# Общее описание.

Данный проект позволяет стримерам организовывать на своих трансляциях «интерактивный режим», при котором зрители вовлекаются в игровой процесс и позволяют генерировать внутриигровые события в прямом эфире.

Основные отличия от других подобных систем интерактива заключаются в следующем:

1. Данная реализация полностью самостоятельна и независима, не требуется никакого другого софта.
2. Команды зрителей берутся из «живого чата» в активной трансляции пользователя, а не из суммы донатов и сообщений донатеров. Благодаря этому любой зритель может принять участие в игре, что является преимуществом для не очень популярных стримеров с небольшим числом подписчиков.
3. Гибкость и расширяемость. Правкой конфига можно настраивать параметры и доступность команд, а лежащий в основе проекта парсер полностью автономен и при желании может быть использован для организации интерактивного режима в любой другой игре.

# Поддерживаемые игры

На данный момент поддерживаются игры серии S.T.A.L.K.E.R. и сделанные на базе их моды. Для адаптации скрипта интерактива на конкретный мод необходимо cкопировать содержимое папки gamedata проекта в папку gamedata, расположенную в корне игры (если такой папки там нет – следует создать её). Далее в папке gamedata\scripts игры (или мода) нужно открыть файл bind\_stalker.script (если его там нет – взять из распакованной игры), найти функцию actor\_binder:update и в её конце добавить строку interactive\_commands.update(). После этого для проверки правильности адаптации можно запустить игру и скопировать в $app\_data\_root$ файл из директории tests (сохраняя неизменным его имя). Если все сделано правильно – скрипт интерактива сгенерирует соответствующе команде в имени файла событие, а сам файл будет удалён.

# Общая архитектура и устройство системы



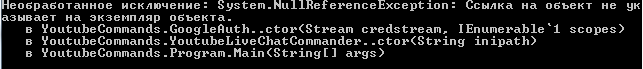
Как видим, система состоит из трёх основных блоков. Рассмотрим более подробно каждый из них вместе с процедурами настройки.

# Парсер команд

Представлен исполняемым файлом YoutubeCommands.exe. Данная программа мониторит все сообщения зрителей в чате трансляции (это возможно благодаря предоставляемым Google API средствам), и при обнаружении среди новых сообщений команд чата создаёт конфигурационный файл in.ini, в который записывает информацию обо всех обнаруженных в чате новых командах. После этого парсер запускает исполняемый файл обработчика команд и передаёт ему на вход путь к файлу с командами, а также путь к файлу, в который обработчик команд будет записывать результаты их выполнения (out.ini).

Для настройки парсера первым делом необходимо сгенерировать файл client\_secrets.json, содержащий в себе настройки, необходимые для получения доступа к Google API через OAuth2. Подробнее о том, как это сделать, написано в файле «Настройка OAuth2.docx». После успешного создания и копирования файла client\_secrets.json можно пробовать запускать парсер. При запуске парсера можно в аргументе указать ему путь к конфигурационному файлу StalkerInteractiveParser.ini, содержащему настройки парсера.

После запуска парсера с валидным client\_secrets.json должно открыться окно браузера, в котором будет предложено залогиниться в YouTube-аккаунт, с которого осуществляется трансляция. Если файл client\_secrets.json некорректен или пуст, при попытке запуска парсера произойдёт ошибка наподобие следующей:



Если после логина в YouTube-аккаунт парсер выдает следующую ошибку:



Это значит, что на данный момент в аккаунте не ведётся никаких трансляций. В случае успешного обнаружения активной трансляции парсер автоматически подключится к чату и начнёт мониторинг сообщений.

# Обработчик команд

Представлен исполняемым файлом StalkerInteractiveCommandsProcessor.exe. Данный файл запускается при поступлении в чат от зрителей команд интерактива и принимает на вход путь к конфигурационному файлу с параметрами этих команд и их отправителя, а также путь к выходному конфигурационному файлу с результатами выполнения команд. После запуска логика обработчика команд проходит по всем найденным командам и определяет, известна ли данная команда и доступна ли она в данный момент. Если команда доступна и пользователь может её запустить, обработчик записывает в выходной конфиг статус успешного выполнения команды и отправляет эту команду скрипту интерактива на исполнение. Для этого в директории $app\_data\_root$ создаётся файл с расширением \*.ic, содержащий всю необходимую скрипту интерактива информацию. Если команда недоступна – в выходной файл записывается статус, соответствующий причине, по которой команда не может быть выполнена.

Настройки обработчика команд хранятся в конфигурационном файле StalkerInteractiveCommandsProcessor.ini. Для передачи команд скрипту интерактива необходимо в параметре game\_appdata\_path секции [main] указать путь к игровой директории $app\_data\_root$. Расположение этой директории может отличаться в зависимости от игры или мода. Узнать его можно из файла fsgame.ltx, расположенного в корневой директории игры или мода. Например:



Каждая поддерживаемая команда должна иметь в файле StalkerInteractiveCommandsProcessor.ini свою секцию, название которой формируется как[названиекоманды\_command]. В данной секции можно задать минимальный промежуток времени между двумя последовательными командами этого типа (параметр timeout). Одноименный параметр в секции [main] задаёт минимальный промежуток времени между двумя любыми командами, которые будут исполнены.

Также в конфигурационном файле StalkerInteractiveCommandsProcessor.ini можно отключить выполнение команд – для этого достаточно удалить соответствующую команде секции или сделать её комментарием путём размещения в начале всех строк символа точки с запятой. Например, так:



# Скрипт интерактива

Непосредственно исполнительная часть, работающая внутри игры. Следит за появлением в директории $app\_data\_root$ новых файлов с расширением \*.ic, извлекает из них команды и безусловно выполняет соответствующие командам действия. Никаких проверок доступности команд на данном этапе не производится – этим занимается обработчик команд.

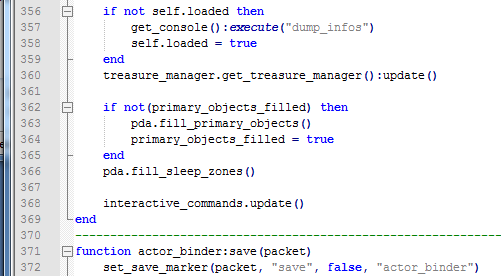
Для установки скрипта интерактива в игру необходимо cкопировать содержимое папки gamedata проекта в папку gamedata, расположенную в корне игры (если такой папки там нет – следует создать её). Затем нужно открыть в текстовом редакторе файл bind\_stalker.script, находящийся в директории gamedata\scripts игры (если его там нет, следует взять его из распакованной игры). Далее поиском по файлу необходимо найти в нём следующую строку:

function actor\_binder:update(delta)

Это – так называемая функция апдейта, периодически вызываемая движком игры. Необходимо добавить в её конец следующую строку:

interactive\_commands.update()

Для оригинальной игры версии 1.6.02 результат внесения правок должен выглядеть примерно так:



После установки в игру скрипт не требует какого-либо дополнительного конфигурирования.