RTM

0.1.0

Создано системой Doxygen 1.8.13

Оглавление

Глава 1

Титульная страница

Этот проект представляет из себя систему для моделирования дорожного движения.
В нём можно тестировать системы управления светофорами, машинами (в ближайшем будущем это понадобится) и многое другое.
А также можно просто позалипать на машинки, интересно катающиеся по дорогами, которые может создать любой человек! с:

Версия

0.1.0

Автор

Владимир Северов (Vladimir Severov)

Необходимо сделать Предстоит ещё много работы для серьезного использования данной системы, однако в рамках курсового проекта этот проект можно считать успешным.

Глава 2

Список задач

page Титульная страница

Предстоит ещё много работы для серьезного использования данной системы, однако в рамках курсового проекта этот проект можно считать успешным.

Глава 3

Алфавитный указатель пространств имен

3.1 Пространства имен

Полный список документированных пространств имен.

Глава 4

Иерархический список классов

4.1 Иерархия классов

Иерархия классов.

Глава 5

Алфавитный указатель классов

5.1 Классы

Классы с их кратким описанием.

Глава 6

Пространства имен

6.1 Пространство имен rtm

Пространство имен для проекта RTM.

Классы

- class BuildingObject

  Класс, описывающий строения (здания)
- class BushObject

  Класс, описывающий кусты
- class CarObject

  Класс, описывающий машины
- class CoatingObject

  Класс покрытия секции карты
- class CoatingUnion

  Класс объединения покрытий
- class ControlUnit

  Класс управляющего блока (светофор)
- class CrossroadObject

  Класс пересечения дорог
- class DrivewayObject

  Класс прямой дороги
- class DynamicObject

  Класс динамического объекта (который двигается, обновляется)
- class MapObject

  Класс статического объекта карты
- class PuddleCoating

  Класс, описывающий лужи
- class RoadCoating

  Класс, описывающий дороги
- struct SpawnType

  Структура, описывающая параметры точки генерации объектов
- class StaticObject

  Класс статического объекта (который не обновляется)

- class TurnObject

  Класс поворота дороги
- class VehicleObject

  Класс транспорта (динамического объекта карты)
- class WorldController

  Класс контроллера карты, связующее звено всех объектов
- class WorldObject

  Класс объекта мира (родитель всех условно объемных объектов)
- class WorldScene

  Класс главной сцены, на которой всё и происходит (для отрисовки)

Определения типов

- using WorldControllerUnique = std::unique_ptr< WorldController >

  Умный указатель для класса WorldController.
- using SpawnVector = std::vector< SpawnType >

  Массив точек генерации объектов
- using CoatingUnique = std::unique_ptr< CoatingObject >

  Умный указатель для класса CoatingObject.
- using CoatingVector = std::vector< CoatingUnique >

  Массив объектов класса CoatingObject.
- using CoatingMatrix = std::vector< CoatingVector >

  Матрица объектов класса CoatingObject.
- using CoatingUnionShared = std::shared_ptr< CoatingUnion >

  Умный указатель для класса CoatingUnion.
- using CoatingUnionVector = std::vector< CoatingUnionShared >

  Массив объектов класса CoatingUnion.
- using CoatingUnionMatrix = std::vector< CoatingUnionVector >

  Матрица объектов класса CoatingUnion.
- using ControlUnitShared = std::shared_ptr< ControlUnit >

  Умный указатель для класса ControlUnit.
- using ControlUnitVector = std::vector< ControlUnitShared >

  Массив объектов класса ControlUnit.
- using StaticShared = std::shared_ptr< StaticObject >

  Умный указатель для класса StaticObject.
- using StaticVector = std::vector< StaticShared >

  Массив объектов класса StaticObject.
- using StaticMatrix = std::vector< StaticVector >

  Матрица объектов класса StaticObject.
- using DynamicShared = std::shared_ptr< DynamicObject >

  Умный указатель для класса DynamicObject.
- using DynamicVector = std::vector< DynamicShared >

  Массив объектов класса DynamicObject.
- using Directions = std::array< bool, 8 >

  Массив возможных направлений движений по кусочку объекта
- using LinesCounts = std::array< size_t, 4 >

  Массив количества полос в каждом напрвлении для перекрестков
- using DirectionSignals = std::array< SignalType, 4 >

  Массив сигналов из одного напрвлениия в каждое
- using CrossroadSignals = std::array< DirectionSignals, 4 >

Массив сигналов для всех напрвлений перекрестка

- using SignalSprites = std::array< cocos2d::Sprite *, 5 >

  Массив всех текстур сигналов из одного направления в одно
- using SignalsSprites = std::array< SignalSprites, 3 >

  Массив всех текстур сигналов из одного направления в каждое (вперед, влево, вправо)
- using DirectionsSignalSprites = std::array< SignalsSprites, 4 >

  Массив всех текстур сигналов для перекрестка

Перечисления

- enum AngleType {
  NullAngle = -1, Up = 0, Right, Down,
  Left, UpRight, DownRight, DownLeft,
  UpLeft }

  Тип для определения положения некоторых объектов и индикации разрешенных направлений на кусочке объекта
- enum DirectionType {
  NullDirection = -1, Upward = 0, Rightward, Downward,
  Leftward }

  Тип для задания направления движения транспорта
- enum CoatingUnionType {
  NoCoatingUnion = -1, DrivewayType, CrossroadType, TCrossroadType,
  TurnType }

  Возможные типы дорожных объединений
- enum DirectionSignalIndex { ForwardSignalIndex = 0, LeftwardSignalIndex, RightwardSignalIndex }

  Индексы массива для каждого типа сигнала
- enum SignalType {
  NotWorking = 0, Allowed, Warning, Forbidden,
  Closed }

  Возможные сигналы светофора
- enum StateType { NotStarted, MustStart, Started, MustStop }

  Возможные состояния для манёвров (движение, поворот, перестроение)
- enum CoatingType { AsphaltCoating = 0, IceAsphaltCoating = 1 }

  Типы покрытий
- enum RoadType {
  RoadTypeNo0 = 0, RoadTypeNo1, RoadTypeNo2, RoadTypeNo3,
  RoadTypeNo4, RoadTypeNo5, RoadTypeNo6, RoadTypeNo7,
  RoadTypeNo8, RoadTypeNo9, RoadTypeNo10, RoadTypeNo11,
  RoadTypeNo12, RoadTypeNo13, RoadTypeNo14, RoadTypeNo15,
  RoadTypeNo16, RoadTypeNo17 }

  Типы дорог
- enum SignalFileId { ForwardSignalId = 1, LeftwardSignalId = 6, RightwardSignalId = 11 }

  Индексы, начиная с которых начинаются текстуры сигналоа определенного типа

Функции

- void CheckCollisions (WorldController *const world)

  Функция для вычисления столкновений в мире
- std::string GetFilename (std::string const &mask, size_t number)

  Функция для получения пути к файлу по маске
- AngleType SumAngleTypes (AngleType a, AngleType b)

Функция для суммирования двух угловых типов

- float CountDeceleration (float maxSpeed)

  Функция для подсчёта рекомендуемого коэффициента замедления транспорта

Функции для работы с параметрами положения объектов

- bool SameCoordinates (float a, float b, float delta=COORD_DELTA)

  Функция для сравнения двух координат с определенной точностью
- float RoundCoordinate (float coordinate, float delta=COORD_DELTA)

  Функция пытается округлить координаты до центра клетки
- float RoundToCenter (float coordinate)

  Функция для округления координаты до центра клетки
- bool InCenter (float coordinate, float delta=COORD_DELTA)

  Функция для проверки координаты на центральность
- float DistanceToNextCenter (float x, float y, float angle)

  Функция для нахождения расстояния до следующего центра клетка по ходу движения
- bool CenterIsCrossed (float x, float y, float angle, float lastDelta)

  Функция проверяет, пересек ли объект центр клетки (центральную линию, перпендикулярную направлению движения)
- bool SameAngles (float a, float b, float delta=ANGLE_DELTA)

  Фукнция для сравнения двух углов
- float RoundAngle (float angle, float delta=ANGLE_DELTA)

  Фукнция пытается округлить угол до одно из главных направлений (период $\pi/4$, т.е. 0, $\pi/4$, $\pi/2$, ...)
- float NormalizeAngle (float angle)

  Функция для нормализации угла до диапазона [-$\pi/2$;$\pi/2$)

Конверторы схожих типов

- int PixelToCell (float coordinate)

  Функция для конвертации координаты в номер ячейки
- float CellToPixel (int cellNumber)

  Функция для конвертации номера ячейки в координату центра
- AngleType AngleToAngleType (float angle)

  Функция для конвертации угла в угловой тип
- DirectionType AngleToDirection (float angle)

  Функция для конвертации угла в напраление
- float AngleTypeToAngle (AngleType angle)

  Функция для конвертации углового типа в угол
- DirectionType AngleTypeToDirection (AngleType angle)

  Функция для конвертации углового типа в направление
- float DirectionToAngle (DirectionType direction)

  Функция для конвертации направления в угол
- AngleType DirectionToAngleType (DirectionType direction)

  Функция для конвертации направления в угловой тип

Переменные

- std::array< size_t, 2 > const COATING_INDEXES

  Индексы, начиная с которых начинаются текстуры покрытий определенного типа

Константы для флага isRight

- bool const LEFT { false }

  Влево

- bool const RIGHT { true }
  Вправо

Заранее посчитанные операции над π

- float const F_PI_8 { 0.39269908169872415480 8f }
  π / 8
- float const F_PI_4 { 0.78539816339744830961 6f }
  π / 4
- float const F_PI_2 { 1.57079632679489661923f }
  π / 2
- float const F_PI { 3.14159265358979323846f }
  π
- float const F_2_PI { 6.28318530717958647692f }
  2 $*$ π

Константы для конвертации углов из радиан в градусы и обратно

- float const DEG_RAD { F_PI / 180.f }
  Коэффициент для перевода из градусов в радианы
- float const RAD_DEG { 180.f / F_PI }
  Коэффициент для перевода из радиан в градусы

Заранее посчитанные углы

- float const ANGLE_UP { 0.f }
  Угол вверх
- float const ANGLE_RIGHT { F_PI_2 }
  Угол вправо
- float const ANGLE_DOWN { -F_PI }
  Угол вниз
- float const ANGLE_LEFT { -F_PI_2 }
  Угол влево
- float const ANGLE_UP_RIGHT { F_PI_4 }
  Угол по диагонали вверх вправо
- float const ANGLE_DOWN_RIGHT { F_PI - F_PI_4 }
  Угол по диагонали вниз вправо
- float const ANGLE_DOWN_LEFT { -F_PI + F_PI_4 }
  Угол по диагонали вниз влево
- float const ANGLE_UP_LEFT { -F_PI_4 }
  Угол по диагонали вверх влево

Допустимые погрешности

- float const ANGLE_DELTA { 1.f $*$ DEG_RAD }
  Погрешность для углов
- float const COORD_DELTA { 1.f }
  Погрешность для координат
- float const NEAR_DELTA { 1.f }
  Максимальное расстояние до объектов, которые недалеко

Парамметры карт

- size_t const CELL_SIZE { 30 }
  Длина (ширина) ячейки карты
- size_t const ROTATION_RADIUS { CELL_SIZE }

Желаемый радиус поворота транспорта
- float const MIN_TIME_FACTOR { 0.5f }
  Минимальный коэффициент ускорения времени. Если меньше 1, то замедлениие
- float const MAX_TIME_FACTOR { 4.f }
  Максимальный коэффициент ускорения времени

Номера слоев для разных объектов. Чем больше, тем выше (ближе к нам)

- int const BACKGROUND_LAYER_Z_ORDER { -1 }
  Номер слоя для слоя фона
- int const MAIN_LAYER_Z_ORDER { 0 }
  Номер слоя для главного слоя (на нем все объекты)
- int const COATING_OBJECT_Z_ORDER { -2 }
  Номер слоя для покрытий (дорог)
- int const SIGNAL_Z_ORDER { -1 }
  Номер слоя для стрелок светофора
- int const VEHICLE_OBJECT_Z_ORDER { 0 }
  Номер слоя для транспорта
- int const MAP_OBJECT_Z_ORDER { 1 }
  Номер слоя для статичных объектов карты

Область видимости при движении вперед

- float const VIEW_RADIUS { 60.f }
  Радиус
- float const VIEW_ANGLE { 25.f * DEG_RAD }
  Ширина угла в каждую сторону
- float const VIEW_ANGLE_SHIFT { 0.f }
  Сдвиг области обзора

Область видимости при повороте

- float const ROTATION_VIEW_RADIUS { 50.f }
  Радиус
- float const ROTATION_VIEW_ANGLE { 30.5f * DEG_RAD }
  Ширина угла в каждую сторону
- float const ROTATION_VIEW_ANGLE_SHIFT { 29.5f * DEG_RAD }
  Сдвиг области обзора

Область видимости незадолго до поворота

- float const TURN_VIEW_RADIUS { 60.f }
  Радиус
- float const TURN_VIEW_ANGLE { 30.f * DEG_RAD }
  Ширина угла в каждую сторону
- float const TURN_VIEW_ANGLE_SHIFT { 10.f * DEG_RAD }
  Сдвиг области обзора

Область видимости на нерегулируемом перекрестке

- float const CROSSROAD_VIEW_RADIUS { 58.f }
  Радиус
- float const CROSSROAD_VIEW_ANGLE { 57.5f * DEG_RAD }
  Ширина угла в каждую сторону
- float const CROSSROAD_VIEW_ANGLE_SHIFT { -17.5f * DEG_RAD }
  Сдвиг области обзора

Область видимости до перестроения

- float const LINE_CHANGING_VIEW_RADIUS { 60.f }
  Радиус
- float const LINE_CHANGING_VIEW_ANGLE { 30.f * DEG_RAD }
  Ширина угла в каждую сторону
- float const LINE_CHANGING_VIEW_ANGLE_SHIFT { 20.f * DEG_RAD }
  Сдвиг области обзора

Значения по умолчанию

- DirectionSignals const DEFAULT_DIRECTIONS_SIGNALS = { NotWorking, NotWorking, NotWorking }
  Значения по умолчанию для массива сигналов в одном направлении (светофора в этом направлении нет)
- CrossroadSignals const DEFAULT_CROSSROAD_SIGNALS
  Значения по умолчанию для массива сигналов всего перекрестка (светофора на перекрестке нет)
- SignalSprites const DEFAULT_SIGNAL_SPRITES = { nullptr, nullptr, nullptr, nullptr, nullptr }
  Пустой массив текстур сигналов для одного типа сигнала одного направления
- SignalsSprites const DEFAULT_SIGNALS_SPRITES = { DEFAULT_SIGNAL_SPRITES, DEFAULT_SIGNAL_SPRITES, DEFAULT_SIGNAL_SPRITES }
  Пустой массив текстур сигналов для одного направления
- DirectionsSignalSprites const DEFAULT_DIRECTIONS_SIGNAL_SPRITES
  Пустой массив текстур сигналов для всего перекрестка

Маски названий файлов

- std::string const BACKGROUND_FILENAME_MASK { "res/background/Background↩No%No%.png" }
  Маска файлов фонов
- std::string const MAP_FILENAME_MASK { "res/map/MapNo%No%.rtmm" }
  Маска файлов карт
- std::string const ROAD_FILENAME_MASK { "res/coating/road/RoadNo%No%.png" }
  Маска файлов текстур дорог
- std::string const PUDDLE_FILENAME_MASK { "res/coating/puddle/PuddleNo%No%.png" }
  Маска файлов текстур грязи
- std::string const SIGNAL_FILENAME_MASK { "res/signal/SignalNo%No%.png" }
  Маска файлов текстур сигналов
- std::string const BUILDING_FILENAME_MASK { "res/static/building/Building↩No%No%.png" }
  Маска файлов текстур зданий
- std::string const BUSH_FILENAME_MASK { "res/static/bush/BushNo%No%.png" }
  Маска файлов текстур кустов
- std::string const CAR_FILENAME_MASK { "res/dynamic/vehicle/CarNo%No%.png" }
  Маска файлов текстур машин

Параметры дорог

- std::array< float, 2 > const ROADS_RESISTANCES
  Массив коэффициентов трения для каждого типа объекта
- std::array< Directions, 18 > const ROADS_DIRECTIONS
  Массив возможных направлений для каждой типа кучоска дороги

Параметры машин

- std::array< float, 6 > const CARS_MAX_SPEEDS
  Массив максимальных скоростей для машин
- std::array< float, 6 > const CARS_ACCELERATIONS
  Массив ускорений для машин

6.1.1 Подробное описание

Пространство имен для проекта RTM.

6.1.2 Типы

6.1.2.1 Directions

using rtm::Directions = typedef std::array<bool, 8>

Массив возможных направлений движений по кусочку объекта

См. также

AngleType

6.1.2.2 LinesCounts

using rtm::LinesCounts = typedef std::array<size_t, 4>

Массив количества полос в каждом напрвлении для перекрестков

См. также

DirectionType

6.1.2.3 DirectionSignals

using rtm::DirectionSignals = typedef std::array<SignalType, 4>

Массив сигналов из одного напрвлениия в каждое

См. также

DirectionType

6.1.2.4 CrossroadSignals

using rtm::CrossroadSignals = typedef std::array<DirectionSignals, 4>

Массив сигналов для всех напрвлений перекрестка

См. также

DirectionType

6.1.2.5 SignalSprites

using rtm::SignalSprites = typedef std::array<cocos2d::Sprite*, 5>

Массив всех текстур сигналов из одного направления в одно

См. также

SignalType

6.1.2.6 SignalsSprites

using rtm::SignalsSprites = typedef std::array<SignalSprites, 3>

Массив всех текстур сигналов из одного направления в каждое (вперед, влево, вправо)

См. также

DirectionSignalIndex

6.1.2.7 DirectionsSignalSprites

using rtm::DirectionsSignalSprites = typedef std::array<SignalsSprites, 4>

Массив всех текстур сигналов для перекрестка

См. также

DirectionType

6.1.3 Перечисления

6.1.3.1 AngleType

enum rtm::AngleType

Тип для определения положения некоторых объектов и индикации разрешенных направлений на кусочке объекта

Элементы перечислений

NullAngle	Неинициализированный угол
Up	Вверх
Right	Вправо
Down	Вниз
Left	Влево
UpRight	По диагонали вверх вправо
DownRight	По диагонали вниз влево
DownLeft	По диагонали вниз влево
UpLeft	По диагонали вверх вправо

6.1.3.2 DirectionType

enum rtm::DirectionType

Тип для задания направления движения транспорта

Элементы перечислений

NullDirection	Неинициализированное напрвление
Upward	Направление вверх
Rightward	Направление вправо
Downward	Направление вниз
Leftward	Направление влево

6.1.3.3 CoatingUnionType

enum rtm::CoatingUnionType

Возможные типы дорожных объединений

Элементы перечислений

NoCoatingUnion	Неинициализированный тип
DrivewayType	Прямая дорога
CrossroadType	Обычный перекресток
TCrossroadType	Т-образный перекресток
TurnType	Поворот

6.1.3.4 DirectionSignalIndex

enum rtm::DirectionSignalIndex

Индексы массива для каждого типа сигнала

Элементы перечислений

ForwardSignalIndex	Сигнал в прямом напрвлении
LeftwardSignalIndex	Сигнал в при повороте налево
RightwardSignalIndex	Сигнал в при повороте направо

6.1.3.5 SignalType

enum rtm::SignalType

Возможные сигналы светофора

Элементы перечислений

NotWorking	Светофор не работает (равносильно его отсутствию)
Allowed	Зеленый сигнал
Warning	Желтый сигнал
Forbidden	Красный сигнал
Closed	В данном напрвлении движение запрещено

6.1.3.6 StateType

enum rtm::StateType

Возможные состояния для манёвров (движение, поворот, перестроение)

Элементы перечислений

NotStarted	Не начато (не выполняется)
MustStart	Необходимо начать
Started	Начато
MustStop	Необходимо закончить

6.1.3.7 CoatingType

enum rtm::CoatingType

Типы покрытий

Элементы перечислений

AsphaltCoating	Асфальтовое покрытие
IceAsphaltCoating	Асфальтовое покрытие со льдом

6.1.3.8 RoadType

enum rtm::RoadType

Типы дорог

Элементы перечислений

RoadTypeNo0	Однополосная прямая
RoadTypeNo1	Однополосная левая прямая
RoadTypeNo2	Однополосная средняя прямая
RoadTypeNo3	Перекресток 1 на 1.
RoadTypeNo4	Угол перекрестка 1 на N.
RoadTypeNo5	Угол перекрестка N на N.
RoadTypeNo6	Центральная часть перекрестка
RoadTypeNo7	Т-образный перекресток 1 на 1.
RoadTypeNo8	Левый угол Т-образного перекрестка 1 на N.
RoadTypeNo9	Заблокированный край Т-образного перекрестка N на N.
RoadTypeNo10	Правый угол Т-образного перекрестка 1 на N.
RoadTypeNo11	Расширение дороги
RoadTypeNo12	Сужение дороги
RoadTypeNo13	Однополосный поворот
RoadTypeNo14	Внешний ряд поворота
RoadTypeNo15	Средний ряд поворота
RoadTypeNo16	Внутренний ряд поворота
RoadTypeNo17	Обочина поворота (угла перекрестка)

6.1.3.9 SignalFileId

enum rtm::SignalFileId

Индексы, начиная с которых начинаются текстуры сигналоа определенного типа

Элементы перечислений

ForwardSignalId	Индекс сигнала для движения вперед
LeftwardSignalId	Индекс сигнал для поворота налево
RightwardSignalId	Индекс сигнал для поворота направо

6.1.4 Функции

6.1.4.1 CheckCollisions()

```
void rtm::CheckCollisions (
            WorldController *const world )
```

Функция для вычисления столкновений в мире

Аргументы

world	контроллер мира, в котором будут происходить вычисления

6.1.4.2 SameCoordinates()

```
bool rtm::SameCoordinates (
            float a,
            float b,
            float delta = COORD_DELTA )
```

Функция для сравнения двух координат с определенной точностью

Аргументы

a,b	координаты, которые будут сравниваться
delta	максимальная разность между координатами

Возвращает

результат сравнения

6.1.4.3 RoundCoordinate()

```
float rtm::RoundCoordinate (
            float coordinate,
            float delta = COORD_DELTA )
```

Функция пытается округлить координаты до центра клетки

Аргументы

coordinate	координата, которую будем пытаться округлить
delta	максимальное расстояние до центра клетки

Возвращает

если координата достаточно близка к центру, то координаты центра, иначе саму координату

6.1.4.4 RoundToCenter()

```
float rtm::RoundToCenter (
        float coordinate )
```

Функция для округления координаты до центра клетки

Аргументы

coordinate	округляемая координата

Возвращает

координата ближайшего центра клетки

6.1.4.5 InCenter()

```
bool rtm::InCenter (
        float coordinate,
        float delta = COORD_DELTA )
```

Функция для проверки координаты на центральность

Аргументы

coordinate	координата, которую проверяем
delta	максимальное расстояние до центра клетки

Возвращает

true, если в центре клетки, иначе false

6.1.4.6 DistanceToNextCenter()

```
float rtm::DistanceToNextCenter (
                float x,
                float y,
                float angle )
```

Функция для нахождения расстояния до следующего центра клетка по ходу движения

Аргументы

x,y	координаты объекта
angle	направление движения (угол)

Возвращает

расстояние до центра

6.1.4.7 CenterIsCrossed()

```
bool rtm::CenterIsCrossed (
                float x,
                float y,
                float angle,
                float lastDelta )
```

Функция проверяет, пересек ли объект центр клетки (центральную линию, перпендикулярную направлению движения)

Аргументы

x,y	координаты объекта
angle	направление движения (угол)
lastDelta	расстояние, которое объект прошёл за последнее перемещение

Возвращает

true, если пересек какой-либо центр

6.1.4.8 SameAngles()

```
bool rtm::SameAngles (
                float a,
                float b,
                float delta = ANGLE_DELTA )
```

Фукнция для сравнения двух углов

Аргументы

a,b	углы, которые будут сравниваться
delta	максимальная разность между углами

Возвращает

результат сравнения

6.1.4.9 RoundAngle()

```
float rtm::RoundAngle (
            float angle,
            float delta = ANGLE_DELTA )
```

Фукнция пытается округлить угол до одно из главных направлений (период $\pi/4$, т.е. 0, $\pi/4$, $\pi/2$, ...)

Аргументы

angle	угол, который будем пытаться округлить
delta	максимальная разность между исходным углом и округленным углом

Возвращает

округленный угол, если исходный был достаточно близок, иначе исходный угол

6.1.4.10 NormalizeAngle()

```
float rtm::NormalizeAngle (
            float angle )
```

Функция для нормализации угла до диапазона [-$\pi/2$;$\pi/2$)

Аргументы

angle	угол, который будем нормализовывать

Возвращает

нормализованный угол

6.1.4.11 PixelToCell()

int rtm::PixelToCell (
float coordinate)

Функция для конвертации координаты в номер ячейки

Аргументы

coordinate	координата, которая будет конвертирована

Возвращает

номер ячейки

6.1.4.12 CellToPixel()

float rtm::CellToPixel (
int cellNumber)

Функция для конвертации номера ячейки в координату центра

Аргументы

cellNumber	номер ячейки, который будет конвертирован

Возвращает

координата центра

6.1.4.13 AngleToAngleType()

rtm::AngleType rtm::AngleToAngleType (
float angle)

Функция для конвертации угла в угловой тип

Аргументы

angle	угол, который будет конвертирован

Возвращает

соответствующий угловой тип

6.1.4.14 AngleToDirection()

rtm::DirectionType rtm::AngleToDirection (
float angle)

Функция для конвертации угла в напраление

Аргументы

angle	угол, который будет конвертирован

Возвращает

соответствующее напраление

6.1.4.15 AngleTypeToAngle()

float rtm::AngleTypeToAngle (
AngleType angle)

Функция для конвертации углового типа в угол

Аргументы

angle	уголовой тип, который будет конвертирован

Возвращает

соответствующий угол

6.1.4.16 AngleTypeToDirection()

rtm::DirectionType rtm::AngleTypeToDirection (
AngleType angle)

Функция для конвертации углового типа в направление

Аргументы

angle	уголовой тип, который будет конвертирован

Возвращает

соответствующее направление

6.1.4.17 DirectionToAngle()

float rtm::DirectionToAngle (
DirectionType direction)

Функция для конвертации направления в угол

Аргументы

direction	направление, которое будет конвертировано

Возвращает

соответствующий угол

6.1.4.18 DirectionToAngleType()

rtm::AngleType rtm::DirectionToAngleType (
DirectionType direction)

Функция для конвертации направления в угловой тип

Аргументы

direction	направление, которое будет конвертировано

Возвращает

соответствующий угловой тип

6.1.4.19 GetFilename()

std::string rtm::GetFilename (
std::string const & mask,
size_t number)

Функция для получения пути к файлу по маске

Аргументы

mask	маска названия файла
number	номер объекта

Возвращает

путь к файлу

6.1.4.20 SumAngleTypes()

```
rtm::AngleType rtm::SumAngleTypes (
                AngleType a,
                AngleType b )
```

Функция для суммирования двух угловых типов

Аргументы

a,b	угловые типы, которые будут складываться

Возвращает

сумма угловых типов (a + b)

6.1.4.21 CountDeceleration()

```
float rtm::CountDeceleration (
                float maxSpeed )
```

Функция для подсчёта рекомендуемого коэффициента замедления транспорта

Аргументы

maxSpeed	максимальная скорость транспорта

Возвращает

рекомендуемый коэффициент замедления

6.1.5 Переменные

6.1.5.1 NEAR_DELTA

float const rtm::NEAR_DELTA { 1.f }

Максимальное расстояние до объектов, которые недалеко

См. также

DynamicObject

6.1.5.2 DEFAULT_CROSSROAD_SIGNALS

CrossroadSignals const rtm::DEFAULT_CROSSROAD_SIGNALS

Инициализатор

```
= {
    DEFAULT_DIRECTIONS_SIGNALS
    , DEFAULT_DIRECTIONS_SIGNALS
    , DEFAULT_DIRECTIONS_SIGNALS
    , DEFAULT_DIRECTIONS_SIGNALS
  }
```

Значения по умолчанию для массива сигналов всего перекрестка (светофора на перекрестке нет)

6.1.5.3 DEFAULT_DIRECTIONS_SIGNAL_SPRITES

DirectionsSignalSprites const rtm::DEFAULT_DIRECTIONS_SIGNAL_SPRITES

Инициализатор

```
= {
    DEFAULT_SIGNALS_SPRITES
    , DEFAULT_SIGNALS_SPRITES
    , DEFAULT_SIGNALS_SPRITES
    , DEFAULT_SIGNALS_SPRITES
  }
```

Пустой массив текстур сигналов для всего перекрестка

6.1.5.4 COATING_INDEXES

std::array<size_t, 2> const rtm::COATING_INDEXES

Инициализатор

```
= {
      1
    , 21
  }
```

Индексы, начиная с которых начинаются текстуры покрытий определенного типа

6.1.5.5 ROADS_RESISTANCES

std::array<float, 2> const rtm::ROADS_RESISTANCES

Инициализатор

```
= {
      1.f
    , 0.8f
  }
```

Массив коэффициентов трения для каждого типа объекта

См. также

CoatingType

6.1.5.6 ROADS_DIRECTIONS

std::array<Directions, 18> const rtm::ROADS_DIRECTIONS

Инициализатор

```
= {
       Directions{ true, false, true, false, false, false, false, false }
     , Directions{ true, false, true, false, true, true, false, false }
     , Directions{ true, false, true, false, true, true, true, true }
     , Directions{ true, true, true, true, false, false, false, false }
     , Directions{ true, true, true, true, false, false, false, false }
     , Directions{ true, true, true, true, false, false, false, false }
     , Directions{ true, true, true, true, false, false, false, false }
     , Directions{ false, true, true, true, false, false, false, false }
     , Directions{ false, true, true, true, false, false, false, false }
     , Directions{ false, true, true, true, false, false, false, false }
     , Directions{ false, true, true, true, false, false, false, false }
     , Directions{ true, false, false, false, false, false, true, true }
     , Directions{ false, false, true, false, false, false, true, true }
     , Directions{ false, true, true, false, false, false, false, false }
     , Directions{ false, true, true, false, false, false, false, false }
     , Directions{ false, true, true, false, false, false, false, false }
     , Directions{ false, true, true, false, false, false, false, false }
     , Directions{ false, false, false, false, false, false, false, false }
  }
```

Массив возможных направлений для каждой типа кучоска дороги

См. также

RoadCoating

6.1.5.7 CARS_MAX_SPEEDS

std::array<float, 6> const rtm::CARS_MAX_SPEEDS

Инициализатор

```
= {
        0.f
      , 21.f
      , 24.f
      , 30.f
      , 33.f
      , 36.f
  }
```

Массив максимальных скоростей для машин

См. также

CarObject

6.1.5.8 CARS_ACCELERATIONS

std::array<float, 6> const rtm::CARS_ACCELERATIONS

Инициализатор

```
= {
        0.f
      , 3.f
      , 4.f
      , 6.f
      , 8.25f
      , 12.f
  }
```

Массив ускорений для машин

См. также

CarObject

Глава 7

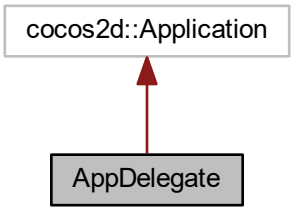
Классы

7.1 Класс AppDelegate

Приложение, основанное на Cocos2d.

#include <AppDelegate.h>

Граф наследования:AppDelegate:



Открытые члены

- AppDelegate ()
  Конструктор по умолчанию
- virtual ~AppDelegate ()
  Деструктор
- virtual void initGLContextAttrs ()
  Функция для установки атрибутов OpenGL (красный, зеленый, синий, альфа-канал...)
- virtual bool applicationDidFinishLaunching ()
  Функция для инициализации Director'а и Scene'ы
- virtual void applicationDidEnterBackground ()
  Функция вызывается, когда приложение скрывается
- virtual void applicationWillEnterForeground ()
  Функция вызывается при первом запуске приложения

7.1.1 Подробное описание

Приложение, основанное на Cocos2d.

7.1.2 Методы

7.1.2.1 applicationDidFinishLaunching()

bool AppDelegate::applicationDidFinishLaunching ( ) [virtual]

Функция для инициализации Director'a и Scene'ы

Возвращает

true Инициализация успешна, приложение продолжает выполняться
false Инициализация провалилась, приложение закроется

Объявления и описания членов классов находятся в файлах:

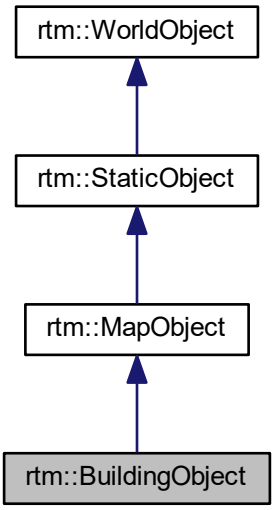
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/AppDelegate.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/AppDelegate.cpp

7.2 Класс rtm::BuildingObject

Класс, описывающий строения (здания)

#include <BuildingObject.h>

Граф наследования:rtm::BuildingObject:



Открытые члены

- BuildingObject ()
  Конструктор по умочанию
- BuildingObject (cocos2d::Sprite *const sprite, int column, int row, float angle)
  Конструктор с использованием уже готового спрайта
- BuildingObject (std::string const &filename, int column, int row, float angle)
  Конструктор из файла
- BuildingObject (size_t type, int column, int row, float angle)
  Конструктор стандартного строения
- virtual ~BuildingObject ()=default
  Деструктор по умолчанию

Дополнительные унаследованные члены

7.2.1 Подробное описание

Класс, описывающий строения (здания)

7.2.2 Конструктор(ы)

7.2.2.1 BuildingObject() [1/3]

```
rtm::BuildingObject::BuildingObject (
                cocos2d::Sprite *const sprite,
                int column,
                int row,
                float angle )
```

Конструктор с использованием уже готового спрайта

Аргументы

sprite	указатель на готовый спрайт
column	колонка, в которой необходимо отрисовать строение
row	строка, в которой необходимо отрисовать строение
angle	угол поворота строения

7.2.2.2 BuildingObject() [2/3]

```
rtm::BuildingObject::BuildingObject (
                std::string const & filename,
```

```
                int column,
                int row,
                float angle )
```

Конструктор из файла

Аргументы

filename	путь к файлу инициализации
column	колонка, в которой необходимо отрисовать строение
row	строка, в которой необходимо отрисовать строение
angle	угол поворота строения

7.2.2.3 BuildingObject() [3/3]

```
rtm::BuildingObject::BuildingObject (
                size_t type,
                int column,
                int row,
                float angle )
```

Конструктор стандартного строения

Аргументы

type	стандартный тип строения
column	колонка, в которой необходимо отрисовать строение
row	строка, в которой необходимо отрисовать строение
angle	угол поворота строения

Объявления и описания членов классов находятся в файлах:

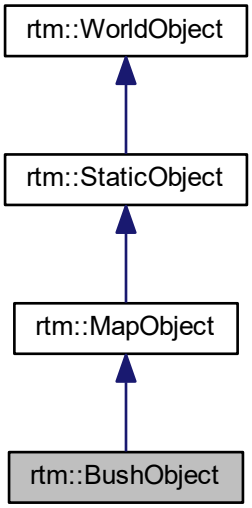
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/BuildingObject.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/BuildingObject.cpp

7.3 Класс rtm::BushObject

Класс, описывающий кусты

#include <BushObject.h>

Граф наследования:rtm::BushObject:



Открытые члены

- BushObject ()
  Конструктор по умочанию
- BushObject (cocos2d::Sprite *const sprite, int column, int row, float angle)
  Конструктор с использованием уже готового спрайта
- BushObject (std::string const &filename, int column, int row, float angle)
  Конструктор из файла
- BushObject (size_t type, int column, int row, float angle)
  Конструктор стандартного куста
- virtual ~BushObject ()=default
  Деструктор по умолчанию

Дополнительные унаследованные члены

7.3.1 Подробное описание

Класс, описывающий кусты

7.3.2 Конструктор(ы)

7.3.2.1 BushObject() [1/3]

```
rtm::BushObject::BushObject (
                cocos2d::Sprite *const sprite,
                int column,
                int row,
                float angle )
```

Конструктор с использованием уже готового спрайта

Аргументы

sprite	указатель на готовый спрайт
column	колонка, в которой необходимо отрисовать куст
row	строка, в которой необходимо отрисовать куст
angle	угол поворота куста

7.3.2.2 BushObject() [2/3]

```
rtm::BushObject::BushObject (
                std::string const & filename,
                int column,
                int row,
                float angle )
```

Конструктор из файла

Аргументы

filename	путь к файлу инициализации
column	колонка, в которой необходимо отрисовать куст
row	строка, в которой необходимо отрисовать куст
angle	угол поворота куста

7.3.2.3 BushObject() [3/3]

```
rtm::BushObject::BushObject (
                size_t type,
                int column,
                int row,
                float angle )
```

Конструктор стандартного куста

Аргументы

type	стандартный тип куста
column	колонка, в которой необходимо отрисовать куст
row	строка, в которой необходимо отрисовать куст
angle	угол поворота куста

Объявления и описания членов классов находятся в файлах:

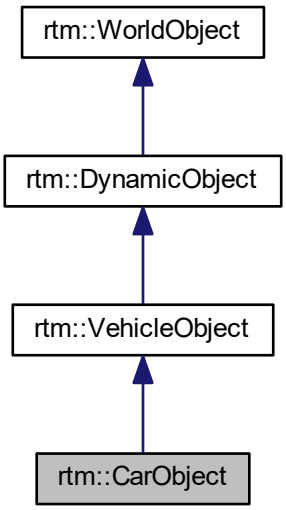
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/BushObject.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/BushObject.cpp

7.4 Класс rtm::CarObject

Класс, описывающий машины

#include <CarObject.h>

Граф наследования:rtm::CarObject:



Открытые члены

- CarObject ()

  Конструктор по умочанию
- CarObject (cocos2d::Sprite *const sprite, int column, int row, float angle, float maxSpeed, float acceleration)

  Конструктор с использованием уже готового спрайта

- CarObject (std::string const &filename, int column, int row, float angle, float maxSpeed, float acceleration)

  Конструктор из файла
- CarObject (size_t type, int column, int row, float angle)

  Конструктор стандартной машины
- virtual ~CarObject ()=default

  Деструктор по умолчанию

Защищенные члены

- virtual bool MovementStart_ (WorldController *const world) override

  Функция, которая просто пропускает выполнение родителя
- virtual bool MovementTick_ (WorldController *const world) override

  Функция для вычисления скорости
- virtual bool MovementEnd_ (WorldController *const world) override

  Функция обнуляет финальную скорость
- virtual bool LineChangingStart (WorldController *const world) override

  Функция, описывающая движение перед перестроением

Закрытые члены

- void SetDesiredSpeed_ (float speed)

  Функция для установки желаемой скорости
- void ResetDesiredSpeed_ ()

  Функция для сброса желаемой скорости
- void CheckCoatingAhead_ (WorldController *const world)

  Функция для проверки объекта (повороты и т.д.)
- void CheckCoatingUnionAhead_ (WorldController *const world)

  Функция для проверки объединения покрытий (заранее тормозим перед светофорами и т.д.)
- void CheckRoadAhead_ (WorldController *const world)

  Функция для проверки дороги спереди (принятие решений)

Закрытые статические члены

- static float GetClassMaxSpeed_ (size_t id)

  Функция для получения максимальной скорости стандартной машины по номеру
- static float GetClassAcceleration_ (size_t id)

  Функция для получения ускорения стандартной машины по номеру

Закрытые данные

- float recommendedSpeed_

  Рекомендованная скорость
- float desiredSpeed_

  Желаемая скорость (приоритетнее рекомендуемой)
- bool hasDesiredSpeed_

  Задана ли желаемая скорость
- bool isTurnNear_

  Далеко ли следующий поворот
- bool isRightTurn_

  Напрвление следующего поворота
- bool waitForSignal_

  Происходит ли сейчас ожидание сигнала светофора
- bool waitForTurn_

  Происходит ли сейчас ожидание освобождения нерегулируемого перекрестка
- AngleType desiredDirection_

  Желаемое направление движения (при первой возможности машина повернет)

7.4.1 Подробное описание

Класс, описывающий машины

7.4.2 Конструктор(ы)

7.4.2.1 CarObject() [1/3]

```
rtm::CarObject::CarObject (
                cocos2d::Sprite *const sprite,
                int column,
                int row,
                float angle,
                float maxSpeed,
                float acceleration )
```

Конструктор с использованием уже готового спрайта

Аргументы

sprite	указатель на готовый спрайт
column	колонка, в которой необходимо отрисовать машину
row	строка, в которой необходимо отрисовать машину
angle	угол поворота машины
maxSpeed	максимальная скорость машины
acceleration	ускорение машины

7.4.2.2 CarObject() [2/3]

```
rtm::CarObject::CarObject (
                std::string const & filename,
                int column,
                int row,
                float angle,
                float maxSpeed,
                float acceleration )
```

Конструктор из файла

Аргументы

filename	путь к файлу инициализации
column	колонка, в которой необходимо отрисовать машину
row	строка, в которой необходимо отрисовать машину
angle	угол поворота машины
maxSpeed	максимальная скорость машины
acceleration	ускорение машины

7.4.2.3 CarObject() [3/3]

```
rtm::CarObject::CarObject (
                size_t type,
                int column,
                int row,
                float angle )
```

Конструктор стандартной машины

Аргументы

type	стандартный тип машины
column	колонка, в которой необходимо отрисовать машину
row	строка, в которой необходимо отрисовать машину
angle	угол поворота машины

7.4.3 Методы

7.4.3.1 MovementStart_()

bool rtm::CarObject::MovementStart_ (
WorldController *const world) [override], [protected], [virtual]

Функция, которая просто пропускает выполнение родителя

Аргументы

world	контроллер мира, в котором находится объект

Переопределяет метод предка rtm::VehicleObject.

7.4.3.2 MovementTick_()

bool rtm::CarObject::MovementTick_ (
WorldController *const world) [override], [protected], [virtual]

Функция для вычисления скорости

Аргументы

world	контроллер мира, в котором находится объект

Переопределяет метод предка rtm::VehicleObject.

7.4.3.3 MovementEnd_()

bool rtm::CarObject::MovementEnd_ (
WorldController *const world) [override], [protected], [virtual]

Функция обнуляет финальную скорость

Аргументы

world	контроллер мира, в котором находится объект

Переопределяет метод предка rtm::VehicleObject.

7.4.3.4 LineChangingStart()

bool rtm::CarObject::LineChangingStart (
WorldController *const world) [override], [protected], [virtual]

Функция, описывающая движение перед перестроением

Аргументы

world	контроллер мира, в котором находится объект

Переопределяет метод предка rtm::VehicleObject.

7.4.3.5 CheckCoatingAhead_()

void rtm::CarObject::CheckCoatingAhead_ (
WorldController *const world) [private]

Функция для проверки объекта (повороты и т.д.)

Аргументы

world	контроллер мира, в котором находится объект

7.4.3.6 CheckCoatingUnionAhead_()

void rtm::CarObject::CheckCoatingUnionAhead_ (
WorldController *const world) [private]

Функция для проверки объединения покрытий (заранее тормозим перед светофорами и т.д.)

Аргументы

world	контроллер мира, в котором находится объект

7.4.3.7 CheckRoadAhead_()

void rtm::CarObject::CheckRoadAhead_ (
WorldController *const world) [private]

Функция для проверки дороги спереди (принятие решений)

Аргументы

world	контроллер мира, в котором находится объект

7.4.3.8 GetClassMaxSpeed_()

float rtm::CarObject::GetClassMaxSpeed_ (
size_t id) [static], [private]

Функция для получения максимальной скорости стандартной машины по номеру

Аргументы

id	номер стандартной машины

Возвращает

максимальная скорость машины

7.4.3.9 GetClassAcceleration_()

float rtm::CarObject::GetClassAcceleration_ (
size_t id) [static], [private]

Функция для получения ускорения стандартной машины по номеру

Аргументы

id	номер стандартной машины

Возвращает

ускорение машины

Объявления и описания членов классов находятся в файлах:

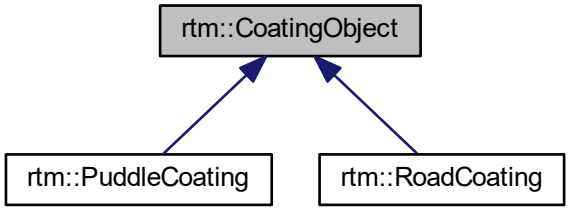
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/CarObject.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/CarObject.cpp

7.5 Класс rtm::CoatingObject

Класс покрытия секции карты

#include <CoatingObject.h>

Граф наследования:rtm::CoatingObject:



Открытые члены

- CoatingObject ()

  Конструктор по умолчанию
- CoatingObject (cocos2d::Sprite *const sprite, int column, int row, AngleType angle, float resistance, Directions directions)

  Конструктор с использованием уже готового спрайта
- CoatingObject (std::string const &filename, int column, int row, AngleType angle, float resistance, Directions directions)

  Конструктор из файла
- virtual ~CoatingObject ()=default

  Деструктор по умолчанию
- cocos2d::Sprite * GetSprite () const

  Функция для получения спрайта
- float GetResistance () const

  Функция для получения коэффициента сопротивления на покрытии
- bool HasDirection (AngleType angle) const

  Функция для проверки существования направления
- bool IsDirectionAvailable (AngleType angle) const

  Функция для проверки разрешенности направления
- void SetDirectionAvailability (AngleType angle, bool status)

  Функция установки разрешенности направления

Защищенные члены

- void SetSprite_ (cocos2d::Sprite *const sprite)

  Функция для установки спрайта

Закрытые члены

- void SetX_ (float x)

  Функция для установки абсциссы
- void SetY_ (float y)

  Функция для установки ординаты

Закрытые данные

- cocos2d::Sprite * sprite_
  Указатель на спрайт
- float x_
  Абсцисса
- float y_
  Ордината
- float resistance_
  Сопротивление на покрытии
- Directions directions_
  Доступные направления
- Directions availableDirections_
  Разрешенные направления

7.5.1 Подробное описание

Класс покрытия секции карты

7.5.2 Конструктор(ы)

7.5.2.1 CoatingObject() [1/2]

```
rtm::CoatingObject::CoatingObject (
                cocos2d::Sprite *const sprite,
                int column,
                int row,
                AngleType angle,
                float resistance,
                Directions directions )
```

Конструктор с использованием уже готового спрайта

Аргументы

sprite	указатель на готовый спрайт
column	колонка, в которой необходимо отрисовать объект
row	строка, в которой необходимо отрисовать объект
angle	угол поворота объекта
resistance	коэффициент сопротивления на покрытии
directions	доступные направления для движения

7.5.2.2 CoatingObject() [2/2]

```
rtm::CoatingObject::CoatingObject (
            std::string const & filename,
            int column,
            int row,
            AngleType angle,
            float resistance,
            Directions directions )
```

Конструктор из файла

Аргументы

filename	путь к файлу инициализации
column	колонка, в которой необходимо отрисовать объект
row	строка, в которой необходимо отрисовать объект
angle	угол поворота объекта
resistance	коэффициент сопротивления на покрытии
directions	доступные направления для движения

7.5.3 Методы

7.5.3.1 GetSprite()

```
cocos2d::Sprite * rtm::CoatingObject::GetSprite ( ) const
```

Функция для получения спрайта

Возвращает

указатель на спрайт

7.5.3.2 GetResistance()

```
float rtm::CoatingObject::GetResistance ( ) const
```

Функция для получения коэффициента сопротивления на покрытии

Возвращает

сопротивление

7.5.3.3 HasDirection()

```
bool rtm::CoatingObject::HasDirection (
            AngleType angle ) const
```

Функция для проверки существования направления

Аргументы

angle	направление

Возвращает

true, если доступно (существует), иначе false

7.5.3.4 IsDirectionAvailable()

```
bool rtm::CoatingObject::IsDirectionAvailable (
            AngleType angle ) const
```

Функция для проверки разрешенности направления

Аргументы

angle	направление

Возвращает

true, если разрешено ехать в данном направлении, иначе false

7.5.3.5 SetDirectionAvailability()

```
void rtm::CoatingObject::SetDirectionAvailability (
            AngleType angle,
            bool status )
```

Функция установки разрешенности направления

Аргументы

angle	направление
status	разрешено ли ехать

7.5.3.6 SetSprite_()

```
void rtm::CoatingObject::SetSprite_ (
            cocos2d::Sprite *const sprite )   [protected]
```

Функция для установки спрайта

Аргументы

sprite	указатель на спрайт

7.5.3.7 SetX_()

void rtm::CoatingObject::SetX_ (
float x) [private]

Функция для установки абсциссы

Аргументы

x	абсцисса

7.5.3.8 SetY_()

void rtm::CoatingObject::SetY_ (
float y) [private]

Функция для установки ординаты

Аргументы

y	ордината

Объявления и описания членов классов находятся в файлах:

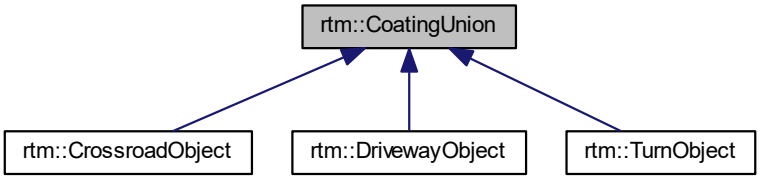
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/CoatingObject.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/CoatingObject.cpp

7.6 Класс rtm::CoatingUnion

Класс объединения покрытий

#include <CoatingUnion.h>

Граф наследования:rtm::CoatingUnion:



Открытые члены

- CoatingUnion ()
  Конструктор по умолчанию
- CoatingUnion (CoatingUnionType type, int column, int row, CoatingMatrix &&objects)
  Конструктор по матрице покрытий
- virtual ~CoatingUnion ()=default
  Деструктор по умолчанию
- CoatingUnionType GetType () const
  Функция для получения типа объединения
- size_t GetWidth () const
  Функция для получения ширины объединения
- size_t GetHeight () const
  Функция для получения высоты объединения
- CoatingUnique const & GetCoatingObject (int column, int row) const
  Функция для получения объекта
- virtual float GetLength () const
  Функция для получения длины (количества покрытий)
- bool IsCorrectColumn (int column) const
  Функция для проверки корректности колонки в данном объединении
- bool IsCorrectRow (int row) const
  Функция для проверки корректности строки в данном объединении
- virtual void ShowSprites (cocos2d::Layer *const layer)
  Функция для добавления спрайтов на сцену
- virtual void ReleaseSprites (cocos2d::Layer *const layer)
  Функция для удаления спрайтов со сцены

Защищенные члены

- int GetColumn_ () const
  Функция для получения левой колонки объединения
- int GetRow_ () const
  Функция для получения нижней строки объединения

Закрытые данные

- CoatingUnionType type_
  Тип объединения покрытий (получившегося элемента)
- int column_
  Левая колонка объединения
- int row_
  Нижняя строка объединения
- size_t width_
  Ширина объединения
- size_t height_
  Высота объединения
- CoatingMatrix objects_
  Матрица объектов покрытий

7.6.1 Подробное описание

Класс объединения покрытий

См. также

CoatingObject

7.6.2 Конструктор(ы)

7.6.2.1 CoatingUnion()

```
rtm::CoatingUnion::CoatingUnion (
                CoatingUnionType type,
                int column,
                int row,
                CoatingMatrix && objects )
```

Конструктор по матрице покрытий

Аргументы

type	тип объединения покрытий (получившегося элемента)
column	левая колонка объединения
row	нижняя строка объединения
objects	матрица объектов покрытий

7.6.3 Методы

7.6.3.1 GetType()

rtm::CoatingUnionType rtm::CoatingUnion::GetType ( ) const

Функция для получения типа объединения

Возвращает

тип объединения

7.6.3.2 GetWidth()

size_t rtm::CoatingUnion::GetWidth ( ) const

Функция для получения ширины объединения

Возвращает

ширина объединения

7.6.3.3 GetHeight()

size_t rtm::CoatingUnion::GetHeight ( ) const

Функция для получения высоты объединения

Возвращает

высота объединения

7.6.3.4 GetCoatingObject()

rtm::CoatingUnique const & rtm::CoatingUnion::GetCoatingObject (
int column,
int row) const

Функция для получения объекта

Аргументы

column	колонка (относительно всей карты), в которой находится объект
row	строка (относительно всей карты), в которой находится объект

Возвращает

умный указатель на объект

7.6.3.5 GetLength()

float rtm::CoatingUnion::GetLength ( ) const [virtual]

Функция для получения длины (количества покрытий)

Возвращает

длина

Переопределяется в rtm::DrivewayObject.

7.6.3.6 IsCorrectColumn()

bool rtm::CoatingUnion::IsCorrectColumn (
int column) const

Функция для проверки корректности колонки в данном объединении

Возвращает

true, если объединение содержит данную колонку, иначе false

7.6.3.7 IsCorrectRow()

bool rtm::CoatingUnion::IsCorrectRow (
int row) const

Функция для проверки корректности строки в данном объединении

Возвращает

true, если объединение содержит данную строку, иначе false

7.6.3.8 ShowSprites()

void rtm::CoatingUnion::ShowSprites (
cocos2d::Layer *const layer) [virtual]

Функция для добавления спрайтов на сцену

Аргументы

layer	слой, на который надо добавить спрайты управляющего блока

Переопределяется в rtm::CrossroadObject.

7.6.3.9 ReleaseSprites()

void rtm::CoatingUnion::ReleaseSprites (
cocos2d::Layer *const layer) [virtual]

Функция для удаления спрайтов со сцены

Аргументы

layer	слой, с которого надо удалить спрайты управляющего блока

Переопределяется в rtm::CrossroadObject.

7.6.3.10 GetColumn_()

int rtm::CoatingUnion::GetColumn_ ( ) const [protected]

Функция для получения левой колонки объединения

Возвращает

левая колонка объединения

7.6.3.11 GetRow_()

int rtm::CoatingUnion::GetRow_ ( ) const [protected]

Функция для получения нижней строки объединения

Возвращает

нижняя строка объединения

Объявления и описания членов классов находятся в файлах:

- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/CoatingUnion.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/CoatingUnion.cpp

7.7 Класс rtm::ControlUnit

Класс управляющего блока (светофор)

#include <ControlUnit.h>

Открытые члены

- ControlUnit ()
  Конструктор по умолчанию
- ControlUnit (size_t type, int column, int row, LinesCounts linesCounts)
  Конструктор управляющего блока перекрестком
- ControlUnit (size_t type, int column, int row, LinesCounts linesCounts, AngleType nullDirection)
  Конструктор управляющего блока перекрестком
- virtual ~ControlUnit ()=default
  Деструктор по умолчанию
- void Update (WorldController *const world)
  Функция обновления
- operator bool () const
  Оператор преобразования в логический тип
- SignalType GetSignal (DirectionType from, DirectionType to) const
  Функция для получения сигнала (статуса направления)
- void ShowSprites (cocos2d::Layer *const layer)
  Функция для добавления спрайтов на сцену
- void ReleaseSprites (cocos2d::Layer *const layer)
  Функция для удаления спрайтов со сцены

Закрытые члены

- void InitSignals_ ()
  Функция для инициализации сигналов
- void ResetSprites_ ()
  Функция для отображения спрайтов в зависимости от массива сигналов
- void UpdateSignal_ (size_t i, size_t j, SignalType signal)
  Функция для безопасной установки сигнала (если направление не закрыто)
- void IncState_ ()
  Функция для инкремента номера состояния
- void SetState_ (size_t state)
  Функция для установки номера состояния
- void ResetState_ ()
  Функция для сброса номера состояния

Закрытые данные

- size_t type_
  Номер типа управляющего блока (задает логику)
- int column_
  Левая колонка перекрестка
- int row_
  Нижняя строка перекрестка
- LinesCounts linesCounts_
  Количество полос в каждом направлении
- AngleType nullDirection_
  Сторона, в направлении которой нельзя двигаться
- CrossroadSignals signals_
  Все сигналы управляющего блока
- DirectionsSignalSprites sprites_
  Все спрайты управляющего блока
- float time_
  Время, прошедшее с последней смены сигнала
- size_t state_
  Номер состояния (для последовательного включения сигналов разных направлений)

7.7.1 Подробное описание

Класс управляющего блока (светофор)

7.7.2 Конструктор(ы)

7.7.2.1 ControlUnit() [1/2]

```
rtm::ControlUnit::ControlUnit (
            size_t type,
            int column,
            int row,
            LinesCounts linesCounts )
```

Конструктор управляющего блока перекрестком

Аргументы

type	номер типа управляющего блока
column	левая колонка перекрестка
row	нижняя строка перекрестка
linesCounts	количество полос в каждом направлении

7.7.2.2 ControlUnit() [2/2]

```
rtm::ControlUnit::ControlUnit (
                size_t type,
                int column,
                int row,
                LinesCounts linesCounts,
                AngleType nullDirection )
```

Конструктор управляющего блока перекрестком

Аргументы

type	номер типа управляющего блока
column	левая колонка перекрестка
row	нижняя строка перекрестка
linesCounts	количество полос в каждом направлении
nullDirection	сторона, в направлении которой нельзя двигаться

7.7.3 Методы

7.7.3.1 Update()

```
void rtm::ControlUnit::Update (
                WorldController *const world )
```

Функция обновления

Аргументы

world	контроллер мира, в котором находится объект

7.7.3.2 operator bool()

```
rtm::ControlUnit::operator bool ( ) const
```

Оператор преобразования в логический тип

Возвращает

true, если управляющий блок работает (меняются сигналы), иначе false

7.7.3.3 GetSignal()

```
rtm::SignalType rtm::ControlUnit::GetSignal (
                DirectionType from,
                DirectionType to ) const
```

Функция для получения сигнала (статуса направления)

Аргументы

from	направление, в котором транспорт движется
to	направление, в котором транспорт поедет дальше

Возвращает

сигнал в нужном направлении

7.7.3.4 ShowSprites()

```
void rtm::ControlUnit::ShowSprites (
                cocos2d::Layer *const layer )
```

Функция для добавления спрайтов на сцену

Аргументы

layer	слой, на который надо добавить спрайты управляющего блока

7.7.3.5 ReleaseSprites()

```
void rtm::ControlUnit::ReleaseSprites (
                cocos2d::Layer *const layer )
```

Функция для удаления спрайтов со сцены

Аргументы

layer	слой, с которого надо удалить спрайты управляющего блока

7.7.3.6 UpdateSignal_()

```
void rtm::ControlUnit::UpdateSignal_ (
```

size_t i,
size_t j,
SignalType signal) [private]

Функция для безопасной установки сигнала (если направление не закрыто)

Аргументы

i	индекс массива для исходного направления
j	индекс массива для конечного направления
signal	новый сигнал

7.7.3.7 SetState_()

void rtm::ControlUnit::SetState_ (
size_t state) [private]

Функция для установки номера состояния

Аргументы

state	новый номера состояния

Объявления и описания членов классов находятся в файлах:

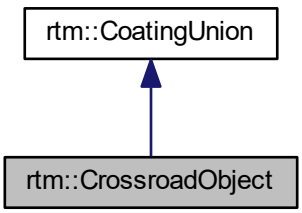
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/ControlUnit.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/ControlUnit.cpp

7.8 Класс rtm::CrossroadObject

Класс пересечения дорог

#include <CrossroadObject.h>

Граф наследования:rtm::CrossroadObject:



Открытые члены

- CrossroadObject ()

  Конструктор по умолчанию
- CrossroadObject (CoatingType type, int column, int row, LinesCounts linesCounts, size_t control↩ UnitType=0)

  Конструктор для обычного перекрестка
- CrossroadObject (CoatingType type, int column, int row, LinesCounts linesCounts, AngleType nullDirection, size_t controlUnitType=0)

  Конструктор для Т-образного перекрестка
- virtual ~CrossroadObject ()=default

  Деструктор по умолчанию
- AngleType GetNullDirection () const

  Функция для получения стороны, в направлении которой нельзя двигаться
- ControlUnitShared GetControlUnit () const

  Функция для получения управляющего блока, привязанного к данному объекту
- virtual void ShowSprites (cocos2d::Layer *const layer) override

  Функция для добавления спрайтов на сцену
- virtual void ReleaseSprites (cocos2d::Layer *const layer) override

  Функция для удаления спрайтов со сцены

Открытые статические члены

- static CoatingMatrix CrossroadMatrix (CoatingType type, int column, int row, LinesCounts lines↩ Counts)

  Функция для получения матрицы покрытий перекрестка
- static CoatingMatrix TCrossroadMatrix (CoatingType type, int column, int row, LinesCounts linesCounts, AngleType nullDirection)

  Функция для получения матрицы покрытий Т-образного перекрестка

Закрытые данные

- LinesCounts linesCounts_

  Количество полос в каждом направлении
- AngleType nullDirection_

  Сторона, в направлении которой нельзя двигаться
- ControlUnitShared controlUnit_

  Умный указатель на управляющий блок

Дополнительные унаследованные члены

7.8.1 Подробное описание

Класс пересечения дорог

7.8.2 Конструктор(ы)

7.8.2.1 CrossroadObject() [1/2]

```
rtm::CrossroadObject::CrossroadObject (
                CoatingType type,
                int column,
                int row,
                LinesCounts linesCounts,
                size_t controlUnitType = 0 )
```

Конструктор для обычного перекрестка

Аргументы

type	тип покрытия
column	левая колонка перекрестка
row	нижняя строка перекрестка
linesCounts	количество полос в каждом направлении
controlUnitType	номер типа управляющего блока

7.8.2.2 CrossroadObject() [2/2]

```
rtm::CrossroadObject::CrossroadObject (
                CoatingType type,
                int column,
                int row,
                LinesCounts linesCounts,
                AngleType nullDirection,
                size_t controlUnitType = 0 )
```

Конструктор для Т-образного перекрестка

Аргументы

type	тип покрытия
column	левая колонка перекрестка
row	нижняя строка перекрестка
linesCounts	количество полос в каждом направлении
nullDirection	сторона, в направлении которой нельзя двигаться
controlUnitType	номер типа управляющего блока

7.8.3 Методы

7.8.3.1 CrossroadMatrix()

```
rtm::CoatingMatrix rtm::CrossroadObject::CrossroadMatrix (
                CoatingType type,
```

int column,
int row,
LinesCounts linesCounts) [static]

Функция для получения матрицы покрытий перекрестка

Аргументы

type	тип покрытия
column	левая колонка перекрестка
row	нижняя строка перекрестка
linesCounts	количество полос в каждом направлении

Возвращает

матрица покрытий

7.8.3.2 TCrossroadMatrix()

rtm::CoatingMatrix rtm::CrossroadObject::TCrossroadMatrix (
CoatingType type,
int column,
int row,
LinesCounts linesCounts,
AngleType nullDirection) [static]

Функция для получения матрицы покрытий T-образного перекрестка

Аргументы

type	тип покрытия
column	левая колонка перекрестка
row	нижняя строка перекрестка
linesCounts	количество полос в каждом направлении
nullDirection	сторона, в направлении которой нельзя двигаться

Возвращает

матрица покрытий

7.8.3.3 GetNullDirection()

rtm::AngleType rtm::CrossroadObject::GetNullDirection ( ) const

Функция для получения стороны, в направлении которой нельзя двигаться

Возвращает

угол, соответствующий запрещенной стороне

7.8.3.4 GetControlUnit()

rtm::ControlUnitShared rtm::CrossroadObject::GetControlUnit ( ) const

Функция для получения управляющего блока, привязанного к данному объекту

Возвращает

умный указатель на управляющий блок

7.8.3.5 ShowSprites()

void rtm::CrossroadObject::ShowSprites (
cocos2d::Layer *const layer) [override], [virtual]

Функция для добавления спрайтов на сцену

Аргументы

layer	слой, на который надо добавить спрайты управляющего блока

Переопределяет метод предка rtm::CoatingUnion.

7.8.3.6 ReleaseSprites()

void rtm::CrossroadObject::ReleaseSprites (
cocos2d::Layer *const layer) [override], [virtual]

Функция для удаления спрайтов со сцены

Аргументы

layer	слой, с которого надо удалить спрайты управляющего блока

Переопределяет метод предка rtm::CoatingUnion.

Объявления и описания членов классов находятся в файлах:

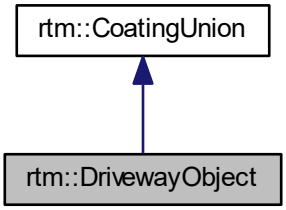
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/CrossroadObject.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/CrossroadObject.cpp

7.9 Класс rtm::DrivewayObject

Класс прямой дороги

#include <DrivewayObject.h>

Граф наследования:rtm::DrivewayObject:



Открытые члены

- DrivewayObject ()

  Конструктор по умолчанию
- DrivewayObject (CoatingType type, int column, int row, size_t width, size_t height, AngleType angle)

  Конструктор по размерам
- virtual ~DrivewayObject ()=default

  Деструктор по умолчанию
- virtual float GetLength () const override

  Функция для получения длины объекта (для вычисления кратчайшего пути)
- size_t GetLinesCount () const

  Функция для получения количества полос
- bool isRightLine (int column, int row) const

  Функция для проверки: находится ли объект в правой полосе
- bool isRightLine (float x, float y) const

  Функция для проверки: находится ли объект в правой полосе
- bool isLeftLine (int column, int row) const

  Функция для проверки: находится ли объект в левой полосе
- bool isLeftLine (float x, float y) const

  Функция для проверки: находится ли объект в левой полосе

Открытые статические члены

- static CoatingMatrix DrivewayMatrix (CoatingType type, int column, int row, size_t width, size_t height, AngleType angle)

  Функция для получения матрицы покрытий дороги

Закрытые статические члены

- static float CountLength_ (size_t width, size_t height, AngleType angle)
  Функция для вычисления длины объекта
- static size_t CountLines_ (size_t width, size_t height, AngleType angle)
  Функция для получения количества полос

Закрытые данные

- AngleType angle_
  Направление движения
- float length_
  Длина объекта (для вычисления кратчайшего пути)
- size_t linesCount_
  Количество полос

Дополнительные унаследованные члены

7.9.1 Подробное описание

Класс прямой дороги

7.9.2 Конструктор(ы)

7.9.2.1 DrivewayObject()

```
rtm::DrivewayObject::DrivewayObject (
                CoatingType type,
                int column,
                int row,
                size_t width,
                size_t height,
                AngleType angle )
```

Конструктор по размерам

Аргументы

type	тип покрытия
column	левая колонка объекта
row	нижняя строка объекта
width	ширина объекта
height	высота объекта
angle	направление движения

7.9.3 Методы

7.9.3.1 DrivewayMatrix()

```
rtm::CoatingMatrix rtm::DrivewayObject::DrivewayMatrix (
                CoatingType type,
                int column,
                int row,
                size_t width,
                size_t height,
                AngleType angle )    [static]
```

Функция для получения матрицы покрытий дороги

Аргументы

type	тип покрытия
column	левая колонка объекта
row	нижняя строка объекта
width	ширина объекта
height	высота объекта
angle	направление движения

Возвращает

матрица покрытий

7.9.3.2 GetLength()

```
float rtm::DrivewayObject::GetLength ( ) const    [override], [virtual]
```

Функция для получения длины объекта (для вычисления кратчайшего пути)

Возвращает

длина

Переопределяет метод предка rtm::CoatingUnion.

7.9.3.3 GetLinesCount()

size_t rtm::DrivewayObject::GetLinesCount ( ) const

Функция для получения количества полос

Возвращает

количество полос

7.9.3.4 isRightLine() [1/2]

bool rtm::DrivewayObject::isRightLine (
int column,
int row) const

Функция для проверки: находится ли объект в правой полосе

Аргументы

column	колонка, в которой находится объект
row	строка, в которой находится объект

Возвращает

true, если объект находится в правой полосе, иначе false

7.9.3.5 isRightLine() [2/2]

bool rtm::DrivewayObject::isRightLine (
float x,
float y) const

Функция для проверки: находится ли объект в правой полосе

Аргументы

x	абсцисса объекта
y	ордината объекта

Возвращает

true, если объект находится в правой полосе, иначе false

7.9.3.6 isLeftLine() [1/2]

```
bool rtm::DrivewayObject::isLeftLine (
                int column,
                int row ) const
```

Функция для проверки: находится ли объект в левой полосе

Аргументы

column	колонка, в которой находится объект
row	строка, в которой находится объект

Возвращает

true, если объект находится в левой полосе, иначе false

7.9.3.7 isLeftLine() [2/2]

```
bool rtm::DrivewayObject::isLeftLine (
                float x,
                float y ) const
```

Функция для проверки: находится ли объект в левой полосе

Аргументы

x	абсцисса объекта
y	ордината объекта

Возвращает

true, если объект находится в левой полосе, иначе false

7.9.3.8 CountLength_()

```
float rtm::DrivewayObject::CountLength_ (
                size_t width,
                size_t height,
                AngleType angle )    [static], [private]
```

Функция для вычисления длины объекта

Аргументы

width	ширина объекта
height	высота объекта
angle	направление движения

Возвращает

длина объекта

7.9.3.9 CountLines_()

```
size_t rtm::DrivewayObject::CountLines_ (
            size_t width,
            size_t height,
            AngleType angle )    [static], [private]
```

Функция для получения количества полос

Аргументы

width	ширина объекта
height	высота объекта
angle	направление движения

Возвращает

количество полос

Объявления и описания членов классов находятся в файлах:

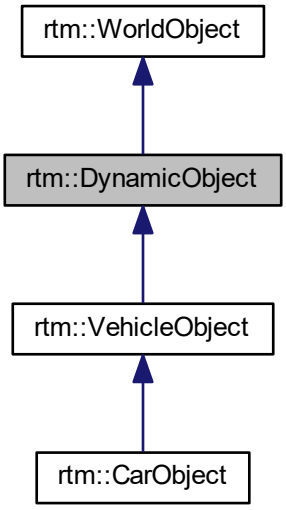
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/DrivewayObject.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/DrivewayObject.cpp

7.10 Класс rtm::DynamicObject

Класс динамического объекта (который двигается, обновляется)

#include <DynamicObject.h>

Граф наследования:rtm::DynamicObject:



Открытые члены

- DynamicObject ()
  Конструктор по умочанию
- DynamicObject (cocos2d::Sprite *sprite, float x, float y, float angle, float speed)
  Конструктор с использованием уже готового спрайта
- DynamicObject (std::string const &filename, float x, float y, float angle, float speed)
  Конструктор из файла
- virtual ~DynamicObject ()=default
  Деструктор по умолчанию
- float GetSpeed () const
  Функция для получения скорости
- float GetLastDelta () const
  Функция для получения последнего приращения положения
- bool HasCollision () const
  Функция для проверки наличия столкновений у данного объекта
- virtual void Update (WorldController *const world)
  Функция обновления
- bool IsNearOthers (WorldController *const world)
  Функция для поиска объектов неподалеку

Защищенные члены

- void SetSpeed_ (float speed)
  Функция для установки скорости
- void SetCollisionFlag_ (bool flag)

Функция для сохранения информации о столкновениях

- bool IsBeholding_ (WorldObject const *const other, float radius=VIEW_RADIUS, float angle=VIEW_ANGLE, float angleShift=VIEW_ANGLE_SHIFT) const

  Функция для проверки попадания объекта в зону видимости
- bool IsIntersecting_ (WorldObject const *const other) const

  Функция для проверки наличия столкновения с other.

Закрытые члены

- bool IsNear_ (WorldObject const *const other) const

  Функция для проверки, находится ли other рядом с данным объектом

Закрытые данные

- float speed_

  Текущая скорость
- float lastDelta_

  Длина последнего смещения
- bool hasCollision_

  Наличие столкновений у данного объекта

Друзья

- void CheckCollisions (WorldController *const world)

  Функция для вычисления столкновений в мире

7.10.1 Подробное описание

Класс динамического объекта (который двигается, обновляется)

7.10.2 Конструктор(ы)

7.10.2.1 DynamicObject() [1/2]

```
rtm::DynamicObject::DynamicObject (
                cocos2d::Sprite * sprite,
                float x,
                float y,
                float angle,
                float speed )
```

Конструктор с использованием уже готового спрайта

Аргументы

sprite	указатель на готовый спрайт
x	абсцисса будущего объекта
y	ордината будущего объекта
angle	угол поворота строения
speed	первоначальная скорость

7.10.2.2 DynamicObject() [2/2]

```
rtm::DynamicObject::DynamicObject (
                std::string const & filename,
                float x,
                float y,
                float angle,
                float speed )
```

Конструктор из файла

Аргументы

filename	путь к файлу инициализации
x	абсцисса будущего объекта
y	ордината будущего объекта
angle	угол поворота строения
speed	первоначальная скорость

7.10.3 Методы

7.10.3.1 GetSpeed()

```
float rtm::DynamicObject::GetSpeed ( ) const
```

Функция для получения скорости

Возвращает

скорость объекта

7.10.3.2 GetLastDelta()

float rtm::DynamicObject::GetLastDelta ( ) const

Функция для получения последнего приращения положения

Возвращает

длина последнего смещения

7.10.3.3 HasCollision()

bool rtm::DynamicObject::HasCollision ( ) const

Функция для проверки наличия столкновений у данного объекта

Возвращает

true, если после последней проверки была столкновение, иначе false

7.10.3.4 Update()

void rtm::DynamicObject::Update (
WorldController *const world) [virtual]

Функция обновления

Аргументы

world	контроллер мира, в котором находится объект

Переопределяется в rtm::VehicleObject.

7.10.3.5 IsNearOthers()

bool rtm::DynamicObject::IsNearOthers (
WorldController *const world)

Функция для поиска объектов неподалеку

Аргументы

world	контроллер мира, в котором находится объект

Возвращает

true, если какой-нибудь объект находится рядом, иначе false

7.10.3.6 SetSpeed_()

```
void rtm::DynamicObject::SetSpeed_ (
            float speed )   [protected]
```

Функция для установки скорости

Аргументы

speed	новая скорость

7.10.3.7 SetCollisionFlag_()

```
void rtm::DynamicObject::SetCollisionFlag_ (
            bool flag )   [protected]
```

Функция для сохранения информации о столкновениях

Аргументы

flag	есть ли столкновение

7.10.3.8 IsBeholding_()

```
bool rtm::DynamicObject::IsBeholding_ (
            WorldObject const *const other,
            float radius = VIEW_RADIUS,
            float angle = VIEW_ANGLE,
            float angleShift = VIEW_ANGLE_SHIFT ) const   [protected]
```

Функция для проверки попадания объекта в зону видимости

Аргументы

other	указатель на второй объект
radius	радиус видимости
angle	угол видимости (в каждую из сторон)
angleShift	сдвиг области видимости

Возвращает

true, если other находится в области видимости данного объекта, иначе false

7.10.3.9 IsIntersecting_ ()

bool rtm::DynamicObject::IsIntersecting_ (
WorldObject const *const other) const [protected]

Функция для проверки наличия столкновения с other.

Аргументы

other	указатель на второй объект

Возвращает

true, если объект пересекается с other, иначе false

7.10.3.10 IsNear_ ()

bool rtm::DynamicObject::IsNear_ (
WorldObject const *const other) const [private]

Функция для проверки, находится ли other рядом с данным объектом

Аргументы

other	указатель на второй объект

Возвращает

true, если other рядом, иначе false

7.10.4 Документация по друзьям класса и функциям, относящимся к классу

7.10.4.1 CheckCollisions

void CheckCollisions (
WorldController *const world) [friend]

Функция для вычисления столкновений в мире

Аргументы

world	контроллер мира, в котором будут происходить вычисления

Объявления и описания членов классов находятся в файлах:

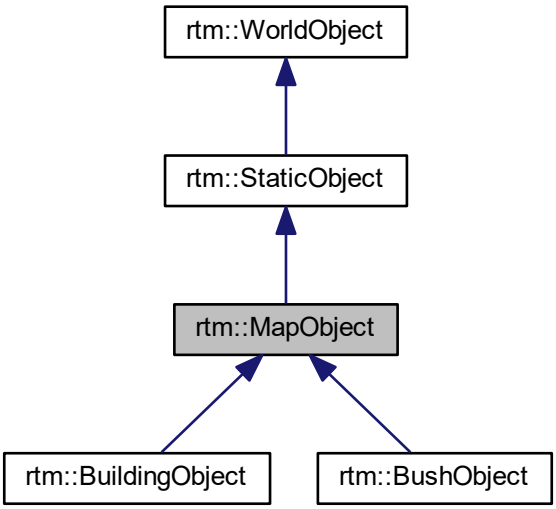
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/DynamicObject.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/DynamicObject.cpp

7.11 Класс rtm::MapObject

Класс статического объекта карты

#include <MapObject.h>

Граф наследования:rtm::MapObject:



Открытые члены

- MapObject ()
  Конструктор по умочанию
- MapObject (cocos2d::Sprite *sprite, int column, int row, float angle)
  Конструктор с использованием уже готового спрайта
- MapObject (std::string const &filename, int column, int row, float angle)
  Конструктор из файла
- virtual ~MapObject ()=default
  Деструктор по умолчанию

Дополнительные унаследованные члены

7.11.1 Подробное описание

Класс статического объекта карты

7.11.2 Конструктор(ы)

7.11.2.1 MapObject() [1/2]

```
rtm::MapObject::MapObject (
            cocos2d::Sprite * sprite,
            int column,
            int row,
            float angle )
```

Конструктор с использованием уже готового спрайта

Аргументы

sprite	указатель на готовый спрайт
column	колонка, в которой необходимо отрисовать объект
row	строка, в которой необходимо отрисовать объект
angle	угол поворота объекта

7.11.2.2 MapObject() [2/2]

```
rtm::MapObject::MapObject (
            std::string const & filename,
            int column,
            int row,
            float angle )
```

Конструктор из файла

Аргументы

filename	путь к файлу инициализации
column	колонка, в которой необходимо отрисовать объект
row	строка, в которой необходимо отрисовать объект
angle	угол поворота объекта

Объявления и описания членов классов находятся в файлах:

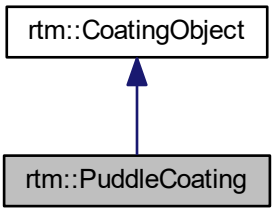
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/MapObject.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/MapObject.cpp

7.12 Класс rtm::PuddleCoating

Класс, описывающий лужи

#include <PuddleCoating.h>

Граф наследования:rtm::PuddleCoating:



Открытые члены

- PuddleCoating ()
  Конструктор по умочанию
- PuddleCoating (cocos2d::Sprite *const sprite, int column, int row, AngleType angle)
  Конструктор с использованием уже готового спрайта
- PuddleCoating (std::string const &filename, int column, int row, AngleType angle)
  Конструктор из файла
- PuddleCoating (size_t id, int column, int row, AngleType angle)
  Конструктор стандартного лужи
- virtual ~PuddleCoating ()=default
  Деструктор по умолчанию

Дополнительные унаследованные члены

7.12.1 Подробное описание

Класс, описывающий лужи

7.12.2 Конструктор(ы)

7.12.2.1 PuddleCoating() [1/3]

```
rtm::PuddleCoating::PuddleCoating (
                cocos2d::Sprite *const sprite,
                int column,
                int row,
                AngleType angle )
```

Конструктор с использованием уже готового спрайта

Аргументы

sprite	указатель на готовый спрайт
column	колонка, в которой необходимо отрисовать лужу
row	строка, в которой необходимо отрисовать лужу
angle	угол поворота лужи

7.12.2.2 PuddleCoating() [2/3]

```
rtm::PuddleCoating::PuddleCoating (
                std::string const & filename,
                int column,
                int row,
                AngleType angle )
```

Конструктор из файла

Аргументы

filename	путь к файлу инициализации
column	колонка, в которой необходимо отрисовать лужу
row	строка, в которой необходимо отрисовать лужу
angle	угол поворота лужи

7.12.2.3 PuddleCoating() [3/3]

```
rtm::PuddleCoating::PuddleCoating (
                size_t id,
                int column,
                int row,
                AngleType angle )
```

Конструктор стандартного лужи

Аргументы

id	номер стандартной лужи
column	колонка, в которой необходимо отрисовать лужу
row	строка, в которой необходимо отрисовать лужу
angle	угол поворота лужи

Объявления и описания членов классов находятся в файлах:

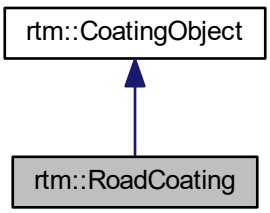
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/PuddleCoating.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/PuddleCoating.cpp

7.13 Класс rtm::RoadCoating

Класс, описывающий дороги

#include <RoadCoating.h>

Граф наследования:rtm::RoadCoating:



Открытые члены

- RoadCoating ()
  Конструктор по умочанию
- RoadCoating (cocos2d::Sprite *const sprite, int column, int row, AngleType angle, float resistance, Directions directions)

Конструктор с использованием уже готового спрайта

- RoadCoating (std::string const &filename, int column, int row, AngleType angle, float resistance, Directions directions)

  Конструктор из файла
- RoadCoating (CoatingType type, size_t id, int column, int row, AngleType angle)

  Конструктор стандартного дороги
- virtual ~RoadCoating ()=default

  Деструктор по умолчанию

Закрытые статические члены

- static float GetClassResistance_ (CoatingType type)

  Функция для получения коэффициента сопротивления на стандартной дороге по номеру
- static Directions const & GetClassDirections_ (size_t id)

  Функция для получения доступных направлений стандартной дороги по номеру

Дополнительные унаследованные члены

7.13.1 Подробное описание

Класс, описывающий дороги

7.13.2 Конструктор(ы)

7.13.2.1 RoadCoating() [1/3]

```
rtm::RoadCoating::RoadCoating (
                cocos2d::Sprite *const sprite,
                int column,
                int row,
                AngleType angle,
                float resistance,
                Directions directions )
```

Конструктор с использованием уже готового спрайта

Аргументы

sprite	указатель на готовый спрайт
column	колонка, в которой необходимо отрисовать дорогу
row	строка, в которой необходимо отрисовать дорогу
angle	угол поворота дороги
resistance	коэффициент сопротивления на дороге
directions	доступные направления для движения

7.13.2.2 RoadCoating() [2/3]

```
rtm::RoadCoating::RoadCoating (
                std::string const & filename,
                int column,
                int row,
                AngleType angle,
                float resistance,
                Directions directions )
```

Конструктор из файла

Аргументы

filename	путь к файлу инициализации
column	колонка, в которой необходимо отрисовать дорогу
row	строка, в которой необходимо отрисовать дорогу
angle	угол поворота дороги
resistance	коэффициент сопротивления на дороге
directions	доступные направления для движения

7.13.2.3 RoadCoating() [3/3]

```
rtm::RoadCoating::RoadCoating (
                CoatingType type,
                size_t id,
                int column,
                int row,
                AngleType angle )
```

Конструктор стандартного дороги

Аргументы

type	стандартный тип покрытия
id	номер стандартной дороги
column	колонка, в которой необходимо отрисовать дорогу
row	строка, в которой необходимо отрисовать дорогу
angle	угол поворота дороги

7.13.3 Методы

7.13.3.1 GetClassResistance_()

float rtm::RoadCoating::GetClassResistance_ (
CoatingType type) [static], [private]

Функция для получения коэффициента сопротивления на стандартной дороге по номеру

Аргументы

type	тип покрытия

Возвращает

сопротивление

7.13.3.2 GetClassDirections_()

rtm::Directions const & rtm::RoadCoating::GetClassDirections_ (
size_t id) [static], [private]

Функция для получения доступных направлений стандартной дороги по номеру

Аргументы

id	номер стандартной дороги

Возвращает

доступные направления

Объявления и описания членов классов находятся в файлах:

- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/RoadCoating.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/RoadCoating.cpp

7.14 Структура rtm::SpawnType

Структура, описывающая параметры точки генерации объектов

#include <General.h>

Открытые атрибуты

- int column
  Номер столбца
- int row
  Номер строки
- float angle
  Первоначальный угол для транспорта

7.14.1 Подробное описание

Структура, описывающая параметры точки генерации объектов

Объявления и описания членов структуры находятся в файле:

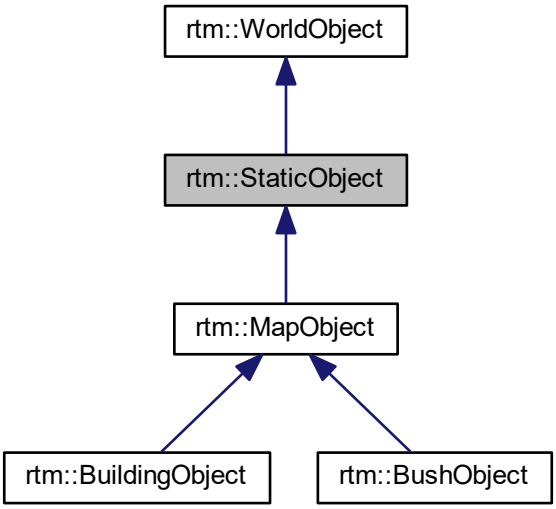
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/General.h

7.15 Класс rtm::StaticObject

Класс статического объекта (который не обновляется)

#include <StaticObject.h>

Граф наследования:rtm::StaticObject:



Открытые члены

- StaticObject ()
  Конструктор по умочанию
- StaticObject (cocos2d::Sprite *sprite, float x, float y, float angle)
  Конструктор с использованием уже готового спрайта
- StaticObject (std::string const &filename, float x, float y, float angle)
  Конструктор из файла
- virtual ~StaticObject ()=default
  Деструктор по умолчанию

Дополнительные унаследованные члены

7.15.1 Подробное описание

Класс статического объекта (который не обновляется)

7.15.2 Конструктор(ы)

7.15.2.1 StaticObject() [1/2]

```
rtm::StaticObject::StaticObject (
                cocos2d::Sprite * sprite,
                float x,
                float y,
                float angle )
```

Конструктор с использованием уже готового спрайта

Аргументы

sprite	указатель на готовый спрайт
x	абсцисса будущего объекта
y	ордината будущего объекта
angle	угол поворота строения

7.15.2.2 StaticObject() [2/2]

```
rtm::StaticObject::StaticObject (
                std::string const & filename,
                float x,
                float y,
                float angle )
```

Конструктор из файла

Аргументы

filename	путь к файлу инициализации
x	абсцисса будущего объекта
y	ордината будущего объекта
angle	угол поворота строения

Объявления и описания членов классов находятся в файлах:

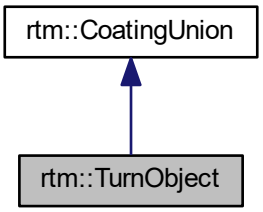
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/StaticObject.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/StaticObject.cpp

7.16 Класс rtm::TurnObject

Класс поворота дороги

#include <TurnObject.h>

Граф наследования:rtm::TurnObject:



Открытые члены

- TurnObject ()

  Конструктор по умолчанию
- TurnObject (bool isRight, CoatingType type, int column, int row, size_t linesCount, AngleType angle)

  Конструктор по размерам
- virtual ~TurnObject ()=default

  Деструктор по умолчанию
- bool IsRight () const

  Функция для получения типа поворота
- AngleType GetAngle () const

  Функция для получения угла поворота объекта

Открытые статические члены

- static CoatingMatrix RightTurnMatrix (CoatingType type, int column, int row, size_t linesCount, AngleType angle)

  Функция для получения матрицы правого поворота
- static CoatingMatrix LeftTurnMatrix (CoatingType type, int column, int row, size_t linesCount, AngleType angle)

  Функция для получения матрицы левого поворота

Закрытые данные

- bool isRight_
  Тип поворота
- AngleType angle_
  Угол поворота объекта

Дополнительные унаследованные члены

7.16.1 Подробное описание

Класс поворота дороги

7.16.2 Конструктор(ы)

7.16.2.1 TurnObject()

```
rtm::TurnObject::TurnObject (
                bool isRight,
                CoatingType type,
                int column,
                int row,
                size_t linesCount,
                AngleType angle )
```

Конструктор по размерам

Аргументы

isRight	тип поворота (false - левый, true - правый)
type	тип покрытия
column	левая колонка объекта
row	нижняя строка объекта
linesCount	количество полос
angle	угол поворота объекта

7.16.3 Методы

7.16.3.1 RightTurnMatrix()

```
rtm::CoatingMatrix rtm::TurnObject::RightTurnMatrix (
                CoatingType type,
```

int column,
int row,
size_t linesCount,
AngleType angle) [static]

Функция для получения матрицы правого поворота

Аргументы

type	тип покрытия
column	левая колонка объекта
row	нижняя строка объекта
linesCount	количество полос
angle	угол поворота объекта

Возвращает

матрица покрытий

7.16.3.2 LeftTurnMatrix()

rtm::CoatingMatrix rtm::TurnObject::LeftTurnMatrix (
CoatingType type,
int column,
int row,
size_t linesCount,
AngleType angle) [static]

Функция для получения матрицы левого поворота

Аргументы

type	тип покрытия
column	левая колонка объекта
row	нижняя строка объекта
linesCount	количество полос
angle	угол поворота объекта

Возвращает

матрица покрытий

7.16.3.3 IsRight()

bool rtm::TurnObject::IsRight ( ) const

Функция для получения типа поворота

Возвращает

true, если правый поворот, false, если левый

7.16.3.4 GetAngle()

rtm::AngleType rtm::TurnObject::GetAngle ( ) const

Функция для получения угла поворота объекта

Возвращает

угол поворота объекта

Объявления и описания членов классов находятся в файлах:

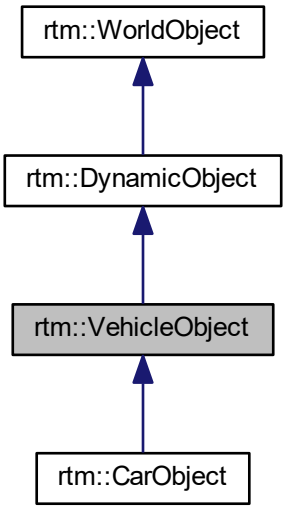
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/TurnObject.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/TurnObject.cpp

7.17 Класс rtm::VehicleObject

Класс транспорта (динамического объекта карты)

#include <VehicleObject.h>

Граф наследования:rtm::VehicleObject:



Открытые члены

- VehicleObject ()

  Конструктор по умочанию
- VehicleObject (cocos2d::Sprite *const sprite, int column, int row, float angle, float maxSpeed, float acceleration, float deceleration)

  Конструктор с использованием уже готового спрайта
- VehicleObject (std::string const &filename, int column, int row, float angle, float maxSpeed, float acceleration, float deceleration)

  Конструктор из файла
- virtual ~VehicleObject ()=default

  Деструктор по умолчанию
- virtual void Update (WorldController *const world) override

  Функция обновления

Защищенные члены

- bool MoveForward_ ()

  Функция для сообщения о необходимости начать движение
- bool Stop_ ()

  Функция для сообщения о необходимости остановиться
- bool Rotate_ (float angle=ANGLE_RIGHT)

  Функция для сообщения о необходимости повернуть
- bool ChangeLine_ (bool isRight=LEFT)

  Функция для сообщения о необходимости перестроиться
- bool IsMovement_ () const

  Функция, сообщающая о движении объекта
- bool IsRotation_ () const

  Функция, сообщающая о повороте объекта
- bool IsLineChanging_ () const

  Функция, сообщающая о перестроении объекта
- bool IsBraking_ () const

  Функция, сообщающая о торможении объекта перед светофором и т.д.
- float GetMaxSpeed_ () const

  Функция для получения максимальной скорости
- float GetFinalSpeed_ () const

  Функция для получения финальной скорости (к которой объект будет стремиться)
- void SetFinalSpeed_ (float speed)

  Функция для установки финальной скорости (к которой объект будет стремиться)
- void SetBrakingFactor_ (float factor)

  Функция для установки коэффициента торможения
- void StopAtDistance_ (float distance)

  Функция для установки тормозого пути (объект будет пытаться остановиться за данную дистанцию)

Зрение у машин

Функции для просмотра окружающего мира

- CoatingObject * CheckForwardCoating_ (WorldController *const world, int delta=1)

  Функция для получения следующего по ходу движения покрытия
- CoatingUnion * CheckForwardCoatingUnion_ (WorldController *const world, int delta=1)

Функция для получения следующего по ходу движения объединения покрытий

- DynamicObject * CheckForwardArea_ (WorldController *const world, float radius, float angle, float angleShift)

  Функция для проверки области видимости спереди
- DynamicObject * CheckMovingArea_ (WorldController *const world)

  Функция для проверки области видимости во время движения по прямой
- DynamicObject * CheckTurnArea_ (WorldController *const world, bool isRight)

  Функция для проверки области видимости перед поворотом
- DynamicObject * CheckRotationArea_ (WorldController *const world)

  Функция для проверки области видимости во время поворота
- DynamicObject * CheckCrossroadArea_ (WorldController *const world)

  Функция для проверки области видимости перед нерегулируемым перекрестком
- DynamicObject * CheckLineChangingArea_ (WorldController *const world)

  Функция для проверки области видимости перед перестроением

Маневры

Функции выполняющиеся во время маневров

- virtual void BeforeMoving_ (WorldController *const world)

  Функция, выполняющаяся непосредственно перед перемещением объекта
- virtual void AfterMoving_ (WorldController *const world)

  Функция, выполняющаяся непосредственно после перемещением объекта
- virtual bool MovementStart_ (WorldController *const world)

  Функция, выполняющаяся перед началом движения
- virtual bool MovementTick_ (WorldController *const world)

  Функция, выполняющаяся во время движения
- virtual bool MovementEnd_ (WorldController *const world)

  Функция, выполняющаяся после движения (перед началом остановки)
- virtual bool RotationStart_ (WorldController *const world)

  Функция, выполняющаяся перед поворотом
- virtual bool RotationTick_ (WorldController *const world)

  Функция, выполняющаяся во время поворота
- virtual bool RotationEnd_ (WorldController *const world)

  Функция, выполняющаяся после поворота
- virtual bool LineChangingStart (WorldController *const world)

  Функция, выполняющаяся перед перестроением
- virtual bool LineChangingTick_ (WorldController *const world)

  Функция, выполняющаяся во время перестроения
- virtual bool LineChangingEnd_ (WorldController *const world)

  Функция, выполняющаяся после перестроения

Закрытые члены

- void LineChanging_ (WorldController *const world)

  Функция, выполняющая различные этапы при перестроении
- void Rotation_ (WorldController *const world)

  Функция, выполняющая различные этапы при повороте
- void Movement_ (WorldController *const world)

  Функция, выполняющая различные этапы при движении
- void SpeedChanging_ (WorldController *const world)

  Функция, выполняющая изменение скорости в зависимости от ускорения, тормозного пути и т.д.
- void SmoothBrakingCounter (WorldController *const world)

  Функция, выполняющая декрементирование тормозного пути (если задан)

Закрытые данные

- StateType isMovement_
  Этап выполнения движения (стоит, движется)
- StateType isRotation_
  Этап выполнения поворота
- StateType isLineChanging_
  Этап выполнения перестроения
- float const maxSpeed_
  Максимальная скорость
- float const acceleration_
  Ускорение
- float const deceleration_
  Скорость замедления
- float finalSpeed_
  Финальная скорость
- float brakingFactor_
  Коэффициент торможения
- float brakingDistance_
  Тормозной путь
- float rotationAngle_
  Угол, на который надо повернуться
- float rotationRadius_
  Радиус окружности, по которой двигается объект
- float remainingOffset_
  Оставшийся перпендикулярный движению сдвиг при перестроении
- float remainingOffsetAngle_
  Направление перестроения (перпендикулярно движению)

7.17.1 Подробное описание

Класс транспорта (динамического объекта карты)

7.17.2 Конструктор(ы)

7.17.2.1 VehicleObject() [1/2]

```
rtm::VehicleObject::VehicleObject (
            cocos2d::Sprite *const sprite,
            int column,
            int row,
            float angle,
            float maxSpeed,
            float acceleration,
            float deceleration )
```

Конструктор с использованием уже готового спрайта

Аргументы

sprite	указатель на готовый спрайт
column	колонка, в которой необходимо отрисовать машину
row	строка, в которой необходимо отрисовать машину
angle	угол поворота машины
maxSpeed	максимальная скорость машины
acceleration	ускорение машины
deceleration	скорость замедления машины

7.17.2.2 VehicleObject() [2/2]

```
rtm::VehicleObject::VehicleObject (
                std::string const & filename,
                int column,
                int row,
                float angle,
                float maxSpeed,
                float acceleration,
                float deceleration )
```

Конструктор из файла

Аргументы

filename	путь к файлу инициализации
column	колонка, в которой необходимо отрисовать машину
row	строка, в которой необходимо отрисовать машину
angle	угол поворота машины
maxSpeed	максимальная скорость машины
acceleration	ускорение машины
deceleration	скорость замедления машины

7.17.3 Методы

7.17.3.1 Update()

```
void rtm::VehicleObject::Update (
                WorldController *const world )   [override], [virtual]
```

Функция обновления

Аргументы

world	контроллер мира, в котором находится объект

Переопределяет метод предка rtm::DynamicObject.

7.17.3.2 MoveForward_()

bool rtm::VehicleObject::MoveForward_ ( ) [protected]

Функция для сообщения о необходимости начать движение

Возвращает

true, если возможно начать движение, иначе false

7.17.3.3 Stop_()

bool rtm::VehicleObject::Stop_ ( ) [protected]

Функция для сообщения о необходимости остановиться

Возвращает

true, если возможно остановиться, иначе false

7.17.3.4 Rotate_()

bool rtm::VehicleObject::Rotate_ (
float angle = ANGLE_RIGHT) [protected]

Функция для сообщения о необходимости повернуть

Возвращает

true, если возможно повернуть, иначе false

7.17.3.5 ChangeLine_()

bool rtm::VehicleObject::ChangeLine_ (
bool isRight = LEFT) [protected]

Функция для сообщения о необходимости перестроиться

Возвращает

true, если возможно перестроиться, иначе false

7.17.3.6 IsMovement_()

bool rtm::VehicleObject::IsMovement_ ( ) const [protected]

Функция, сообщающая о движении объекта

Возвращает

true, если объект движется, иначе false

7.17.3.7 IsRotation_()

bool rtm::VehicleObject::IsRotation_ ( ) const [protected]

Функция, сообщающая о повороте объекта

Возвращает

true, если объект поворачивает, иначе false

7.17.3.8 IsLineChanging_()

bool rtm::VehicleObject::IsLineChanging_ ( ) const [protected]

Функция, сообщающая о перестроении объекта

Возвращает

true, если объект перестраивается, иначе false

7.17.3.9 IsBraking_()

bool rtm::VehicleObject::IsBraking_ ( ) const [protected]

Функция, сообщающая о торможении объекта перед светофором и т.д.

Возвращает

true, если объект тормозит, иначе false

7.17.3.10 GetMaxSpeed_()

float rtm::VehicleObject::GetMaxSpeed_ ( ) const [protected]

Функция для получения максимальной скорости

Возвращает

максимальная скорость

7.17.3.11 GetFinalSpeed_()

float rtm::VehicleObject::GetFinalSpeed_ ( ) const [protected]

Функция для получения финальной скорости (к которой объект будет стремиться)

Возвращает

конечная скорость

7.17.3.12 SetFinalSpeed_()

void rtm::VehicleObject::SetFinalSpeed_ (
float speed) [protected]

Функция для установки финальной скорости (к которой объект будет стремиться)

Аргументы

speed	новая скорость

7.17.3.13 SetBrakingFactor_()

void rtm::VehicleObject::SetBrakingFactor_ (
float factor) [protected]

Функция для установки коэффициента торможения

Аргументы

factor	новый коэффициент торможения (1 - торможение с обычным ускорением)

7.17.3.14 StopAtDistance_()

void rtm::VehicleObject::StopAtDistance_ (
float distance) [protected]

Функция для установки тормозого пути (объект будет пытаться остановиться за данную дистанцию)

Аргументы

distance	дистанция, за которую необходимо остановиться

7.17.3.15 CheckForwardCoating_()

rtm::CoatingObject * rtm::VehicleObject::CheckForwardCoating_ (
WorldController *const world,
int delta = 1) [protected]

Функция для получения следующего по ходу движения покрытия

Аргументы

world	контроллер мира, в котором находится объект
delta	сдвиг в клетках относительно данного объекта (1 - следующая, 2 - через одну)

Возвращает

указатель на покрытие

7.17.3.16 CheckForwardCoatingUnion_()

```
rtm::CoatingUnion * rtm::VehicleObject::CheckForwardCoatingUnion_ (
        WorldController *const world,
        int delta = 1 )  [protected]
```

Функция для получения следующего по ходу движения объединения покрытий

Аргументы

world	контроллер мира, в котором находится объект
delta	сдвиг в клетках относительно данного объекта (1 - следующее, 2 - через одно)

Возвращает

указатель на объединение покрытий

7.17.3.17 CheckForwardArea_()

```
rtm::DynamicObject * rtm::VehicleObject::CheckForwardArea_ (
        WorldController *const world,
        float radius,
        float angle,
        float angleShift )  [protected]
```

Функция для проверки области видимости спереди

Аргументы

world	контроллер мира, в котором находится данный объект
radius	радиус видимости
angle	угол видимости (в каждую из сторон)
angleShift	сдвиг области видимости

Возвращает

указатель на объект, находящийся в области видимости
nullptr, если нет объектов в области видимости

7.17.3.18 CheckMovingArea_()

```
rtm::DynamicObject * rtm::VehicleObject::CheckMovingArea_ (
        WorldController *const world )  [protected]
```

Функция для проверки области видимости во время движения по прямой

Аргументы

world	контроллер мира, в котором находится данный объект

Возвращает

указатель на объект, находящийся в области видимости
nullptr, если нет объектов в области видимости

7.17.3.19 CheckTurnArea_()

```
rtm::DynamicObject * rtm::VehicleObject::CheckTurnArea_ (
        WorldController *const world,
        bool isRight )  [protected]
```

Функция для проверки области видимости перед поворотом

Аргументы

world	контроллер мира, в котором находится данный объект
isRight	сторона, в которую совершается поворот (тип поворота)

Возвращает

указатель на объект, находящийся в области видимости
nullptr, если нет объектов в области видимости

7.17.3.20 CheckRotationArea_()

```
rtm::DynamicObject * rtm::VehicleObject::CheckRotationArea_ (
        WorldController *const world )  [protected]
```

Функция для проверки области видимости во время поворота

Аргументы

world	контроллер мира, в котором находится данный объект

Возвращает

указатель на объект, находящийся в области видимости
nullptr, если нет объектов в области видимости

7.17.3.21 CheckCrossroadArea_ ()

rtm::DynamicObject * rtm::VehicleObject::CheckCrossroadArea_ (
WorldController *const world) [protected]

Функция для проверки области видимости перед нерегулируемым перекрестком

Аргументы

world	контроллер мира, в котором находится данный объект

Возвращает

указатель на объект, находящийся в области видимости
nullptr, если нет объектов в области видимости

7.17.3.22 CheckLineChangingArea_ ()

rtm::DynamicObject * rtm::VehicleObject::CheckLineChangingArea_ (
WorldController *const world) [protected]

Функция для проверки области видимости перед перестроением

Аргументы

world	контроллер мира, в котором находится данный объект

Возвращает

указатель на объект, находящийся в области видимости
nullptr, если нет объектов в области видимости

7.17.3.23 BeforeMoving_ ()

void rtm::VehicleObject::BeforeMoving_ (
WorldController *const world) [protected], [virtual]

Функция, выполняющаяся непосредственно перед перемещением объекта

Аргументы

world	контроллер мира, в котором находится данный объект

7.17.3.24 AfterMoving_()

void rtm::VehicleObject::AfterMoving_ (
WorldController *const world) [protected], [virtual]

Функция, выполняющаяся непосредственно после перемещением объекта

Аргументы

world	контроллер мира, в котором находится данный объект

7.17.3.25 MovementStart_()

bool rtm::VehicleObject::MovementStart_ (
WorldController *const world) [protected], [virtual]

Функция, выполняющаяся перед началом движения

Аргументы

world	контроллер мира, в котором находится данный объект

Возвращает

true, если шаг успешно завершён (в следующий раз выполнится MovementTick)
false, если необходимо повторить этот шаг (в следующий раз выполнится опять эта функция)

Переопределяется в rtm::CarObject.

7.17.3.26 MovementTick_()

bool rtm::VehicleObject::MovementTick_ (
WorldController *const world) [protected], [virtual]

Функция, выполняющаяся во время движения

Аргументы

world	контроллер мира, в котором находится данный объект

Возвращает

true, если шаг успешно завершён (в следующий раз выполнится MovementEnd)
false, если необходимо повторить этот шаг (в следующий раз выполнится опять эта функция)

Переопределяется в rtm::CarObject.

7.17.3.27 MovementEnd_()

bool rtm::VehicleObject::MovementEnd_ (
WorldController *const world) [protected], [virtual]

Функция, выполняющаяся после движения (перед началом остановки)

Аргументы

world	контроллер мира, в котором находится данный объект

Возвращает

true, если шаг успешно завершён (движение на этом закончится)
false, если необходимо повторить этот шаг (в следующий раз выполнится опять эта функция)

Переопределяется в rtm::CarObject.

7.17.3.28 RotationStart_()

bool rtm::VehicleObject::RotationStart_ (
WorldController *const world) [protected], [virtual]

Функция, выполняющаяся перед поворотом

Аргументы

world	контроллер мира, в котором находится данный объект

Возвращает

true, если шаг успешно завершён (в следующий раз выполнится RotationTick)
false, если необходимо повторить этот шаг (в следующий раз выполнится опять эта функция)

7.17.3.29 RotationTick_()

bool rtm::VehicleObject::RotationTick_ (
WorldController *const world) [protected], [virtual]

Функция, выполняющаяся во время поворота

Аргументы

world	контроллер мира, в котором находится данный объект

Возвращает

true, если шаг успешно завершён (в следующий раз выполнится RotationEnd)
false, если необходимо повторить этот шаг (в следующий раз выполнится опять эта функция)

7.17.3.30 RotationEnd_()

bool rtm::VehicleObject::RotationEnd_ (
WorldController *const world) [protected], [virtual]

Функция, выполняющаяся после поворота

Аргументы

world	контроллер мира, в котором находится данный объект

Возвращает

true, если шаг успешно завершён (поворот на этом закончится)
false, если необходимо повторить этот шаг (в следующий раз выполнится опять эта функция)

7.17.3.31 LineChangingStart()

bool rtm::VehicleObject::LineChangingStart (
WorldController *const world) [protected], [virtual]

Функция, выполняющаяся перед перестроением

Аргументы

world	контроллер мира, в котором находится данный объект

Возвращает

true, если шаг успешно завершён (в следующий раз выполнится LineChangingTick)
false, если необходимо повторить этот шаг (в следующий раз выполнится опять эта функция)

Переопределяется в rtm::CarObject.

7.17.3.32 LineChangingTick_()

bool rtm::VehicleObject::LineChangingTick_ (
WorldController *const world) [protected], [virtual]

Функция, выполняющаяся во время перестроения

Аргументы

world	контроллер мира, в котором находится данный объект

Возвращает

true, если шаг успешно завершён (в следующий раз выполнится LineChangingEnd)
false, если необходимо повторить этот шаг (в следующий раз выполнится опять эта функция)

7.17.3.33 LineChangingEnd_()

bool rtm::VehicleObject::LineChangingEnd_ (
WorldController *const world) [protected], [virtual]

Функция, выполняющаяся после перестроения

Аргументы

world	контроллер мира, в котором находится данный объект

Возвращает

true, если шаг успешно завершён (перестроение на этом закончится)
false, если необходимо повторить этот шаг (в следующий раз выполнится опять эта функция)

7.17.3.34 LineChanging_()

void rtm::VehicleObject::LineChanging_ (
WorldController *const world) [private]

Функция, выполняющая различные этапы при перестроении

Аргументы

world	контроллер мира, в котором находится данный объект

7.17.3.35 Rotation_()

void rtm::VehicleObject::Rotation_ (
WorldController *const world) [private]

Функция, выполняющая различные этапы при повороте

Аргументы

world	контроллер мира, в котором находится данный объект

7.17.3.36 Movement_()

void rtm::VehicleObject::Movement_ (
WorldController *const world) [private]

Функция, выполняющая различные этапы при движении

Аргументы

world	контроллер мира, в котором находится данный объект

7.17.3.37 SpeedChanging_()

void rtm::VehicleObject::SpeedChanging_ (
WorldController *const world) [private]

Функция, выполняющая изменение скорости в зависимости от ускорения, тормозного пути и т.д.

Аргументы

world	контроллер мира, в котором находится данный объект

7.17.3.38 SmoothBrakingCounter()

void rtm::VehicleObject::SmoothBrakingCounter (
WorldController *const world) [private]

Функция, выполняющая декрементирование тормозного пути (если задан)

Аргументы

world	контроллер мира, в котором находится данный объект

Объявления и описания членов классов находятся в файлах:

- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/VehicleObject.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/VehicleObject.cpp

7.18 Класс rtm::WorldController

Класс контроллера карты, связующее звено всех объектов

#include <WorldController.h>

Открытые члены

- WorldController ()
  Конструктор по умолчанию
- WorldController (WorldScene *const scene)
  Конструктор без загрузки какой-либо карты
- WorldController (WorldScene *const scene, std::string const &filename)
  Конструктор с загрузкой карты
- WorldController (WorldScene *const scene, size_t mapNumber)
  Конструктор с загрузкой карты
- void Update (float time)
  Функция обновления
- cocos2d::Layer * GetLayer () const
  Функция для получения основного слоя сцены (на нём вся движуха)
- size_t GetColumnsCount () const
  Функция для получения количества столбцов карты
- size_t GetRowsCount () const
  Функция для получения количества строк карты
- float GetDeltaTime () const
  Функция для получения последней разницей между обновлениями
- float GetTimeFactor () const
  Функция для получения коэффициента ускорения времени
- CoatingObject * GetCoatingObject (int column, int row)
  Функция для получения объекта в определенной клетке
- CoatingUnion * GetCoatingUnion (int column, int row)
  Функция для получения объединения покрытий в определенной клетке
- StaticObject * GetStaticObject (int column, int row)
  Функция для получения статического объекта в определенной клетке
- std::vector< DynamicShared > & GetDynamicObjects ()
  Функция для получения массива движущихся объектов
- bool IsPause ()
  Функция сообщает, происходят ли обновления
- bool IsCorrectColumn (int column)

Функция проверяет корректность столбца

- bool IsCorrectRow (int row)

  Функция проверяет корректность строки
- bool IsAllowableColumn (int column)

  Функция проверяет, можно ли двигаться в столбце
- bool IsAllowableRow (int row)

  Функция проверяет, можно ли двигаться в строке
- bool IsVisibleColumn (int column)

  Функция проверяет видимость столбца
- bool IsVisibleRow (int row)

  Функция проверяет видимость строки
- void SetTimeFactor (float factor)

  Функция для установки коэффициента ускорения времени
- bool LoadMap (std::string const &filename)

  Функция для загрузки карты из файла
- bool LoadMap (size_t number)

  Функция для загрузки карты по номеру
- void SpawnCar ()

  Функция для добавления машины на карту
- void RemoveAccidents ()

  Функция для удаления аварий
- void RemoveVehicles ()

  Функция для удаления всего транспорта
- void Play ()

  Функция для продолжения выполнения обновлений
- void Pause ()

  Функция для временной остановки обновлений
- void Reset ()

  Функция для перезагрузки карты

Закрытые члены

- bool IsEmpty_ (int column, int row, size_t width=1, size_t height=1)

  Функция проверяет доступность зоны для генерации статических объектов и объектов объекта
- bool GenerateObject_ (uint8_t *params, uint8_t count)

  Функция для парсинга параметров и генерации объектов
- bool AddCoatingUnion_ (int column, int row, CoatingUnionShared coatingUnion)

  Функция для генерации объединения покрытий
- bool AddDriveway_ (CoatingType type, int column, int row, size_t width, size_t height, AngleType angle)

  Функция для генерации прямой односторонней дороги
- bool AddCrossroad_ (CoatingType type, int column, int row, LinesCounts linesCounts, size_t controlUnitType=0)

  Функция для генерации перекрестка
- bool AddTCrossroad_ (CoatingType type, int column, int row, LinesCounts linesCounts, AngleType nullDirection, size_t controlUnitType=0)

  Функция для генерации т-образного перекрестка
- bool AddLeftTurt_ (CoatingType type, int column, int row, size_t linesCount, AngleType angle)

  Функция для генерации левого поворота
- bool AddRightTurt_ (CoatingType type, int column, int row, size_t linesCount, AngleType angle)

Функция для генерации правого поворота

- bool AddControlUnit_ (ControlUnitShared controlUnit)

  Функция для добавления управляющего блока в общий массив (для обновлений)
- bool AddStaticObject_ (int column, int row, StaticShared staticObject)

  Функция для генерации статического объекта
- bool AddBuilding_ (size_t type, int column, int row, float angle)

  Функция для генерации строения
- bool AddBush_ (size_t type, int column, int row, float angle)

  Функция для генерации куста
- bool AddDynamicObject_ (int column, int row, DynamicShared dynamicObject)

  Функция для генерации динамического объекта
- bool AddCar_ (size_t type, int column, int row, float angle)

  Функция для генерации машины
- size_t GetVectorColumn_ (int column)

  Функция для получения столбца в массиве
- size_t GetVectorRow_ (int row)

  Функция для получения строки в массиве
- int GetRealColumn_ (size_t column)

  Функция для получения столбца объекта
- int GetRealRow_ (size_t row)

  Функция для получения строки объекта
- void CloseMap_ ()

  Функция закрытии карты
- void ClearSpawns_ ()

  Функция для очистки массива точек генерации
- void ClearCoatingObjects_ ()

  Функция для очистки матрицы покрытий
- void ClearControlUnits_ ()

  Функция для очистки движущихся объектов
- void ClearStaticObjects_ ()

  Функция для очистки статических объектов
- void ClearDynamicObjects_ ()

  Функция для очистки движущихся объектов

Закрытые данные

- WorldScene * scene_

  Сцена, к которой привязан контроллер
- cocos2d::Layer * mainLayer_

  Основной слой сцены (на нём вся движуха)
- bool isPause_

  Происходят ли обновления (точнее стоит ли пауза)
- uint8_t hiddenArea_

  Размер скрытой зоны
- uint16_t columnsCount_

  Количество колонок (включая скрытую зону)
- uint16_t rowsCount_

  Количество строк (включая скрытую зону)
- SpawnVector spawns_

  Массив точек генерации транспорта

- float deltaTime_
  Последняя разница между обновлениями
- float spawnTime_
  Время, прошеднее с последней автоматического генерации транспорта
- float cleanTime_
  Время, прошеднее с последнего автоматического удаления аварий
- float timeFactor_
  Коэффициент ускорения времени
- std::string lastMapFile_
  Последняя загруженная карта (путь к файлу)
- CoatingUnionMatrix coatingUnions_
  Матрица покрытий
- ControlUnitVector controlUnits_
  Матрица объединений покрытий
- StaticMatrix staticObjects_
  Матрица статических объектов
- DynamicVector dynamicObjects_
  Массив движущихся объектов

7.18.1 Подробное описание

Класс контроллера карты, связующее звено всех объектов

7.18.2 Конструктор(ы)

7.18.2.1 WorldController() [1/3]

```
rtm::WorldController::WorldController (
            WorldScene *const scene )
```

Конструктор без загрузки какой-либо карты

Аргументы

scene	сцена, к которой привязан контроллер

7.18.2.2 WorldController() [2/3]

```
rtm::WorldController::WorldController (
            WorldScene *const scene,
            std::string const & filename )
```

Конструктор с загрузкой карты

Аргументы

scene	сцена, к которой привязан контроллер
filename	путь к карте

7.18.2.3 WorldController() [3/3]

```
rtm::WorldController::WorldController (
            WorldScene *const scene,
            size_t mapNumber )
```

Конструктор с загрузкой карты

Аргументы

scene	сцена, к которой привязан контроллер
mapNumber	номер стандартной карты

7.18.3 Методы

7.18.3.1 Update()

```
void rtm::WorldController::Update (
            float time )
```

Функция обновления

Аргументы

time	время, прошедшее с момента прошлого обновления (в секундах)

7.18.3.2 GetLayer()

```
cocos2d::Layer * rtm::WorldController::GetLayer ( ) const
```

Функция для получения основного слоя сцены (на нём вся движуха)

Возвращает

основной слой

7.18.3.3 GetColumnsCount()

size_t rtm::WorldController::GetColumnsCount ( ) const

Функция для получения количества столбцов карты

Возвращает

количество столбцов

7.18.3.4 GetRowsCount()

size_t rtm::WorldController::GetRowsCount ( ) const

Функция для получения количества строк карты

Возвращает

количество строк

7.18.3.5 GetDeltaTime()

float rtm::WorldController::GetDeltaTime ( ) const

Функция для получения последней разницей между обновлениями

Возвращает

разница во времени между обновлениями (в секундах)

7.18.3.6 GetTimeFactor()

float rtm::WorldController::GetTimeFactor ( ) const

Функция для получения коэффициента ускорения времени

Возвращает

коэффициент ускорения времени (1 - реальная скорость)

7.18.3.7 GetCoatingObject()

```
rtm::CoatingObject * rtm::WorldController::GetCoatingObject (
            int column,
            int row )
```

Функция для получения объекта в определенной клетке

Возвращает

указатель на объект объекта

7.18.3.8 GetCoatingUnion()

```
rtm::CoatingUnion * rtm::WorldController::GetCoatingUnion (
            int column,
            int row )
```

Функция для получения объединения покрытий в определенной клетке

Возвращает

указатель на объект объединения покрытий

7.18.3.9 GetStaticObject()

```
rtm::StaticObject * rtm::WorldController::GetStaticObject (
            int column,
            int row )
```

Функция для получения статического объекта в определенной клетке

Возвращает

указатель на статический объект

7.18.3.10 GetDynamicObjects()

```
std::vector< rtm::DynamicShared > & rtm::WorldController::GetDynamicObjects ( )
```

Функция для получения массива движущихся объектов

Возвращает

массив движущихся объектов

7.18.3.11 IsPause()

bool rtm::WorldController::IsPause ( )

Функция сообщает, происходят ли обновления

Возвращает

true, если происходят не происходят (стоит пауза), иначе false

7.18.3.12 IsCorrectColumn()

bool rtm::WorldController::IsCorrectColumn (
int column)

Функция проверяет корректность столбца

Аргументы

column	номер проверяемой колонки

Возвращает

true, если столбец корректный, иначе false

7.18.3.13 IsCorrectRow()

bool rtm::WorldController::IsCorrectRow (
int row)

Функция проверяет корректность строки

Аргументы

row	номер проверяемой строки

Возвращает

true, если строка корректная, иначе false

7.18.3.14 IsAllowableColumn()

bool rtm::WorldController::IsAllowableColumn (
int column)

Функция проверяет, можно ли двигаться в столбце

Аргументы

column	номер проверяемой колонки

Возвращает

true, если в столбце можно двигаться, иначе false

7.18.3.15 IsAllowableRow()

bool rtm::WorldController::IsAllowableRow (
int row)

Функция проверяет, можно ли двигаться в строке

Аргументы

row	номер проверяемой строки

Возвращает

true, если в строке можно двигаться, иначе false

7.18.3.16 IsVisibleColumn()

bool rtm::WorldController::IsVisibleColumn (
int column)

Функция проверяет видимость столбца

Аргументы

column	номер проверяемой колонки

Возвращает

true, если столбец виден, иначе false

7.18.3.17 IsVisibleRow()

bool rtm::WorldController::IsVisibleRow (
int row)

Функция проверяет видимость строки

Аргументы

row	номер проверяемой строки

Возвращает

true, если строка видна, иначе false

7.18.3.18 SetTimeFactor()

void rtm::WorldController::SetTimeFactor (
float factor)

Функция для установки коэффициента ускорения времени

Аргументы

factor	коэффициент ускорения времени (1 - реальная скорость)

7.18.3.19 LoadMap() [1/2]

bool rtm::WorldController::LoadMap (
std::string const & filename)

Функция для загрузки карты из файла

Аргументы

filename	полный путь к файлу с картой

7.18.3.20 LoadMap() [2/2]

```
bool rtm::WorldController::LoadMap (
                size_t number )
```

Функция для загрузки карты по номеру

Аргументы

number	номер стандартной карты

7.18.3.21 IsEmpty_()

```
bool rtm::WorldController::IsEmpty_ (
                int column,
                int row,
                size_t width = 1,
                size_t height = 1 )   [private]
```

Функция проверяет доступность зоны для генерации статических объектов и объектов объекта

Аргументы

column	левая колонка проверяемой зоны
row	нижняя строка проверяемой зоны
width	ширина проверяемой зоны
height	высота проверяемой зоны

Возвращает

true, если можно сгенерировать объект в данной зоне, иначе false

7.18.3.22 GenerateObject_()

```
bool rtm::WorldController::GenerateObject_ (
                uint8_t * params,
                uint8_t count )   [private]
```

Функция для парсинга параметров и генерации объектов

Аргументы

params	массив параметров генерации
count	количество параметров генерации в массиве

Возвращает

true, если объект получилось сгенерировать, иначе false

7.18.3.23 AddCoatingUnion_()

```
bool rtm::WorldController::AddCoatingUnion_ (
        int column,
        int row,
        CoatingUnionShared coatingUnion )   [private]
```

Функция для генерации объединения покрытий

Аргументы

column	левая колонка объединения покрытий
row	нижняя строка объединения покрытий
coatingUnion	умный указатель на объединение дорог

Возвращает

true, если объект получилось сгенерировать, иначе false

7.18.3.24 AddDriveway_()

```
bool rtm::WorldController::AddDriveway_ (
        CoatingType type,
        int column,
        int row,
        size_t width,
        size_t height,
        AngleType angle )   [private]
```

Функция для генерации прямой односторонней дороги

Аргументы

type	тип объекта (асфальт, грязь...)
column	левая колонка объекта дороги
row	нижняя строка объекта дороги
width	ширина объекта дороги
height	высота объекта дороги
angle	направление, в котором разрешенно движение

Возвращает

true, если объект получилось сгенерировать, иначе false

7.18.3.25 AddCrossroad_()

```
bool rtm::WorldController::AddCrossroad_ (
        CoatingType type,
        int column,
        int row,
        LinesCounts linesCounts,
        size_t controlUnitType = 0 )   [private]
```

Функция для генерации перекрестка

Аргументы

type	тип объекта (асфальт, грязь...)
column	левая колонка перекрестка
row	нижняя строка перекрестка
linesCounts	количество полос в каждом направлении
controlUnitType	тип управляющего модуля перекрестком (тип светофора)

Возвращает

true, если объект получилось сгенерировать, иначе false

7.18.3.26 AddTCrossroad_()

```
bool rtm::WorldController::AddTCrossroad_ (
        CoatingType type,
        int column,
        int row,
        LinesCounts linesCounts,
        AngleType nullDirection,
        size_t controlUnitType = 0 )   [private]
```

Функция для генерации т-образного перекрестка

Аргументы

type	тип объекта (асфальт, грязь...)
column	левая колонка перекрестка
row	нижняя строка перекрестка
linesCounts	количество полос в каждом направлении
nullDirection	сторона, в направлении которой нельзя двигаться
controlUnitType	тип управляющего модуля перекрестком (тип светофора)

Возвращает

true, если объект получилось сгенерировать, иначе false

7.18.3.27 AddLeftTurt_()

```
bool rtm::WorldController::AddLeftTurt_ (
            CoatingType type,
            int column,
            int row,
            size_t linesCount,
            AngleType angle )    [private]
```

Функция для генерации левого поворота

Аргументы

type	тип объекта (асфальт, грязь...)
column	левая колонка поворота
row	нижняя строка поворота
linesCount	количество полос
angle	угол поворота (самого поворота)

Возвращает

true, если объект получилось сгенерировать, иначе false

7.18.3.28 AddRightTurt_()

```
bool rtm::WorldController::AddRightTurt_ (
            CoatingType type,
            int column,
            int row,
            size_t linesCount,
            AngleType angle )    [private]
```

Функция для генерации правого поворота

Аргументы

type	тип объекта (асфальт, грязь...)
column	левая колонка поворота
row	нижняя строка поворота
linesCount	количество полос
angle	угол поворота (самого поворота)

Возвращает

true, если объект получилось сгенерировать, иначе false

7.18.3.29 AddControlUnit_()

bool rtm::WorldController::AddControlUnit_ (
ControlUnitShared controlUnit) [private]

Функция для добавления управляющего блока в общий массив (для обновлений)

Аргументы

controlUnit	умный указатель на управляющий блок

Возвращает

true, если объект получилось добавить, иначе false

7.18.3.30 AddStaticObject_()

bool rtm::WorldController::AddStaticObject_ (
int column,
int row,
StaticShared staticObject) [private]

Функция для генерации статического объекта

Аргументы

column	колонка статического объекта
row	строка статического объекта
staticObject	умный указатель на статический объект

Возвращает

true, если объект получилось сгенерировать, иначе false

7.18.3.31 AddBuilding_()

bool rtm::WorldController::AddBuilding_ (
size_t type,

int column,
int row,
float angle) [private]

Функция для генерации строения

Аргументы

type	тип строения
column	колонка строения
row	строка строения
angle	угол поворота строения

Возвращает

true, если объект получилось сгенерировать, иначе false

7.18.3.32 AddBush_ ()

bool rtm::WorldController::AddBush_ (
size_t type,
int column,
int row,
float angle) [private]

Функция для генерации куста

Аргументы

type	тип куста
column	колонка куста
row	строка куста
angle	угол поворота куста

Возвращает

true, если объект получилось сгенерировать, иначе false

7.18.3.33 AddDynamicObject_ ()

bool rtm::WorldController::AddDynamicObject_ (
int column,
int row,
DynamicShared dynamicObject) [private]

Функция для генерации динамического объекта

Аргументы

column	колонка динамического объекта
row	строка динамического объекта
dynamicObject	умный указатель на динамический объект

Возвращает

true, если объект получилось сгенерировать, иначе false

7.18.3.34 AddCar_()

```
bool rtm::WorldController::AddCar_ (
            size_t type,
            int column,
            int row,
            float angle )   [private]
```

Функция для генерации машины

Аргументы

type	тип машины
column	колонка машины
row	строка машины
angle	угол поворота машины

Возвращает

true, если объект получилось сгенерировать, иначе false

7.18.3.35 GetVectorColumn_()

```
size_t rtm::WorldController::GetVectorColumn_ (
            int column )   [inline], [private]
```

Функция для получения столбца в массиве

Аргументы

column	столбец объекта

Возвращает

столбец в массиве

7.18.3.36 GetVectorRow_()

size_t rtm::WorldController::GetVectorRow_ (
int row) [inline], [private]

Функция для получения строки в массиве

Аргументы

row	строка объекта

Возвращает

строка в массиве

7.18.3.37 GetRealColumn_()

int rtm::WorldController::GetRealColumn_ (
size_t column) [inline], [private]

Функция для получения столбца объекта

Аргументы

column	столбец в массиве

Возвращает

столбец объекта

7.18.3.38 GetRealRow_()

int rtm::WorldController::GetRealRow_ (
size_t row) [inline], [private]

Функция для получения строки объекта

Аргументы

row	строка в массиве

Возвращает

строка объекта

Объявления и описания членов классов находятся в файлах:

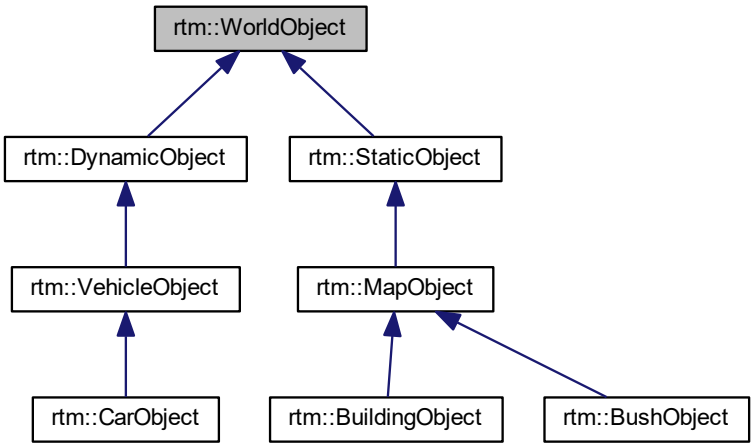
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/WorldController.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/WorldController.cpp

7.19 Класс rtm::WorldObject

Класс объекта мира (родитель всех условно объемных объектов)

#include <WorldObject.h>

Граф наследования:rtm::WorldObject:



Открытые члены

- WorldObject ()
  Конструктор по умолчанию
- WorldObject (cocos2d::Sprite *const sprite, float x, float y, float angle)
  Конструктор с использованием уже готового спрайта
- WorldObject (std::string const &filename, float x, float y, float angle)
  Конструктор из файла

- virtual ~WorldObject ()=default

  Деструктор по умолчанию
- cocos2d::Sprite * GetSprite () const

  Функция для получения спрайта
- float GetX_ () const

  Функция для получения абсциссы
- float GetY_ () const

  Функция для получения ординаты
- float GetAngle () const

  Функция для получения угла поворота
- float GetWidth () const

  Функция для получения ширины
- float GetHeight () const

  Функция для получения высоты

Защищенные члены

- void SetSprite_ (cocos2d::Sprite *const sprite)

  Функция для установки спрайта
- void SetX_ (float x)

  Функция для установки абсциссы
- void SetY_ (float y)

  Функция для установки ординаты
- void SetAngle_ (float angle)

  Функция для установки угла поворота
- void SetWidth_ (float width)

  Функция для установки ширины
- void SetHeight_ (float height)

  Функция для установки высоты
- virtual void PositionInit_ ()

  Функция, выполняемая во время инициализации
- virtual void PositionUpdate_ ()

  Функция, выполняемая во время обновления положения
- virtual void OnXUpdate_ ()

  Функция, выполняемая во время обновления абсциссы
- virtual void OnYUpdate_ ()

  Функция, выполняемая во время обновления ординаты
- virtual void OnAngleUpdate_ ()

  Функция, выполняемая во время обновления угла поворота
- virtual void OnWidthUpdate_ ()

  Функция, выполняемая во время обновления ширины
- virtual void OnHeightUpdate_ ()

  Функция, выполняемая во время обновления высоты

Закрытые члены

- void SetSpriteX_ (float x)
  Функция для установки абсциссы спрайта
- void SetSpriteY_ (float y)
  Функция для установки ординаты спрайта
- void SetSpriteAngle_ (float angle)
  Функция для установки угла поворота спрайта
- void SetSpriteWidth_ (float width)
  Функция для установки ширины спрайта
- void SetSpriteHeight_ (float height)
  Функция для установки высоты спрайта

Закрытые данные

- cocos2d::Sprite * sprite_
  Указатель на спрайт
- float x_
  Абсцисса
- float prevX_
  Абсцисса для отслеживания изменений
- float y_
  Ордината
- float prevY_
  Ордината для отслеживания изменений
- float angle_
  Угол поворота
- float prevAngle_
  Угол поворота для отслеживания изменений
- float width_
  Ширина
- float prevWidth_
  Ширина для отслеживания изменений
- float height_
  Высота
- float prevHeight_
  Высота для отслеживания изменений

7.19.1 Подробное описание

Класс объекта мира (родитель всех условно объемных объектов)

7.19.2 Конструктор(ы)

7.19.2.1 WorldObject() [1/2]

```
rtm::WorldObject::WorldObject (
            cocos2d::Sprite *const sprite,
            float x,
            float y,
            float angle )
```

Конструктор с использованием уже готового спрайта

Аргументы

sprite	указатель на готовый спрайт
x	абсцисса
y	ордината
angle	угол поворота объекта

7.19.2.2 WorldObject() [2/2]

```
rtm::WorldObject::WorldObject (
                std::string const & filename,
                float x,
                float y,
                float angle )
```

Конструктор из файла

Аргументы

filename	путь к файлу инициализации
x	абсцисса
y	ордината
angle	угол поворота объекта

7.19.3 Методы

7.19.3.1 GetSprite()

```
cocos2d::Sprite * rtm::WorldObject::GetSprite ( ) const
```

Функция для получения спрайта

Возвращает

указатель на спрайт

7.19.3.2 GetX_()

```
float rtm::WorldObject::GetX_ ( ) const
```

Функция для получения абсциссы

Возвращает

абсцисса

7.19.3.3 GetY_()

float rtm::WorldObject::GetY_ ( ) const

Функция для получения ординаты

Возвращает

ордината

7.19.3.4 GetAngle()

float rtm::WorldObject::GetAngle ( ) const

Функция для получения угла поворота

Возвращает

угол поворота

7.19.3.5 GetWidth()

float rtm::WorldObject::GetWidth ( ) const

Функция для получения ширины

Возвращает

ширина

7.19.3.6 GetHeight()

float rtm::WorldObject::GetHeight ( ) const

Функция для получения высоты

Возвращает

высота

7.19.3.7 SetSprite_()

void rtm::WorldObject::SetSprite_ (
cocos2d::Sprite *const sprite) [protected]

Функция для установки спрайта

Аргументы

sprite	указатель на спрайт

7.19.3.8 SetX_()

void rtm::WorldObject::SetX_ (
float x) [protected]

Функция для установки абсциссы

Аргументы

x	абсцисса

7.19.3.9 SetY_()

void rtm::WorldObject::SetY_ (
float y) [protected]

Функция для установки ординаты

Аргументы

y	ордината

7.19.3.10 SetAngle_()

void rtm::WorldObject::SetAngle_ (
float angle) [protected]

Функция для установки угла поворота

Аргументы

angle	угол поворота

7.19.3.11 SetWidth_()

void rtm::WorldObject::SetWidth_ (
float width) [protected]

Функция для установки ширины

Аргументы

width	ширина

7.19.3.12 SetHeight_()

void rtm::WorldObject::SetHeight_ (
float height) [protected]

Функция для установки высоты

Аргументы

height	высота

7.19.3.13 SetSpriteX_()

void rtm::WorldObject::SetSpriteX_ (
float x) [private]

Функция для установки абсциссы спрайта

Аргументы

x	абсцисса спрайта

7.19.3.14 SetSpriteY_()

void rtm::WorldObject::SetSpriteY_ (
float y) [private]

Функция для установки ординаты спрайта

Аргументы

y	ордината спрайта

7.19.3.15 SetSpriteAngle_()

void rtm::WorldObject::SetSpriteAngle_ (
float angle) [private]

Функция для установки угла поворота спрайта

Аргументы

angle	угол поворота спрайта

7.19.3.16 SetSpriteWidth_()

void rtm::WorldObject::SetSpriteWidth_ (
float width) [private]

Функция для установки ширины спрайта

Аргументы

width	ширина спрайта

7.19.3.17 SetSpriteHeight_()

void rtm::WorldObject::SetSpriteHeight_ (
float height) [private]

Функция для установки высоты спрайта

Аргументы

height	высота спрайта

Объявления и описания членов классов находятся в файлах:

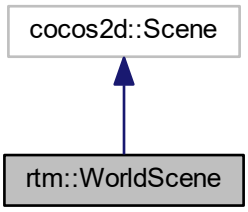
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/WorldObject.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/WorldObject.cpp

7.20 Класс rtm::WorldScene

Класс главной сцены, на которой всё и происходит (для отрисовки)

#include <WorldScene.h>

Граф наследования:rtm::WorldScene:



Открытые члены

- ~WorldScene ()=default

  Деструктор по умолчанию
- virtual bool init () override

  Функция для инициализации полей
- virtual void update (float time) override

  Функция для обновления сцены
- cocos2d::Layer * GetMainLayer () const

  Функция для получения основного слоя, на котором находятся объекты

Функции для установки фона

- void SetBackground (std::string const &filename)

  Функции для установки фона из файла
- void SetBackground (size_t number)

  Функции для установки фона по номеру

Открытые статические члены

- static WorldScene * Create ()

  Конструктор класса, поддерживающий RAII.

Закрытые члены

- void OpenMap_ ()
  Функция открытия карты
- void Restart_ ()
  Функция перезагрузки карты
- void SetDefaultPosition_ ()
  Функция для установки первоначальной позиции просмотра
- void ShiftUp_ ()
  Функция сдвига области просмотра вверх
- void ShiftRight_ ()
  Функция сдвига области просмотра вправо
- void ShiftDown_ ()
  Функция сдвига области просмотра вниз
- void ShiftLeft_ ()
  Функция сдвига области просмотра влево
- void UpdatePosition_ ()
  Функция для обновления положения главного слоя в зависимости от области просмотра
- void SetDefaultScale_ ()
  Функция для установки масштаба просмотра по умолчанию
- void IncreaseScale_ ()
  Функция для увеличения масштаба просмотра
- void DecreaseScale_ ()
  Функция для уменьшения масштаба просмотра
- void SetDefaultSpeed_ ()
  Функция для установки скорости обработки (скорости объектов) по умолчанию
- void IncreaseSpeed_ ()
  Функция для увеличения скорости обработки (скорости объектов)
- void DecreaseSpeed_ ()
  Функция для уменьшения скорости обработки (скорости объектов)
- WorldControllerUnique & GetMap_ ()
  Функция для получения контроллера данной сцены

Закрытые статические члены

- static void KeyPressed_ (cocos2d::EventKeyboard::KeyCode code, cocos2d::Event *event)
  Функция-обработчик нажатий клавиш клавиатуры
- static void KeyReleased_ (cocos2d::EventKeyboard::KeyCode code, cocos2d::Event *event)
  Функция-обработчик отпусканий клавиш клавиатуры

Закрытые данные

- cocos2d::Layer * mainLayer_
  Основной слой, на котором располагаются объекты
- cocos2d::Layer * backgroundLayer_
  Слой для фона, находится позади основного
- cocos2d::Sprite * background_
  Картинка фона
- WorldControllerUnique map_
  Контроллер мира, привязанный к данной сцене

- float clickTime_

  Время, прошедшее с последнего нажатия клавиш перемещения по карте (стрелочек)
- int viewColumn_

  Сдвиг по горизонтали при просмотре
- int viewRow_

  Сдвиг по вертикали при просмотре
- bool isCtrlPressed_

  Состояние клавиши CTRL.
- bool isAltPressed_

  Состояние клавиши ALT.
- bool isUpArrowPressed_

  Состояние клавиши "вверх".
- bool isRightArrowPressed_

  Состояние клавиши "вправо".
- bool isDownArrowPressed_

  Состояние клавиши "вниз".
- bool isLeftArrowPressed_

  Состояние клавиши "влево".

Закрытые статические данные

- static WorldScene * globalScene_ { nullptr }

  Основная сцена, к которой адресуются нажатия клавиш и т.д.

7.20.1 Подробное описание

Класс главной сцены, на которой всё и происходит (для отрисовки)

7.20.2 Методы

7.20.2.1 Create()

rtm::WorldScene * rtm::WorldScene::Create ( ) [static]

Конструктор класса, поддерживающий RAII.

Возвращает

указатель на созданный объект

7.20.2.2 init()

bool rtm::WorldScene::init ( ) [override], [virtual]

Функция для инициализации полей

Возвращает

true в случае успешной инициализации, иначе false

7.20.2.3 update()

void rtm::WorldScene::update (
float time) [override], [virtual]

Функция для обновления сцены

Аргументы

time	время, прошедшее с момента прошлого обновления

7.20.2.4 GetMainLayer()

cocos2d::Layer * rtm::WorldScene::GetMainLayer ( ) const

Функция для получения основного слоя, на котором находятся объекты

Возвращает

основной слой

7.20.2.5 SetBackground() [1/2]

void rtm::WorldScene::SetBackground (
std::string const & filename)

Функции для установки фона из файла

Аргументы

filename	полный путь к файлу с фоном (картинка)

7.20.2.6 SetBackground() [2/2]

void rtm::WorldScene::SetBackground (
size_t number)

Функции для установки фона по номеру

Аргументы

number	номер стандартного фона

7.20.2.7 GetMap_()

rtm::WorldControllerUnique & rtm::WorldScene::GetMap_ ( ) [private]

Функция для получения контроллера данной сцены

Возвращает

контроллер сцены

Объявления и описания членов классов находятся в файлах:

- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/WorldScene.h
- C:/Users/Vladimir/Documents/Visual Studio 2017/Projects/RTM/Classes/WorldScene.cpp

Предметный указатель