|  |  |
| --- | --- |
|  | Task for JavaScript OOP  Vector Editor |
| EPMR-TCSR |

ИСТОРИЯ ИЗМЕНЕНИЙ

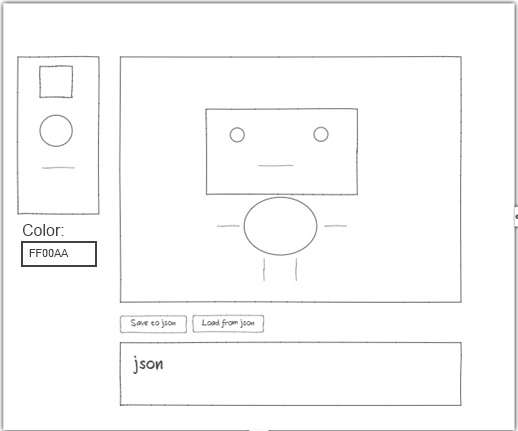
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Версия | Описание изменений | Автор | Дата | Утверждено   |  |  | | --- | --- | | Имя | Дата | | |
| <1.0> | Initial version | Михаил Семичев | <26-Июля-2014> |  |  |
| <1.1> | Final version | Alexander\_Kuznetsov1 | <30-Июля-2014> |  |  |

СВЯЗАННЫЕ ДОКУМЕНТЫ

|  |  |
| --- | --- |
| Номер | Название документа |
|  |  |
| 1 |  |

Практическое задание

Реализовать с использованием JavaScript простейший векторный редактор.



## Виды фигур:

* Квадрат
* Эллипс
* Линия

## Операции с фигурами:

* Добавлять
* Выделять
* Перемещать выделенную фигуру мышью
* Удалять выделенную фигуру кнопкой DELETE
* Определить цвет рисования фигур
* Перекрашивать выделенные фигуры
* Сохранять в textarea все нарисованные фигуры в формате JSON по кнопке «Save to JSON»
* Загрузить сохраненные фигуры из JSON строки на холст по кнопке «Load from JSON»

## Обязательно:

* Использовать ООП. Должен быть базовый класс для всех фигур Shape с методом получения уникального номера фигуры, а так же методом select’a фигуры и получения её статуса isSelected. Все остальные классы наследуют от Shape, переопределяют метод сохранения, определяют рендеринг и другие дополнительные методы.
* Методы фигур хранить в прототипах (для эффективного потребления памяти)
* Использовать Canvas.

## Дополнительные плюсы

1. Использование паттерна MVC:

* Model – сами фигуры и некоторый класс хранилище
* View – отвечает за рендеринг фигур на канве
* Controller – отвечает за обработку действий пользователя, реакции на нажатия клавиш, обработка действий мыши

1. Вращение фигур относительно центра
2. Реализация группировки фигур.