

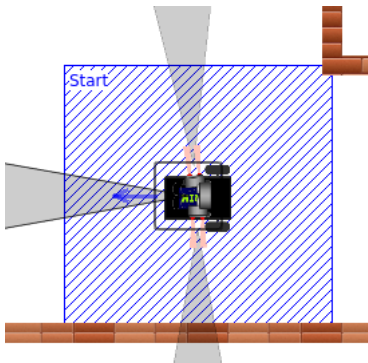
# Облачное тестирование программ TRIK Studio в среде Travis CI на примере задачи локализации в лабиринте

Балашов Илья Вадимович  
243 группа

Научный руководитель: ст. преп. Кириленко Я.А.

## О задаче локализации

- Известна карта
- Неизвестна точка и направление старта
- Доступны три датчика расстояния на роботе
- Надо определить координаты робота



Датчики на роботе

## Олимпиада НТИ, проблемы

- В финале надо запрограммировать работа
- Нет средств автоматической проверки решений
- Участникам нужно проверять каждый тест вручную в симуляторе
- “Эталонное” решение задачи локализации (сезон 2017-18 гг.) нечитабельно

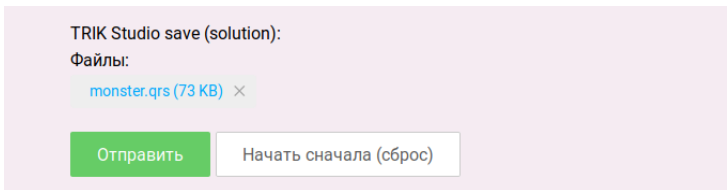
Целями работы являются:

- Сделать обзор существующих систем автоматического тестирования
- “Красиво” реализовать решение задачи локализации
- Создать систему облачного тестирования
- Интегрировать её с задачей локализации
- Провести проверку эффективности системы

# Обзор существующих решений 1

Решение на Stepik:

- + Готовая система тестирования в облаке
- Участникам нельзя загружать свои тесты
- Нестабильное поведение: зависание, прием не с первого раза



Окно сдачи решения на Stepik

## Обзор существующих решений 2

### Система EJudge:

- + Надежность
- + Решения тестируются в “песочнице”
- Неинтуитивность для участников
- Нужны серьёзные доработки для достижения поставленных целей

## Чекер ТРИК

- Разработан командой ТРИК
- Тестирует скрипт для конструктора на предопределённом наборе полей
- Генерирует отчет и путь робота по каждому тесту

## Реализации. Общая схема



Прецедентная диаграмма системы



## Реализация задачи локализации

- Метод “виртуального колеса” для прямолинейного движения
- Проблема “180/-180”
- Локализация методом гипотез
- Избегание циклов

# Управляющий скрипт

- Сборка образа Ubuntu с чекером
- Конфигурирование чекера, подгрузка тестов и решения
- Интерпретация и выгрузка результатов

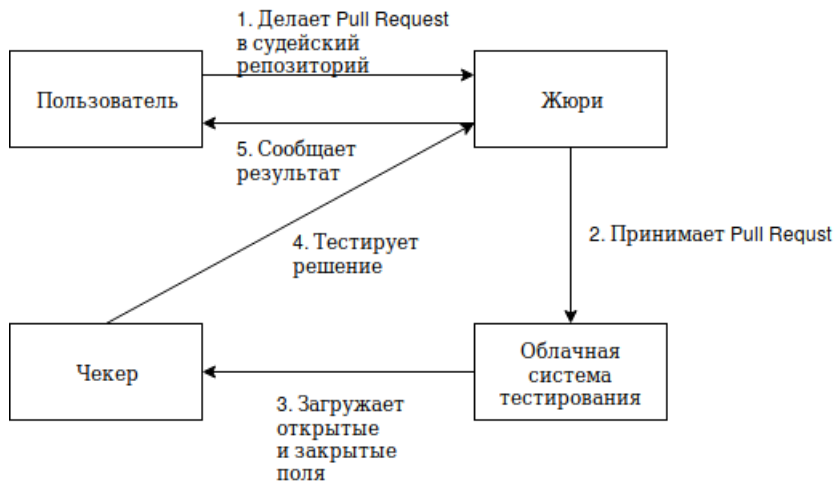
```
[ ' +-----+ ' ]
[ ' .....*.* ' ]
[ ' ***.***.*** ' ]
[ ' *.***.*.* ' ]
[ ' ***.*.***.*** ' ]
[ ' .....*.*.* ' ]
[ ' ***.*.***.*** ' ]
[ ' *.***.*.*.* ' ]
[ ' ***.***.*** ' ]
[ ' .....*.*.* ' ]
[ ' ***.***.*** ' ]
[ ' *.***.*.* ' ]
[ ' ***.***.*** ' ]
[ ' .....*.*.* ' ]
[ ' +-----+ ' ]
[ ' |.|..... ' ]
[ ' +-----+ ' ]
```

Пример задания карты в ASCII Art

## “Закрытые” поля

- Доступны жюри, недоступны участникам
- Travis CI Encrypted Values
- Pull Request в судейский репозиторий

## Общая схема сдачи решения

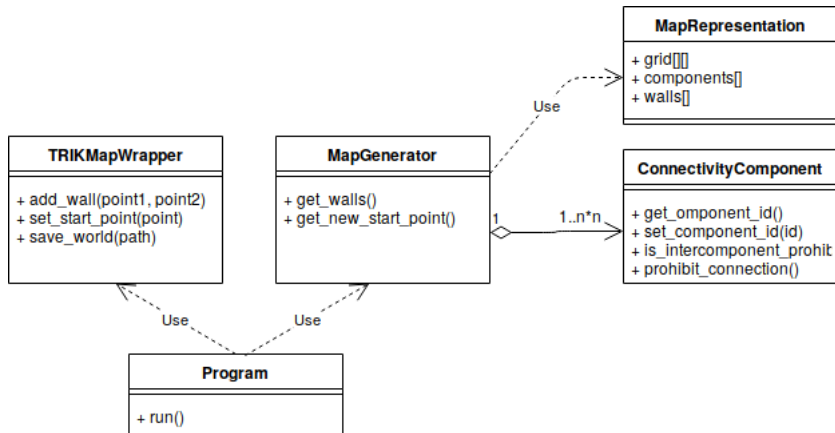


Последовательная диаграмма работы с системой тестирования

## Интеграция с задачей локализации

- Реализован генератор полей для задачи локализации
- Гарантирует:
  - Возможность локализации
  - Замкнутость, связность поля
  - Наличие циклических структур
- Консольная утилита и модуль для подключения
- Интегрирован с системой тестирования

# Архитектура генератора



Архитектура MapGenerator

## Анализ эффективности использования

- 3 студента матмеха СПбГУ
- 30 тестов
- **20 минут** – в среднем при ручном тестировании
- **7 минут** – при автоматическом

## Результаты

- Выполнен обзор существующих решений для автоматического тестирования
- Реализовано “эталонное” решение задачи локализации
- Создана система облачного тестирования
- Система интегрирована с задачей локализации
- Проведена проверка эффективности системы
  - Показано существенное преимущество автоматической системы