Санкт-Петербургский государственный электротехнический университет

«ЛЭТИ» им. В.И. Ульянова (Ленина) (СПбГЭТУ)

Кафедра МО ЭВМ

Системы реального времени на основе Linux

Отчёт по курсовой работе

на тему «Использование стандартных средств платформы ROS»

**Выполнили**: студенты гр. 1303

Завгороднев А. В.

Николаева К. Д.

**Проверил**: преподаватель

Филатов А. Ю.

Санкт-Петербург

2016

### Исходная постановка задачи

Зерги должны удержать аванпосты под атакой протоссов.

### Исходные данные

Зерги подверглись массированной атаке тамплиеров протоссов. Задача удержать аванпосты до тех пор, пока королева Керриган не уничтожит тамплиеров.

### Ограничения на исходные данные

* Зерги имеют только маленьких беспомощных зерлингов, которые не могут оказать сопротивление организованной атаке тамплиеров. Они лишь могут создавать живой щит, который может поглотить ограниченное число урона и имеет перезарядку.
* Зерглинги рождаются на аванпосте.
* Тамплиеры прилетают на шаттле в любую точку заранее определённой прямоугольной области, не содержащей аванпосты
* Тамплиеры протоссов уничтожают зерглингов с одного удара не по щиту.
* Тамплиеров уничтожает королева Керриган с одного удара.
* Зерлингов неограниченное число, но подкрепление может не успеть дойти к аванпосту, когда он окажется захвачен протоссами.
* Тамплиеров протоссов ограниченное число.
* Двое существ одной расы не могут располагаться в одной точке (они должы стоять рядом)
* Зерги побеждают, к**о**гда все протоссы уничтожены; протоссы побеждают, когда захвачен хотя бы один аванпост
* Аванпост считается захваченным, когда в радиусе контроля нет ни одной единицы фракции Зергов, но есть хотя бы один тамплиер.
* Боевые единицы не умеют обходить препятствия.

### Описание основного принципа ROS

В платформе ROS любая исполнительная единица называется «узлом». Для коммуникаций между узлами используется механизм сообщений и, так называемых, «топиков» (topic). Топик – это интерфейс узла для коммуникаций. В топик можно посылать сообщения (в этом случае говорится, что узел публикует сообщения), или считывать сообщения из топика (в этом случае говорят, что узел подписан на топик). В ROS существует множество стандартных сообщений, также есть возможность создавать свои структуры сообщений и использовать их. Ниже приведен пример в виде схемы:

C:\Users\ckyxc\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Untitled Diagram.png

Рис. 1. Пример узлов и топиков ROS.

Т.е. на данном примере узел Robot подписан на топик /cmd\_vel, принимает команды от узла Enviroment и посылает сообщение о результате выполнения команды в топик /cmd\_result. Узел Enviroment публикует команды в топик /cmd\_vel, и читает результат выполнения команды роботом из топика /cmd\_result.

### Описание стандартного пакета tf

Пакет tf служит для упрощения определения в пространстве координат различных объектов. Например известно, что относительно объекта 1 объект 2 имеет координаты (x,y). А объект 3 относительно объекта 2 - (m,n). Для того, чтобы выяснить взаимное расположение объектов 1 и 3 можно применить формулы высшей математики, однако в этом нет необходимости, поскольку именно для этого предназначен пакет tf.

Tf – это пакет, который предоставляет набор топиков. Один из них - топик, в который можно посылать сообщения о координатах объекта относительно другого объекта. Другой – топик, из которого можно считать координаты объекта, причем относительно третьего объекта, который также посылает свои координаты в tf. В таком случае пакет tf сам произведет все вычисления, которые необходимо выполнить при вычислении относительного положения объектов.

### Описание структуры программы

При выполнении задания была разработана программа, которая использует особенности ROS, приведенные выше.

Программа имеет следующую иерархию классов:

C:\Users\ckyxc\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Untitled Diagram (1).png

Рис. 2. Иерархия классов.

На данной схеме класс RenderWindow не входит в разработку и является частью библиотеки SFML, которая использовалась для визуализации.

Зерглинги являются объектами класса DefenceUnit, а тамплиеры объектами класса AttackUnit. Названия остальных классов говорят сами за себя.

Фракции зерглингов и протоссов изолированы друг от друга, т.е. единицы фракции зерглингов не имеют возможности узнать положение протоссов на карте непосредственно через объекты и наоборот. Для того, чтобы взаимодействовать с единицами из другой фракции каждая единица (в том числе аванпосты) публикуют свои координаты в tf, что предоставляется функционалом класса Unit. Также, факт получения урона отслеживается через топик /input, который так же предоставляется классом Unit. Таким образом, при атаке, единицы просто публикуют сообщения в соответствующий топик.

Классы фракций объединяют логику всех единиц, относящихся к ней, собирают из tf’а координаты вражеской фракции и управляют своими единицами на основе этой информации. Класс World просто соединяет в себе логику фракции и логику визуализации. Логика визуализации состоит только в том, что необходимо обрабатывать коллизии единиц, путем проверки границ спрайтов и отображении мертвых единиц «под» живыми.

### Заключение

Использование платформы ROS при реализации данной задачи внесло свои сложности, но в то же время такой подход позволил обеспечить изолированную логику, так что изменения в логике одной фракции никогда не повлекут изменений в другой.