Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего профессионального образования

**«Уральский федеральный университет
Имени первого Президента России Б.Н. Ельцина»**
Институт математики и компьютерных наук

Разработка браузерной версии логической игры Sherlock

Курсовая работа
Студента 4 курса
Группы КБ-401
Маценко О.Н.

Научный руководитель:
Игумнов А.С.

Екатеринбург
2015

Оглавление

Постановка задачи:

Шерлок — логическая игра, в которой нужно расположить 36 карт в нужной последовательность, используя 2 типа подсказок: вертикальные и горизонтальные. Для облегчения восприятия карт обычно используются 36 картинок, которые можно объединить в 6 групп. К примеру — 6 цифр, 6 букв и тд. Карты одной группы могут быть расположены только в одной строке. Вертикальные подсказки показывают какие карты находятся/не находятся в одном столбце. Горизонтальные — по строкам. Также горизонтальные ключи могут показывать последовательность карт на поле (карта X идет левее карты Y).

Игра ведется на поле размером 6x6 карт.

При первоначальном запуске — расположение карт по порядку, поля с ключами пустые. При нажатии кнопки старт - генерация уровня и его отрисовка.

Поле: если карта, на месте не инициализирована изначально (значение карты на данном месте должен выбрать пользователь), то показываются все возможные значения карт на этом месте, иначе — выбранная карта.

Подсказки:

- вертикальные — карты, которые расположены друг под другом, либо если карты не могут быть друг под другом, то одна из карт перечеркнута.
- горизонтальные: три карты, расположенные рядом.
  - Если это подсказка о порядке карт на поле, то сверху 3 карты стрелка, указывающая на то, что карты расположены в таком порядке, который может начинаться как слева, так и справа.
  - Если подсказка о порядке 2-х карт (карта X левее карты Y), то межжу этими двумя картами пиктограмма порядка.
  - Если подсказка о расположении соседних карт, то просто отрисовка 3 карт.
  - Если о том, что карта X не может находится в соседнем столбуе с картой Y, то одна из карт перечеркнута.

Игра ведется до тех пор, пока все карты на поле не будут расположены. В случае верного расположения — сообщение игроку о победе, иначе сообщение об ошибке и предложение отменить ходы до последней верной позиции карт. Также игру в любой момент может прекратить игрок с сохранением прогресса игры.

Игра запускается на клиенте без использования сервера. Желательно использование браузера Google Chrome, но возможна игра и в других браузерах. (Поддержка Chrome гарантирована, а вот в других браузерах могут быть ошибки).

Требования к интерфейсу:

1. понятность для любого пользователя
2. минимальная ширина игрового поля 600px (необходимо отобразить в ряд 15 карт, и их размер должен быть удобен, чтобы игрок мог совершить клик мыши на нужной карте)
3. минимальная высота игрового поля 500px
4. специальное поле для отображения информации об игре (имя игрока, уровень игры, таймер)
5. поле с управляющими кнопками
    1. старт новой игры / пауза
    2. начать текущую игру заново
    3. уровень / завершить текущую игру
    4. имя игрока / отменить ход
    5. статистика игры / подсказка
    6. настройки
    7. справка
    8. выход
6. 2 поля с подсказками
    1. вертикальные

        1. отображаются снизу от игрового поля
        2. минимальный размер одной подсказки: 60*160px
    2. горизонтальные
        1. отображаются справа от игрового поля 3 в ряд
        2. минимальный размер одной подсказки: 180*80px
7. При запуске игры — верное разложение карт на игровом поле

Функциональность:

Минимальный набор функций

1. **старт новой игры** — генерация игрового поля, генерация ключей (заполнение массивов) и отображение начальной позиции на экране игрока.

2. **Подсказать** — генерация всплывающего окна с подсказкой. В случае, если игрок не верно расположил карты, подсказать об ошибке и дать возможность отката до последней верной игровой позиции

3. **Отменить ход** — отмена хода

4. **Отменить ходы до верного расположения карт**

5. **клик левой кнопкой мыши** на игровом поле:

   на варианте заполнения — выбрать эту карту

6. **клик правой кнопкой мыши**

   на варианте заполения — убрать/показать этот вариант

7. **изменить имя игока**

8. **изменить уровень игры**

9. **начать текущую игру заново** — очистка массивов и старт новой игры

10. завершить текущую игру — очистка массивов и возврат к начальному состоянию (все карты разложены верно, поля ключей пустые)

11. пауза — скрыть поля с ключами, остановить таймер

12. статистика — всплывающее окошко со статистикой

13. настройки — всплывающее окошко с настройками

14. справка - сплывающее окошко со справкой

15. выход — закрыть игру

Сохранение статистики игры: В LocalStorage на локальной машине пользователя. Формат записи:

- номер,
- уровень,
- имя игрока,
- время когда была игра,
- время игры

Храним только лучшие результаты для каждого игрока.

Для генерации уровней — файл level-dec.js в котором массив с данными. Формат — количество заданных изначально карт, их перечисление с позициями, количество вертикальных ключей, из перечисление, количество горизонтальных ключей, их перечисление.

Для програмной реализации использовать javascript, для реализации интерфейса — html. Возможно использование jQuery.

Набор файлов:

1. main.html — сама страничка, на которой отрисовывается поле
2. main.js — функционал, обеспечивающий отклик игры на действия пользователя.
3. Sherlock.js — игровой функционал
4. hint.js — подсказки для игрока
5. level-dec.js — файл с массивом, в котором заданы уровни (начальное расположение карт и ключи для решения).

Для легкой смены интерфейса функционал разделен на 2 части: логика игры и действия. Соответственно, при необходимости смены интерфейса необходимо заменить html страницу и main.js, который отрисовывает поле.

Рассмотрим как происходит реакция на событие вызванное игроком.

1. Игрок нажал на кнопку/на карту
2. Вызов функции A из main.js, которая отвечает за это событие
3. A вызывает функцию B из sherlock.js, которая изменяет игровые массивы в зависимости от вызвавшей ее функции.
4. При изменение массивов, влияющих на изменение раскладки карт на поле (выбор карты, карта не может быть на это месте и тд.) происходит вызов функции draw_field() из main.js, которая перерисовывает игровое поле.
5. Ожидание следующего действия игрока.

Рассмотрим отдельно функциональность всех js-файлов

main.js

- при загрузке страницы вызывается функция window.onload, которая отрисовывает игровое поле и стандартное расположение карт (по порядку)

- send_level — функция, отрисовывающая заданный игроком уровень игры. Вызывается при нажатии кнопки Set в окошке смены уровня
- change_level — открывает окошко смены уровня
- send_name — отрисовывает новое имя игрока
- show_set_name — показывает окошко смены имени игрока
- draw_big — отрисовывает выбранную карту. (нажатие на левую кнопку мыши)
- draw_variants — отрисовывает предполагаемые варианты заполнени данной клетки
- draw_field — отрисовка поле целиком
- myTimer - таймер
- correct_length — корректировка длины. Используется таймером, чтобы время всегда отражалось в формате dd:dd:dd
- start_new_game — запуск таймера игры, заполение массивов и отрисовка поля, в случае корректного заполнения массивов для игры
- td_right_click — клик правой ко=нопки мыши на одном из вариантов заполнения. В случае, если вариант уже убран игроком, то он появляется и наоборот. Этот ход записывается в историю ходов.
- td_click — клик левой кнопкой мыши. Выбирает карту как правильную для этой клетки.
- next_hint — подсказка

level-dec.js

- в данном файле единственный массив FBlevels, с данными для уровней.

sherlock.js

<u>глобальные переменные:</u>

- FlevelMap — массив, i элемент которого — индекс начала данных i-го уровня в массиве FBlevels

- FField - массив объектов, описывающий текущую раскладку карт на поле.
    - FField[i][j].UserValue — карта, которую выбрал пользователь для места i,j
    - FField[i][j].CorrectVlue — карта, которая должна быть на этом месте
    - Ffiled[i][j].Variants — возможные варианты
- FMainHClues — массив, i-ый элемент которого — i-ый горизонтальный ключ
    - FmainHClues[i].Card1, FmainHClues[i].Card2, FmainHClues[i].Card3 — карты ключа
    - FmainHClues[i].ClueType — тип ключа
        - hcNextTo — карты находятся в соседних столбцах. Card1 = Card3
        - hcNotNextTo - карты не находятся в соседних столбцах. Card1 = Card3
        - hcTriple - порядок столбцов. Card1 и Card2, Card2 и Card3 -в соседних столбцах
        - hcNotTriple - порядок столбцов. Card1 и Card2, Card2 и Card3 -ен могут находится в соседних столбцах
        - hcOrder — порядок Card1 левее Card3.
        - HcNone — пустой ключ
- FMainVClues — массив с вертикальными ключами
    - FmainVClues[i].Card1,FmainVClues[i].Card2 — карты ключа;
    - FmainVClues[i].ClueType — тип ключа
        - vcTogether — карты находятся в одном столбце
        - vcNotTogether — карты не могут находиться в разных столбцах
        - vcNone — пустой ключ

- steps_history - массив с историей ходов
    - steps_history[i].col — колонка
    - steps_history[i].row - строка
    - steps_history[i].was — при выборе большой карты, содержит информацию о том, в каких клетках данная карта была в вариантах заполнения, при выборе варианта — пустой массив
    - steps_history[i].var — при выборе большой карты — инормация о вариантах заполнения клетки [col][row], иначе пустой
    - steps_history[i].right — правильный или нет ход
    - steps_history[i].card.type — с какой картой действие:
        - big - большая
        - small - маленькая
    - steps_history[i].card.action - само действие (добавили, удалили)
    - steps_history[i].card.number — номер карты
- error_flag — была ли совершена ошибка.

Функции:

- choose_big — определили местположение карты. Изменяет массив Ffield:
    - удаляет выбранную карту из вариантов для заполнения в данной строке
    - проставляет флаг, что пое инициализировано
    - устанавливает userValue
    - если расставлены все карты, то вывод выиграли или нет
- add_step — добавляет шаг в историю ходов (steps_history) и проверяет на правильность ходов. Если текущий ход противоречит правильному заполнению поля, то выставляет флаг ошибки. Все последующие ходы помечаются как ошибочные

- remove_many — откат ходов до ошибки. Пока ход помечен как неверный, вызывает откат хода

- remove_step — откат одного хода. Удаляет один элемент из истории ходов и производит его откат. В случае отката хода определения местоположения карты, убирает ее, возвращает варианты заполнения данного места и добавляет в этот вариант заполнения в те ячейки строки, в которых была данная карта

- InitLevel — основная функция игры. Инициализирует все значения массивов.

- CheckPossibility — проверяет возможна ли карта на этом месте. Возвращает значения

    - cpCannotBe — не возможна (нет в предполагаемых значения, неверно указанное место)

    - cpCanBe — возможна (есть в предполагаемых вариантах)

    - cpIsHere — уже стоит на этом месте

- CheckCorrectness — проверяет правильность заполнения поля. Проверяет, что все карты могут быть расположены на данных местах

- delete_variants — удаляет карту из предполагаемых значений

- add_variants — добавляет картц в предполагаемые значения

- CreateFField — инициализирует массив Ffield.

    - Ffield[i][j].Inittial = false;

    - Ffield[i][j].UserValue =0

    - Ffield[i][j].CorrectValue=0

- solve_game — решает игру. Заполняет массив Ffield.UserValue правильной расстановкой карт. Для решения игры используются подсказки. В случае если игру по каким-то причинам невозможно решить выдает сообщение об ошибке.

- Div — целочисленное деление

- FirstInitField — инициплизация массивов Ffield, FmainHClues, FmainVClues
- CreateLevelMap — Создание карты уровней. Заполняет массив FlevelMap

hint.js

глобальные переменные:

- solved — решена ли игра. Используется для вывода подсказок если игра решена.

Функции

- GetRow - возвращает строку, в которой может быть карта
- все следующие функции проверяют возможность применения ключа заданного типа в данной ситуации.
- Как работают подсказки
    - Проверка правильности заполнения поля
    - Поиск противоречий между заполненным полем и вертикальными ключами
    - Поиск противоречий между заполненным полем и горизонтальными ключами
    - Поиск подсказок по вертикальным ключам (проверяем относительно верхней карты)
    - Поиск подсказок по горизонтальным ключам (проверяем относительно первой и второй карты)
    - Поиск подсказок по вертикальным ключам (проверяем относительно нижней карты)
    - Поиск подсказок по горизонтальным ключам (проверяем относительно второй и третьей карты)
    - Если не нашли, то выдаем ошибку

Источники

1. Исходные коды на языке Delphy - https://www.quadrathell.cn.ua/forum/32-372-1