

Лёвкин Вадим Евгеньевич

канд. филос. наук, доцент

ФГАОУ ВО «Тюменский государственный университет»

г. Тюмень, Тюменская область

СОЗДАНИЕ СТИМУЛЬНОГО МАТЕРИАЛА ДЛЯ НАУЧНЫХ ИССЛЕДОВАНИЙ В ПРОГРАММНОМ РЕДАКТОРЕ НЕОВООК

Аннотация: в общественных и гуманитарных науках применение экспериментальных методов исследования затруднено не только в силу специфики предмета исследования (активность и формируемость психической, культурной реальности), но и потому, что исследователи не информированы о возможностях современных программных средств, позволяющих осуществлять контролирование заданных переменных. В настоящей статье вниманию гуманитариев предложен инструмент, уже несколько лет применяемый в ТюмГУ и позволяющий проводить исследования констатирующего и формирующего характера – это программный редактор NeoBook, предназначенный для создания компьютерных программ непрофессиональными программистами. На примере организации исследования визуального восприятия с помощью программного тахистоскопа показано, как просто и эффективно можно обеспечить стимульным материалом экспериментальное исследование. По приведенной в статье инструкции можно создать программный тахистоскоп за один час, впервые в жизни открыв NeoBook.

Ключевые слова: стимульный материал, NeoBook, эксперимент в общественных науках, эксперимент в гуманитарных науках, программирование с нуля, тахистоскоп, информатика для гуманитариев.

Благодарности. «Проект-победитель» Грантового конкурса Стипендиальной программы Владимира Потанина 2016/2017.

Введение. Поле исследовательских вопросов в социально-гуманитарных науках чрезвычайно обширно. Заранее невозможно не только заготовить, но даже вообразить все необходимые средства материально-технического

обеспечения для проведения научных экспериментальных исследований в науках о человеке и обществе. Исследователи, столкнувшиеся с такого рода ограничениями в своем конкретном исследовании, часто самостоятельно создают как стимульный материал, так и средства фиксации результатов эксперимента, что требует наличия специальных технических навыков и оборудования для изготовления средств проведения эксперимента, либо требует дополнительных затрат для оплаты работы соответствующего мастера-изготовителя экспериментальной установки. Например, первые исследования скорости зрительного восприятия на тахистоскопе могли быть воспроизведены или продолжены только при наличии соответствующего приспособления – тахистоскопа (прибора, позволяющего демонстрировать изображения заданное время). В простом варианте тахистоскоп – это непрозрачная панель с прямоугольным окошком, в котором на краткое и заранее определенное время появляется изображение. Картинка предъясвляется на заданное время с помощью различных средств. Это может быть падающая пластина с прорезью или вращающийся барабан, на котором закреплены изображения. Со временем простые механические устройства сменились сложными оптическими проекторами с целой системой дополнительных приспособлений и электронной начинкой.

Однако в настоящее время задача демонстрации изображений испытуемому многократно облегчается, если воспользоваться компьютером и выводить изображение на экран монитора. При этом процесс получает новые степени свободы (количество, размер, цветность и другие характеристики изображений, гибкость настройки интервалов предъясвления стимулов, и мн. др.), которые могут быть проконтролированы и оперативно изменены в эксперименте. Возникающие при использовании компьютера трудности обычно сводятся к умению подключать к совместной работе профессиональных программистов и налаживать с ними эффективную коммуникацию. Именно эта трудность зачастую оказывается непреодолимой.

К счастью, в начале XXI века достаточно развились программные редакторы – это компьютерные программы, напоминающие текстовые и графические

редакторы, но позволяющие создавать не текстовый или графический файл, а исполняемый файл программы. Важно, что при работе с такими программными редакторами не требуются профессиональные знания в области программирования или какие-либо глубокие технические знания. Достаточно изучить интерфейс программы и несложный скриптовый язык, овладеть которым можно ровно настолько, насколько требуется в данном текущем проекте.

После анализа различных программных редакторов для целей обучения гуманитариев основам программирования в ТюмГУ была избрана среда быстрой разработки приложений – NeoBook [2]. Разработчик NeoBook: Дэвид Райлей, компания Неософт [4].

На примере программного редактора NeoBook покажем, как можно создать элементарный программный тахистоскоп. Для того, чтобы познакомиться со средой NeoBook достаточно будет выполнить все описанные ниже действия, получив на выходе программный тахистоскоп, которым можно пользоваться в реальных научных исследованиях. Созданная на пробной версии NeoBook программа будет работать только на компьютере с установленной NeoBook (в режиме тестовой компиляции), а программа, собранная (скомпилированная) на лицензионной версии, сможет работать на любом компьютере с Windows и для своей работы не требует установленной NeoBook. Следует отметить, что пробная версия (которую можно скачать с сайта разработчика) полнофункциональна и позволяет осваивать NeoBook неограниченно долго по времени, при этом разработчик предоставляет очень значительные скидки для образовательных учреждений, что делает покупку лицензий доступным мероприятием даже для самых малых образовательных учреждений.

Создание программного тахистоскопа в NeoBook

Для начала необходимо установить NeoBook. Детальная инструкция (в т.ч. с видеоуроком) размещена в форме Практикума №1 на странице разработанного автором настоящей статьи бесплатного онлайн курса, полностью посвященного программированию в NeoBook [1]. Издан и учебник по NeoBook [3].

Благодаря Петру Павлову NeoBook имеет дружественный русскоязычный интерфейс [5], что очень помогает в процессе овладения программой. Запустив NeoBook Вы обнаружите сверху главное меню, а под ним панель инструментов с кнопками быстрого доступа к командам и кнопками выбора создаваемого объекта. Отдельно размещены окна панели стилей, списка объектов, крупного вида. Подробнее об интерфейсе NeoBook Вы сможете узнать из Справки к ней [1], а также из Практикума №1 [1].

Итак, откройте программный редактор NeoBook и создайте новый проект, после чего сразу сохраните его. Будет создан обычный текстовый файл, но с расширением *.pub*, в котором NeoBook будет хранить описание используемых нами объектов (рисунков, текста) и команд, которые должны будут исполниться при наступлении запрограммированного нами события (например, клика левой кнопкой мыши по пункту главного меню нашего приложения). При финальной компиляции проекта, т.е. в момент создания уже исполняемого файла программы, NeoBook соберет в единый файл (с расширением *.exe*) и файл проекта, и все прочие детали проектируемого приложения (в нашем случае – файлы картинок) и превратит всё это в самостоятельное работающее приложение.

Чтобы создать проект в главном меню NeoBook выберите команду «Файл\Новый». В появившемся окне выбора размера главного окна программы просто нажмите кнопку ОК, тем самым согласившись с размером 640 x 480 пикселей (при желании этот размер потом можно изменить). В рабочей области программного редактора NeoBook мы увидим пока пустое окно будущей программы – это область, в которой будут отображаться картинки с заданной экспозицией.

Чтобы сохранить проект выберите команду «Файл\Сохранить». В появившемся окне выберите место для сохранения файла проекта. Лучше всего для каждого проекта создавать отдельную папку и в ней хранить как файл проекта, так и прочие «запчасти» (в текущем проекте это будут файлы изображений). Такую папку можно хранить в любом удобном месте на диске компьютера, например, в папке «Мои документы» или в папке с установленной NeoBook.

Представим, что нас в будущем эксперименте будет интересовать вопрос: с какой скоростью нужно предъявлять изображения, чтобы испытуемый успевал опознавать фрукты?

Возьмём для примера 10 изображений, хотя их может быть и 10 тысяч, сколько угодно. Скачать изображения со свободной лицензией можно, например, с сайта pixabay.com (к каждому изображению указан тип лицензии). Сами изображения поместим в папку с проектом и переименуем простыми числами от 1 до 10 (это удобно тем, что мы можем таким образом отметить последовательность предъявления картинок, если это в эксперименте имеет значение).

Теперь поместите на пока что пустое окно программы объект «Рисунок». Для этого нужно единожды кликнуть по иконке «Рисунок» (см. всплывающие подсказки к иконкам на панели инструментов), после чего наметить прямоугольник в том месте окна будущей программы, где Вы желаете получить изображение. Для того, чтобы наметить прямоугольник нажмите левую кнопку мыши в окне создаваемой программы и проведите мышью по диагонали, после чего отпустите левую кнопку мыши. Так в NeoBook создаются и другие объекты, выбираемые на панели инструментов.

В открывшемся окне выбора файла изображения впишите *0.jpg* (это имя файла изображения, но файла с таким именем нет в папке с проектом, поэтому объект «Рисунок» хотя и будет создан, но ничего не отобразит, что нам и требуется, поскольку изображения мы будем показывать потом, по специальной команде, которую пропишем позже).

Нажав кнопку «Открыть» получим серый прямоугольник на месте будущего изображения. Прямоугольник объекта «Рисунок» серый, поскольку именно такой цвет заливки был выбран в программе по умолчанию, и он имеет черную рамку, поскольку такие свойства объекта «Рисунок» установлены по умолчанию (если, конечно, Вы не успели изменить их). Цвет заливки во время работы приложения будет заменен на само изображение и может отображаться полоской сверху и полоской снизу только если пропорции изображения будут отличаться от пропорций окна создаваемого нами приложения. А вот рамку мы можем

убрать прямо сейчас, если она не нужна. Для этого достаточно в разделе «Рамка» (на панели стилей) выбрать вместо линии значение «нет».

Чтобы мы могли в объекте «Рисунок» показывать картинку разного фактического размера, автоматически подгоняя их под размер объекта «Рисунок» нужно сделать следующее: нажмите по созданному объекту «Рисунок» правой кнопкой мыши, появится окно настройки свойств этого объекта. В разделе «Вид», в параметре «Режим отображения» выберите свойство «Авто» (чтобы размер рисунка автоматически увеличивался или уменьшался до размеров объекта «Рисунок») и отметьте галочкой свойство «Центрировать рисунок» (это позволит автоматически размещать изображение по центру окна создаваемого приложения). Не закрывая этого окна, в разделе «Общие» в параметре «Привязка» выберите свойство «По центру» (для того, чтобы объект «Рисунок» автоматически подстроился под размер окна, заполнив всё его пространство). Нажмите ОК.

В результате, объект «Рисунок» растянется на всё окно проектируемого приложения. Будем считать, что «окошко» нашего тахистоскопа готово. Испытуемому есть куда смотреть, однако теперь нужно сделать так, чтобы в этом «окошке» отображался ряд изображений с заданной скоростью.

Представим, что нам нужно, чтобы изображения предъявлялись на заданное нами время (в миллисекундах) при нажатии кнопки на экране или клавиатуре. Создадим в самом низу окна приложения эту кнопку и назначим ей, для удобства, горячую клавишу.

Чтобы создать кнопку нужно выбрать инструмент «Кнопка», т.е. единожды кликнуть на панели инструментов NeoBook по соответствующей иконке (см. всплывающие подсказки к кнопкам на панели инструментов NeoBook), а затем наметить прямоугольник там, где планируется разместить кнопку (так же, как мы намечали прямоугольник при создании объекта «Рисунок»).

В открывшемся окне настройки свойств кнопки в параметре «Заголовок» напишите ПУСК, это будет надпись на кнопке, в поле «Горячая клавиша» поставьте курсор и нажмите на клавиатуре клавишу «Пробел» (появится надпись Space), а в выпадающем списке «Привязка» выберите пункт «Снизу», чтобы

кнопка растянулась по всей нижней части окна приложения и «прилепилась» к ней. Не закрывая этого окна, перейдите в раздел «Вид» и выберите «Стиль кнопки» – «Тема Windows» (просто для того, чтобы она смотрелась симпатичнее). Нажмите ОК, чтобы сохранить изменения. Вы получите в окне NeoBook изображение кнопки внизу окна создаваемого приложения, во всю его ширину.

Чтобы по нажатию кнопки (или её горячей клавиши) выполнялись какие-то команды их нужно прописать. Для этого кликните по созданной кнопке «ПУСК» правой кнопкой мыши и в открывшемся окне настройки свойств объекта перейдите в раздел «Команды», на вкладку «Левый клик мышью» (Left Click). Все команды, прописанные на этой вкладке, будут исполняться при каждом клике левой кнопкой мыши по данному объекту «Кнопка».

Чтобы объект «Рисунок» отобразил нужный нам файл изображения ему нужно указать имя этого файла. Для этого воспользуемся командой `SetObjectFileName`, позволяющей изменить имя файла объекта. Пока команды не известны на память удобно пользоваться «Селектором команд» (т.е. их упорядоченным списком с подсказками, который вызывается по нажатию кнопки «Команды» в правой верхней части окна редактора кода, например, на вкладке «Left Click»).

Щелчок по имени команды в списке команд автоматически открывает окно помощника настройки параметров этой команды. В открывшемся окне щелкните по иконке, напоминающей шестеренку, и выберите имя объекта «Рисунок» (в данном случае это «Picture1»).

А в качестве имени файла впишите имя первого файла рисунка `[PubDir]1.jpg`, перед которым (без пробела) в квадратных скобках указана «глобальная переменная» `[PubDir]` («Директория проекта»). Эта переменная всегда хранит актуальный путь к той папке (директории) в которой лежит запущенная на исполнение программа (а поскольку мы решили хранить изображения рядом с файлом самой программы, то переменная `[PubDir]` будет указывать одновременно и на папку с рисунками, ведь они лежат в той же папке в какой и запускаемый файл проекта).

Нажав кнопку ОК получаем следующую запись в редакторе кода:

```
SetObjectFileName «Picture1» «[PubDir]1.jpg»
```

Первое слово в строке – это команда, она подсвечивается синим цветом, а далее в кавычках указаны параметры этой команды, т.е. то, что можно настроить или изменить в работе команды.

Далее, чтобы создать паузу на заданное число миллисекунд (перед показом следующего изображения) воспользуемся командой Delay («Пауза»). В селекторе команд эта команда находится в разделе «Управление».

Кликнете по имени команды. В автоматически открывшемся окне настройки параметров команды впишите вместо конкретного числа миллисекунд имя переменной [Пауза], чтобы иметь в будущем возможность оперативно изменять длительность паузы.

В NeoBook имена переменных можно придумывать любые и даже называть их по-русски (как в приведенном примере), кроме тех имен, которые уже зарезервированы самой NeoBook (они называются «Глобальные переменные», как уже известная нам переменная [PubDir]), но имена переменных в любом случае пишутся всегда в квадратных скобках). Полный список глобальных переменных легко найти в Справке к NeoBook, которая полностью русифицирована благодаря Сергею Ионову.

Нажав ОК в окне настройки команды Delay получим строку с кода с паузой:

```
Delay «[Пауза]»
```

Аналогичным образом добавим команды для показа остальных рисунков с такой же паузой между ними (только в самой последней команде SetObjectFileName уберём имя рисунка, чтобы получить снова просто серый фон, для чего достаточно оставить пустыми кавычки обозначающие последний параметр команды):

```
SetObjectFileName «Picture1» «[PubDir]1.jpg»
```

```
Delay «[Пауза]»
```

```
SetObjectFileName «Picture1» «[PubDir]2.jpg»
```

```
Delay «[Пауза]»
```



```

SetObjectFileName «Picture1» «[PubDir]3.jpg»
Delay «[Пауза]»
SetObjectFileName «Picture1» «[PubDir]4.jpg»
Delay «[Пауза]»
SetObjectFileName «Picture1» «[PubDir]5.jpg»
Delay «[Пауза]»
SetObjectFileName «Picture1» «[PubDir]6.jpg»
Delay «[Пауза]»
SetObjectFileName «Picture1» «[PubDir]7.jpg»
Delay «[Пауза]»
SetObjectFileName «Picture1» «[PubDir]8.jpg»
Delay «[Пауза]»
SetObjectFileName «Picture1» «[PubDir]9.jpg»
Delay «[Пауза]»
SetObjectFileName «Picture1» «[PubDir]10.jpg»
Delay «[Пауза]»
SetObjectFileName «Picture1»

```

Каждая команда в NeoBook пишется с новой строки. Имя команды в редакторе кода автоматически подсвечивается синим цветом, а имена переменных – красным.

Конечно, проще было бы реализовать то же самое через так называемый цикл. Например, с помощью команды Loop и команды завершения цикла EndLoop. Тогда код выглядел бы так (и выполнял бы ровно то же самое):

```

Loop «1» «10» «[i]»
SetObjectFileName «Picture1» «[PubDir][i].jpg»
Delay «[Пауза]»
EndLoop
SetObjectFileName «Picture1»

```

Но и предыдущий вариант не слишком громоздкий и его не трудно написать, поэтому, если циклы Вам пока не понятны, можно писать так, как понятно

на данный момент (хотя все команды NeoBook прекрасно описаны в Справке). Всё же, главное в программе – это чтобы она работала корректно, т.е. делала то, ради чего создавалась. А уж насколько кратким окажется код – это при современных мощностях компьютеров уже не важно.

Нам осталось сделать так, чтобы переменная [Пауза] получала то значение, которое мы решим оперативно установить прямо во время работы приложения. Как и любую другую задачу, эту можно решить разными способами. Воспользуемся одним из них.

Сделаем в главном меню нашей программы команду «Настройки», чтобы клик по этому меню вызывал окошко, в котором мы могли бы изменять число миллисекунд, управляя, таким образом, длительностью паузы.

Для этого в главном меню самой NeoBook выберите команду «Проект\Свойства проекта\Главное меню». В открывшемся окне (в крайней правой и верхней его части) нажмите кнопку «Добавить заголовок меню» (см. всплывающую подсказку).

Появится созданное меню, но со служебным заголовком, который замените на свой, например, вписав в поле «Заголовок» слово «Настройки».

Чтобы созданный заголовок меню при клике по нему делал работу нужно прописать соответствующие команды. Для этого в том же окне «Свойства проекта\Главное меню» перейдите на вкладку «Команды» и через «Селектор команд» выберите в разделе «Сообщения/Диалоги» команду InputBox, которая будет выводить окошко с полем ввода.

В открывшемся окне помощника настройки параметров команды InputBox введите в поле «Заголовок» слово «Настройки», в поле «Сообщение» введите текст «Требуемое число миллисекунд (1 сек = 1000 мс)», а в поле «Сохранить результат в переменной» введите имя переменной: [Пауза]. Пока не закрывайте это окно. Если теперь нажать на кнопку «Тест» (она находится в левом нижнем углу окна настройки параметров команды InputBox), то можно увидеть, как будет выглядеть окно, создаваемое командой InputBox. Нажмите кнопку «Тест», чтобы проверить как это работает.

Теперь, после нажатия кнопки ОК, но уже в работающем приложении, переменная [Пауза] примет указанное в данном поле число миллисекунд, которое будет действительным на протяжении всего текущего запуска программы, пока мы снова не поменяем это значение. А менять его мы теперь можем сколько угодно раз.

Нажмите ОК последовательно для того, чтобы сохранить все изменения. Теперь в окне создаваемого приложения появилось главное меню «Настройки». Сейчас (в режиме редактирования) надпись «Настройки» тусклая и не реагирует на клик мышью. Однако, это меню будет кликабельным во время тестового запуска или после финальной компиляции проекта, которая делается командой меню «Проект\Компиляция...» для того, чтобы получить самостоятельный исполняемый файл созданной программы.

Теперь мы можем осуществить тестовый запуск и проверить – так ли работает приложение, как мы задумали. Чтобы осуществить тестовый запуск выберите команду «С начала проекта» в меню «Проект» (или нажмите горячую клавишу F9).

Откроется приложение (при этом в тестовом запуске всегда присутствует и окно отладчика, которое позволяет видеть последовательность исполняемых команд и текущие значения переменных, что очень удобно при отладке программ). В открывшемся по команде тестового запуска приложении нажмите на меню «Настройки» и введите значение 100. После чего нажмите ОК. Теперь, если Вы нажмёте кнопку ПУСК (или клавишу «Пробел» на клавиатуре), то картинки будут отображены по порядку с паузой в одну десятую секунды (100 мс).

Уже можно попробовать продемонстрировать испытуемым изображения с вопросом «Сколько было показано фруктов и каких именно?». Постепенно увеличивая длительность можно добиться того, что испытуемые смогут уверенно называть фрукты, что даст нам достоверную информацию о том, с какой скоростью человек может опознать изображение. Собрав информацию по многим испытуемым, мы сможем выявить скорость характерную для большинства из них. А если параллельно проверим испытуемых на тип темперамента, то сможем

проверить гипотезу о том, что скорость опознания визуальных образов связана с типом темперамента. Например, мы смогли бы ответить на вопрос: верно ли, что холерики опознают визуальные образы быстрее флегматиков? И если разница есть, то существенна ли она? Например, в отношении распознавания дорожных знаков во время движения на автомобиле?

Чтобы изменить заголовок, который отображается в верхней части окна нашей программы нужно вернуться в режим редактирования (для этого просто закройте запущенное для тестирования приложение) и в главном меню NeoBook выбрать команду «Проект\Свойства проекта\Общие». В открывшемся окне остаётся лишь заменить служебный заголовок «Безымянный проект NeoBook» на свой, например, «Тахистоскоп».

Итак, мы только что сделали программу «Тахистоскоп» своими руками. Осталось лишь произвести финальную компиляцию, чтобы получить самостоятельный файл программы (меню «Проект\Компиляция...»). Если действовать по приведенной инструкции на весь проект у Вас уйдёт не больше часа неспешной работы (и это уже при первом знакомстве с NeoBook!).

Ясно, что Вы не ограничены числом изображений, их размером, можете теперь выбирать любой период экспозиции для показа изображений испытуемому, для запуска тахистоскопа достаточно нажать клавишу «Пробел». Это удобно. Однако, можно и дальше развивать программу. В следующих версиях можно добавить больше полезных функций. Например, можно добавить горячую клавишу на команду меню «Настройки» (для быстрого вызова окошка с настройкой времени паузы), можно научить программу распознавать не заранее известное число изображений, а все те изображения, которые окажутся в папке с ней. Можно добавить кнопку остановки показа (на случай, если потребуется остановить длинную демонстрацию). Можно в начале добавить страницу опроса для сбора анкетных данных испытуемого и в конце – страницу с вопросами обратной связи, научить программу сохранять все полученные данные в файл, добавить в качестве стимульного материала звуки и многое другое. Однако, созданный

программный тахистоскоп полезен и в том виде в каком уже есть, поскольку избавляет от необходимости покупать или создавать сложные физические устройства.

Заключение. Итак, на примере создания программного тахистоскопа видно, насколько просто и быстро можно создавать стимульный материал для научных исследований в общественных и гуманитарных науках. Кроме того, NeoBook позволяет с легкостью автоматизировать обычные бланковые методики (тесты, опросники, анкеты и т. д.) и создавать новые (в т.ч. используя аудио и видео элементы), существенно облегчая труд исследователя и предотвращая возможные ошибки ручной обработки данных.

Список литературы

1. Быстрое программирование с нуля для гуманитариев / Разработчик курса В.Е. Лёвкин. Свободный онлайн курс по программированию в NeoBook [Электронный ресурс]. Режим доступа: <https://distant.orgpsiholog.ru/course/view.php?id=20> (короткая ссылка: <https://goo.gl/K6H47f>).

2. Лёвкин В.Е. Обучение магистрантов-психологов инновационным технологиям: возможности среды программирования NeoBook // В мире научных открытий. – 2014. – №12.2 (60). – Красноярск: Научно-инновационный центр, 2014. – 360 с. [Электронный ресурс]. – Режим доступа: <http://orgpsiholog.ru/neobook-programming-the-humanities.htm>

3. Лёвкин В.Е. NeoBook. Быстрое программирование с нуля для гуманитариев: учебник. – М.; Берлин: Директ-Медиа, 2016. – 218 с. [Электронный ресурс]. – Режим доступа: <http://biblioclub.ru/index.php?page=book&id=450198>

4. Официальный сайт среды программирования NeoBook / Разработчик языка NeoBook: Дэвид Райлей, компания Неософт (David Riley, NeoSoft Corporation) [Электронный ресурс]. – Режим доступа: www.neosoft.com

5. Форум «Русский дом для NeoBook» [Электронный ресурс]. – Режим доступа: <http://neobooker.ru> NeoBook