

Обяснение към кода на чекъра

Повечето централни настройки се намират в *main.hpp*.

Някои по-интересни настройки (изискват Rebuild):

1. *DEBUG_LEVEL* – ако е 2 всички дебъг съобщения ще се изпринтват.
Ако е 1 – само по-важните дебъг съобщения ще се изпринтват.
Ако е 0 – само резултата ще бъде изпечатан.
2. *OS_PATH_SEPARATOR* – за Windows '\\', иначе '/'. **Моля променете го спрямо вашата Операционна система.**
3. *FORMAT_int64* – указва правилния форматиращ стринг за printf/scanf, когато се чете unsigned long long. За Windows "I64u", за Linux – "llu". **Моля променете спрямо вашата ОС.**
4. Следват съответните константи за инструкциите и ограниченията.

По файлове: (по двойки *.hpp/.cpp*, като *.cpp* може да не присъства)

main – логиката на тестването (*.cpp*) и конфигурация на целия симулатор (*.hpp*)

Mob – описва едно животно.

RunningMob – описва животно, което е част от вълна. Разликата с Mob е, че тук се пази къде е животното на картата, какъв е неговият моментен живот (а не само максималният му, както в Mob) и т.н.

Turret – описва кула.

PlacedTurret – еквивалента на RunningMob. Описва кула с локация на картата и т.н.

Field – описва полето, пази статуса на всяко поле (заето с кула, част от пътя, блокирано, свободно). Помагачи структури са FieldCell (описва за полета, част от пътя, кои виждат това поле, а за заети полета – коя кула от максимум 64те заема това поле). Ако кула N вижда поле, то в съответната FieldCell структура за това поле, бит N ще е 1.

SegmentTree, Quadtree – помагачи дървета, подпомагачи бързото вземане на всички кули, които ще видят едно животно за 1 секунда (по време на пътя от началната до крайната позиция), както и с колко трябва да се променят бързината и щетата на кула.

Game – клас, който управлява инстанция на Field, пази животите, парите, кулите и т.н. Поддържа закупуване на кула и продаване на кула (но без ъпгрейди).

TextGame – наследява Game и поддържа ъпгрейди за кулите. Пази информация не само за закупените кули (наследена от Game), но и за всички кули, които могат да бъдат закупени, дори и да не са. Аналогично пази информация за животните и кулите.

TextGameReader – чете входящият файл във описания в условието файл и връща TextGame*.

WaveRunner – Съдържа логиката това как всяка вълна се симулира. Тъй като се тестваха два вида логики, тествайте само с waveRunLogicManyCalls (кода, който ще бъде предоставен прави това по подразбиране).

WaveInfo – съдържа статистика, връщана от изпълнението на дадена вълна.

ManyRunsLogicInfo – съдържа информация за зададените подредби и максималните щети за всяко поле.

Моля, ако намерете някой бъг, пишете по някой мейл ☺ Ако искате да правите нещо сложно по кода и нещо не ви е ясно, питайте в сайта.