

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. АНАЛИЗ ТЕХНИЧЕСКОГО ЗАДАНИЯ.....	5
2. ВЫБОР МЕТОДА РЕШЕНИЯ ЗАДАЧИ.....	7
3. ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ ЧАСТЕЙ	9
4. РАЗРАБОТКА ПРОГРАММЫ	10
5. ТЕСТИРОВАНИЕ И АНАЛИЗ РЕЗУЛЬТАТОВ.....	14
5.1 Разработка тестовых примеров.....	14
5.2 Тестирование программы	15
ЗАКЛЮЧЕНИЕ	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	20

ПРИЛОЖЕНИЕ А. СТРУКТУРНАЯ СХЕМА ПРОГРАММЫ

ПРИЛОЖЕНИЕ Б. ТЕКСТ ПРОГРАММЫ

ВВЕДЕНИЕ

Не секрет, что с появлением массы компьютерных игр и красочных игровых наборов из магазинов, игры на бумаге незаслуженно и совершенно напрасно были забыты. Многие взрослые наверняка еще помнят, как пытались скрасить скучный урок или лекцию игрой в «Пятнашки». Для этого не нужен был компьютер или планшет, а только сам набор для игры. Зато время пролетало невероятно быстро и интересно. Такие игры развивают способность к логическому мышлению, улучшают внимание, память, приучают к усидчивости и терпению.

Предположительно, история этой игры началась в конце 19 века с появлением квадратных костяшек с нанесенными числами, заключенных в квадратную коробку.

Согласно другой версии, «Пятнашки» были изобретены в 1878 году Ноем Чэпмэном, решившим таким образом отвлечь себя и товарищей от изнуряющего, тяжелого труда. Судя по всему, игра не только понравилась, но получила широкое распространение среди других слоев населения и успешно дошла до наших дней. [1]

Задача данного курсового проекта состоит в том, чтобы создать игру «Морской бой», используя язык программирования JavaScript, HTML, CSS.

Игра должна содержать:

- HTML-страница презентация игры, размещенная на github pages;
- верстка для мобильных устройств;
- css в отдельном файле;
- репозиторий на github;
- размещение в сети (heroku или любой другой сервер);

Функционал игры:

- начать новую игру;
- таймер затраченного времени;
- проверка окончания игры;
- информация о результате игры.

1 АНАЛИЗ ТЕХНИЧЕСКОГО ЗАДАНИЯ

Цель данного курсового проекта создать игру «Пятнашки», используя язык программирования JavaScript, HTML, CSS.

JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript (стандарт ECMA-262). JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам. Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса. На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке. Название «JavaScript» является зарегистрированным товарным знаком компании Oracle Corporation.

JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам – функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания – что придаёт языку дополнительную гибкость. Несмотря на схожий с Си синтаксис, JavaScript по сравнению с языком Си имеет коренные отличия: объекты с возможностью интроспекции, функции как объекты первого класса, автоматическое приведение типов, автоматическая сборка мусора, анонимные функции. В языке отсутствуют такие полезные вещи, как: стандартная библиотека: в частности, отсутствует интерфейс программирования приложений по работе с файловой системой, управлению потоками ввода-вывода, базовых типов для бинарных данных; стандартные интерфейсы к веб-серверам и базам данных; система управления пакетами, которая бы отслеживала зависимости и автоматически устанавливала их. [2]

HTML (от англ. HyperText Markup Language – «язык гипертекстовой разметки») – стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства. Язык HTML до 5-й версии определялся как приложение SGML (стандартного обобщённого языка разметки по

стандарту ISO8879). Спецификации HTML5 формулируются в терминах DOM (объектной модели документа). Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием шифрования. [3]

Данная игра носит развлекательный характер, тренирует логическое мышление.

Игра будет реализована в виде html-страницы. Для удобства работы в программе будет организован интерфейс в виде меню, так как именно эта форма интерфейса позволяет одним нажатием клавиши, не вводя никаких команд, выполнить ту или иную процедуру. В меню можно будет найти разделы «Играть», «Примеры», «Правила», «Об авторе».

Для ознакомления с краткой справочной информацией по правилам игры в программе будет реализован раздел «Правила». Естественно, если игрок забыл правила игры или вообще их не знает, то ему нужно будет прочитать правила, чтобы понять, как играть.

Раздел «Об авторе» будет включать в себя информацию об авторе разработанной программы. Этот раздел можно будет найти в меню, а перейти можно будет, нажав на название раздела.

Раздел «Играть» будет содержать саму игру «Пятнашки», то есть для того, чтобы начать играть, игрок должен будет кликнуть на раздел «Играть», чтобы перейти на html-страницу с игрой. Игра автоматически запускается, подтверждением этого будет служить запущенный таймер.

Игра будет представлять собой квадрат, содержащий игровое поле четыре на четыре и заполнено цифрами от 1 до 15.

Когда игрок захочет сыграть сначала, то необходимо будет реализовать процедуру «Заново», которая позволит игроку начать игру заново. Если же он закончил играть, либо же захотел вернуться в главное меню, то необходимо будет добавить процедуру «Меню», которая вернёт игрока к главной странице.

Как и любой программный продукт, данная программа будет иметь информационную заставку, из которой можно будет узнать название игры.

2 ВЫБОР МЕТОДА РЕШЕНИЯ ЗАДАЧИ

Данный курсовой проект реализован с помощью Visual Studio Code – редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией.

Visual Studio Code основан на Electron – фреймворк, позволяющий с использованием Node.js разрабатывать настольные приложения, которые работают на движке Blink. Несмотря на то, что редактор основан на Electron, он не использует редактор Atom. Вместо него реализуется веб-редактор Monaco, разработанный для Visual Studio Online.

Visual Studio Code – это редактор исходного кода. Он поддерживает ряд языков программирования, подсветку синтаксиса, IntelliSense, рефакторинг, отладку, навигацию по коду, поддержку Git и другие возможности. Многие возможности Visual Studio Code не доступны через графический интерфейс, зачастую они используются через палитру команд или JSON файлы (например, пользовательские настройки). Палитра команд представляет собой подобие командной строки, которая вызывается сочетанием клавиш.

Visual Studio также позволяет заменять кодовую страницу при сохранении документа, символы перевода строки и язык программирования текущего документа.

С 2018 года появилось расширение Python для Visual Studio Code с открытым исходным кодом. Оно предоставляет разработчикам широкие возможности для редактирования, отладки и тестирования кода.

Visual Studio Code имеет поддержку плагинов, доступных через Visual Studio Marketplace. Они могут включать в себя дополнения к редактору, поддержку дополнительных языков программирования, статические анализаторы кода. [4]

Цель данного курсового проекта создать игру «Пятнашки», используя язык программирования JavaScript, HTML, CSS. JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам – функции

как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания – что придаёт языку дополнительную гибкость. [2]

Язык HTML был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1986–1991 годах в стенах ЦЕРНа в Женеве в Швейцарии. HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки. HTML успешно справлялся с проблемой сложности SGML путём определения небольшого набора структурных и семантических элементов – дескрипторов. Дескрипторы также часто называют «тегами». С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в HTML внесена поддержка гипертекста. Мультимедийные возможности были добавлены позже.

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью (цветной экран современного компьютера, монохромный экран органайзера, ограниченный по размерам экран мобильного телефона или устройства и программы голосового воспроизведения текстов). Однако современное применение HTML очень далеко от его изначальной задачи. Например, тег `<table>` предназначен для создания в документах таблиц, но иногда используется и для оформления размещения элементов на странице. С течением времени основная идея платформонезависимости языка HTML была принесена в жертву современным потребностям в мультимедийном и графическом оформлении. [3]

CSS (/si:es/ англ. Cascading Style Sheets – каскадные таблицы стилей) – формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL. До появления CSS оформление веб-страниц осуществлялось исключительно средствами HTML, непосредственно внутри содержимого документа. Однако с появлением CSS стало возможным принципиальное разделение содержания и представления документа. За счёт этого нововведения стало возможным лёгкое применение единого стиля оформления для массы схожих документов, а также быстрое изменение этого оформления. Преимущества: несколько дизайнов страницы для разных устройств просмотра; уменьшение времени загрузки страниц сайта за счет переноса правил представления данных в отдельный CSS-файл; простота последующего изменения дизайна; дополнительные возможности оформления. [5]

3 ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ ЧАСТЕЙ

На основании выбранного метода решения можно выделить следующие примерные функциональные части:

- процедура открытия программы будет производиться запуском html-файла;
- организация интерфейса в виде меню с такими разделами, как «Играть», «Примеры», «Правила», «Об авторе»;
- процедура открытия новой html-страницы с игрой «Пятнашки»;
- процедура нажатия на ячейку игрового поля «Пятнашки» нажатием левой кнопки мыши на ячейку игрового поля;
- процедура перемещение цифр на игровом поле;
- процедура случайного расставления цифр на игровом поле будет осуществляться после того, как игрок нажмет на кнопку «Заново»;
- процедура «начать игру заново» будет осуществляться нажатием левой кнопки мыши на кнопку «New Game»;
- процедура завершения игры.
- процедура вывода информации о завершении игры. Будет реализована как окно, выводящее информацию о результате игры;
- процедура, осуществляющая работу таймера. Таймер будет запускаться после того, как игрок нажмёт на кнопку «New Game», автоматически перейдя в игру.

4 РАЗРАБОТКА ПРОГРАММЫ

Игра «Пятнашки» была создана при помощи языка программирования JavaScript, HTML, CSS в редакторе исходного кода Visual Studio Code.

Основные этапы проектирования игры приведены на рисунке 4.1.

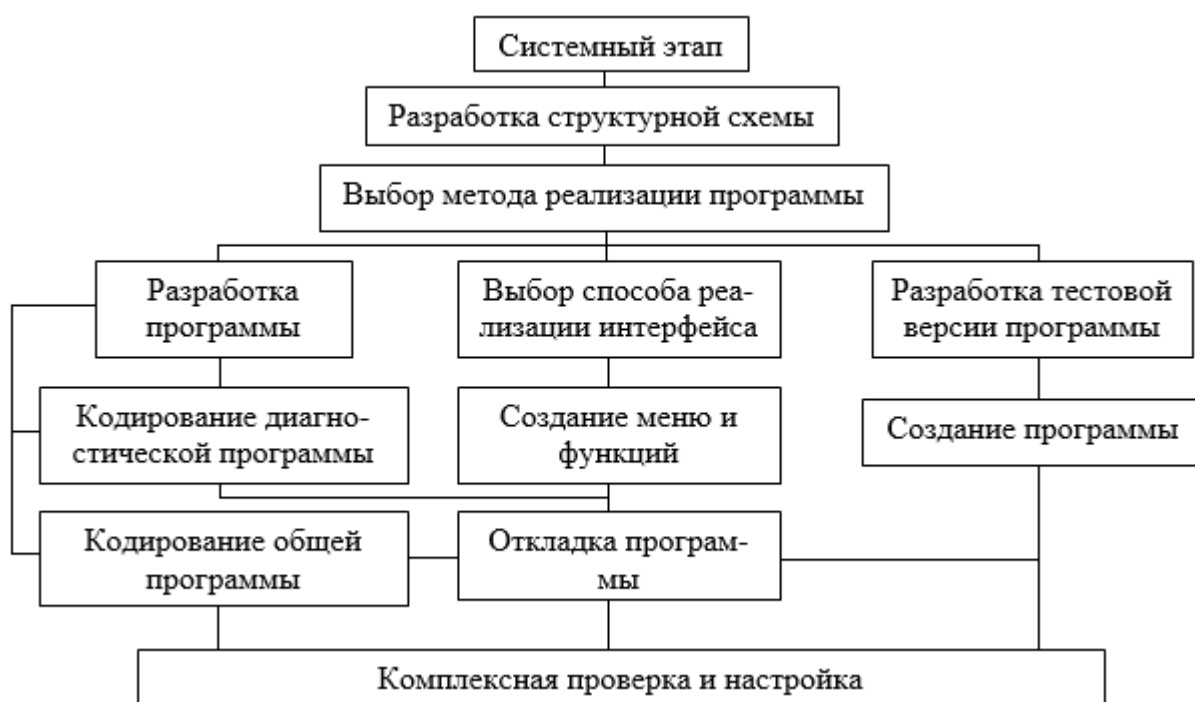


Рисунок 4.1 – Этапы проектирования

Игра для большей наглядности и структурированности состоит из ряда функциональных частей и меню, которое используется для облегчения работы с программой, т.е. для того, чтобы пользователь мог выбирать какое действие должна выполнить игра. Меню реализовано в графическом режиме.

Фрагмент кода перемещение цифры в пустую клетку приведён на рис.

4.2:

```

var event = event || window.event,
    el = event.srcElement || event.target;
i = el.id.charAt(0),
j = el.id.charAt(2);
if((i == index_i && Math.abs(j - index_j) == 1) || (j == index_j && Math.abs(i - index_i) == 1)){
    document.getElementById(index_i + " " + index_j).innerHTML = el.innerHTML;
    el.innerHTML = "";
    index_i = i;
    index_j = j;
  }

```



```

var q = true;
for(i = 0; i<4; ++i)
    for(j = 0; j<4; ++j)
        if(i + j != 6 && document.getElementById(i + " " + j).innerHTML != i*4 + j + 1){
            q = false;
            break;
        }
    if(q) alert("Victory!");
}

```

Рисунок 4.2 – Фрагмент кода перемещение цифры в пустую клетку

Фрагмент кода добавление игрового поля при загрузке страницы на рисунке 4.3:

```

window.onload = function() {
    game_field = document.getElementById("game_field");
    new_Game();
    document.getElementById("reset").onclick = new_Game;
}

```

Рисунок 4.3 – Фрагмент кода добавление игрового поля при загрузке страницы

Фрагмент кода функциональной части «Таймер» приведён на рисунке 4.4:

```

function ClearClock() {
    clearTimeout(clocktimer);
    h=1;m=1;tm=1;s=0;ts=0;ms=0;
    init=0;
    readout='00:00:00.00';
    document.MyClock.innerHTML = readout; }
function StartTIME() {
    var cdateObj = new Date();
    var t = (cdateObj.getTime() - dateObj.getTime())-(s*1000);
    if (t>999) s++;
    if (s>=(m*base)) { ts=0; m++; }
    else { ts=parseInt((ms/100)+s);
        if(ts>=base) { ts=ts-((m-1)*base); } }
    if (m>(h*base)) { tm=1; h++; }
    else { tm=parseInt((ms/100)+m);
        if(tm>=base) { tm=tm-((h-1)*base); } } ms = Math.round(t/10);
    if (ms>99) ms=0;
    if (ms==0) ms='00';
    if (ms>0&&ms<=9) ms = '0'+ms;
    if (ts>0) { ds = ts;
        if (ts<10) ds = '0'+ts; }
    else

```

```

ds = '00'; dm=tm-1;
if (dm>0) {
    if (dm<10) dm = '0'+dm; }
else dm = '00'; dh=h-1;
if (dh>0) {
    if (dh<10) dh = '0'+dh; }
else
dh = '00';
readout = dh + ':' + dm + ':' + ds + '.' + ms;
document.MyClock.innerHTML = readout;
clocktimer = setTimeout("StartTIME()",1); }
function StartStop() {
    if (init==0) {
        ClearClock();
        dateObj = new Date();
        StartTIME();
        init=1; }
    else {
        clearTimeout(clocktimer);
        init=0;}
}

```

Рисунок 4.4 – Фрагмент кода функциональной части «Таймер»

В программе использовались такие структуры, как циклы, которые организовывались таким оператором, как `for`.

С помощью цикла `for` можно выполнять оператор или блока выписок повторно до тех пор, пока указанное выражение не будет оценки к `false`. Этот тип цикла полезен для перебора массивами и для других приложений, в которых известны заранее, сколько раз необходимо цикл повторных.

При использовании в качестве типа возвращаемого значения для метода, `void` указывает, что метод не возвращает значение.

Оператор `if` служит для того, чтобы выполнить какую-либо операцию в том случае, когда условие является верным. Условная конструкция всегда записывается в круглых скобках после оператора `if`.

Внутри фигурных скобок указывается тело условия. Если условие выполнится, то начнется выполнение всех команд, которые находятся между фигурными скобками.

Конструкция `switch` заменяет собой сразу несколько `if`. Она представляет собой более наглядный способ сравнить выражение сразу с несколькими вариантами. Переменная `x` проверяется на строгое равенство первому значению `value1`, затем второму `value2` и так далее. Если соответствие установлено – `switch` начинает выполняться от соответствующей директивы `case` и далее, до ближайшего `break` (или до конца `switch`). Если ни один `case` не совпал – выполняется (если есть) вариант `default`. При этом `case` называют вариантами `switch`.

Зачастую нам надо повторять одно и то же действие во многих частях программы. Например, красиво вывести сообщение необходимо при

приветствии посетителя, при выходе посетителя с сайта, ещё где-нибудь. Чтобы не повторять один и тот же код во многих местах, придуманы функции (function). Функции являются основными «строительными блоками» программы. Примеры встроенных функций – это `alert(message)`, `prompt(message, default)` и `confirm(question)`. Но можно создавать и свои. Объявление функции происходит следующим образом: вначале идет ключевое слово `function`, после него имя функции, затем список параметров в скобках и тело функции – код, который выполняется при её вызове.

Функция может содержать локальные переменные, объявленные через `var`. Такие переменные видны только внутри функции. Блоки `if/else`, `switch`, `for`, `while`, `do..while` не влияют на область видимости переменных. При объявлении переменной в таких блоках, она всё равно будет видна во всей функции.

Функция может вернуть результат, который будет передан в вызвавший её код. Для возврата значения используется директива `return`. Она может находиться в любом месте функции. Как только до неё доходит управление – функция завершается и значение передается обратно. Директива `return` может также использоваться без значения, чтобы прекратить выполнение и выйти из функции.

Имя функции следует тем же правилам, что и имя переменной. Основное отличие – оно должно быть глаголом, т.к. функция – это действие. Как правило, используются глагольные префиксы, обозначающие общий характер действия, после которых следует уточнение. Функции, которые начинаются с `"show"` – что-то показывают. Функции, начинающиеся с `"get"` – получают, и т.п. Это очень удобно, поскольку, взглянув на функцию – мы уже примерно представляем, что она делает, даже если функцию написал совсем другой человек, а в отдельных случаях – и какого вида значение она возвращает.

Методы для создания узлов: `document.createElement(tag)` – создает элемент, `document.createTextNode(value)` – создает текстовый узел, `elem.cloneNode(deep)` – копирует элемент, если `deep == true`, то со всеми потомками, если `false` – без потомков. Вставка и удаление узлов: `parent.appendChild(elem)`, `parent.insertBefore(elem, nextSibling)`, `parent.removeChild(elem)`, `parent.replaceChild(newElem, elem)`. Все эти методы возвращают `elem`.

Схема алгоритма программы расположена в Приложении А.

Текст программы расположен в Приложении Б.

5 ТЕСТИРОВАНИЕ И АНАЛИЗ РЕЗУЛЬТАТОВ

5.1 Разработка тестовых примеров

Тест №1. Первый запуск и выход из программы

Запуск программы на экран осуществляется нажатием на html-страницу index.html. Должно произойти открытие страницы в браузере с основной информацией об игре и меню. Чтобы перейти к самой игре, нужно нажать на кнопку «Играть». В итоге должна открыться страница с игрой Пятнашки.

Для выхода необходимо нажать кнопку «X», после чего web-страница закроется.

Тест №2. Начало игры. Ход игры

Чтобы начать играть в «Пятнашки», необходимо нажать на ячейку игрового поля. После нажатия на ячейку цифра переместится в свободную ячейку.

Тест №3. Заново

Допустим, игрок завершил игру, тогда необходимо начать игру заново. Для этого необходимо нажать на кнопку «New Game». После нажатия кнопки поле придёт к первоначальному виду, и игрок сможет заново начать играть.

Тест №4. Таймер

Когда игрок сыграет в «Пятнашки», ему будет интересно за какой промежуток времени он это сделал. Для этого необходимо добавить таймер. Таймер должен включиться, как только игрок нажмёт на кнопку «Играть», перейдя в саму игру.

Тест №5. Окончание игры. Вывод результата

После того, как игрок расставит цифры в правильно порядке, необходимо вывести результат о выигрыше, количестве затраченных ходов и затраченном времени. Также предложить начать заново.

5.2 Тестирование программы

Тест №1. Первый запуск и выход из программы

Запуск программы на экран осуществляется нажатием на html-страницу index-html. Должно произойти открытие страницы в браузере с основной информацией об игре и меню (рис.5.1):



Рисунок 5.1 – Открытая html-страница в браузере

Чтобы перейти к самой игре, нужно нажать на кнопку «Play Game» (рис.5.2):



Рисунок 5.2 – Нажатие на кнопку «Play Game»

В итоге должна открыться страница с игрой Пятнашки (рис.5.3):

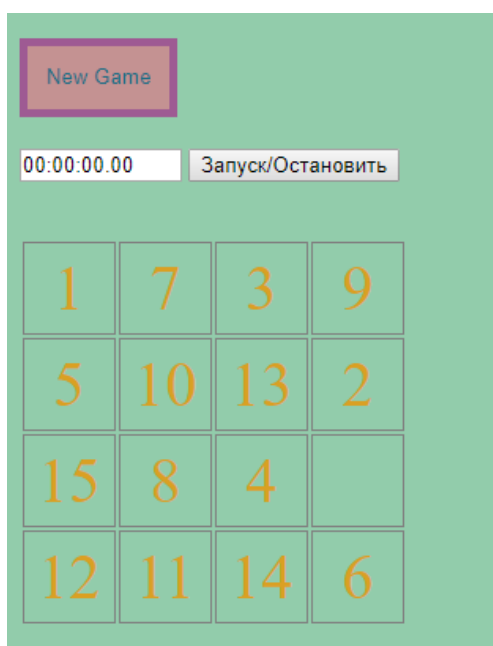


Рисунок 5.3 – Страница с игрой «Пятнашки»

Тест №2. Начало игры. Ход игры

Чтобы начать играть в «Пятнашки», необходимо нажать на ячейку с цифрой, которая находится рядом с пустой клеткой. (рис.5.4):

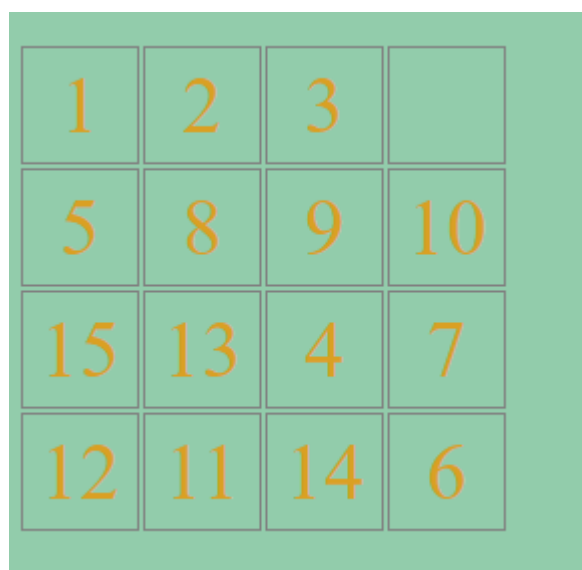


Рисунок 5.4 – Ход игры

Тест №3. Заново

Допустим, игрок завершил игру, тогда необходимо начать игру заново. Для этого необходимо нажать на кнопку «New Game» (рис 5.5):

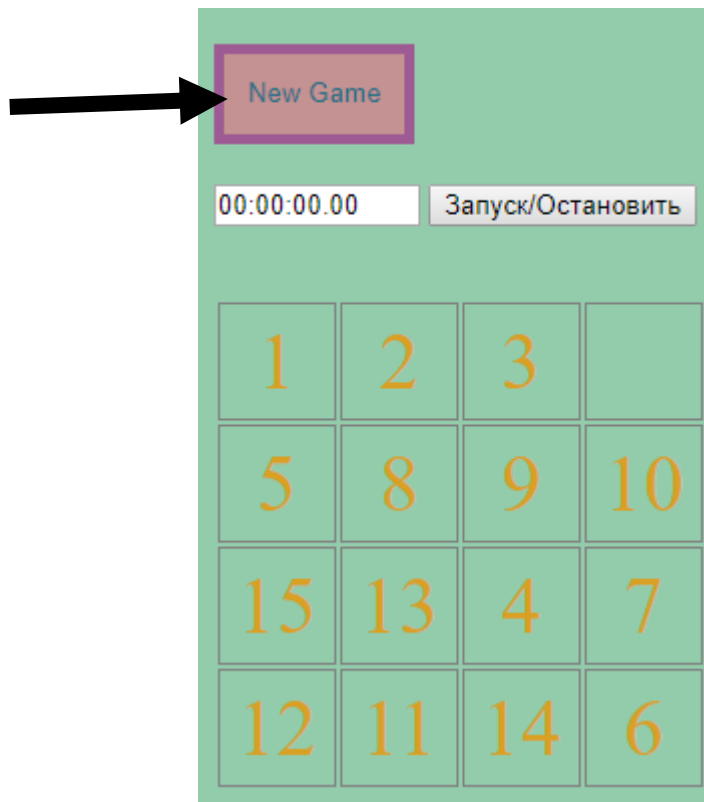


Рисунок 5.5 – Нажатие на кнопку New Game

Тест №5. Таймер

Когда человек играет в «Пятнашки», ему интересно за какой промежуток времени он это сделал. Для этого необходимо добавить таймер. Таймер должен включиться, как только игрок нажмёт на кнопку «Запуск», перейдя в саму игру (рис 5.6):

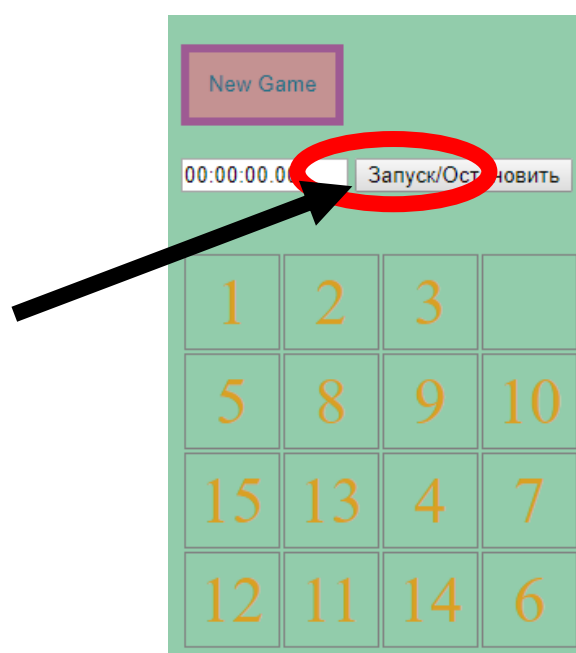


Рисунок 5.6 – Таймер

Тест №6. Окончание игры. Вывод результата

После того, как игрок соберет правильную последовательность из цифр, необходимо вывести результат о выигрыше, и затраченном времени. (рис 5.7):

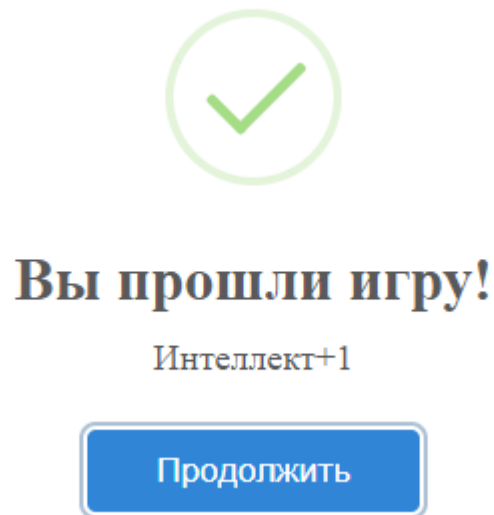


Рисунок 5.7 – Окончание игры

ЗАКЛЮЧЕНИЕ

В результате проделанной работы был разработан алгоритм и написана сама программа, реализующая работу игры «Пятнашки», которая была создана при помощи среды быстрой разработки приложений – Visual Studio Code с использованием языка программирования JavaScript, HTML, CSS. Данная программа реализует запуск новой игры, проверку на наличие пустой клетки рядом, вывод данных о выигрыше, затраченном времени, также html-страницу с меню, в котором есть краткая информация об игре, правилах игры и авторе игры.

Для запуска программы необходимо открыть html-файл index.html.

Данная программа может быть использована в учебных целях.

СПИСОК ЛИТЕРАТУРЫ

1. Введение [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/15>. – Дата доступа 04.11.18.
2. JavaScript [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/JavaScript>. – Дата доступа 14.11.18.
3. HTML [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/HTML>. – Дата доступа 14.11.18.
4. Visual Studio Code [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/Visual_Studio_Code. – Дата доступа 14.11.18.
5. CSS [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/CSS>. – Дата доступа 14.11.18.