



CODESIDE 2019

ПРАВИЛА

Версия 0.1.1

Декабрь 2019

Оглавление

1	О мире CodeSide 2019	3
1.1	Общие положения игры и правила проведения турнира	3
1.2	Описание игрового мира	5
1.3	Уровень	5
1.4	Движение	6
1.5	Лут	6
1.6	Стрельба	7
1.7	Мины	8
1.8	Управление	8
2	Описание API	9
2.1	MyStrategy	9
2.2	Описание объектов	9
2.2.1	Bullet	10
2.2.2	BulletParams	10
2.2.3	Color	10
2.2.4	ColoredVertex	10
2.2.5	CustomData	11
2.2.6	ExplosionParams	11
2.2.7	Game	11
2.2.8	Item	12
2.2.9	JumpState	12
2.2.10	Level	12
2.2.11	LootBox	12
2.2.12	Mine	13
2.2.13	MineState	13

2.2.14	Player	13
2.2.15	Properties	13
2.2.16	TextAlignment	14
2.2.17	Tile	14
2.2.18	Unit	14
2.2.19	UnitAction	15
2.2.20	Vec2	15
2.2.21	Weapon	15
2.2.22	WeaponParams	16
2.2.23	WeaponType	16

Глава 1

О мире CodeSide 2019

1.1 Общие положения игры и правила проведения турнира

Данное соревнование предоставляет вам возможность проверить свои навыки программирования, создав искусственный интеллект (стратегию), управляющий командой юнитов в специальном игровом мире (подробнее об особенностях мира CodeSide 2019 можно узнать в следующих пунктах этой главы).

В каждой игре вы будете противостоять стратегии другого участника. Задача вашей команды — набирать очки, уничтожая и нанося урон противнику. Команда, набравшая большее количество очков побеждает. Игры также могут закончиться ничьей, если обе команды набрали одинаковое количество очков в конце.

Время в игре дискретное и поделено на “тики”. А начале каждого тика, игровой симулятор отправляет данные о состоянии мира стратегиям участников, получает действия от них и обновляет состояние мира в соответствии с полученными действиями. Затем считает изменения в игровом мире и объектах в нем, после чего процесс повторяется с обновленными данными. Максимальная длительность любой игры равна 10000 тиков, но игра может закончиться раньше если все стратегии “упали”.

“Упавшая” стратегия больше не может управлять своими юнитами. Стратегия считается “упавшей” в следующих случаях:

- Процесс стратегии непредвиденно завершился, или произошла ошибка в протоколе взаимодействия стратегии с игровым сервером.
- Стратегия превысила один (или несколько) из ограничений на время. Стратегии на один тик дается не больше 1 секунды реального времени. Суммарно на всю игры стратегии выделено

$$20 \times \langle \text{длительность_игры_в_тиках} \rangle + 20000 \quad (1.1)$$

миллисекунд реального времени. Формула учитывает максимальную длительность игры. Ограничение времени остается тем же даже если настоящая длительность игры отличается от этого значения. Все ограничения по времени относятся не только к коду участника, но и к взаимодействию клиента-оболочки с игровым симулятором.

- Стратегия превысила ограничение памяти. В любой момент процесс стратегии не должен использовать более 256 MB оперативной памяти.

Турнир проводится в несколько этапов, которым предшествует квалификация в Песочнице. Песочница — соревнование, которое проходит на протяжении всего чемпионата. В рамках каждого этапа игроку соответствует некоторое значение рейтинга — показателя того, насколько успешно его стратегия участвует в играх.

Начальное значение рейтинга в Песочнице равно 1200. По итогам игры это значение может как увеличиться, так и уменьшиться. При этом победа над слабым (с низким рейтингом) противником даёт небольшой прирост, также и поражение от сильного соперника незначительно уменьшает ваш рейтинг. Со временем рейтинг в Песочнице становится всё более инертным, что позволяет уменьшить влияние случайных длинных серий побед или поражений на место участника, однако вместе с тем и затрудняет изменение его положения при существенном улучшении стратегии. Для отмены данного эффекта участник может сбросить изменчивость рейтинга до начального состояния при отправке новой стратегии, включив соответствующую опцию. В случае принятия новой стратегии системой рейтинг участника сильно упадёт после следующей игры в Песочнице, однако по мере дальнейшего участия в играх быстро восстановится и даже станет выше, если ваша стратегия действительно стала эффективнее. Не рекомендуется использовать данную опцию при незначительных, инкрементальных улучшениях вашей стратегии, а также в случаях, когда новая стратегия недостаточно протестирована и эффект от изменений в ней достоверно не известен.

Начальное значение рейтинга на каждом основном этапе турнира равно 0. За каждую игру участник получает определённое количество единиц рейтинга в зависимости от занятого места (система, аналогичная используемой в чемпионате “Формула-1”). Если два или более участников делят какое-то место, то суммарное количество единиц рейтинга за это место и за следующие **количество_таких_участников** — 1 мест делится поровну между этими участниками. Например, если два участника делят первое место, то каждый из них получит половину суммы единиц рейтинга за первое и второе места. При делении округление всегда совершается в меньшую сторону. Более подробная информация об этапах турнира будет предоставлена в анонсах на сайте проекта.

Сначала все участники могут участвовать только в играх, проходящих в Песочнице. Игроки могут отправлять в Песочницу свои стратегии, и последняя принятая из них берётся системой для участия в квалификационных играх. Каждый игрок участвует примерно в одной квалификационной игре за час. Жюри оставляет за собой право изменить этот интервал, исходя из пропускной способности тестирующей системы, однако для большинства участников он остаётся постоянной величиной. Существует ряд критериев, по которым интервал участия в квалификационных играх может быть увеличен для конкретного игрока. За каждую N-ю полную неделю, прошедшую с момента отправки игроком последней стратегии, интервал участия для этого игрока увеличивается на N базовых интервалов тестирования. Учитываются только принятые системой стратегии. За каждое “падение” стратегии в 10 последних играх в Песочнице начисляется дополнительный штраф, равный 20% от базового интервала тестирования. Интервал участия игрока в Песочнице не может стать больше суток.

Игры в Песочнице проходят по набору правил, соответствующему правилам случайного прошедшего этапа турнира или же правилам следующего (текущего) этапа. При этом чем ближе значение рейтинга двух игроков в рамках Песочницы, тем больше вероятность того, что они окажутся в одной игре. Песочница стартует до начала первого этапа турнира и завершается через некоторое время после финального (смотрите расписание этапов для уточнения подробностей). Помимо этого Песочница замораживается на время проведения этапов турнира. По итогам игр в Песочнице происходит отбор для участия в Раунде 1, в который попадут 1080 участников с наибольшим рейтингом на момент начала этого этапа турнира (при равенстве рейтинга приоритет отдаётся игроку, раньше отправившему последнюю версию своей стратегии), а также дополнительный набор в следующие этапы турнира, включая Финал.

Этапы турнира

- В **Раунде 1** вам предстоит изучить правила игры и освоить управление юнитами. Вам даётся 1 юнит, как и вашему противнику. Уровень в первом раунде достаточно простой, что не нужен продвинутый алгоритм поиска пути. Задача — нанести как можно больше урона противнику! Все просто. Раунд 1, как и последующие этапы, состоит из двух частей, между которыми будет небольшая пауза (с возобновлением работы Песочницы), позволяющая улучшить стратегию. Последняя отосланная стратегия перед началом каждой части выбирается для игр в соответствующей части. Игры проходят волнами. В каждой волне, каждый участник играет ровно одну игру. Количество волн в каждой части определено способностями тестирующей системы, но гарантируется, что их будет не меньше десяти. 300 участников с наибольшим рейтингом проходят в Раунд 2. Также в Раунд 2 проходят дополнительные 60 участников с наибольшим рейтингом в Песочнице (на момент старта Раунда 2), среди тех кто не прошел по результатам Раунда 1.
- В **Раунде 2** вам предстоит освоить командную работу. Теперь в вашей команде уже 2 юнита. Задача осложняется тем, что после Раунда 1, часть слабых участников будет отсеяна и вам придется сражаться с более сильными соперниками. По результатам Раунда 2 50 лучших стратегий проходят в Финал. Также в Финал проходят дополнительные 10 участников с наибольшим рейтингом в Песочнице (на момент старта Финала), среди тех кто не прошел основной отбор.
- **Финал** — самый важный этап. После первых двух раундов остаются только сильнейшие. Также, игровой уровень теперь более сложный, и требует хорошего алгоритма поиска пути. Система проведения Финала имеет свои особенности. Этап будет также разделен на две части, но игры будут проходить не волнами. В каждой части, игры будут проходить между каждой парой участников Финала. Если время позволит, операция будет повторена.

Все финалисты сортируются по неубыванию рейтинга после окончания Финала. Если рейтинг одинаковый, более высокое место получает тот, чья стратегия была отправлена раньше. Призы по итогам Финала раздаются в соответствии с такой сортировкой.

После окончания Песочницы, все участники кроме победителей Финала, сортируются по неубыванию рейтинга. Если рейтинги одинаковые, более высокое место получает тот, чья последняя версия стратегии была отправлена раньше. Призы по итогам Песочницы раздаются в соответствии с такой сортировкой.

1.2 Описание игрового мира

Игровой мир двумерный. Ось X направлена горизонтально, ось Y вертикально, направлена вверх. Уровень состоит из тайлов, которые отличаются механикой коллизий. Все сущности (юниты, пули, взрывы, лут, тайлы уровня) прямоугольные, со сторонами параллельными осям X и Y .

В начале каждой игры, юниты располагаются на заданных позициях и не имеют оружия. Также на уровне появляется лут — оружие, аптечки и мины.

Игра обновляется несколько раз за тик. Каждый апдейт продвигает игровое время на $\frac{1}{\text{кол} - \text{вотиковвсекунду} * \text{кол} - \text{воапдейтоввтик}}$.

1.3 Уровень

Уровень представляет собой сетку из тайлов. Каждый тайл может быть одним из:

- Пусто.
- Стена — блокирует передвижение и пули.
- Платформа — на платформе можно стоять, но можно и двигаться сквозь нее.
- Лестница — на лестнице можно свободно передвигаться вертикально. Считается, что юнит находится на лестнице, если отрезок от центра юнита к середине верхней границе юнита пересекается с тайлом.
- Батут — дает высокий прыжок. Активируется если юнит пересекается с тайлом. Прыжок с батута нельзя отменить.

1.4 Движение

Юниты не имеют ускорения, горизонтальное / вертикальное движения независимы. Каждый апдейт, сперва происходит горизонтальное движение, затем вертикальное.

Скорость горизонтального движения задается действием вашей стратегии. Во время апдейта игры, позиция юнита меняется в соответствии с заданной скоростью, до тех пор пока движение не заблокировано тайлом или другим юнитом.

Вертикальное движение немного сложнее. Юнит может находиться в состоянии падения или состоянии прыжка. Прыжок может быть отменен в любой момент, за исключением прыжков с батута.

Если юнит находится в состоянии падения, он движется вниз с постоянной скоростью. Если падение заблокировано тайлом уровня или другим юнитом, падение прекращается. Какие тайлы блокируют падение зависит от того, выставлено ли “jump_down” в стратегии. Если да, только стены блокируют падение. Если нет, платформы и лестницы также блокируют. Если падение заблокировано чем либо, юнит меняет состояние на состояние прыжка. Состояние прыжка может быть мгновенно отменено, если в стратегии не выставлено поле “jump”.

Если юнит находится в состоянии прыжка, он движется вверх с постоянной скоростью. Состояния прыжка ограничено временем. После данного времени, состояние меняется на падение. Если прыжок заблокирован тайлом стены или другим юнитом, состояния также меняется на падение. Прыжок может быть отменен в любой момент, если поле “jump” не выставлено.

Время и скорость прыжка зависят от того, что спровоцировало прыжок. Прыжок может начаться вручную после падения на землю или батутом. В случае прыжка с батута, прыжок нельзя отменить. Это значит, что состояния прыжка поменяется на падение только в случае столкновения с потолком или истечением времени прыжка.

1.5 Лут

В начале игры, лут случайным образом появляется на уровне. Лут может содержать:

- Аптечка. Подбирается только в случае неполного здоровья юнита. Мгновенно увеличивает здоровье.
- Оружие. Подбирается автоматически если у юнита нет оружия. Также возможно поменять текущее оружие юнита на новое (старое останется лежать на земле).

- Мина. Юниты хранят мины в инвентаре, они могут быть установлены и активированы позже.

Взаимодействие с лутом происходит в случае пересечения с юнитом.

1.6 Стрельба

Если юнит имеет оружие, он может целиться и стрелять в противника. Каждое оружие обладает следующими параметрами:

- Размер обоймы. Патроны в игре бесконечны, однако оружие должно быть перезаряжено после использования обоймы.
- Скорострельность. Время между выстрелами.
- Скорость перезарядки.
- Минимальный и максимальный разброс. Оружие имеет также текущее значение разброса, находящееся между минимальным и максимальным. Угол выстрела изменяется случайно в пределах текущего значения разброса.
- Отдача. Текущее значение разброса увеличивается на величину отдачи после каждого выстрела.
- Скорость прицеливания. Текущее значение разброса уменьшается постоянно с данной скоростью.
- Параметры пули — скорость, урон и размер пули.
- Параметры взрыва — может отсутствовать. Если присутствует, пули взрываются, нанося урон от взрыва юнитам в заданном радиусе.

Пули являются квадратами. Они попадают в другие юниты при пересечении. Пули так же могут попасть в стены. В любом случае, пуля мгновенно исчезает, нанося урон в случае попадания в юнита. Также, если присутствуют параметры взрыва, создается взрыв, наносящий дополнительный урон всем юнитам пересекающимся с квадратом с центром в позиции пули и заданным радиусом (то есть, размер квадрата взрыва равен двум радиусам).

Эти параметры являются константами для каждого типа оружия. Также есть следующие изменяющиеся параметры оружия:

- Обойма — количество патронов оставшихся в текущей обойме.
- Разброс — текущее значение разброса. Изначально равно минимальному. Угол выстрела изменяется случайным образом в отрезке $[-\text{разброс}, \text{разброс}]$.
- Таймер выстрела — время до следующего выстрела. Может отсутствовать если выстрел уже возможен.
- Последний угол — последний угол прицеливания. Когда вы меняете направление прицеливания, текущее значение разброса увеличивается на разницу углов. Во время первого тика после подбора оружия, это значение отсутствует.

Перезарядка оружия происходит автоматически. Первый раз — при подборе оружия. Затем, в случаях если текущая обойма пустая, а так же если стратегия указывает юниту прекратить огонь при неполной обойме.

1.7 Мины

Мины могут быть установлены только если юнит находится на земле. После установки, некоторое время необходимо для подготовки мины. В это время мину не получится взорвать. После подготовки мина переходит в режим ожидания, и находится в нем до тех пор, пока юнит не окажется в радиусе обнаружения. После обнаружения юнита, через некоторое время (время запуска), мина взорвется, нанеся урон всем юнитам в заданном радиусе взрыва.

1.8 Управление

Управление юнитом происходит с помощью правильной установки действия (*Action*), возвращаемого вашей стратегией.

Поле *velocity* устанавливает горизонтальную скорость юнита. Значение обрезается до максимального значения скорости юнита при превышении.

Если поле *jump* установлено в *true*, юнит начнет или продолжит прыжок. Если оно установлено в *false*, и текущий прыжок можно отменить, юнит прекращает прыжок.

Если поле *jump_down* установлено в *true*, юнит начинается двигаться вниз по лестницам и спрыгивать с / падать сквозь платформы.

Поле *aim* контролирует направление прицеливания. Если юнит не имеет оружия, значение не используется. Установите поле *shoot* в *true* чтобы открыть огонь.

Если поле *swap_weapon* установлено в *true*, юнит поменяет текущее оружие на то что находится рядом, если есть.

Наконец, *plant_mine* контролирует установку мин. Инвентарь игрока должен содержать хотя бы одну мину.

Глава 2

Описание API

2.1 MyStrategy

В пакете для вашего языка программирования вы можете найти файл `MyStrategy/my_strategy.<ext>`. Этот файл содержит класс `MyStrategy` с методом `get_action`, где должна быть реализована логика вашей стратегии.

Этот метод будет вызываться каждый тик, отдельно для каждого из ваших юнитов.

Метод принимает следующие аргументы:

- Текущий юнит для которого надо вычислить действие (объект *Unit*)
- Текущее состояние игрового мира (объект *Game*)
- Вспомогательный отладочный объект (объект *Debug*). Этот объект позволяет вам отрисовывать отладочную информацию прямо из кода вашей стратегии. Заметьте, что использование этого объекта не имеет эффекта при тестировании на сервере, поэтому использовать его стоит только для локальной отладки.

Метод должен вернуть объект *UnitAction*, определяющий действие для заданного юнита.

2.2 Описание объектов

В этой секции, некоторые поля могут быть опциональными (обозначается как *Optional<type>*). Способ реализации зависит от языка. При возможности используется специальный опциональный (*nullable*) тип, иначе другие методы могут быть использованы (например *nullable* указатели).

Некоторые объекты могут принимать несколько различных форм. Способ реализации зависит от языка. Если возможно, используется специальный (алгебраический) тип данных, иначе другие методы могут быть использованы (например варианты представлены классами унаследованными от абстрактного базового класса).

float32 — 32-битный вещественный тип данных, может называться *float*, и *float64* может называться *double* в некоторых языках.

2.2.1 Bullet

Определяет пулю. Поля:

- $weapon_type : WeaponType$ — тип оружия из которого была выстрелена пуля
- $unit_id : int$ — id выстрелившего юнита
- $player_id : int$ — id игрока выстрелившего юнита
- $position : Vec2\langle float64 \rangle$ — позиция пули (центр)
- $velocity : Vec2\langle float64 \rangle$ — скорость пули (в единицах в секунду)
- $damage : int$ — урон наносимый юниту при попадании
- $size : float64$ — размер пули (сторона квадрата определяющего пулю)
- $explosion_params : Optional\langle ExplosionParams \rangle$ — параметры взрыва, если применимо

2.2.2 BulletParams

Параметры определяющие способ создания пули при выстреле. Поля:

- $speed : float64$ — скорость (длина вектора скорости)
- $size : float64$ — размер
- $damage : int$ — урон

2.2.3 Color

Цвет (используется для отладочной отрисовки). Поля:

- $r : float32$ — красная компонента
- $g : float32$ — зеленая компонента
- $b : float32$ — синяя компонента
- $a : float32$ — альфа компонента (непрозрачность)

2.2.4 ColoredVertex

Вершина полигона (используется для отладочной отрисовки). Поля:

- $position : Vec2\langle float32 \rangle$ — положение
- $color : Color$ — цвет

2.2.5 CustomData

Тип данных используемый для отправки через объект *Debug* (используется для отладочной отрисовки). Может принимать следующие формы:

- *Log* — логирование. Поля:
 - *text* : *string* — текст для отображения
- *Rect* — прямоугольник. Поля:
 - *pos* : *Vec2* \langle *float32* \rangle — положение (левый нижний угол)
 - *size* : *Vec2* \langle *float32* \rangle — размер
 - *color* : *Color* — цвет для заполнения
- *Line* — отрезок. Поля:
 - *p1* : *Vec2* \langle *float32* \rangle — первая точка
 - *p2* : *Vec2* \langle *float32* \rangle — вторая точка
 - *width* : *float32* — ширина
 - *color* : *Color* — цвет
- *Polygon* — выпуклый полигон. Каждая вершина может быть покрашена отдельно — цвет будет интерполирован между вершинами. Поля:
 - *vertices* : *List* \langle *ColoredVertex* \rangle — список вершин
- *PlacedText* — текст. Поля:
 - *text* : *string* — текст для отображения
 - *pos* : *Vec2* \langle *float32* \rangle — позиция (в игровых координатах)
 - *alignment* : *TextAlignment* — способ выравнивания текста
 - *size* : *float32* — размер текста в пикселях
 - *color* : *Color* — цвет

2.2.6 ExplosionParams

Параметры взрыва от мины или от взрывной пули. Поля:

- *radius* : *float64* — радиус взрыва (половина стороны квадрата области взрыва)
- *damage* : *int* — урон

2.2.7 Game

Текущее состояние игры. Поля:

- *current_tick* : *int* — индекс текущего тика
- *properties* : *Properties* — параметры игры (константы)
- *level* : *Level* — уровень (карта)
- *players* : *List* \langle *Player* \rangle — список игроков (стратегий), участвующих в игре

- *units* : *List*⟨*Unit*⟩ — список живых юнитов
- *bullets* : *List*⟨*Bullet*⟩ — список летящих пуль
- *mines* : *List*⟨*Mine*⟩ — список **установленных** мин
- *loot_boxes* : *List*⟨*LootBox*⟩ — список объектов лута

2.2.8 Item

Предмет, являющийся лутом. Может принимать следующие формы

- *HealthPack* — аптечка. Поля:
 - *health* : *int* — пополняемое здоровье
- *Weapon* — оружие. Поля:
 - *weapon_type* : *WeaponType* — тип оружия
- *Mine* — мина. Без полей.

2.2.9 JumpState

Состояние прыжка юнита. Поля:

- *can_jump* : *boolean* — может ли юнит начать/продолжить прыжок
- *speed* : *float64* — скорость прыжка (в единицах в секунду)
- *max_time* : *float64* — максимальная длительность прыжка (в секундах)
- *can_cancel* : *boolean* — можно ли отменить текущий прыжок

2.2.10 Level

Уровень (карта). Поля:

- *tiles* : *List*⟨*List*⟨*Tile*⟩⟩ — двумерный список тайлов уровня

2.2.11 LootBox

Лут. Поля:

- *position* : *Vec2*⟨*float64*⟩ — положение (середина нижней границы)
- *size* : *Vec2*⟨*float64*⟩ — размер
- *item* : *Item* — предмет

2.2.12 Mine

Установленная мина. Поля:

- $player_id : int$ — id игрока, установившего мину
- $position : Vec2\langle float64 \rangle$ — положение (середина нижней границы)
- $size : Vec2\langle float64 \rangle$ — размер
- $state : MineState$ — текущее состояние
- $timer : Optional\langle float64 \rangle$ — время до смены состояния. Может отсутствовать
- $trigger_radius : float64$ — радиус запуска
- $explosion_params : ExplosionParams$ — параметры взрыва

2.2.13 MineState

Состояние мины. Варианты:

- Preparing — Подготовка
- Idle — Ожидание
- Triggered — Запуск

2.2.14 Player

Игрок (стратегия), участвующая в игре. Поля:

- $id : int$
- $score : int$ — текущий счет

2.2.15 Properties

Параметры игры (константы). Поля:

- $max_tick_count : int$ — максимальная длительность игры в тиках
- $ticks_per_second : float64$ — количество тиков в “секунду”
- $updates_per_tick : int$ — количество обновлений в тик. Каждое обновление продвигает игровое время на $\frac{1}{ticks_per_second \times updates_per_tick}$
- $loot_box_size : Vec2\langle float64 \rangle$ — размер коробок с лутом
- $unit_size : Vec2\langle float64 \rangle$ — размер юнитов
- $unit_max_horizontal_speed : float64$ — максимальная горизонтальная скорость юнитов
- $unit_fall_speed : float64$ — скорость падения юнитов
- $unit_jump_time : float64$ — обычное время прыжка

- *unit_jump_speed* : *float64* — обычная скорость прыжка
- *jump_pad_jump_time* : *float64* — время прыжка с трамплина
- *jump_pad_jump_speed* : *float64* — скорость прыжка с трамплина
- *unit_max_health* : *int* — максимальное (стартовое) здоровье юнита
- *weapon_params* : *Map*(*WeaponType* → *WeaponParams*) — параметры для каждого типа оружия
- *mine_size* : *Vec2*(*float64*) — размер **установленных** мин
- *mine_explosion_params* : *ExplosionParams* — параметры взрыва мин
- *mine_prepare_time* : *float64* — время подготовки мины
- *mine_trigger_time* : *float64* — время запуска мины
- *mine_trigger_radius* : *float64* — радиус запуска мины
- *kill_score* : *int* — очки за уничтожение вражеского юнита

2.2.16 TextAlignment

Выравнивание текста (используется для отладочной отрисовки). Варианты:

- Left — выравнивание по левому краю
- Center — выравнивание по центру
- Right — выравнивание по правому краю

2.2.17 Tile

Тайл уровня. Варианты:

- Empty — пусто
- Wall — стена
- Platform — платформа
- Ladder — лестница
- JumpPad — трамплин

2.2.18 Unit

Юнит. Поля:

- *player_id* : *int* — id игрока
- *id* : *int* — id юнита
- *health* : *int* — здоровье
- *position* : *Vec2*(*float64*) — позиция (середина нижней границы)

- *size* : *Vec2* \langle *float64* \rangle — размер
- *jump_state* : *JumpState* — текущее состояние прыжка
- *mines* : *int* — количество мин в инвентаре
- *weapon* : *Optional* \langle *Weapon* \rangle — текущее оружие, если есть

2.2.19 UnitAction

Действие юнита. Поля:

- *velocity* : *float64* — целевая горизонтальная скорость
- *jump* : *boolean* — начать/продолжить прыжок/вверх по лестнице
- *jump_down* : *boolean* — спрыгивать с/сквозь платформы/вниз по лестнице
- *aim* : *Vec2* \langle *float64* \rangle — направление прицеливания. Игнорируется если длина меньше 0.5
- *shoot* : *boolean* — контролирует стрельбу/перезарядку
- *swap_weapon* : *boolean* — поменять оружие
- *plant_mine* : *boolean* — установить мину

2.2.20 Vec2

Двумерный вектор. Поля:

- *x* : *float*
- *y* : *float*

2.2.21 Weapon

Оружие в руках юнита. Поля:

- *type* : *WeaponType* — тип оружия
- *params* : *WeaponParams* — параметры (константы)
- *magazine* : *int* — текущая обойма
- *spread* : *float64* — текущее значение разброса
- *fire_timer* : *Optional* \langle *float64* \rangle — время до следующего выстрела (отсутствует если выстрел возможен)
- *last_angle* : *Optional* \langle *float64* \rangle — последний угол прицеливания (отсутствует при подборе)

2.2.22 WeaponParams

Параметры оружия (константы). Поля:

- *magazine_size* : *int* — максимальный размер обоймы
- *fire_rate* : *float64* — время между выстрелами
- *reload_time* : *float64* — время перезарядки
- *min_spread* : *float64* — минимальный разброс
- *max_spread* : *float64* — максимальный разброс
- *recoil* : *float64* — значение отдачи
- *aim_speed* : *float64* — скорость прицеливания
- *bullet* : *BulletParams* — параметры пуль
- *explosion* : *Optional*⟨*ExplosionParams*⟩ — параметры взрыва пуль (если применимо)

2.2.23 WeaponType

Варианты:

- Pistol
- AssaultRifle
- RocketLauncher