Новые наработки и пояснения к ним. Смотрим...

Новые наработки и пояснения к ним.

Вышла новая версия - 2.4.0

Версия 2.3.0

+ **onDblClick** - Сообщения приходящие от мыши
+ **onMouseDown**
+ **onMouseUp**
+ **onMouseMove**
* Переименовано
ProgressBarPercent -> ProgressBarPercentSection
* **ProgressBarPercentSection** теперь Double
+ **ProgressBarPercentFull** (ReadOnly) - количество выполненного в процентах относительно всего прогрессбара
+ **Свойство Visible** для анимации
+ **WaitModeStart** - Метод вводящий прогрессбар и иконки в режим ожидания с соответствующей индикацией
+ **WaitModeEnd** - Выход из режима ожидания
+ **isWaitMode** - Находимся ли мы в режиме ожидания
+ **onWaitModeCustomDraw** - Событие позволяющее самому рисовать в прогрессбаре режим ожидания, если не устраивает дефолтная прорисовка.
* Изменен порядок прорисовки, теперь сначала рисуется анимация, затем иконки, потом прогрессбар и наконец текст.

Новая демка **ASFDemoPool** демонстрирующая работу с мышью (несколько курсоров, перемещения, клики, обработка событий связанных с ней). Так же демонстрируется реализация процесса ожидания для прогрессбара. В этой же демке можно увидеть каким образом можно использовать сплэш для создания так называемого окна About.

Обновлен **SplashCreator** до версии 1.2
Поддержка свойства Visible для анимации и мелкие исправления.

Скачать новую версию модулей, как всегда, можно **тут**

Еще несколько пояснений.
По событиям мыши, я думаю, все ясно.

Переименовано свойство ProgressBarPercent в связи с тем, что появилось новое **ProgressBarPercentFull**
показывающее процент заполнения всего прогрессбара, соответственно свойство **ProgressBarPercentSection**
показывает процент секции.
Анимацию теперь можно включать и выключать в процессе работы.
Теперь про режим ожидания (**WaitMode**). Был введен для того, чтобы вводить прогрессбар и иконки в специальный режим, при котором на на них выполняется зацикленная анимация, что в свою очередь должно свидетельствовать о том, что ничего не висит, а усердно работает. Так как часто неизвестно ни время выполнения (LoadingSectionTime), ни тем более процент выполнения (ProgressBarPartComplite), например при коннекте с удаленным сервером, то можно перейти в режим ожидания. Реализацию смотрите в демке ASFDemoPool .
Также, если вас не устраивает нарисованная мной анимация прогрессбара, то но событии **onWaitModeCustomDraw** вы можете нарисовать свою. В обработчик передается канва (TGPGraphics), регион (TGPRegion), в котором будет происходить рисование (он соответствует размеру текущей секции) и переменная var DefaultDraw будет говорить дефолтному обработчику, рисовать ли мою анимацию после вашей или нет.

Ну вот, собственно, и всё. Теперь, думаю, надо сесть за хелп, т.к. класс действительно разросся и даже я сам начинаю забывать свойства и методы.

{jcomments on}