

## Практическая работа № 31. Создание анимации в программе Macromedia Flash (работа с объектом).

**1. Цель работы:** освоение приемов создания анимации в программе Macromedia Flash.

**2. Оборудование, приборы, аппаратура, материалы:** персональный компьютер, программа Macromedia Flash.

**3. Краткие теоретические сведения.**

Macromedia Flash MX 2004 предоставляет несколько способов создания анимационных последовательностей:

- анимационные эффекты — программа сама создает последовательность кадров, имитирующую тот или иной эффект применительно к некоторому объекту;
- покадровая анимация — пользователь создает каждый кадр будущей анимации;
- автоматическая tweened-анимация, или анимация трансформации, — пользователь задает начальный и конечный кадр, а программа сама создает промежуточные кадры на основе программной интерполяции.

### 1. Задание

#### Покадровая анимация

Рассмотрим простейший пример — листочек перемещается из одной точки экрана в другую с поворотом вокруг своей оси.

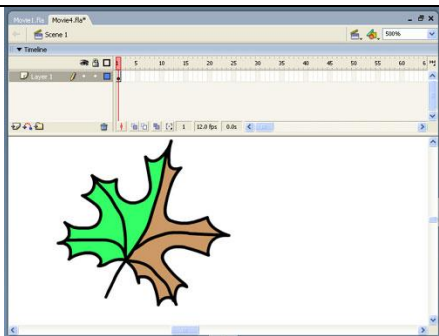


Рис. 6. Первый ключевой кадр анимации

Нарисуем кленовый листочек, например такой, как показано на рис. 6, — соответствующий кадр на панели **Timeline** окрасится в серый цвет и внутри него появится точка, указывающая на то, что это ключевой кадр. Ключевой кадр — это кадр, в котором происходит помещение содержимого или его изменение.

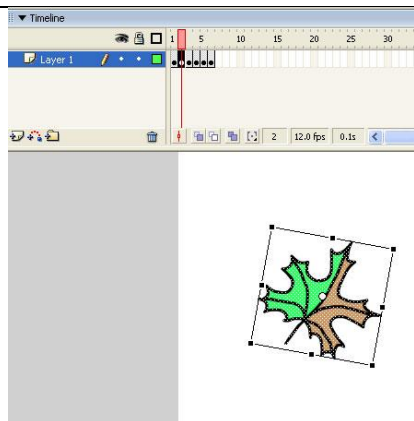


Рис. 7. Второй кадр создадим перетаскиванием и трансформацией первого

Затем щелкнем правой кнопкой мыши по соседнему кадру и вставим еще один ключевой кадр, используя команду **Insert Keyframe**. В результате в этом кадре появится копия листочка; переместим ее вниз (пользуясь инструментом Arrow) и повернем, используя команду **Modify => Transform => Free Transform** (рис. 7).

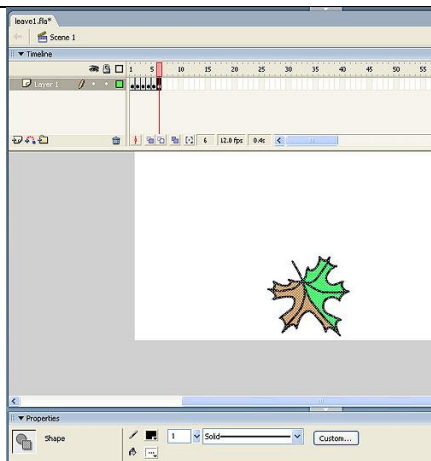


Рис. 8. Последний кадр анимации

Повторим процедуру таким образом, чтобы в 6-м кадре лист занимал конечное положение (рис. 8).

Обратите внимание на панель **Properties** (рис. 8) — в левой ее части указывается тип объекта. В каждом кадре листок является объектом типа **Shape** (форма), о других типах объектов будет рассказано чуть позже.

Для того чтобы экспортировать фильм в виде SEF-файла (родной формат Macromedia для Flash-фильмов), выполним команду **File => Export => Export Movie**. В результате получим следующий фильм. Просмотреть получившийся фильм можно не покидая программы Flash по команде **Control => Test Movie**.

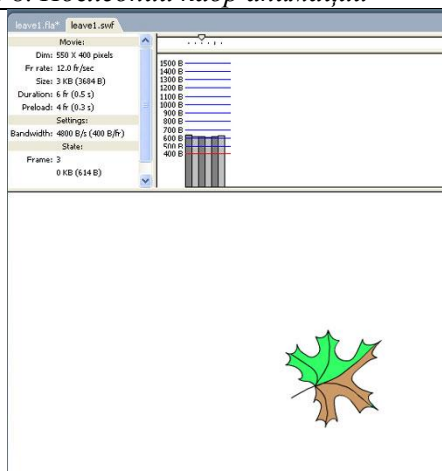


Рис. 9. Просмотр фильма в режиме **Bandwidth Profiler**

Причем для того, чтобы просмотреть, какой объем занимают отдельные кадры фильма, следует выполнить команду . В результате мы увидим, что каждый из шести кадров занимает около 600 Кбайт (рис. 9). Таким образом, объем всего фильма составляет 3686 байт.

### Автоматическая анимация

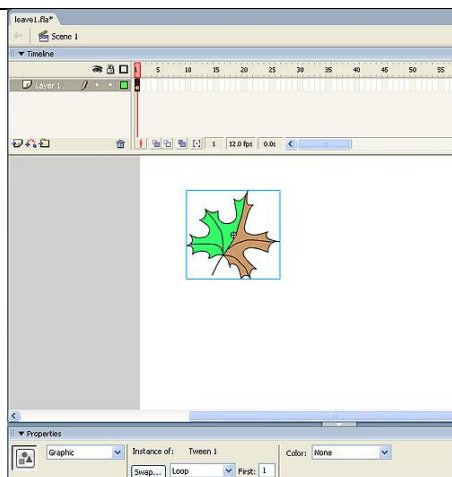


Рис. 10. Объект автоматически превращается в графический символ

Рассмотрим, как можно сделать ту же анимацию и получить более компактный результирующий файл. Выделим на экране нарисованный листок инструментом **Arrow** и выполним команду **Insert => Timeline => Create Motion Tween**, в результате листочек будет помещен в рамочку, а в панели **Properties** появится сообщение, что выделенный объект имеет свойства **Graphic** (рис. 10). Это означает, что анимируемый объект автоматически преобразован в графический символ. Теперь его уже невозможно произвольно редактировать инструментом **Arrow** как объект типа **Shape**.

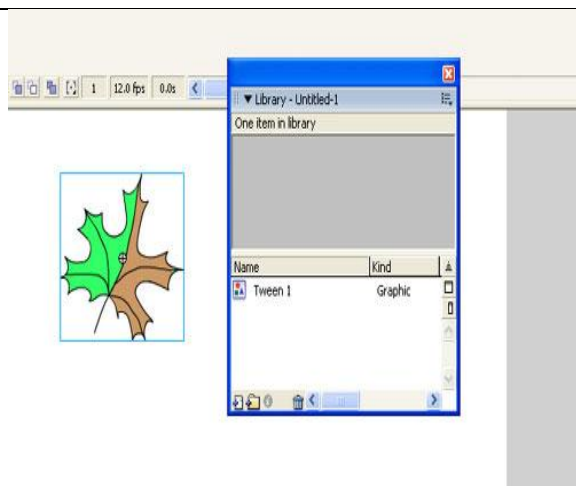


Рис. 11. Каждый новый символ становится частью библиотеки

Использование символов является важным понятием во Flash. Однажды создав символ, его можно использовать несколько раз в фильме, не увеличивая размер результирующего файла. Символы делятся на графические (graphic), символы-кнопки (button) и символы-мувклипы (movie clip). В данном уроке рассмотрим графический символ, а к другим типам символов вернемся позднее. Каждый новый символ сразу становится частью библиотеки текущего документа (рис. 11). Если выполнить команду **Window => Library**, то можно убедиться, что в библиотеке появился символ и ему по умолчанию присвоено имя Tween 1. Для того чтобы присвоить символу другое имя, достаточно дважды щелкнуть по названию и заменить его на желаемое. После того как мы сформировали графический символ, перейдем в конечный кадр нашей анимации (пусть это будет 15-й кадр) и вставим ключевой кадр (по команде **Insert Keyframe**). В этом кадре появится копия символа, которую мы переместим и повернем вокруг оси (по команде **Modify Transform=>Free Transform**), как в предыдущем примере.

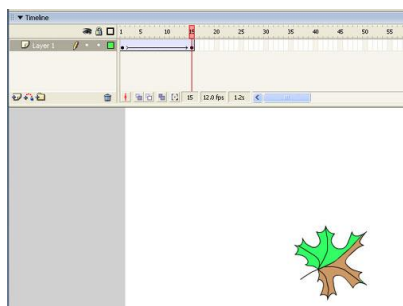


Рис. 12. Стрелка на голубом фоне свидетельствует о создании анимации **Motion Tween**

Как видно из рис. 12, все кадры между двумя ключевыми окрасились в голубой цвет и от первого ключевого кадра к последнему протянулась стрелка, что указывает на создание анимации **Motion Tween**.

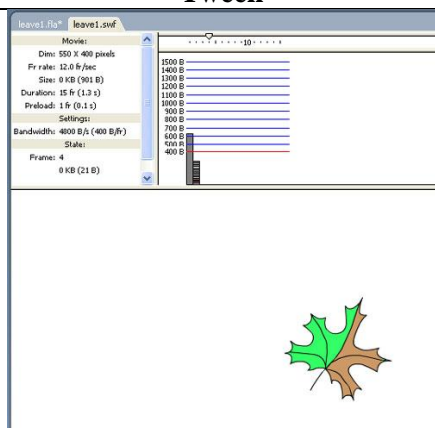


Рис. 13. Просмотр фильма в режиме **Bandwidth Profiler**

Выполнив команду **Control => Test Movie**, получим информацию, представленную на рис. 13.

Несмотря на то что в этом примере у нас 15, а не шесть кадров, как в предыдущем, и анимация получилась более плавная, размер результирующего файла оказывается меньше — всего 900 байт. Как видно из диаграммы (рис. 13), информация об объекте хранится только в первом кадре, а в каждом новом кадре необходимо запоминать лишь новые положения листка. На это уходит в среднем всего по 20 байт.

Для того чтобы проиллюстрировать падение листа с поворотом вокруг плоскости листа, повторим предыдущий пример, только при модификации последнего ключевого кадра добавим команду **Modify => Transform => Flip Horizontal**. В результате получим следующий **фильм**.

Теперь рассмотрим пример, когда листочек приближается к зрителю. Для этого в конечном кадре вместо зеркального отображения (**Flip Horizontal**) будем увеличивать размер листка. Для того чтобы не возникало ощущения замедления движения при приближении объекта к зрителю, скорость его движения необходимо увеличивать. Для того чтобы добиться этого эффекта, необходимо щелкнуть по первому кадру и обратиться к разделу **Ease** в панели **Properties**. Положительные значения параметра Ease приводят к замедлению движения, а отрицательные — к ускорению. Выберем максимальное ускорение объекта.

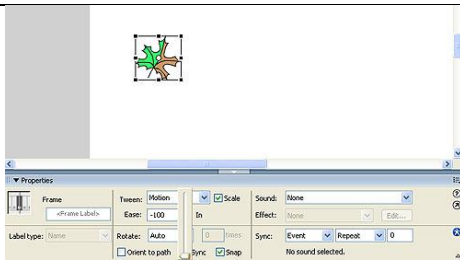


Рис. 14. Выберем максимальное ускорение объекта  $Ease = -100$

Для этого выставим значение параметра **Ease** равным  $-100$  (рис. 14). В результате получим **фильм**. Отметим, что первый кадр можно поместить и за сценой, тогда мы получим фильм, в котором листок будет влетать в кадр и двигаться навстречу зрителю. Можно моделировать вращение листа вокруг смещенного центра симметрии. Надеюсь, читатель сам сможет поэкспериментировать, усложняя модификацию конечного кадра и изменяя таким образом характер движения листа.

Из представленных примеров очевидно, что автоматическая анимация движения эффективна, когда трансформация объекта при его движении задается простыми функциями (поворот, масштабирование и т.п.). Если же необходимо анимировать сложные движения (например, движение руки героя мультфильма), то здесь анимация трансформации движения не применима. Каждый кадр приходится рисовать вручную, то есть применять покадровую анимацию, состоящую из набора ключевых кадров. Таким образом, покадровая анимация — это наиболее универсальный, но вместе с тем и самый трудоемкий вид анимации, кроме того, он создает наиболее «тяжелые» файлы. Всегда, когда можно заменить покадровую анимацию автоматической, — это предпочтительно. Рассмотрим ряд примеров, которые позволяют использовать автоматическую анимацию движения при имитации полета.

### Движение по заданной траектории

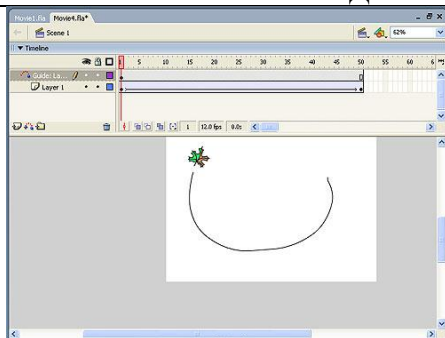


Рис. 15. Пример задания траектории движения

Flash позволяет задать движение объекта вдоль заданной траектории. Для того чтобы задать эту траекторию, выполните команду **Insert => Timeline => Motion Guide**.

В результате над текущим слоем появится специальный слой, который по умолчанию будет иметь имя **Guide Layer 1**.

Щелкнем мышью по слою траектории и с помощью инструмента «карандаш» нарисуем линию, вдоль которой планируется перемещение листка (рис. 15).

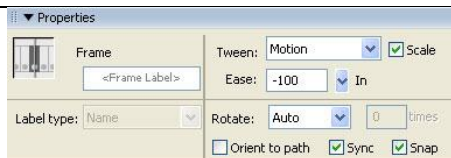


Рис. 16. Параметр **Snap** задает режим привязки к траектории движения

Теперь перейдем в первый кадр (щелкнем по нему мышью) и в появившейся панели **Properties** установим флажок **Snap** (задает режим привязки к траектории движения) — рис. 16.

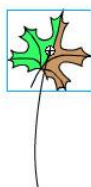


Рис. 17. Объект как бы прилипает к траектории движения своим центром

После того как вы поставите флажок **Snap**, центр листа совместится с траекторией движения. Пользуясь инструментом **Arrow**, можно переместить листочек вдоль кривой движения, но если вы попытаетесь оторвать листок от траектории и расположить его рядом с ней, он будет притягиваться назад и вновь «прилипает» к траектории движения (рис. 17).

Перейдите на последний кадр и аналогично привяжите листок к конечной точке траектории движения. Для придания фильму объемности добавим трансформацию **Flip Horizontal**, — в результате получим **фильм**.



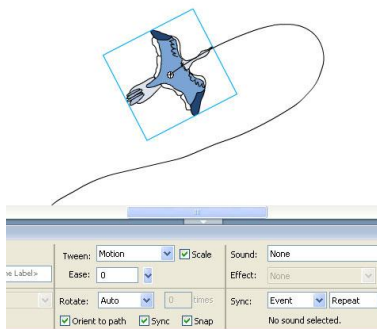


Рис. 18. Если поставить флажок **Orient to Path**, птица будет лететь головой вперед

При движении листа вдоль траектории нам не важно, как он будет повернут в направлении движения. Но если мы аналогичным образом задаем траекторию полета птицы, то на некоторых участках кривой окажется, что птица летит хвостом **вперед**.

Очевидно, что если мы хотим анимировать полет самолета или птицы, то нам нужно, чтобы они все время двигались носом вперед. Во Flash такой характер движения задать очень просто (рис. 18).

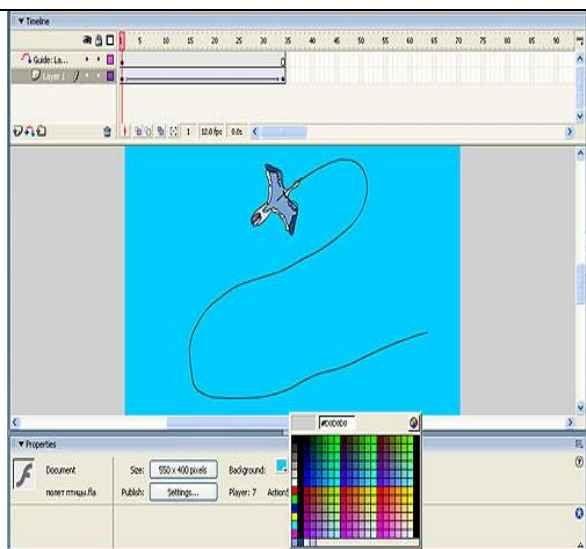


Рис. 19. Добавление одноцветного фона

Необходимо поставить флажок **Orient to Path**, и движение птицы изменится на вполне привычное (источник к данному ролику — [полет птицы.flv](#)).

Если вы хотите добавить одноцветный фон, щелкните мышью по фону и в появившейся панели **Properties** (рис. 19) в поле **Background** выберите необходимый цвет фона.

Если мы хотим добавить фоновый рисунок, нам понадобится для этого отдельный слой. В принципе, программа Flash предлагает возможность создания системы слоев, сходной с теми, что используются в классической анимации, при которой фон и различные подвижные объекты рисуются каждый на своем слое прозрачной пленки.

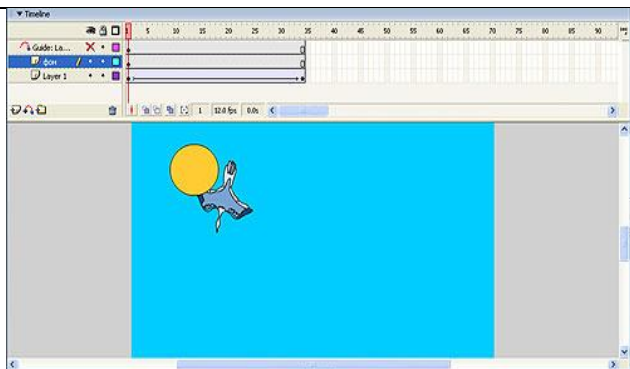


Рис. 20. На новом слое создадим неподвижный фоновый объект

Расположив фоновое изображение и каждый анимируемый объект на своем слое, добиться контроля над ними гораздо проще. Итак, для того чтобы добавить фоновый рисунок, создадим для него новый слой. Для этого щелкнем правой кнопкой мыши по слою, который на рис. 19 обозначен как layer 1, и в выпадающем меню выберем строку **Insert layer**. В добавленном слое нарисуем солнце. Чтобы не запутаться в номерах слоев, дадим слою название «фон». Для этого необходимо щелкнуть на текущем названии и ввести необходимое имя (рис. 20).

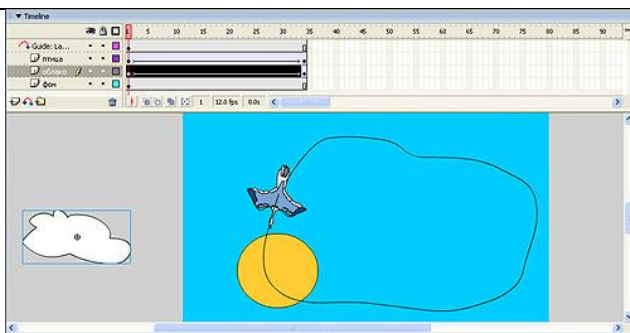


Рис. 21. Для анимации облака создадим отдельный

Как видно из рис. 20, птица находится за солнцем, что противоречит здравому смыслу. Для того чтобы поменять местами слои, достаточно в режиме **drag-and-drop** перетащить слой с именем «фон» вниз.

Подредактируем объекты фильма (для того чтобы птица кружила на фоне солнца, изменим траекторию ее полета и поменяем соотношения размеров солнца и птицы) и получим следующий **ролик**.

Теперь добавим к нашей анимации подвижные объекты, например облако. Для облака создадим новый слой и на нем зададим анимацию

<i>слой</i>	трансформации движения. Для того чтобы облако влетало в сцену, расположим его, как показано на рис. 21. В результате получим следующий <b>фильм</b> (источник к данному фильму находится в прилагаемом файле <b><u>полет птицы-с облаком.fla</u></b> ).
-------------	---

## **5. Содержание отчета**

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и его решение.
4. Вывод по работе.

## **6. Контрольные вопросы**

1. Что такое ключевой кадр? Как вставить ключевой кадр?
2. Как сделать покадровую анимацию?
3. Как просмотреть созданный ролик?
4. Как сохранить анимацию в формате .swf?
5. Как создать автоматическую анимацию?
6. Как создать движение по заданной траектории?