

# КОНСТРУКТОР ИГР CLICKTEAM FUSION

С. Л. Углев

12+



КАК БЫСТРО СОЗДАТЬ СВОЮ ИГРУ

Семён Углев

# **Конструктор игр Clickteam Fusion**

«Автор»

2018

**Углев С. Л.**

Конструктор игр Clickteam Fusion / С. Л. Углев — «Автор», 2018

ISBN 978-5-532-96567-6

Книга была создана в рамках проекта "Clickteam по-русски". Автор выражает надежду, что новая книга оправдает себя, облегчит работу и поможет творчеству читателя в таком непростом деле.

ISBN 978-5-532-96567-6

© Углев С. Л., 2018

© Автор, 2018

# Содержание

Предисловие	5
Часть I. Введение в программу	6
Урок 1. Интерфейс программы	6
Урок 2. "Движок" для игры	19
Урок 3. Стартовое меню	29
Урок 4. Инвентарь в игре	34
Часть II. Функционал игр	48
Урок 5. Приближение в играх	48
Конец ознакомительного фрагмента.	52

# Семён Углев

## Конструктор игр Clickteam Fusion

### Предисловие

Эти уроки были созданы в период с 2013 по 2018 год. Одной из первых моих работ был урок по программе Multimedia Fusion, которая является предшественником Clickteam Fusion 2.5. Затем потребовался урок об основах Clickteam, в частности, об интерфейсе программы. Следом было выпущено ещё несколько уроков. Так и родилась эта книга.

В данной книге уроки идут в последовательности, максимально понятной для изложения, поскольку создание уроков – процесс творческий и очень сложный. Безусловно, для этого требовалось вдохновение, поэтому до создания следующего урока шли дни, недели, а иногда и месяцы. В последнее время ассортимент литературы и видеообзоров был расширен и появилось несколько обзоров популярных игр, когда-либо выпущенных на движке Clickteam.

Разумеется, за 5 лет создания уроков многое изменилось. В основном, менялся административный веб-интерфейс Google Play, AppStore, Windows Phone Store, но даже с этими изменениями всё остаётся доступным и понятным. При создании роликов я пользовался улучшенной модификацией программы Clickteam Fusion Developer, но старался, чтобы все функции работали и в базовой версии. Если что-то не работает, можно обратиться непосредственно к готовым файлам, созданным в ходе урока и опубликованным на сайте издательства.

Первоначально мои уроки были ориентированы на людей моего склада: быстро заглянул, промотал, что-то подсмотрел и начал работу. Однако, оказалось, что есть вдумчивые, серьёзные люди – быть может, никогда в жизни не сталкивающиеся с программированием. Таким образом и родилась эта книга.

Честно признаюсь, несмотря на простоту использования данной программы и достаточно широкий функционал, я рекомендую всё же ознакомиться с основами программирования. Это очень поможет в случае проектирования структуры игры, поможет избежать многих ошибок и значительно упростит её алгоритм. Впрочем, если для читателя главное – что-то творить для других, хватит и информации в самой книге. Обращаю внимание, что эта книга посвящена созданию 2D-игр и не включает в себя уроки создания изображений и анимаций в различных графических программах. В книге использованы скриншоты программ Clickteam Fusion, Far Manager, скриншоты игр, разработанных автором: «Dizzy One», «Dizzy Two», «Caramba», «Ligo-Lambi», личные арты и фотографии, а также иные изображения, авторство которых указано в тексте отдельно.

В завершение хотелось сказать, что данная книга – не первая у автора на тему программирования, до этого вышли "Уроки ассемблера", включающие в себя 20 уроков для DOS, а также множество статей, посвящённых программированию и системному администрированию. Выражаю надежду, что новая книга оправдает себя, облегчит работу и поможет творчеству читателя.

*Автор*

## Часть I. Введение в программу

### Урок 1. Интерфейс программы

Это наш самый первый урок. Мы познакомимся с интерфейсом программы, рассмотрим основные возможности и даже напишем свою первую мини-игру. Для этого откроем нашу программу и нажмём сочетание клавиш **Ctrl-N** (создание нового документа).



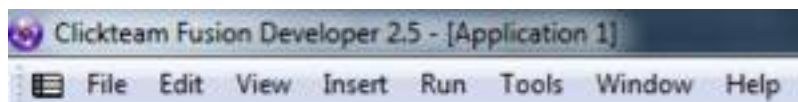
Окно программы можно условно разделить на несколько частей. Первая – это меню, находящееся вверху. Вторая строка, чуть ниже, на которой находятся иконки – меню быстрого доступа, представляющее собой графическое представление некоторых возможностей программы. На третьей строчке расположен ползунок масштаба и иконки визуального представления фрейма, находящегося в правой части экрана.

Боковое меню, как мы видим, разбито на две части, верхнюю и нижнюю. В верхней части расположена панель **Workspace Toolbar**, которая предоставляет быстрый доступ к настройкам приложения, фреймов и элементов, находящихся в нём. Нижнюю часть называют **Properties Toolbar**, и она пока недоступна. Информация в ней будет размещена при выборе элементов рабочей панели или элементов фрейма.

В центральной части находится большое рабочее окно, которое можно переключать в разные режимы. Сейчас мы видим список фреймов (пока он у нас один) и основные настройки. Благодаря ему мы можем переходить в выбранные нами фрейм, а также в редактор событий, о котором расскажем позже.

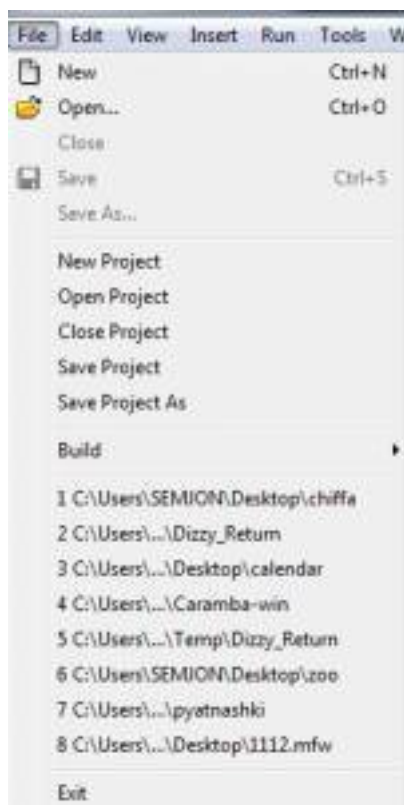
В правом краю экрана расположена малозаметная панель слоёв, о которой мы подробно расскажем в процессе создания игр.

Итак, рассмотрим верхнее меню.



В крайней правой части находится иконка свойств самой программы (заккрыть, развернуть и пр.). Её можно скрыть, дважды кликнув по ней. За ней идут элементы меню, способные раскрываться, предоставляя доступ к различным функциям.

Первое меню – **File** (работа с файлами).



Структура выпадающего меню такова. Слева находится иконка данного пункта меню – если она предусмотрена и находится в строке ниже. Правее – текстовое описание действия пункта. В правом краю находятся «горячие» клавиши, благодаря которым можно совершить нужные действия, не используя выпадающее меню.

Первое меню представляет возможности:

**New (Ctrl-N)** – создать новый документ с чистого листа;

**Open (Ctrl-O)** – открыть новый документ;

**Close** – закрыть документ;

**Save (Ctrl-S)** – сохранить документ (чуть ниже находится пункт Save as... «Сохранить как», предусматривающий возможность сохранения в нужном месте);

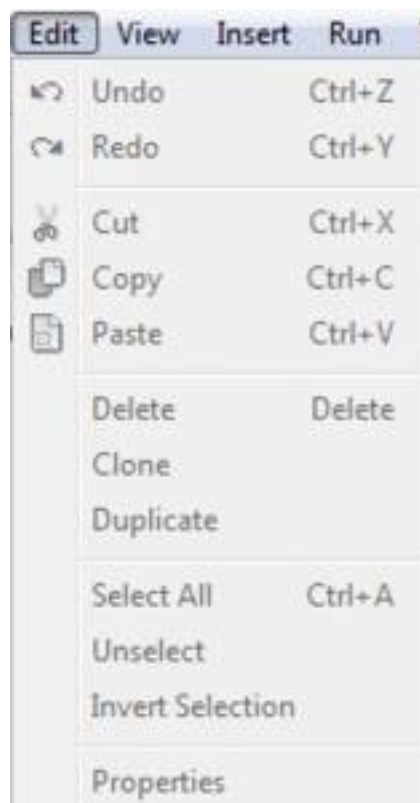
**New Project** – новый проект;

Другие действия с проектом (Open Project, Close Project, Save Project, Save Project as... – аналогично документам – «открыть», «закрыть», «сохранить» и «сохранить как»);

**Build** – собрать. Собрать проект можно в виде готового приложения или программы-инсталлятора (а также и того, и другого). В зависимости от имеющихся экспортёров, мы можем собирать проекты для разных операционных систем;

Ниже находится перечень последних открытых проектов, а в самом низу – возможность выхода из программы (Exit).

Второе меню – **Edit** (редактирование).



Сейчас все пункты неактивны, поскольку документ только что открыт, и мы не производили над ним никаких действий. Тем не менее вот перечень пунктов:

**Undo (Ctrl-Z)** – отменить последнюю операцию;

**Redo (Ctrl-Y)** – повторить последнюю операцию;

**Cut (Ctrl-X)** – вырезать элемент (с возможностью последующей вставки с удалением элемента);

**Copy (Ctrl-C)** – копировать элемент (с возможностью последующей вставки без удаления данного элемента);

**Paste (Ctrl-V)** – вставить элемент;

**Delete (Delete)** – удалить элемент;

**Clone** – клонировать элемент. При этом будет создан элемент с новым именем (к наименованию добавляется цифра), с которым можно проводить независимые операции в редакторе событий;

**Duplicate** – дублировать элемент. Элемент с новым именем программа не создаёт, в редакторе событий он остаётся один. При этом на экране размещены для одинаковых элемента с одинаковыми именами и параметрами;

**Select All (Ctrl-A)** – выделить всё;

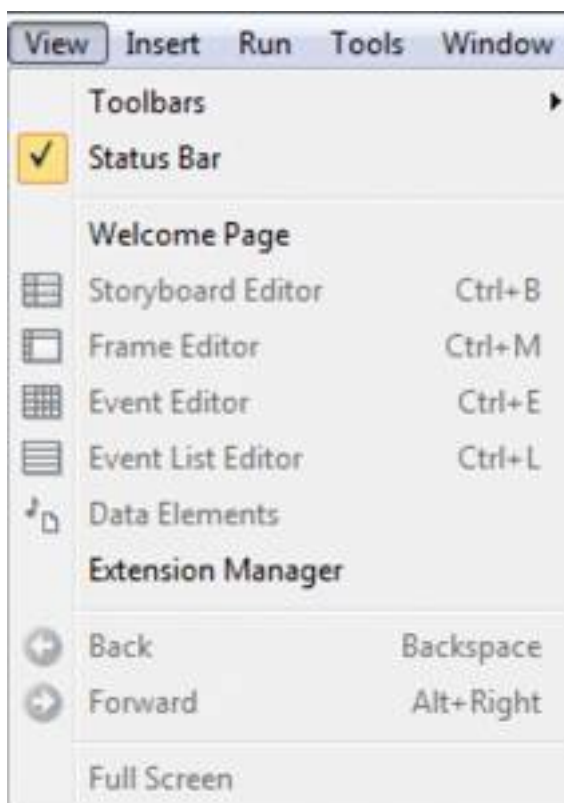
**Unselect** – сбросить выделение;

**Invert Selection** – инвертировать выделение;

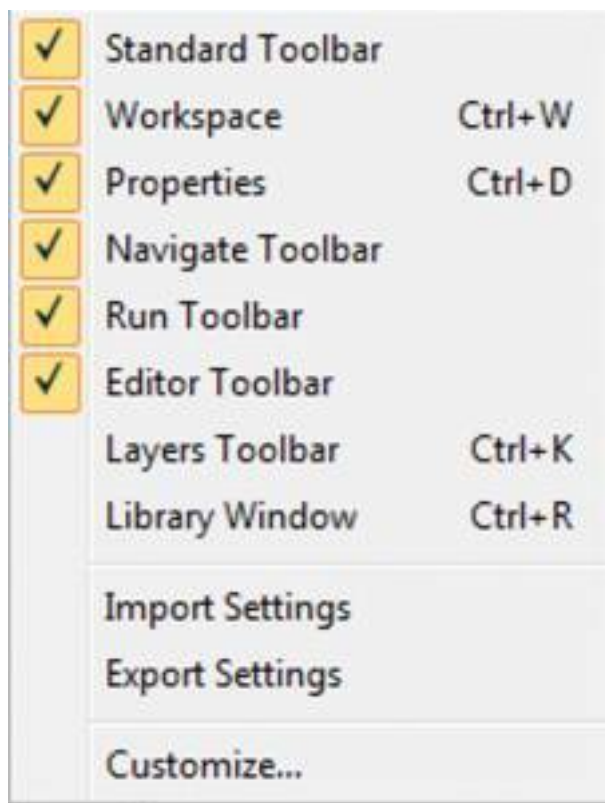
**Properties** – открыть окно настроек.

Третье меню – **View** (просмотр).





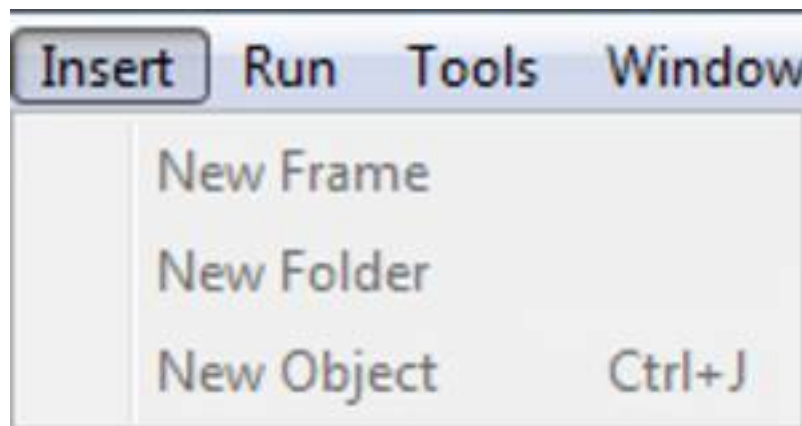
И сразу же первый пункт меню, **Toolbars**, предоставляет нам возможность открытия ещё одного пункта меню:



Эти инструменты позволяют нам открывать или закрывать на основном экране панели инструментов, предназначенные для быстрого доступа. Попробуйте снять или установить галочку с любого из них, и вы увидите, как появляется или исчезает та или иная панель. Можно настроить программу так, чтобы на экране были расположены только те панели инструментов, с которыми проходит активная работа.

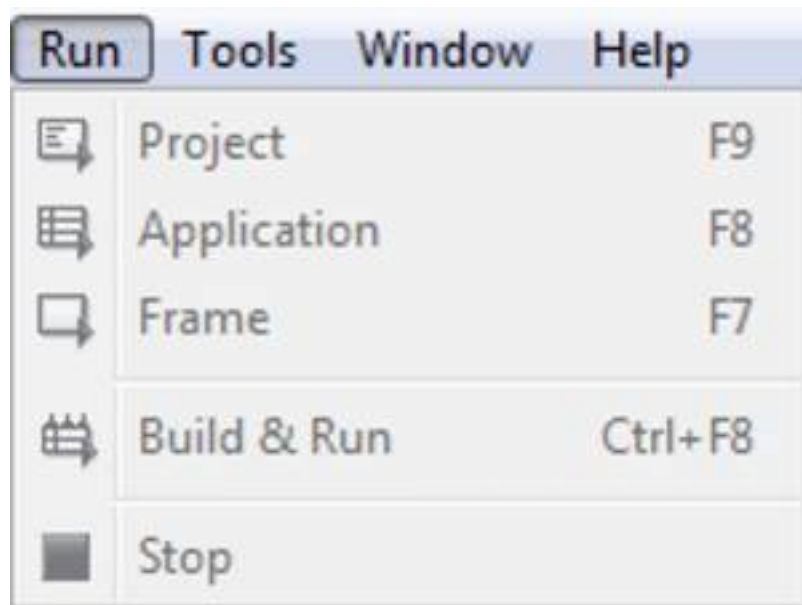
Также, с помощью пунктов **Import Settings** и **Export Settings** можно сохранить и при необходимости восстановить сделанные настройки. Пункт меню **Customize** даёт нам возможность расширенных настроек каждой панели инструментов, вплоть до добавления и удаления любого пункта.

Четвёртое меню – **Insert** (вставка).



Меню позволяет вставить новый фрейм (**New Frame**), новую папку (**New Folder**), новый объект (**New Object Ctrl-J**) соответственно.

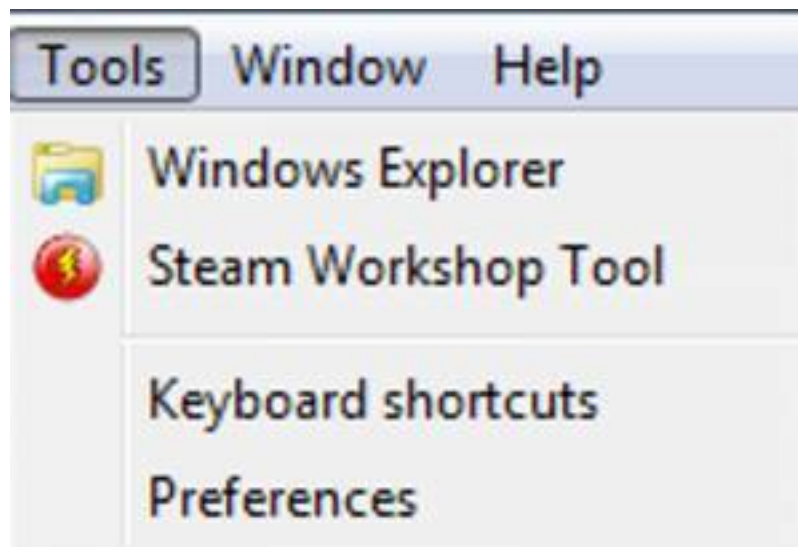
Пятое меню – **Run** (запуск).



Меню Run позволяет запускать весь проект, отдельное приложение и отдельный фрейм. При этом надо отметить, что при запуске одного фрейма по кнопке F7 будет запущен непосредственно выбранный фрейм, а в случае наличия в нём инструкций по переходу на другой фрейм он будет просто закрывать приложение.

Пункт **Build & Run** Ctrl-F8 позволяет собрать и запустить наше приложение, а пункт **Stop** – остановить его работу.

Шестое меню – **Tools** (утилиты).



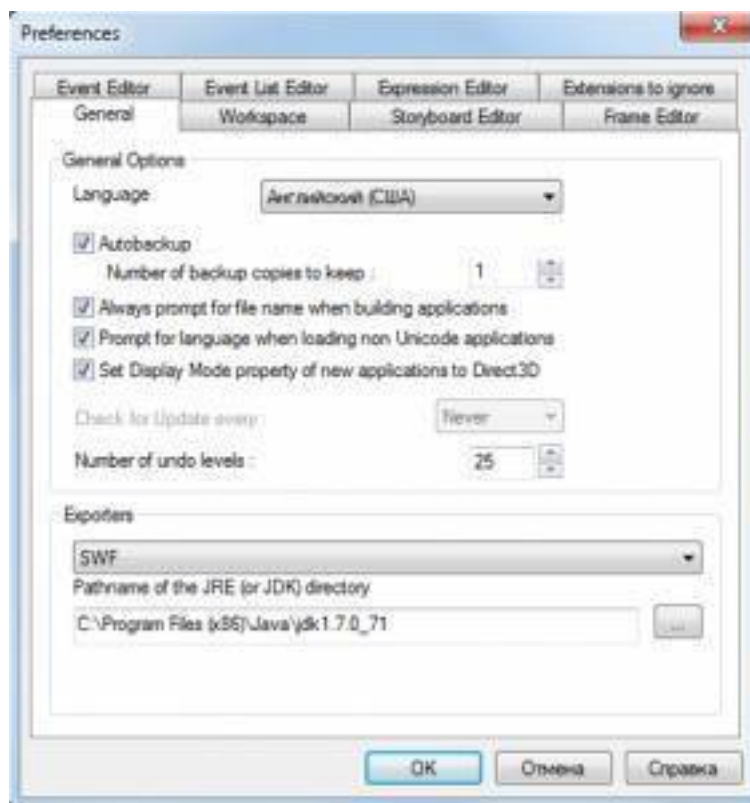
**Windows Explorer** – открывает в проводнике Windows папку установки программы Clickteam Fusion;

**Steam Workshop Tool** – предоставляет доступ к хранилищу различных приложений (как правило, в формате \*.mfa), которые можно скачивать и открывать программой Clickteam Fusion;

**Keyboard shortcuts** – клавиатурные ярлыки – открывает окно «Помощь по клавиатуре», где можно посмотреть те или иные сочетания клавиш;

**Preferences** – предпочтения – открывается окно настроек программы, состоящее из нескольких вкладок.

В последнем пункте мы рассмотрим самое главное.

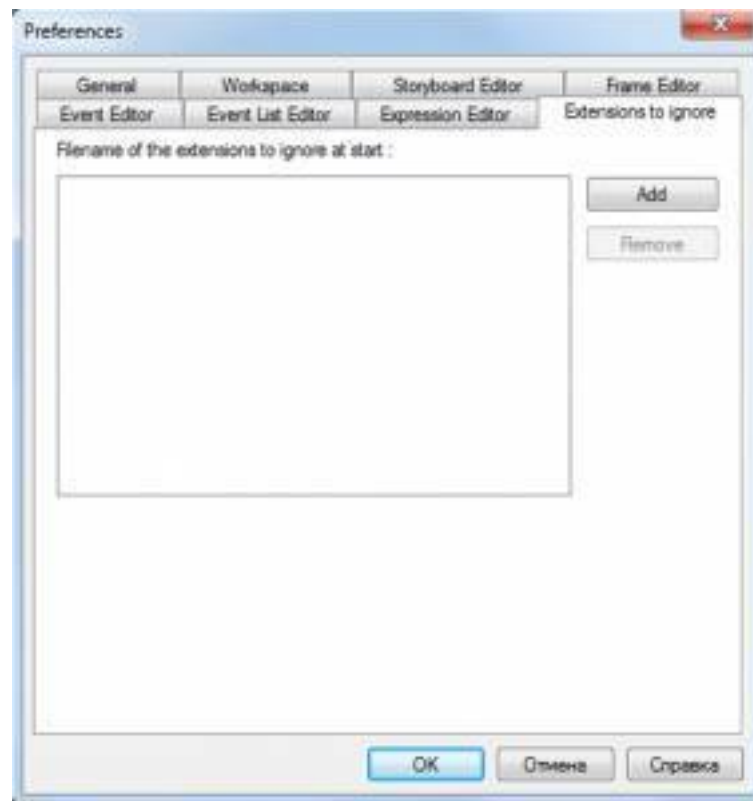


Во вкладке **General** – основные настройки для нас важно меню выбора языка (требуется перезапуск) и нижнее меню **Exporters** – настройки экспортёров. В первом его пункте, **Android**, нам необходимо установить и указать путь к директории JDK (скачать можно по ссылке [www.oracle.com/technetwork/java/javase/downloads/](http://www.oracle.com/technetwork/java/javase/downloads/)), а также Android SDK ([developer.android.com/intl/ru/sdk/index.html](http://developer.android.com/intl/ru/sdk/index.html)).

Во втором пункте, **BlackBerry Java**, нужно установить и указать путь к BlackBerry JDE.

Третий пункт, **HTML5**, равно как и четвёртый, **SWF**, просят нас указать путь к директории JDK, о котором было написано в первом пункте.

Ещё одна немаловажная вкладка – это **Extensions to ignore**.

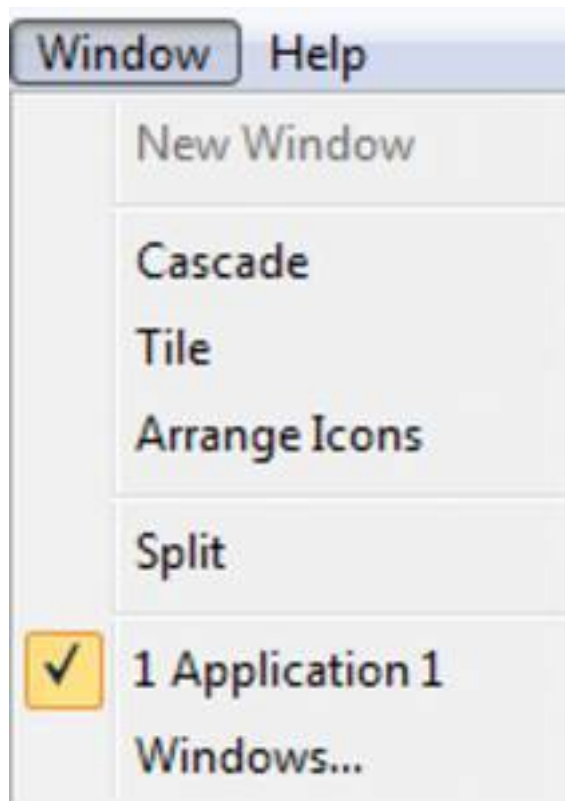


Иногда так случается, что при открытии скачанного проекта мы наблюдаем такое окно:



Пока в данном случае не будет установлено расширение **Control X** (путём нажатия на кнопку **Extension Manager** и выбора расширения), мы не сможем открыть проект. В данном случае мы можем помочь этому, добавив расширение **ctrlx.mfx** во вкладку **Extensions to ignore**. При этом нужно помнить, что часть функционала, связанная с этим расширением, будет потеряна.

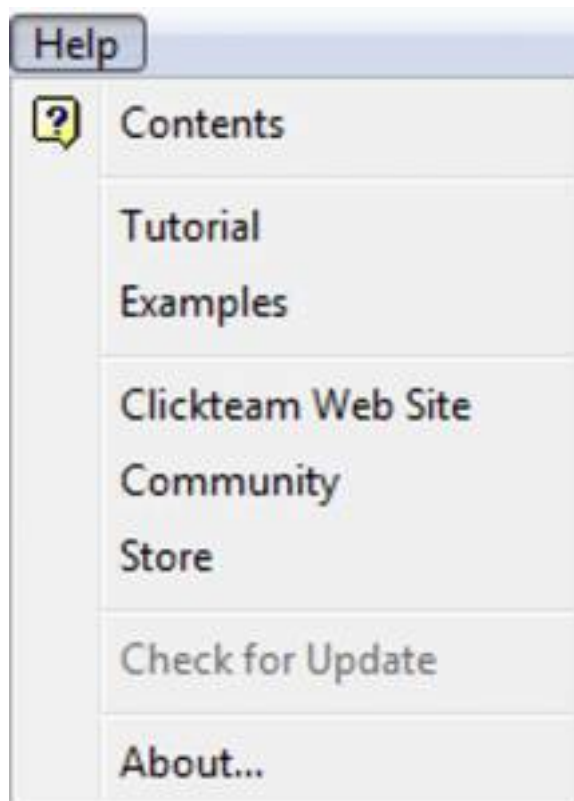
Седьмое меню – **Window** (окно).



Его задачи: открыть новое окно (**New Window**), расположить окна каскадом (**Cascade**), выделить окно фреймов отдельным окном (**Tile**), расположить иконки (**Arrange Icons**) или разделить фрейм на несколько частей (**Split**).

С помощью данного меню можно выбрать текущим новый документ из списка, либо с помощью выбора меню **Windows...**

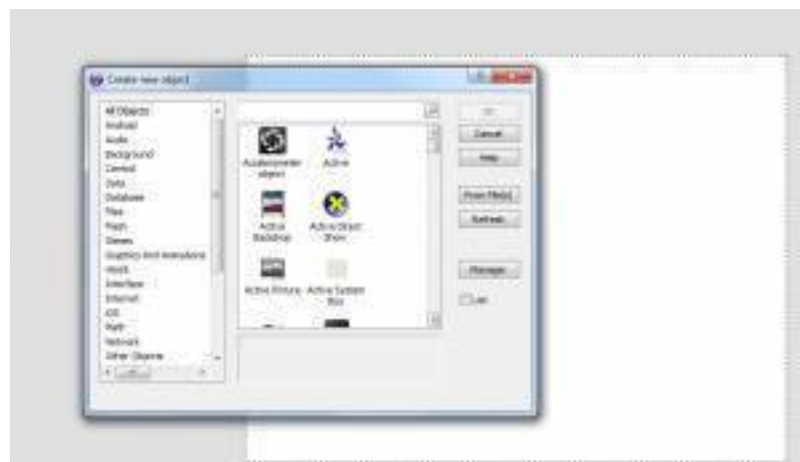
И последнее, восьмое меню, отвечает за помощь в работе с программой (совсем как этот урок).



Это собственно доступ к справке Windows (**Contents**), справочник по примерам (**Tutorial**), сами примеры (будет открыт проводник Windows), переход на официальный сайт, форум, магазин, проверка обновлений, а также окно «О программе» (**About**). На этом мы закончим знакомство с основными элементами интерфейса и будем возвращаться или заново изучать их в других уроках.

Ну и на последнем небольшом примере можно создать небольшую игру, чтобы понять, как работает программа.

Дважды кликнем по нашему фрейму и дважды щёлкнем левой кнопкой мыши по любому месту в белом прямоугольнике.





Перед нами будет открыто окно создания нового объекта, чем мы и воспользуемся. Дважды кликнем по объекту **Active**, и появится перекрестие, при котором, при нажатии на левую кнопку мыши, объект будет размещён в нашем фрейме.

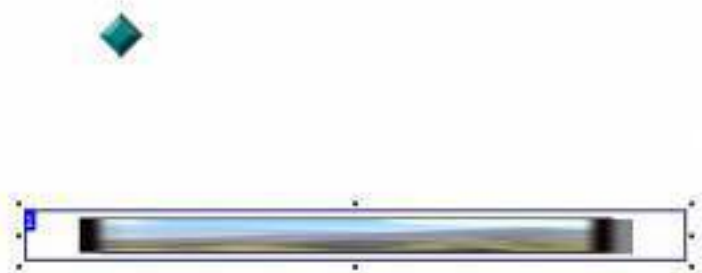
Снова аналогичным образом воспользуемся этим окном, выбрав слева вкладку **Background**, и разместим в нашем фрейме объект **Backdrop** на высоте ниже нашего объекта.



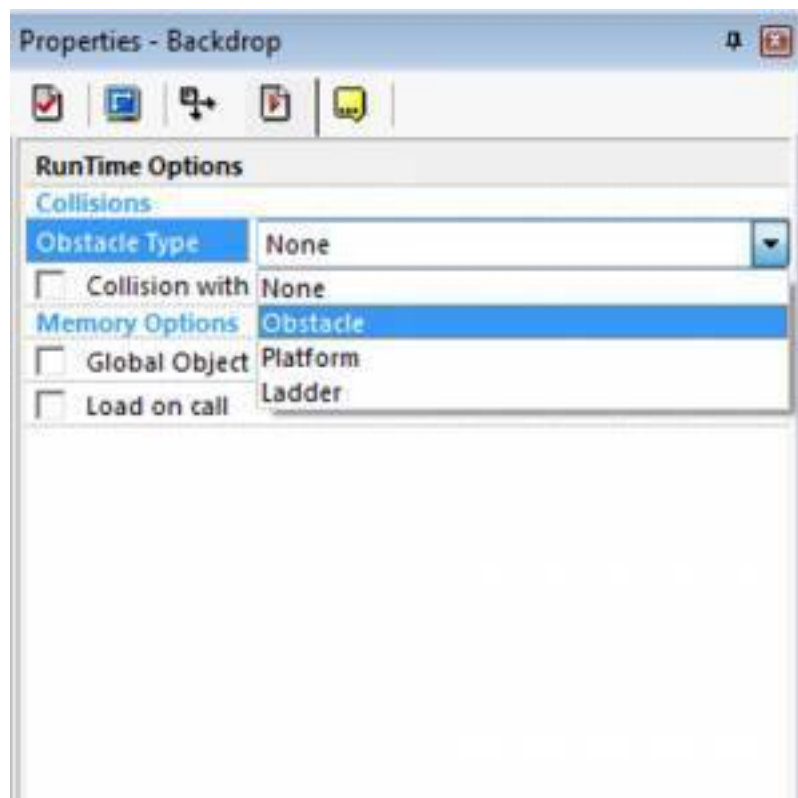
Щёлкнем левой кнопкой мыши по нему один раз и потянем вправо за правую точку, растягивая объект на ширину фрейма.



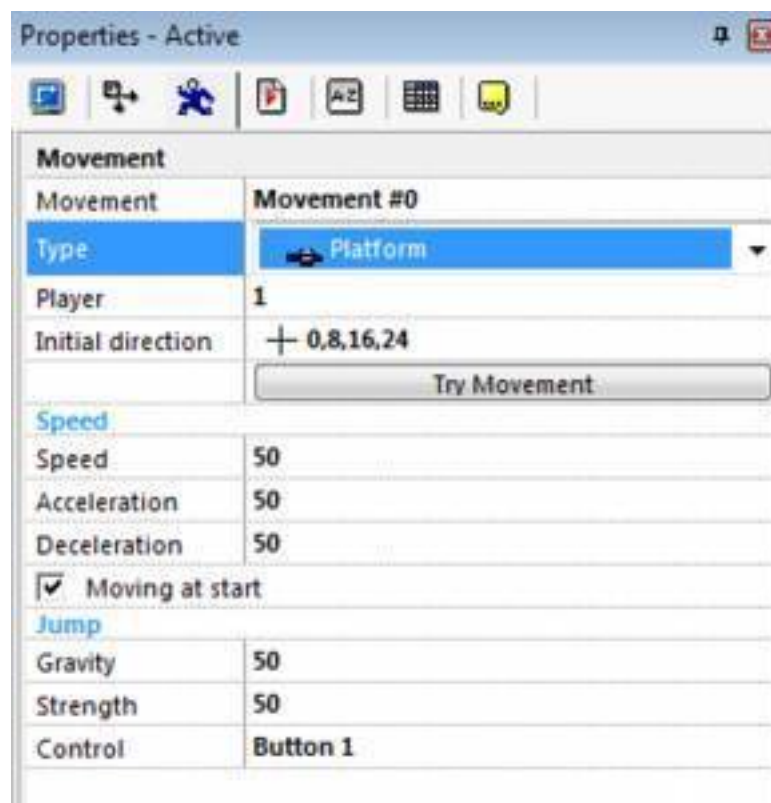




В окне настроек слева мы увидим настройки выбранного объекта. Нам нужно перейти в пункт **RunTime Options** и в выпадающем меню **Obstacle Type** выбрать тип **Obstacle**.

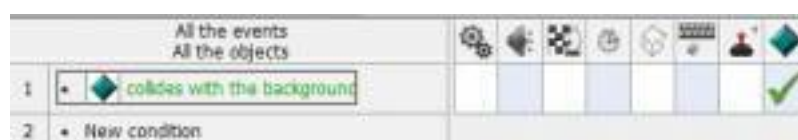


В свою очередь кликнем мышкой на наш ромбовидный объект так, чтобы в левой нижней части отобразились его настройки, и выберем тип перемещения **Platform**.



Затем перейдём в редактор событий (**Event Editor**) с помощью горячей комбинации клавиш **Ctrl-E**. И создадим свою первую программу, кликнув по надписи **New Condition** любой кнопкой мыши.

С помощью очередного клика на нашем ромбовидном объекте и выборе пунктов **Collision** и **Backdrop** создадим условие соприкосновения объекта с нашей «землёй». Теперь в правом столбце редактора найдём наш ромб и, кликнув правой кнопкой мыши по клетке ниже выберем пункты **Movement** и **Stop**.



Для большей наглядности перейдём в **Event List Editor** (Ctrl-L):



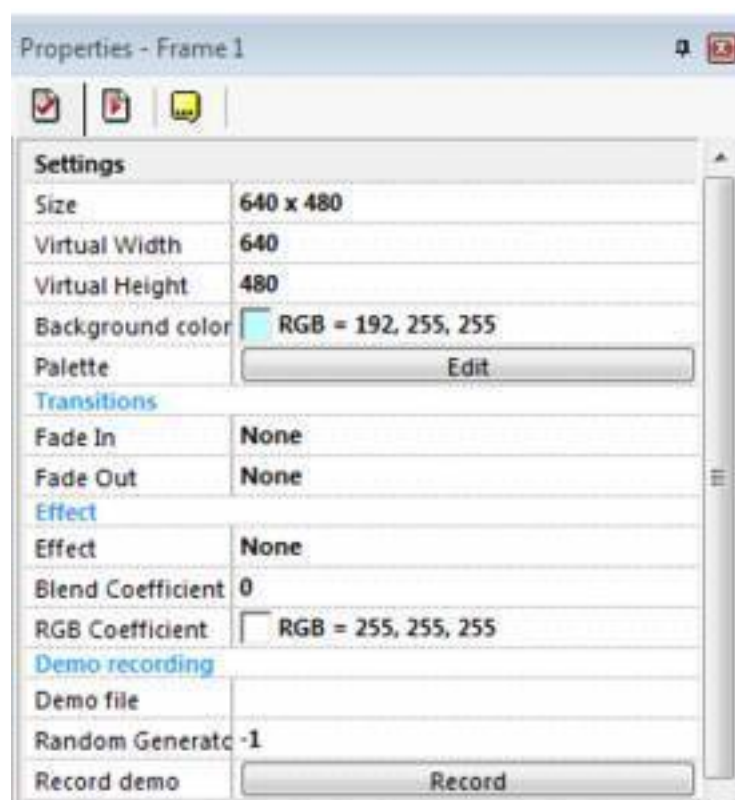
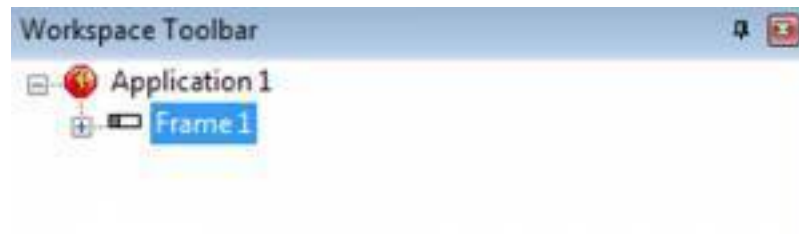
Если изображение на экране компьютера совпадает с картинкой, значит, программа создана правильно. Теперь проверим работу игры, нажав **F8**.

Клавиши управления: стрелки лево, право, Shift = прыжок.

## Урок 2. "Движок" для игры

Сегодня у нас снова довольно большой урок, демонстрирующий возможности программы Clickteam Fusion. Мы напишем ни много ни мало, а собственный движок для игры.

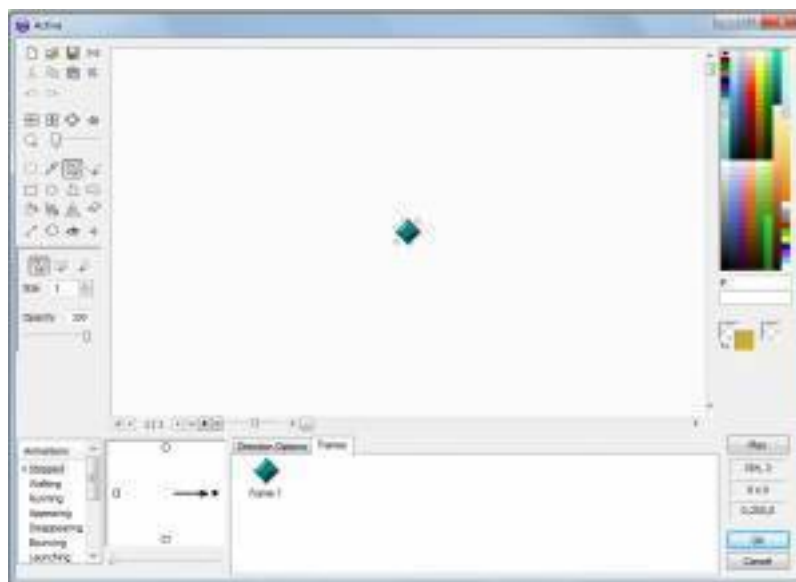
Как мы уже умеем из предыдущего урока, создадим новый документ и установим цвет фона, к примеру, голубого цвета. Делаем это в свойствах фрейма:





Кликнем дважды левой кнопкой мыши по любому месту и выберем создание активного объекта (**Active**). Его мы будем использовать в большинстве случаев.

Разместим его на экране и дважды кликнем теперь по нему, попав в редактор объекта. Об этом окне стоит рассказать поподробнее.



Это окно можно разбить на несколько условных частей. В левой части представлена своя панель инструментов, представляющая собой простейший графический редактор. В правой части мы видим цветовую палитру. В нижней части мы видим редактор анимаций. Давайте рассмотрим кнопки из левой части более детально.

Первый ряд.



**Clear (Ctrl-N)** – очистить текущий рисунок;

**Import (Ctrl-O)** – импортировать изображение или набор анимаций;

**Export (Ctrl-S)** – экспортировать текущий рисунок;

**Options** – выбор предпочтений.

Второй ряд.



**Cut (Ctrl-X)** – вырезать выделенную часть;

**Copy (Ctrl-C)** – скопировать выделенную часть;

**Paste (Ctrl-V)** – вставить предварительно вырезанную или скопированную часть;

**Delete** – выполнить процедуру удаления.

Третий ряд.



Третий ряд представлен кнопками «Восстановить действие» (**Undo Ctrl-Z**) и повторить отменённое действие (**Redo Ctrl-Y**).

Четвёртый ряд.



**Flip horizontally (Ctrl-I)** – отразить горизонтально;

**Flip vertically (Ctrl-J)** – отразить вертикально;

**Crop (Ctrl-K)** – обрезать пустое прозрачное место вокруг объекта;

**Transparency (Y)** – произвести действия с прозрачностью объекта.

В пятом ряду расположен значок лупы и ползунок, с помощью которого можно изменять масштаб представления объекта. Также, изменение масштаба можно произвести с помощью колёсика мыши.



Шестой ряд.



**Selection tool (B)** – выделяет часть объекта с помощью мыши;

**Color picker (P)** – пипетка, определяет цвет внутри данного окна и устанавливает его в окне цветовой палитры;

**Brush tool (D)** – кисть для рисования произвольных линий;

**Line tool (L)** – кисть для рисования прямых линий.

Седьмой ряд.



**Rectangle tool (R)** – рисование прямоугольника;  
**Circle tool (E)** – рисование окружности;  
**Polygon tool (G)** – рисование многогранников;  
**Shape tool (M)** – создание очертаний.  
Восьмой ряд.



**Fill tool (F)** – инструмент «Заливка»;  
**Spray tool (S)** – инструмент «Аэрозоль»;  
**Text tool (T)** – инструмент для создания текста;  
**Eraser tool (U)** – стирательная резинка.  
Девятый ряд.



**Size (W)** – изменение размера;  
**Rotate (A)** – поворот объекта;  
**View hot spot (H)** – показывает место отсчёта координат объекта;  
**View action point (Q)** – просмотр и изменение «активной точки», например, места, откуда будут вылетать другие объекты.

Ниже идёт рабочая область, изменяющая содержание в зависимости от выбранного выше пункта.

Цветовая палитра в особом представлении не нуждается, поэтому перейдём к рассмотрению редактора анимаций.



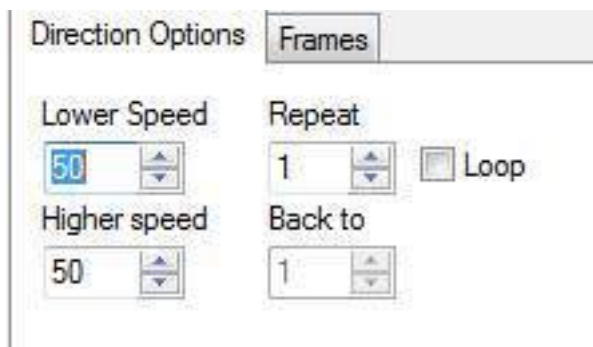
В самом левом окне мы видим названия анимаций. Вот они:

**Stopped** – объект в состоянии покоя;  
**Walking** – объект идёт;  
**Running** – бежит;  
**Appearing** – появляется;  
**Disappearing** – исчезает;  
**Bouncing** – отскакивает;  
**Launching** – запускает;  
**Jumping** – подпрыгивает;  
**Falling** – падает;  
**Climbing** – лезет по лестнице;  
**Crouch down** – наклон вниз;  
**Stand up** – подъём вверх.

Эти названия мы можем переименовывать, а также устанавливать свои с помощью меню, которое можно вызвать щелчком по любому названию **правой кнопкой мыши**.

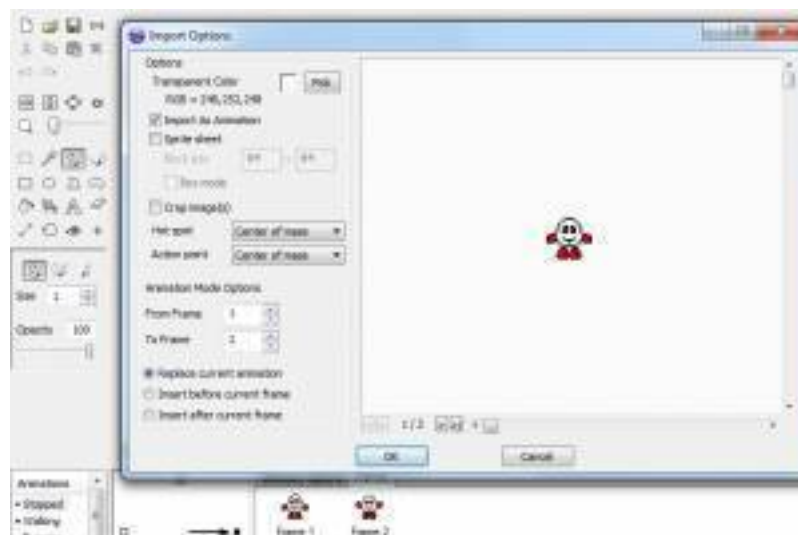
Чуть правее находятся стрелки, указывающие направление текущей анимации (для каждого названия можно выбрать не более четырёх – вверх, вниз, влево и вправо). Их мы можем отрисовывать сами, либо импортировать уже готовые. Справа находится кнопка «**Play**», с помощью которой мы можем проверить работу анимации, не закрывая окна.

Переключим режим средней части с вкладки **Frames** на **Direct Options**.



Здесь мы можем задавать скорость смены кадров (причём можно указывать нижний и верхний порог), количество повторений или, если кадры меняются всегда, то указать цикл.

Давайте попробуем анимировать нашего героя и заполнить анимации Stopped и Walking. У меня сохранились нарисованные мной анимации персонажа старой игры про Диззи для ZX Spectrum, которые я и использую.



И хотя для понимания нашего урока это необязательно, в будущем мы обязательно будем использовать этот навык.

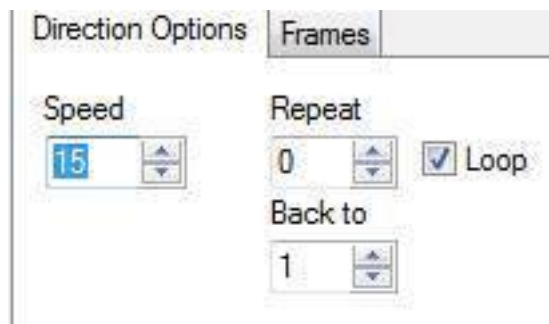
Для начала нам нужно найти готовые анимации героя. Как правило, это отдельные кадры png с прозрачным слоем или анимации gif. В последнем случае программа автоматически определит и импортирует все кадры. В случае png-кадров убедитесь, что в конце названия каждого кадра присутствует номер в порядке увеличения. Тогда программа сможет импортировать все пронумерованные кадры сразу. В противном случае придётся добавлять их вручную.

Если установлен чекбокс **Sprite sheet**, то мы сможем вырезать кадры из одного рисунка, предварительно установив ширину и высоту каждого кадра (строка ниже).

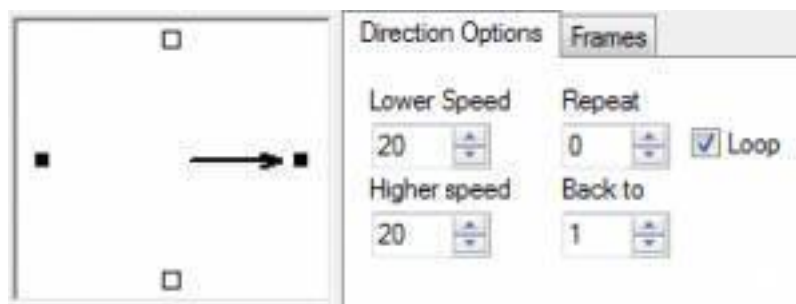
Откроем **Import (Ctrl-O)** – импортировать изображение или набор анимаций, укажем путь, а также дополнительные опции, как представлено на рисунке.

Дополнительно мы можем при импорте осуществить **Crop** (в последующем нажатием на иконку или горячими клавишами Ctrl-K), установить точку отсчёта координат и «активную» точку, а также произвести импорт выделенной части кадра.

После импорта перейдём из вкладки **Frames** во вкладку **Directions Options** и установим цикл и скорость смены кадров, например, в 15.



Аналогичные действия произведём и с серией кадров **Walking**, только сделаем это уже для направлений «вправо» и «влево».



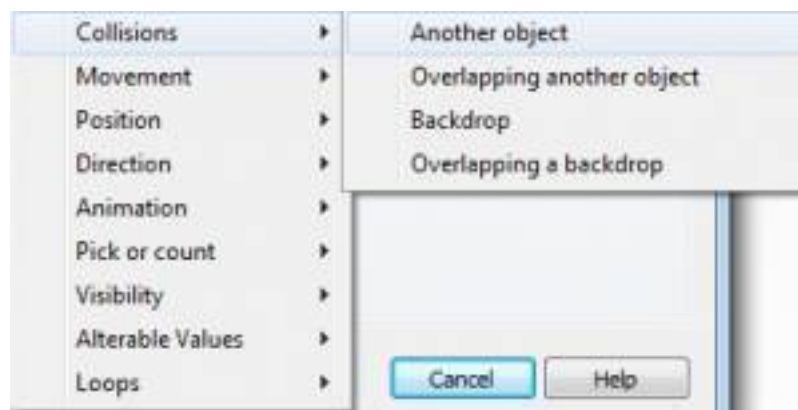
Заполненные кадры анимации отмечены на окне слева чёрными точками, а незаполненные – контурными.

Если имеются анимации прыжка и падения, заполним и их.

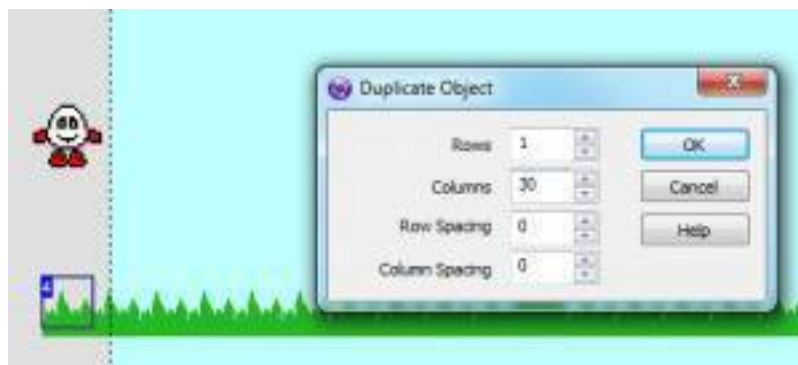
После создания героя мы должны указать его тип перемещения, благо что в программе имеется богатый выбор. Как и в прошлом уроке, мы выставяем тип **Platform** во вкладке свойств объекта (там, где нарисован синий бегущий человек) и выставяем другие значения скорости, гравитации и силы. Я их выставил по значениям 45, 45 и 80, оставив параметры ускорения – замедления без изменений.

Создадим новый объект – землю для нашего героя. Как и в прошлом уроке, мы можем создать её из вкладки **Background – Backdrop**, но можно использовать в качестве земли и активный объект. Тогда наше условие соприкосновения (предотвращение падения объекта) будет выглядеть так (**Collision – Another object**):

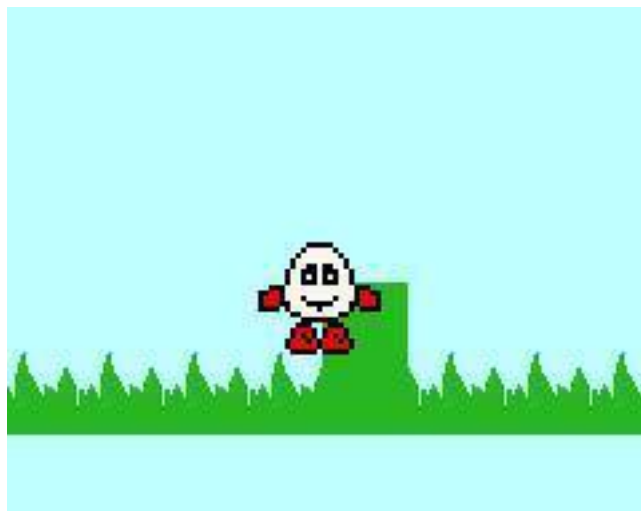




Дополнительно добавим к нашей скупой поверхности траву, создав её в качестве ещё одного активного объекта, разместим её над землёй и растиражируем на всё оставшееся незаполненное место через функцию **Duplicate**.



Казалось бы, наш урок должен быть завершён, ведь при запуске проекта по **F8** мы увидим корректную работу программы. Однако при создании героя мы использовали движущиеся объекты, а это значит, что при соприкосновения движущейся руки, ноги или иной части героя с более-менее вертикальной плоскостью, герой будет совершать непредвиденные движения, например, подлетать вверх. Герой сможет взбираться на вертикальные стенки, совсем как человек-паук, и существенно нарушить игровой процесс. Проверим это, создав посередине нашей земли выступающий прямоугольник и указав это в редакторе событий (условие, аналогичное остановке объекта с землёй):

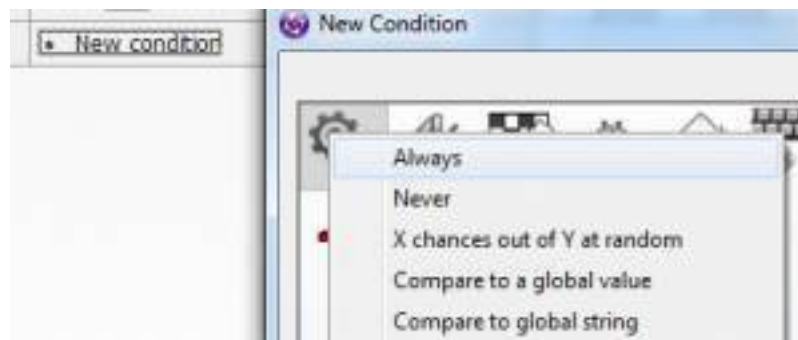


Чтобы этого не происходило, укажем для нашего героя Диззи тип перемещения **Stop** (он изначально таковым и был), и создадим новый активный объект прямоугольной формы, которому мы и присвоим тип **Platform** и все те же свойства главного героя.

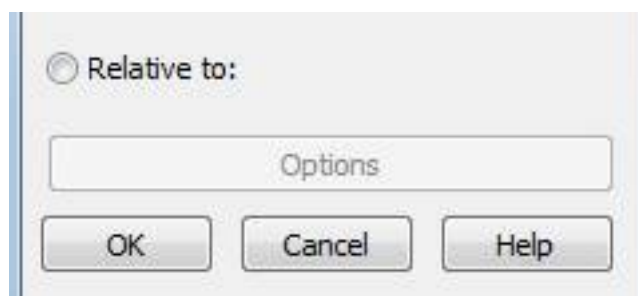
Соответственно в редакторе событий заменим все условия соприкосновения с объектами с Диззи на наш новый прямоугольник. Делаем это просто: в редакторе событий дважды кликаем на иконку Диззи и выбираем вместо него новый объект.

Проверим работу игры. Если при запуске проекта наш прямоугольник сможет перемещаться по земле, продолжим работу и перейдём в редактор событий.

Кликнем снова на надпись **New condition** и выберем из выпадающего меню пункт **Always** («Всегда»):



В столбце напротив героя выберем **Position – Select Position**, а затем в новом окне надпись **Relative to**:



В новом окне выберем нашего героя – Диззи. Таким образом, анимации нашего героя будут находиться поверх нашего нового прямоугольника.

Теперь, если запустить игру, станет ясно, что осталось направить анимации героя в нужную сторону. Сделаем это, определяя направление движения героя и устанавливая соответствующую анимацию. Если герой стоит, будем присваивать ему анимацию Stopped:



В последующем мы для удобства будем приводить такой листинг (**Ctrl-L**). Он удобен тем, что показывает нам все действия, которые запрограммированы в редакторе событий. В левой колонке указан номер события, напротив чёрной точки – условия, а ниже с небольшим абзацем – сами действия.

Разберём ещё раз действия нашей программы по условиям.

1: Проверяем соприкосновение нашего основного объекта с типом Platform с землёй. Если он соприкасается, мы останавливаем движение.

2: Проверяем соприкосновение нашего основного объекта с типом Platform с препятствием аналогично пункту 1.

3: Всегда делаем так, чтобы персонаж с анимациями (Диззи) находился в центре нашего основного объекта.

4: Если основной объект, которым мы управляем, стоит, показываем анимацию персонажа Stopped.

5: Если направление движения объекта вправо, при этом объект движется, направляем Диззи вправо, при этом выбираем анимацию ходьбы.

6: Аналогично для движения влево.

Вот и всё, взаимодействие персонажа с окружающим миром закончено. Как обычно, клавиши управления: стрелки лево, право, Shift = прыжок.

## Урок 3. Стартовое меню

Если задать себе вопрос, с чего начинается игра, то обычно мы ответим: с меню. Создание стартового меню и выбор пунктов мы и рассмотрим в этом уроке.

Как обычно, создадим новый документ, дважды кликнем левой кнопкой мыши по появившемуся окну и найдём текстовый объект «**String**». Чтобы его найти быстрее, достаточно ввести начальные буквы в строке поиска.

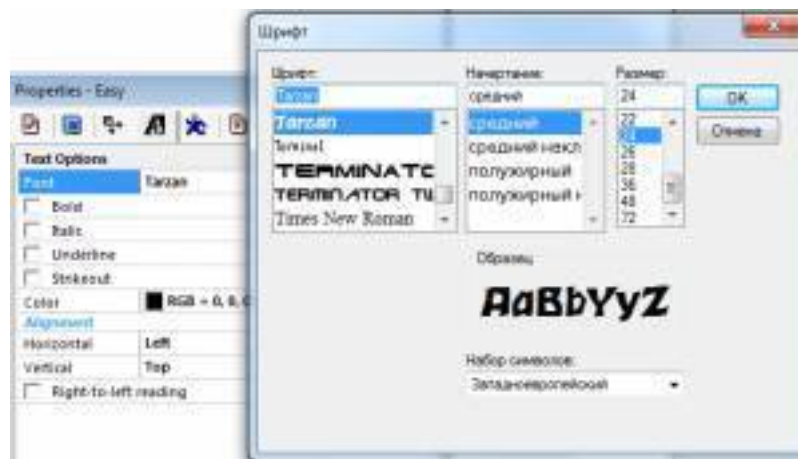


Объект можно найти и другим способом: кликнуть в правом меню по группе «**Text**».

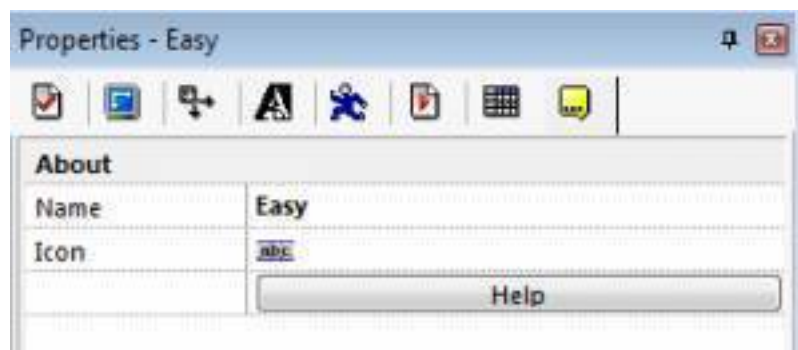
Растянем побольше наш объект до размера читаемого пункта и перейдём в окно свойств. Обращаю внимание на слово «**Paragraph**» в окне свойств – в данном случае мы могли бы ввести несколько вариантов строки через кнопку «**New**», если бы делали, скажем, игру на разных языках.



Но мы идём дальше в текстовые опции и изменяем тип и размер шрифта. Например, выберем шрифт Tarzan 24-го размера.



Поменяем, чтобы не забыть, и имя объекта. Вместо стандартного «String» напишем имя объекта, пусть это будет «Easy» – лёгкий уровень сложности нашей игры. Кроме него, у нас будут уровни «Medium» и «Hard».



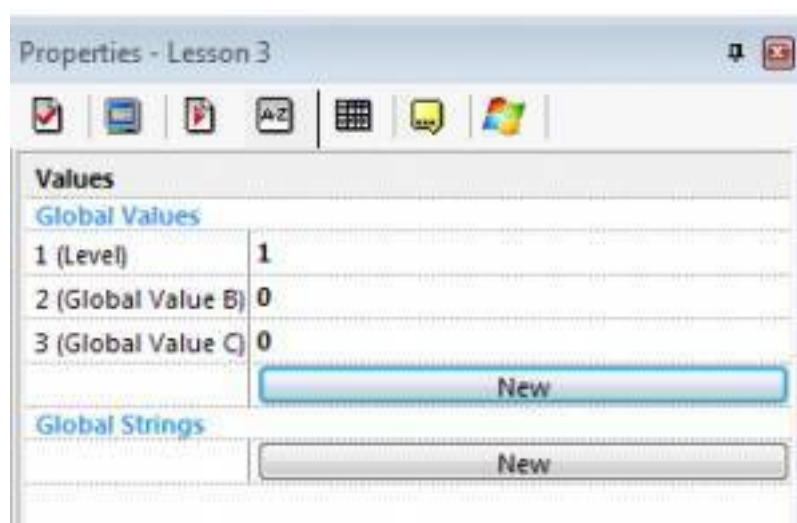
Давайте будем учиться всегда при создании нового объекта менять имя по умолчанию («String», «Active» и др.) на осмысленное, соответствующее функциональности нашего объекта. Поверьте, это очень поможет в дальнейшем!

Выберем пункт «Клонирование объекта» и введём количество строк – 3. Мы видим три одинаковых объекта, но с разными именами в окне свойств. Поменяем имена объектов, как мы только что сделали выше, а затем и текст в самих строках.



Теперь создадим активный объект, который будет у нас в роли переключателя пунктов. Нарисуем в нём, например, оранжевый круг. Снова, как мы вспомним, введём имя объекта.

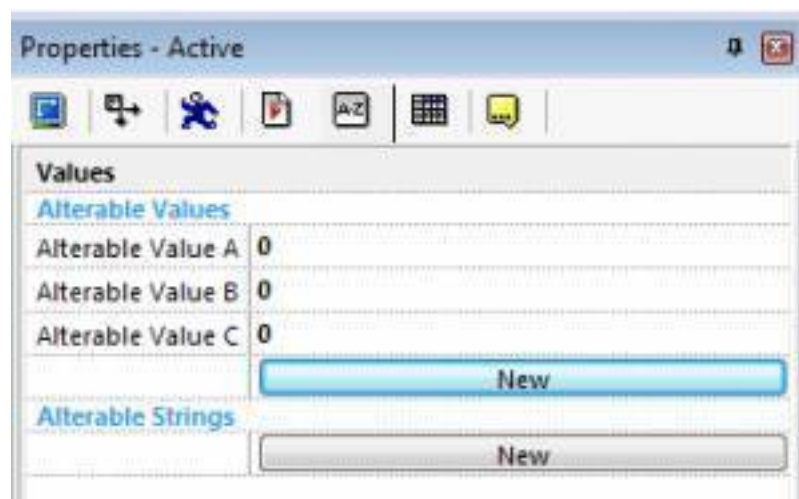
Теперь поговорим о переменной, которая будет иметь значение текущего уровня. Во-первых, для нашей задачи мы можем использовать глобальные переменные. Они расположены во вкладке «**Values**» свойств нашего документа (не фрейма!).



Они имеют значения вида «1 – Global Value A», и их можно переименовывать. Например, на рисунке выше мы создали три переменные и указали имя первой из них «Level». Их количество ограничено.

Также, мы можем использовать переменных из свойств объекта. Возьмём, например, наш активный объект, в котором мы нарисовали оранжевый круг, и заглянем в его свойства на вкладку «**Values**».





Во вкладке расположены переменные объекта. Каждый объект с уникальным именем имеет свой набор переменных. Таким образом, мы имеем неограниченное количество переменных, однако они действуют только в пределах нашего фрейма. В другом фрейме их использовать по умолчанию нельзя.

И наконец, для этой цели мы можем использовать счётчики. Это самый наглядный способ, и их значение проще всего отслеживать, если разместить их в рабочей области фрейма.

Сейчас мы используем первый способ, то есть перейдём к названию нашей игры в окне рабочего пространства и выберем имя нашей переменной – **Level**. Присвоим ей значение 1.

Вызываем редактор событий.

Создадим первое условие. Если наша переменная равна единице, то значение Y для нашего оранжевого круга должно равняться значению Y нашего объекта с текстом.

То же самое делаем и для остальных значений.

Создадим условия перемещения оранжевого круга. Добавляем ещё одно условие – если нажата клавиша «Вниз» на клавиатуре.

В этом случае к переменной **Level** мы прибавляем единицу.

Копируем предыдущее условие и изменяем его так, чтобы при нажатии клавиши «Вверх» из переменной **Level** вычитали единицу.

Теперь всё? Давайте подумаем. Согласно нашему алгоритму значение **Level** может вырасти больше трёх или уменьшиться до минусовых значений. Что делать? Так как мы работаем с шагом на один, то есть прибавляем или отнимаем единицу, то следующее значение после трёх будет 4, а до 1 – 0. Так мы и поправим.

Если значение **Level** равно 0, устанавливаем его в 3 (чтобы цикл работал по кругу), а если 4 – то одному. Проверяем работу.

Оранжевый круг стоит не совсем ровно по отношению к надписям. Дело в том, что в этой версии редактора точка объекта, по которой идёт расчёт положения в пространстве, по умолчанию находится в центре. А для текста – в левом верхнем углу. Поэтому дважды кликнем по оранжевому кругу и в настройках поменяем положение этой точки (на изображении глаза) в левый верхний угол. Проверим работу, теперь всё нормально.

Давайте теперь проверим, а сохраняется ли в новом фрейме наша переменная **Level**. Создадим новый фрейм и создадим в нём надпись об успешном окончании работы программы. И надпись о значении счётчика. Отмечу, что название можно менять и не в самом объекте, кликая дважды по нему, а и в окне свойств. Теперь нужно создать счётчик.

Найти его сложно, поэтому просто используем строку поиска.



Перейдём во второй фрейм в редактор событий и создадим условие – при начале работы фрейма значение счётчика нужно установить равное переменной Level.

Добавим значение перехода на следующий фрейм при нажатии кнопки Enter. Для этого нам нужно вернуться в окно редактора событий первого фрейма и добавить условие. Проверим.

Итак, всё работает, а покажу ещё один способ. Как можно сделать выбор меню при левом клике мыши на каждой из строк.

Создаём условие. Если пользователь кликает по первому объекту, причём делает это левой кнопкой мыши и одинарным кликом, значит, значение переменной мы установим в 1. И перейдём в следующий фрейм.

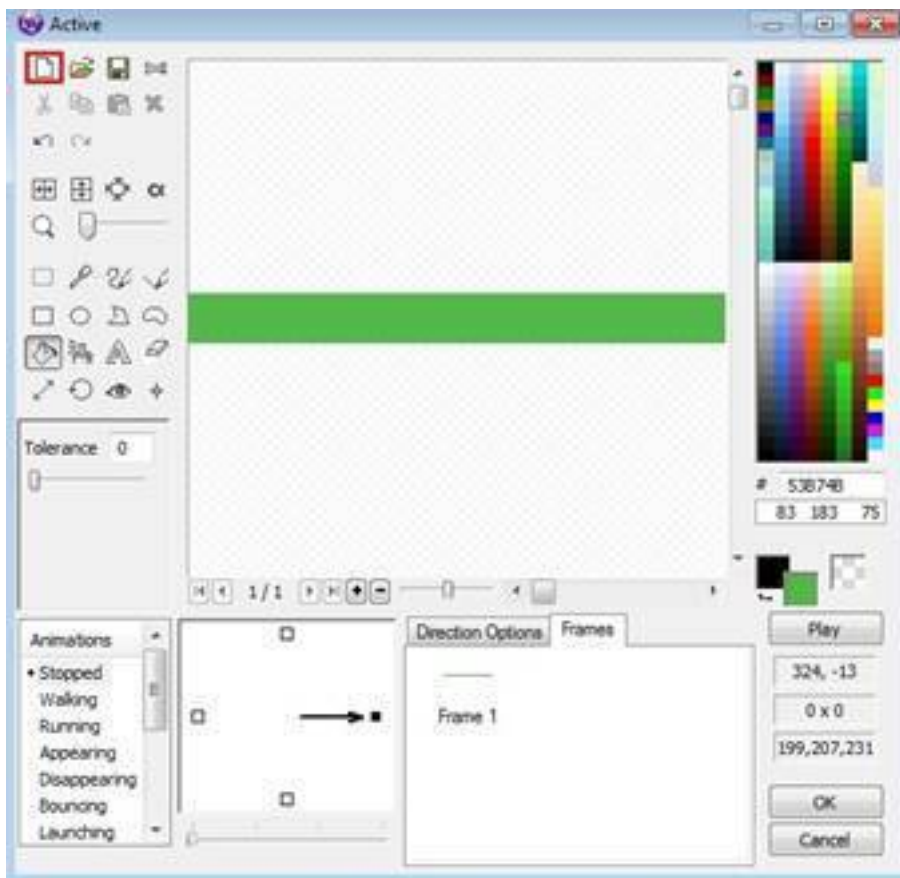
То же самое мы сделаем для второго и третьего объекта.

Проверяем. Всё работает (а если нет, здесь и далее можно скачать готовый пример в описании видео соответствующего урока).

## Урок 4. Инвентарь в игре

В этом уроке мы рассмотрим наиболее сложный момент, который встречается при создании игр, – инвентарь.

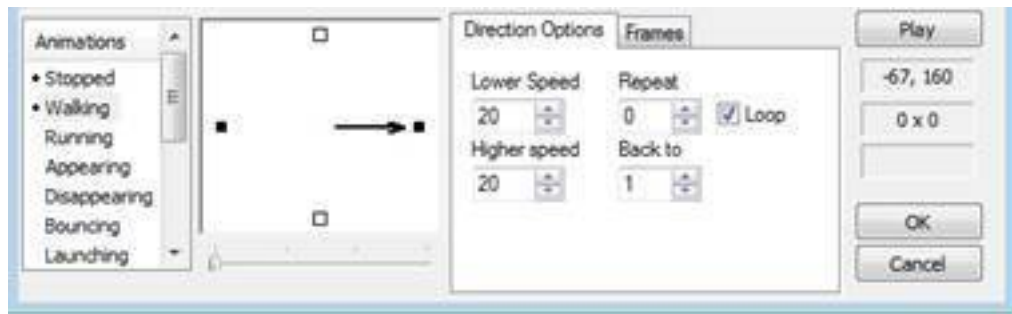
Создадим новый документ. Сначала нарисуем поверхность и героя, как мы уже делали в уроке 2. Ничего страшного, повторение – мать учения. Растянем первое изображение в длинный прямоугольник, нажмём **Clear** и воспользуемся зелёной заливкой.



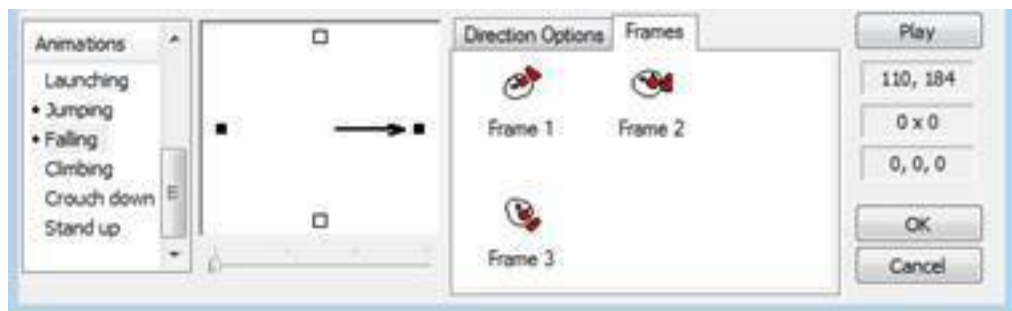
Создадим героя, назовём его “Hero” и загрузим для него спрайт Диззи. В **Animations** – **Stopped** установим:



Для **Walking** установим следующие значения для движения вправо и влево (а также загрузим по 3 анимации Диззи для движения в этих направлениях):



Точно такие же значения зададим для **Jumping** и загрузим по 7 анимаций. Для **Falling** будут те же значения и по 3 анимации.



Теперь создадим два предмета, которые будем брать в инвентарь. Пусть это будут жёлтый и красный круги.

Вот что у нас пока получилось:



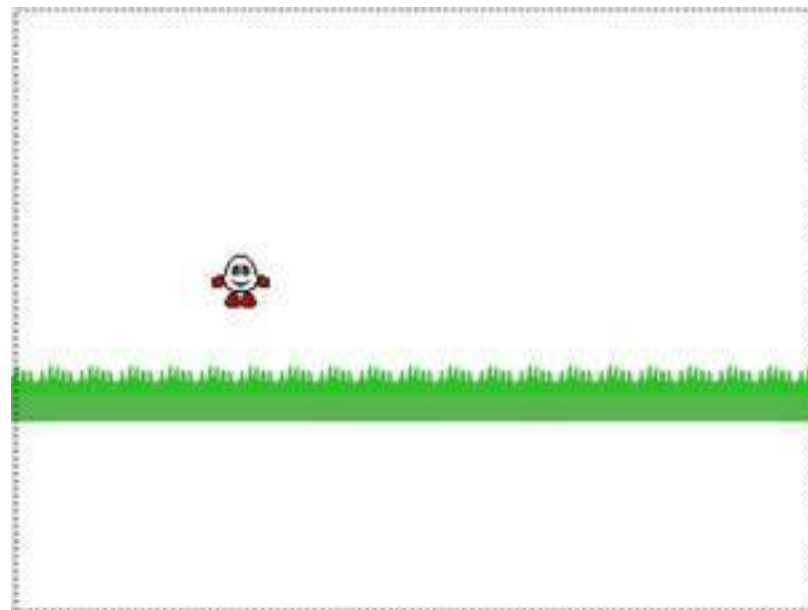
Кликаем на **Event Editor – New condition – Hero – Collisions – Another object** и выбираем нашу поверхность. Затем задаём для нашего героя **Movement – Type – Platform**. Возвращаемся в **Event Editor**, кликаем на столбец героя и задаём **Movement – Stop**.



Создаём еще один объект Active. В его окне выбираем Fill tool, Clear, Brush tool и рисуем фрагмент травы.

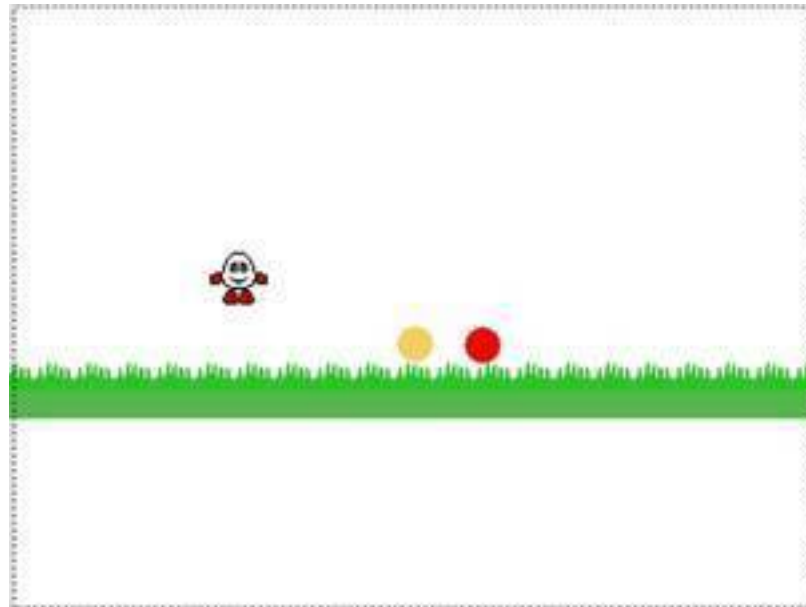


Называем её Green и с помощью функции **Duplicate** создаём 22 таких же изображения. Размещаем их так, чтобы они располагались на нашей поверхности от одного конца экрана до другого.



Запускаем приложение (**F8**) и убеждаемся, что Диззи может ходить вперед и назад по траве.

Теперь создадим 2 предмета, которые будем брать в инвентарь. Пусть это будут жёлтый и красный круги. Создаём новые предметы, называем их Item 1 и Item 2, рисуем круги и раскрашиваем их в жёлтый и красный цвета соответственно.



Находим в списке объектов (**All objects**) **Popup menu** и сразу же вводим первую позицию (**Add**). Пусть она будет называться **Nothing** («Ничего»). Мы дадим нашему герою возможность выбора – ничего не брать и закрыть меню.



Привыкайте, что сами предметы лучше сразу включать в группы. Тогда мы сможем обращаться не к конкретному объекту, а к любому объекту в группе. Следующий пример покажет, как это работает.

Кликаем на красный круг, **Events – Qualifier(s) – Edit – Add – Friends**.



А пока откроем редактор событий. Создадим первое условие, чтобы наш герой не падал вниз, а перемещался по земле. **New condition – The mouse pointer and keyboard – The keyboard – Upon pressing a key**, нажимаем клавишу Enter.

И сразу создаём второе условие. Давайте при нажатии клавиши Enter выводить наше меню в заранее указанное место. Кликаем на столбец **Popup menu**, выбираем **Show At** и перетаскиваем меню в нужное место.

Копируем второе условие и создаём третье. Ведь если мы нажали Enter и в это время герой перекрывает объект, его нужно добавить в наше меню. Так мы и поступим: кликаем на **Upon pressing “Enter” – Insert – Hero – Collision – Overlapping another object – Item 1** (жёлтый круг). Далее кликаем на столбец **Popup menu – Add item**.

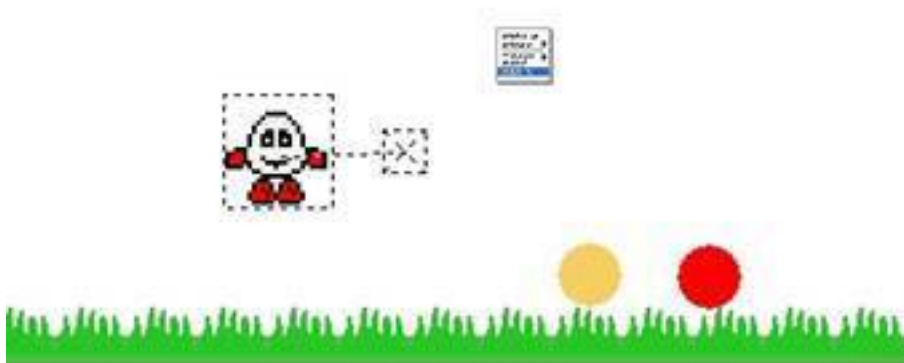
Присвоим жёлтому кругу номер 2 (первый у нас уже есть – это «Ничего») и название “Yellow circle” (в появляющихся окнах нужно написать 0, 2 и Yellow circle соответственно). Сам объект перенесём в область за пределы игрового пространства: кликаем на столбец жёлтого круга, **Position – Select position** и задаем его координаты.

Создадим следующее условие на основе предыдущего: вновь скопируем строку условия, поменяем в ней жёлтый круг на красный и переставим галочку на столбец красного. При этом мы изменим номер объекта (3) и его название (Red circle).



Теперь научим программу выкладывать предметы: **New condition – Popup menu – Item selected – 2**. Если в нашем меню выбран Item 2, удалим его из меню и перенесём его снова

рядом с главным героем: кликаем на столбец жёлтого круга – **Position – Select position – Relative to – Hero**. Далее перетащим квадрат с изображения героя на положение справа от него.



Теперь в столбце **Popup menu** выберем **Delete item – 2**. Всё то же самое сделаем и для красного круга.

All the events All the objects		Settings	Hero	Grass	Yellow Circle	Red Circle	Inventory
1	Collision between  and						
2	Upon pressing "Enter"						
3	Upon pressing "Enter"						
4	Upon pressing "Enter"						
5	Item 2 Selected						
6	Item 3 Selected						

Проверяем (F8). Итак, у нас есть меню, в котором ничего нет. Возьмём поочерёдно жёлтый и красный круги, а потом всё вместе. Всё работает!

Закроем этот проект и создадим новый, в котором мы создадим наглядный инвентарь для квестовой сцены. Создадим объект (**Active**).

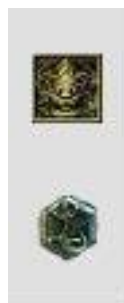
Загрузим заранее подготовленную картинку с пустыми окнами для предметов и два предмета. В данном примере использовалось изображение игры «Secrets of the Dark: Eclipse Mountain», взятое с сайта [www.metacritic.com](http://www.metacritic.com). Давайте узнаем размер картинки, создадим экран такого же размера (для картинки ниже это **Frame 1 – Settings – 560 x 420**), а саму картинку установим в координаты 0:0, то есть она должна полностью совпасть с экраном.





Назовем картинку “Font”.

Создадим два активных объекта, загрузим заранее заготовленные изображения предметов и назовём их Item 1 и Item 2. Не забываем поместить их в общую группу (**Events – Qualifier(s) – Edit – Add – Friends**).



Перетаскиваем их на нашу картинку.

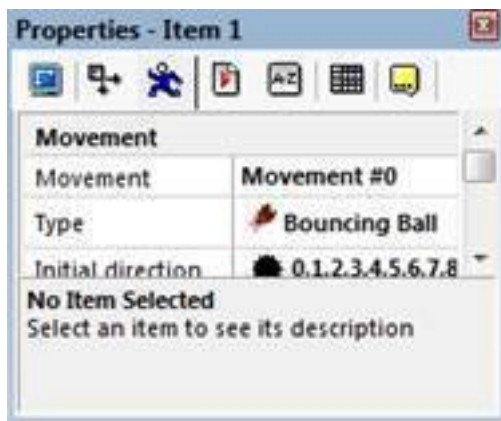




Теперь создадим новый объект, который будет активным местом для нашего меню. Сделаем его зеленым квадратом по размеру ячейки в нижней части нашей картинки.



Кстати, не забудем указать для наших предметов инвентаря тип перемещения **Bouncing ball** («прыгающий мяч»).



Переходим в редактор событий.

Создаём первое условие: **New condition – Storyboard controls – Start of frame**, в столбцах Item 1 и Item 2 выбираем **Movement – Stop**. Так при старте фрейма мы останавливаем наши объекты, так как изначально скорость движения у них указана 60 (в настройках типа перемещения).




Второе условие: если мы кликаем мышью по объекту, мы направляем его в сторону активного квадрата. **New condition – The mouse pointer and keyboard – The mouse – User clicks on an object**. Нажимаем ОК.



В появившемся окне выбираем **Friends**.




Далее кликаем на столбец друзей – **Direction – Look in the direction of – Relative to** – Зеленый квадрат. Нажимаем «ОК» в следующем окне, подтверждая координаты 0:0. Далее снова кликаем на столбец друзей – **Movement – Set speed** и указываем скорость движения 60.

All the events All the objects		⚙	🔊	🎮	🕒	📅	🎞	👤	🏠	🌳	🚪	🏠	🟢
1	• Start of Frame											✓	✓
2	• User clicks with left button on 											✓	

Заметьте, мы указываем не конкретный предмет инвентаря, а сразу всю группу.

Теперь давайте остановим объект и разместим его в ячейке. Сделаем это в виде условия накладывания объекта – здесь мы опять используем группу – на наш активный квадрат. При этом мы размещаем предмет по центру квадрата, останавливаем его, а сам квадрат отодвигаем в сторону так, чтобы он был размещён по центру следующей ячейки. **New object – Friends – Collisions – Overlapping another object – Зеленый квадрат.** Кликаем на столбец друзей – **Position – Select position – Relative to – Квадрат.** Еще раз кликаем на столбец друзей – **Movement – Stop.** Далее столбец квадрата – **Position – Select position – Relative to – Квадрат.**



All the events All the objects		⚙	🔊	🎮	🕒	📅	🎞	👤	🏠	🌳	🚪	🏠	🟢
1	• Start of Frame											✓	✓
2	• User clicks with left button on 											✓	
3	•  is overlapping 											✓	✓

Проверяем работу – всё должно работать. Предметы летят к квадрату, сам квадрат смещается на одну ячейку.





Но что будет, если будет заполнено всё место в инвентаре?

Сделаем наш квадрат невидимым. Расклонируем любой наш объект, чтобы проверить заполненность инвентаря.



Сначала научим программу удалять объекты из инвентаря. Условие: если мы кликаем левой кнопкой мыши по объекту, при этом объект находится в зоне меню, его можно удалить.

**New condition – The mouse pointer and keyboard – The mouse – User clicks on an object – OK – Friends.** Далее кликаем на условие – **Insert – The mouse pointer and keyboard – The mouse – Check for mouse pointer in a zone** и выделяем нижнюю часть нашей картинке (ячейки).



Кликаем на столбец друзей – **Destroy**.

Следующее условие. При выходе активного квадрата за пределы нашего инвентаря его нужно вернуть на первую позицию. Но сделать это нужно в том случае, если объектов в инвентаре меньше или равно семи. **New condition – Квадрат – Position – Compare X position to a value**, затем задаем значение и выбираем **Greater or equal**.

И ещё. Почему на первую позицию, ведь она может быть занята предметом? Посмотрим ещё раз наш алгоритм. Если первая позиция будет занята предметом, то квадрат сразу сместится на вторую, а если занята вторая – на третью, и так далее.

Кликаем на условие – **Insert – Friends – Pick or count – Compare to the number or “Group.Friends” objects in a zone** и снова перемещаем окно на наши ячейки внизу. В появившемся окне выбираем **Lower or equal, 7**.



Первую координату X активного квадрата мы узнаем из его свойств.



Кликаем на столбец квадрата – **Position – Set X coordinate** и вводим координаты.

Так как у нас уже есть условие по клику мыши, исключим для него возможность срабатывания, когда предмет уже находится в инвентаре (кликнем на координаты и выберем в выпадающем списке **Lower**). В Start of frame добавим опцию **Stop** для **Item 3-10**.



All the events All the objects																		
1 • Start of Frame										✓	✓	✓	✓	✓	✓	✓	✓	✓
2 • User clicks with left button on										✓								
3 •  is overlapping										✓	✓							
4 • User clicks with left button on • Mouse pointer lays within zone (121,370) to (440,411)										✓								
5 • X position of  >= 440 • Number of  in zone (122,370) to (438,415) <= 7											✓							

Проверяем. Заполняем инвентарь предметами, удаляем их и снова заполняем. Всё должно работать.



Программу, которую мы создали, можно ещё и оптимизировать. Какие наши действия были лишними?

А в следующий раз мы познакомимся с приближением в играх.

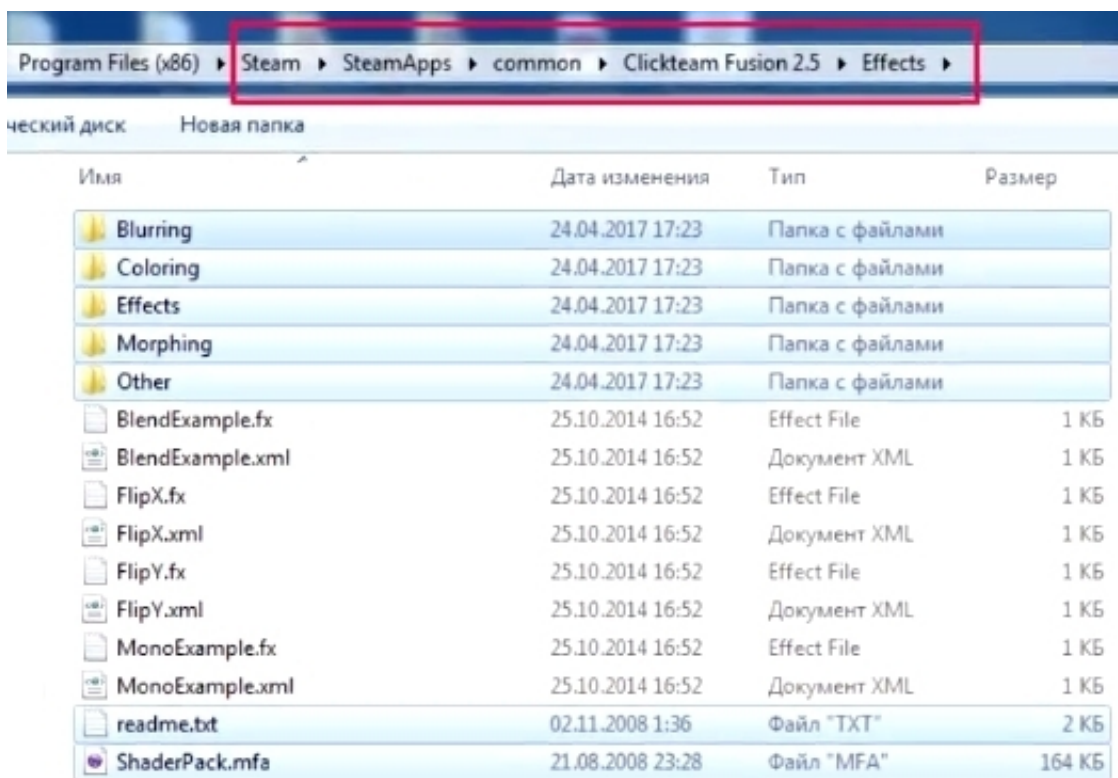
## Часть II. Функционал игр

### Урок 5. Приближение в играх

В этом уроке мы узнаем, что такое zoom-zoom. Мы рассмотрим три способа, но сначала зайдём на официальный сайт **Clickteam – Clickstore** и загрузим Mega Shader Pack.



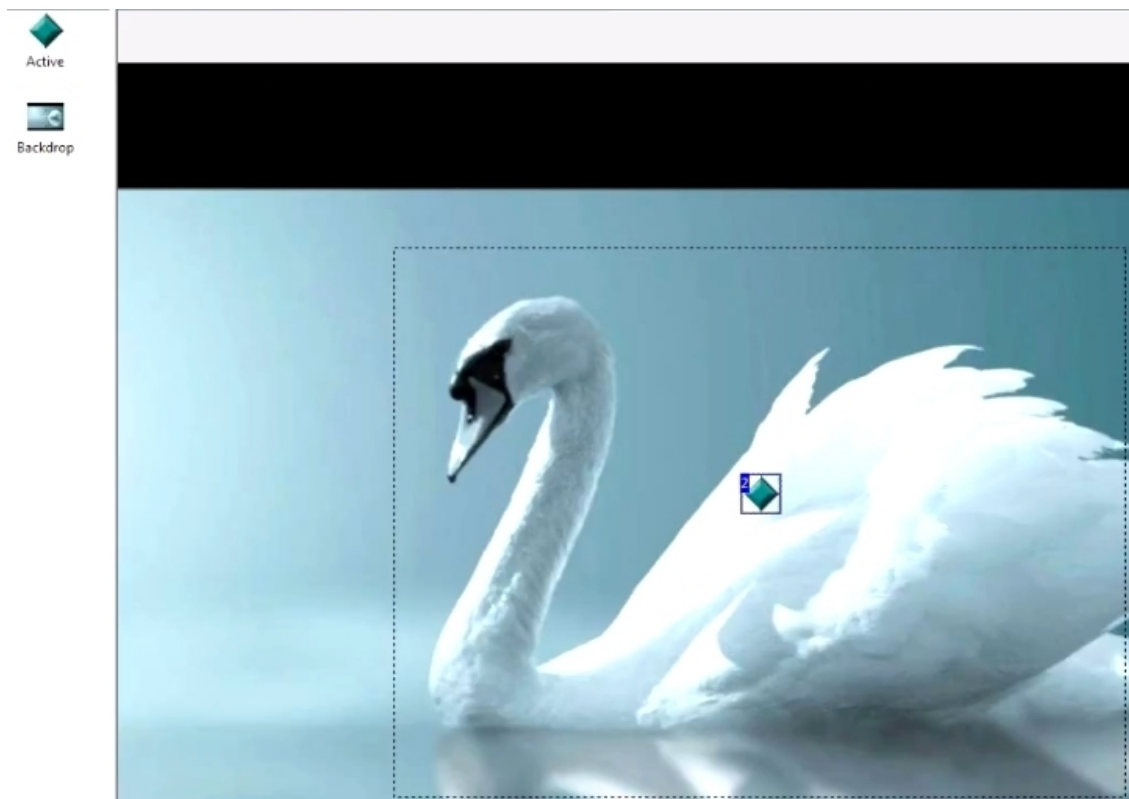
Затем перенесём файлы из архива в папку с Clickteam Fusion.



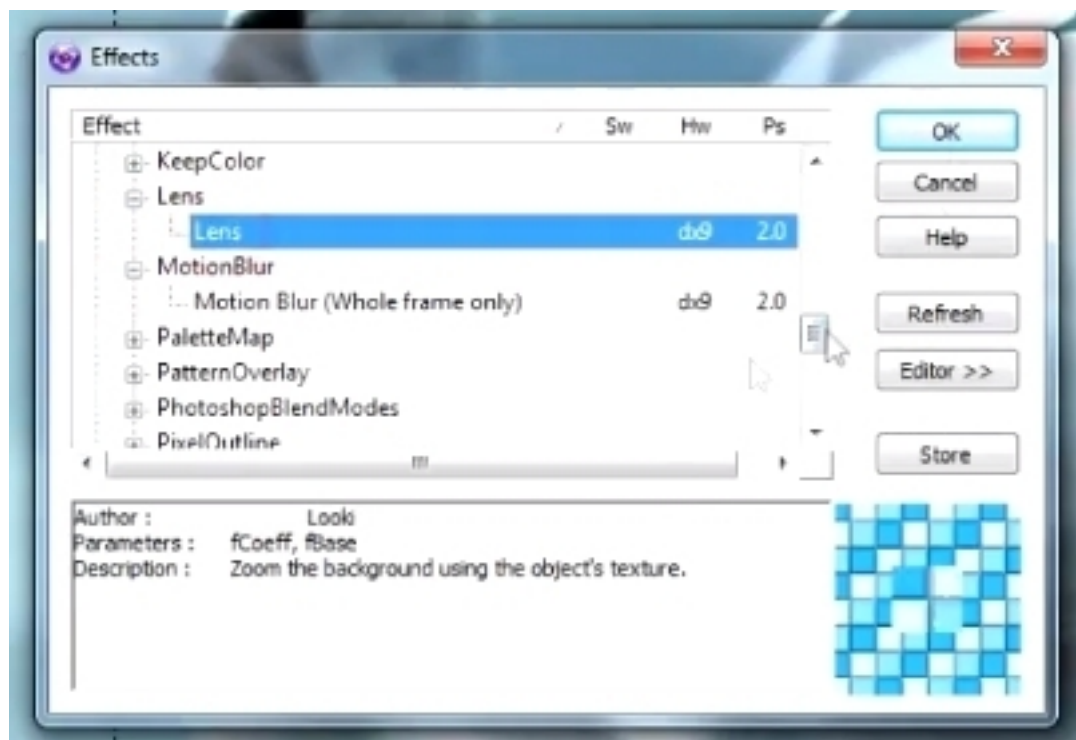


### Первый способ

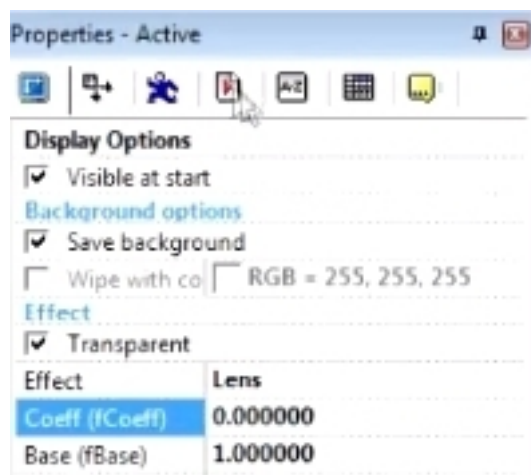
Откроем новый проект, загрузим любой понравившийся фон (например, свободно распространяемое изображение с сайта 10wallpaper.com) и добавим активный объект и Backdrop.



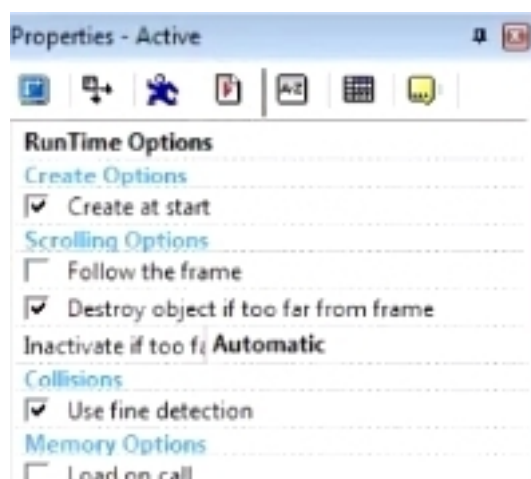
Добавим для активного объекта эффект линзы: **Properties – Effects – Lens – OK.**



Также снимем чекбокс в настройках напротив слов «следовать за фреймом».



Также перетащим активный объект и зададим ему такие параметры, чтобы он занимал все поле.





Затем перейдём в редактор событий и будем увеличивать первую переменную при прокручивании колёсика мыши вверх и уменьшать при прокручивании колеса вниз. **New condition – The keyboard – When mouse wheel is moved up – OK**, повторяем для следующего условия, выбирая «**Moved down**».

All the events All the objects								
1	• On mouse wheel up							
2	• On mouse wheel down							
3	• New condition							

Кликаем на столбец объекта в первом условии – **Alterable values – Add to** – (вводим 1) – **OK**. Повторяем для второго условия, заменяя на «**Subtract from**».

В последнем событии «Всегда» мы будем присваивать второй переменной значение по показанной формуле и применять нужный эффект. Для третьего условия кликаем на **Special conditions – Always**. Кликаем на столбец объекта – **Alterable values – Set** – (выбираем в выпадающем списке Alterable Value B, кликаем на объект) – **Values – Values A to M – Retrieve alterable value B** –

## **Конец ознакомительного фрагмента.**

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.