

Как установить ботов для StarCraft II



(c) *Blizzard Entertainment*

В StarCraft II есть встроенные боты, и все с ними хорошо, за исключением того, что они немного тупые, но речь пойдет не о них. В 2017 году разработчик игры компания Blizzard Entertainment [опубликовала](#) API, позволяющий создавать внешних ботов. Однако, по какой-то странной причине Blizzard рассматривает этот API как сугубо исследовательский проект, где боты должны сражаться только друг с другом. Нормальной возможности поиграть человеку с ботами они почему-то не сделали, хотя многие геймеры рассматривают ботов как довольно неплохой инструмент тренировок.

Тем не менее все не так плохо. Раз штатной возможности нет, то будем использовать нештатную. Мы все же настроим ботов и повоюем против них. Потом мы с вами, движимые исследовательским порывом, разберемся как устроен API, и поднимем окружение разработки, в котором сможем сами писать подобных ботов.

Лабораторный стенд

В качестве лабораторного стенда использовалась виртуальная машина. Это было сделано по множеству причин, одна из которых заключалась в необходимости запускать неизвестные исполняемые файлы «из Интернета». Игра на виртуалке запустилась нормально и даже вполне

играбельно, правда с минимальными настройками графики. Использование виртуалки в качестве стенда дает ряд преимуществ: во-первых, можно запускать игру на не-Windows компьютерах, во-вторых, виртуалку можно использовать как основу для необнаруживаемых ботов-кликеров.

Виртуальная машина, используемая в экспериментах, работала под управлением **(важно)** Windows 10 x64. Операционная система устанавливалась с параметрами по умолчанию (Next, Next, Next...), и в ней был создан единственный пользователь «user», обладающий правами администратора.

Установка и подготовка игры

StarCraft II [бесплатная](#) игра, и процесс ее установки довольно простой:

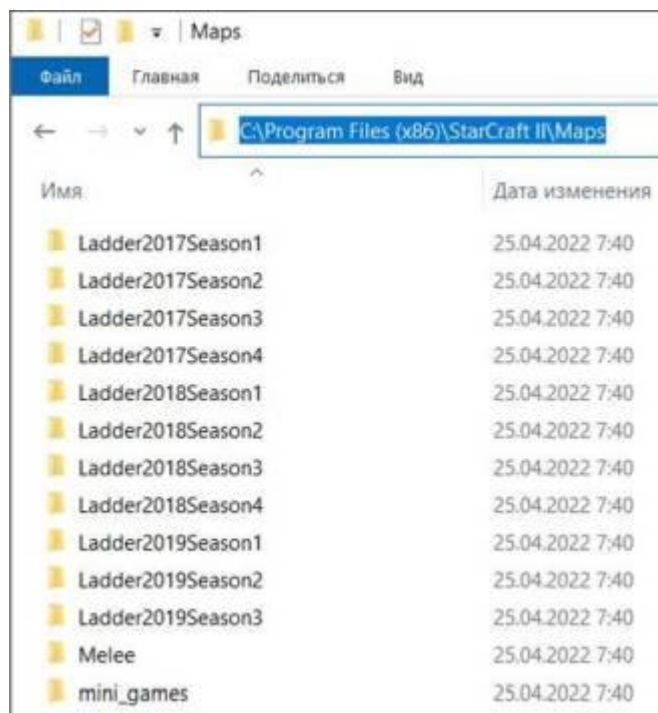
1. На [сайте Battle.net](#) создаем учетную запись.
2. Скачиваем и устанавливаем [Battle.net клиент](#).
3. Авторизуемся в клиенте под ранее созданной учетной записью и устанавливаем StarCraft II **(важно)** в каталог «C:\Program Files (x86)\StarCraft II\» (это каталог по умолчанию).

Для использования внешних ботов нам необходимо установить локальные игровые карты. По идее боты могут работать на любых картах, хоть на тех, что вы создадите с помощью встроенного редактора. Однако, для первого знакомства рекомендуется скачать карты, на которых проводятся [соревнования](#) ботов между собой (да-да, и такое есть). Дело в том, что в некоторых ботах названия карт жестко забиты в код, и поначалу будет довольно сложно понять, почему тот или иной бот не запускается.

Скачивать карты будем со специального [репозитория](#) Blizzard на GitHub. Рекомендуется скачать все доступные карты. Они там разложены по архивам на каждый сезон.

(Важно) В каталоге игры нужно сделать подкаталог «Maps» и распаковать архивы с картами туда. Пароль на архивы «iagreeetotheeula». Распаковывать архивы надо с сохранением структуры каталогов.

В нашем случае игра установлена в «C:\Program Files (x86)\StarCraft II\», соответственно каталог с картами будет «C:\Program Files (x86)\StarCraft II\Maps\». После распаковки карт все должно получиться как на скриншоте ниже:



Играем против ботов

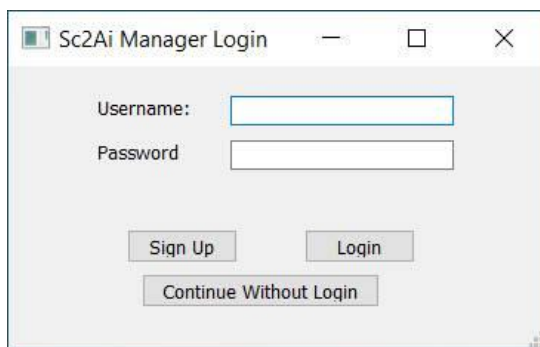
Для того чтобы попробовать свои силы против внешних ботов, нам потребуется специальный лаунчер — [SC2AI Mannager](#) (только Win10) от Cryptious админа [Discord канала SC2 AI](#). Справедливости ради стоит сказать, что некоторых ботов можно запустить и без лаунчера, но это, как говорится, следующий уровень сложности, и мы его рассмотрим чуть позже.

SC2AI Mannager позволяет как играть против ботов, так и срабатывать различных ботов между собой. Запускать лаунчер с произвольными ботами мы в данной статье не будем, а вместо этого возьмем [готовый набор \(лаунчер, карты, боты\)](#), который использовался в демонстрационных боях [ProBots 2021 Season 3 — ProBots vs Humans Exhibition](#).

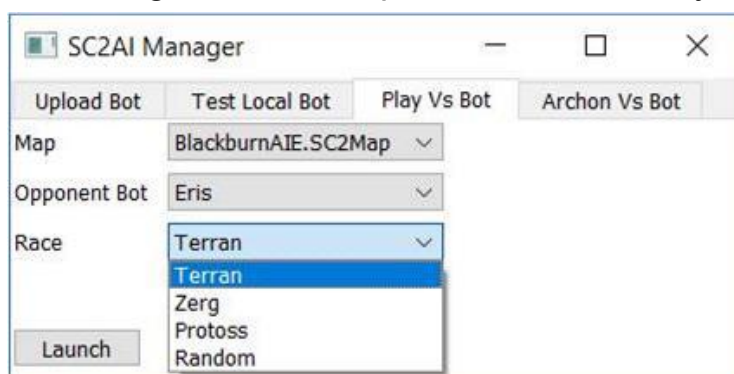
Использование набора сводится к следующим простым шагам:

Скачиваем набор и распаковываем его в произвольную папку на диске.

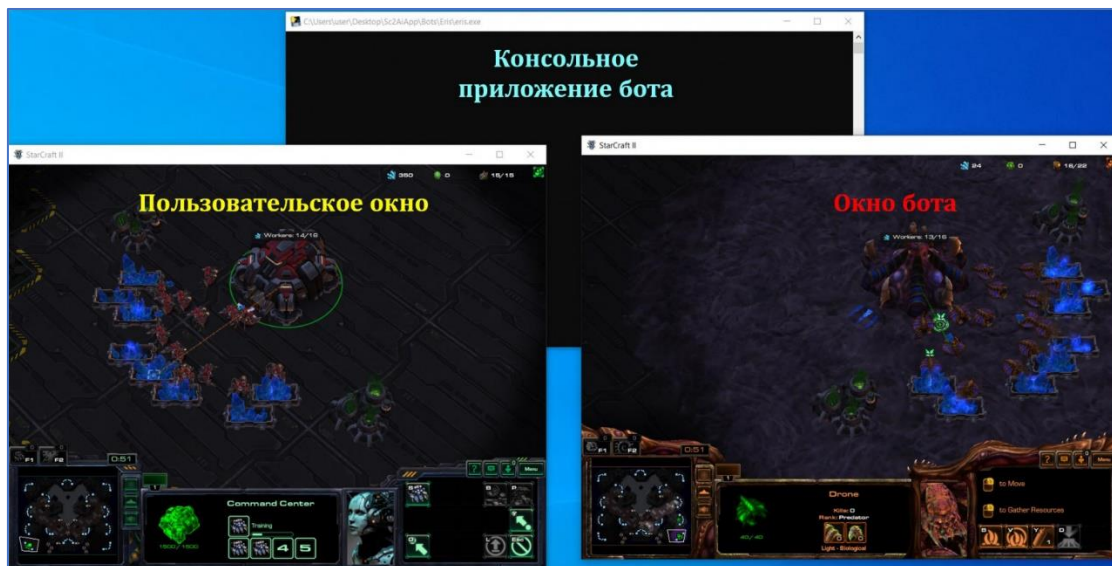
1. Карты из набора, то есть файлы имеющие расширение *.SC2Map, копируем в «C:\Program Files (x86)\StarCraft II\Maps».
2. Запускаем лаунчер Sc2AiApp.exe.



3. Он просит авторизоваться, но можно и без авторизации, ждем «Continue Without Login», затем переходим на вкладку «Play Vs Bot».



4. Выбираем расу, которой хотим играть и нажимаем «Launch». В результате должны запуститься 2 игровых клиента и консольное приложение бота.



5. Для нормальной игры в пользовательском окне нужно включить отображение на полный экран. Для этого необходимо нажать F10 (Option)->Graphics выбрать Windowed (Fullscreen)



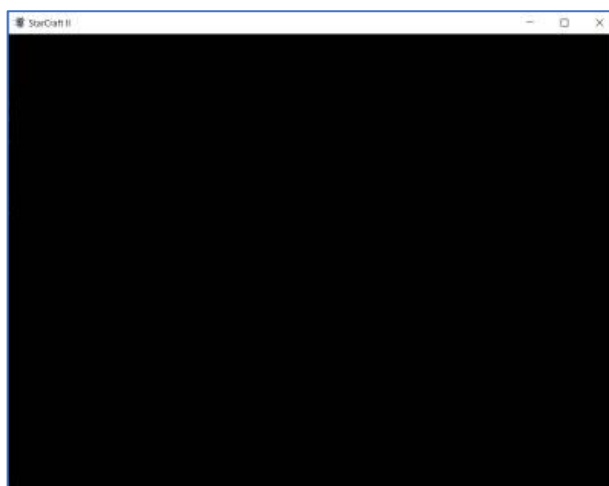
Архитектура StarCraft II и организация API

Что ж, быстрый старт позади. Давайте теперь разберемся чуть глубже. По умолчанию API внешних ботов в игре не работает, для его активации необходимо запустить клиент с особыми параметрами командной строки. Процедура запуска выглядит следующим образом:

1. Необходимо определить путь установки SC2 и определить разрядность клиента
2. Необходимо сделать рабочим каталогом директорию в которой хранятся библиотеки для клиента соответствующей разрядности. Для x64 клиента это «c:\Program Files (x86)\StarCraft II\Support64\»
3. Запустить клиент с указанием параметров командной строки. Например так: "«C:\Program Files (x86)\StarCraft II\Versions\Base87702\SC2_x64.exe» -listen «127.0.0.1» -port «18080» -dataDir «C:\Program Files (x86)\StarCraft II» -tempDir «C:\Users\user\AppData\Local\Temp\SC2_xp3zu_33» -displayMode «0» -verbose"

Примечание. Указанный пример взят из исходного текста одного из исследованных ботов.

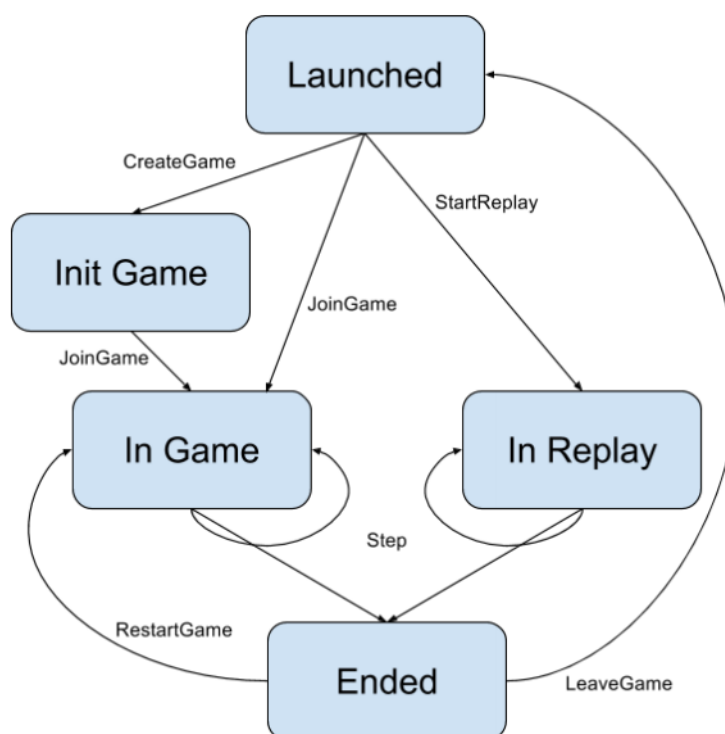
Если все сделали правильно, то запустится просто черное окно с заголовком StarCraft II, и больше ничего происходить не будет.



По факту клиент запустился нормально, он перешел в режим ожидания подключения по сетевому сокету (IP адрес + порт), который мы указали в качестве параметров командной строки. В примере выше это адрес 127.0.0.1 и порт 18080.

Для управления клиентом применяется специальный сетевой протокол, использующий в качестве транспорта [WebSockets](#), поверх которого реализован протокол [Google Protocol Buffers](#), параметры сериализации которого определены Blizzard и описаны в [спецификации](#).

API рассматривает игру как конечный автомат, имеющий следующие состояния:



(c) Blizzard Entertainment

Соответственно, когда мы запустили игру с помощью примера выше, она вошла в состояние «Launched» и далее ждала запросов на перевод в следующее состояние. Вот поэтому и был просто черный экран.

Этих знаний нам хватит для первого знакомства с устройством внешних SC2 ботов и того, как они могут работать в принципе. В дальнейшем, если будет желание что-то написать самому, то необходимо будет подробно ознакомиться с документацией на API из GitHub репозитория [Blizzard/s2client-proto](https://github.com/Blizzard/s2client-proto).

Развертывание среды разработки (Python)

За счет того что API основан на стандартном сетевом протоколе, разработка ботов может вестись на любом языке программирования, но мы рассмотрим только Python. Это обусловлено следующими причинами:

- Во-первых, Python довольно популярен среди разработчиков ботов, и для него написаны неплохие библиотеки (например, [BurnySc2/python-sc2](https://github.com/BurnySc2/python-sc2), [deepmind/pysc2](https://github.com/deepmind/pysc2)), помогающие в разработки собственных ботов.
- Во-вторых, подготовить подобную среду будет достаточно просто и доступно даже новичкам.

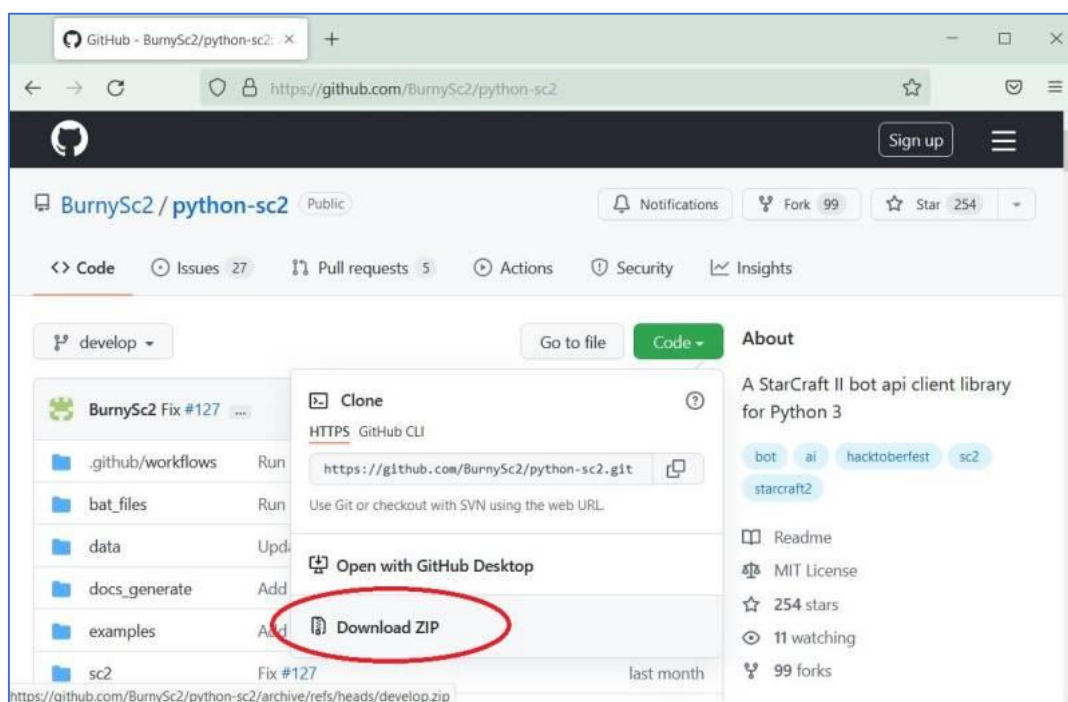
Приступим к подготовке среды:

1. Установим Python, скачав его с [официального сайта](https://www.python.org/). В текущем примере использовалась версия 3.10.4, однако некоторые боты могут требовать другую версию, так что перед началом посмотрите, какую версию рекомендует разработчик бота. Во время установки не забудьте добавить путь к python в переменную окружения PATH. Если вы все сделали правильно, то после установки, запустив консоль (cmd) и набрав «python», у вас должен запуститься интерпретатор.
2. Следующий этап — это развертывание одной из Python-библиотек, используемых для написания ботов. В этом примере мы воспользуемся BurnySc2/python-sc2. Для её установки в консоли наберем: `pip install burnysc2`

Если библиотека установилась правильно, то в консоли не должно быть сообщений об ошибках, а в каталоге Python-библиотек (например,

c:\users\user\appdata\local\programs\python\python310\lib\site-packages\) должен появиться каталог «sc2».

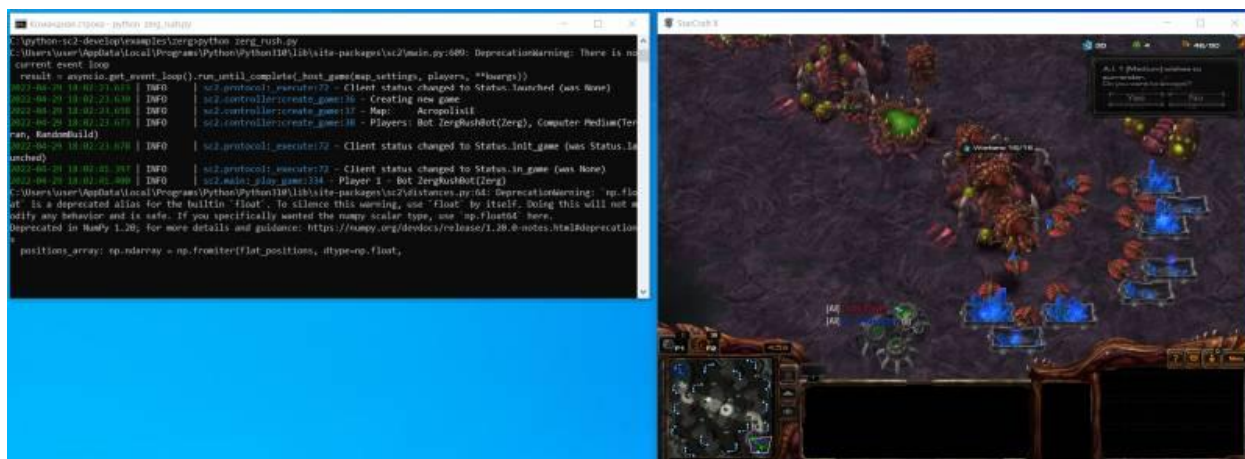
3. Установим дополнительную библиотеку six, которая потребуется для работы burnysc2
`pip install six`
4. Перейдем в Web-интерфейс GitHub-репозитория BurnySc2 и скачаем его весь как архив. Для этого нажмем на зеленую кнопку «Code», а затем в появившемся меню «Download ZIP».



Скачанный архив распакуем в произвольную папку, например, в c:\python-sc2-develop\.

Среда разработки готова. Теперь попробуем запустить бот-пример «Zerg-rush». Для этого откроем консоль и перейдем в каталог «C:\python-sc2-develop\examples\zerg\». После этого введем команду: `python zerg_rush.py`

В результате у нас должно запуститься консольное окно бота и окно клиента StarCraft II, где в ускоренном режиме бот «Zerg-rush» будет воевать со встроенным ботом среднего уровня сложности. По окончании боя будет записан реплей ZvT.SC2Replay.



В библиотеке BurnySc2 есть еще множество примеров ботов, но отдельного внимания стоит уделить примеру `play_tvz.py`. В нем показано, как запустить бота против человека.

Примечание. Пример содержит ошибку: в первой строке указан некорректный путь для импорта. Исправленный пример, предназначенный для запуска из каталога «C:\python-sc2-developer\examples\» выглядит следующим образом:

```
from zerg.zerg_rush import ZergRushBot
from sc2 import maps
from sc2.data import Race
from sc2.main import run_game
from sc2.player import Bot, Human

def main():
    run_game(maps.get("Abyssal Reef LE"), [Human(Race.Terran), Bot(Race.Zerg,
        ZergRushBot())], realtime=True)

if __name__ == "__main__":
    main()
```

Успешный запуск данного примера приведет к запуску 2-х клиентов SC2, в одном из которых будет бот, а во втором можно будет играть человеку.

Дополнительные материалы

- [SC2 AI Arena ladder](#) — сайт, посвященный соревнованиям ботов между собой.
- [SC2 AI Wiki](#) — Вики, по всему что связано с ботами для SC2.

- [GitHub компании Blizzard](#) — в нем следует обратить внимание на репозитории s2client-proto, s2client-api и s2protocol.
- [Статья компании DeepMind: AlphaStar: Mastering the real-time strategy game StarCraft II](#) — в ней говорится об опыте создания ботов для SC2 на базе глубоких нейронных сетей. [Русский перевод](#) этой же статьи.