

Боты на страже качества



Шуков Александр

QA Automation Tech Lead

shadow40kwrk@gmail.com



WARGAMING.NET

LET'S BATTLE

- В чем проблемы автоматизации тестирования игр?
- Какие подходы есть?
- Что такое bot-net?
- Как он устроен и почему именно так?
- Ограничения bot-net.



Что нам стоит автоматизировать тестирование игры?

Немного о играх и тестах

Автотесты в играх

Бизнес и состояние в отрасли



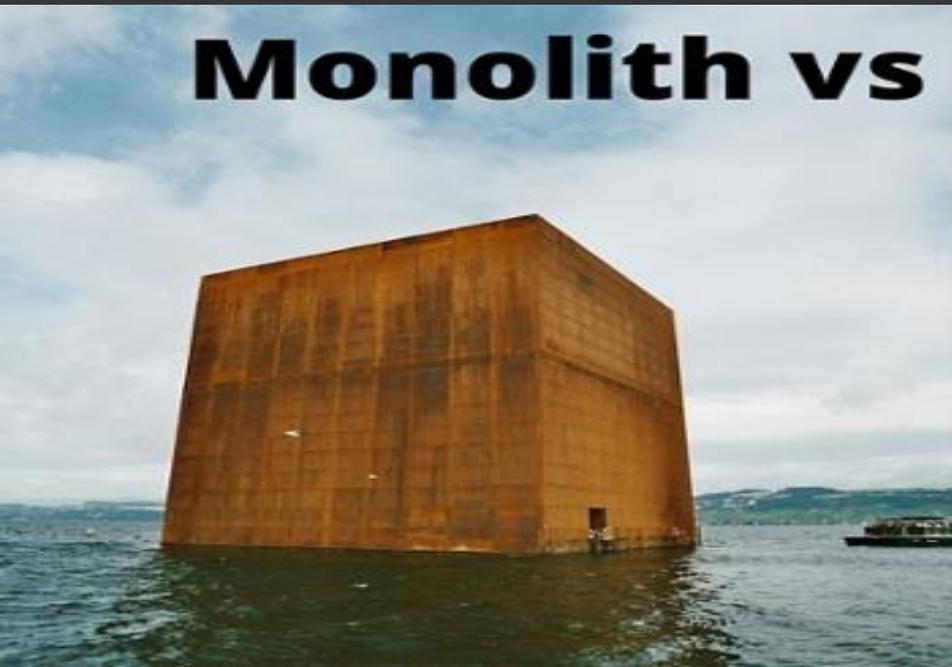
- коробочные игры — оплот ручного QA
- MMO | PaaS* игры — двигатель автоматизации*

**Product As A Service*

**автотесты игровой логики*

1. оптимизация и монолитная архитектура против всех

Monolith vs



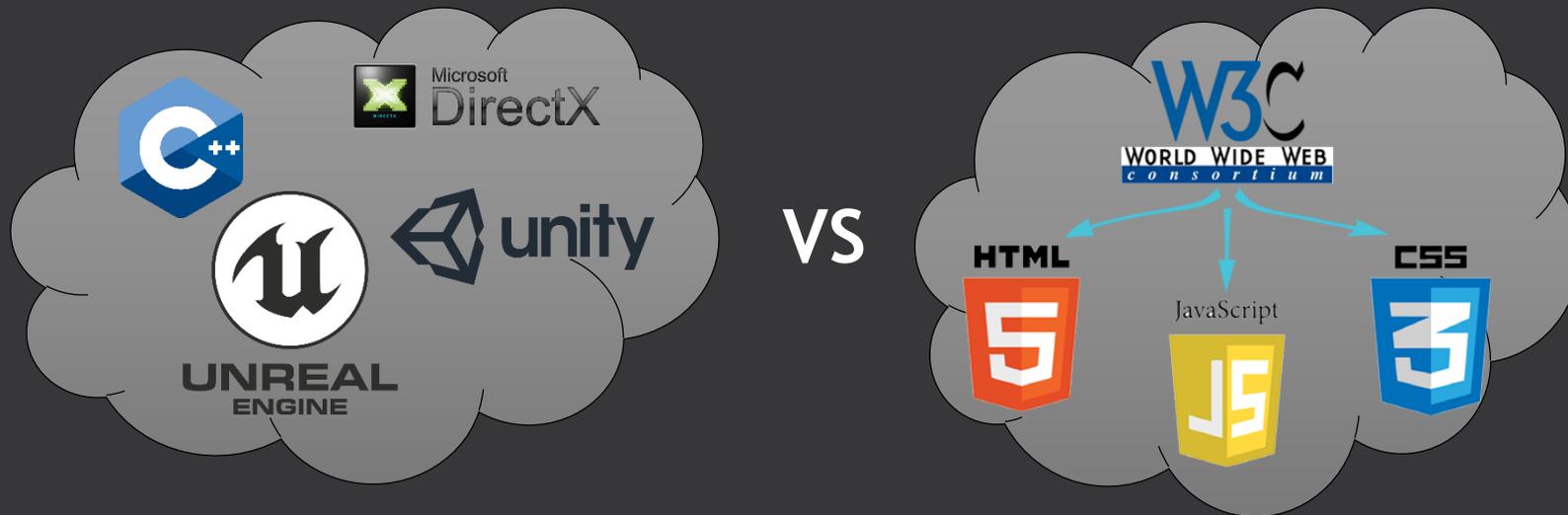
Microservices



Автотесты в играх

Технические сложности

2. мало стандартизации



Автотесты в играх

Технические сложности

3. слабо применимы наработки WEB-а





Что мы пробовали?

Технологии автоматизации тестирования игр

Кликеры

Autolt, WinTask, Sikuli...

+

- просты в освоении
- подходят под любое приложение

-

- сложно делать работающие тесты
- сложно^{^2} поддерживать
- сложно^{^3} анализировать результаты

Утилиты

WorldEditor-ы, админки и все, все, все...

+

- просты в использовании
- отлично покрывают — контент тестирование

-

- не успевают за игровой логикой
- сильно ограничивают область проверок
- сильная завязка на оператора

Инъекции кода

Хакерские техники на страже качества

- +
 - полное покрытие игровой логики
 - полный доступ к данным
 - реальный объект тестирования

- - очень сложны в освоении
 - сложны^{^2} в поддержке — работают только первый раз

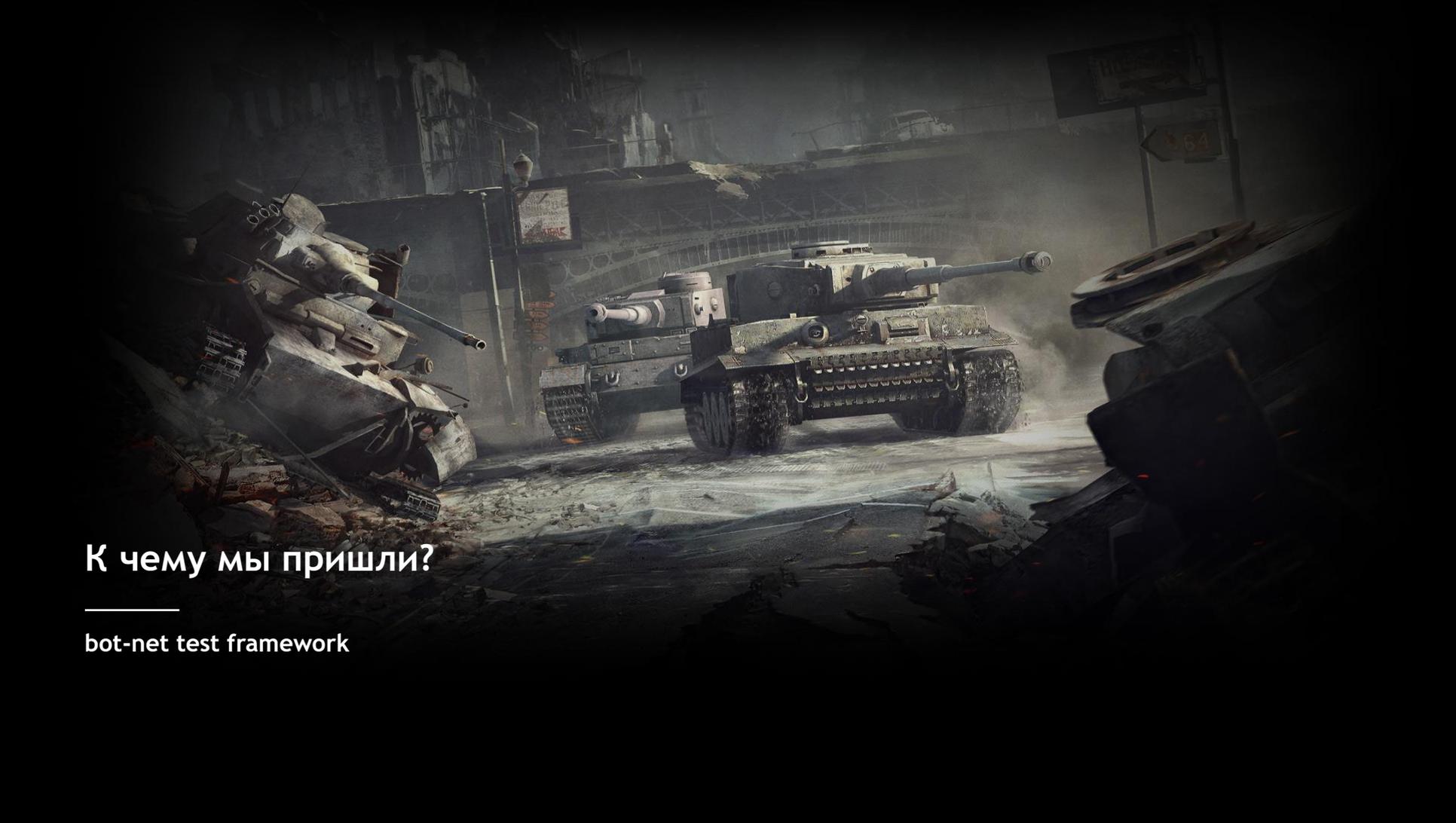
<https://www.slideshare.net/infodox/steelcon-2014-process-injection-with-python>

Юнит тесты

PyTest, Nose...

- +
 - быстрое выполнение
 - хорошо покрывают ядро (формулы, расчеты, форматы)
 - слабо завязаны на тестовую инфраструктуру

- - зачастую нужно править исходной код
 - сложно покрывать в середине жизни проекта
 - мало говорят о работе логики «в проде»



К чему мы пришли?

bot-net test framework

Что мы хотим?

Требования к системе

- работа с живым сервером WoT
- создание массовки
- оперирование игровыми сущностями (купить танк, захватить базу, проехать в точку, уничтожить цель)
- воспроизводимость багов
- вменяемая поддержка*

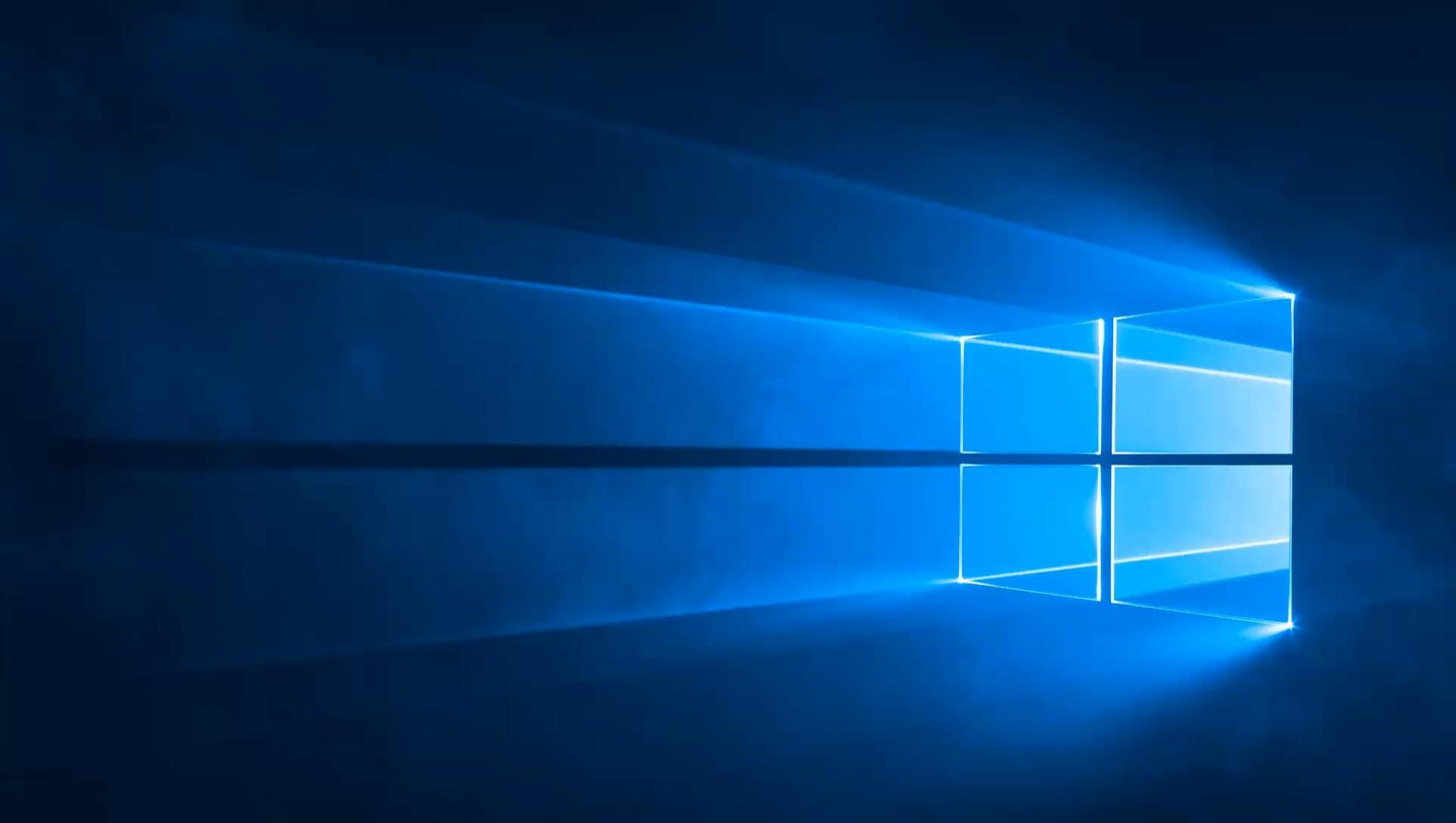
**рефакторинги, ревью, прозрачная связь с кодом*

Что мы получили?

bot-net!

bot-net — это:

1. эмуляторы игрока в WoT (боты)
2. пошагово выполняющие инструкции (не AI)
3. описанные в сценариях (в декларативном стиле)



Что такое bot-net?

Как мы его используем

- автотесты игровой логики (смоук, регресс, приемочный)
- нагрузочные тесты сервера
- подготовка тестового окружения



Под капотом у bot-net

разбираем систему шаг за шагом

Кто будет работу работать?

Среда исполнения

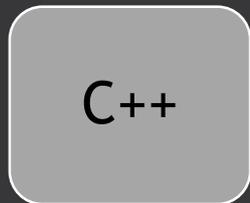
Как сделать эмулятор клиента?

... а зачем если можно взять сам клиент!

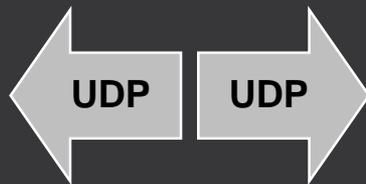
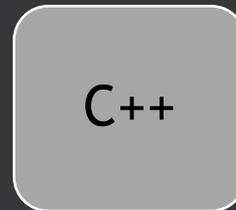
Что имеем?

Общие сведения о клиенте WoT

WoT Client

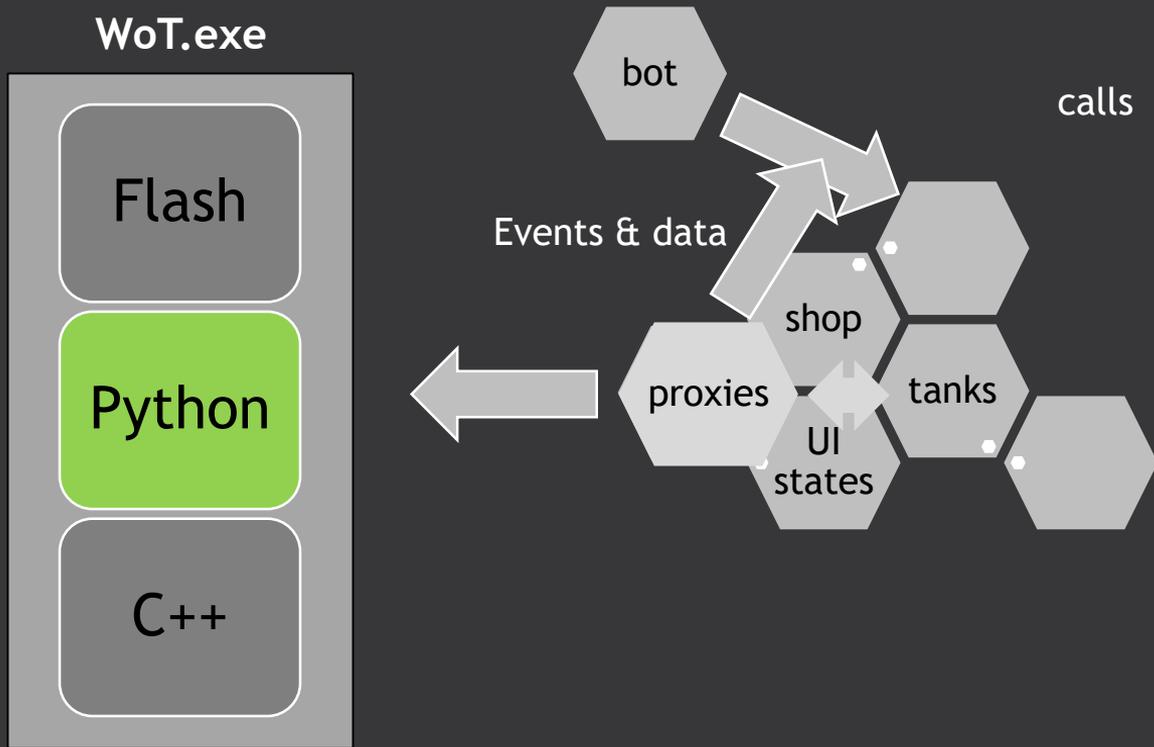


WoT Server



Кто будет работу работать?

bot-net ClientBot



Кто будет работу работать?

bot-net ClientBot – этого

+

- полноценный клиент
- полный доступ к игровой логике и данным
- end-to-end тесты

-

- 1 бот — 1 клиент
- тяжелый клиент
- медленный клиент



bot1_1 



bot1_1 Tiger II

1600/1600

FOREVER ALONE



Скриншот сохранен:
d:\ISVN_WOT\Trunk\bin\client\win32\screenshots\shot_002.jpg

 25	 0	 17
--	--	--



Что мы хотим?

Требования к системе

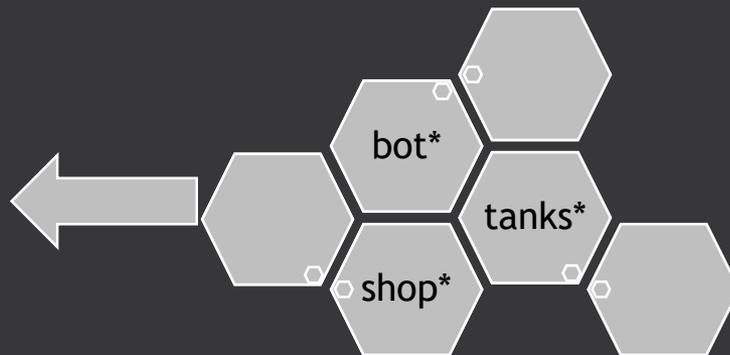
- ✓ работа с живым сервером WoT
- создание массовки
- оперирование игровыми сущностями
- воспроизводимость багов
- вменяемая поддержка

Кто будет работу работать?

bot-net HeadlessBot

bots.exe

WoT.exe



**легковесная реализация, без UI кода*

Кто будет работу работать?

bot-net HeadlessBot – умого

+

- выглядит как клиент для сервера
- много ботов внутри 1го процесса (100+)
- очень легкий и быстрый

-

- ничего не видно :(
- нельзя проверять: рендер, звуки, UI
- нужно писать легковесную реализацию игровой логики

bot1_1

- bot1_10
- bot1_11
- bot1_12
- bot1_13
- bot1_14
- bot1_15
- bot1_16
- bot1_2
- bot1_3
- bot1_4
- bot1_5
- bot1_6
- bot1_7
- bot1_8
- bot1_9



bot1_1 Tiger II

1600/1600

38

0 km/h

25 0 17



Скриншот сохранен:
 G:\SVN_WOT\11\trunk\bin\client\win32\screenShots\shot_001.jpg

Что мы хотим?

Требования к системе

- ✓ работа с живым сервером WoT
- ✓ создание массовки
- оперирование игровыми сущностями
- воспроизводимость багов
- вменяемая поддержка

Как писать сценарии?

Варианты:

- XML

Как писать сценарии?

XML scenario @ bot-net 0.1

```
<action>
  <!-- buy vehicle -->
  <mandatory> False </mandatory>
  <arg> CMD_BUY_VEHICLE </arg>
  <arg> shopRevision </arg>
  <arg> 3681 </arg>
  <arg> 1 </arg>
  <arg> 2 </arg>
</>
```

Как писать сценарии?

Варианты:

- XML
- язык движка игры
 - C
 - C++
 - C#
 - Python
 - LUA

Как писать сценарии?

Python scenario @ bot-net 1.0

```
def main(self):  
    vId = yield self.getTank('ussr:T-34')  
    botNames = yield self.waitBots(team_size=3)  
    botNames = sorted(botNames.keys())  
    creator = botNames[0]  
    roster = botNames[1:]  
    yield self.selectBattle('SQUAD', vId, creator, roster)  
    yield self.arena.waitForBattleStarted()  
    ...
```

Зачем на прошлом слайде **yield**?



Как все успеть?

Подружить ежа с ужом

Есть 2 задачи:

- всегда* ждать ответа от сервера
- никогда* не блокировать ожиданием

...решение — кооперативная многозадачность!

**на все «важные» операции*

**дольше 1кадра | 100мс*

Поддержка асинхронной работы

Синхронный код

- выполнение 1 кадр или 100мс
- не ждать ответа
- вызываем как обычные функции

```
def isClose2D(p1, p2, maxDistance=2.):  
    distance = Math.Vector3(p1).flatDistTo(Math.Vector3(p2))  
    return distance <= maxDistance
```

```
reachEnemyBase = isClose2D(base.position, veh.position)
```

Поддержка асинхронной работы

Асинхронный код

- выполняется >>1 кадра или 100мс
- нужно ждать ответ
- вызываем как future объекты ([asyncio](#) для python 3.5+)
 - или генераторы через **yield** ([PEP 342](#) для python 2.5+)

Асинхронный код

Мощь генераторов

...

```
vid = yield self.getTank('ussr:T-34')  
yield self.joinBattle(vid)  
yield self.moveTo(enemyBasePoint)
```

1 frame = 1 tick = 1 generator next

`@generator`

```
def joinBattle(self, vehicleID):  
    res = [None, None]
```

```
    def onJoined(*_):  
        res[0] = True
```

```
    def onFailed(errorCode, errorStr):  
        res[0] = False  
        res[1] = errorStr
```

```
    events.onEnqueuedRandom += onJoined
```

```
    events.onEnqueueRandomFailure += onFailed
```

```
    self.bot.player.enqueueBattle(vehicleID)
```

```
    yield self._waitForCondition(lambda: res[0] is not None)
```

Что мы хотим?

Требования к системе

- ✓ работа с живым сервером WoT
- ✓ создание массовки
- ✓ оперирование игровыми сущностями
- ✓ воспроизводимость багов
- ✓ вменяемая поддержка

... а что с поиском багов?

Как обеспечить поиск багов?



steps control

asserts

log parsing

Как обеспечить поиск багов?

steps control: @checked

```
"""Test BUY PREMIUM"""  
result = yield setMoney(gold=200)  
assert not result, 'Fail 2 get Gold!'  
result = yield buyPremium(1) # 1day = 150g  
assert not result, 'Fail 2 buy 1d Prem!'  
result = yield buyPremium(7) # 1w = 600g  
assert not result, 'Fail 2 buy 1w Prem!'
```

Как обеспечить поиск багов?

steps control: @checked

```
def checked(func):
    def _checked(self, *args, **kwargs):
        needCheck = kwargs.pop('check', True)
        res = yield func(self, *args, **kwargs)
        if needCheck and type(res) is _CheckedError:
            raise ScenarioError("{}: {} -
                                {}".format(func.__name__, res.message, res.data))
        yield self._return(res)
    return _checked
```

```
@checked
@generator
def buyPremium(self, days, arenaUID=0):
    return self._doCmd(AccountCommands.CMD_PREMIUM, (days, arenaUID))
```

Как обеспечить поиск багов?

steps control: @checked

```
"""Test BUY PREMIUM"""  
yield setMoney(gold=200)  
yield buyPremium(1) # 1day = 150g  
yield buyPremium(7, check=False) # 1w = 600g
```

Как обеспечить поиск багов?

scenario asserts

```
"""Test BUY PREMIUM"""  
yield setMoney(gold=200)  
yield buyPremium(1) # 1day = 150g  
result = yield buyPremium(7, check=False) # 1w = 600g  
assert result, 'Buy 1w without enough Gold!'
```

Как обеспечить поиск багов?

log's parsing

client.log

```
ERROR: [NvAPI] Get GDI primary display id: NVAPI_NVIDIA_DEVICE_NOT_FOUND
INFO: particle_lib: tPS_Shaders ::freeAll()
INFO: particle_lib: tPS_Shaders ::freeAll()
INFO: [Moo] TextureManager::delInternal: Could not find texture 'FlashBang' at 0x047C0460 to delete it
ERROR: [NvAPI] Unloading NvAPI: NVAPI_ERROR
INFO: [Connection] ServerConnection::createBasePlayer: id 4570
DEBUG: [DEBUG] (scripts/client/Account.py, 65): client Account.init
DEBUG: [DEBUG] (scripts/client/Account.py, 263): Account.onBecomePlayer()
DEBUG: [DEBUG] (scripts/client/account_helpers/Shop.py, 89): Shop.synchronize: cli_rev=0, serv_rev=None
WARNING: [WARNING] (scripts/client/messenger/proto/bw_chat2/VOIPChatController.py, 207): Initialize. Vivox is
not supported
DEBUG: [DEBUG] (scripts/bot/common/BotLog.py, 107): [BOT: bot1_1@test.iv][INFO: AccountController::create]
ERROR: [EXCEPTION] (scripts/client/MapActivities.py, 490):
Traceback (most recent call last):
  File "scripts/client/MapActivities.py", line 487, in __playSound
AttributeError: 'PyNullSoundObject' object has no attribute 'volume'
DEBUG: [DEBUG] (scripts/bot/common/BotLog.py, 107): [BOT: bot1_1@test.iv][ACTION: START_SCENARIO][scenario:
t1.py]
DEBUG: [DEBUG] (scripts/bot/common/BotLog.py, 107): [BOT: bot1_1@test.iv][ACTION: START_THREAD][pid: 1]
```

Как обеспечить поиск багов?

log's parsing

error_client.log

```
ERROR: [NvAPI] Get GDI primary display id:  
                NvAPI_NVIDIA_DEVICE_NOT_FOUND
```

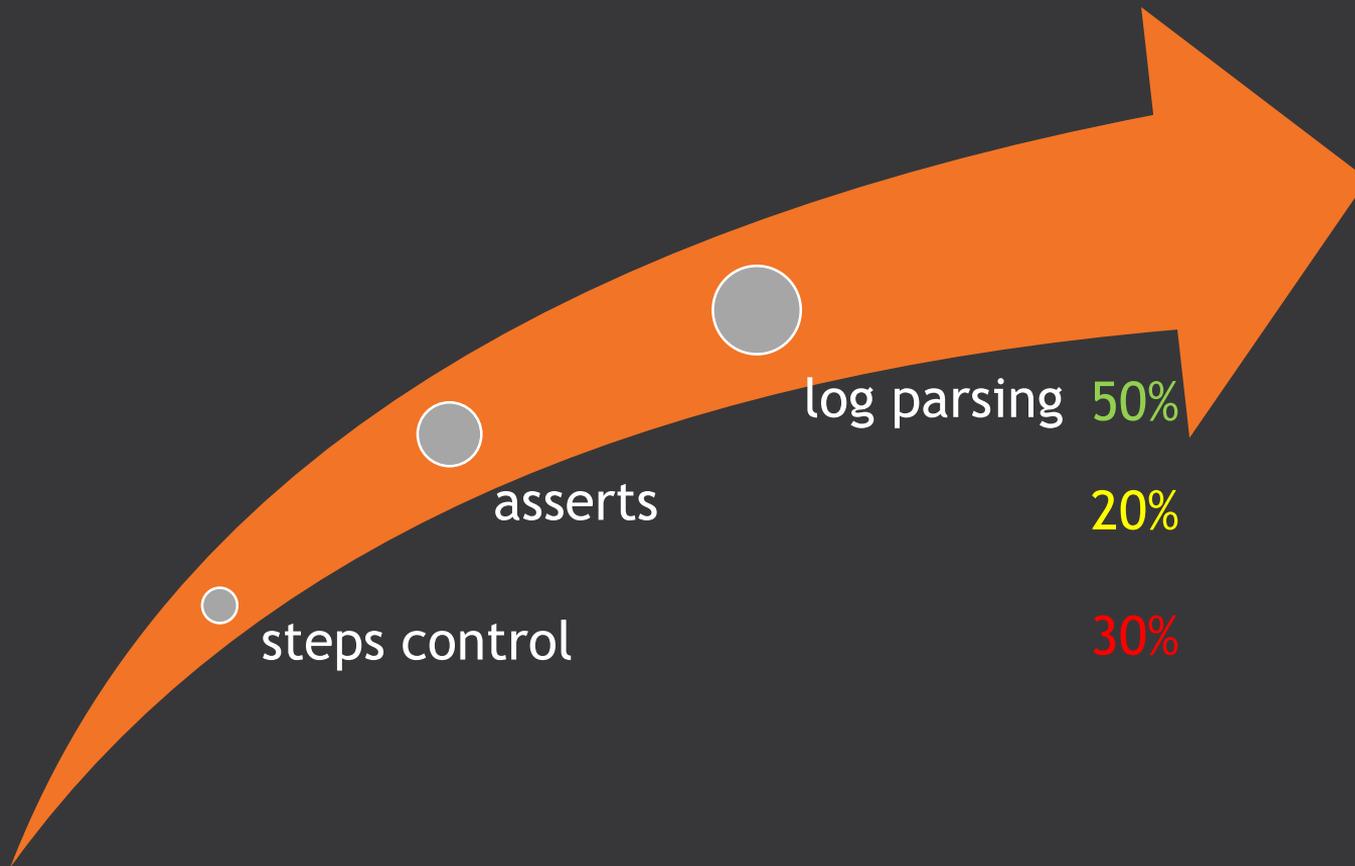
```
ERROR: [NvAPI] Unloading NvAPI: NvAPI ERROR
```

```
ERROR: [EXCEPTION] (scripts/client/MapActivities.py, 490):
```

```
Traceback (most recent call last):
```

```
  File "scripts/client/MapActivities.py", line 487, in __playSound  
AttributeError: 'PyNullSoundObject' object has no attribute 'volume'
```

Как обеспечить поиск багов?





Ограничения

bot-net может не только лишь все



Геопривязка ботов

```
POI = {  
  '02_malinovka': {  
    'attack': {  
      1: [(-371, 11, -204)],  
      2: [(-374, 12, -243),  
          (-161, 7.6, -373)],  
      3: [(-319, 11, -109),  
          (-154, 8, -372)]},  
    'center': [((203, 28, 230)],  
    'cover': [((-161, 7.9, -373.9),  
              -0.59), ]}]}
```

Навигация

Варианты решения

Доработка WorldEditor-а:

- указание точек с тэгами
- указание областей с тэгами
- экспорт в json

+ API со стороны bot-net:

- `getPointByTag`
- `getPointFromArea`

Параллельное выполнение генераторов

И врага подбить и корму не подставить

...

```
yield self.arena.moveTo(enemyBase)
```

```
yield self.arena.killEnemies()
```

...

Параллельное выполнение генераторов

Варианты решения

...

```
f1 = self.arena.moveTo(enemyBase)
```

```
f2 = self.arena.killEnemies()
```

```
yield self.waitAll(f1, f2)
```

...



Выводы

Как дальше жить и что с этим делать?

Выводы

Чему мы научились?

1. Прежде чем делать свой велосипед попробуй из магазина!



Выводы

Чему мы научились?

2. Если решился делать — продумай:

- где выполняется тест?
- как он описан?
- как он уживается с объектом тестирования (асинхронность)?
- как он будет находить баги?

Выводы

Чему мы научились?

3. Используй любую возможность найти баг!

- ответы от объекта тестирования
- ассерты
- логи
- скриншоты
- ...

Спасибо за стойкость!



Шуков Александр

shadow40kwrk@gmail.com

<https://www.linkedin.com/in/alexander-shukov-8628b7ba/>



WARGAMING.NET

LET'S BATTLE