

Применение средств языка JavaScript при разработке игрового приложения

Добровольская Олеся Михайловна,
10 «Б» класса, МБОУ «Школа №48»
Научный руководитель: Петухина И.В.,
учитель информатики

Школьный курс информатики подразумевает изучение основ языка программирования. До сих пор одним из самых популярных языков является Pascal. В данной работе рассматривается выполнение игрового приложения "Игра в 15" на более современном и востребованном языке программирования JavaScript с графическим интерфейсом, окном приветствия с уточнением правил, секундомером, отсчитывающим время в игре, окном окончания игры

В теоретической части работы рассмотрены классические правила игры, рассмотрена ее история.

В практической части работы описана реализация данной игры на языке JavaScript.

Оригинальность написанного текста, проверенного в системе Antiplagiat.ru, составляет 97,84%.

Прекрасное средство для развития логики и творческого мышления – это головоломки, однако детям и взрослым в наш век больше не хочется проводить часы, передвигая фишки в небольшой деревянной коробочке, ведь на смену настольным играм и головоломкам пришли компьютерные развлечения. Вот почему «Игра в 15», работающая через обыкновенный браузер, может стать чудесной альтернативой «Пятнашкам» обыкновенным.

Я уже давно увлекаюсь веб-версткой, а на определенном этапе мне захотелось создать полноценное приложение, которое будет работать на веб-странице. Прекрасным вариантом для реализации является «Игра в 15», ведь для ее создания требуется хорошее владение средствами веб-разработки.

Классические правила

В настольном варианте головоломки нужно добиться упорядочения перемешанных фишек, то есть расположить их по возрастанию. Существует много различных версий головоломки: где-то нужно собрать картинку, в других использованы буквы вместо цифр, иногда целью является «магический квадрат».

Есть ли решение у задачи? Оказывается, в 50% случаев «Игра в 15» по классическим правилам не имеет решения. Одним из таких случаев является перестановка фишек 14 и 15.

Как же доказать неразрешимость задачи? Нам следует определить инвариант для наших «Пятнашек». Инвариант – это свойство объекта, которое не меняется при выполнении определенных преобразований. Если инвариант для двух объектов разный, то превратить один объект в другой путем указанных преобразований не удастся.

В случае «Игры в 15» инвариантной является четность суммы двух чисел: первое – это число пар плиток, в которых перед меньшим числом идет большее, а второе – номер строки с пустой ячейкой. При этом любую четную позицию можно свести к другой четной, а вот привести к нечетной – нельзя.

Правила моей игры

Учитывая, что при хаотичной расстановке плиток их расположение вполне может оказаться таким, которое нельзя упорядочить, выигрышными будут считаться два варианта группировки. При четном инварианте игра считается оконченной при «классической» расстановке плиток: по возрастанию от 1 до 15 и заканчивая пустым местом. При нечетном же инварианте игра считается оконченной, если числа от 1 до 13 выстроены в порядке возрастания, а далее стоят 15, 14 и пустая клетка.

Таким образом наша задача будет иметь решение при любом начальном расположении фишек.

В моей игре будет запущен секундомер, фиксирующий время решения головоломки, счетчик ходов, реализовано автоматическое определение инварианта и уточнение правил в соответствии с ним и краткое описание игры перед стартом.

При победе игрок сможет узнать свое время и число ходов и стараться улучшить результаты. Если он улучшит их в новой игре, об этом он узнает при победе и сможет сравнить новый и старый рекорды.

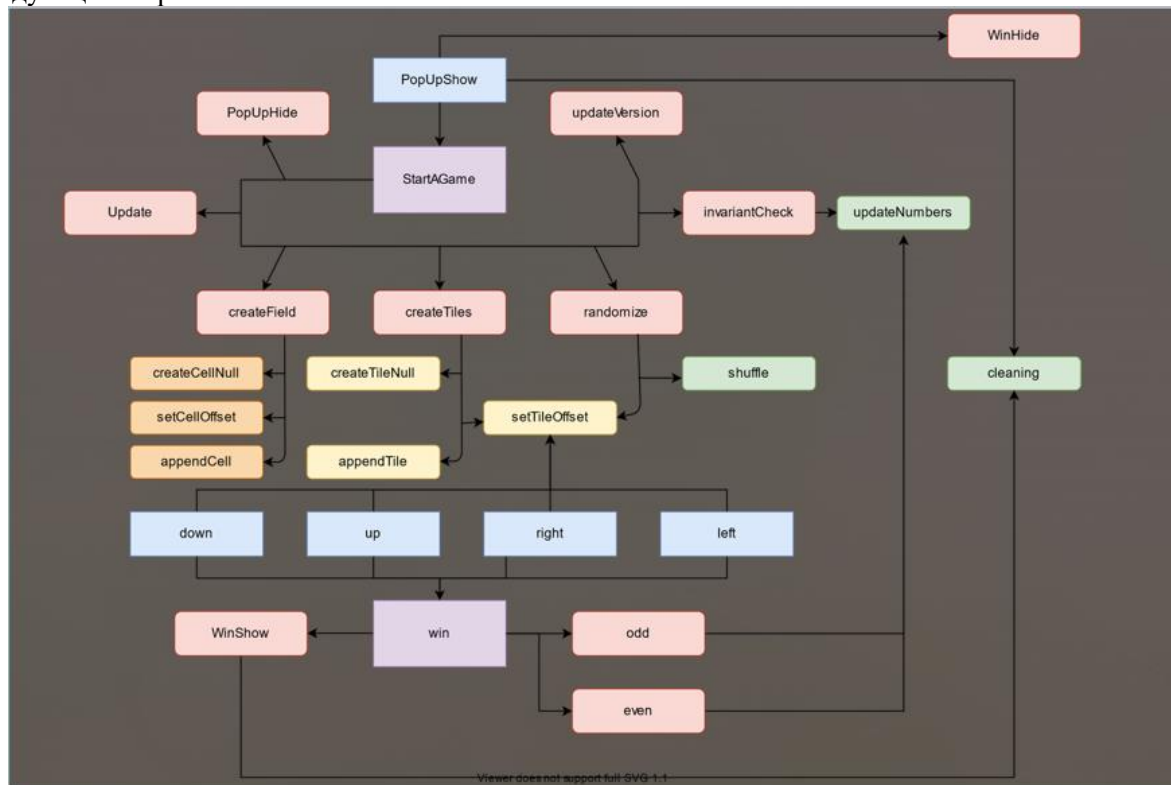
Определение концепции игры

Передо мной стояла задача создать полноценную браузерную игру средствами веб-разработки и непосредственно языка JavaScript. Мои «Пятнашки» были написаны с использованием разметки HTML-5 и библиотек стилей CSS.

Структура программы

Структуру моей программы условно можно разделить на три части: объявление глобальных переменных, объектов, массивов и строк, описание функций и обработчики событий для кнопок, которые эти функции будут запускать.

Теперь можем перейти к функциям, используемых в игре. Их визуально можно представить следующим образом:



PopUpShow, PopUpHide

Клик по кнопке «Новая игра!» вызывает функцию PopUpShow, где с помощью JQuery мы установим видимость окна приветствия. Определим наше окно видимым при загрузке страницы.

PopUpHide с помощью JQuery делает окно приветствия невидимым.

StartAGame

Функция StartAGame – это самая важная в нашей игре функция, запускаемая по клику на кнопку «Начать игру!». Внутри нее будет осуществляться запуск других функций, необходимых для работы игры.

Заберем время клика по кнопке, чтобы затем передать в таймер, как время начала игры. Внутри StartAGame запустим с небольшой задержкой встроенную функцию setInterval(), которая будет перезапускать функцию таймера через определенные промежутки времени.

Update

Функция, обновляющая значения таймера, будет работать при переданном ей начальном времени.

Чтобы найти время в игре, из текущего времени вычтем время начальное.

createField, createCellNull, setCellOffset, appendCell

В функции createField будет происходить отрисовка игрового поля с помощью двух циклов, считающих координаты от 0 до 3 по x и по y. Она запустит createCellNull, где будет создана ячейка, setCellOffset, где будет установлен сдвиг ячейки вправо и вниз относительно родительского элемента, и appendCell, которая добавит ячейку в родительский элемент.

createTiles, createTileNull, setTileOffset, appendTile

Эта функция работает аналогично функции createField с той разницей, то каждой костяшке присваивается номер – это число, отображаемое на ней, если номер не равен 16. Кроме того, добавим каждой костяшке обработчик событий по клику, который будет запускать функцию сдвига фишки. Шестнадцатая фишка так же не будет добавлена в родительский элемент, поэтому не будет отображаться в дереве.

down

Функции *down*, *right*, *left*, *up* отвечают за обработку клавиатурных событий. Функция *down* будет определять координаты свободной ячейки и костяшки, которую можно сдвинуть вниз, и, если такие есть, будет менять их координатами. Остальные функции работают аналогично.

move

Функция *move* отвечает за сдвиг костяшки в пустую ячейку при клике мышкой по ней. Она очень похожа на функции обработки клавиатурных событий.

randomize

Функция *randomize* будет перемешивать костяшки перед началом игры. Внутри нее по алгоритму «Тасование Фишера-Йетса» работает функция *shuffle*. Тасование Фишера-Йетса – это несмещенный алгоритм, так что каждая перестановка в нем генерируется с одинаковой вероятностью.

invariantCheck, updateNumbers, UpdateVersion

Запустим функцию *invariantCheck*, которая будет определять четность инварианта. Внутри функции введем числа *A* и *B*: первое будет отвечать за количество пар чисел, в которых идет сначала костяшка с большим номером, а затем с меньшим, а вторая – за строку, в которой находится пустая ячейка.

Если сумма *A* и *B* четна, то присвоим *invariant* значение *true*, в противном случае – *false*.

win

Запустим функцию *win*, которая запустит проверку положения фишек в зависимости от четности инварианта. Как только одна из функций проверки условия вернет истину, функция *win* запустит всплывающее окно победы.

При четном инварианте номера костяшек должны располагаться от 1 до 15, как принято.

При нечетном инварианте фишки от 1 до 13 должны располагаться по возрастанию, а затем должны следовать 15 и 14.

WinShow, WinHide

Как только одна из проверяющих функций вернет истину, функция *win* запустит функцию *WinShow* и с помощью JQuery сделает окно победы видимым. *WinHide* абсолютно аналогична *PopUpHide*. Вне функций определим блок победы невидимым при загрузке документа.

Для определения рекорда нам понадобятся переменная под время и под количество ходов, изначально равные 0. Если они равны 0, присвоим им результаты текущей игры, то есть время и количество ходов. Игрок может побить рекорд по времени, по количеству ходов и по времени и количеству ходов. Кнопка «Новая игра!» во всплывающем окне победы будет вызывать *PopUpShow*.

Результаты

В результате проведенных работ я создала браузерную игру, которая поддерживает всплывающие окна, плавную анимацию костяшек, два варианта развития событий (при четном и нечетном инвариантах). В ней реализован таймер решения головоломки, в конце игры выводится «Победа» и учитывается улучшение результатов. Управлять фишками можно как при помощи клавиатуры, так и мышью, а при нажатии на *Enter* запускается новая игра.

В будущем я планирую опубликовать мою игру на каком-либо хостинге, возможно, добавлю на сайт и другие головоломки, а в «Пятнашках» таблица рекордов будет обновляться для всех пользователей.

Я выполнила цель своей работы, создав полноценную игру средствами JavaScript. Игра имеет красивый дизайн и понятное управление. Я узнала много нового при работе с JavaScript и непосредственно библиотекой JQuery.

Кроме того, выполнение цели моей работы доказывает, что на языке JavaScript можно создавать хорошие браузерные приложения, не используя специализированных средств, а спектр возможностей этого языка по-настоящему велик.