

Классы:

- **Canvas:**
 - C.load(filename) - загружает картинку filename
 - C.save(filename) - сохраняет текущую картинку в filename
 - C(int x, int y) - создаёт пустой холст x на y
 - C.fill_canvas(int R, int G, int B) - заливает весь холст цветом R.G.B
 - C.put_pixel(int x, int y, int R, int G, int B) - закрашивает пиксель (x, y) цветом R.G.B (Ось 0x вправо, Ось 0y вниз)
- **Reader:**
 - R(string scene, string objs) - настраивает ридер на чтение параметров сцены в scene и объектов в objs
 - R.read_scene(vector<double>& camera, ...) - читает параметры сцены в переданные переменные
 - R.read_objects(vector<figure*>& objs) - читает объекты в переданный вектор
- **Raytracer:**
 - R.calc_shadow(...) - private - служебная функция, расчёт теней
 - R.calc_light(...) - private - служебная функция, расчёт света (диффузный / глянец)
 - R(string scene, string objs, string saveto, int thread_num) - конструктор...
 - R.raytrace() - берёт инструкции из scene, объекты из objs, рассчитывает на CPU в thread_num потоках и сохраняет в saveto
 - R.stripe_trace(...) - служебная функция, расчёт фрагмента экрана, распоточивается
- **Renderer:**
 - R(string scene_tmpl, string obj_tmpl, string save_tmpl, int thread_num) - конструктор...
 - R.render(int frame_number, bool logs) - рендерит frame_number кадров по сценам scene_tmpl, объектам string_tmpl, сохраняет кадры по save_tmpl, рендер на CPU в thread_num потоках
 - R.render_scope(...) - вспомогательная функция, рендерит часть кадров, распоточивается
 - R.join_to_video(string frame_tmpl, string video_name, int framerate, int quality, string ffmpeg_dir) - склеивает кадры из frame_tmpl в видео video_name, FPS = framerate, качество quality: 1 = лучшее, 30 = худшее, ffmpeg_dir - путь к необходимой утилите.
(<https://www.gyan.dev/ffmpeg/builds/ffmpeg-release-essentials.zip>)
- **figure:**
 - uint8_t surface: 0 = диффузный объект, 1 = глянцевый объект
 - virtual bool traceback(s_point, trace_ray, res_v, mode) - трассирует луч trace_ray из точки пространства s_point в режиме mode (нужно для оптимизации...), сохранение результата в вектор res_v
 - res_v = [x, y, z, R, G, B, Nx, Ny, Nz, stype, otype] - координаты пересечения, Цвет точки пересечения, нормаль к пересеченной поверхности, тип поверхности (диффуз / глянец), тип пересеченного объекта (1 = сфера, 2 = коробка, 3 = пирамидка)
- **box:**

- double half_diag_quad - квадрат половины диагонали (= квадрат радиуса описанной сферы...) - значение для оптимизации, вычисляется при инициализации объекта
- bool internal(...) - private - служебная функция, определяет принадлежность точки грани.
- **tetra:**
 - vector<double> optimize_center - центр описанной сферы - значение для оптимизации, вычисляется при инициализации
 - double circumscribed_RAD - радиус описанной сферы - значение для оптимизации, вычисляется при инициализации
 - bool internal(...) - private - служебная функция, определяет принадлежность точки полигону тетраэдра
- **spinlock:**
 - Самописный мьютекс, .lock() / .unlock()

Функции:

- split_by_space(...) - парсит строку на составляющие, нужна для чтения файлов
- vectmul(vect_1, vect_2) - векторное произведение, возвращает вектор
- scalmul(vect_1, vect_2) - скалярное произведение
- dist_quad(point_1, point_2) - квадрат расстояния между точками (без корня для оптимизации)
- volume_vect(vect_1, vect_2, vect_3) - объем параллелепипеда, натянутого на три вектора (не помню зачем нужна... :()
- vectmul_area(vect_1, vect_2) - площадь векторного произведения
- int closest_point(...) - ближайшая точка к данной - необходима при наложении трассируемых объектов
- to_length(vector, len) - приводит вектор vector к длине len
- void check_system_cores() - вывод оптимального числа потоков текущей системы

Форматы:

- Параметры сцены в текстовом файле:

camera X Y Z	Положение камеры
normal_ X Y Z	Направление “взгляда”
upvector X Y Z	Направление “вверх”
light X Y Z	Положение лампочки
screen_dist D	Расстояние до экрана
view_depth VD	Глубина зрения
screen_width SW	Разрешение экрана X
screen_height SH	Разрешение экрана Y

Порядок строк в файле не важен.

- Объекты сцены в текстовом файле:

sphere R G B x y z Radius Surface
box R G B x_min y_min z_min x_max y_max z_max Surface
tetra R G B x1 y1 z1 ... x4 y4 z4 Surface