**Война ботов.**

1. **О соревновании**
   1. **Обзор**

В войне ботов у Вас есть уникальная возможность применить свои навыки программирования против других участников. В процессе соревнований в игровом мире сражаются боты-машины. Каждый участник пишет своего бота – игровую стратегию. Затем Ваша машина будет помещена в игровой мир вместе с машинами других игроков и будет бороться с ними за победу.

Соревнование будет состоять из Nтуров. В каждом туре принимают участие Mботов. В начале каждая машина неподвижна, имеет количество жизни равное (K), количество нитро равное (L).

Машины двигаются в двумерном прямоугольном мире. Они могут собирать бонусы. За каждый бонус, соответственно его типу игрок получает:

* за аптечку P1 очков
* за нитро P2 очков
* за флаг P3 очков

Также очки начисляются за столкновение с машинами противников. В игровом мире одновременно присутствует случайное число разных бонусов.

Машина может получить урон и потерять жизнь, сталкиваясь с другими машинами, препятствиями или границами мира.

Цель игры набрать как можно большее количество очков за матч.

Есть 3 способа набрать очки.

1. Собирание бонусов.
2. Соударения с противниками(количество очков зависит от урона, причиненного сопернику).
3. Также игрок получает дополнительное количество очков в конце матча, равное количеству оставшейся жизни (0, если у игрока ее не осталось).

Для написания стратегии используется язык Java. Каждый участник получает

//TODO что каждый получит на руки.

Далее более подробно описаны правила соревнования и приведен пример бота. Это простейший бот - пример, предназначенный для того, чтобы продемонстрировать его создание и использование. Далее также приведена документация, которая описывает класс и его поля, которые используются для создания стратегии.

* 1. **Правила.**
     1. Правила проведения турнира

// тут про то, как будет проводиться турнир

* + 1. Игровые правила

Игра проходит следующим образом. В начале игры (инициализация) у каждого игрока есть возможность выбрать имя своего бота (оно будет отображаться в статистике и является необходимым атрибутом). Затем начинается игра и боты игроков осуществляют свои действия. В игре время дискретно и один тик времени означает один ход бота игрока, в течение которого он должен выбрать действия на текущее состояние мира. Таким образом, во время инициализации игровое время равно 0. В дальнейшем время увеличивается на 1 с каждым тиком. Игра длится Xтиков.

Игровой мир это прямоугольник размером R , находящийся на двумерной плоскости.

X

Y

π/4

0

(1024,640)

(0,0)

Размер мира не зависит от разрешения экрана и наоборот.

При старте игры машины появляются в игровом мире в некотором случайном месте. Каждый игрок управляет одной машиной. При этом у каждой машины имеется жизнь в количестве nочков. Машина считается живой, если количество жизни больше нуля. Когда количество жизни достигает нуля, машина считается неактивно. Управление неактивной машиной не выполняется, т.е. код, написанный в ее стратегии не выполняется и машина не едет (но может при этом двигаться по инерции). Однако, если машина каким-либо образом сумеет подобрать бонус с жизнью то машина со следующего тика считается живой и ее стратегия продолжает выполняться.

Всего в игре 4 типа объектов: машина, бонус, препятствие. Каждый объект это круг с определенным радиусом .Радиусы объектов приведены в таблице.

|  |  |  |
| --- | --- | --- |
| Название класса | Объект | Радиус |
| Vehicle | Машина |  |
| Bonus | Бонус |  |
| Obstacle | Препятствие |  |

Препятствия в игре подвижны и вначале игры генерируются случайным образом.

Столкновение машины с игровым объектом происходит, если расстояние между центрами машин меньше или равно сумме их радиусов.

В таблице приведено описание взаимодействий, которые происходят при столкновении машины с другими игровыми объектами.

|  |  |
| --- | --- |
| Объект | Взаимодействия |
| Машина | Когда машины сталкиваются, обе теряют жизнь, но приобретают очки. Их скорости и направления движения изменяются. |
| Бонус | Машина подбирает бонус, бонус исчезает с игрового поля. Потеря жизни не происходит. |
| Препятствие | Машина теряет жизнь, ее скорость и направление движения меняются |

Машина может столкнуться с границами игрового мира. В этом случае она также теряет жизнь и ее скорость и направление изменяются. Машина не может покинуть игровой мир.

В игре присутствует 2 типа бонусов: аптечка и нитро. Аптечка добавляет очки жизни машине, а нитро увеличивает скорость движения. Размеры изменений приведены в таблице.

|  |  |  |
| --- | --- | --- |
| Тип бонуса | Очки | Особенность |
| Аптечка | 10 | Увеличивает жизнь машины на 20 |
| Нитро | 15 | Увеличивает максимальное ускорение машины до 2 |
| Флаг | 100 | - |

Все удары в игровом мире абсолютно упруги. Нет никакого ограничения на скорость машины, но существуют ограничения на ускорение(n) и поворот (m). Также в игре присутствует сила трения.

1. **Реализация бота**
   1. **Как написать своего бота.**

Для того чтобы написать бота Вы должны реализовать класс Bot. Класс содержит 2 метода, которые и нужно реализовать. Конечно, Вы можете добавить свои необходимые поля и классы к этому классу.

Когда машина помещается в игровой мир, вызывается метод init() из Вашего класса Bot. Этот метод возвращает имя Вашего бота и для того, чтобы задать придуманное Вами имя перепишите директиву return (к примеру return "MyBotName";). Имя должно содержать от 3 до 50 символов.

Код в методе move() определяет поведение вашей машины. Этот метод будет вызываться каждый тик игры. При этом в этот метод передаются 3 параметра:

World world – этот параметр содержит всю информацию о состоянии игрового мира. Например, расположение бонусов, размеры мира, текущий тик и тд. (см документацию)

Vehicle self – с помощью этого параметра можно получать информацию о состоянии самой машины игрока, например о количестве оставшейся жизни и положении в пространстве. (см документацию)

BotAction action – посредством этого класса игрок управляет действиями своей машины в мире. Существуют 2 действия , которые определяет игрок. Это изменение скорости вдоль направления машины (определяемого углом angle) и поворот на угол. В классе BotAction существуют 2 поля: power и turn. Для того, чтобы повлиять на скорость машины нужно соответствующее значение напрямую присвоить полю power. Соответственно, для поворота на угол надо присвоить значение полю turn.

Для каждого игрока дается только Nсекунд для выполнения метода move() его бота. Если время выполнения кода превышает это ограничение, бот перестает участвовать в текущей игре. Если в коде бота возникают ошибки времени исполнения то бот также прекращает участие.

* 1. **Документация**