

Убийственные круги

Сегодня вам предстоит погрузиться в мир кругов. Но нет, это будет не унылая оптимизационная задача на геометрию, а весёлая и задорная игра. Теперь вместо скучных алгоритмов и занудных оптимизаций будут жизнерадостные круги.

Абсолютно всё вокруг будет кругами. Начнём с игрового поля: оно представляет собой круг с центром в начале координат и заданным начальным радиусом. На поле располагаются объекты трёх типов: мусор, роботы и пули. Объекты всех типов, как несложно догадаться, круги. Все объекты одного типа имеют одинаковый радиус. Но радиус разных типов объектов — разный.

Изначально на игровом поле находится много мусора: это статичные нейтральные объекты. Мусор равномерно сгенерирован по всему полю, он не перемещается, и в ходе игры новый не появляется. У мусора есть начальное количество здоровья (одинаковое для всех объектов). За нанесение урона (с помощью стрельбы или тарана) и уничтожение мусора игрок получает очки, которые необходимы для улучшения робота, а и которым после завершения игры ранжируются команды.

Каждая команда управляет ровно одним роботом. Робот обладает рядом характеристик, часть из которых можно будет улучшать. У роботов также есть начальное количество здоровья (одинаковое для всех роботов). Вам требуется следить, чтобы здоровье вашего робота не опустилось до нуля: в этом случае ваш робот будет безвозвратно уничтожен.

Игрок может отдавать этому роботу команды на перемещение, стрельбу и улучшение характеристик.

При отдаче команды стрельбы из робота вылетает пуля. При столкновении с любым объектом, имеющим здоровье, пуля наносит этому объекту урон и исчезает с поля. Нанесение урона заключается в уменьшении количества здоровья объекта на некоторую заданную константу. Объект, у которого после очередного нанесения урона количество здоровья перестало быть положительным, с поля исчезает. Если этот объект — робот, то игрок, управлявший им, выходит из игры.

Роботы имеют характеристики, которые в начале игры для всех игроков одинаковы:

- максимальное количество здоровья;
- максимальная скорость перемещения;
- скорость стрельбы (минимальное количество итераций, которые необходимо подождать с момента последнего выстрела, чтобы произвести новый выстрел);
- урон от пуль игрока;
- радиус стрельбы.

В игре присутствует система прокачки указанных характеристик: по мере получения очков появляется возможность улучшать их значения. Каждая из этих характеристик может быть улучшена максимум 5 раз.

С ходом игры игровое поле сжимается: с каждым игровым тактом радиус поля уменьшается на заданную константу. Робот может покидать границы игрового поля. Однако, роботы и мусор, у которых хотя бы часть круга находится за границами поля, получают заданное значение урона после каждой итерации.

1 Моделирование игры

Моделирование игры состоит из заранее заданного количества игровых тактов. Каждый такт состоит из этапов, на котором по очереди выполняются действия каждого из игроков.

В начале такта стратегия каждого робота определяет куда он будет перемещаться, стрелять и будет ли выполнено улучшение. В том числе, робот может делать все три действия на одном такте.

Первым этапом выполняются улучшения. Робот может сказать какую из характеристик хочет улучшить. Чтобы получить возможность выбрать первое улучшение, необходимо набрать 4 единицы опыта. Каждое следующее улучшение требует, чтоб суммарно набранный опыт был в два раза больше: т.о. следующее обновлением будет доступно ещё после набора 4 единиц опыта, следующее после ещё 8 и т.д. Если робот отдаёт команду на улучшение, но при этом её невозможно выполнить (недостаточно опыта или указанная характеристика уже была улучшена 5 раз), то такая команда игнорируется.

Следующим этапом выполняется стрельба. Для всех роботов, которые указали действие стрельбы (задали направление) и которые могут его совершить (прошло необходимое количество итераций с момента последнего выстрела), на поле появляются пули с координатами в центре робота. На текущем такте игрок может отказаться стрелять, даже если с момента предыдущего выстрела прошло достаточно времени, чтобы совершить выстрел. Далее данные пули будут лететь в строго указанном направлении.

Следующий этап — перемещение игроков. Игрок может выбрать произвольный вектор скорости, по модулю не превосходящий своей максимальной скорости перемещения (в противном случае перемещение будет ограничено скоростью). Перемещение происходит дискретно в несколько шагов. На каждом шаге робот сдвигается в указанном направлении, но на расстояние, в несколько раз меньше указанного. Если после очередного шага робот игрока столкнётся с мусором или другим роботом, то движение будет остановлено, робот переместится в свое положение на предыдущем шаге, и все объекты получают урон от столкновения. Если же ни на одном шаге столкновения не произошло, то в итоге робот переместится на указанное расстояние (но не дальше, чем максимальная скорость игрока).

Следующий этап — перемещение пуль. Движение пуль так же происходит дискретно, как это описано для игроков. Если на одном из шагов при движении пуля сталкивается с игроком или мусором, то эти объекты получают урон. Тот игрок, чья пуля нанесла урон, получает очки. Пуля исчезает с поля при выполнении любого из следующих условий: либо когда она столкнётся с некоторым другим объектом, либо когда пройдёт определённое количество тактов после её появления.

Следующий этап — сжатие игрового поля. По ходу игры радиус игрового поля равномерно сокращается. Если после очередного сокращения хотя бы часть робота или мусора оказался за кругом, то этот объект получает фиксированный урон.

Все оставшиеся в конце игрового такта роботы получают восполнение здоровья на фиксированную величину, но не более чем до своего максимального запаса здоровья.

Любое нанесение урона (когда пуля игрока попадает в другой объект; когда игрок сталкивается с другим объектом при движении, в том числе, вы нанесёте урон и получите опыт, когда другой робот столкнулся с вашим роботом — в этом случае объекты нанесут друг другу одинаковый урон) даёт определённое количество опыта. Дополнительные очки даются за нанесение урона, приведшего к исчезновению объекта. Но опыт будет получен только в том случае, если робот ещё жив. Например, если вы сделали выстрел, затем ваш робот умер, а уже после этого пуля нанесла урон, то такой опыт получен уже не будет.

Точные числовые константы всех обозначенных параметров можно найти в файле `params.ini`, который лежит в пакете с примером решения.

2 Среда исполнения

В файле `api.h` описаны доступные для вашей программы функции. Когда-нибудь у жюри будут ресурсы и возможности готовить задачи не в последний момент, а дать в условии понятное и исчерпывающее описание доступных для вас функций. Но, увы, не в этот раз. Сегодня вам предстоит разобраться в доступном интерфейсе лишь по описанию типов и сигнатур функций в указанном заголовочном файле. Впрочем, к материалам приложен пример простейшего решения, который должен значительно облегчить знакомство с игрой.

3 Ограничения

Ограничение по памяти составляет 64 мегабайт. При превышении этого ограничения ваша программа будет отключена (робот при этом останется на поле, но не сможет двигаться, стрелять и прокачивать характеристики).

Существует мягкое ограничение на суммарное процессорное время работы вашей программы за все прошедшие такты: это 1000 миллисекунд, плюс количество прошедших тактов, умноженное на 20 миллисекунд. При превышении мягкого ограничения ваша программа будет пропускать дальнейшие такты, пока квота не вырастет достаточно, чтобы покрыть использованное время.

Дополнительно, каждый вызов функции `doTurn` должен завершиться за 5 секунд. Если программа не завершается в течение этого времени, то она будет принудительно отключена (робот при этом станет неактивен, как и при превышении программой лимита по памяти).

В случае, если ваша программа завершится до окончания игры, то робот останется на поле (и даже сможет набирать опыт), однако, никаких действий он совершать не будет.

Вашей программе запрещается выполнять любые вычисления между вызовами `doTurn`.

4 Отправка решений и система оценки

Очки робота за игру равны количеству набранного опыта.

На туре будет включен бесконечный прогон игр с текущими решениями (все против всех на одном поле), но в финальный рейтинг результаты таких игр не попадают.

После тура жюри запускает определённое (предположительно, 100) количество симуляций (опять же всех против всех). В итоге команды ранжируются по убыванию среднего количества очков по всем играм, смоделированным после завершения тура.

5 Материалы

Материалы состоят из двух архивов: пример решения и визуализатор.

«««< .mine Пакет с проектом для Visual Sudio содержит базовый пример стратегии управления роботом (проект «Solution») и встроенный симулятор игры (проект «GameRunner») для локального тестирования (в точности такой же, какой и используется при тестировании). ||||| .l11823 Пакет со проектом для Visual Sudio содержит базовый пример стратегии управления роботом (проект «Solution») и встроенный симулятор игры (проект «GameRunner») для локального тестирования (в точности такой же, какой и используется при тестировании). ===== Пакет со проектом для Visual Studio содержит базовый пример стратегии управления роботом (проект «Solution») и встроенный симулятор

игры (проект «GameRunner») для локального тестирования (в точности такой же, какой и используется при тестировании). »»»> .r11826

После открытия проекта он должен быть сразу в состоянии, готовом для запуска эмулятора с одним-единственным роботом, которым управляет ваша стратегия. Если проект не открылся должным образом и вы не можете сразу запустить игру, то обратитесь с вопросом к жюри.

6 Запуск решений

Локально решение запускается посредством запуска приложения GameRunner. В GameRunner из функции main можно позвать одну из функций:

- `runSingleSolution` — запустить симуляцию с одним роботом, управляемым вашим решением;
- `runSolutionWithBots` — запустить симуляцию с роботом, управляемым вашим решением, и ботами;
- `runSeveralSolutions` — запустить симуляцию с несколькими роботами, управляемыми вашим решением;
- `runDebugServer` — запустить сервер для подключения к нему визуализатора.

«««< .mine В коде данных функций вы можете править набор роботов, с которым запустится решение. Также вы можете вносить правки в параметры игры, изменяя файл `param.ini`. ||||| .r11823 В коде данных функций вы можете править набор роботов, с которым запустится. Так же вы можете вносить правки в параметры игры, изменяя файл `param.ini`. ===== В коде данных функций вы можете править набор роботов, с которым следует запуститься. Так же вы можете вносить правки в параметры игры, изменяя файл `param.ini`. »»»> .r11826

7 Визуализатор

В выданных вам материалах имеется божественный визуализатор. Он умеет работать в двух режимах: воспроизведение лога игры (удобно для просмотра ранее смоделированных решений или решений с сервера) и подключение к локальному серверу игры (удобно при отладке своего решения).

Для воспроизведения лога требуется запустить исполняемый файл визуализатора, нажав на кнопку выбора файла в левом нижнем углу и наслаждаться воспроизведением игры.

Для выбора второго режима запустите игру через функцию `runDebugServer`, которая предварительно запустит локальный сервер. Затем запустите визуализатор и нажмите на кнопку “Connect”. После подключения используйте элементы графического интерфейса визуализатора, чтобы проиграть визуализацию игры: вы можете ставить процесс моделирования на паузу, проигрывать по одной итерации и смотреть состояние поля на предыдущих итерациях.

«««< .mine Визуализатор поддерживает масштабирование и перемещение по полю с помощью мышки. Вы можете кликнуть на панели слева на робот, чтобы автоматически центрировать и отмасштабировать на нём камеру, а так же начать слежение за ним. ||||| .r11823 Визуализатор поддерживает масштабирование и перемещение по полю с помощью мышки. Так же вы можете кликнуть на панели слева на работ чтобы автоматически центрировать и отмасштбировать на нём камеру, а так же начать слежение за ним. ===== Визуализатор поддерживает масштабирование и перемещение по полю с помощью мышки. Так же вы

можете кликнуть на панели слева на робот чтобы автоматически центрировать и отмасштабировать на нём камеру, а так же начать слежение за ним. »»»> .r11826

В интерфейсе игры есть набор функций, начинающихся со слова “draw”. Данные функции позволяют нарисовать вспомогательные элементы поверх игрового поля. Вызовы не будут ничего делать при моделировании игры на сервере, однако, при локальном запуске помогают вывести дополнительную отладочную информацию.

Если при запуске визуализатора операционная система выдаст предупреждение о недоверенном источнике программы, нажмите «подробнее» и затем выберите «разрешить».

8 Просмотр игр из системы тестирования

Скачать истории игр из системы тестирования можно в браузере из директории:

<https://olympic.nsu.ru/nsuts-new/ws02024/games/>