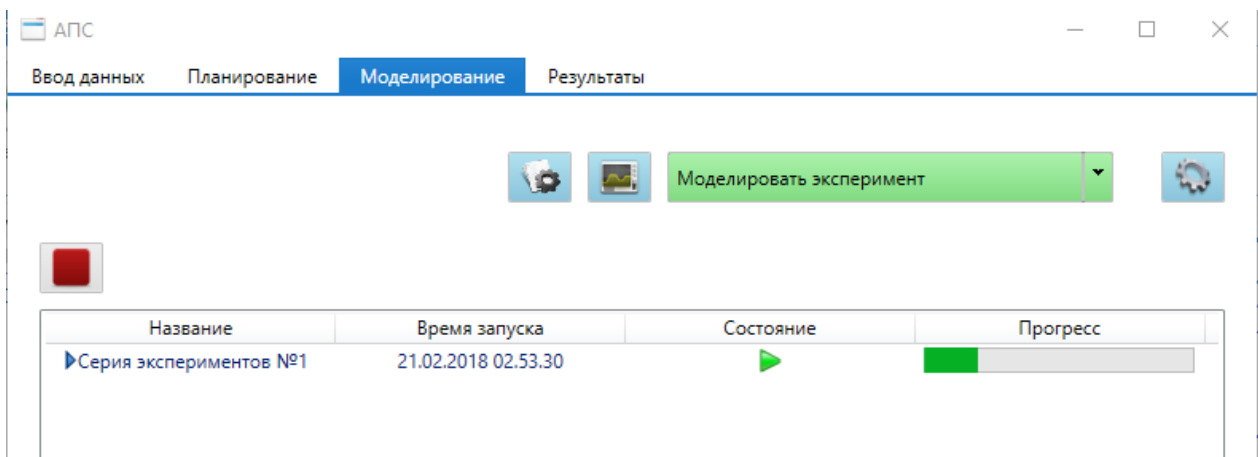


Рисунок 70 – Проведение экспериментов

После завершения имитационных экспериментов во вкладке «**Результаты**» отобразятся все результаты по проведенным экспериментам. Для просмотра анимации выберем вкладку «**Анимационный ролик**» (рис. 71), на которой отображается анимационный ролик, при просмотре которого можно менять скорость, изменять масштаб и выбирать необходимый период просмотра анимации (рис. 72).

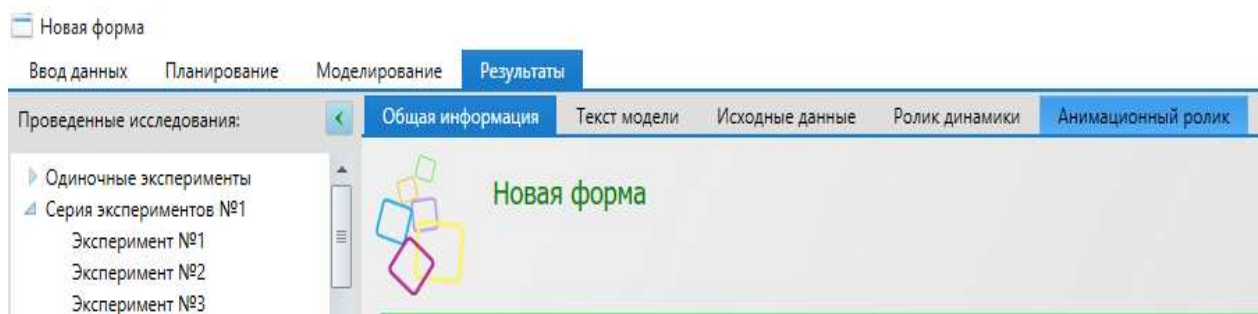


Рисунок 71 – Выбор раздела «Анимационный ролик»

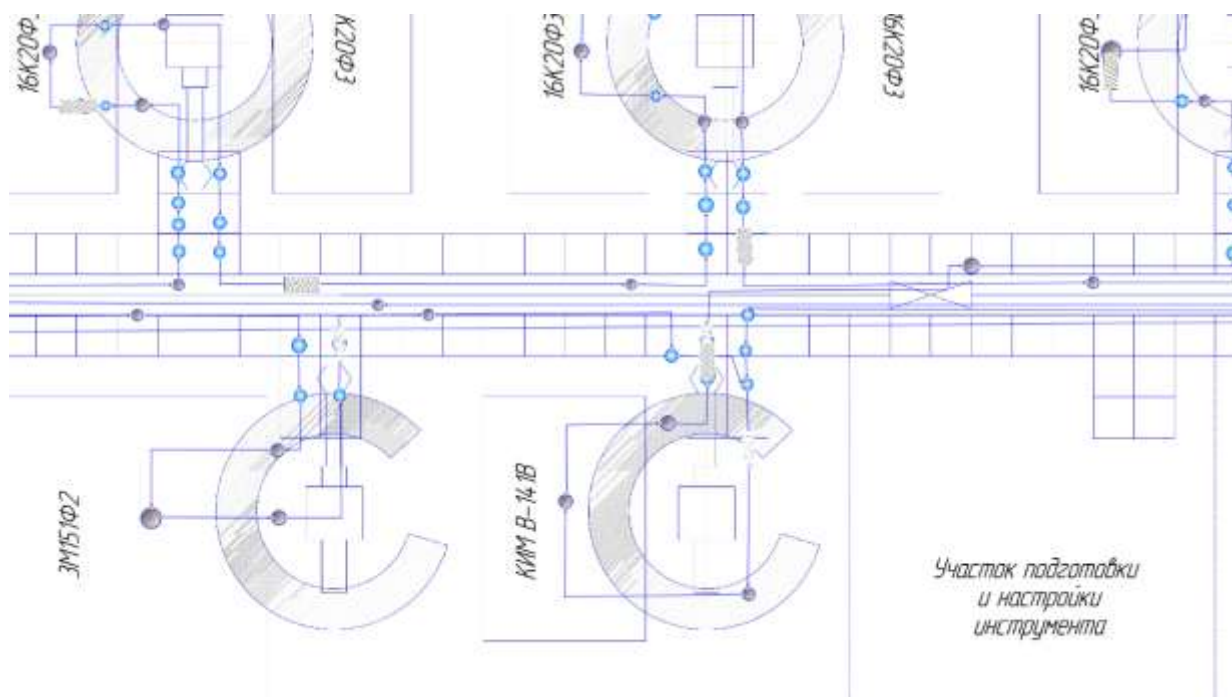


Рисунок 72 – Фрагмент работы анимации АПС

Имитационные модели позволяют исследовать влияние параметров технологии и внешних воздействий на основные характеристики АПС. Так, например, внецикловые потери времени оказывают существенное влияние на производительность системы. На указанной модели проведена серия экспериментов по оценке производительности АПС с краном-штабелёром без учёта и с учётом внецикловых потерь при различном времени моделирования. Результаты экспериментов представлены в виде графика (рис. 73).

На графике сплошной линией показана производительность системы без учёта потерь, а пунктиром – с учётом их.

Как видно из графика количество деталей, изготовленных системой за определённый период времени, с учётом потерь меньше, чем при стандартном нормировании времени обработки. Причём, если на коротких промежутках времени это различие незначительно – 14 штук при 500 мин работы, то при 50000 мин – 1044 штуки. В то время, когда согласно расчётам АПС должна будет произвести заданное количество деталей – 200000 штук, на самом деле будет изготовлено на 8641 штуку меньше.

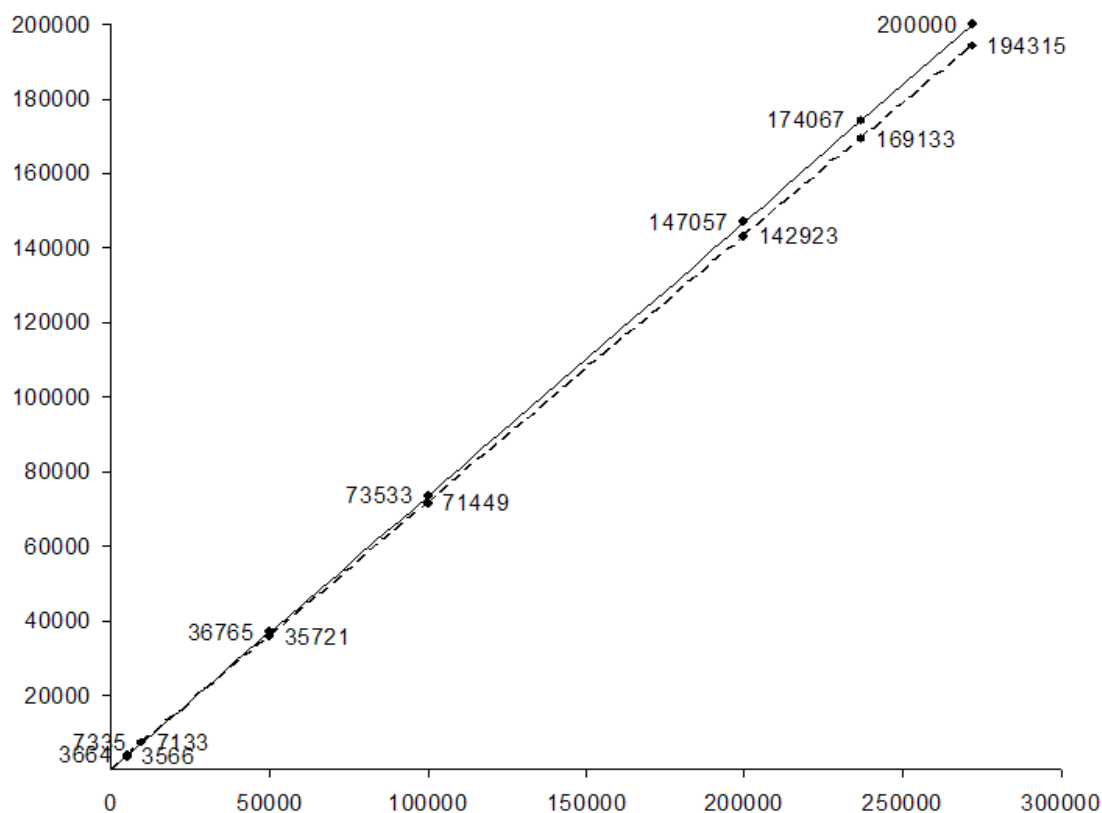


Рисунок 73 – Производительность АПС без учёта и с учётом внецикловых потерь

КОНТРОЛЬНЫЕ ВОПРОСЫ К РАЗДЕЛУ 3

1. В чем особенность построения анимации в GPSS Studio?
2. Какие статистические элементы присутствуют при построении анимации?
3. Дайте определение понятию «примитив».
4. Особенность связи статистических элементов между собой.
5. Каким образом осуществляется запуск анимации в GPSS Studio?

ЗАКЛЮЧЕНИЕ

Компьютерная имитация и анимация в настоящее время находят применение практически во всех отраслях и сферах деятельности человека: рекламе, обучении, исследованиях, искусстве, фильмах, видео, интернете, мультимедийных презентациях. Наиболее известными методами визуализации процессов в наши дни являются 3D-анимация и 3D-модели. Часто объединяют видеоматериалы и компьютерную графику. Дизайнерские возможности визуализации очень велики. Однако сегодня всё большей популярностью пользуется анимация технологических процессов. В данном случае анимация – это отображение динамики работы оборудования на компьютерной мнемосхеме технологического процесса, формируемое имитационной моделью процесса.

Зачастую визуальное отображение технологического процесса необходимо не только самим конструкторам как помощь в разработке, но и для «внешних» нужд предприятия: клиенты и партнёры должны уяснить суть и разобраться в этапах процесса, даже не имея специальных знаний в области деятельности предприятия. Визуализация стала наиболее эффективным и выгодным методом доступного демонстрирования действующих на предприятии машин, установок и производственного процесса в целом.

Визуализация незаменима в том случае, когда требуется внедрение или разработка инновационной и ранее не применявшейся технологии. При помощи компьютерной графики и 3D-анимации можно наглядно показать все преимущества и перспективы технологии доступно и эффективно.

Разработка и создание динамических моделей, наглядно представляющих технологический процесс, преследует несколько целей:

- привлечение инвесторов за счёт наглядной демонстрации принципа действия объекта;
- знакомство с новыми технологиями, которые планируется вводить в целях оптимизации производственного процесса, а также возможность сравнить старые способы производства с новыми, наглядно отслеживая изменения;
- проверка адекватности модели объекту в деталях и в целом;

- выявление трудноуловимых ошибок имитационного моделирования;
- «проигрывание» различных ситуаций для непрограммирующего пользователя.

При анимации технологических процессов недостаточно представить только визуализацию операций и оборудования. Современные средства позволяют связывать компьютерные имитационные модели процессов с их анимационной моделью, что даёт более широкие возможности для исследователя, проектировщика или заказчика. Такая «связка» позволяет не только понять, как работает система, но и проводить имитационные эксперименты на модели объекта, отслеживая его параметры, а также визуально оценивать динамику при различных условиях и входных воздействиях.

В самом простом случае возможно применение языка компьютерной анимации Proof Animation, который является универсальным программным обеспечением системы анимации для персональных компьютеров, позволяющим проектировщику создавать любые цветные двухмерные и изометрические рисунки, формировать набор команд для движения элементов рисунка, редактировать последовательность движений. Простые инструкции языка Proof Animation сводятся к заданию времени и конечных точек перемещения нарисованных объектов. Изображение процесса строят из статических и динамических элементов. Основным недостатком такого подхода является сложность проведения экспериментов, т. к. каждый раз придется переписывать отдельную часть файла управления анимацией. Если перед разработчиком стоит задача создания не анимационного ролика, отображающего технологический процесс, а полноценной его модели, то целесообразнее разработать имитационную модель на специализированном языке компьютерной имитации (например, GPSS/H) и ввести в код модели команды управления анимацией. В этом случае появится возможность исследовать модель системы при варьировании параметров технологии и различных входных воздействиях. При этом все варианты можно оценить на анимационной модели.

В более поздних версиях среды моделирования GPSS особое внимание уделяется способам графического конструирования моделей, автоматизированному созданию интерфейсов ввода исходных данных в модель, анимации результатов, интерактивному анализу

результатов экспериментов с моделью и документированию результатов исследования. Пользователю предоставлена возможность строить анимационные схемы различных типов, от абстрактных до приближенных к реальным, используя реальные схемы, рисунки, карты.

В зависимости от поставленных задач у современного исследователя имеются широкие возможности, позволяющие создавать анимационные модели различных систем: от дизайнерских рекламных анимационных роликов до полноценных моделей сложных технических систем, разрабатываемых с использованием математических аппаратов и специализированных программных средств.

**СРАВНИТЕЛЬНАЯ ТАБЛИЦА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ДЛЯ СОЗДАНИЯ 2D-АНИМАЦИИ**

Название ПО	Производитель	Стоимость лицензии	Решаемые задачи	Достоинства	Недостатки
Toon Boom	Toon Boom Animation, Inc. www.toonboom.com	От 19,6\$ / месяц	Программа для создания анимации, используемая многими профессиональными студиями. Приложение позволяет переносить объекты с кальки в интерфейс утилиты	<ul style="list-style-type: none"> • Поддержка нескольких методов создания анимации • Отличный функционал 	<ul style="list-style-type: none"> • Требовательность к ресурсам компьютера • Сложный интерфейс для новичков
TV Paint Animation	TVPaint Developpement www.tvpaint.com	От 612,4\$	Приложение базируется на мощном движке для обработки частиц и включает ряд интересных возможностей – функцию очистки фоновой подложки ScanCleaner, инструмент для создания комплексной анимации с применением нелимитированного числа слоев Multiplane Camera и специальную опцию для обработки теней ToonShading. Готовые работы можно экспортировать в анимированные изображения или видеофайлы.	<ul style="list-style-type: none"> • Широкие возможности для импорта, включая многослойные изображения и звуковые файлы • Существуют специальные версии приложения для Linux, MacOS и Android • Удобная тайм-линия • Поддержка FPS до 60 	<ul style="list-style-type: none"> • Отсутствует полноценная поддержка 3D-формата • Существенное ограничение бесплатной версии • Довольно высокая цена лицензии • Нет русской локализации
Anime Studio Pro	Smith Micro Software Inc http://my.smithmicro.com/	От 299.99\$	Anime Studio Pro позволяет создавать мультипликационные фильмы – как короткометражные, так и полнометражные. Пользователю доступны интегрированные персонажи, набор эффектов, продвинутые инструменты для работы с деталями и фоном.	<ul style="list-style-type: none"> • Удобная работа со слоями и покадровая обработка • Импорт PSD-файлов из Adobe Photoshop • Совместимость с движком Unity 3D • Автоматическое шевеление губами при наложении аудиоряда 	<ul style="list-style-type: none"> • Нельзя использовать в течение триал-периода без добавления данных банковской карты • Интерфейс не переведен на русский язык

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ 1

Название ПО	Производитель	Стоимость лицензии	Решаемые задачи	Достоинства	Недостатки
Adobe Animate (Macromedia/Flash Professional)	Adobe https://www.adobe.com/	От 11,39\$	Flash – одновременно мощное по функционалу и простое в использовании средство для создания анимированных проектов на основе векторной графики с встроенной поддержкой интерактивности. Flash – хороший рабочий инструмент для художников и дизайнеров, который позволяет дополнять web-проекты звуком и анимацией.	<ul style="list-style-type: none"> • Плавная анимация, которая не требует ожидания загрузки и не «съедает» трафик • Звуковое сопровождение для действий пользователя, интегрирование видео- и аудио-вставок во флеш-ролики • Встроенный язык программирования ActionScript, наделяющий флеш-элементы высокой интерактивностью • Возможности создания программ и видеоролики для ПК 	<ul style="list-style-type: none"> • Для просмотра флеш-анимации в браузерах необходимо установить на ПК Adobe Flash player. Вместо анимированной картинке посетитель увидит пустое поле • Контент, выполненный во Flash, тяжеловесен и долго кэшируется, на его загрузку влияет скорость интернет соединения. • Flash-технология нагружает процессор и потребляет много оперативной памяти, а значит, плавное воспроизведение всех элементов анимации гарантировано только для владельцев современных ПК
Synfig Studio	https://www.synfig.org/	Бесплатно	<ul style="list-style-type: none"> - векторная анимация морфингом - перекладная анимация - скелетная анимация - встроенные фильтры и эффекты (размытие, деформация и пр.) 	<ul style="list-style-type: none"> • Многофункциональность • Частичный перевод на русский • Плагины • Бесплатная 	<ul style="list-style-type: none"> • Сложность управления

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ 1

Название ПО	Производитель	Стоимость лицензии	Решаемые задачи	Достоинства	Недостатки
Vectorian Giotto	http://vectorian.com/	Бесплатно	Бесплатная программа для создания флеш-роликов практически любой степени сложности! Включает в себя практически все необходимые инструменты для повседневной работы с флеш-анимацией и может быть использована как новичками, так и профессионалами. Основным отличием и достоинством данной программы является поддержка ActionScript 2.0, без которого просто нельзя представить создание профессионального флеш-контента	<ul style="list-style-type: none"> • Бесплатность • Наличие всех необходимых инструментов и схожесть интерфейса с Adobe Flash • Удобная работа с цветом • Модульность интерфейса • Наличие библиотеки эффектов и шаблонов 	<ul style="list-style-type: none"> • Большая ресурсоемкость при работе с несколькими слоями • Нет русского языка • Не совсем удобная работа с ActionScript • Нет поддержки FLA-файлов • Недоработанный алгоритм сжатия роликов
Мультатор	https://multator.ru/	Бесплатно	онлайн-редактор мультиков. Ресурс подойдёт для баловства или быстрых зарисовок, с которыми вы можете делиться сразу же после публикации.	<ul style="list-style-type: none"> • Простой и понятный интерфейс, отличная идея 	<ul style="list-style-type: none"> • Нет возможности рисовать с планшета или смартфона
GPSS World (Studio)	Элина-Компьютер http://elina-computer.ru/	От 80,66\$	Система GPSS World – это мощная среда компьютерного моделирования общего назначения, разработанная для профессионалов в области моделирования. Это комплексный моделирующий инструмент, охватывающий области как дискретного, так и непрерывного компьютерного моделирования, обладающий высоким уровнем интерактивности и визуального представления информации.	<ul style="list-style-type: none"> • Высокая производительность • Гибкость языка • Простое управление очередью • Автоматически предоставляет базовые выходные данные моделирования • Универсальный язык сценариев для проведения экспериментов 	<ul style="list-style-type: none"> • Сложные внутренние инструменты • Необходимо знание языка программирования

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ 1

Название ПО	Производитель	Стоимость лицензии	Решаемые задачи	Достоинства	Недостатки
PROOF Animation	Wolverine Software Corporation http://www.wolverinesoftware.com/	Бесплатно	<p>Применение компьютерной анимации даёт возможности:</p> <ul style="list-style-type: none"> - проверить адекватность модели объекту в деталях и в целом; - выявить трудноуловимые ошибки имитационного моделирования; - доказать заказчику правильность работы имитационной модели; - проиграть различные ситуации для непрограммирующего пользователя; - вести обучение на модели. 	<ul style="list-style-type: none"> • Простые инструкции языка Proof Animation сводятся к заданию времени и конечных точек перемещения нарисованных объектов • Изображение процесса строят из статических и динамических элементов • Возможны выбор цвета, ширины и типа линии; просмотр размеров и ориентации объекта до его ввода в рисунок; изображение кривой любой формы путём передвижения «мыши» 	Отсутствие англоязычный интерфейс

ПРИЛОЖЕНИЕ 2

СРАВНИТЕЛЬНАЯ ТАБЛИЦА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СОЗДАНИЯ 3D-АНИМАЦИИ

Название ПО	Производитель	Стоимость лицензии	Решаемые задачи	Достоинства	Недостатки
Zbrush	Pixologic http://pixologic.com/	от 500\$, пробный период – 45 дней	Программа для 3D-моделирования. Применяется для создания искусственных персонажей в мультипликации и кинематографе Утилита позволяет выбрать материал объекта. Это профессиональное приложение для художественной лепки объектов	<ul style="list-style-type: none"> • Принцип лепки фигур напоминает создание объектов из пластилина • Отличное качество прорисовки мельчайших деталей • Автоматическая реакция готовых моделей на свет • Огромное количество кистей и палитр 	<ul style="list-style-type: none"> • Программу лучше использовать, уже обладая базовыми навыками лепки • Отсутствует официальная русификация • Приложение нельзя применять для воспроизведения сюжетных линий
CrazyTalk	Reallusion Inc	\$59	Программа для генерирования анимированных персонажей из лиц на фотографиях. Приложение позволяет создавать говорящие объекты, способные выражать эмоции и озвучивать текст	<ul style="list-style-type: none"> • Опция речевого синтеза • Очень простой алгоритм создания актёров • Возможность экспорта персонажей в другие приложения • Отличный генератор эмоций 	<ul style="list-style-type: none"> • Утилита не подойдет для создания полнометражных мультфильмов • Скачать программу CrazyTalk нельзя на русском языке
iClone Pro	Reallusion Inc	от \$170	Программа для производства анимационных фильмов. Утилита обладает набором различных инструментов для работы с отдельными персонажами и жизненными сценами	<ul style="list-style-type: none"> • Возможность обработки отдельных компонентов • Огромная библиотека реквизитов • Редактируемая степень прозрачности • Качественная прорисовка деталей, включая тени 	<ul style="list-style-type: none"> • Интерфейс может показаться сложным для новичков • Требуется регистрация на сайте разработчика

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ 2

Название ПО	Производитель	Стоимость лицензии	Решаемые задачи	Достоинства	Недостатки
Autodesk Maya	Autodesk	123\$	Программа для 3D-моделирования отдельных объектов и их взаимодействия в сценах. Приложение содержит четыре собственных визуализатора, работающих по разным алгоритмам	<ul style="list-style-type: none"> • Четыре собственных ядра для визуализации • Огромный набор моделей • Открытый код • В Интернет для Maya 3D можно найти и скачать множество уроков (на торрент-трекерах, например), в том числе и на русском языке 	<ul style="list-style-type: none"> • Требовательность к производительности видеокарты и процессора • Autodesk Maya нельзя скачать с интерфейсом на русском языке
Autodesk 3Ds Max	Autodesk https://www.autodesk.ru/	97,21\$ / месяц	Эта система может использоваться во всех отраслях, начиная от архитектуры и дизайна интерьеров и заканчивая созданием мультфильмов и анимированных видеороликов. Autodesk 3Ds Max идеален для статичной графики. С помощью него быстро и технологично создаются реалистичные картинки интерьеров, экстерьеров, отдельных предметов	<ul style="list-style-type: none"> • Универсальность продукта позволяет применять её практически в любых отраслях трёхмерного моделирования • Понятная логика работы • Наличие русскоязычной локализации • Широкие возможности полигонального моделирования • Удобный и функциональный инструментарий в работе со сплайнами • Возможность детальной настройки расположения текстур 	<ul style="list-style-type: none"> • Бесплатная демо-версия имеет ограничения • Интерфейс усложнен большим количеством функций • Некоторые стандартные примитивы не подходят для работы, вместо них лучше использовать сторонний 3D-модели

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ 2

Название ПО	Производитель	Стоимость лицензии	Решаемые задачи	Достоинства	Недостатки
Blender	Blender Foundation https://www.blender.org/	Бесплатно	Позволяет создавать полноценные 3D игры благодаря набору инструментов для моделирования частиц (вода, огонь, дым), волос и движения тела. Программа предоставляет разнообразные предустановленные шаблоны движений – пользователю не придется самому прописывать простую анимацию походки персонажа. Имеет большую библиотеку, из которой пользователи могут загрузить объекты природы (деревья, местность, облака) и разнообразные фактуры. Преимуществом программы является встроенная возможность редактировать видео	<ul style="list-style-type: none"> • Бесплатна • Открытый код • Постоянное развитие • Небольшой размер установщика • Возможность создания игр • Кроссплатформенность • Большое количество модификаторов • Возможность создания анимации • Возможность риггинга (анимация с помощью «арматуры») • Настройка фона • Монтаж видео • Скининг • Трекинг видео • Возможность работы с хромакеем 	Отсутствие документации в базовой поставке, но её можно найти на сайте программы или в других источниках

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ 2

Название ПО	Производитель	Стоимость лицензии	Решаемые задачи	Достоинства	Недостатки
Modo	The Foundry Visionmongers Ltd https://www.foundry.com/products/modo	\$1799	Программа для трёхмерного конструирования объектов, рисования динамических сцен, рендеринга, сетевой визуализации, «лепки» скульптур и создания архитектурных проектов	<ul style="list-style-type: none"> • Создание анимационных моделей • Практически неограниченные инструменты для рендеринга • Предустановленные профили управления • Гибкие настройки свойств элементов • Подключаемые модули 	<ul style="list-style-type: none"> • Сложная процедура активации триал-периода • Нет перевода интерфейса на русский язык • Высокая стоимость
Adobe Animate	Adobe https://www.adobe.com/	\$10 в месяц	Программа для создания интерактивной векторной анимации в современных форматах. Приложение используется веб-разработчиками, телевизионными и рекламными студиями	<ul style="list-style-type: none"> • Быстрый обмен заготовками с другими приложениями библиотеки Creative Cloud • Возможность доступа к проектам с различных устройств • Поддержка 4K-разрешения • Преобразование работ в HTML5 • Меню на русском языке 	<ul style="list-style-type: none"> • Нет поддержки 32-битных платформ • На момент обзора недоступны справочные материалы • На освоение интерфейса требуется немало времени

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ 2

Название ПО	Производитель	Стоимость лицензии	Решаемые задачи	Достоинства	Недостатки
AutoCAD	Autodesk https://www.autodesk.ru/	97,21\$ в месяц	Эта программа обладает мощнейшим функционалом для двухмерного черчения, а также проектирования трёхмерных деталей разной сложности и назначения.	<ul style="list-style-type: none"> • Огромный набор инструментов и функций для профессионального проектирования • Поддержка работы с облачными сервисами • Возможность интеграции с электронными таблицами Excel • Большое количество поддерживаемых ОС • Работа с макетами • Возможность работы с 3D-принтерами 	<ul style="list-style-type: none"> • Высокая стоимость базовой версии ПО • Высокие системные требования
Sketch Up	https://www.sketchup.com/	695\$	Интуитивная программа для дизайнеров и архитекторов, которая используется для быстрого создания трёхмерных моделей предметов, конструкций, зданий и интерьеров. Благодаря интуитивному процессу работы, пользователь может воплотить свой замысел достаточно точно и графически понятно	<ul style="list-style-type: none"> • Полноценная возможность моделирования • Понятный и простой интерфейс • Мало занимает места (180 Мб) • Экспорт и импорт 3D-моделей • Рендеринг эффектов в виде набросков и тумана • Возможность установки плагинов 	<ul style="list-style-type: none"> • Очень ограниченные возможности моделирования поверхностей сложной кривизны («органики») • Отсутствие продвинутых инструментов текстурирования (UV-маппинга) сложных форм • Отсутствие управляемых дополнительных точечных (то есть кроме Солнца) источников света • Отсутствие механизмов фотореалистичного рендеринга, полноценных анимационных (презентационных) возможностей

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

Основная литература

1. Зиновьев, В. В. Моделирование процессов и систем: учеб. пособие / В. В. Зиновьев, А. Н. Стародубов, П. И. Николаев; КузГТУ. – Кемерово, 2016. – 143 с.

<http://library.kuzstu.ru/meto.php?n=91460&type=utchposob:common>

2. Зиновьев, В. В. Моделирование систем при помощи компьютерной имитации и анимации : учеб. пособие / В. В. Зиновьев, А. Н. Стародубов; КузГТУ. – Кемерово, 2010. – 118 с.

<http://library.kuzstu.ru/meto.php?n=90464&type=utchposob:common>

3. Полетаев, В. А. Автоматизация управления жизненным циклом продукции: учеб. пособие / В. А. Полетаев; КузГТУ. – Кемерово, 2014. – 125 с.

<http://library.kuzstu.ru/meto.php?n=90165&type=utchposob:common>

4. Полетаев, В. А. Проектирование компьютерно-интегрированных производственных систем учеб. пособие / В. А. Полетаев; КузГТУ. – Кемерово, 2015. – 200 с.

<http://library.kuzstu.ru/meto.php?n=91337&type=utchposob:common>

5. Проектирование компьютерно-интегрированных производственных систем / В. А. Полетаев, В. В. Зиновьев, А. Н. Стародубов, И. В. Чичерин; под ред. В. А. Полетаева. – Москва: Машиностроение, 2011. – 324 с.

6. Методология сквозного изготовления изделий машиностроения: в 2 т. Т. 2 / В. А. Полетаев, В. В. Зиновьев, А. Н. Стародубов, А. Н. Трусов; под ред. В. А. Полетаева; КузГТУ. – Кемерово, 2013. – 225 с.

7. Кобелев, Н. Б. Имитационное моделирование: учеб. пособие / Н. Б. Кобелев, В. А. Половников, В. В. Девятков; под общ. ред. Н. Б. Кобелева. – Москва: КУРС, НИЦ ИНФРА-М, 2013. – 368 с.

8. Девятков, В. В. Имитационные исследования в среде моделирования GPSS STUDIO: учеб. пособие / В. В. Девятков, Т. В. Девятков, М. В. Федотов. – Москва: ИНФРА-М, Вузовский учебник, 2018 – 283 с.

Дополнительная литература

9. Боресков, А. В. Компьютерная графика: учебник и практикум для прикладного бакалавриата / А. В. Боресков, Е. В. Шикин. – Москва: Юрайт, 2018. – 219 с. – (Серия: Бакалавр).

10. Аверилл, М. Лоу. Имитационное моделирование. Классика сс. – 3-е изд. / М. Лоу Аверилл, В. Дэвид Кельтон. – Санкт-Петербург: Питер; Киев: Издательская группа ВНУ, 2004. – 847 с.
11. Варжапетян, А. Г. Имитационное моделирование на GPSS/H: учеб. пособие; ГУАП. – Санкт-Петербург, 2007. – 384 с.
12. Зиновьев, В. В. Комплекс программ для поддержки принятия решений по выбору варианта автоматизированного производства / В. В. Зиновьев, А. Н. Стародубов // Вестник КузГТУ. – 2009. – № 3. – С. 135–140.
13. Конюх, В. Л. Дискретно-событийное моделирование подземных горных работ / В. Л. Конюх, В. В. Зиновьев. – Новосибирск: Изд-во СО РАН, 2011. – 243 с.
14. Коровин, А. М. Моделирование систем: учеб. пособие к лабораторным работам. – Челябинск: ЮУрГУ, 2010. – 47 с.
15. Королев, А. Г. Моделирование систем средствами GPSS – Future. Практический подход в примерах и задачах: учеб. пособие. Интернет-ресурс <http://objectgpss.narod.ru> – Украина, Северодонецк, 2009.
16. Исследование характеристик автоматизированных производственных систем с использованием имитационного моделирования / В. А. Полетаев, В. В. Зиновьев, А. Н. Стародубов, А. И. Цигельников // Вестник КузГТУ. – 2013. – № 2. – С. 74–76.
17. Алгоритм работы модуля оптимизации структуры автоматизированной производственной системы / В. А. Полетаев, А. И. Цигельников, В. В. Зиновьев, А. Н. Стародубов // Вестник КузГТУ. – 2011. – № 6. – С. 79–81.
18. Анимационное представление автоматизированных производственных систем (АПС) / В. А. Полетаев, В. В. Зиновьев, А. Н. Стародубов // Вестник КузГТУ. – 2010. – № 6. – С. 102–104.
19. Определение оптимального варианта автоматизированных производственных систем / В. А. Полетаев, В. В. Зиновьев, А. Н. Стародубов // Вестник КузГТУ. – 2010. – № 4. – С. 144–147.
20. Создание концепции имитационного моделирования автоматизированных производственных систем / В. А. Полетаев, В. В. Зиновьев, А. Н. Стародубов // Вестник КузГТУ. – 2010. – № 5. – С. 113–119.

21. Советов, Б. Я. Моделирование систем: учеб. для вузов / Б. Я. Советов, С. А. Яковлев. – 3-е изд., перераб. и доп. – Москва: Высш. шк., 2001. – 343 с.
22. Шеннон, Р. Имитационное моделирование систем – искусство и наука. – Москва: Мир, 1978. – 420 с.
23. Миронов, Д. Ф. Компьютерная графика в дизайне: учеб. для вузов / Д. Ф. Миронов. – Санкт-Петербург: БХВ-Петербург, 2008. – 560 с.
24. Мураховский, В. И. Компьютерная графика / В. И. Мураховский. – Москва: АСТ-Пресс, СКД, 2002. – 640 с.
25. Рейнбоу, В. Компьютерная графика. Энциклопедия / В. Рейнбоу. – Санкт-Петербург: Питер, 2003. – 768 с.
26. Петров, М. Н. Компьютерная графика: учеб. для вузов / М. Н. Петров, В. П. Молочков. – 2-е изд. – Санкт-Петербург: Питер, 2004. – 544 с.
27. Петров М. Н. Компьютерная графика: учебник для вузов. – 2-е изд. / М. Н. Петров, В. П. Молочков. – Санкт-Петербург: Питер, 2006. – 811 с.
28. Сайт 3D MAX: 3D MAX – Компьютерная графика и анимация в деталях [Электронный ресурс]. – Электрон. дан. – Б.м., 2009. – Режим доступа: <http://3dmax.ru>, свободный.
29. Уотролл, Э. Эффективная работа: Macromedia Flash MX / Э. Уотролл, Н. Гербер. – Санкт-Петербург, Киев: БХВ-Петербург, 2003. – 720 с.
30. Чепмен, Н. Цифровые технологии мультимедиа / Н. Чепмен, Д. Чепмен. – 2-е изд. – Москва: Вильямс, 2006. – 624 с.
31. Никулин, Е. А. Компьютерная геометрия и алгоритмы машинной графики / Е. А. Никулин. – Санкт-Петербург: БХВ-Петербург, 2003. – 560 с.
32. Капранова, М. Н. Macromedia Flash MX. Компьютерная графика и анимация / М. Н. Капранова. – Москва: СОЛОН-ПРЕСС, 2010. – 96 с.: ил. – («Элективный курс» профильное обучение).
33. Платонова Н. С. Создание компьютерной анимации в Adobe Flash CS3 Professional [Электронный ресурс] – Электрон. текстовые данные. – Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 175 с.
34. Классическая анимация: нарисованное движение: учеб. пособие / А. А. Петров. – Москва: ВГИК, 2009. – 197 с.

Стародубов Алексей Николаевич
Зиновьев Василий Валентинович
Кузнецов Игорь Сергеевич

**КОМПЬЮТЕРНАЯ ИМИТАЦИЯ И АНИМАЦИЯ
ПРОИЗВОДСТВЕННЫХ СИСТЕМ**

Учебное пособие

Печатается в авторской редакции

Подписано в печать 16.04.2018. Формат 60×84/16
Бумага офсетная. Гарнитура «Times New Roman». Уч.-изд. л. 5,8
Тираж 100 экз. Заказ
КузГТУ, 650000, Кемерово, ул. Весенняя, 28
Издательский центр УИП КузГТУ, 650000, Кемерово, ул. Д. Бедного, 4а