Министерство образования и науки Российской Федерации

Санкт-Петербургский государственный электротехнический университет

“ЛЭТИ”

Отчет по курсовой работе

«Битва Роботов»

Выполнили: Шевский В.С.

Факультет: КТИ

Группа: № 1303

Санкт-Петербург

2016

Оглавление

[1.Постановка задачи 3](#__RefHeading___Toc308_1795183420)

[2.Описание реализации роботов 3](#__RefHeading___Toc310_1795183420)

[3.Описание классов 5](#__RefHeading___Toc312_1795183420)

[3.1.Классы роботов 5](#__RefHeading___Toc314_1795183420)

[3.2.Классы оружия. 6](#__RefHeading___Toc439_1795183420)

[3.3.Класс потоков 7](#__RefHeading___Toc441_1795183420)

[4.Взаимодействие топиков 7](#__RefHeading___Toc318_1795183420)

[Выводы 8](#__RefHeading___Toc443_1795183420)

# 1.Постановка задачи

Вариант № 4. Битва роботов   
Исходная постановка задачи:   
Две армии роботов бьются друг с другом.   
Исходные данные:   
Есть несколько типов роботов, (быстрый, сильный, далеко стреляющий) которые должны расположится на карте и одолеть команду противника. Они подчиняются командиру, который раздает приказы, решает, когда робота нужно перебросить на другой фланг и прочее. Необходимо уничтожить вражеского командира.   
Ограничения на исходные данные:

* У роботов есть полоска здоровья.
* Роботам известна карта.
* Командир знает координаты каждого из своих роботов.
* Роботы могут видеть вражеских юнитов, но изначально не знают их координат.

# 2.Описание реализации роботов

В данной курсовой работе были реализованы следующие классы роботов :робот командир ,лучник и робот ближнего боя.

В задачи робота командира входит отдача приказаний роботам .В данной реализации у командира имеется только один приказ который он может отдавать — это приказ атаковать всем роботам.

Роботы подчиненные получая данный приказ начинают искать цель из списка целей который имеется у всех роботов и выбрав ближайшую начинают двигаться к ней и доехав до дистанции атаки атакуют цель.

Все роботы в отличие от атаки имеют одинаковую модель движения до противника. Задана она следующем образом. Как только робот узнает позицию врага — он сначала разворачивается на угол до противника затем двигается по этому углу до него.

Также роботы могут двигаться на заданную позицию но в данной реализации отдача командиром таких приказания не предусмотрено. Такое движение осуществляется следующим образом — сначала происходит движение по оси абсцис а затем по оси ординат.

Как только робот достиг дистанции атаки — он начинает атаковать. Все роботы атакуют в направлении куда они смотрят. Робот лучник атакует на дистанцию равную 5 перед собой но с углом разброс ±0.4 от направления взгляда. Робот ближнего боя атакует перед собой на дистанцию 1.Робот командир не атакует.

Робот лучник и командир имеет одну единицу здоровья .Робот ближнего боя — 3 единицы. Если позиция атаки куда про атаковал робот совпадает с позицией робота врага то он получает повреждения. У всех роботов во время повреждения отнимается по единице здоровья.

# 3.Описание классов

## 3.1.Классы роботов

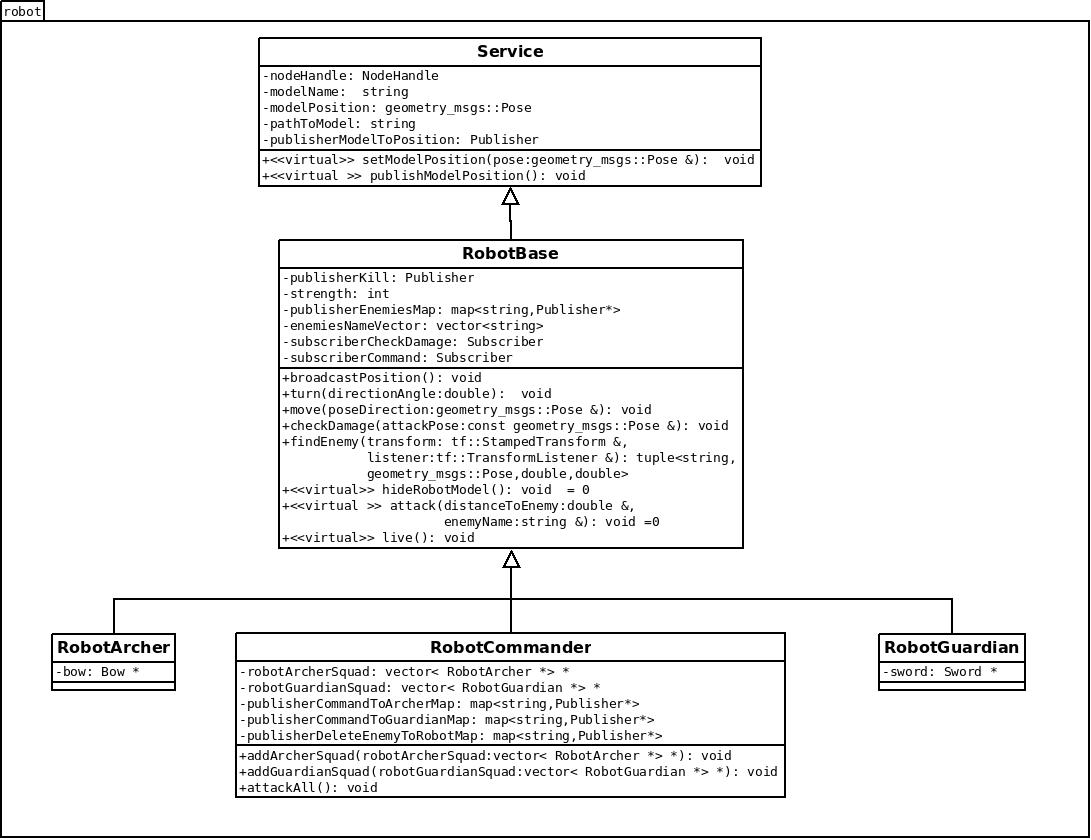


Рис.1 Диаграмма классов роботов

1)Класс Service — служит для создания моделей роботов ,оружия и снарядов в gazebo.

2)Класс RobotBase — базовый класс для всех роботов,который задает им основные операции характерные для всех роботов,такие как :

* broadcastPosition — посылает координаты позиции робота в tf.
* findEnemy- ищет ближайшего к роботу врага.
* сhechDamage — проверяет полученные повреждения
* turn — поворот на заданный угол.
* move — движение до заданной точки

следующие методы для каждого робота воина реализуются по своему:

* attack — в данном методе производится анимация атаки и отсылается врагу информация о позиции атаки.
* live — прослушивает приказы и начинает их выполнять.
* hideRobotModel — имитирует уничтожение робота путем скрытие модели под полем для битвы.

3)Класс RobotCommander -служит для отдачи приказов роботам, которые являются его подчиненными. Подчиненных ему роботов может добавлять следующими методами для лучников -метод addArcher и для ближнего боя -метод addGuardian и отдавать им приказ атаковать-метод attackAll.

4)Классы RobotArcher и RobotGuardian являются реализацией лучника и бойца ближнего боя

## 3.2.Классы оружия.

Данные классы представляют скелет дерева классов предметов в котором единственными значимым классом является Weapon который задает дистанцию атаки оружия .Остальные классы служат для отрисовки конкретной модели оружия .В данном случае это лук и меч(классы Bow и Sword).Класс Arrow отвечает за от рисовку модели стрелы.

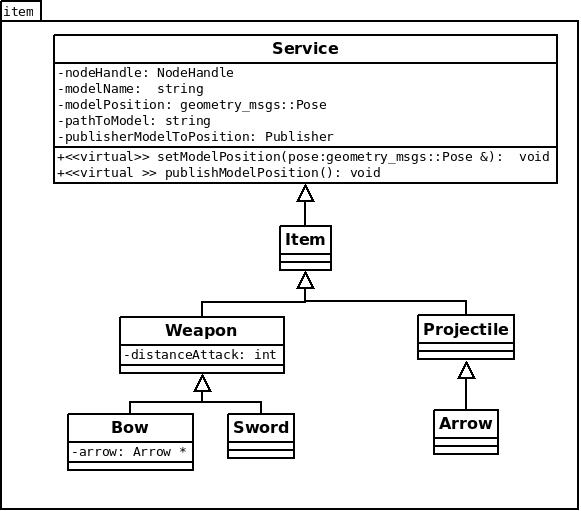


Рис.2 Диаграмма классов оружия

## 3.3.Класс потоков

Класс ThreadPool ставит каждого робота в поток в котором он принимает команды и выполняет их. Также этот класс останавливает потоки и удаляет роботов если они получили критические повреждения.

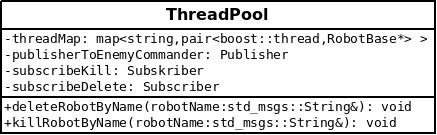


Рис.3 Класс потоков

# 4.Взаимодействие топиков

В программе имеется следующие топики:

* /имя\_робота/command — служит для передачи команд атаки. Тип передаваемых сообщения message:: CommandInformation.
* /имя\_робота/delete\_enemy — для передачи имени врага чтоб впоследствии удалить его из списка целей. Тип передаваемых сообщения std\_msgs::String.
* /имя\_робота/check\_damage — для передачи позиции атаки которая дальше сравнивается с текущим положением и в зависимости от этого повреждения считаются либо нет. Тип передаваемыхсообщения geometry\_msgs::Pose attackPose.
* /kill\_номер\_команды — для передачи имя робота , поток которого должен быть остановлен. Тип передаваемыхсообщения std\_msgs::String.
* /kill\_номер\_команды/delete - для передачи имя робота , объект которого должен быть удален. Тип передаваемых сообщений std\_msgs::String.

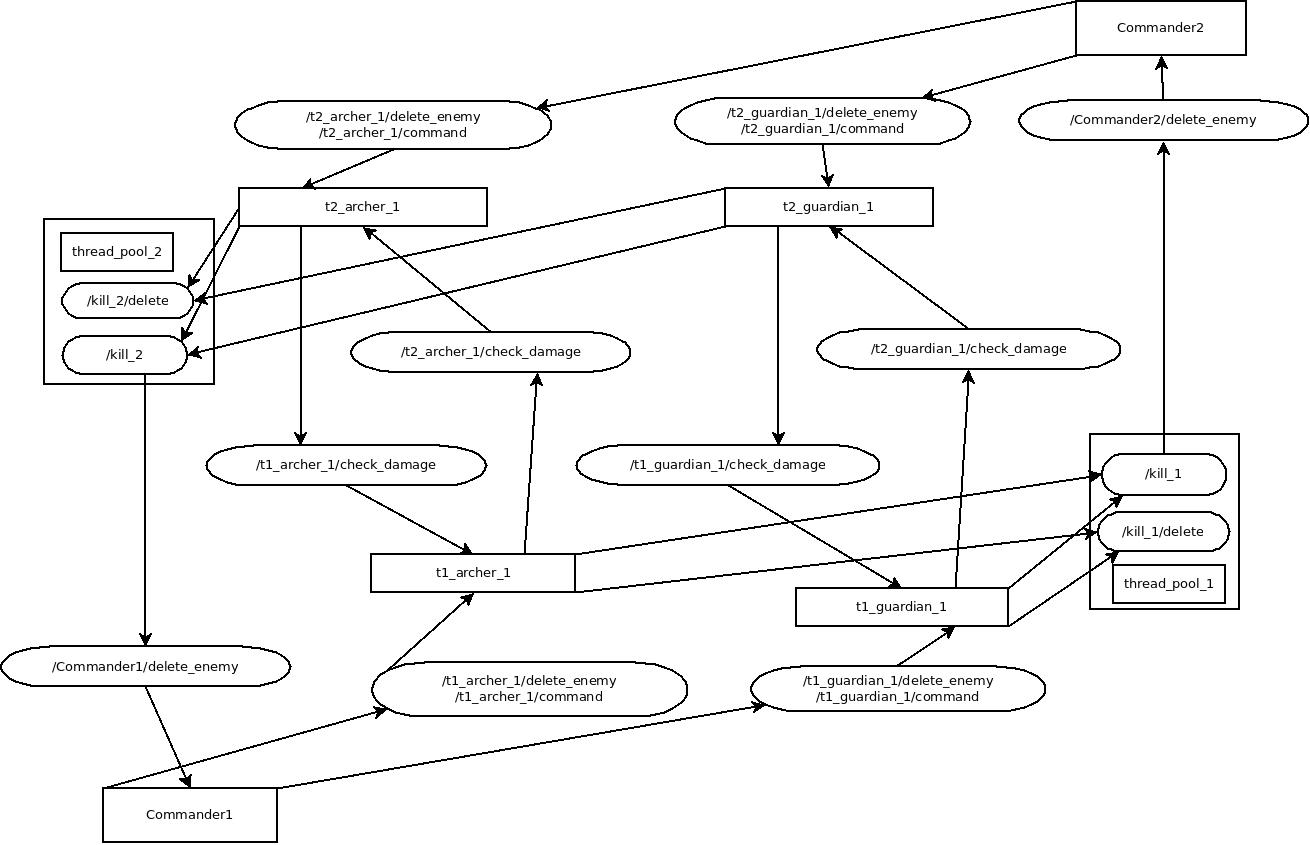


Рис.4 Схема взаимодействия топиков.

Рассмотрим поведение топиков на примере первой команды (на схеме внизу) где командир(Commander1)шлет команду атаковать. Разберем пример на одном работе лучнике t1\_archer\_1 так как для других роботов поведение будет идентично. Робот лучник через топик /t1\_archer1/command получает команду атаковать. Он выбирает ближайшую цель двигается к ней и атакует. После его атаки он посылает позицию атаки в топик /t2\_archer\_1/check\_damage(предположим он атаковал робота t2\_archer\_1).Робот t2\_archer\_1 проверяет повреждения и если он убит то посылает в топик /kill\_2 свое имя чтоб был остановлен поток в котором он выполняется.После остановки его потока он пошлет свое имя в топик /kill\_2/delete чтобы его объект был удален.После этого объект thread\_pool\_2 посылает сообщение с именем удаленного робота командиру первой команды в топик /Commander1/delete\_enemy.Командир рассылает это имя своим подчиненным это имя чтобы они удалиили его из списка целей.

# Выводы

Выполнено:

* 3 класса робота :робот командир ,лучник ,робот ближнего боя(сильный).
* Основные модели движения — движения до точки и до противника,поворот на заданный угол.
* Каждый робот живет в отдельном потоке , слушает приказы атаки и выполняет ее.
* Проверка повреждений и удаление роботов после критических повреждений.

Не реализовано:

* Отправка робота в указанную позицию командиром.
* Приказ атаки командиром врага по выбору.
* Атака командира.
* Командир пока не может являться целью и стало быть не шлет свои координаты в tf.
* Робот ближнего боя (**быстрый).**

**Во время битвы роботов было замечено что некоторые роботы не удаляют уничтоженного врага из списка целей а также не происходит своевременное удаление самих роботов. Связано это с тем что передача имен роботов в соответствующие топики для выполнения удаления врагов из списка целей и удаления роботов выполняется единожды и некоторые сообщения не доходят из-за чего могут возникать вышесказанные проблемы.**

Данную проблему можно исправить добавив вторичную отправку сообщений и соответственно проверку уже принятых сообщений чтобы не выполнять действия дважды.

Сама проблема заключается в том что хоть Ros и позиционирует себя системой реального времени ,но наделе таковой не является и потеря сообщений лишнее тому доказательство.