

# Звёздные Голодные Игры

Летняя Олимпиадная Школа

Июнь 2019

## 1 Правила игры

Действие игры происходит на обречённой платформе, несущейся сквозь пространство бушующего космоса будущего.

Платформа представляет из себя прямоугольное поле размера  $N \times M$ ,  $300 \leq N, M \leq 1500$ , состоящее из клеточек размера  $1 \times 1$ .

На платформе находятся различные объекты, в том числе игроки **Звёздных Голодных Игр**. Каждый игрок занимает  $3 \times 3$  клеточки и имеет направление взгляда (с точностью до 90 градусов: на север, восток, запад или юг).

Игрок видит всё, что находится в полуплоскости его зрения на расстоянии не более 150 единиц от него.

Игроки могут перемещаться по клеточкам, взаимодействовать с собой (путём питания), друг с другом (путём атаки световым мечом) и с игровым миром (путём перемещения или выстрела из лазерного бластера).

Каждый игрок имеет следующие характеристики:

- **Очки здоровья:**  $0 \leq player_h \leq 250$ . Они расходуются при получении урона от других игроков или окружающего мира и восполняются при достаточном количестве жизненной энергии. Потеря всех очков здоровья приводит к смерти игрока (и возможному перерождению на некоторых картах). При старте  $player_h = 150$ .

- **Очки жизненной энергии:**  $0 \leq player_s \leq 300$ . Они расходуются при действиях игрока и пополняются при потреблении энергетических батончиков. При старте  $player_s = 200$ .
- **Очки результатов:**  $-\infty < player_r < \infty$ . Они вычисляются по некоторой формуле и используются для определения победителя в турнире стратегий. Чем выше количество очков, тем лучше результат стратегии. При старте  $player_r = 0$ .
- **Энергетическая ценность** всех энергетических батончиков в инвентаре:  $0 \leq player_f < \infty$ . Энергетическая ценность батончиков может быть сконвертирована в очки жизненной энергии. При старте  $player_f = 0$ .
- **Энергетическая ёмкость** всех энергетических батарей в инвентаре:  $0 \leq player_b < \infty$ . Энергетическая ёмкость батарей может быть использована для выпуска лазерных зарядов. При старте  $player_b = 0$ .

Платформа разрушается со временем: периодически некоторые из её клеток становятся непригодными для существования игрока.

Помимо игрока, на клетках платформы могут находиться следующие объекты:

- **Другой игрок:** попытка игрока выполнить движение так, что хотя бы одна из его клеток окажется в клетке с другим игроком, приведёт к пропуску хода. Про игрока известен его номер  $player_{id}$ , некоторая функция от его здоровья  $smooth(player_h)$  и количество его очков  $player_r$ .
- **Препятствие:** попытка игрока выполнить движение так, что хотя бы одна из его клеток окажется в клетке с препятствием, приведёт к пропуску хода. Препятствия статичны: они помещены на карту до начала первого хода, не исчезают, не появляются и не перемещаются.
- **Энергетический батончик:** если в результате движения хотя бы одна из клеток игрока находится в клетке с энергетическим батончиком, игрок подбирает его и кладёт в инвентарь. Различные энергетические батончики имеют различную энергетическую ценность

$10 \leq food_s \leq 30$ , которая не видна на карте, но станет известна в момент получения батончика в инвентарь.

- **Батарейка:** если в результате движения хотя бы одна из клеток игрока находится в клетке с батареей, игрок подбирает её и кладёт в инвентарь. Различные батарейки имеют различную энергетическую ёмкость  $2 \leq battery_p \leq 4$ , которая не видна на карте, но станет известна в момент получения батарейки в инвентарь.
- **Лазерный луч:** если в результате движения игрока или лазерного луча хотя бы одна из клеток игрока находится в клетке с лазерным лучом, значение  $player_h$  уменьшается на 30. Лазерный луч имеет длину 10 (то есть  $(min(x_i) - max(x_i))^2 + (min(y_i) - max(y_i))^2 \leq 10^2$ , где  $(x_i, y_i)$  – множество клеточек, занимаемых лучом в данный момент времени). Лазерный луч выпускается игроком и самостоятельно передвигается в заданном направлении, но не дальше 50 единиц расстояния от выпустившего его игрока. Лазерный луч перемещается на 10 единиц расстояния за ход (то есть лазерный луч не имеет общих клеточек с собой же на прошлом ходу, но имеет как минимум одну касающуюся клеточку). Направление движения лазерного луча известно. Лазерный луч проходит сквозь все остальные объекты, включая игроков и другие препятствия.
- **Разрушенная платформа:** если в результате движения игрока или разрушения платформы хотя бы одна из клеток игрока находится в клетке с разрушенной платформой, значение  $player_h$  уменьшается на 30.
- **Повреждённая платформа:** клетка такого типа превратится в разрушенную платформу в течение 2 ходов игры.

Функция  $smooth$ , описанная выше, обладает следующими свойствами:  $0 \leq smooth(x) \leq 5$ , если  $x < y$ , то  $smooth(x) < smooth(y)$ .

Каждый игровой ход значение  $player_s$  уменьшается на 1 у каждого игрока.

Игроки ходят в порядке, заданным кортежом  $(player_s, player_h, (player_{id} + round \bmod |player|))$  (т.е. сначала игроки с максимальным количеством жизненной силы, среди них – игроки с максимальным количеством очков здоровья, среди них – по очереди).

За каждый ход игрок может выполнить одно из следующих действий:

- **Ничего:** игрок, не делающий ничего, остаётся на месте.
- **Питание:** игрок может потребить  $1 \leq \Delta f \leq player_f$  единиц энергии. При этом  $player_f$  уменьшится на  $\Delta f$ , а  $player_s$  увеличится на  $\Delta f$ .
- **Движение:** игрок может передвинуться на одну клетку прямо, вправо или влево относительно направления своего взгляда.
- **Быстрое движение:** игрок может передвинуться на две клетки прямо, вправо или влево относительно направления своего взгляда. При этом дополнительно расходуется 2 единицы  $player_s$ . Игрок может двигаться только в одном направлении (например, переход на две клетки прямо за один ход разрешён, но переход на одну клетку прямо и на одну клетку вправо – нет).
- **Поворот:** игрок может повернуться на 90 градусов вправо или влево.
- **Разворот:** игрок может развернуться на 180 градусов. При этом дополнительно расходуется 2 единицы  $player_s$ .
- **Атака:** если игрок имеет хотя бы одну общую грань с другим игроком, игрок может выполнить атаку световым мечом. Для атакующего игрока,  $attacker_s$  уменьшится на 5, а для атакуемого игрока  $victim_h$  уменьшится на 30.
- **Выстрел:** если игрок имеет хотя бы одну единицу энергетической ёмкости, он может выпустить лазерный луч в полуплоскость, совпадающую с направлением его взгляда.  $player_b$  уменьшается на 1, а  $player_s$  уменьшается на 5. Выстрел можно выполнять не чаще, чем раз в 10 ходов (так как бластеру требуется время на перезарядку)
- **Комбинированное действие:** игрок может выполнить одну из следующих комбинаций: **Движение и Атака**, **Движение и Выстрел**, **Поворот и Атака** или **Поворот и Выстрел**. Другие комбинации не разрешены.

В конце хода игрока происходит изменение здоровья в зависимости от значения жизненной энергии:

$$\Delta player_h = \begin{cases} -10 & \text{если } player_s = 0 \\ -1 & \text{если } 0 < player_s < 50 \\ 0 & \text{если } 50 \leq player_s \leq 250 \\ +10 & \text{если } 250 < player_s \end{cases}$$

Если в результате обработки хода игрока количество очков жизни стало равно 0, игрок теряет всё содержимое своего инвентаря и возражается в случайной точке с начальными значениями  $player_h$  и  $player_s$ .

Пример формулы расчёта количества очков результата:  $player_r = 2000 \times K_a + 3000 \times K_s - 5000 \times D + \sum_{step} \frac{player_h \times player_s}{100 \times 100}$ , где  $K_a$  – количество противников, убитых световым мечом,  $K_s$  – количество противников, убитых из бластера,  $D$  – количество смертей,  $\sum_{step}$  – сумма значений по всем шагам игры.

## 2 Протокол Взаимодействия

Система проведения турнира использует протокол взаимодействия, сходный с интерактивными задачами на олимпиадах по программированию: последовательное взаимодействие через стандартный ввод-вывод.

Перед каждым ходом программа получает на стандартный ввод строку:

$player_{id} \ player_h \ player_s \ player_f \ player_b \ x \ y \ D \ E_n \ O_n$

Где  $player_*$  – это параметры игрока,  $x$  и  $y$  – координаты центральной клетки игрока ( $-10^9 \leq x, y \leq 10^9$ ),  $D$  – направление взгляда ("N": в сторону увеличения  $y$ , "E": в сторону увеличения  $x$ , "S": в сторону уменьшения  $y$ , "W": в сторону уменьшения  $x$ ).

Далее следуют  $E_n$  строк – описания событий, произошедших с последнего хода:

- `pickup_battery batteryp` (была поднята батарея).
- `pickup_food foods` (был поднят батончик).
- `attack playerid` (была атака мечом от игрока  $player_{id}$ ).

- `attack_ray playerid` (была атака лазером от игрока `playerid`).
- `attack_platform` (было потеряно здоровье из-за нахождения на разрушенной платформе).
- `recover` (было восстановлено здоровья из-за высокого количества энергии).
- `starve` (было потеряно здоровье из-за низкого количества энергии).
- `death` (была смерть; игрок был восстановлен в случайном месте поля).
- `failure` (последняя команда завершилась неудачей).
- `finish` (турнир окончен; стратегия должна завершить работу).

Далее следует  $O_n$  строк – описания объектов, видимых игроком. Каждая строка имеет один из следующих возможных форматов:

- `player x y D playerid smooth(playerh) playerr`
- `food N x1 y1 ... xN yN`
- `battery N x1 y1 ... xN yN`
- `ray playerid N x1 y1 ... xN yN`
- `obstacles N x1 y1 ... xN yN`
- `damaged_platform N x1 y1 ... xN yN`
- `destroyed_platform N x1 y1 ... xN yN`

Все однотипные объекты задаются вместе перечислением их координат в одной строке. Игрок видит только объекты, находящиеся на расстоянии не дальше 150 единиц и в полуплоскости направления его взгляда.

В ответ решение должно вывести единственную строку, описание действия:

- `idle`

- `move x y` (где  $x, y$  – новые координаты центра игрока).
- `turn D` (где  $D$  – новое направление взгляда игрока "N" "E" "W" или "S").
- `attack player_id` (где  $player_id$  – ID атакуемого игрока).
- `shoot x y` (лазерный луч выйдет из центральной точки игрока в направлении точки  $(x, y)$ ).
- `eat  $\Delta f$`  (где  $\Delta f$  – количество потребляемой еды).
- `both  $A_1$  and  $A_2$`  (где  $A_1, A_2$  – описание разрешённых к комбинированию действий; например, комбинированное действие «вернуться на север и атаковать игрока 4» записывается как `both turn N and attack 4`).

Обратите внимание, что Движение и Быстрое движение (а так же поворот и разворот) кодируются одним и тем же ключевым словом: система проведения турнира решит, какое действие нужно выполнить, в зависимости от разницы координат или направлений взгляда.

Строка обязана завершаться символом перевода строки, буфер вывода должен быть очищен после каждого вывода (с помощью `cout.flush()`).