**Разработка программного обеспечения для развлечения с использованием ООП и технологии Ray Casting.**

**Цель:** создать мир, построенный на квадратной сетке из стен однородной высоты и ровных сплошных полов и потолков. Для отрисовки игрового уровня из области наблюдения «выпускается» луч, который проходит через каждый столбец пикселей на экране. Далее алгоритм проверяет, не пересёк ли луч стену, и если пересёк, то в соответствии с этим выбирает и масштабирует текстуру стены согласно тому, где на уровне луч «столкнулся» со стеной и насколько далеко он распространился до этого. Далее создается одномерный буфер, в котором располагаются масштабированные спрайты, которые изображают врагов, бонусы и предметы[[6]](https://ru.wikipedia.org/wiki/Ray_casting#cite_note-6).

Создание уровней, основанных на сетке, преследовало две цели — столкновения луча и стены могут быть найдены более быстро, так как потенциальные столкновения становятся более предсказуемыми и сокращается потребление памяти. Однако обработка открытых пространств становится затруднительной.

**Главная цель (обязательное к разработке):** создание 3-мерного мира по 2-мерной карте без текстур с возможностью для объекта передвигаться по карте.

**Второстепенная цель (возможная реализация или реализация в будущем):** создать объекты-монетки, появляющиеся в случайном месте, и объекты-враги, которые могут двигаться и атаковать.

**Языки для разработки:** C#, Python.

**Ray casting**, **рейкастинг**, **метод «бросания лучей»** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *ray casting* — бросание лучей) — один из [методов](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4) [рендеринга](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BD%D0%B4%D0%B5%D1%80%D0%B8%D0%BD%D0%B3) в [компьютерной графике](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0), при котором сцена строится на основе замеров пересечения лучей с визуализируемой поверхностью. Этот термин впервые использовался в компьютерной графике в 1982 году в публикации Скотта Рота ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Scott Roth*), который применил его для описания метода рендеринга [CSG](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D1%81%D0%BF%D0%BB%D0%BE%D1%88%D0%BD%D0%B0%D1%8F_%D0%B3%D0%B5%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%8F)-моделей[[1]](https://ru.wikipedia.org/wiki/Ray_casting#cite_note-1).

Хотя термины «ray casting» (метод бросания лучей) и «ray tracing» ([трассировка лучей](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D1%81%D1%81%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0_%D0%BB%D1%83%D1%87%D0%B5%D0%B9)) часто использовались как синонимы в ранней литературе[[2]](https://ru.wikipedia.org/wiki/Ray_casting#cite_note-2), посвящённой компьютерной графике, в современной компьютерной графике эти термины различны и описывают разные методики, которые, тем не менее, имеют много общего[[3]](https://ru.wikipedia.org/wiki/Ray_casting#cite_note-3).

**Концепция**

Рейкастинг не является синонимом к рейтрейсингу (трассировке лучей), но он может быть представлен как сокращённая и существенно более быстрая версия алгоритма трассировки лучей. Оба алгоритма являются «image order» и используются в компьютерной графике для рендеринга трёхмерных сцен на двухмерный экран с помощью проекционных лучей, которые проецируются от глаз наблюдателя к [источнику света](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA_%D1%81%D0%B2%D0%B5%D1%82%D0%B0). Метод бросания лучей не вычисляет новые тангенсы лучей света, которые возникнут после того, когда луч, который проецируется от глаза к источнику света, пересечётся с поверхностью. Эта особенность делает невозможным точный рендеринг отражений, преломлений и естественной проекции теней с помощью рейкастинга. Однако все эти особенности могут быть добавлены с помощью «фальшивых» (обманных, аппроксимационных) методик, например, через использование текстурных карт или другие методы. Высокая скорость вычисления сделала рейкастинг удобным методом рендеринга в ранних компьютерных играх с трёхмерной графикой реального времени.

В реальной природе [источник света](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA_%D1%81%D0%B2%D0%B5%D1%82%D0%B0) испускает [луч](https://ru.wikipedia.org/wiki/%D0%9B%D1%83%D1%87) света, который, «путешествуя» по пространству, в конечном счёте «натыкается» на какую-либо преграду, которая прерывает распространение этого светового луча. Луч света можно представить в виде потока [фотонов](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%82%D0%BE%D0%BD), который движется вдоль вектора луча. В какой-либо точке пути с лучом света может случиться любая комбинация трёх вещей: [поглощение](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B3%D0%BB%D0%BE%D1%89%D0%B5%D0%BD%D0%B8%D0%B5), [отражение](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) и [преломление](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%BB%D0%BE%D0%BC%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5). Поверхность может отразить весь световой луч или только его часть в одном или нескольких направлениях. Поверхность может также поглотить часть светового луча, что приводит к потере интенсивности отраженного и/или преломлённого луча. Если поверхность имеет свойства [прозрачности](https://ru.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D0%B7%D1%80%D0%B0%D1%87%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0&action=edit&redlink=1), то она преломляет часть светового луча внутри себя и изменяет его направление распространения, поглощая некоторый (или весь) спектр луча (и, возможно, изменяя цвет). Суммарная интенсивность светового луча, которая была «потеряна» вследствие поглощения, преломления и отражения, должна быть в точности равной исходящей (начальной) интенсивности этого луча. Поверхность не может, например, отразить 66 % входящего светового луча, и преломить 50 %, так как сумма этих порций будет равной 116 %, что больше 100 %. Отсюда вытекает, что отраженные и/или преломлённые лучи должны «стыкаться» с другими поверхностями, где их поглощающие, отражающие и преломляющие способности снова вычисляются, основываясь на результатах вычислений входящих лучей. Некоторые из лучей, сгенерированных источником света, распространяются по пространству и, в конечном счете, попадают на [область просмотра](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%80%D1%82_%D0%BF%D1%80%D0%BE%D1%81%D0%BC%D0%BE%D1%82%D1%80%D0%B0) (глаз человека, объектив фото- или видеокамеры и т. д.). Попытка симулировать физический процесс распространения света путём трассировки световых лучей, используя компьютер, является чрезмерно расточительной, так как только незначительная доля лучей, сгенерированных источником света, попадает на область просмотра.

Первый алгоритм рейкастинга (не рейтрейсинга), используемый для рендеринга, был представлен Артуром Аппелем ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Arthur Appel*) в 1968 году[[5]](https://ru.wikipedia.org/wiki/Ray_casting#cite_note-5). В основе рейкастинга стои́т идея испускать лучи из «глаз» наблюдателя, один луч на [пиксель](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D0%BA%D1%81%D0%B5%D0%BB%D1%8C), и находить самый близкий объект, который блокирует путь распространения этого луча. Используя свойства материала и эффект света в сцене, алгоритм рейкастинга может определить затенение данного объекта. Предположение в упрощении основано на том, что если поверхность размещена перед светом, то свет достигнет поверхности и не будет блокирован или находиться в тени. Затенение поверхности вычисляется, используя алгоритмы затенения традиционной компьютерной трёхмерной графики. Одним из преимуществ рейкастинга является способность легко обрабатывать неплоские поверхности и сплошные тела вроде сферы или конуса. Если математическая поверхность может быть пересечена лучом, то она может быть отрендерена, используя рейкастинг. Сложные объекты могут быть созданы с использованием методов моделирования сплошных тел и легко отрендерены.

Рейкастинг для создания компьютерной графики впервые был применён учёными из [Mathematical Applications Group, Inc.](https://ru.wikipedia.org/w/index.php?title=Mathematical_Applications_Group,_Inc.&action=edit&redlink=1" \o "Mathematical Applications Group, Inc. (страница отсутствует)) ([англ.](https://en.wikipedia.org/wiki/Mathematical_Applications_Group,_Inc.)) (MAGI) из Элмсфорда, Нью-Йорк ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Elmsford, New York*). Эта компания была создана в 1966 году для проведения вычислений по радиологической экологии для [Министерства обороны США](https://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D0%B5%D1%80%D1%81%D1%82%D0%B2%D0%BE_%D0%BE%D0%B1%D0%BE%D1%80%D0%BE%D0%BD%D1%8B_%D0%A1%D0%A8%D0%90). Программное обеспечение разработки MAGI не только вычисляло то, как гамма-лучи отражаются от поверхностей (рейкастинг для радиации был сделан ещё в 1940-х), но и то, как проникают и преломляются внутри материалов. Под руководством доктора Филиппа Миттелмана ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Dr. Philip Mittelman*) учёные разработали метод генерирования изображений, взяв за основу то же самое программное обеспечение. В 1972 году MAGI стала коммерческой студией по созданию анимации. Она использовала рейкастинг для генерирования трёхмерной компьютерной анимации телевизионных реклам, образовательных фильмов, и, в конечном счете, для художественных кинофильмов. Компания MAGI ответственна за создание большей части анимации в научно-фантастическом фильме «[Трон](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%BE%D0%BD_(%D1%84%D0%B8%D0%BB%D1%8C%D0%BC))». Для создания анимации использовался исключительно метод рейкастинга. В 1985 году MAGI обанкротилась.