

Задачи проекта

- продемонстрировать реализацию математической головоломки на одном из языков программирования
- проверить на практике теорему о несобираемости 15-шек при определенных начальных параметрах раскладки
- Разработать макет и сценарий приложения
- Разработать интерфейс приложения
- Изучить ПО необходимое для реализации проекта: js,html,css
- Написать код приложения на языке программирования javascript
- Провести тестирование
- Исправить ошибки
- продемонстрировать финансовые возможности математики (создатель 15-шек пообещал тому кто их соберет - 1млн долларов, люди скупили миллионы наборов этих 15-шек в попытках заработать этот миллион и обогатили создателя 15-шек

<h1 color='red'>Заголовок 1-ого уровня</h1> HTML

<div>

<h1 class='my' style='color:red;'>Заголовок 1-ого уровня</h1> CSS

<h1 class='my' style='color:black;'>Заголовок 1-ого уровня</h1> CSS

<h1 class='my2'></h1>

</div>

//.css

div->h1.my

{

font-style:italic;

}

-----алгоритм для смертных-----

1. Распрямим пятнашки в линию

например

01	02	05	04
11	06	08	07
13	12	15	—
14	10	09	03

01,02,05,04,11,06,08,07,13,12,15,__,14,10,09,03

в полученном наборе считаем количество беспорядков. Беспорядком будем называть ситуацию, когда 2 числа, стоящие слева направо не обязательно рядом, не стоят по возрастанию. Например, в нашей линии 11 и 03 несоседние образуют беспорядок, или 11 и 06 соседние и тоже образуют беспорядок.

Число беспорядков называется индексом перестановки. Индекс перестановки бывает четным и нечетным.

Главное утверждение:

если свободная клетка стоит в нижнем правом углу 15-шек, то если 15-шки как-то подвигать, а потом снова вернуть свободную клетку на начальную позицию (в правый нижний угол) - четность беспорядков не изменится.

Док-во:

Рассмотрим 2 случая

а) движения пустой клетка слева-направо - очевидно такое движение не меняет числа беспорядков, а значит не меняет и четности
пример

01,02,05,04,11,06,08,07,13,12,15,__,14,10,09,03

движение влево даст следующий результат

01,02,05,04,11,06,08,07,13,12,__,15,14,10,09,03

видим, что на число беспорядков это никак не повлияло

б) движения пустой клетки сверху-вниз.

утверждение: такое движение меняет четность числа беспорядков,
пример

было

01	02	05	04
11	06	08	07
13	12	15	__
14	10	09	03

01,02,05,04,11,06,08,**07**,13,12,15,__,14,10,09,03

стало

01	02	05	04
11	06	08	__
13	12	15	07
14	10	09	03

01,02,05,04,11,06,08,__,13,12,15,**07**,14,10,09,03

дело в том, что такую перестановку можно осуществить как последовательное применению 3-х простейших перестановок (простейшими мы назовем перестановки соседних чисел)

1перестановка)

01,02,05,04,11,06,08,**07**,13,12,15,__,14,10,09,03

01,02,05,04,11,06,08,13,**07**,12,15,__,14,10,09,03

2перестановка)

01,02,05,04,11,06,08,13,**07**,12,15,__,14,10,09,03

01,02,05,04,11,06,08,13,12,**07**,15,__,14,10,09,03

2перестановка)

01,02,05,04,11,06,08,13,12,**07**,15,__,14,10,09,03

01,02,05,04,11,06,08,13,12,15,**07**,__,14,10,09,03

а одна простейшая перестановка изменения количества беспорядков на 1, поэтому 3 простейших изменять количество беспорядков на 3, а значит изменят четность перестановки.

А теперь верно главное утверждение - потому что пустая клетка, начавшая свое путешествие в правом нижнем углу, чтобы вернуться обратно - должна сделать столько же движений вверх, сколько вниз. А значит четность перестановки изменится четное число раз - а значит исходная четность перестановки не изменится.

В силу главного утверждения, если изначально четность была нечетной, то она ей и останется для пустой клетки в правом нижнем углу. А требуемая для сборки комбинация предполагает четную четность в конце. Значит, собрать 15-шки невозможно при изначальной нечетности.

ОЖИДАЕМЫЙ РЕЗУЛЬТАТ РАБОТЫ

1. Создание полезной интерактивной головоломки для развития логического мышления. (первые 3 ряда 15-шек при должном усердии можно собрать), более того можно собрать базовую конфигурацию с нечетной четностью

01 02 03 04
05 06 07 08
09 10 11 12
13 15 14 __

2. Демонстрация того, что наша интуиция бывает обманчива и с виду легкие задачи на деле могут быть безумно трудными. Отсюда следуют вывод, что математические теоремы совершенно необходимо строго доказывать - потому что только так мы дойдем до истины.

3. Демонстрация того факта, что даже знание самые простейших конструкций `html+css+js` позволяет за очень короткий срок программировать фантастически удивительные вещи. (это была просто таблица с раскрашенными ячейками, в которые вписаны цифры. И эти ячейки реагируют на нажатие и меняют свой цвет и содержимое).

ЭТАПЫ РЕАЛИЗАЦИИ

1 Этап

- Подробно изучил материалы по самым знаменитым головоломкам
- Просмотрел существующие приложения в интернете.
- Оценил сложность разработки различных головоломок
- Выбрал из них самую легко реализуемую и одновременно производящую наибольшее впечатление на людей с моменты ее создания.
- Продумал инструментальный `web`-верстки для реализации задуманного.

2 Этап

- Изучил верстку html+css (создание макетов) и среду программирования js
- Разработал дизайн приложения и структуру
- Реализовал модель приложения в браузере

3 Этап

- Осуществил отладку приложения
- Провел тестирование приложения с учащимися школы
- Осуществил коррекцию алгоритма приложения
- Провел финальное тестирование
- Предусмотрел хаотичное начальное положение игры, чтобы было интереснее играть

-----АКТУАЛЬНОСТЬ-----

данная работа очень важна как для развития навыков логического мышления у людей всех возрастов, так и для осознания людьми того факта - что изучение науки очень важно. Оно может как уберечь Ваши деньги (как в случае с 15-шками), так и раскрыть закономерности окружающего мира, которые могут поспособствовать общему прогрессу человечества.

ССЫЛКИ

https://ru.wikipedia.org/wiki/%D0%98%D0%B3%D1%80%D0%B0_%D0%B2_15

<http://scanvord.net/pyatnashki/>

<https://www.youtube.com/watch?v=WxmDbs-eYvE>

-----ЗАКЛЮЧЕНИЕ-----

Я успешно продемонстрировал как на очень базовых конструкциях html+css+js можно реализовывать очень нестандартные и яркие математические идеи. Используя всего

лишь навсего табличную верстку html и простейший процедурный js уже можно создавать целые игры с графическим интерфейсом. Большинство суперсовременных графических эффектов совершенно избыточны для большинства задач. Даже обходясь малым, можно делать поистине прекрасные web-приложения.

-----ПРАКТИЧЕСКАЯ ЗНАЧИМОСТЬ-----

1) Развитие логического мышления.

2) Понимание того факта, что не во всякую игру в принципе можно выиграть. И часто победой следует считать не победу в игре - а строгое обоснование невозможности этой победы.

-----ПЕРСПЕКТИВЫ-----

Математика переполнена нерешенными задачами и головоломками. И каждая из них может стать источником для новых невероятных и очень полезных программ. И помимо эстетического удовольствия может принести реальную финансовую выгоду.