

Задание 3. Трассировка лучей

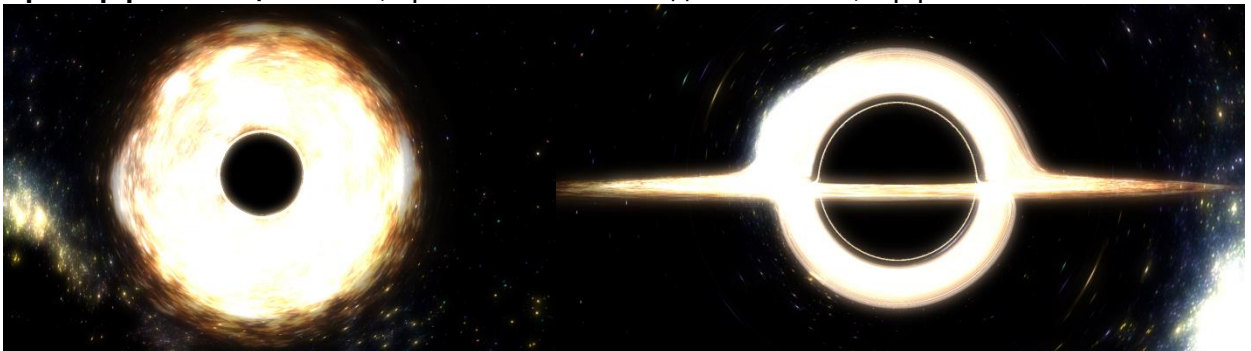
Автор задания: Владимир Афанасьев

Цель задания - реализовать визуализацию сцены с аналитически заданной геометрией при помощи алгоритма трассировки лучей. Предлагается визуализировать гравитационную линзу – чёрную дыру, окружённую аккреционным диском и звёздной панорамой. Необходимо реализовать метод трассировки криволинейных лучей, искажённых под действием гравитации.

Задание не претендует на физическую точность, а призвано продемонстрировать методы визуализации, позволяющие получить реалистичную картинку в современном кинематографе. Также в задании фигурирует понятие “луч”: здесь имеется не геометрический луч, а траектория света, которая может не быть участком прямой.



Пример реализации. База, простые и сложные доп. объекты, эффект свечения.



Пример реализации [7]. Вид сверху и сбоку. База, WebGL.

Правила оформления работы

Внимание! При невыполнении указанных требований работа может не проверяться!

Архив с заданием в формате **zip** должен быть залит в систему курса. В случае превышения максимального размера архива в системе нужно разбить его на части средствами архиватора. Заливать архив на файлообменники можно только в случае невозможности залить его в систему, по предварительному согласованию с проверяющими.

Содержимое архива:

1. Папка **src** (исходный код)

- Файлы исходного кода
- Файлы проекта
- **НЕ нужно** включать в архив папку **ipch**, базы данных программы **.ncb**, **.sdf**.
- Проект должен собираться из папки **src**

2. Папка **bin** (исполняемый код - конфигурация Release, 32 бит). Обязательно проверьте, что программа запускается из папки **bin**. Желательно, на другой машине.

- Исполняемый файл
- Библиотеки, необходимые для запуска
- Данные (модели, текстуры, файл настроек). Дублировать данные в папке **src** не нужно.
- 3 файла настроек.

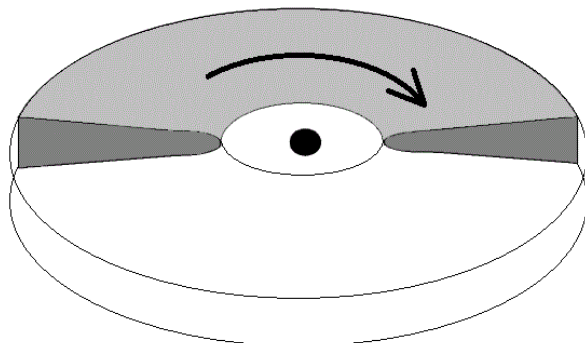
3. Папка **img** (визуализированные изображения сцены)

4. Файл **Readme.txt**

- Фамилия, имя, отчество, группа
- Операционная система
- Оборудование (процессор, видеокарта, объём памяти)
- Управление программой (формат задания настроек в файле настроек, описание интерфейса)
- Время работы программы для каждого варианта настроек
- Реализованные пункты из бонусной части

Введение

Как известно из общей теории относительности, гравитационное поле искажает пространство, и это затрагивает всё, в том числе, световые лучи становятся искривлёнными. Это приводит к тому, что массивные объекты искажают изображение предметов рядом с собой и за собой. Однако для того, чтобы эффект был заметным, объект должен иметь очень большую массу: например, отклонение луча от прямолинейного направления вблизи поверхности Солнца составляет всего $1.75''$ (угловой минуты) или $4,86 \cdot 10^{-4}$ градуса, что не создаёт видимого глазу искажения. Объектами, обладающими достаточной массой для создания видимого искажения, являются чёрные дыры. Чёрная дыра может быть окружена *аккреционным диском* [8] – вращающимся облаком падающего на неё вещества. Из-за вращения это облако стремится к плоской форме, наподобие колец Сатурна. Высокая скорость вещества и внутреннее трение приводит к его разогреву до высокой температуры, и вещество начинает светиться.



На рисунке показан схематично и в разрезе объёмный аккреционный диск вокруг чёрной дыры.

Базовая часть (до 10 баллов)

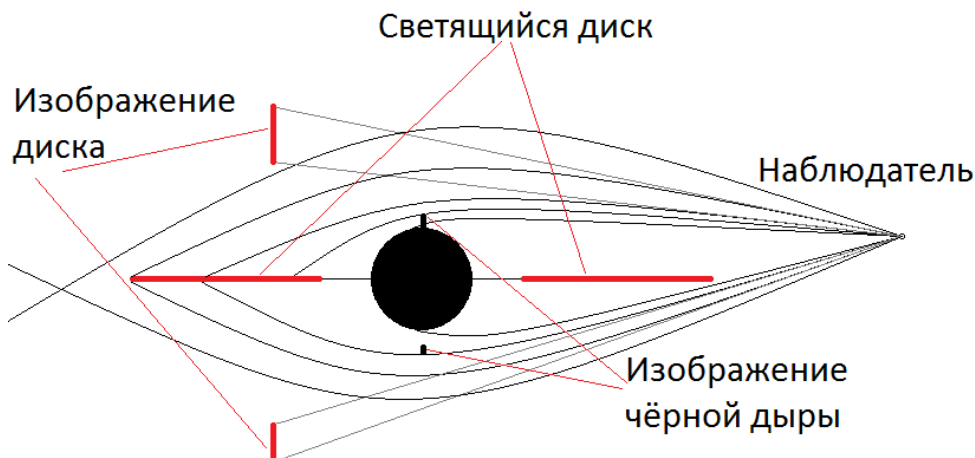
В базовой части требуется реализовать алгоритм обратной трассировки **криволинейных лучей** в сцене с тремя аналитически заданными объектами:

1. Сферическая чёрная дыра.
2. Плоский текстурированный аккреционный диск.
3. Бесконечно удалённая панорама звёздного окружения.

Разрешение результирующего цветного изображения должно быть не менее 512x512 пикселей. Разрешение, как и другие основные параметры, должно задаваться в файле настроек. **Необходимо сделать 3 или более различных файла настроек**, демонстрирующих разные ракурсы и варианты визуализации. Как минимум 1 ракурс должен быть видом сбоку, чтобы можно было оценить искривление изображения. Как минимум один вариант настроек должен позволять получить изображение не дольше, чем за 2 минуты на средней по мощности машине.

Все числа задаются в метрической системе (кг, м, с), для записи больших чисел используйте вид $15e+7$.

1. Масса чёрной дыры и коэффициент радиуса аккреционного диска (во сколько раз он больше чёрной дыры) должны задаваться в файле настроек. Радиус чёрной дыры (горизонта событий) вычисляется как радиус Шварцшильда: $R = 2G \frac{M}{c^2}$. Всякий луч, который пересёк снаружи сферу этого радиуса, теряется, его цвет становится равным 0.
2. Требуется реализовать алгоритм обратной трассировки лучей. Каждый луч начинается в центре проекции камеры и проходит через соответствующий пиксель изображения на плоскости проекции. Далее луч распространяется по определённой траектории в сцене, пока не попадёт в объект сцены, или не промахнётся мимо всех объектов. Цвет объекта, в который попал луч (либо цвет, собранный по пути более сложным способом) присваивается пикселю изображения в камере.
3. Траектория света вычисляется итерационно, исходя из гравитационного притяжения чёрной дыры. Из-за искривления световых лучей изображение аккреционного диска для удалённого наблюдателя окружает всю чёрную дыру (как видно на примерах реализации).



4. Необходимо реализовать камеру с точечной перспективной проекцией. Начало луча выбирается в позиции камеры, а направление выбирается для каждого пикселя так, чтобы луч проходил через соответствующую точку на картинной плоскости перед камерой. Расстояние до виртуальной картинной плоскости определяется выбранным углом обзора камеры (обычно 45-60 градусов по вертикали).

5. Плоскость аккреционного диска должна быть текстурирована. Требуется реализовать простейший метод выборки текстуры “ближайший сосед”.
6. Лучи, промахнувшиеся мимо объектов сцены, попадают в бесконечно удалённую панораму, цвет луча определяется выборкой из текстуры методом “ближайшего соседа”. Пересечение луча с бесконечно удалённой панорамой определяется только его направлением, не нужно пересекать луч с очень большой сферой!
7. Результат работы трассировщика должен сохраняться во внешний файл формата .bmp или .png рядом с исполняемым файлом. Просьба не использовать форматы, сжимающие с потерями.

Дополнительная часть (до 5 баллов)

Внимание! Все реализованные пункты дополнительной части должны конфигурироваться через внешний текстовый файл настроек. У проверяющего должна быть возможность легко модифицировать файл. В readme должно быть пояснение формата файла, как с ним работать. За каждый пункт указано максимальное число баллов, которое можно получить при сложной и качественной реализации.

- **Дополнительные простые объекты – планеты, спутники (+2)**
Нужно добавить в сцену не менее 2 других объектов, состоящих из примитивов, использованных в базе – это плоскости или сферы. Объекты могут быть покрашены сплошным цветом, либо быть зеркальными, либо можно реализовать на них модель освещения (например, Фонга).
- **Сложные полигональные объекты – астероиды, корабли (+3) – не складываются с простыми.**
Нужно добавить не менее 2 дополнительных объектов, состоящих из треугольников. Как реализовывать пересечение луча с такими объектами, описано ниже. Аналогично, эти объекты могут быть сплошного цвета, зеркальными или освещёнными по некоторой модели.
- **Параллелизм (+5)**
 - **+1** за простую реализацию на OpenMP, стандартных потоках и т.п
 - **+3** за Optix, CUDA, OpenCL или шейдеры OpenGL и **+2** за анимацию вращающегося диска
- **Антиалиасинг (+1)**
- **Генерация стереопары (+1) – не заменяет генерацию одного изображения с симметричной пирамидой видимости.**
- **Объёмный аккреционный диск либо затухание света (+4)** Окружающий чёрную дыру газ распределён в некотором объёме, диск имеет конечную толщину. Распределение газа в этом объёме можно сгенерировать с помощью шума Перлина [10]. Объёмный аккреционный диск заменяет плоский диск с текстурой из базы.
Альтернативно, можно реализовать аналогичным способом затухание света в объёмных облаках газа, находящихся между чёрной дырой и наблюдателем. При этом чёрная дыра не должна быть полностью закрыта, и нельзя делать облака полностью чёрными.
- **Альфа-смешивание (+1) – полупрозрачный аккреционный диск.** Цвет диска накладывается на цвет фона за ним: это может быть другая часть диска, звёзды или чёрная дыра.
- **Постобработка результата (+1)** Любые фильтры, применённые к изображению, позволяющие улучшить его визуальные качества. Например, эффект свечения (bloom).
- **Спектральные расчёты + гравитационное смещение спектра (+5)**
Здесь требуется моделирование гравитационного красного смещения [9]. Суть

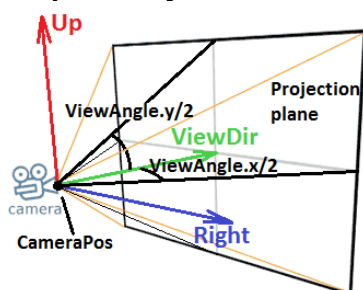
явления состоит в том, что длина волны излучения при изменении гравитационного потенциала тоже меняется. Спектр излучения, испущенного газом вблизи чёрной дыры, для удалённого наблюдателя сместится в красную сторону: свет потратит часть своей энергии на преодоление притяжения. Поэтому необходимо реализовать расчёты не в цвете, как в базовой части, а в интенсивностях излучения на разных длинах волн. Точного физического моделирования эффекта не требуется, подойдёт любая правдоподобная модель, например, описанная ниже.

Подсказки к решению

1. Вычисления

Для расчётов на CPU предлагается использовать библиотеку **glm** [1] для векторных вычислений. Рекомендуется на всякий случай использовать точность вычислений **double** из-за большого порядка величин, однако возможно корректно реализовать визуализацию и на точности **float32** (актуально при реализации на GPU или оптимизации).

2. Генерация луча. Модель камеры



Камера задаётся центром проекции *CameraPos*, ортогональной тройкой $(\vec{Up}, \vec{Right}, \vec{ViewDir})$ и углами обзора по горизонтали и вертикали. Результирующее изображение имеет разрешение $W \times H$ пикселей, и для того, чтобы не было искажения пропорций (пиксели квадратные), нужно соответствие углов обзора разрешению.

Каждый луч, проходящий через пиксель изображения с координатами (cx, cy) задаётся начальной точкой *CameraPos* и направлением

$$\vec{ViewDir} + \left(\frac{cx + 0.5}{W} - 0.5\right) \vec{Right} + \left(\frac{cy + 0.5}{H} - 0.5\right) \vec{Up}$$

Для того, чтобы это направление было верно, нужно сделать длину \vec{Up}, \vec{Right} равной соответствующему размеру плоскости проекции, а длину $\vec{ViewDir}$ – равной расстоянию от центра камеры до плоскости проекции. В числителях к координатам пикселя добавлено 0.5 для того, чтобы луч проходил через центр пикселя, а не его угол.

3. Итерационное вычисление траектории света

Траектория фотона вычисляется итерационно и является ломаной. Простейший способ вычисления координат фотона в следующий момент времени:



- a. По закону всемирного тяготения вычислить ускорение фотона в текущий момент времени: $a = \vec{r}G \frac{M}{r^3}$
- b. Считаем, что весь промежуток Δt от текущего до следующего момента времени на фотон действует постоянное ускорение, вычисленное в пункте а. Обратите внимание, что модуль скорости фотона должен быть постоянным и равным скорости света в вакууме. Поэтому на скорость фотона действует только нормальная составляющая ускорения – вектор скорости поворачивается, а тангенциальная переходит в энергию фотона (в базовой части это можно не учитывать, а в доп. задании по гравитационному смещению спектра следует изменить частоту фотона в соответствии с этой добавкой энергии)
- c. Вычисляем новую скорость и новое положение:

$$V += a \times \Delta t$$

$$Pos += V_0 \Delta t + a \frac{\Delta t^2}{2}$$

Важно – в добавке к Pos используется старая скорость V_0 , что моделирует равноускоренное движение фотона на промежутке времени Δt .

- d. В простейшем варианте промежуток времени Δt можно выбрать постоянным и достаточно маленьким для хорошей точности расчётов. Для увеличения скорости расчётов можно применять неравномерный шаг по времени: например, сделать шаг обратно пропорциональным ускорению в данной точке.

Желающие могут реализовать более корректный метод расчёта траектории света на основе численного решения уравнений ОТО: например, в [2], [3] описывается такой подход.

4. Пересечение луча с объектами

Луч света в предложенном подходе состоит из набора отрезков, проверка пересечений выполняется для каждого нового отрезка.

a. Пересечение отрезка и плоскости

Предлагается решить задачу в параметрическом виде: направленный отрезок \overrightarrow{AB} задан начальной точкой A и направляющим вектором $\vec{m} = B - A$; плоскость задана точкой P и нормалью \vec{n} . Решение нужно искать как $A + \vec{m}t$, где $t \in [0,1]$ – неизвестный параметр. В случае выбора глобальной системы координат так, что аккреционный диск лежит в плоскости XY , решение упрощается.

b. Пересечение отрезка и сферы

Предлагается также прибегнуть к параметрическому заданию прямой и решить квадратное относительно t уравнение пересечения этой прямой и сферы. Уравнение даст 2 решения в случае пересечения, нужно выбрать минимальное t . [4]

c. Пересечение отрезка и треугольника для доп. задания с полигональными объектами.

Здесь необходимо пересечь отрезок с плоскостью треугольника и провести дополнительные тесты на то, что точка лежит внутри треугольника. Это можно сделать как напрямую, так и другими способами [5]. Для пересечения луча с объектом, состоящим из многих треугольников, можно перебрать все треугольники и среди точек пересечения выбрать ближайшую к началу луча.

5. Загрузка текстуры

Загрузка производится с помощью класса `ATL CImage`, как сделано в шаблоне, или любым другим способом. В задание включена 32-битная текстура аккреционного диска с альфа-каналом. В базовой части предлагается учитывать только значения 0 и не-0 альфа-канала: текстура прозрачна – не учитываем пересечение, и луч идёт дальше, либо учитываем пересечение и приписываем лучу RGB цвет текстуры.

Для выполнения доп. задания с альфа-смешиванием нужно использовать

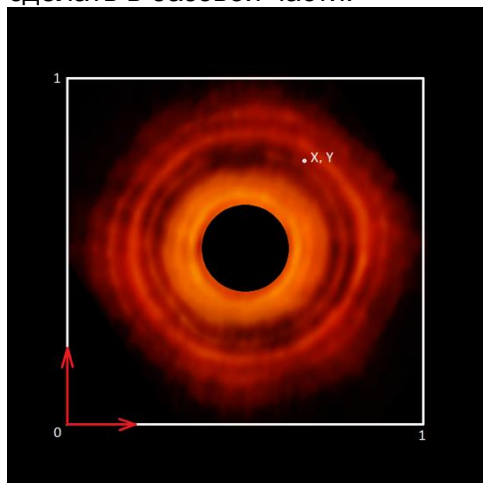
значения из альфа-канала.

Для доп. задания со спектральными расчётами к заданию прилагается спектральная текстура аккреционного диска в виде набора изображений для разных длин волн.

6. **Выборка текстуры**

Текстура представляет собой плоское изображение, “натянутое” на объёмную модель. В пространстве изображения вводятся т.н. текстурные координаты, соответствующие горизонтальному и вертикальному направлению, область значений которых – от 0 до 1. В некоторых точках на поверхности объекта указываются текстурные координаты (x, y) , что задаёт соответствие точки поверхности объекта и точки изображения. В промежутках между такими точками текстурные координаты интерполируются.

Простейший случай наложения текстуры – на плоский квадрат, что и предлагается сделать в базовой части.



В плоскости аккреционного диска должна быть введена система координат, связанная с углом текстуры. Оси лежат в плоскости диска, в центре текстуры находится чёрная дыра, длина координатных векторов задаёт размер аккреционного диска (т.е. все координаты (x, y) , лежащие на текстуре, в этом пространстве находятся в $[0, 1]$). При пересечении лучом плоскости координаты точки пересечения нужно перевести в эту систему координат (получаем текстурные координаты), затем привести их в пиксельные координаты, домножив на длину и ширину текстуры в пикселях. Далее, по методу “ближайшего соседа”, нужно округлить полученные координаты и присвоить лучу цвет соответствующего пикселя.

7. **Выборка из сферической панорамы**

Если луч уже достаточно долго не меняет своё направление на значительную величину, можно считать, что он будет прямым до бесконечности. Необходимо провести финальный тест пересечения этого луча с объектами сцены, и если пересечения нет, сделать выборку из сферической панорамы. Для этого нужно посчитать сферические координаты данного луча $\varphi \in [0, 2\pi)$ и $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$:

$$\varphi = \text{atan2}(x, y)$$

$$\theta = \text{asin}(z)$$

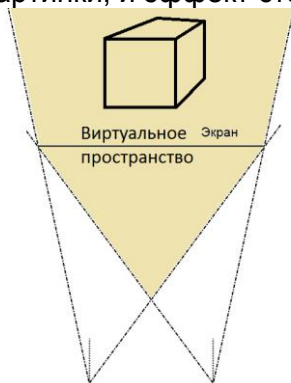
– если принять (x, y, z) за координаты единичного вектора направления луча. Сферическая текстура, как правило, является изображением с соотношением сторон 2:1 – её размеры $2H \times H$. Приводим φ и θ в диапазоны $[0, 2H)$ и $[0, H]$ и аналогично предыдущему пункту выбираем пиксель по координатам методом ближайшего соседа.

8. **Антиалиасинг**

Предлагается устранять ступенчатость методом мультисэмплинга. Для этого через каждый пиксель пропускается не один луч, а несколько – через различные точки этого пикселя, помимо центра. Результирующий цвет этих лучей усредняется и присваивается пикселю.

9. Генерация стереопары

Необходимо реализовать генерацию 2 изображений с разных ракурсов, со сдвинутой пирамидой видимости (в отличие от базовой части, где симметричная пирамида). Это необходимо для моделирования условий показа зрителю на стерео-мониторе. Для получения эффекта стерео на таком большом объекте как чёрная дыра следует установить расстояние между ракурсами достаточно большим – соразмерным другим расстояниям в сцене. Соответственно, и размер виртуального экрана (плоскости проекции камеры) тоже будет большим. Если же использовать параметры человеческого зрения, оба ракурса дадут одинаковые картинки, и эффект стерео не получится.



Изображения можно сохранить в отдельные файлы, либо в один файл рядом друг с другом, либо в специализированный формат для стереопар.

10. Объёмный аккреционный диск/затухание

Предлагается реализовать аккреционный диск в виде облака светящегося газа. Распределение газа можно сгенерировать с помощью шума Перлина [10]. Луч проходит через этот газ, и цвет собирается по объёму равномерной выборкой. Аналогично с затуханием – нужно создать поглощающее облако газа где-то между наблюдателем и светящимися объектами так, чтобы его было видно, и моделировать поглощение по закону Бугера-Ламберта-Бера.

11. Альфа-смешивание

Нужно сделать так, чтобы при пересечении лучом плоскости аккреционного диска он не прекращал трассироваться и попал в какой-то объект за диском. Получаем два цвета: RGB_0 – цвет диска и RGB_1 – цвет луча за диском. Результирующий цвет равен $RGB = \alpha RGB_0 + (1 - \alpha) RGB_1$, где α – значение альфа-канала текстуры диска в соответствующем пикселе, нормированное в отрезок $[0,1]$.

12. Спектральные расчёты и гравитационное смещение спектра

Необходимо испускать из каждого пикселя изображения не один луч, который затем принесёт цвет, а набор лучей для выборки длин волн от 380 до 780 нм. Каждый луч будет нести интенсивность излучения на данной длине волны. Длина волны луча может меняться в процессе трассировки под действием гравитации – т.н. красное и фиолетовое смещение. При попадании в спектральную текстуру выбирается наиболее близкая к данному лучу длина волны в текстуре, и лучу приписывается соответствующая интенсивность. Обратите внимание, что эта интенсивность в камере будет соответствовать той длине волны, для которой был испущен луч, а не длине волны при выборке из текстуры! (В отсутствие гравитации они бы совпадали)

После трассировки всех таких лучей через пиксель и получения интенсивностей излучения на всех длинах волн из выборки (спектра) нужно вычислить RGB цвет этого спектра. Это делается с помощью свёртки со спектральными кривыми CIE. [6] Рекомендуется сначала провести свёртку в цвет XYZ x, затем с помощью матрицы перейти к цвету sRGB.

13. Константы и некоторые величины

Скорость света в вакууме $c \approx 3 \cdot 10^8$ м/с

Гравитационная постоянная $G \approx 6,674 \cdot 10^{-11} \text{ Н} \cdot \frac{\text{м}^2}{\text{кг}^2}$

Типичная масса чёрной дыры (объект в центре нашей галактики) $M \approx 8,57 \cdot 10^{36}$ кг
Радиус горизонта событий при такой массе $R \approx 12,7 \cdot 10^9$ м

Ссылки

1. <http://jasmscole.com/2014/10/04/what-do-black-holes-look-like> Визуализация чёрной дыры
2. <http://spiro.fisica.unipd.it/~antonell/schwarzschild> Визуализация чёрной дыры
3. <http://glm.g-truc.net/0.9.7/index.html> Библиотека glm
4. <http://ray-tracing.ru/articles245.html> Пересечение луча со сферой
5. <http://ray-tracing.ru/articles213.html> Пересечение луча с треугольником
6. https://ru.wikipedia.org/wiki/Цветовая_модель Вычисление цвета по спектру
7. <http://sirxemic.github.io/Interstellar> Реализация на WebGL
8. https://ru.wikipedia.org/wiki/Аккреционный_диск Аккреционный диск
9. https://ru.wikipedia.org/wiki/Гравитационное_красное_смещение Красное смещение
10. https://ru.wikipedia.org/wiki/Шум_Перлина Шум Перлина