Руководство по Site3D

01.07.2020

Все взаимодействие с библиотекой происходит через класс site3d. Все наше 3D пространство мы будем называть сценой, а любой 3D объект в ней – моделью.

Начало работы

Вначале рассмотрим структуру библиотеки (содержимое каталога site3d):

- base – обязательные скрипты
- components – готовые компоненты: наполнения и модели
- modules – дополнительные модули
  - fly.js – полет камеры и моделей по заданной траектории
- pictures – изображения
- translations – интерфейсные фразы на разных языках
- include.js – скрипт подключения
- style.css – файл стилей

По умолчанию папка с библиотекой должна располагаться в корневом каталоге сайта. Если это не так, то необходимо указать путь в параметре site3d_path скрипта подключения, в нем же осуществляется подключение готовых наполнений и дополнительных модулей.

Чтобы начать работу с библиотекой, необходимо подключить js скрипт «site3d/include.js».

Скрипты наполнений содержат метод site3dFill[Имя скрипта](s3d, options), где s3d – объект сцены, а options – дополнительные параметры.

Скрипты моделей содержат метод site3dAdd[Имя скрипта](s3d, name, options), где s3d – объект сцены, name – имя модели, а options – дополнительные параметры.

Быстрый старт

Для демонстрации 3D модели достаточно вызвать статический метод site3d.widget (containerId, name, path, options), передав ему идентификатор HTML-контейнера, имя и путь до модели, а также, при необходимости, дополнительные параметры:

- autoLoad – если истина, то модель загружается автоматически, иначе загрузка происходит по клику (по умолчанию истина)
- isPreload – истина, если нужно показать окно загрузки (по умолчанию истина)
- autoPreload – истина, если нужно скрыть окно загрузки после загрузки модели (по умолчанию истина)

- preloadCompleted – функция завершения предварительной загрузки сцены (необходим вызов метода preload сцены для обозначения окончания этапа предварительной загрузки)
- nav – перечень кнопок навигации (по умолчанию ['full', 'rotate', 'save'], можно передать false для отключения панели)
- pos – позиция модели (по умолчанию [0, 0, 0])
- scale – масштаб модели (по умолчанию 1)
- rotate – углы поворота в виде массива [x, y], где x – угол поворота относительно глобальной оси x, а y - относительно локальной оси y (по умолчанию [0, 0])
- scaleControl – настройка управления масштабом модели: is_enabled (включить или выключить управление, по умолчанию включено), min (минимальное значение, по умолчанию 0.5), max (максимальное значение, по умолчанию 2)
- rotateControl – настройка управления вращением модели (по умолчанию включено): is_enabled (включить или выключить управление, по умолчанию включено), min (минимальный угол поворота, по умолчанию -90), max (максимальный угол поворота, по умолчанию 90)
- autoRotate – истина, если включен автоповорот (по умолчанию истина)
- rotateSpeed – скорость автоповорота относительно оси y (по умолчанию 10)
- environment – настройка окружения (передаются параметры из метода сцены enableEnvironment, по умолчанию включено, чтобы отключить, необходимо передать false)
- shadows – настройка отображения теней (по умолчанию включены, для отключения нужно передать false): angle – угол зрения (по умолчанию 90), near и far – пространство перед камерой (от и до, по умолчанию 0.1 и 100)
- ambientLight – параметры окружающего освещения: color (цвет, по умолчанию белый) и power (мощность, по умолчанию 1)
- directionalLight – параметры направленного освещения: color (цвет, по умолчанию белый), power (мощность, по умолчанию 1), pos (позиция, по умолчанию [1, 1, 1]) и target (позиция направления, по умолчанию [0, 0, 0])
- alpha – истина, если включена прозрачность (по умолчанию истина)
- дополнительные параметры метода importModel

В стилях контейнера необходимо указать ширину и высоту 3D сцены.

Пример вызова:

```
<div id="containerDemo" style="width: 100%; height: 500px;"></div>

<script>site3d.widget(' containerDemo ', 'demo', 'models/demo', {auto_rotate: 60});</script>
```

Метод возвращает объект сцены.

Правила установки параметров

- pos, target, rot – величины, передающие значения по трем осям координат, устанавливаются и возвращаются в формате массива (например, pos = [1, 2, 3] – позиция с координатами: x = 1, y = 2, z = 3)
- углы поворота указываются в градусах, время в секундах, а скорость в единицах перемещения в секунду
- axes – оси в формате строки (например, «xy» – оси x и y)
- color - цвет в формате HTML строки (например, «#ff0000» – красный)
- fill – наполнение модели цветами, текстурами, видео, щейдерами (размеры текстур желательно выбирать кратными степени двойки, например: 128, 256, 512 пикселей и т.д.), возможны следующие варианты:
    - name – имя загруженного ранее наполнения с помощью метода сцены add_fill(name, fill)
    - color – строка цвета
    - texture – строка пути до текстуры в формате JPG или PNG
    - video – идентификатор HTML-тега <video>
    - shader – объект {vars, pixel, vertex} для передачи результата выполнения шейдера (программы на языке GLSL), где vars – массив путей к текстурам или произвольный объект параметров для шейдеров, а pixel и vertex – идентификаторы или строки javascript кода для пиксельного и вершинного шейдеров
    - {value, count, side, color, emissive, shininess, metalness, roughness, blending, transparent, opacity}: value – color, texture, video или shader, далее идут дополнительные свойства: count – на сколько частей размножить (для текстур), side – какие грани заполняются («inside» – только внутри, «out» – только снаружи (по умолчанию), «both» – внутри и снаружи), color – цвет, emissive – цвет светимости (по умолчанию черный), shininess – уровень бликов (от 0 до 150), metalness – металлические свойства (от 0 до 1), roughness – шероховатость (от 0 до 1), blending – режим смешивания («no», «normal» (по умолчанию), «additive», «subtractive», «multiply»), transparent – истина, если поддерживается прозрачность (по умолчанию ложь), opacity – степень прозрачности (от 0 до 1)
    - массив из {name, fill} – только для импортированных моделей, где name – имя части модели, а fill – любое из предыдущих значений наполнения
- vars – переменные для шейдера в виде массива, элементы которого – массивы [name, value] из имени и значения переменной

Конструктор

site3d(canvas, options) – создание сцены, где canvas – идентификатор тега <canvas>, в котором и будет показана наша сцена, а options:

- load(info) – функция обработки процесса загрузки сцены, которая в качестве аргумента возвращает объект со следующими свойствами:
    - countModel – число добавленных для загрузки моделей
    - countModelLoaded – число загруженных моделей
    - isPreloadCompleted – истина, если завершена предварительная загрузка сцены (данный этап помечается методом preload)
- alpha – истина, если включена прозрачность (по умолчанию истина)

Загрузка сцены

- preload() – отметка момента окончания предварительной загрузки сцены

Отрисовка сцены

- render() – отрисовка текущего состояния сцены
- использование одного из дочерних классов для автоматической отрисовки сцены site3dRender
    - site3dRenderFull – класс для постоянного рендера сцены
    - site3dRenderActions – класс для рендера сцены только в момент взаимодействия пользователем

Методы site3dRender:

- constructor(s3d) – конструктор принимает объект сцены
- render() – отрисовка текущего состояния сцены
- start() – начать автоматическую отрисовку сцены
- stop() – остановить автоматическую отрисовку сцены

Настройки сцены

- background(color) – установка цвета фона
- skybox(path, size) – создание объемного фона, где path – путь до файла панорамной текстуры, size – размер объема
- effects(options) – эффекты сцены, где options:
    - brightness – яркость от 0 до 1 (по умолчанию 0.15)
    - contrast – контраст от 0 до 1 (по умолчанию 0.3)
- enableEnvironment(options) – включить окружение, где options: fill – путь до файла панорамной текстуры в формате HDR или одно и предустановленных значений: «grey» (по умолчанию), «sunset», tone – степень наложения текстуры на объекты от 0 до 1 (по умолчанию, 0.5), is_background – истина, если нужно отобразить текстуру в качестве фона сцены (по умолчанию false)

- disableEnvironment() – выключить окружение
- fog(color, near, far) – установка тумана: color – цвет, near и far – параметры густоты тумана в зависимости от расстояния

Добавление моделей и доступ к ним

Все модели имеют уникальное имя (name), набор параметров и наполнение (fill).

Некоторые модели имею функцию load(model) для информирования об окончании процесса загрузки (в качестве параметра передается сама модель). Для таких моделей начало взаимодействия необходимо проводить в данной функции или отслеживая результат метода модели isLoaded().

Рассмотрим методы добавления моделей с описанием специфических параметров:

- plane(name, width, height, fill) – плоскость: width, height – длина и высота
- cube(name, width, height, depth, fill) – куб: width, height, depth – длина, высота и глубина
- sphere(name, radius, detail, fill) – сфера: radius, detail – радиус и уровень детализации
- hemisphere(name, radius, detail, fill) – полусфера: radius, detail – радиус и уровень детализации
- text(name, text, options, fill, load) – 3D текст: text, options – текст и дополнительные параметры (font_path – путь до файла шрифта в формате json, font_size – размер шрифта, text_align – выравнивание относительно позиции модели (доступны значения «left» и «center»))
- importModel(name, path, options) – импорт модели в формате GLB, где path – путь до файла модели, а options – дополнительные параметры:
  - load(model) – функция, вызываемая после окончания загрузки модели
  - progress(model, info) – функция, вызываемая в процессе загрузки модели, где info содержит:
    - percent – процент загрузки модели
- importGroup(name, items, options) – импорт группы моделей в формате GLB, где items – массив с информацией о моделях (возможные значения: {name, model}, {name, path, options}, где name – имя модели, model – объект модели, path и options – параметры как в importModel), а options – дополнительные параметры:
  - isAutoConnect – если истина, то все модели группы соединяются с первой моделью (по умолчанию истина)
  - load – функция, вызываемая после окончания загрузки всех моделей группы

Рассмотрим структуру каталога для импортированных моделей:

- name.glb – файл модели

Доступ к любой модели осуществляется через метод model(name).

Работа с наполнением моделей

- addFill(name, fill) – добавить поименованное наполнение

Добавление источников освещения и доступ к ним

Все источники освещения имеют уникальное имя (name), цвет (color) и мощность (power).

Рассмотрим методы добавления света с описанием специфических параметров:

- ambientLight(name, {color, power}) – общий: color – цвет (по умолчанию белый), power – мощность (по умолчанию 1)
- directionalLight(name, {color, power, pos , target}) – направленный: pos – позиция источника (по умолчанию [1, 1, 1]), target – позиция направления (по умолчанию [0, 0, 0])
- spotLight(name, {color, power, pos , target, angle, blur}) – конусный: angle – угол конуса (по умолчанию 45), blur – сглаженность светового пятна (по умолчанию 0.5)

Доступ к источнику осуществляется через метод light(name).

Базовая работа с камерой

- camera(options) – настройка основных параметров: angle – угол зрения (по умолчанию 50), near и far – пространство перед камерой (от и до, по умолчанию 0.1 и 100), pos – позиция (по умолчанию [0, 0, 1]), target – точка наблюдения (по умолчанию [0, 0, 0])
- getCameraPos() – получить позицию камеры
- cameraPos(pos) или cameraPos(x, y, z) – установка позиции камеры
- isCameraPos(pos) – истина, если камера находится в позиции pos
- moveCamera(step) – движение камеры в направлении взгляда вдоль плоскости xz
- moveCamera(step, pos) – движение камеры в направлении точки pos
- moveCamera(stepX, stepY, stepZ) – движение камеры по трем осям относительно точки наблюдения
- getCameraRot() – получить поворот камеры в формате [x, y, z]
- cameraRot(rot) или cameraRot(x, y, z) – установка состояния поворота камеры
- rotateCamera(pos) – поворот камеры в направлении позиции pos

- rotateCamera(angle, options) – поворот камеры вокруг позиции на угол angle (options: pos – позиция, axe – строка оси вращения (по умолчанию «y»), is_look – истина, если нужно следить взглядом (по умолчанию true), все опции сохраняются и можно вызывать метод без них)
- rotateCamera(stepX, stepY, stepZ) – поворот камеры по трем осям

Полет камеры

fly(object, options, end) – полет камеры к определенной позиции или к их набору (функция end сигнализирует о завершении полета).

Возможные значения object:

- Строка имени модели
- Позиция

Можно указать массив из любых комбинаций данных значений, чтобы камера совершила последовательный облет нескольких позиций.

Возможные значения options:

- direction – направление движения камеры («forward» – вперед (по умолчанию), «back» – назад, «none» – без поворота)
- distance – на каком расстоянии от объекта остановиться (если «none», то перемещения не происходит, только поворот)
- duration – время полета (если указана скорость, то игнорируется)
- speed – скорость полета
- loop – истина, если нужно постоянное повторение всей цепочки полета

Также, каждому object можно передать уникальные свойства: {value: object, options: {direction, distance, duration, speed, end}}, где end – функция завершения этапа полета камеры.

Находится ли камера в состоянии полета, можно узнать через метод isFly(), а остановить полет методом stopFly().

Управление просмотром всей сцены

- enableControls(options) – включить
- disableControls() – выключить

Рассмотрим параметры просмотра (options):

- rotate: {events, axes, speed} – вращение

    - events – массив строк органов управления (возможные значения: «mouse_left» – левая кнопка мыши, «mouse_right» – правая кнопка мыши (по умолчанию: [«mouse_left», «mouse_right»]))
    - axes – оси вращения (x, y или x и y одновременно (по умолчанию: «xy»))
    - speed – скорость (по умолчанию: 1)

Методы для моделей

Загрузка

- isLoaded() – истина, если модель загружена
- hide() – скрыть модель
- show() – показать модель

Базовые методы

- size() – получить информацию о размерах модели, возвращает объект со следующими свойствами:
    - x – длина по оси x
    - y – длина по оси y
    - z – длина по оси z
    - pos – центр модели
- getPos() – получить позицию
- pos(pos) или pos(x, y, z) – установка позиции
- move(pos) или move(stepX, stepY, stepZ) – движение по трем осям
- moveLocal(pos) или moveLocal(stepX, stepY, stepZ) – движение вдоль локальных осей модели
- getScale() – получить масштаб
- setScale(scale) или setScale(x, y, z) – установка масштаба (можно задать одно значение для установки одинакового значения по трем осям)
- scale(scale) или scale(stepX, stepY, stepZ) или scale(step) – масштабирование по трем осям (можно задать одно значение для равномерного масштабирования по трем осям)
- getRot() – получить состояние поворота
- rot(rot) или rot(x, y, z) – установка состояния поворота
- rotate(stepX, stepY, stepZ, options) – поворот по трем локальным для модели или глобальным осям (options – дополнительные параметры (не обязательный параметр): is_local – истина, если оси локальные (по умолчанию истина), duration – время анимации)
- isRotate() – возвращает истину, если осуществляется поворот
- pauseRotate() – поставить анимацию поворота на паузу
- play_rotate() – продолжить анимацию поворота

- stop_rotate() – остановить анимированный поворот
- fill(fill) – редактирование наполнения
- getColor(name) – получить цвет модели (если указано имя, то цвет части модели)
- Полет модели осуществляется через методы fly, isFly и stopFly, как и соответствующие методы полета камеры

Управление просмотром

- enableControls(options) – включить
- disableControls() – выключить

Рассмотрим параметры просмотра (options):

- scale: {min, max, speed} – масштабирование колесом прокрутки
  - min – минимальный масштаб (по умолчанию: 0.5)
  - max – максимальный масштаб (по умолчанию: 2)
  - speed – скорость (по умолчанию: 1)
- rotate: {events, axes, speed} – вращение
  - events – массив строк органов управления (возможные значения: «mouse_left» – левая кнопка мыши, «mouse_right» – правая кнопка мыши (по умолчанию: [«mouse_left», «mouse_right»]))
  - axes – оси вращения (x, y или x и y одновременно (по умолчанию: «xy»))
  - speed – скорость (по умолчанию: 1)

Управление соединением с камерой или моделью

- connect(options) – соединить
- disconnect() – отсоединить

Рассмотрим возможные параметры (options):

- modelName – имя модели (если пустое значение, то соединение с камерой)
- moveLocal - смещение в рамках локальных осей модели
- rotateLocal – поворот относительно локальных осей

Управление ссылкой

- link(click, options) – включить, где click – функция обработки клика (в качестве аргументов возвращает модель (model) и стандартный объект результата клика (event)) или url адрес
- unlink() – выключить

Рассмотрим параметры ссылки (options):

- boundHover – истина, если наведение распространяется на область в виде параллелепипеда (по умолчанию false)
- newWindow – открыть ссылку в новом окне (актуален, если в качестве параметра click передан url адрес)
- hover – событие при наведении мышки
- out – событие при отведении мышки
- move – событие при движении мышки
- hint – строка идентификатора элемента HTML с подсказкой, которая появляется при наведении мышки и исчезает при её отведении

Работа с шейдерами

- setVars(vars) – установить переменные для шейдера

Управление метками

- setLabels(labels) – добавить метки, где labels – массив объектов со следующими свойствами: name (имя метки), content (идентификатор HTML-элемента), pos (позиция относительно модели (по умолчанию [0, 0, 0])) и is_show (истина, если метка видна (по умолчанию истина))
- setLabel(name, options) – установить новые свойства для уже добавленной метки или добавить новую метку, где name – имя метки, а options – свойства аналогичные для объектов метода set_labels
- deleteLabel(name) – удалить метку по имени
- showLabels() – показать все метки
- hideLabels() – скрыть все метки

Методы для источников освещения

Базовые методы

- setColor(color) – установка цвета
- setPower(power) – установка интенсивности освещения (по умолчанию 1)
- getPos() – получить позицию
- pos(x, y, z), pos(pos) – установка позиции
- move(stepX, stepY, stepZ) – движение по трем осям
- getTarget() – получить позицию направления освещения
- target(x, y, z), target(pos) – установка позиции направления освещения
- moveTarget(stepX, stepY, stepZ) – движение направления освещения по трем осям

Работа с тенями

Для отображения теней необходимо настроить источники света, которые будут их генерировать и указать объекты, которые будут отбрасывать и принимать тени.

Настройка источников света:

- enableShadows(options) – включить, где options:
  - size – размер текстуры для теней (от 1 до 3, по умолчанию 2)
  - параметры для направленного света: angle – угол зрения (по умолчанию 90), near и far – пространство перед камерой (от и до, по умолчанию 0.1 и 100)
- disableShadows() – выключить

Настройка моделей:

- enableShadows(actions) – включить, где actions – строки режимов (по умолчанию «cast_receive»):
  - «cast» – отбрасывать тени
  - «receive» – принимать тени
  - «cast_receive» – отбрасывать и принимать тени
- disableShadows() – выключить

Работа с шейдерами

Для того чтобы включить в свой проект шейдеры, достаточно применить соответствующее наполнение {vars, pixel, vertex} к модели (об этом упоминается в правилах установки параметров в начале данного руководства). Если не указывать uniform-переменные vars, то будут автоматически передаваться следующие значения:

- vec3 iResolution – длина и ширина сцены (третий параметр vec3 будет установлен в 1)
- float iTime – время в миллисекундах с момента начала загрузки сцены
- vec4 iMouse – координаты мыши (x и y – клик, z и w – перемещение)
- sampler2D iChannel0 и iChannel1 – текстуры

Работа с HTML-контентом

В Site3D можно назначить моделям свой контент для отображения во всплывающем окне.

Для этого нужно через метод сцены contentParams(container, content) передать идентификатор контейнера окна container и самого контента content (по

умолчанию имена контейнеров беруться из названия canvas из конструктора сцены с окончаниями «_window» и «_window_content» соответственно).

Далее нужно назначить модели идентификатор своего контента через метод setContent(content), а для вывода окна использовать метод showContent().