

**Описание функциональных характеристик  
программного обеспечения “Алгоритмический  
комплекс, обеспечивающий визуализацию данных  
САх процессов” и информация, необходимая для  
установки и эксплуатации программного обеспечения.**

ООО “ВГТ”

2019

# Содержание

<b>Содержание</b>	<b>1</b>
<b>Введение</b>	<b>2</b>
<b>Функциональные характеристики</b>	<b>2</b>
Цели и назначение	2
Ключевые функции	2
<b>Информация необходимая для установки и эксплуатации комплекса</b>	<b>3</b>
Системные требования	3
Требования к рабочему месту пользователя	3
Требования к квалификации пользователя	3
Требования к серверной инфраструктуре	3
<b>Подготовка к работе с использованием API комплекса</b>	<b>4</b>
<b>Установка виджета</b>	<b>4</b>
Запуск	4
Загрузка модели	5
Выделение объектов	5
Выделение извне	6
Выделение внутри виджета	6
Раскрашивание объектов	7
Фильтрация видимости объектов	8
Требования к исходным моделям	8
<b>Техническая поддержка</b>	<b>9</b>

# Введение

Данное руководство пользователя описывает правила, методы и принципы работы программного обеспечения “Алгоритмический комплекс, обеспечивающий визуализацию данных САх процессов” (далее - “комплекс”), разработанным ООО “ВГТ” (далее - “разработчик”).

## 1. Функциональные характеристики

### 1.1. Цели и назначение

Комплекс представляет собой базовый набор JS-модулей, позволяющих создавать интерактивные 3D визуализации, исполняемые в среде web-браузеров с поддержкой WebGL API (версии 1.x или 2.x) и обеспечивает:

- Отображение в web-браузере интерактивной 3D визуализации объёмом до 1 Гб и количеством деталей/объектов до 100 000 шт.;
- Возможность рендера в реальном времени (обработки и визуализации) сложных геометрических конструкций (3D моделей со сложной топологией поверхности) на пользовательском устройстве (ПК, MAC, мобильные устройства с операционной системой IOS 9+, Android 5.1+, Windows 10) в web-браузере.

### 1.2. Ключевые функции

Алгоритмический комплекс реализует следующие функции для конечного пользователя:

- Загрузка 3D модели в комплекс
- Пространственный анализ 3D модели целиком:
  - Вращение
  - Приближение/отдаление
  - Смена точки фокусировки камеры обзора
- Пространственный анализ выбранного отдельного объекта или группы объектов 3D модели:
  - Выбор объекта 3D модели
  - Приближение/отдаление от выбранного объекта
  - Смена точки фокусировки камеры для выбранного объекта
- Работа с иерархией объектов 3D модели:
  - Отображение иерархии объектов в соответствии со структурой 3D модели
  - Отображение названия или id объектов
  - Выбор отдельного объекта 3D модели

- Выделение выбранного объекта 3D модели цветом:
  - Выбор цвета для выделения
  - Выделение цветом отдельного объекта или группы объектов.
- Настройка видимости объектов:
  - Выбор объектов для отображения

## **2. Информация необходимая для установки и эксплуатации комплекса**

### **2.1. Системные требования**

Для разворачивания комплекса предъявляется три основных блока требований:

- требования к рабочим местам пользователей комплекса (разработчики)
- требования к квалификации пользователей комплекса (разработчики)
- требования к серверной инфраструктуре, в которой будет разворачиваться комплекс и создаваемые на его основе 3D визуализации

### **2.2. Требования к рабочему месту пользователя**

Рабочее место должно отвечать следующим требованиям:

- Операционная система:
  - Linux
  - MacOS
  - Windows

### **2.3. Требования к квалификации пользователя**

Пользователь комплекса должен обладать следующими компетенциями:

- Знание JavaScript
- Знание основ 3D моделирования объектов

### **2.4. Требования к серверной инфраструктуре**

Для разворачивания комплекса в рабочем режиме необходимо подготовить сервер, который должен отвечать следующим характеристикам:

- Аппаратные требования:
  - CPU-2
  - Memory - 2GB
  - Swap - 2GB
  - Disk size 100 GB

## 3. Подготовка к работе с использованием API комплекса

### 3.1. Установка виджета

Виджет представляет собой javascript файл.

Для работы виджета в своем веб-приложении необходимо:

- Поместить файлы виджета в папку со статическими файлами (в примерах ниже используется папка “industrial-viewer/” для файлов виджета)
- Подключить javascript файл через тег script

Пример:

```
<!DOCTYPE html>
<html lang="en">
<body>
  <div id="container"></div>

  <script src="industrial-viewer/app.js"></script>
</body>
</html>
```

### 3.2. Запуск

После подключения скрипта, в глобальном пространстве должна появиться переменная `industrialViewer`. Для инициализации приложения необходимо создать инстанс виджета с помощью конструктора, который находится в свойстве `default` в глобальной переменной `industrialViewer`. В конструктор необходимо передать `dom` элемент, в который будет виджет добавляться и настройки виджета.

API:

---

```
/**
 * @param {HTMLElement} container Элемент, в котором разместить виджет
 * @param {Object} config Настройки
 * @param {string} config.apiPath Путь до API по получению моделей
 * @param {selectObjectCallback} config.onSelectObject Описано в разделе
 "Выделение объектов"
 * @returns {Object} Instance проинициализированного виджета
 */
```

---

Пример:

```
const IndustrialViewer = industrialViewer.default;
const container = document.getElementById('container');

const widget3d = new IndustrialViewer(container, {
  apiPath: 'industrial-viewer/assets/',
  onSelectObject(id) {
    console.log('Object selected:', id);
  }
});
```

### 3.3. Загрузка модели

Загружает 3D модель по указанному идентификатору. При этом если была загружена другая модель, то она удаляется со сцены. Требования к исходным моделям описаны в п. 3.7

API:

---

```
/**
 * @param {string} id Идентификатор модели
 * @return {Promise<>}
 */
```

---

Пример:

```
widget3d.loadModel('factory').then(() => {
  // Loaded
}).catch((error) => {
  // Error
});
```

### 3.4. Выделение объектов

Допускается за раз выделение только одного объекта. При выделении нового объекта, у предыдущего выбранного объекта, выделение спадает. Если указанный элемент является узлом дерева с дочерними объектами, то выделяются все объекты в иерархии. Выделение объекта

возможно командой из основного веб приложения, так и может произойти внутри виджета, например, по клику на объекте в 3D сцене.

- **Выделение извне**

API:

---

```
/**
 * @param {string|null} id Идентификатор объекта
 * @return {Promise<>}
 */
```

---

Пример:

```
widget3d.selectObject('object15')
  .then(() => {
    // Selected
  })
  .catch((error) => {
    // Error
  });
```

- **Выделение внутри виджета**

API:

---

```
/**
 * @callback selectObjectCallback
 * @param {string} id Идентификатор объекта
 */
```

---

Пример:

```
widget3d.onSelectObject = (id) => {
  console.log('Object selected:', id);
};
```

### 3.5. Раскрашивание объектов

Метод раскрашивает указанные объекты в указанные цвета. Предыдущие параметры раскрашивания, если таковые применялись ранее, при этом сбрасываются. Если в списке объектов указан элемент, который является узлом дерева с дочерними объектами, то раскрашиваются все объекты в иерархии. Если аргумент методу не передан или передано null, то все расцветки сбрасываются.

API:

---

```
/**
 * @param {Array<ColorDescription>|null} list Список применяемых цветов
 * @return {Promise<>}
 */

/**
 * @type {ColorDescription}
 * @param {string} color Цвет в hex формате RRGGBB
 * @param {Array<string>} objectIds Массив идентификаторов объектов
 */
```

---

Пример:

```
widget3d.colorObjects([
  {
    color: 'ff0000',
    objectIds: ['object15', 'object39520']
  }, {
    color: '00ff00',
    objectIds: ['object39520']
  }
]).then(() => {
  // Colored
}).catch((error) => {
  // Error
});
```



### 3.6. Фильтрация видимости объектов

Метод скрывает все объекты, кроме тех, id которых переданы методу. Предыдущие параметры фильтрации, если таковые применялись ранее, при этом сбрасываются. Если в списке объектов указан элемент, который является узлом дерева с дочерними объектами, то отображаются все объекты в иерархии. Если аргумент методу не передан или передано null, то отображаются все объекты.

API:

---

```
/**
 * @param {Array<string>|null} objectIds Массив идентификаторов объектов
 * @return {Promise<>}
 */
```

---

Пример:

```
widget3d.filterObjects(['ground1', 'ground2']).then(() => {
  // Visibility set
}).catch((error) => {
  // Error
});
```

### 3.7. Требования к исходным моделям

3D модель должна отвечать следующим базовым требованиям:

- A. Модель должна быть преобразована в формат glTF с сохранением идентификаторов объектов с минимально-возможным количеством уникальных геометрических объектов.
- B. Размер файла загруженной модели с учетом текстур для отображение в web-браузере должен составлять для:
  - a. Desktopных устройств - не более 1 Gb до инстанцирования\*
  - b. Мобильных устройств - не более 300 Mb до инстанцирования\*

\*Инстанцирование - замена уникального объекта на экземпляр класса объектов обладающих определенным состоянием, поведением и значениями свойств (атрибутов), а также операций над ними. Принципы инстанцирования описаны ниже в п. F

- C. Геометрический размер модели для визуализации в режиме реального времени при рендере одного кадра должен составлять для:
  - a. Desktopных устройств - не более 8 млн. вершин сетки на кадр
  - b. Мобильных устройств - не более 4 млн. вершин сетки на кадр

- c. Максимальные геометрические размеры модели приведены с учетом инстанцирования объектов, т.е., если в модели есть инстансы, то общий размер модели должен учитывать все созданные экземпляры инстанцируемых объектов.
- D. Рекомендуется проконтролировать количество треугольников в каждом отдельном объекте модели, чтобы он не содержал более 10 тыс. треугольников.
- E. Текстуры (если предполагаются) должны быть разрешением в степени 2 по каждой стороне изображения, но не более 2048, например: 32x128, 256x256, 256x128, 256x1024, 1024x2048 и подобные.
- F. Модель должна быть инстанцирована по следующим принципам:
  - a. Каждый объект в сцене должен быть представлен определенной сеткой, а также положением, поворотом этой сетки в пространстве и её масштабом;
  - b. Если у двух и более объектов сетки идентичны, то необходимо сохранять сетку в памяти лишь один раз. Таким образом данные объекты будут ссылаться на одну и ту же сетку, но при этом обладать собственным масштабом, положением и поворотом в пространстве. В таком случае объекты будут считаться инстанцированными;
  - c. Необходимо инстанцировать максимальное количество объектов в сцене, при этом общий объем сцены с учетом инстансов должен удовлетворять требованию В настоящего пункта;
  - d. Если две сетки различаются количеством и/или порядком расположения их треугольников/вершин настолько, что это невозможно заметить визуально, то необходимо исключить из сцены одну из этих сеток (ту, у которой больше вершин), а все объекты, которые были ей представлены должны быть представлены оставшейся сеткой (той, у которой вершин/треугольников меньше).
- G. Индексы вершин должны быть упорядочены в соответствии с топологией сетки.
- H. На каждом объекте модели должны присутствовать корректные группы сглаживания.
- I. Каждый объект должен содержать корректную информацию о нормалях (верное направление нормалей в заданной группе треугольников образующих единую логически законченную поверхность или форму).
- J. Все объекты должны содержать в своем наименовании только строчные (маленькие) буквы латинского алфавита, знак нижнего подчеркивания «\_» и арабские цифры от 1 до 9.

## 4. Техническая поддержка

Вопросы возникающие в ходе работы с комплексом следует направлять в службу поддержки по адресу [support@wgt3d.com](mailto:support@wgt3d.com)

Все обращения рассматриваются в рабочее время (Europe GMT+3), ответы и оказание поддержки в штатном режиме предоставляются не позднее 48 часов с момента обращения.