

## Игра память

Необходимо разработать браузерную игру “Память”.

При открытии страницы у пользователя есть возможность выбрать в выпадающем списке размер игрового поля (список из нескольких вариантов, например 6 x 6, 8 x 8, 10 x 10). По умолчанию выбирается самое маленькое поле 6 x 6, что будет равно 36 игровым карточкам.

Суть игры заключается в том чтобы отгадать все парные карточки, расположение которых генерируются случайным образом исходя из размера игрового поля. Для каждой карточки может быть только одна пара. После клика на карточку она переворачивается, и на другой стороне показывается цифра или картинка. Следующим кликом на поле выбирается другая карточка и также переворачивается. Если она отличается, то обе они переворачиваются обратно. В случае совпадения обе карточки остаются в таком положении и игра продолжается, пока не будут найдены все пары.

В процессе игры идет подсчет времени. После того, как все совпадающие карточки найдены, пользователю показывается таблица / список с результатами всех его ходов, общее время игры, общее количество шагов и кнопка “Начать заново”, при клике на которую запускается новая игра.

Возможный дизайн-макет:

<https://www.figma.com/file/47pPBg8hjiApqYVA3UMjA5/memory-game?type=design&node-id=0%3A1&mode=design&t=qPWcFPGUtHYNULjy-1>

## Техническое задание

Ниже описано минимальное техническое задание, которое необходимо выполнить для отправки задания. Также ниже представлен список дополнительных возможностей приложения, и чем больше их будет реализовано, тем больше у вас шансов на победу.

### Задачи

Сгенерировать проект при помощи Create React App, Vite или создать собственную конфигурацию Webpack (нет ограничений).

Реализовать приложение, в котором можно будет сыграть в игру “Память”.



### **Важно**

- Для стилизации можно использовать любой подход: простой CSS, препроцессоры (Sass / Less и т. д.) или библиотеки компонентов (MUI, Ant Design и т. д.)
- Расположение элементов и дизайн на ваше усмотрение, важно наличие необходимых компонентов.
- Если потребуется, то можно использовать любой менеджер состояния (Redux, MobX, Zustand и т. д.), либо использовать состояние React и Context.

### **Обязательные компоненты**

Ниже представлен список компонентов, которые в том или ином виде должны присутствовать в проекте. При использовании готовых библиотек компонентов их API может отличаться от описанного ниже, но суть остается та же.

Компонент выбора размера игрового поля

Компонент принимает:

- Массив из элементов списка размеров игрового поля, которые будут отображены
- Текущий **id** выбранного из списка размера игрового поля (если такой есть)
- Функцию **onChange**, которая вызывается при смене размера игрового поля из списка

Компонент игрового поля

Компонент принимает:

- Данные о размере поля и на его основе генерирует игровую сетку
- Функцию **onClick**, которая вызывается при клике на “Начало игры”

Компонент игровой карточки

Компонент принимает:

- Функцию **onClick**, которая будет вызвана при клике на карточку
- Сгенерированный **id** для данной карточки

Компонент отображения таймера

Компонент должен отображать время игровой сессии

Компонент вывода результатов игры

Данный компонент должен принимать данные по всем совершенным ходам во время игровой сессии и отображать их (например, в табличном виде), начиная с первого хода и до последнего

### **Дополнительные возможности**

Вы можете добавлять функционал и всячески улучшать приложение на ваше усмотрение. Чем больше дополнительных возможностей будет реализовано, тем выше шанс на победу. Вот несколько идей, которые можно внедрить:

- Отдельная страница, где показываются история всех сыгранных пользователем игр.
- Возможность регулирования размера игрового поля вводом произвольных данных в поле ввода. Но нужно учитывать максимальный размер (например, 15 x 15).
- Возможность выбора языка для игры.
- Возможность включения и отключения звукового сопровождения (например, фоновая музыка, звук при успешном совпадении карточек и так далее).
- Возможность выбора оформления (например, какие-либо фильмы, книги, города которые будут использоваться в качестве значений на карточках).
- Возможность сохранять текущую игру и возобновлять ее позднее.
- Возможность играть по сети нескольким пользователям с использованием сервера и веб-сокетов.