

## Тестовое задание на позицию Game Backend developer

### Цель:

Необходимо продемонстрировать знания принципов проектирования, обеспечить простоту и расширяемость кода.

Код должен быть понятен разработчику с квалификацией Junior Developer.

В этом должно помочь понимание: KISS, YAGNI, DRY, SOLID, Low Coupling и High Cohesion

\*\* В процессе выполнения задачи вы можете прислать вопросы для уточнения цели и задания. \*

### Задача

Дана карта размера  $H \times W$  клеток, на карте размещается  $N$  воинов, которые могут маршировать в любую клетку.

Марш воина продолжается определенное время в зависимости от расстояния от места расположения воина:

Время в абстрактных тиках =  $\sqrt{(X_b - X_a)^2 + (Y_b - Y_a)^2}$

Воины имеют силу, которая указывается в файле сценария.

Воина можно отправить в марш, марш нельзя отменить.

Марши воинов идут одновременно, независимо и никак не взаимодействуют пока не дойдут до точки.

Когда воин пришел в точку:

- если точка пустая - воин ее занимает
- если сила воина больше, чем сила соперника, то воин его убивает.
- если сила равна, то умирают оба.
- победитель занимает клетку.

Погибшие воины удаляются с карты, их больше нет.

При запуске - приложение получает в **argv** путь к файлу со сценарием работы, в котором описаны **команды**:

```
CREATE_MAP 30 30 // создать карту(может быть ток в самом начале)
SPAWN 1 10 20 100 //разместить воина №1 в точке x=10 y=20 с силой 100
SPAWN 2 11 21 50
SPAWN 3 19 20 102
SPAWN 4 13 23 103
SPAWN 5 13 20 102
```

# SCOREWARRIOR

```
MARCH 1 11 21 // отправить воина №1 в точку 11 21
WAIT 10 // время ожидания в абстрактных тиках, до того, как выполнить следующую команду
MARCH 4 14 24
WAIT 10
MARCH 3 16 20
MARCH 5 16 20
```

Программа должна дождаться окончания всех маршей и завершиться с выводом произошедших **событий** в **stdout**, результат работы в виде:

```
[CURRENT_TICK] MAP CREATED 30 30 // карта создана
[CURRENT_TICK] WARRIOR SPAWNED 1 ON 10 20 // воин создан в координатах 10 20
[CURRENT_TICK] WARRIOR SPAWNED 2 ON 11 21
[CURRENT_TICK] WARRIOR SPAWNED 3 ON 19 20
[CURRENT_TICK] WARRIOR SPAWNED 4 ON 13 23
[CURRENT_TICK] WARRIOR SPAWNED 5 ON 13 20

[CURRENT_TICK] MARCH STARTED 1 TO 11 21 // марш воина 1 запущен в координаты 11 21
[CURRENT_TICK] MARCH 1 FINISHED 11 21 BATTLE 1 2 WINNER IS 1 // марш воина 1 завершён, в битве 1 и 2
воина победил 1
[CURRENT_TICK] MARCH STARTED 4 TO 14 24 // марш воина 4 запущен в координаты 14 24
[CURRENT_TICK] MARCH 4 FINISHED 14 24 // марш пришел, занял пустую клетку
[CURRENT_TICK] MARCH STARTED 3 TO 16 20
[CURRENT_TICK] MARCH STARTED 5 TO 16 20
[CURRENT_TICK] MARCH 3 FINISHED 16 20
[CURRENT_TICK] MARCH 5 FINISHED 16 20 BATTLE 3 5 ALL DEAD // марш воина 5 завершён, в битве 3 и 5
воинов никто не победил, все умерли

[CURRENT_TICK] FINISH // все марши закончены, CURRENT_TICK должен соответствовать времени
окончания последнего марша
```

Формат ввода/вывода именно такой.

*Если вдруг вам стало скучно, то можем предложить так же выполнить задания со звёздочкой 😊*

## Требования со звёздочкой 1\*

К чему должен быть готов проект, если бы он имел в будущем дальнейшее развитие?

1. Количество Команд и Событий будет сильно расширяться множеством людей в команде одновременно
2. Формат ввода и вывода может быть изменен на другой (к примеру, json или даже бинарный)
3. Помимо воинов могут быть введены юниты с другим поведением и новыми характеристиками
  1. к примеру, лучник стреляющий в другого юнита как только он попал в его радиус (характеристика).

Это не нужно реализовывать, но решение должно быть готово к подобным изменениям, нужно будет рассказать на втором собесе как бы вы это сделали.

# SCOREWARRIOR

**Задание со звёздочкой 2\*** - Если воин проигрывает в битве с другим воином, то он не умирает, а идёт автоматически обратно в точку, из которой пришёл. По возвращению марша, может произойти битва по обычным правилам - более слабый воин умирает.

пример:

[CURRENT\_TICK] MARCH 1 FINISHED 11 21 BATTLE 1 2 WINNER IS 2 // марш воина 1 завершён, в битве 1 и 2 воина победил 2

[CURRENT\_TICK] MARCH STARTED 1 TO 10 20 // марш воина 1 запущен обратно в координаты 10 20

Мы ценим время наших кандидатов, поэтому стоит учитывать, что мы рассчитывали предельное время, которое стоит уделять заданию.

*Планируемое время на выполнение основного задания - не более 12 часов*

*Планируемое время на выполнение основного задания + требования со звёздочкой 1 - не более 16 часов*

*Планируемое время на выполнение основного задания задания + задание со звёздочкой 2 - не более 14 часов*

*Максимальное время, которое стоит уделять всем заданиям и требованиям - не более 20 часов.*

**Технические требования:** ОС: Ubuntu

Компилятор: clang

Система сборки: cmake (в проекте должен быть CMakeLists.txt)

Задание необходимо опубликовать в любом гит-репозитории (github, bitbucket, и т д).

- Убедитесь, что к нему есть доступ по ссылке без авторизации