Выполнить задание используя разметку из задания 3.

Разделить логику работы страницы на следующие скрипты:

1. “common.js” (см. вложение)

Константы и часто используемые функции.

1. “Zombie.js”

Должен содержать функцию-класс для создания базового объекта зомби.

В рамках этого скрипта не допускается создание никаких других глобальных объектов.

1. “ZombieMad.js”

Должен содержать функцию-класс для создания объекта зомби.

Должен наследовать все методы функции-класса “Zombie”.

В рамках этого скрипта не допускается создание никаких других глобальных объектов.

1. “ZombieStrong.js”

Должен содержать функцию-класс для создания объекта зомби.

Должен наследовать все методы функции-класса “Zombie”.

В рамках этого скрипта не допускается создание никаких других глобальных объектов.

1. “Engine.js”

Должен содержать функцию-класс для создания объекта по управлению зомби.

Должен содержать только логику управления объектами-зомби.

В рамках этого скрипта не допускается работа с DOM элементами.

В рамках этого скрипта не допускается создание никаких других глобальных объектов.

1. “app.js”

Должен содержать всю логику работы с DOM элементами.

В рамках этого скрипта не допускается обращение напрямую к любому из объектов зомби.

В рамках этого скрипта не допускается создание никаких глобальных объектов.

При первом открытии страница должна выглядеть как показано на рис. 1:

1. Кнопки <Create> должны быть доступны
2. Кнопки <Damage> и <Kill> должны быть недоступны
3. Изображение зомби должно отсутствовать

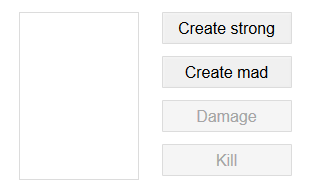


рис. 1

При нажатии на кнопку <Create strong> должен быть создан объект зомби класса “ZombieStrong”.

При нажатии на кнопку <Create mad> должен быть создан объект зомби класса “ZombieMad”.

После создания объекта-зомби должна отобразится соответсвующая ему картинка (рис. 2 и рис. 3)

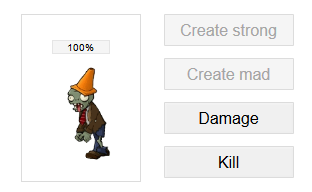


рис. 2

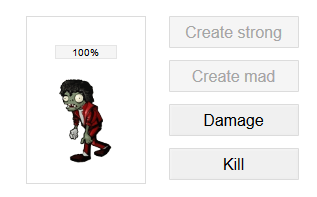


рис. 3

После появления зомби кнопки <Create> должны стать недоступны.

После появления зомби кнопки <Damage> и <Kill>должны стать доступны.

После появления зомби его первоначальный уровень здоровья должен быть установлен в значение константы “ZOMBIE\_MAX\_HEALTH” файла “common.js”.

Текущий уровень здоровья (в %) должен отображаться над зомби в виде полосы здоровья (рис. 4). При этом текущий размер полосы здоровья должен быть пропорционален максимальному (рис. 5)



рис. 6

При нажатии на кнопку <Damage> существующему зомби наносится урон, равный значению константы “ZOMBIE\_HIT\_DAMAGE” файла “common.js”.

При нажатии на кнопку <Kill> уровень здоровья зомби должен быть установлен в значение “0”.

Если у зомби уровень здоровья опустился до значения “0”, то зомби погибает.

Если зомби погибает, то изображение зомби должно измениться (рис. 7).

Если зомби погибает, то кнопки <Damage> и <Kill> должны стать недоступны.

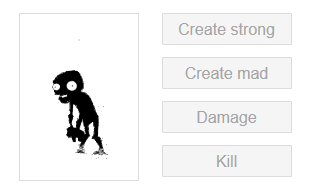


рис. 7

После того, как зомби погиб, должен включится счетчик ожидания для удаления объекта зомби. Время ожидания в миллисекундах равно значению константы ZOMBIE\_DELETE\_TIMEOUT файла “common.js”. По истечении этого времени объект зомби должен быть удален.

При удалении зомби страница возвращается в первоначальное состояние (рис. 1).

Особенности классов зомби:

1. Если у зомби класса “ZombieMad” текущий уровень здоровья опускается ниже 50%, то он погибает
2. Зомби класса “ZombieStrong” вдвое уменьшает урон, получаемый по нажатию на кнопку <Damage>