

Использование ПО PXLab для подготовки экспериментов, включающих регистрацию движений глаз с помощью айтрекеров Sensomotoric Instruments

Александр Владимирович Жегалло

Институт психологии РАН, Москва, Россия

Аннотация. Психологический эксперимент предполагает предъявление стимульного материала и последующую фиксацию ответов испытуемого. Компьютеризованное предъявление изображений и текста с автоматической фиксацией ответов участников может быть реализовано, в частности, с помощью ПО PXLab с открытым исходным кодом. Выполненная доработка PXLab обеспечивает регистрацию движений глаз с использованием оборудования Sensomotoric Instruments. Описываются возможности доработанного программного обеспечения PXLab в части проведения айтрекинговых исследований с автоматической фиксацией ответов испытуемых. Подробно рассматривается пример реализации экспериментального исследования, позволяющий освоить конструирование простых экспериментов с регистрацией движений глаз и ответов испытуемого. Доработанная версия PXLab и примеры использования доступны на <https://sites.google.com/site/azhegallo/main/pxlab>.

Контактная информация: Александр Владимирович Жегалло, zhegs@mail.ru; ул. Ярославская, д.13, Институт психологии РАН, 129366 Москва, Россия.

Ключевые слова: регистрация движений глаз, экспериментальные психологические исследования

© 2016 Александр Владимирович Жегалло. Данная статья доступна по лицензии [Creative Commons "Attribution" \(«Атрибуция»\) 4.0. всемирная](https://creativecommons.org/licenses/by/4.0/), согласно которой возможно неограниченное распространение и воспроизведение этой статьи на любых носителях при условии указания автора и ссылки на исходную публикацию статьи в данном журнале в соответствии с канонами научного цитирования.

Благодарности. Работа выполнена в рамках госзадания ФАНО РФ № 0159-2016-0004.

Статья поступила в редакцию 16 марта 2016 г. Принята в печать 28 сентября 2016 г.

Простейший психологический эксперимент с регистрацией движений глаз предполагает экспозицию стимульного материала на время, достаточное для его рассматривания, и фиксацию некоторого ответа участника, характеризующего рассматриваемое изображение. Традиция проведения подобных исследований восходит к работам А.Л. Ярбуса. При анализе характера рассматривания сложных объектов отмечалось, что глаз останавливается лишь на отдельных элементах

изображения, содержащих «сведения полезные и нужные в момент восприятия» (Ярбус, 1965, с. 125). Далее отмечается, что «в зависимости от задач, которые стоят перед человеком, то есть в зависимости от характера сведений, которые он должен получить, будет соответственно изменяться и распределение точек фиксации на объекте, поскольку различные сведения обычно локализованы в различных частях объекта» (Ярбус, 1965, с. 143). Данное положение иллюстрировалось записями

движений глаз при рассматривании картины И.Е. Репина «Не ждали» при различных инструкциях участникам исследования. Таким образом, обязательное наличие инструкции, даваемой наблюдателю, и фиксации ответов, даваемых в рамках ее выполнения, стали фактическим стандартом при проведении исследований, включающих регистрацию движений глаз.

В последние 10 лет среди российских исследователей широкой популярностью пользуются айтрекеры производства Sensomotoric Instruments (SMI). Линейка оборудования включает стационарную высокоскоростную систему SMI High Speed с частотой регистрации до 1250 Гц в монокулярном режиме и 500 Гц в бинокулярном, используемую с лобно-подбородной опорой; систему RED 250/500, совместимую с компьютерными, телевизионными и проекционными экранами различных размеров и допускающую относительно свободное размещение испытуемого; компактные модули RED250 mobile и REDn scientific; мобильный айтрекер на очковой оправе SMI eyetracking glasses. Вместе с оборудованием может быть приобретено программное обеспечение SMI Experiment Suite™ 360°, включающее SMI Experiment Center™ для проведения экспериментов и SMI BeGaze™ для анализа данных. При покупке программного обеспечения от исследователя требуется выбрать отдельные функциональные элементы, обеспечивающие проведение конкретных типов экспериментов и вариантов анализа данных, причем стоимость приобретаемого программного обеспечения может оказаться сопоставимой со стоимостью оборудования.

В простейшем случае при проведении экспериментального исследования необходима последовательная отработка следующих этапов: предъявление инструкции, предъявление целевого стимульного материала, фиксация ответа. Предъявление инструкции в виде текста на экране и последующая экспозиция целевого изображения не представляют существенных затруднений. В то же время задача фиксации ответа при организации эксперимента с помощью SMI Experiment Center™ решается, на наш взгляд, не вполне адекватно. Элемент конструктора, поддерживающий фиксацию ответов, не допускает изменения шрифтов и расположения текста вопросов. Результат ответа в айтрекинговых данных фиксируется в виде координат нажатия «мыши», что неудобно для дальнейшей обработки. Результат ответа в виде сделанного выбора доступен средствами штатного ПО SMI BeGaze™ и с его использованием может быть экспортирован для дальнейшей обработки сторонним ПО. При жестких требованиях к эргономическим характеристикам отображаемого текста исследователи могут отказаться от использования штатных средств в пользу самостоятельно формируемого экрана со списком вариантов ответов и фиксации реплик испытуемых на диктофон (Бессонова и др., 2015).

При таком подходе происходит потеря информации о точном моменте ответа, что в дальнейшем сужает возможности обработки результатов исследования и предполагает повторный ручной ввод данных для последующего статистического анализа. Автоматическая фиксация ответов, которые дают участники исследования, требует использования специализированного ПО, ориентированного на проведение стан-

дартных психологических исследований. В то же время данное ПО должно поддерживать взаимодействие с аппаратно-программным комплексом, осуществляющим регистрацию движений глаз. Как минимум необходимы следующие операции: запуск регистрации, ее остановка, сохранение результатов в файл, простановка отметок о предъявляемом стимульном материале и действиях участников исследования. В случае айтрекеров SMI такое взаимодействие может быть реализовано с помощью программного обеспечения iView X™ SDK, включающего интерфейсы для популярных программ, предназначенных для проведения психофизических исследований (E-Prime, NBS Presentation, Psychophysics Toolbox, PsychoPy) и API для ряда универсальных языков программирования (C/C++; Python; C#; MATLAB). Также поддерживается взаимодействие с айтрекерами на уровне управляющих команд (Remote Commands), отправляемых по протоколу UDP (iView X System Manual, 2011).

Проблема организации айтрекинговых экспериментов обсуждалась в ходе дискуссии, проходившей на конференции «Айтрекинг в психологической науке и практике» (Москва, 16–17 октября 2015 г.). Можно утверждать, что имеющаяся у исследователей потребность в организации простейших экспериментов, включающих экспозицию статических изображений и последующую автоматическую фиксацию ответов, не покрывается возможностями программного обеспечения SMI Experiment Center™. Целесообразность приобретения дополнительного коммерческого программного обеспечения (E-Prime, NBS Presentation, EventIDE и др.) определяется конкретными целями и задачами исследователя. Высокая стоимость дополнительного программного обеспечения, его относительно малая распространенность, существенные различия в тематике исследований, относительная ограниченность коммуникации с другими исследовательскими группами не позволяют сформировать исчерпывающие рекомендации по оптимальному его использованию. Наш собственный опыт использования E-Prime как среды для подготовки айтрекинговых исследований следует признать скорее неудачным. Единственным преимуществом E-Prime по сравнению со штатным ПО оказалась возможность точного контроля времени экспозиции. Удобный интерактивный способ фиксации ответов в E-Prime не предусмотрен в силу ориентации на психофизические, но не общепсихологические эксперименты.

В качестве удачного примера использования дополнительного ПО можно указать использование EventIDE в айтрекинговом исследовании, направленном на изучение восприятия постепенно проявляющихся изображений (Koltunova, Podladchikova, 2013). Следует отметить, что в данном случае успех был достигнут в значительной степени за счет эффективного взаимодействия с разработчиком ПО.

При наличии у экспериментатора навыков программирования возможны весьма гибкие реализации взаимодействия с оборудованием SMI, обеспечивающие требуемую в конкретном исследовании функциональность. В исследованиях, проводящихся нами и нашими коллегами в Центре эксперименталь-

ной психологии МГППУ, данный способ работы получил широкое распространение. Так, с помощью пакета PsychoPy (Peirce, 2007) успешно была реализована фиксация ответов испытуемых по ходу просмотра видеофрагментов (Жегалло, Хозе, 2015) и парный айтрекинг-эксперимент с синхронным предъявлением стимульного материала двум участникам (Ананьева, Басюл, Харитонов, 2015). На языке Action Script была реализована программа для предъявления стимулов во время саккады (Жердев, Барабанщиков, 2014). Существенным недостатком такого подхода является практическая невозможность изменения функциональности и использования программ без помощи автора оригинальной разработки.

Завершая обзор имеющихся средств проведения айтрекинг-экспериментов с использованием оборудования SMI, сопоставим возможности различных вариантов решения данной задачи. Штатное ПО SMI Experiment Center™ характеризуется высокой стоимостью, удовлетворительной функциональностью, низким порогом освоения, простой повторяемостью экспериментальных исследований. Дополнительное коммерческое ПО (E-Prime, Presentation, Event IDE) характеризуется высокой стоимостью (приобретается, как правило, в дополнение к штатному ПО), средним порогом освоения, сложной повторяемостью экспериментальных исследований. Использование языков и сред программирования общего назначения требует от экспериментатора навыков программирования, характеризуется высоким порогом вхождения, максимальной функциональностью и крайне высокой сложностью повторения экспериментальных исследований.

Целью настоящей статьи является описание альтернативного способа подготовки айтрекинг-экспериментов. Предлагаемый способ характеризуется низкими материальными затратами, средним порогом освоения (сопоставимым с порогом освоения коммерческого ПО), средней функциональностью (в некоторых аспектах заведомо превосходящей штатное ПО), удобством воспроизведения экспериментальных исследований.

В качестве основы для реализации экспериментальных айтрекинг-исследований использовано ПО PXLab (Irtel, 2007) с открытым исходным кодом. PXLab представляет собой коллекцию классов на языке Java, реализующую широкий спектр визуальных компонент, применяемых в психологических экспериментах. Поддерживается вывод текстовых элементов (TextParagraph, TextFile, Message, Instruction, TextInput и др.), простейших геометрических объектов (Arrow, SimpleBar, SimpleDisk, PolyArea, FixationMark и др.), статических изображений (Picture), звуковых сигналов (Beep, SoundFile, SyntheticSound). В состав ПО также входят специализированные классы для изучения цветового зрения, восприятия движения, восприятия пространственного контраста и др. Хорошо проработанная система синхронизации и таймеров обеспечивает точный контроль и высокую стабильность времени экспозиции вплоть до одного кадра видеоразвертки.

PXLab содержит классы, позволяющие эффективно реализовать различные варианты опросников. С помощью класса MultipleChoiceQuestion реализует-

ся выбор одного из нескольких развернутых текстовых вариантов ответа на заданный вопрос. В случае, если вопрос предполагает краткие варианты ответа, например «Да», «Нет», «Не знаю», необходимая функциональность может быть удобно реализована с помощью класса ChoiceResponse. Рейтинговая оценка по дискретной либо непрерывной шкале реализуется с помощью класса RatingScale. Ранжирование объектов, представленных текстовыми строками, по заданному признаку выполняется с помощью класса ItemRanking. В состав ПО входит набор примеров, демонстрирующих широкие возможности реализации экспериментальных исследований.

Структура проводимого эксперимента описывается в специальном файле. Необходимость изучения синтаксиса данного файла, фактически представляющего собой специализированный язык программирования, собственно и задает порог вхождения, требуемый для начала работы. Настоящая статья содержит детальный разбор структуры файла описания эксперимента в объеме, достаточном для реализации простейших экспериментов.

Документация к ПО содержит специальный раздел, адресованный разработчикам, желающим расширить функциональность PXLab, и подробно описывающий детали реализации программы. В большинстве случаев расширение функциональности может быть достигнуто за счет написания дополнительных модулей, без внесения изменений в существующий исходный код.

Модифицированный вариант PXLab и рассмотренные в статье примеры¹ реализации айтрекинг-экспериментов доступны для скачивания по адресу <https://sites.google.com/site/azhegallo/main/pxlab> и в [онлайн-приложении](#) к статье.

Краткая характеристика выполненной доработки ПО PXLab

Для обеспечения совместной работы с айтрекером SMI нами реализована дополнительная компонента SendUDP(), выполняющая отправку заданного UDP-пакета на заданный порт на заданном IP-адресе. С ее помощью осуществляется передача управляющей информации от PXLab к управляющему ПО айтрекеров SMI. Благодаря этому становится возможной синхронизация записи движений глаз с информацией о смене предъявляемого стимульного материала и действиях, выполняемых испытуемым. Таким образом, становится возможным использование полной функциональности PXLab в эксперименте, включающем регистрацию движений глаз.

Адрес и содержание отправляемого сообщения задаются параметрами UdpHost, UdpPort, UdpMessage. Значения IP-адреса и порта определяются настройками айтрекера. Возможные варианты отправляемого UDP-сообщения описаны в документации на айтре-

¹ По техническим причинам (PDF-файл содержит дополнительные символы перевода строк, приводящие к синтаксическим ошибкам при выполнении программы) фрагменты кода, приведенные в PDF-версии статьи, не могут быть непосредственно выполнены в PXLab. При повторении описываемых экспериментов читателям предлагается скачать текстовые файлы с рассмотренными в статье примерами.

кер. Инициация начала записи движений глаз выполняется отправкой сообщения ET_REC\n (\n — здесь символ перевода строки, завершающий отправляемое сообщение). Информация об экспериментальной ситуации передается сообщением ET_REM data\n. Поле data в простейшем случае представляет собой имя файла с предъявляемым изображением, в общем случае — произвольную информацию, характеризующую процесс развертывания экспериментальной ситуации. Завершение записи движений глаз выполняется путем последовательной отправки сообщений ET_STP\n — остановка записи и ET_SAV filename\n — сохранение результатов в файл. Поле filename представляет собой полное имя файла, в который будет выполняться запись.

Интерактивный выбор вариантов ответа требует наличия на экране курсора мыши. В то же время при рассматривании целевого изображения наличие курсора мыши не желательно. Для управления отображения курсором мыши дополнительно реализовано свойство отображаемых объектов MouseOn.

Процедура калибровки, предшествующая проведению эксперимента, реализована с помощью компоненты CalibrationSMI(). Калибровка выполняется в автоматическом режиме, аналогично работе штатного ПО SMI. Результаты калибровки первой (центральной) точки принимаются вручную, нажатием клавиши «пробел». Число точек калибровки задается в процедуре калибровки. Работа процедуры проверялась на айтрекере SMI High Speed (5 и 9 точек) и на айтрекере SMI RED-m (5 точек). Координаты точек калибровки определяются настройками штатного ПО SMI, обеспечивающего работу оборудования. По окончании калибровки автоматически выполняется валидация результатов. В отличие от штатного ПО SMI последних версий, где валидация выполняется по четырем точкам, являющимся вершинами прямоугольника, находящегося в средней части экрана, в нашей процедуре валидация осуществляется для того же набора точек, что и калибровка, что позволяет оценить точность локализации взора на всей площади экрана. По окончании процедуры валидации результат отображается в виде изображения позиций точек калибровки (черные кружки) и фактических позиций локализации взора испытуемого (красные кресты). Принятие результатов процедуры осуществляется клавишей «пробел», повторное выполнение процедуры калибровки — клавишей «г». Калибровочная точка на экране представляет черный кружок с пульсирующим светлым центром. Реализация процедуры происходит в режиме диалога между PXLab и управляющим айтрекером ПО iViewX (iView X System Manual, 2011). PXLab инициирует процедуру калибровки, посылая команду ET_CAL. В ответ iViewX отправляет информацию о параметрах калибровки: число калибровочных точек, размер экрана, координаты каждой из калибровочных точек и в заключение — указание о том, что необходимо отобразить заданную калибровочную точку. Получив данное указание, управляющая программа отображает соответствующую калибровочную точку. Процедура повторяется последовательно для всех точек, после чего iViewX передает указание о завершении процедуры

калибровки. После завершения процедуры калибровки для проверки ее качества PXLab вновь отображает калибровочную точку в каждой из заданных позиций, после чего посылает команду ET_VLX с указанием координат отображаемой точки. В ответ iViewX анализирует текущие координаты взора наблюдателя и сообщает рассогласование с заданными координатами.

Приемы конструирования экспериментов описаны в руководстве по PXLab. В поставку также входят примеры экспериментов, реализованных с помощью данного ПО. В настоящей статье мы ограничимся разбором реализации простейших айтрекинговых экспериментов.

Структура эксперимента

В качестве примера рассмотрим следующий эксперимент. Участникам исследования последовательно экспонируются фотоизображения различных натурщиков. Необходимо рассмотреть каждое изображение и дать ответ, характеризующий пол (мужчина или женщина) и возраст (молодой или пожилой) каждого натурщика. Перед экспозицией изображения на 1 секунду показывается фиксационный крест в центре экрана. Время экспозиции каждого изображения составляет 3 секунды. После завершения экспозиции на экран выводится вопрос: «Кто изображен на картинке» и четыре варианта ответа: «молодой мужчина», «молодая женщина», «пожилой мужчина», «пожилая женщина». Ответ дается путем выбора мышкой одного из вариантов ответа и нажатия клавиши «пробел». В начале работы показывается инструкция участнику исследования. Завершается исследование появлением надписи «Эксперимент закончен. БОЛЬШОЕ СПАСИБО!». Регистрация движений глаз начинается в момент исчезновения инструкции и завершается в момент появления сообщения о завершении эксперимента. Отметки о смене экспериментальной ситуации для айтрекера отправляются в момент появления фиксационного креста, в момент появления целевого изображения, в момент появления экрана с вариантами ответов и в момент, когда испытуемый дал ответ.

В качестве стимульного материала использованы фотоизображения натурщиков, включенные в дистрибутив PXLab². Как уже отмечалось, для реализации эксперимента нам необходимо создать файл, содержащий его описание в специальном виде, определенном разработчиками PXLab. Описание эксперимента имеет следующую структуру :

```
Experiment() {
    Context() {
        // описание контекста эксперимента
    }
    Procedure() {
        // описание процедуры выполнения
    }
}
```

² Следует отметить, что штатный пример, использующий данные изображения, представляет собой более сложный эксперимент — тест имплицитных ассоциаций (см. файл описания эксперимента iat.pxd в дистрибутиве). Доработку указанного примера, связанную с обеспечением регистрации движений глаз, оставляем читателям в качестве самостоятельного упражнения.

Основное назначение раздела `Context` — описание последовательности стимулов, предъявляемых во время отдельной экспериментальной ситуации. Также в разделе `Context` выполняется задание значений глобальных параметров. Раздел `Procedure` служит для задания последовательности сессий, блоков и экспериментальных ситуаций в ходе выполнения эксперимента. В случае простейшего айтрекингового эксперимента раздел `Context` будет содержать следующие четыре подраздела: `AssignmentGroup`, `Block`, `BlockEnd`, `Trial`.

Раздел `AssignmentGroup` включает инициализацию глобальных переменных.

```
AssignmentGroup() {
    ExperimentName = "Simple Eyetracking study";
    SkipBoundingBlockDisplays = 0;
    DataFileTrialFormat = "%SubjectCode%,
%Trial.Picture.FileName%,
%Trial.MultipleChoiceQuestion.Selection%,
%Trial.Picture.ResponseTime%,
%Trial.MultipleChoiceQuestion.ResponseTime%";
    RandomizeTrials = 1;
    new iViewHost = "127.0.0.1";
    new iViewPort = 4444;
    new ImgDir =
"C:\pxlab\pxd\images\old_young\";
    ScreenBackgroundColor = lightGray();
    StopKey = de.pxlab.pxl.KeyCodes.ESCAPE_KEY;
    new CalibrationFormat = "%SubjectCode%,
%Trial:C.CalibrationSMI.vPointsX%,
%Trial:C.CalibrationSMI.vPointsY%,
%Trial:C.CalibrationSMI.gPointsX%,
%Trial:C.CalibrationSMI.gPointsY%";
}
```

Назначение переменных:

`ExperimentName` — название эксперимента.

`SkipBoundingBlockDisplays` — указание на то, что не следует пропускать описания в разделах `Block()` и `BlockEnd()` первого и последнего блоков секции.

`DataFileTrialFormat` — формат выходного файла (основной блок исследования).

`RandomizeTrials` — указание предъявлять экспериментальные ситуации в случайном порядке

`iViewHost` — IP-адрес компьютера, на котором работает управляющее ПО айтрекера.

`iViewPort` — номер порта компьютера, на котором управляющее ПО айтрекера ведет прием команд

`ImgDir` — адрес папки, в которой содержатся предъявляемые изображения.

`ScreenBackgroundColor` — цвет фона.

`StopKey` — выделенная клавиша, код которой отрабатывается в специальных случаях средствами PXLab (см. документацию PXLab).

`CalibrationFormat` — формат выходного файла (процедура калибровки).

Процедура калибровки определяется разделами `Block:C()`, `BlockEnd:C()`, `Trial:C()`. Первые два соответствуют операциям, которые можно было бы выполнять непосредственно до калибровки и после ее заверше-

ния, в данной реализации — пустые. Раздел `Trial:C()` включает единственный подраздел `CalibrationSMI()`, описывающий характеристики проводимой процедуры калибровки.

```
Trial:C(DataFileTrialFormat,CalibrationSMI.
vPointsX,CalibrationSMI.vPointsY,CalibrationSMI.
gPointsX,CalibrationSMI.gPointsY) {
    CalibrationSMI() {
        Timer = de.pxlab.pxl.TimerBitCodes.STOP_KEY_
TIMER_BIT|de.pxlab.pxl.TimerBitCodes.KEY_TIMER_
BIT|de.pxlab.pxl.TimerBitCodes.END_OF_MEDIA_
TIMER_BIT;
        OnOffTimer = de.pxlab.pxl.TimerCodes.VS_CLOCK_
TIMER;
        OnDuration = 15;
        OffDuration = 0;
        Color = Black;
        eSize = 40;
        iColor = White;
        NCPoints = 5;
        InIP = "127.0.0.1";
        OutIP = "127.0.0.1";
        InPort = 5555;
        OutPort = 4444;
        LTime = 10000;
    }
}
```

Назначение переменных:

`Timer` — устанавливает биты таймера, необходимые для работы процедуры калибровки. Изменять не следует.

`OnOffTimer` — устанавливает тип таймера, применяемый для обновления изображения при калибровке. Изменять не следует.

`OnDuration` — продолжительность основной фазы обновления изображения (в миллисекундах). По порядку величины должна соответствовать времени видеоразвертки одного кадра. Увеличение данной величины может привести к нарушению нормальной работы процедуры калибровки.

`OffDuration` — продолжительность дополнительной фазы обновления изображения. Изменять не следует.

`Color` — цвет калибровочной точки.

`iColor` — цвет центральной части калибровочной точки.

`eSize` — диаметр в пикселях внешнего круга калибровочной точки.

`iSize` — диаметр в пикселях внутренней области калибровочной точки.

`NCpoints` — число калибровочных точек. Для SMI High Speed рекомендуется 5 или 9 точек, для SMI RED-m — 5 точек.

`InIP` — IP-адрес, на который `iViewX` будет отсылать пакеты программе, управляющей экспериментом. Имеющаяся настройка соответствует случаю запуска `iViewX` и PXLab на одной машине. См. SMI `iView X System Manual`.

`OutIP` — IP-адрес, на который PXLab будет отправлять пакеты для `iViewX`. Имеющаяся

настройка соответствует случаю запуска iViewX и PXLab на одной машине.

InPort — порт, на который iViewX будет отсылать пакеты программе, управляющей экспериментом.

OutPort — порт, на который PXLab будет отправлять пакеты для iViewX.

LTime — таймаут ожидания пакета от iViewX.

Изменять не рекомендуется.

Следует отметить, что в нашей программе присутствуют наборы однотипных разделов (основанных на одних и тех же классах PXLab), соответствующие двум операциям: калибровка и предъявление стимульного материала. Предъявление стимульного материала задается в разделах Block(), BlockEnd(), Trial(), а калибровка — в разделах Block:C(), BlockEnd:C(), Trial:C(). Чтобы их различать, в случае процедуры калибровки после имени раздела мы произвольно добавили уникальный постфикс :C. Далее по тексту программы добавление различных постфиксов будет использоваться при необходимости использования объектов, основанных на одном и том же классе. Например, за отправку UDP-пакета отвечает реализованный нами класс SendUDP, при этом подразделы, в которых он используется, получают произвольно заданные нами уникальные постфиксы. Имена соответствующих подразделов получают вид: SendUDP:start_record, SendUDP:stop, SendUDP:save, SendUDP:picture, SendUDP:ask_answer.

Итак, процедура предъявления изображения определяется разделами Block(), BlockEnd(), Trial(). Раздел Block() описывает действия, которые следует выполнить в начале блока экспериментальных ситуаций, и включает подразделы TextParagraph и SendUdp:start_record.

```
Block(TrialFactor, RandomizeTrials){
    TextParagraph(){
        Timer = RESPONSE_TIMER;
        ResponseSet = SPACE_KEY;
        Color = black();
        Alignment = CENTER;
        Text = ["ИНСТРУКЦИЯ",
              "Внимательно рассмотрите предъявляемое
              изображение.",
              " ",
              "Если Вам понятно задание и Вы готовы
              начать работу, нажмите <<пробел>>."];
    }
    SendUDP:start_record() {
        Timer = NO_TIMER;
        UdpHost = iViewHost;
        UdpPort = iViewPort;
        UdpMsg = "ET_REC\n";
    }
}
```

Подраздел TextParagraph описывает сообщение — инструкцию, выводимую по центру экрана.

Подраздел SendUDP:start_record отвечает за посылку UDP-пакета, сообщающего ПО айтрекера о необходимости начала записи.

Раздел BlockEnd() описывает действия, которые следует совершить по окончании блока экспериментальных ситуаций, и включает следующие подразделы. SendUDP:stop — посылает UDP-пакет, сообщающий

о необходимости завершения записи. SendUDP:save посылает UDP-пакет, сообщающий, в какой файл следует сохранить данные, имя файла включает уникальный идентификатор испытуемого. Message описывает сообщение, выводимое на экран по завершении эксперимента.

```
BlockEnd() {
    SendUDP:stop() {
        Timer = NO_TIMER;
        UdpHost = iViewHost;
        UdpPort = iViewPort;
        UdpMsg = "ET_STP\n";
    }
    SendUDP:save() {
        Timer = NO_TIMER;
        UdpHost = iViewHost;
        UdpPort = iViewPort;
        UdpMsg = "ET_SAV C:\\Zhegs\\ET\\
SMI_%SubjectCode%.idf\n";
    }
    Message(){
        Timer = de.pxl.pxl.TimerCodes.
CLOCK_TIMER|de.pxl.pxl.TimerCodes.
RESPONSE_TIMER;
        ResponseSet = SPACE_KEY;
        FontSize = 60;
        Alignment = CENTER;
        Color = black();
        Duration = 5000;
        Text = ["Эксперимент закончен.", "БОЛЬШОЕ
СПАСИБО!"];
    }
}
```

Раздел Trial() описывает последовательность предъявления стимульного материала в ходе экспериментальной ситуации и включает следующие подразделы. SendUdp:fixpoint — отправка UDP-пакета, информирующего о предъявлении фиксационного креста. FixationMark — предъявление фиксационной отметки. Отметка имеет вид креста (FIXATION_CROSS), показывается на заданное время (что определяется типом таймера), время экспозиции составляет 1000 мс (параметр Duration), цвет — черный.

SendUdp:picture — отправка UDP-пакета, информирующего о предъявлении целевого изображения (передается имя изображения Trial.Picture.FileName).

Picture — предъявление целевого изображения. Время экспозиции составляет 3000 мс, изображение находится в папке ImgDir. Название файла с изображением определяется параметром Trial.Picture.FileName. ClearScreen:after показывает пустой экран на 15 мс и включает отображение курсора мыши (MouseOn = 1).

SendUdp:ask_answer — отправка UDP-пакета, информирующего о предъявлении списка вариантов ответа. MultipleChoiceQuestion — предъявление списка вариантов ответа. Задаваемый вопрос определяется параметром Text, варианты ответа — параметром Choices. Параметр Unique = 1 указывает, что допустим только один вариант ответа. Параметр SelectionColor = White определяет цвет подсветки выбранного варианта ответа. Параметр ResponseSet = SPACE_KEY указывает, что завершение выбора выполняется нажатием клавиши пробел.

SendUDP:snd_answer отправляет UDP-пакет, содержащий информацию о выполненном выборе, содержащуюся в полях

Trial.MultipleChoiceQuestion.Selection и Trial.MultipleChoiceQuestion.ResponseTime.

```

Trial(Picture.FileName,
MultipleChoiceQuestion.Selection,
MultipleChoiceQuestion.ResponseTime) {
  SendUDP:fixpoint() {
    Timer = NO_TIMER;
    UdpHost = iViewHost;
    UdpPort = iViewPort;
    UdpMsg = "ET_REM fix.bmp\n";
  }
  FixationMark(){
    MouseOn = 0;
    Timer = CLOCK_TIMER;
    Type = FIXATION_CROSS;
    Color = Black;
    Duration = 1000;
  }
  SendUDP:picture() {
    Timer = NO_TIMER;
    UdpHost = iViewHost;
    UdpPort = iViewPort;
    UdpMsg = "ET_REM %Trial.Picture.FileName%\n";
  }
  Picture() {
    Timer = CLOCK_TIMER;
    Directory = ImgDir;
    Duration = 3000;
  }
  ClearScreen:after(){
    MouseOn = 1;
    JustInTime = 1;
    Timer = CLOCK_TIMER;
    Duration = 15;
  }
  SendUDP:ask_answer() {
    Timer = NO_TIMER;
    UdpHost = iViewHost;
    UdpPort = iViewPort;
    UdpMsg = "ET_REM ASK.bmp\n";
  }
  MultipleChoiceQuestion() {
    LocationX = -200;
    Width = 600;
    Text = ["Кто изображен на картинке?"];
    Choices = ["молодой мужчина", "молодая женщина", "пожилой мужчина", "пожилая женщина"];
    Timer = MOUSE_TRACKING_KEY_TIMER;
    Color = black();
    Unique = 1;
    SelectionColor = White;
    ResponseSet = SPACE_KEY;
  }
  SendUDP:snd_answer() {
    JustInTime = 1;
    Timer = NO_TIMER;
    UdpHost = iViewHost;
    UdpPort = iViewPort;
    UdpMsg = "ET_REM %Trial.
MultipleChoiceQuestion.Selection%,
%Trial.MultipleChoiceQuestion.ResponseTime|
answer.bmp\n";
  }
}

```

Управление последовательностью предъявлений

Раздел Procedure описывает последовательность разветвления сценария эксперимента и имеет следующий вид.

```

Procedure() {
  Session() {
    Block:C() {
      Trial:C(CalibrationFormat, ?, ?, ?, ?);
    }
    Block(1, 1) {
      Trial("adlerbr0.jpg", ?, ?);
      Trial("augusan0.jpg", ?, ?);
      Trial("babelgi0.jpg", ?, ?);
      Trial("braukra0.jpg", ?, ?);
      Trial("dieckro0.jpg", ?, ?);
      Trial("dzembde0.jpg", ?, ?);
      Trial("eich_lu0.jpg", ?, ?);
    }
  }
}

```

Эксперимент содержит единственную сессию, внутри которой находятся два блока: процедуры калибровки (Block:C) и основного эксперимента (Block).

Для экспериментальной ситуации калибровки (Trial:C) в блоке калибровки задается нестандартный формат выходных данных и четыре выходных параметра, соответствующих X и Y координатам использовавшихся точек калибровки (и одновременно валидации), а также X и Y координатам точек фактических позиций взора наблюдателя, полученным в результате процедуры валидации.

Основной блок предъясняется однократно (первый параметр вызова), причем экспериментальные ситуации (Trial) следует показывать в случайном порядке (второй параметр вызова). Всего эксперимент включает семь экспериментальных ситуаций (соответствующие строки Trial). Первый аргумент, соответствующий параметру Picture.FileName, задает имя файла с экспонируемым изображением. При этом предполагается, что ПО PXLab установлено в полном объеме в папку C:\pxlab. Вторым и третьим аргументами обозначены знаками вопроса, указывающими, что соответствующие параметры MultipleChoiceQuestion.Selection и MultipleChoiceQuestion.ResponseTime являются выходными и их значения формируются по завершении экспериментальной ситуации.

Установка PXLab и запуск простейшего эксперимента

Использование PXLab требует предварительной установки Java Runtime Environment (тестировалось с версиями 6, 7, 8 JRE). ПО PXLab скачивается с сайта <http://irtel.uni-mannheim.de/pxlab/> (архив PXLabRT.zip) и разархивируется в папку c:\pxlab. Файл pxlab.jar, содержащийся в архиве, заменяется на модифицированный вариант. Инструкция по установке и использованию PXLab также доступна на сайте www.pxlab.de. Для оптимальной организации процедуры эксперимента рекомендуется следующая последовательность действий.

1. Разархивировать архив PXLabRT.zip в папку C:\pxlab.
2. Модифицированный файл pxlab.jar помещается в папку C:\ProgramFiles\Java\%java_version%\lib\ext, где %java_version% соответствует установленной версии Java.
3. На компьютер устанавливается текстовый редактор Notepad++.

4. В произвольную папку помещаются: прилагаемый файл сценария эксперимента ET_minimal.txt. В нее же копируются файлы jxinput.dll, pxlab.dll, JNIsiapp.dll из C:\pxlab\bin.

5. В той же папке создается стартовый файл runexp.bat, содержащий команду java de.pxlab.px1.run.ExRun -f ET_minimal.txt -S5. Параметр -f указывает на выполняемый файл сценария, -S5 — задает предъявление в монопольном полноэкранном режиме на втором экране компьютера. Параметры видеорежима при запуске не изменяются.

6. Запуск эксперимента выполняется файлом runexp.bat, после чего в открывшемся окне следует ввести произвольный Subject Identification Code, уникальным образом идентифицирующий испытуемого. По завершении эксперимента файл с протоколом эксперимента будет находиться в поддиректории dat, а файл с дополнительной отладочной информацией — в поддиректории dtr.

7. Для обеспечения взаимодействия с айтрекером следует задать значения iViewHost и iViewPort в блоке AssignmentGroup эксперимента в соответствии с настройками айтрекера. В подразделе SendUDP:save следует задать желаемый путь, по которому будет выполняться сохранение данных айтрекера. В подразделе CalibrationSMI следует задать значения InIP, OutIP, InPort, OutPort, NCPoints.

Ограничения PXLab

Существенной проблемой ПО PXLab являются ограничения на форматы воспроизводимых видеофайлов и звуковых файлов, характерные для любого ПО, реализуемого на языке Java. Воспроизведение звукового файла реализовано с помощью класса **SoundControl**, запись с микрофона — классами **SoundRecorder** и **SoundRecorderControl**, фиксация момента устного ответа — классом **VoiceKey**. Практические возможности данной функциональности нами не проверялись. Воспроизведение видеофайла реализовано с помощью класса **Movie**. На практике необходим подбор кодека, обеспечивающего стабильность воспроизведения, что может привести к неоправданному увеличению объема воспроизводимого файла. Отсутствие собственного практического опыта не позволяет нам рекомендовать PXLab для проведения исследований, включающих запись и воспроизведение аудиофайлов, воспроизведение видеофайлов.

PXLab может быть запущена под Linux, по крайней мере на Debian-подобных дистрибутивах (тестировалась на Ubuntu 15.10), при условии установки проприетарной версии Java от Oracle. Запуск возможен только в полноэкранном режиме, смена разрешения экрана при запуске возможна, смена частоты кадровой развертки — нет. Следует иметь в виду, что под Linux не реализована возможность синхронизации с видеоразверткой экрана и, соответственно, время экспозиции стимульного материала может быть задано с точностью плюс/минус кадр видеоразвертки.

В штатную поставку выполняемого модуля PXLab входит реализация интерактивного конструктора экспериментов Design Editor. Однако исходный код, предоставленный разработчиками PXLab, не включает реализацию данного компонента. Таким образом, при

самостоятельной сборке PXLab получаемый выполняемый модуль не будет включать Design Editor. Из-за этого конструктор Design Editor отсутствует в доработанной нами версии выполняемого модуля PXLab.

Обработка результатов айтрекингового эксперимента, реализованного в PXLab

Результатом айтрекингового эксперимента, выполняемого под управлением ПО PXLab, являются два файла. Текстовый файл с расширением .dat содержит протокол эксперимента в формате, указанном при задании сценария. Его обработка выполняется стандартными средствами, например Excel. В случае рассмотренного эксперимента содержание .dat файла (определяемое переменной DataFileTrialFormat в разделе AssignmentGroup) будет иметь следующий вид:

```
Z,[640, 320, 1267, 960, 13],[512, 10, 256, 1014, 768],[643, 309, 1257, 949, 57],[505, 35, 250, 1016, 765]
Z,babelgi0.jpg,4,3005.1037,4230.0406
Z,dzembde0.jpg,3,3004.918,3141.6332
Z,adlerbr0.jpg,1,3005.0568,2206.2532
Z,braukra0.jpg,1,3004.6906,1110.462
Z,dieckro0.jpg,1,3004.7201,817.7815
Z,augusan0.jpg,4,3004.6691,2637.1474
Z,eich_lu0.jpg,3,3004.7284,1967.5415
```

Первая строка содержит результат выполнения процедуры калибровки и позволяет оценить точность локализации взора в разных зонах экрана. Дальнейшие строки соответствуют данным по отдельным экспериментальным ситуациям. Поля в строке разделяются запятыми. Первое поле %SubjectCode% — уникальный идентификатор участника, задаваемый экспериментатором. Второе поле %Trial.Picture.FileName% — имя экспонируемого файла с изображением. Третье поле — %Trial.MultipleChoiceQuestion.Selection% — код ответа испытуемого. Четвертое поле %Trial.Picture.ResponseTime% — фактическое время предъявления изображения. Пятое поле %Trial.MultipleChoiceQuestion.ResponseTime% — время, затраченное на ответ (с момента показа возможных вариантов ответа).

Файл с расширением .idf содержит результаты регистрации движений глаз во внутреннем формате SMI IView Data File (iView X System Manual, 2011). В нем содержится информация о положении взора наблюдателя в координатах экрана компьютера, на котором демонстрируется рассматриваемое изображение. Величина временного интервала между последовательными записями определяется частотой регистрации айтрекера. Также в файле содержатся отметки о предъявлении очередного изображения. В случае рассмотренного эксперимента файл данных будет содержать отметки о показе фиксационной точки, целевого изображения, списка вариантов ответа и данном пользователем ответе. Для проведения дальнейшего анализа движений глаз может быть использовано штатное ПО BeGaze, поставляемое с айтрекерами Sensomotoric Instruments. В этом случае исследователь должен самостоятельно обеспечить подготовку изображений — подложек, размер которых соответствует разрешению экрана при проведении эксперимента, а содержание — фактически экспонировавшимся на экране изображениям. В рассмотренном

примере для анализа в BeGaze нужно будет подготовить полноэкранный образ с фиксационной точкой (имя файла — fix.bmp); полноэкранные изображения с рассматриваемыми фотографиями (имена файлов совпадают с именами файлов, использованных в качестве стимульного материала); полноэкранный образ со списком вариантов ответов (имя файла ASK.bmp). При этом анализ ответов, даваемых участниками исследования, средствами BeGaze осуществить не удастся.

Выбор альтернативных вариантов свободно расширяемого программного обеспечения для обработки результатов эксперимента определяется потребностями исследователя. Экспорт данных из idf-файла в текстовый формат выполняется с помощью штатной программы IDF-Converter. Полученный файл с «сырыми» данными может быть, например, загружен в программу OGAMA (Open Gaze and Mouse Analyzer) (Vofskühler et al, 2008). Еще один возможный вариант — использование среды статистической обработки R с дополнительно разработанным нами пакетом ETRAN (Мармалюк и др., 2015; Zhegallo, Marmalyuk, 2015).

Заключительные комментарии

Доработанная версия ПО PXLab рекомендуется для проведения айтрекинговых исследований с использованием регистрирующего оборудования Sensomotoric Instruments, в первую очередь при проведении экспериментов, включающих рассмотрение статических изображений и текстов и последующую автоматическую фиксацию ответов. В качестве примера такого исследования можно привести решение задачи опознания эмоционального состояния по фотоизображению (Жегалло, 2015). Участники исследования рассматривали статические фотоизображения лиц натурщиков, выражающих различные эмоциональные экспрессии; ответ давался в форме выбора одного из возможных названий эмоции из закрытого списка³. В качестве дополнительного примера возможностей PXLab на сайте разработчика помещен пример tst_AVX, содержащий реализацию дискриминационной AVX-задачи с регистрацией движений глаз. Данная методика используется при изучении категориальности восприятия эмоциональных экспрессий (Барабанщиков, Жегалло, Королькова, 2016). Принципы реализации в целом аналогичны рассмотренному примеру. Одновременная экспозиция двух изображений достигается за счет использования режима Overlay = Join. Комбинированная фиксация ответа как во время экспозиции последнего изображения (X), так и после него, обеспечивается предварительным формированием значений переменных ADuration и AText в подразделах SetParameter:IsAnswer() и SetParameter:IsAnswer2(). Контроль момента выполнения дополнительно обеспечивается использованием настроечного параметра JustInTime.

Мы позиционируем PXLab как один из элементов экосистемы программного обеспечения с открытым

исходным кодом, используемого для проведения экспериментальных исследований и последующей обработки результатов.

Для облегчения освоения ПО в Приложении 1 приводится справочная информация, описывающая классы PXLab, использованные при реализации рассмотренного в статье эксперимента, а в Приложении 2 — пример простейшего эксперимента, предназначенного для тестирования наличия связи между PXLab и айтрекером.

Литература

- Ананьева К.И., Басюл И.А., Харитонов А.Н. Координация взоров участников парного эксперимента и успешность решения перцептивно-коммуникативной задачи // Айтрекинг в психологической науке и практике / Под ред. В.А. Барабанщикова. М.: Когито-Центр, 2015. С. 340–344.
- Барабанщиков В.А., Жегалло А.В., Королькова О.А. Перцептивная категоризация выражений лица. М.: Когито-Центр, 2016.
- Бессонова Ю.В., Обознов А.А., Лобанова Л.А. Использование айтрекинга для диагностики мотивации личности // Айтрекинг в психологической науке и практике. М.: Когито-Центр, 2015. С. 147–157.
- Жегалло А.В. Специфика содержания зрительных фиксаций при опознании эмоциональных экспрессий по выражению лица // Айтрекинг в психологической науке и практике / Под ред. В.А. Барабанщикова. М.: Когито-Центр, 2015. С. 240–255.
- Жегалло А.В., Хозе Е.Г. Движение глаз при оценке лица, передающего достоверную и недостоверную информацию // Айтрекинг в психологической науке и практике. М.: Когито-Центр, 2015. С. 256–259.
- Жердев И.Ю., Барабанщиков В.А. Аппаратно-программный комплекс для исследований зрительного восприятия сложных изображений во время саккадических движений глаз человека // Экспериментальная психология. 2014. Т. 7. №1. С. 123–137.
- Мармалюк П.А., Жегалло А.В., Юрвев Г.А., Панфилова А.С. Принципы построения программного обеспечения с открытым исходным кодом для анализа результатов окулографических исследований // Экспериментальная психология. 2015. Т. 8. №1. С. 127–144.
- Ярбус А.Л. Роль движений глаз в процессе зрения. М.: Наука, 1965.
- Irtel H. PXLab: The Psychological Experiments Laboratory [online]. Version 2.1.11. Mannheim, Germany: University of Mannheim, 2007.
- iView X System Manual. Version 2.8. SensoMotoric Instruments GmbH, 2011.
- Koltunova T.I., Podladchikova L.N. Distractor effect at initial stages of recognition depends on visual image properties // Journal of Integrative Neuroscience. 2013. Vol. 12. No.1. P. 91–101. doi:10.1142/S021963521350009X
- Langner O., Dotsch R., Bijlstra G., Wigboldus D.H., Hawk S.T., van Knippenberg A. Presentation and validation of the Radboud Faces Database // Cognition and Emotion. 2010. Vol. 24. No.8. P. 1377–1388. URL: http://www.tandfonline.com/doi/abs/10.1080/02699930903485076. doi:10.1080/02699930903485076
- Peirce J.W. PsychoPy—psychophysics software in Python // Journal of Neuroscience Methods. 2007. Vol. 162. No.1. P. 8–13. doi:10.1016/j.jneumeth.2006.11.017
- Vofskühler A., Nordmeier V., Kuchinke L., Jacobs A.M. OGAMA — OpenGazeAndMouseAnalyzer: Open source software designed to analyze eye and mouse movements in slideshow study designs // Behavior Research Methods. 2008. Vol. 40. No.4. P. 1150–1162. doi:10.3758/BRM.40.4.1150
- Zhegallo A.V., Marmalyuk P.A. ETRAN — R extension package for Eye Tracking Results Analysis // Perception. 2015. Vol. 44. No.8-9. P. 1129–1135. doi:10.1177/0301006615594944

³ Реализация данного эксперимента принципиально не отличается от рассмотренного в статье примера, но в качестве стимульного материала в нем используются фотоизображения лиц натурщиков из базы RaFD (Langer et al, 2010). Лицензионные ограничения на данные изображения не допускают их свободного распространения, что затрудняет воспроизведение данного исследования.

Приложение 1. Справочная информация по ПО PXLab, описывающая классы, использованные при реализации демонстрационного эксперимента

Настоящее приложение основывается на фрагментарном переводе материалов, содержащихся в PXLab manual (<http://irtel.uni-mannheim.de/pxlab/doc/manual/>) и PXLab API reference (<http://irtel.uni-mannheim.de/pxlab/doc/api/>).

Класс **Display**. Используемые в эксперименте стимулы представляют собой последовательность объектов, класс которых является производным от класса Display. С классом Display связан список параметров, определяющих специфику отображения объектов. Данные параметры доступны для всех классов, производных от класса Display.

Параметры класса Display

Duration — задает продолжительность отображения в миллисекундах.

Execute — по умолчанию устанавливается значение Execute=1, что разрешает выполнение кода соответствующего объекта и его отображение.

JustInTime — управляет моментом времени, когда вычисляются свойства отображаемого объекта.

Если значение параметра true (не равно 0), свойства вычисляются непосредственно перед отображением объекта. Если значение параметра false (равно 0), то свойства объекта вычисляются заранее.

Overlay — каждый отображаемый объект имеет фоновый элемент, перекрывающий весь экран. В результате одновременное отображение нескольких объектов оказывается невозможным. Параметр Overlay = de.pxlab.pxl.OverlayCodes.JOIN указывает, что фон отображать не следует. При этом ранее отображавшийся объект остается видимым.

ResponseChar — код клавиши ответа, остановившей отображение объекта.

ResponseCode — код, формируемый по окончании отображения объекта.

ResponsePosition — позиция указателя «мыши» в момент, когда зафиксирован ответ, данный с помощью «мыши».

ResponseSet — список допустимых кодов ответа, завершающих предъявление объекта.

ResponseTime — фактическая продолжительность предъявления (в миллисекундах).

Screen — код выбора экрана. Используется в случае, когда объект должен показываться на многоэкранной системе.

TimeControl — фактический момент времени с начала проведения эксперимента, когда начато предъявление (в миллисекундах).

TimeError — возвращает ошибку тайминга для объектов, которые должны отображаться заданное время (для таймера установлен флаг CLOCK_TIMER).

Timer — определяет тип таймера, управляющего продолжительностью отображения объекта. Итоговое значение таймера определяется комбинацией отдельных битов.

NO_TIMER_BIT — не использовать таймер.

CLOCK_TIMER_BIT — контролировать время отображения.

MOUSE_BUTTON_RESPONSE_BIT — использовать нажатие клавиши «мыши» как таймер.

KEY_TIMER_BIT — использовать нажатие клавиатуры как таймер.

STOP_KEY_TIMER_BIT — используется для регистрации повторных нажатий клавиатуры, клавиша, определяемая глобальным параметром StopKey используется для завершения экспозиции.

XBUTTON_TIMER_BIT — в качестве таймера используется нажатие клавиши на внешнем регистрирующем устройстве, подключенном к управляющей линии последовательного порта.

DOWN_TIMER_BIT — указывает, что в качестве таймера используется *нажатие* кнопки или клавиши.

UP_TIMER_BIT — указывает, что в качестве таймера используется *отпускание* кнопки или клавиши.

MOUSE_TRACKING_TIMER_BIT — указывает, что в качестве таймера используется перемещение «мыши».

STORE_TIMER_BIT — указывает, что результаты выполнения таймера должны сохраняться во внешних параметрах.

START_RESPONSE_TIMER_BIT — используется для задания интервалов ответа, превышающих время отображения одного объекта. Указывает, что начало измерения интервала запускается со стартом таймера.

STOP_RESPONSE_TIMER_BIT — используется для задания интервалов ответа, превышающих время отображения одного объекта. Указывает, что измерение интервала завершается по окончании работы таймера.

VIDEO_SYNCHRONIZATION_BIT — запуск таймера должен синхронизироваться с началом очередного интервала вертикальной видеоразвертки.

WATCH_SPURIOUS_RESPONSES_BIT — ложные ответы, зафиксированные за время работы таймера, должны фиксироваться.

SERIAL_LINE_TIMER_BIT — ожидается последовательность символов с активированного последовательного порта. Последовательность должна завершаться нажатием клавиши ВВОД.

VOICE_KEY_TIMER_BIT — используется для задания таймера, работающего по «голосовой клавише», активируемой микрофоном.

SYNC_TO_MEDIA_TIMER_BIT — используется для задания времени ожидания при проигрывании медиафайла.

END_OF_MEDIA_TIMER_BIT — таймер завершается по окончании проигрывания медиафайла.

Для упрощения работы с таймером наиболее употребительные комбинации битов таймера получили собственные обозначения.

NO_TIMER — код, указывающий не использовать таймер. Используется на практике для одно-временного отображения нескольких объектов. В таком случае для всех объектов кроме последнего значение таймера должно быть **NO_TIMER**; для всех объектов кроме первого должен быть установлен параметр `Overlay = JOIN`.

CLOCK_TIMER — таймер с фиксированным временным интервалом. Результаты выполнения сохраняются.

RAW_CLOCK_TIMER — таймер с фиксированным временным интервалом. Результаты выполнения не сохраняются.

VS_CLOCK_TIMER — таймер с фиксированным временным интервалом, запускается в момент начала очередного цикла вертикальной видеоразвертки.

WATCHING_CLOCK_TIMER — таймер с фиксированным временным интервалом. Во время работы таймера запоминаются коды и моменты срабатывания кнопок.

KEY_RESPONSE_TIMER — таймер, использующий нажатие клавиатуры для завершения. Сохраняются код и время нажатия.

RELEASE_KEY_RESPONSE_TIMER — таймер, использующий отпускание кнопки клавиатуры для завершения. Сохраняются код кнопки и время отпускания.

RESPONSE_TIMER — таймер, использующий для завершения нажатие «мыши», клавиатуры или кнопки внешнего устройства. Сохраняются код кнопки и время нажатия.

VS_RESPONSE_TIMER — таймер, использующий для завершения нажатие «мыши», клавиатуры или кнопки внешнего устройства. Сохраняются код кнопки и время нажатия. Запуск таймера синхронизируется с началом очередного цикла вертикальной видеоразвертки.

RELEASE_RESPONSE_TIMER — таймер, использующий для завершения отпускание «мыши», клавиатуры или кнопки внешнего устройства. Сохраняются код кнопки и время отпускания.

LIMITED_RESPONSE_TIMER — продолжительность определяется либо действиями испытуемого, также как для **RESPONSE_TIMER**, либо заданным временным интервалом, в зависимости от того, какое событие наступит раньше. Сохраняются код и время работы.

GO_TIMER — продолжительность определяется действиями испытуемого. Результаты не сохраняются.

LIMITED_GO_TIMER — продолжительность определяется действиями испытуемого либо заданным временным интервалом. Результаты не сохраняются.

MOUSE_TRACKING_TIMER — таймер непрерывно контролирует положение «мыши». Завершение по отпусканию кнопки «мыши».

STOP_KEY_TIMER — завершается по нажатии заранее определенной кнопки `StopKey`. Сохраняются код и время работы.

MOUSE_TRACKING_STOP_KEY_TIMER — комбинация **MOUSE_TRACKING_TIMER** и **STOP_KEY_TIMER**.

MOUSE_TRACKING_KEY_TIMER — комбинация **MOUSE_TRACKING_TIMER** и **KEY_RESPONSE_TIMER**.

MOUSE_TRACKING_RELEASE_KEY_TIMER — комбинация **MOUSE_TRACKING_TIMER** и **RELEASE_KEY_RESPONSE_TIMER**.

SERIAL_LINE_INPUT_TIMER — таймер ожидает ввода из последовательного порта. Ввод должен завершиться кодом `<CR>`. Полученная строка сохраняется в поле `ResponseCode`.

VOICE_KEY_TIMER — таймер срабатывает по нажатие «голосовой клавиши».

START_RESPONSE_TIMER — используется для задания интервалов ответа, включающих более одного отображаемого объекта. Указывает, что при запуске таймера начинается измерение времени ответа.

STOP_RESPONSE_TIMER — используется для задания интервалов ответа, включающих более одного отображаемого объекта. Указывает, что при запуске таймера завершается измерение времени ответа.

END_OF_MEDIA_TIMER — используется для ожидания завершения проигрывания или записи медиафайла.

WATCHING_END_OF_MEDIA_TIMER — используется для ожидания завершения проигрывания или записи медиафайла с фиксацией действий испытуемого в процессе действия.

SYNC_TO_MEDIA_TIMER — используется для синхронизации с медиаобъектом. В ходе проигрывания или записи должно быть достигнуто заданное время.

Класс **FontDisplay** — производный от класса **Display**. Определяет свойства объектов, связанные с отображением шрифтов. Непосредственно при описании экспериментов не используется.

Параметры класса **FontDisplay**

Color — цвет отображаемого текста.

FontFamily — используемое семейство шрифтов.

FontSize — размер шрифта.

FontStyle — стиль шрифта.

Класс **TextDisplay** — производный от класса **FontDisplay**. Определяет свойства объектов, связанные с отображением текста. Непосредственно при описании экспериментов не используется.

Параметры класса **TextDisplay**

LocationX — горизонтальная позиция текста.

LocationY — вертикальная позиция текста.

ReferencePoint — Референсная точка, для которой задаются X и Y координаты. Допустимые значения: **BASE_CENTER**, **BASE_LEFT**, **BASE_RIGHT**, **MIDDLE_CENTER**, **MIDDLE_LEFT**, **MIDDLE_RIGHT**, **TOP_CENTER**,

TOP_LEFT, TOP_RIGHT. По умолчанию: BASE_LEFT.

Text — отображаемый текст.

Класс **TextParagraph** — производный от класса **TextDisplay**. Определяет отображаемый на экране параграф текста. Контролируются выравнивание, ширина текста, расстояние между строками. Текст может быть принудительно разбит на отдельные строки, длина которых не превышает заданной.

Параметры класса TextParagraph

Alignment — выравнивание текста внутри параграфа.

EmphasizeFirstLine — выделение первой строки текста.

LineSkipFactor — расстояние между строками.

Width — ширина параграфа.

Wrapping — автоматическое разбиение текста на отдельные строки.

Класс **Message** — производный от класса **TextParagraph**. Отображает на экране текст по центру экрана. В отличие от **TextParagraph** ориентирован на отображение одной строки текста. Несколько строк или строка, разделенная символом перевода ('\n'), отображаются с выравниванием по центру экрана. Вертикальная позиция при этом соответствует нижней строке текста.

Класс **FixationMark** — производный от класса **Display**. Отображает на экране фиксационную метку.

Параметры класса FixationMark

Color — цвет метки.

EnclosedHeight — высота прямоугольника, обрамляемого метками, в случае метки типа CORNER_MARKS.

EnclosedWidth — ширина прямоугольника, обрамляемого метками, в случае метки типа CORNER_MARKS.

LineWidth — толщина линий.

LocationX — горизонтальная позиция центра метки.

LocationY — вертикальная позиция центра метки.

Size — горизонтальный и вертикальный размеры метки.

Type — тип фиксационной метки.

CORNER_MARKS — метки по углам заданного прямоугольника.

FIXATION_CROSS — фиксационный крест.

FIXATION_DOT — фиксационная точка.

NO_FIXATION — не показывать фиксационную метку.

Класс **Picture** — производный от класса **Display**. Отображает на экране изображение, содержащееся в файле.

Параметры класса Picture

ClearFileBaseOnRecompute — в документации описание отсутствует. Очистка содержимого перед перезагрузкой изображения.

Color — в документации описание отсутствует. Цвет, используемый для первоначальной инициализации изображения.

Directory — папка, в которой содержится изображение.

FileName — название файла с изображением.

Height — высота изображения в формате scalable vector graphics (SVG).

Histogram — в документации описание отсутствует. При установке параметра в true в консоль во время инициализации эксперимента выдается отладочная информация об изображении.

LocationX — горизонтальная позиция изображения.

LocationY — вертикальная позиция изображения.

ReferencePoint — Референсная точка, для которой задаются X и Y координаты. Допустимые значения: BASE_CENTER, BASE_LEFT, BASE_RIGHT, MIDDLE_CENTER, MIDDLE_LEFT, MIDDLE_RIGHT, TOP_CENTER, TOP_LEFT, TOP_RIGHT. По умолчанию MIDDLE_CENTER.

Width — ширина изображения в формате scalable vector graphics (SVG).

Класс **ClearScreen** — производный от класса **Display**. Отображает пустой экран.

Класс **MultipleChoiceQuestion** — производный от класса **TextParagraph**. Отображает список выбираемых вариантов.

BulletSize — размер маркера, отмечающего выбираемые варианты.

Choices — массив строк, задающих альтернативные варианты выбора.

DisableNonSelection — если параметр установлен, то в момент выбора должен быть отмечен по крайней мере один из вариантов.

Indent — в документации указано «choice alternatives indentation». Установка различных значений параметра не приводит к изменениям отображаемого объекта.

ParSkip — расстояние между отображаемыми вариантами по вертикали.

Selection — список выбранных вариантов.

SelectionColor — цвет, используемый для выделения выбранного варианта.

Unique — флаг, указывающий, что только один из альтернативных вариантов может быть выбран.

Приложение 2. Минимальный айтрекинг-эксперимент, тестирующий наличие связи с оборудованием

Недостатком реализованной схемы взаимодействия с айтрекером является однонаправленный характер взаимодействия. Управляющие пакеты посылаются ПО PXLab, контролирующим ход эксперимента, и принимаются управляющим ПО айтрекера. Обратная связь, обеспечивающая контроль состояния айтрекера, не реализована. В результате эксперимент может выполняться в отсутствие фактической связи с айтрекером. От оператора требуется вручную контролировать подключение оборудования и соответствие IP-адреса и порта, через которые происходит коммуникация.

Для проверки нормального функционирования связи можно использовать приведенный далее простейший тестовый эксперимент, содержащий единственную тестовую экспериментальную ситуацию. В ходе выполнения сценария иницируется запись движений глаз, отсылается единственная отметка о выполнении экспериментальной ситуации, останавливается запись и выполняется сохранение файла с данными. При наличии связи с айтрекером в результате выполнения эксперимента в целевой папке, задаваемой параметром `UdpMsg` в подразделе `SendUDP:save`, должен появиться файл с расширением `.idf`, содержащий результаты регистрации движений глаз.

```
Experiment() {
  Context() {
    AssignmentGroup() {
      ExperimentName = "Eyetracking
connection test";
      SkipBouncingBlockDisplays = 0;
      DataFileTrialFormat =
"%SubjectCode%";
      new iViewHost = "127.0.0.1";
      new iViewPort = 4444;
      ScreenBackgroundColor = lightGray();
    }

    Block() {
      TextParagraph() {
        Timer = RESPONSE_TIMER;
        ResponseSet = SPACE_KEY;
        Color=black();
        Alignment = de.pxlab.pxl.
AlignmentCodes.CENTER;
        Text = ["Нажмите <<ПРОБЕЛ>>
для запуска теста"];
      }
      SendUDP:start_record() {
        Timer = NO_TIMER;
        UdpHost = iViewHost;
        UdpPort = iViewPort;
        UdpMsg = "ET_REC\n";
      }
    }
  }
}
```

```
BlockEnd() {
  SendUDP:stop() {
    Timer = NO_TIMER;
    UdpHost = iViewHost;
    UdpPort = iViewPort;
    UdpMsg = "ET_STP\n";
  }

  SendUDP:save() {
    Timer = NO_TIMER;
    UdpHost = iViewHost;
    UdpPort = iViewPort;
    UdpMsg = "ET_SAV C:\\Zhegs\\ET\\
SMI_%SubjectCode%.idf\n";
  }
  Message() {
    Timer = CLOCK_TIMER|RESPONSE_
TIMER;

    ResponseSet = SPACE_KEY;
    FontSize = 60;
    Alignment = CENTER;
    Color=black();
    Duration = 5000;
    Text = ["Тест закончен"];
  }
}

Trial() {
  SendUDP:fixpoint() {
    Timer = NO_TIMER;
    UdpHost = iViewHost;
    UdpPort = iViewPort;
    UdpMsg = "ET_REM fix.bmp\n";
  }
  AssignmentGroup();
}

Procedure() {
  Session() {
    Block() {
      Trial();
    }
  }
}
```

software and equipment

PXLab Software as an Instrument for Eye-Tracking Research Using SMI Eye Trackers

Alexander Zhegallo

Institute of Psychology RAS, Moscow, Russia

Abstract. PXLab software is an open source tool for the automated presentation of stimuli images and text as well as the recording of responses in psychological research. We present a modified version of the PXLab software that can be used as a tool for eye-tracking experiments with SensoMotoric Instruments (SMI) systems. This paper describes the features of the modified software and implementation of the automated recording of eye-tracking responses. A detailed, example-based tutorial is provided. The modified version of PXLab and examples are available on <https://sites.google.com/site/azhegallo/main/pxlab>.

Correspondence: Alexander Zhegallo, zhegs@mail.ru; Yaroslavskaya str., 13, Institute of Psychology RAS, 129366 Moscow, Russia

Keywords: eye-tracking, eye movements, experimental psychology

Copyright © 2016. Alexander Zhegallo. This is an open-access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/) (CC BY), which permits unrestricted use, distribution, and reproduction in any medium, provided that the original author is credited and that the original publication in this journal is cited, in accordance with accepted academic practice.

Acknowledgements. This study is supported by FASO Russia, № 0159-2016-0004.

Received 16 March 2016, accepted 28 September 2016.

References

- Anan'eva, K.I., Basyul, I.A., & Kharitonov, A.N. (2015). Koordinatsiya vzorov uchastnikov parnogo eksperimenta i uspesnost' resheniya pertseptivno-kommunikativnoy zadachi [Gaze coordination and performance in cognitive and communicative task in participants of dyadic experiment]. In V.A. Barabanshchikov (Ed.), *Aitrekning v psikhologicheskoi nauke i praktike [Eyetracking in psychological science and practice]* (pp. 340–344). Moscow: Kogito-Centr Publ. (In Russian).
- Barabanshchikov, V.A., Zhegallo, A.V., & Korolkova, O.A. (2016). *Pertseptivnaya kategorizatsia vyrazhenij litsa [Perceptual categorization of facial expressions]*. Moscow: Kogito-Centr Publ. (In Russian).
- Bessonova, Y.V., Oboznov, A.A., & Lobanova, L.A. (2015). Ispol'zovanie aytrekinga dlya diagnostiki motivatsii lichnosti [Using eye-tracking for assessment of motivation]. In V.A. Barabanshchikov (Ed.), *Aitrekning v psikhologicheskoi nauke i praktike [Eyetracking in psychological science and practice]* (pp. 147–157). M.: Kogito-Centr Publ. (In Russian).
- Irtel, H. (2007). *PXLab: The Psychological Experiments Laboratory [online]*. Version 2.1.11. Mannheim, Germany: University of Mannheim.
- iView X System Manual. Version 2.8.*. SensoMotoric Instruments GmbH.
- Koltunova, T.I., & Podladchikova, L.N. (2013). Distractor effect at initial stages of recognition depends on visual image properties. *Journal of Integrative Neuroscience*, 12(1), 91–101. [doi:10.1142/S021963521350009X](https://doi.org/10.1142/S021963521350009X)
- Langner, O., Dotsch, R., Bijlstra, G., Wigboldus, D.H., Hawk, S.T., & van Knippenberg, A. (2010). Presentation and validation of the Radboud Faces Database. *Cognition and Emotion*, 24(8), 1377–1388. Retrieved from <http://www.tandfonline.com/doi/abs/10.1080/02699930903485076>. [doi:10.1080/02699930903485076](https://doi.org/10.1080/02699930903485076)
- Marmalyuk, P.A., Zhegallo, A.V., Yuryev, G.A., & Panfilova, A.S. (2015). Principles of construction of open-source software for oculography data analysis. *Experimental Psychology (Russia)*, 8(1), 127–144. (In Russian).
- Peirce, J.W. (2007). PsychoPy—psychophysics software in Python. *Journal of Neuroscience Methods*, 162(1), 8–13. [doi:10.1016/j.jneumeth.2006.11.017](https://doi.org/10.1016/j.jneumeth.2006.11.017)
- Vofßkühler, A., Nordmeier, V., Kuchinke, L., & Jacobs, A.M. (2008). OGAMA — OpenGazeAndMouseAnalyzer: Open source software designed to analyze eye and mouse movements in slideshow study designs. *Behavior Research Methods*, 40(4), 1150–1162. [doi:10.3758/BRM.40.4.1150](https://doi.org/10.3758/BRM.40.4.1150)
- Yarbus, A.L. (1967). *Eye movements and vision*. New York: Plenum Press. [doi:10.1007/978-1-4899-5379-7](https://doi.org/10.1007/978-1-4899-5379-7)

- Zhegallo, A.V. (2015). Spetsifika sodержaniya zritel'nykh fiksatsiy pri opoznanii emotsional'nykh ekspressiy po vyrazheniyu litsa [Specifics of fixation content in identification of emotional facial expressions]. In V.A. Barabanshchikov (Ed.), *Aitreking v psikhologicheskoi nauke i praktike [Eyetracking in psychological science and practice]* (pp. 240–255). M.: Kogito-Centr Publ. (In Russian).
- Zhegallo, A.V., & Khoze, E.G. (2015). Dvizhenie glaz pri otsenke litsa, peredayushchego dostovernuyu i nedostovernuyu informatsiyu [Eye movements in perception of faces communicating true or false statements]. In V.A. Barabanshchikov (Ed.), *Aitreking v psikhologicheskoi nauke i praktike [Eyetracking in psychological science and practice]* (pp. 256–259). M.: Kogito-Centr Publ. (In Russian).
- Zhegallo, A.V., & Marmalyuk, P.A. (2015). ETRAN — R extension package for Eye Tracking Results Analysis. *Perception*, 44(8-9), 1129–1135. doi:10.1177/0301006615594944
- Zherdev, I.Y., & Barabanshchikov, V.A. (2014). Hardware-software system for study of complex images' visual perception during saccadic eye movements in man. *Experimental Psychology (Russia)*, 7(1), 123–137. (In Russian).