

ТЗ к проекту «Memory»

Описание проекта

Приложение тренирует память на разных наборах изображений. Пользователь должен составить пары из карточек с одинаковыми изображениями. Но сделать это не так просто.

Программа в простой и удобной форме обучает запоминанию. Используя разные наборы изображений, мы можем тренировать разные ассоциативные цепочки. Запоминая положение разных карточек мы кодируем наборы образов. Сначала доведите до совершенства один набор, который нужно запомнить, а затем переключитесь на другой. Прогресс результата означает прогресс в обучении.

Придумайте свою тему, подключите набор изображений и проводите дни и ночи за экраном, стараясь удержать в памяти эти картинки! Подойдут любые карточки, на которых изображены однотипные предметы. При этом недопустимо, чтобы изображения разных предметов были похожи друг на друга или походили на что-то другое.

Задача

Вам нужно запрограммировать версию для одного игрока популярной игры на развитие памяти и внимания. Мы уже подготовили вёрстку и графику. Используйте React для декларативного создания интерфейса игры.

Приложение состоит из трех экранов: *начальный экран*, *экран игры* и *экран результатов*.

- После загрузки страницы пользователь видит **начальный экран**. Начальный экран отображает правила игры и позволяет выбрать тему изображений на карточках.
- **Экран игры** отображает поле с карточками и текущую статистику игры (количество открытых карточек и затраченное число ходов).
- **Экран результатов** отображает таблицу с результатами игр пользователей.

Начальный экран

- При клике на кнопку выбора темы изображений пользователь переходит к экрану игры с выбранным комплектом карточек.
- Для тренировки есть наборы «цветы», «котики» и «машины».
- При каждом старте игры набор карточек перемешивается в случайном порядке.

Экран игры

- При старте игры пользователь видит все карточки рубашками вверх. Изображение на рубашке карточек соответствует выбранной теме.
- Пользователь переворачивает карточку кликом.
- Можно выбрать любую карточку в произвольном порядке. Для завершения хода пользователь выбирает вторую закрытую карточку. Повторный клик по открытой карточке игнорируется.
- Приложение проверяет открытую пару карточек, и если изображения на лицевой стороне карточек не совпадают, то обе карточки переворачиваются рубашкой вверх после короткой задержки, заданной в настройках приложения.
- Если изображения открытой пары совпадают, то карточки отображаются на поле лицевой стороной и далее в игре не участвуют. Такие карточки выделяются стилями.
- При выборе третьей карточки обе открытые карточки закрываются, третья карточка не открывается.
- Приложение показывает текущее количество переворотов пар карточек (ходов) и количество пар карточек, которые правильно собраны, во время игры.
- На экране показан индикатор прогресса игры. Содержимое индикатора закрашивается в ходе игры. Закрашенная часть индикатора пропорциональна доле собранных пар.
- Можно сделать не ограниченное количество ходов (переворотов пар карточек) за игру.

- Когда все карточки собраны в пары, пользователь видит **модальное окно (popup)** с сообщением об окончании игры и кнопкой перехода к экрану результатов. Переход к следующему экрану происходит после нажатия на кнопку.

Экран результатов

- Результаты игры других пользователей поступают из внешнего источника данных (файла).
- Результаты в таблице отсортированы по возрастанию количества ходов.
- Результат текущей игры отображается в строке «Ваш результат».
- Строки, в которых результат совпадает с текущим, выделены стилями.
- Экран результатов содержит кнопку начала новой игры. при нажатии на кнопку пользователь переходит к начальному экрану.
- При переходе к начальному экрану текущий результат сбрасывается.
- Приложение правильно склоняет числительные («1 шаг», «2 шага», «5 шагов»).

Технические требования

- Для разработки интерфейса используйте React. Код библиотеки подключите к странице тегом `<script>`.
- Приложение должно работать без сборки проекта. Для парсинга JSX-кода используйте библиотеку `babeljs` версии `standalone`.
- Для получения корректного склонения числительных используйте библиотеку `@dubaua/get-declension`.
- Код приложения пишите в теге `<script>` файла `index.html` для возможности локального запуска из файловой системы, без использования сервера сборки.
- Переход между экранами приложения происходит без перезагрузки страницы.
- В разметке приложения может находиться только корневой элемент (`#root`). Вся разметка интерфейса должна генерироваться из JS.

- При отключенном в браузере JS пользователь видит сообщение «*You need to enable JavaScript to run this app.*».
- Готовая разметка шаблонов передается разработчику в HTML-файлах. Вам нужно самостоятельно перенести разметку шаблонов в JSX-компоненты.
- Разработчик получает готовые стили и графику приложения.
- Стили и разметку изменять нельзя!
- Используйте в проекте синтаксис стандарта `ES6`.
- Для добавления обработчиков события используйте только механизмы React. Использование методов `addEventListener` и `querySelector` не допускается!
- Код для генерации случайного набора изображений карточек должен быть оформлен в функцию в файле `data.js`.
- Для удобной отладки приложения в файле настроек игры `settings.js` можно отключить перемешивание карточек.
- Задержку обратного переворачивания карточек `TIMEOUT` установите в файле настроек.
- Приложение сравнивает изображения по их `url`, который указан в данных в файле `data.js`.