|  |  |
| --- | --- |
| Министерство образования и науки Российской Федерации  Федеральное государственное автономное образовательное учреждение высшего образования «Российский государственный профессионально-педагогический университет» | |
| Институт инженерно-педагогического образования  Кафедра информационных систем и технологий | |
|  | |
| **КУРСОВАЯ РАБОТА**  по дисциплине  **«ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ»** | |
| на тему  **«Разработка игры: «Морской бой»»** | |
|  | |
| **Направление подготовки** 09.03.02 Информационные системы и технологии  **Профиль** Информационные технологии в медиаиндустрии | |
|  |  |
| Выполнил: | Студенты группы ИТм-403  Мустакимова М.В.  Захарова Д.В.  Максимова А.В.  Куноф П.А.  Стройкина Е.С |
|  |  |
| Проверил: | Старший преподаватель  Черноскутов М.Ю. |
|  |  |
| Екатеринбург 2018 | |

Содержание

[Введение 4](#_Toc535467891)

[1 Предпроектное описание игры «Морской бой» 6](#_Toc535467892)

[1.1 Анализ предметной области 6](#_Toc535467893)

[1.2 Обзор аналогов 7](#_Toc535467894)

[1.3 Постановка задачи 9](#_Toc535467895)

[2 Техническое задание игры «Морской бой» 11](#_Toc535467896)

[3 Проектирование интерфейса игры «Морской бой» 14](#_Toc535467897)

[3.1 Выбор инструмента 14](#_Toc535467898)

[3.2 План работы 15](#_Toc535467899)

[4 Звуковые эффекты игры «Морской бой» 20](#_Toc535467900)

[5 Разработка анимаций игры «Морской бой» 24](#_Toc535467901)

[6 Разработка програмного кода игры «Морской бой» 29](#_Toc535467902)

[6.1 Выбор средств разработки 29](#_Toc535467903)

[6.2 Описание клиентской части программы 30](#_Toc535467904)

[6.3 Описание функций серверной части программы 31](#_Toc535467905)

[6.4 Алгоритм запуска программы 33](#_Toc535467906)

[7 Тестирование программного обеспечения 35](#_Toc535467907)

[7.1 Понятие тестирования 35](#_Toc535467908)

[7.2 Результаты тестирования 39](#_Toc535467909)

[8 Документация 45](#_Toc535467910)

[8.1 Руководство программиста 45](#_Toc535467911)

[8.1.1 Назначение и условия применения программы 45](#_Toc535467912)

[8.1.2 Характеристика программы 46](#_Toc535467913)

[8.1.3 Обращение к программе 46](#_Toc535467914)

[8.1.4 Входные и выходные данные 46](#_Toc535467915)

[8.2 Инструкция пользователя 46](#_Toc535467916)

[Заключение 49](#_Toc535467917)

[Список использованных источников 50](#_Toc535467918)

Введение

Данная работа посвящена разработке программного продукта, реализующего классическую версию настольной игры «Морской бой».

После проведения анализа предметной области была подтверждена необходимость разработки данного программного продукта. Целевой аудиторией будут являться люди любого возраста, желающие сыграть в «морской бой».

При выполнении работы были рассмотрены различные аналогичные решения, и, после выделения и сравнения их основного функционала и особенностей, были выработаны основные идеи проекта, способные сделать разрабатываемый продукт конкурентоспособным. Игра будет поддерживать полную визуализацию игрового процесса, можно играть с другом по локальной сети поделившись ссылкой. Выделяться среди конкурентов игра будет за счет удобного, красочного и выполненного в космической тематике интерфейса без лишних элементов и рекламы, также за счет подобранных звуковых эффектов помогающим пользователям «погрузиться» в атмосферу игры. Ни один из рассмотренных аналогов не выполняет все эти идеи, что должно повысить интерес среди людей к данному программному продукту.

Объект курсовой работы – игра-стратегия для отдыха и совместного времяпровождения с друзьями.

Предмет курсовой работы – разработка игры «Морской бой».

Целью работы является создание и разработка игры «Морской бой» с музыкальной подборкой и индивидуальным интерфейсом.

В соответствии с поставленной целью в работе определены следующие задачи:

1. Анализ предметной области
2. Обзор аналогов продукта
3. Техническое задание
4. Разработка интерфейса
5. Поиск звукового сопровождения
6. Разработка анимаций
7. Разработка программного продукта
8. Тестирование программного продукта
9. Составление документации

1 Предпроектное описание игры «Морской бой»

1.1 Анализ предметной области

Изначальный вариант игры «Морской бой» это настольная игра на двух участников, которые по очереди выбирают координаты на неизвестном им поле противника. Поле – это набор клеток размеров 10 на 10, на которые устанавливаются корабли. Если у соперника по этим координатам имеется корабль, то корабль или его часть «топится». Цель – первым поразить все корабли соперника.

Всего кораблей в игре 10: 1 четырёхпалубный, 2 трёхпалубных, 3 двухпалубных и 4 однопалубных. Между кораблями должно быть расстояние как минимум 1 пустая клетка.

Данный проект будет представлять собой приложение, реализующее игру «морской бой» по локальной сети с другим участником. Проект будет конкурировать не только с другими реализациями морского боя, но и с другими простыми логическими играми, такими как, например, «крестики-нолики» или «сапёр».

Целевую аудиторию составляют люди, которые хотят играть в «морской бой». Игроки будут иметь возможность играть со своими друзьями и знакомыми.

Проект будет отличаться от конкурентов отсутствием ненужных элементов, которые неблагоприятно сказываются на внешнем виде программы. Также игра будет отличаться интересной концепцией, которая реализована за счёт тематического графического интерфейса и музыкальных эффектов.

Данный проект будет полезен для тех, кто хочет немного отдохнуть и приятно провести время, сыграв пару боёв в космический «морской бой». Для игры не требуются какие-то специфические знания компьютера, поэтому играть в неё сможет абсолютно любой человек.

1.2 Обзор аналогов

В настоящее время уже существует большое количество игр-стратегий «Морской бой». Для разработки конкурентоспособной игры, были проанализированы аналогичные игры:

* Battleships – GeneralQuartersII.
* Battleship The Beginning.
* Naval Clash.
* Морской бой, разработанный по заказу Министерства обороны РФ.
* «Морской бой» на ОС Android.

Battleships – GeneralQuartersII.

Игра «Морской бой», выполненная не в классической форме. Всего можно расставить 5 кораблей: один на 2 клетки, два на 3 клетки, один на 4 клетки и один на 5 клеток. Корабли можно расставлять вплотную друг к другу. После каждого попадания на короткое время появляется комментарий от капитана.

Ссылка на официальную веб-страницу: http://www.ubattleships.com/

Battleship The Beginning.

Объёмный «Морской бой» с приятной графикой и звуковым сопровождением. Главной особенностью является наличие различных способностей, таких как подзорная труба, горящий снаряд, взрывные бочки, залповый обстрел. Также игра имеет кампанию, в ходе которой зарабатываются очки на приобретение способностей.

Ссылка на официальную веб-страницу: http://www.coolbuddy.com/games/game.asp?gid=2128

Naval Clash.

Классический «Морской бой» на ОС Android со стилистикой рисовки ручкой по бумаге. Присутствует возможность играть по Bluetooth или по сети Интернет. Довольно быстрый игровой процесс.

Ссылка на официальную веб-страницу: https://play.google.com/store/apps/details?id=northern.captain.seabattle

Морской бой, разработанный по заказу Министерства обороны РФ.

Игра сделана в классическом стиле. Несмотря на огромные расходы, потраченные на разработку, игра имеет не очень дружелюбный интерфейс. Не отмечаются клетки вокруг уничтоженного корабля. Очень быстрый игровой процесс.

Ссылка на официальную веб-страницу: http://мультимедиа.минобороны.рф/multimedia/games/seabattle.htm

BattleShip Hunter.

«Морской бой» на ОС Android. Не классический стиль. Игровой процесс имеет очень простое оформление: при стрельбе по полю отмечаются либо зелёные, либо красные кружочки, в зависимости от того, было ли попадание.

Ссылка на официальную веб-страницу: https://apkpure.com/ru/warship-hunter/go.english.BattleShipHunter

Для сравнения аналогов были выбраны несколько параметров:

* Классический стиль игры.
* Дружелюбный интерфейс.
* Возможность игры по сети.
* Звуковое сопровождение.
* Удобная расстановка кораблей.

Дружелюбный интерфейс подразумевает интуитивную простоту и отсутствие различных ненужных элементов. Этот параметр очень сильно влияет на впечатление от игры и иногда определяет дальнейшее желание в неё поиграть.

Возможность игры по сети. Наличие реального противника для такой игры, как Морской бой является значимым элементом.

Звуковое сопровождение не обязательный, но имеющий большое значение параметр, который приятно сказывается на впечатлении от игры, а также лучшей ассоциацией с данной игрой.

Удобная расстановка кораблей подразумевает наличие разметки или выделение пространства, куда можно поставить корабль, удобный выбор кораблей из списка и так далее. Плохая реализация отрицательно сказывается на впечатлении об игре.

Сравнение аналогов представлено в таблице 1.

Таблица 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Признаки сравнения | Battleships – GeneralQuartersII | Battleship The Beginning | NavalClash | Морской бой, разработанный по заказу Министерства обороны РФ | BattleShipHunter |
| Классический  стиль | – | – | + | + | – |
| Дружелюбный  интерфейс | + | + | + | – | – |
| Возможность  игры по сети | – | – | + | – | – |
| Звуковое сопровождение | + | + | – | – | + |
| Удобная расстановка кораблей | + | + | + | – | – |

Из проведённого обзора следует, что в играх такого рода редко реализовывается тематический интерфейс, который может заинтересовать пользователя «нестандартным» подходом к созданию графической составляющей. В некоторых играх присутствует либо навязчивая реклама, либо лишние элементы, которые только портят впечатление от игры. Почти у всех рассмотренных аналогов присутствует звуковое сопровождение, служащее для создания атмосферы, однако не везде оно приятное, также как и чаще всего очень похожее друг на друга.

1.3 Постановка задачи

Исходя из анализа предметной области можно сделать вывод, что у данного типа приложений есть спрос. Все реализации морского боя имеют свои недостатки и преимущества, так что разработка этого проекта будет иметь успех, если он не будет в себе содержать основные недостатки аналогичных решений, а также будет иметь свои уникальные особенности и нововведения.

Основные идеи проекта:

1. Возможность играть с другом по сети.
2. Интерфейс выполнен в космической тематике.
3. Звуковое сопровождение, подходящее по тематике интерфейса, запоминающаяся фоновая музыка и погружающие в атмосферу игры звуковые эффекты.
4. Все действия будут визуализированы: перетаскивание кораблей, наведение на клетки и надписи, результаты выстрелов.

2 Техническое задание игры «Морской бой»

Наименование программы – компьютерная многопользовательская игра в жанре стратегия, морской бой.

Краткая характеристика области применения программы или программного изделия: данная программа предназначена для приятного времяпровождения пользователя в свободное время, для эксплуатации на персональных компьютерах пользователей.

Основанием для проведения разработки являются документы, разработанные и утверждённые Российским государственным профессионально-педагогическим университетом. В список документов входит:

1. Учебный план.
2. Устав Университета.
3. Положение о курсовой работе.

Наименование и условное обозначение темы разработки – «Создание компьютерной игры «Морской бой» (далее по тексту – Игра)».

Функциональное назначение программы: предоставление пользователю возможности проведения досуга в игровой форме.

Эксплуатационное назначение: использование на подходящих под минимальные системные требования персональных компьютерах. Конечные потребители – обычные люди, не нуждающиеся в дополнительном инструктаже и обучении.

Игра должна обеспечивать возможность выполнения следующих функций:

1. Копирование ссылки в буфер обмена.
2. Управление боем.
3. Перемещение кораблей по полю.
4. Атака.
5. Воспроизведение звуковых эффектов.
6. Воспроизведение фоновой музыки.
7. Возможность вернуться в главное меню.
8. Метод ввода – манипулятор мышь.
9. Организация выходных данных (методы вывода) – монитор, аудиосистема.

Программа должна быть устойчива к ошибкам, то есть не допускать их возникновения.

Условия эксплуатации:

* Температура окружающего воздуха: от 5 до 28 градусов Цельсия.
* Относительная влажность: от 40 до 70%.

Требования к составу и параметрам технических средств.

Минимальная конфигурация ПК:

1. частота процессора не менее 1 ГГц;
2. операционная система Windows 7 или выше;
3. 1024 Мбайт оперативной памяти;
4. манипулятор мышь;
5. не менее 920 Мбайт свободного пространства на диске;
6. рабочее разрешение экрана, 1280x720 точек;
7. звуковая карта, колонки и/или наушники;
8. сетевая карта.

Требования к информационной и программной совместимости: исходные коды программы должны быть реализованы на мультипарадигменном языке программирования javaScript на платформе node.js. Используются также: язык гипертекстовой разметки страницы - html и каскадные таблицы стилей - css. Требования программной совместимости: программа должна обеспечивать совместимость с ОС Windows.

Требования к маркировке и упаковке не регламентируются ввиду планируемого распространения программы без физического носителя.

Требования к транспортированию и хранению – ссылка в общем доступе на Игру в облачном хранилище.

Требования к программной документации:

* Пользовательское лицензионное соглашение.
* Руководство пользователя.
* Техническое задание.

Технико-экономические показатели:

* Ориентировочная экономическая эффективность не рассчитывается в виду некоммерческого статуса программы;
* Предполагаемая годовая потребность: 365 запусков по 20 мин в среднем;
* Экономические преимущества разработки по сравнения с лучшими отечественными и зарубежным образцами не рассчитываются в виду некоммерческого статуса программы.

Стадии и этапы разработки:

1. Разработка технического задания.
2. Разработка Игры.
3. Тестирование Игры.
4. Доработка документации.

Сроки разработки – до 20 декабря 2018 года.

Исполнителями являются студенты группы ИТм-403:

* Мустакимова Мария Владимировна – ведение документации, звуковое сопровождение.
* Куноф Павел Александрович – ведущий разработчик.
* Захарова Дарья Владимировна – цифровой художник, дизайнер, арт-директор.
* Максимова Анна Сергеевна – тестировщик.
* Стройкина Екатерина Сергеевна – разработчик анимаций.

3 Проектирование интерфейса игры «Морской бой»

3.1 Выбор инструмента

В настоящее время существует большое количество инструментов для создания прототипов сайтов и веб-приложений. Кроме современных профильных приложений большой популярностью пользуются тяжёлые универсальные инструменты, например Adobe Photoshop. Таким образом был составлен список инструментов, подходящих для решения поставленной задачи:

* Adobe Photoshop.
* Adobe Illustrator.
* CorelDraw.
* Sketch.
* Figma.

Adobe Photoshop - является многофункциональным графическим редактором, который позволяет работать как с растровой, так и с векторной графикой. Однако основной инструментарий рассчитан именно для работы с растром. Распространяется с помощью платной подписки Adobe Creative Cloud, стоимостью 1288 рублей в месяц.

Adobe Illustrator - инструмент для работы с исключительно векторной графикой. Также распространяется с помощью Adobe Creative Cloud.

CorelDraw - другой продукт для работы с векторной графикой, разработанный компанией-конкурентом. Входит в пакет программного обеспечения CorelDRAW Graphics Suite, стоимость годовой подписки которого составляет 13930 рублей.

Sketch - графическией редактор векторной графики, разработанный для MacOS. Пользуется широкой популярностью для создания прототипов и макетов веб-сайтов, а также веб и мобильных приложений. Годовая подписка составляет 99$.

Figma - это веб-приложение для создания макетов и прототипов, работающее в облаке, что позволяет использовать его независимо от платформы. Имеет как платную, так и бесплатную подписки. Бесплатная подписка ограничена лишь длиной истории и максимальным количеством людей в команде.

Для реализации данного проекта была выбрана Figma, так как данный инструмент является бесплатным, позволяет сохранять проект в облаке и открыть его с любого устройства. Также, основным преимуществом является возможность просматривать и работать над проектом сразу нескольким людям. Это позволяет другим членам команды наблюдать изменения в проекте в реальном времени, а также, получать семантически верный код, генерируемый инструментом.

3.2 План работы

Для реализации графической составляющей проекта была составлена последовательность действий:

* Поиск референсов для формулирования стилистики проекта.
* Поиск свободных для использования материалов, применимых для создания графической части проекта, соответствующие выбранной стилистике.
* Создание недостающих графических элементов.
* Редактирование имеющихся компонентов и объединение их в единую систему.
* Создание прототипа на основе разработанной дизайн-системы.

Изначальной идеей проекта было создание игры “Морской бой” в необычной для этого жанра стилистике. Для выбора подходящего сеттинга были проанализированы различные браузерные игры. Наиболее подходящей стилистикой в рамках заданного жанра была выбрана тема космических боёв.

Для ускорения разработки проекта было принято решение использовать бесплатно доступные векторные изображения. Найденные материалы представлены на рисунках 1, 2.

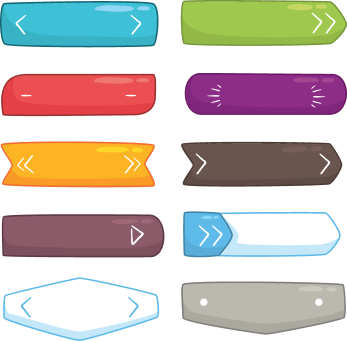


Рисунок 1 – бесплатно доступные векторные изображения.

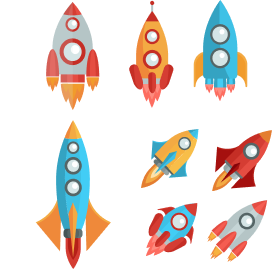


Рисунок 2 – бесплатно доступные векторные изображения.

Также, в процессе создания проекта были отрисованы уникальные элементы, представленные на рисунках 3, 4.

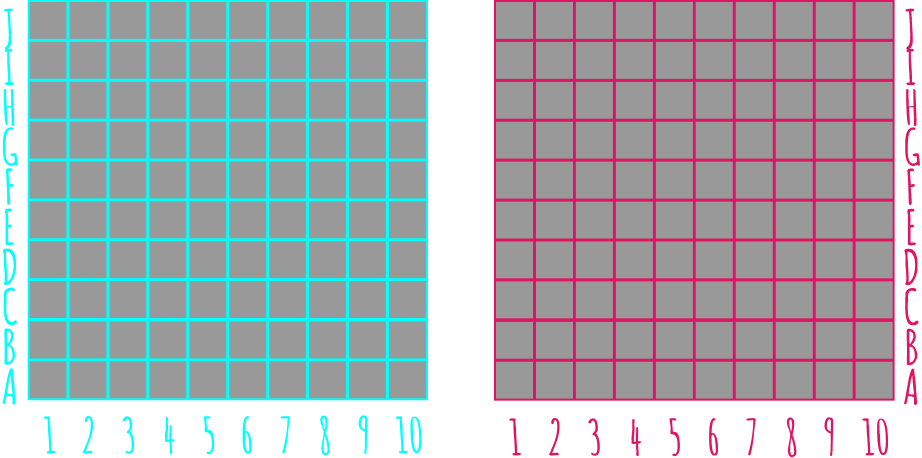


Рисунок 3 – уникальные элементы.



Рисунок 4 – уникальные элементы.

Далее был разработан дизайн-система (рисунок 5) и собран прототип, некоторые экраны собранного прототипа представлены на рисунках 6, 7.

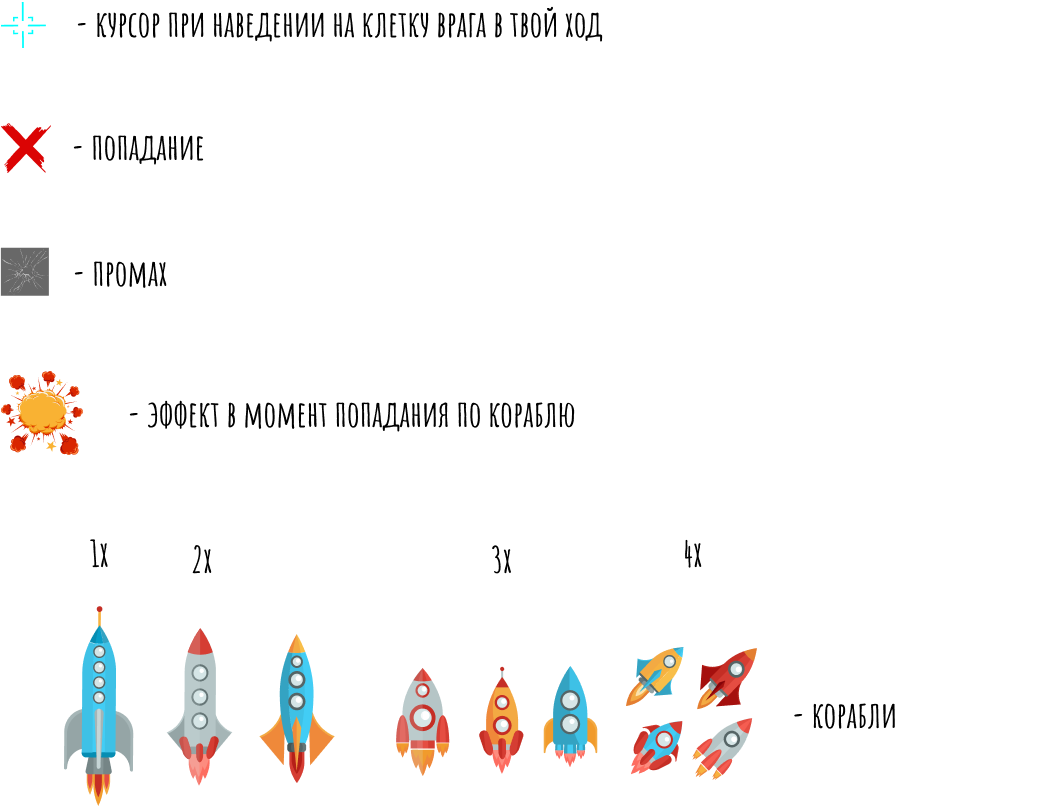


Рисунок 5 – дизайн-система.



Рисунок 6 – экран собранного прототипа.

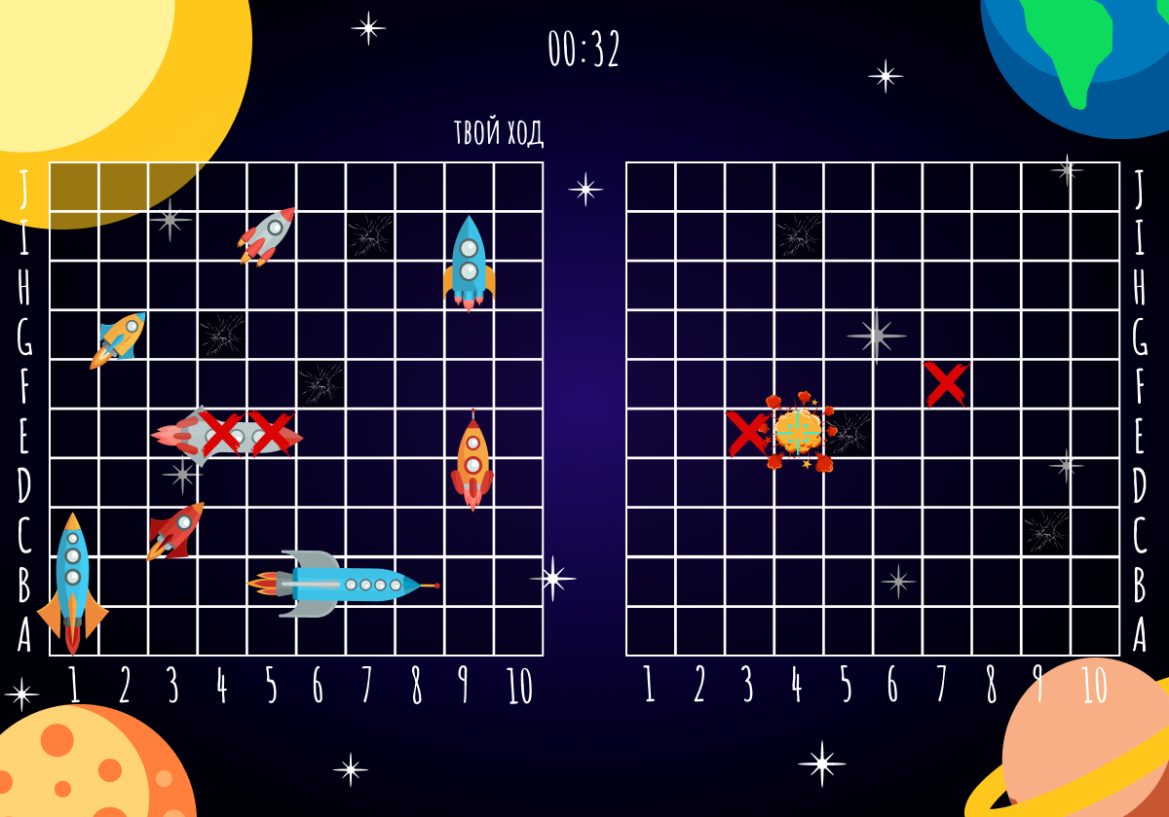


Рисунок 7 – экран собранного прототипа.

4 Звуковые эффекты игры «Морской бой»

Изначально для поиска звуковых эффектов надо определиться с: тематикой для чего именно они подбираются, продолжительностью аудиофайлов, для фоновой музыки плавной сменой «начало-конец», то есть цикличностью. Также аудиофайлы должны быть без авторских прав, для бесплатного использования. Изначально были отобраны, загружены на личный ПК и прослушаны 120 звуковых эффектов, после нескольких прослушиваний каждого трека был отобран «лист» треков необходимых эффектов наиболее подходящих для использования.

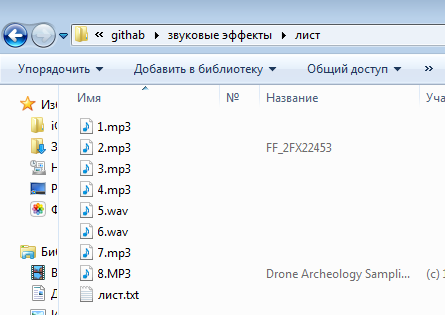


Рисунок 8 – отсортированные звуковые эффекты.

Дальнейшая работа с аудиофайлами осуществляется в программе Adobe Audition – программе для профессионального редактирования, микширования, записи и создания аудио контента без потери его качества. Adobe Audition предоставляет в распоряжение все средства, необходимые для быстрой и эффективной работы. Удобный интерфейс позволяет выполнять необходимые операции в кратчайшие сроки, а плавающие рабочие панели обеспечивают быструю настройку рабочего пространства. Также возможна работа с аудио-файлами во всех распространенных форматах (включая WAV, AIFF, MP3, mp3PRO и WMA).



Рисунок 9 – программа Adobe Audition.

Для обработки звуков, загружаем аудиофайлы в программу. В Adobe Audition увеличиваем или же уменьшаем громкость нужных нам участков, для более приятного прослушивания.

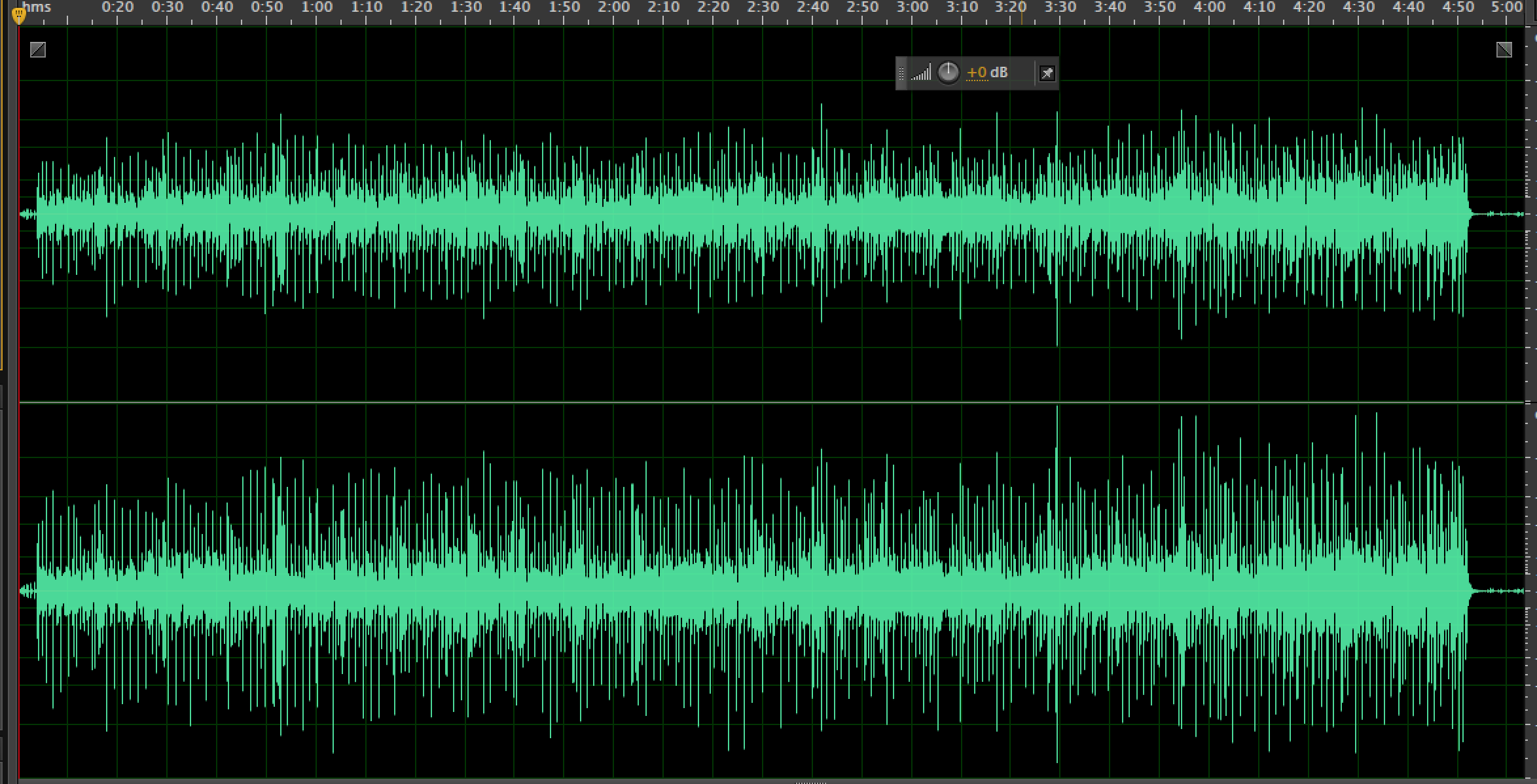


Рисунок 10 – увеличение и уменьшение звука аудиодорожек.

Далее проверяем аудиодорожки на наличие шумов, при слишком сильных и постоянных шумах уменьшаем их или же вовсе удаляем, оставляя нужную часть аудиотрека.

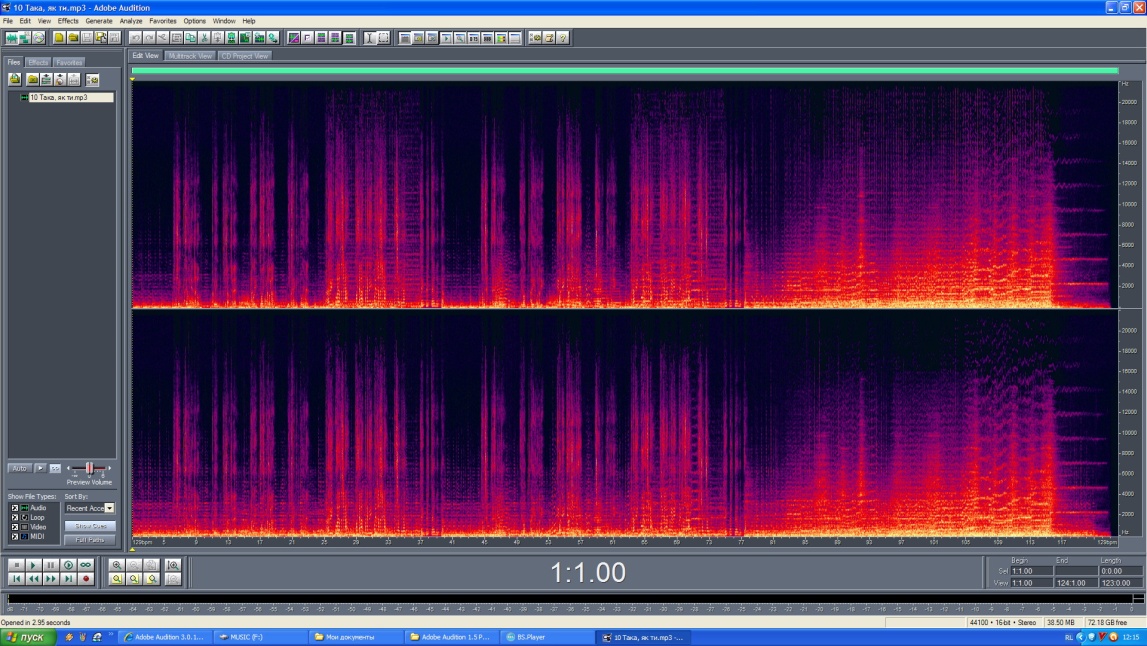


Рисунок 11 – редактирование шумов.

После редактирования всех треков по отдельности и их настройки. Необходимо сделать плавное появление или же наоборот, затухания аудио.

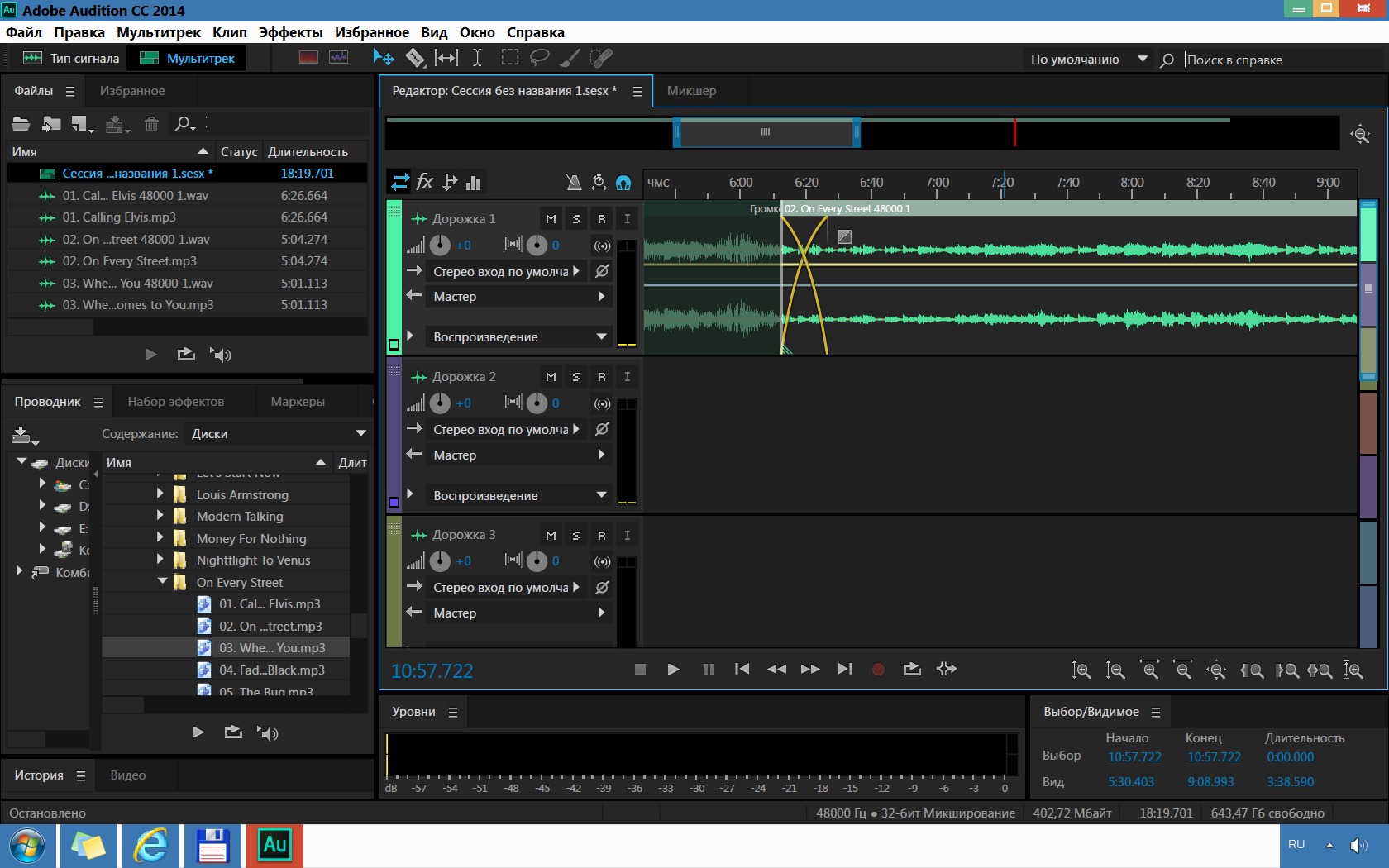


Рисунок 12 – применение эффектов затухания.

После сохранения аудиофайлов их легко можно открыть в плеере на любом ПК. Отредактированные файлы для дальнейшей работы необходимо загрузить в облачное хранилище, а ссылку передать ведущему разработчику.

5 Разработка анимаций игры «Морской бой»

Роль медиатехнологий в передаче информации увеличивается. Постепенно текстовая информация уступает часть позиций графической и мультимедийной. Компьютерная анимация - это процесс, используемый для создания анимированных изображений.

Были рассмотрены несколько популярных инструментов подходящих для решения создания анимации:

* Canvas.
* GIF.
* CSS.

Canvas – HTML5 элемент, с помощью которого можно добавлять растровую графику в код через JavaScript. CANVAS имеет особенность - производительность его элементов, скорость обработки и т.п. зависит напрямую от возможностей браузера и компьютера.

GIF – один из самых популярных форматов растровых графических изображений. Основной его минус заключается в том, что он не поддерживает векторную графику.

CSS – технология описания внешнего вида документа, оформленного языком разметки. CSS3-анимация придаёт сайтам динамичность. Она оживляет веб-страницы, улучшая взаимодействие с пользователем. Cоздание анимации базируется на ключевых кадрах, которые позволяют автоматически воспроизводить и повторять эффекты на протяжении заданного времени, а также останавливать анимацию внутри цикла.

Для создания данного проекта был выбран формат CSS, т.к. с его помощью можно анимировать векторную графику, что является необходимостью создания данного проекта.

CSS3-анимация придаёт сайтам динамичность. С её помощью улучшается взаимодействие с пользователем. Создание CSS анимации базируется на ключевых кадрах, которые позволяют автоматически воспроизводить и повторять эффекты на протяжении заданного времени. CSS3-анимация может применяться практически для всех html-элементов.

Создание анимации начинается с установки ключевых кадров правила @keyframes. Правило @keyframes устанавливает ключевые кадры при анимации элемента. Кадры определяют, какие свойства на каком шаге будут анимированы. Каждый кадр может включать один или более блоков объявления из одного или более пар свойств и значений. Правило @keyframes содержит имя анимации элемента, которое связывает правило и блок объявления элемента. Ключевые кадры создаются с помощью ключевых слов from и to (эквивалентны значениям 0% и 100%) или с помощью процентных пунктов, которых можно задавать сколько угодно.  На рисунке 13 представлена функция scale3d, которая задает масштаб элемента. В зависимости от значения, элемент либо увеличивается, либо уменьшается. Если значение меньше единицы – уменьшается, если больше единицы – увеличивается.



Рисунок 13 – функция scale3d

На рисунках 14 и 15 представлен пример данного кода – постепенное увеличение и уменьшение таймера.

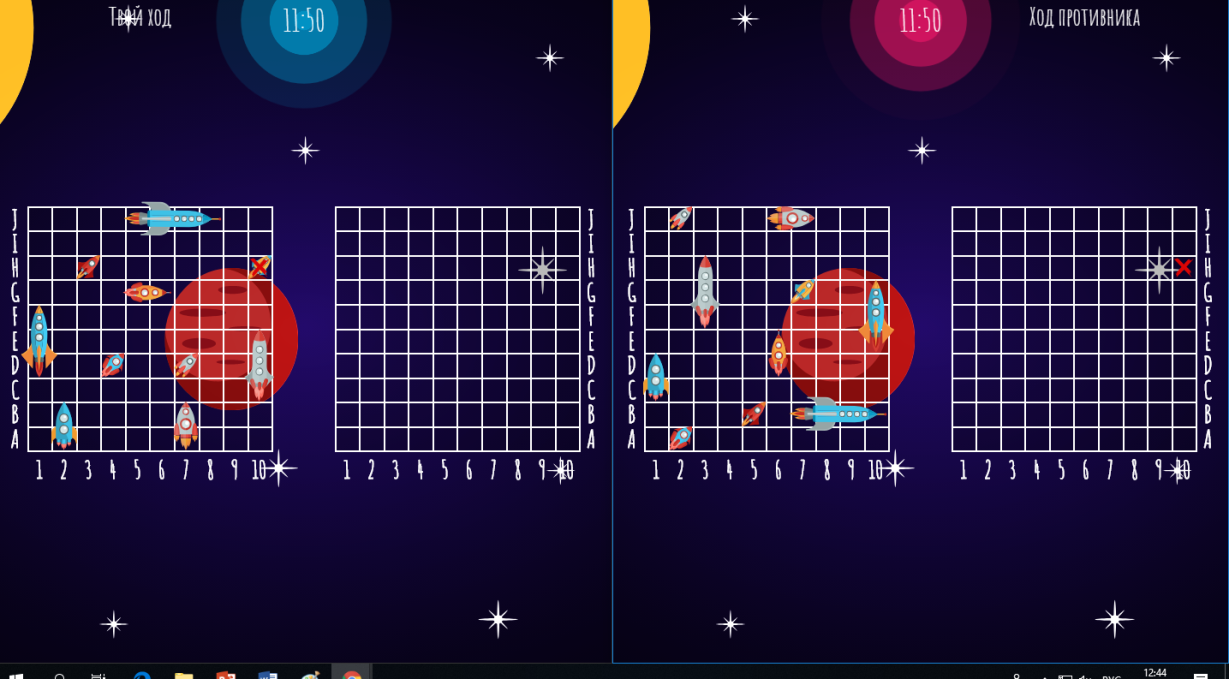


Рисунок 14 - постепенное уменьшение и увеличение изображения

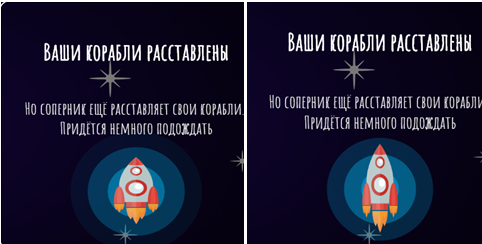


Рисунок 15 – постепенное уменьшение и увеличение изображения

**На рисунке 16 представлено правило @-webkit-keyframes**, позволяющее пользователю описать анимацию CSS свойств в виде перечня ключевых кадров. Оно поддерживается современными бразерами Safari и Chrome, в то время как правило **@keyframes** не поддерживается ни одним браузером.

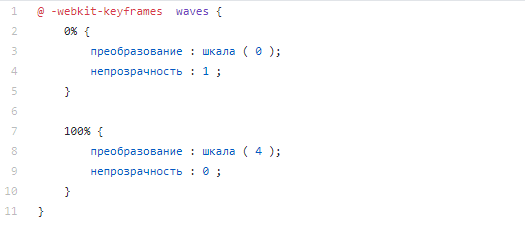


Рисунок 16 – **правило @-webkit-keyframes.**

С помощью данного кода можно увидеть увеличение и исчезновение фигуры.

С помощью данного кода можно увидеть увеличение и исчезновение фигуры. На рисунках 17 и 18 представлен пример данного кода – увеличение и исчезновение изображения.

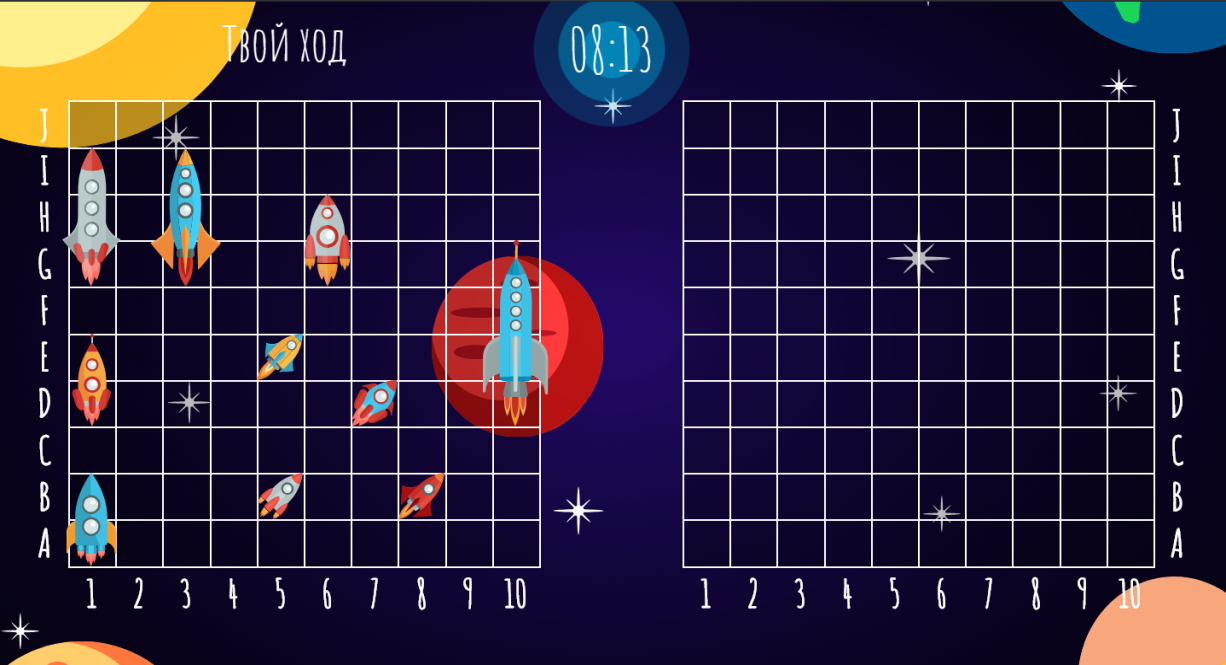


Рисунок 17- увеличение и исчезновение фигуры

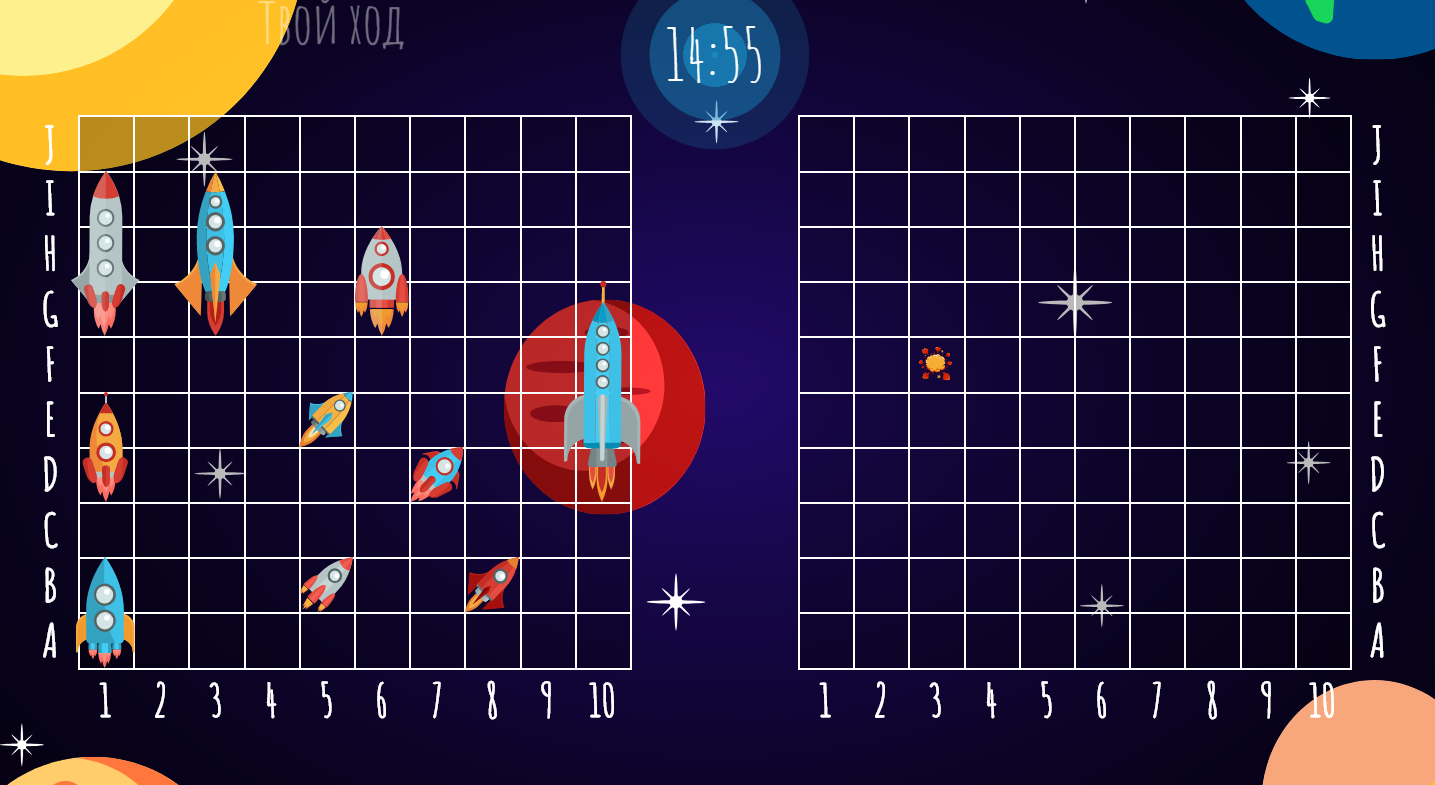


Рисунок 18 – увеличение и исчезновение фигуры

6 Разработка програмного кода игры «Морской бой»

6.1 Выбор средств разработки

По ходу разработки программного кода были изучены такие такие программные средства как PhpStorm, Docker, git, программирование на языке JavaScript. Получен опыт разработки приложения с поддержанием сессии через сокеты.

PhpStorm — коммерческая кросс-платформенная интегрированная среда разработки. Разрабатывается компанией JetBrains на основе платформы IntelliJ IDEA.

PhpStorm представляет собой интеллектуальный редактор для PHP, HTML и JavaScript с возможностями анализа кода на лету, предотвращения ошибок в коде и автоматизированными средствами рефакторинга для PHP и JavaScript. Автодополнение кода в PhpStorm поддерживает спецификацию PHP 5.3, 5.4, 5.5, 5.6, 7.0, 7.1 и 7.2 (современные и традиционные проекты), включая генераторы, сопрограммы, пространства имен, замыкания, типажи и синтаксис коротких массивов. Имеется полноценный SQL-редактор с возможностью редактирования полученных результатов запросов.

Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в среде виртуализации на уровне операционной системы. Позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть перенесён на любую Linux-систему с поддержкой cgroups в ядре, а также предоставляет среду по управлению контейнерами. Изначально использовал возможности LXC, с 2015 года применял собственную библиотеку, абстрагирующую виртуализационные возможности ядра Linux — libcontainer. С появлением ​Open Container Initiative начался переход от монолитной к модульной архитектуре.

Git — распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года. На сегодняшний день его поддерживает Джунио Хамано.

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.

6.2 Описание клиентской части программы

Клиентская часть программы была выполнена на языке программирования JavaScript, а отображение графического интерфейса программы выполнена на языке гипертекстовой разметки HTML и при помощи каскадных таблиц стилей CSS.

В приложении все JavaScript функции поделены на файлы для каждой из стадий игры.

Каждый файл содержит функции, отвечающие за какой-либо из этапов игры.

Список используемых функций:

* copyLink() – отвечает за копирование ссылки в буфер обмена;
* initStartScreen() – инициализирует стартовый экран;
* toPlacementShips() – отвечает за переход от стартового экрана к экрану расстановки кораблей;
* placementShipsScreen() – включается в себя множество подфункций, отвечает за расстановку кораблей;
* generateFields() – динамически генерирует поля 10х10 клеток;
* checkFields() – выполняет проверку ячеек на принадлежность к нужной строке;
* ifInRow() – выполняет проверку что ячейка входит в отведенную ей строку;
* waitingToStart() – определение игрока, который ходит первый, и инициализирует экран ожидания;
* checkRival() – переход программы к режиму готовности к бою;
* battleStart() – отвечает за ведение боя;
* showEnemyField() – показывает поле противника и скрывает экран ожидания;
* shot() – выстрел по полю противника;
* changeTurn() – смена хода по выстрелу или по окончанию таймера;
* shotMyField() – выстрел по своему полю;
* endBattle() – показывает окно с сообщением о победе или о поражении.

6.3 Описание функций серверной части программы

Серверная часть программы так же была написана на JavaScript. Для реализации этого использовалась программная платформа Node.js

Перечень Node.js компонентов для создания игры:

* App-root-path – для определения корневой директории приложения
* Express – для упрощения работы с приложением
* Qr-image – для генерации QR кода
* Socket.io – для поддержание сессии при помощи сокетов

Серверная часть программы расположена в единственном файле под названием server.js в корневой директории проекта.

При запуске программы сервер подключает все необходимые зависимости:

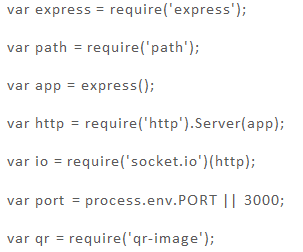


Рисунок 119 – пример кода**.**

Затем определяет для себя корневую директорию и определяет файл, который выдает пользователю при запросе к серверу

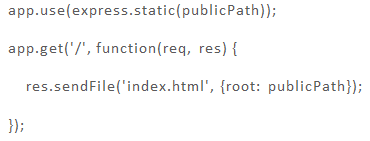


Рисунок 20 – пример кода**.**

Сервер запускает прослушивание 3000 порта по HTTP, и выводит сообщение об этом.

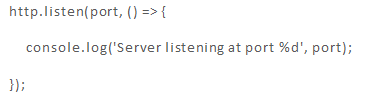


Рисунок 21 – пример кода**.**

При присоединении пользователя к серверу, сервер пытается определить, имеет ли пользователь ссылку с существующей комнатой для присоединения к пользователю, который его ожидает. Иначе сервер определяет пользователю номер комнаты, и дает пользователю ссылку на подключение к его комнате. Номер комнаты определяется текущим временем.



Рисунок 22 – пример кода**.**

6.4 Алгоритм запуска программы

Запуск программы осуществляется следующим образом: необходимо загрузить файлы программы на сервер с установленными на нем Node.js и Docker.

После этого необходимо подключиться к серверу по SSH и переключиться в корневую папку проекта, в которой содержится файл .Dockerfile.

Dockerfile — это простой текстовый файл, в котором содержится список команд Докер-клиента. Это простой способ автоматизировать процесс создания образа.

Теперь можно создать образ. Команда docker build занимается сложной задачей создания образа на основе Dockerfile.

Пример команды docker build: docker build -t spacebattle .

Последний шаг — запустить образ и проверить его работоспособность (замените username на свой): $ docker run -p 80:3000 spacebattle

Зайдите на URL сервера и увидите приложение в работе.

7 Тестирование программного обеспечения

7.1 Понятие тестирования

**Тестирование программного обеспечения** — проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. В более широком смысле, тестирование — это одна из техник контроля качества, включающая в себя активности по планированию работ (Test Management), проектированию тестов (Test Design), выполнению тестирования (Test Execution) и анализу полученных результатов (Test Analysis).

Тестирование – очень важный и трудоемкий этап процесса разработки программного обеспечения, так как правильное тестирование позволяет выявить подавляющее большинство ошибок, допущенных при составлении программ.

Различные наборы тест-кейсов и стратегий тестирования направлены на достижение одной общей цели – устранение багов и ошибок в коде, и обеспечения точной и оптимальной производительности программного обеспечения.

Методика тестирования

Широко используемыми методами тестирования являются

* модульное тестирование;
* интеграционное тестирование;
* приемочное тестирование;,
* тестирование системы.

В связи с особенностями проектирования и разработки тестирования программного продукта было задокументировано приёмочное тестирование, в которое входили

* Тестирование на кроссбраузерность;
* Тестирование пользовательского интерфейса и логики игры;
* Тестирование орфографии и грамматики;
* Тестирование звуковых эффектов;
* Тестирование работы по сети.

Тестирование на кроссбраузерность. Тест на кроссбраузерность – вид тестирования, который проводится для того, чтобы понять, одинаково ли ведет себя программный продукт при взаимодействии с разными версиями браузеров. Пользователь должен обращаться к сервисам программного продукта независимо от того, каким браузером пользуется и какую аппаратную платформу предпочитает. Если каждый десятый человек использует нестандартный браузер, в котором тестируемое приложение меняет внешний вид или отступает от заданного поведения, разработчик рискует потерять аудиторию в 10%.

При тестировании веб-приложений в разных браузерах, как правило, большая часть дефектов выявляется непосредственно в интерфейсе пользователя. Но не стоит забывать про функциональные возможности сайтов или приложений. При проведении кроссбраузерного тестирования могут быть выявлены функциональные проблемы, связанные с неработоспособностью:

* форм;
* фильтров;
* динамических элементов;
* отдельных модулей и разделов сайта.

Постоянное внедрение новых стандартов и следование им при разработке само по себе дает достаточно высокий уровень совместимости. Однако не все браузеры работают одинаково с этими стандартами. А разработчики в свою очередь зачастую пренебрегают ими, потому как практически невозможно, создавая продукт, учесть абсолютно все факторы и поддержать все разновидности клиентского ПО. Только тест продукта в разных браузерах поможет выявить все проблемы совместимости, оптимизировать производительность и добиться однообразия представления информации в поддерживаемых версиях. Плюс ко всему: не все из наиболее часто используемых браузеров оперативно выпускают новые версии, которые соответствуют новейшим стандартам, и, что более важно, не все пользователи следят и регулярно обновляют свое ПО. Если такие версии браузеров отвечают запросам целевой аудитории системы, то тест на кроссбраузерность просто необходим.

GUI и Usability тестирование.

Тестирование usability подразумевает проверку эргономичности системы, логики взаимодействия пользователей с системой, удобства использования функциональности продукта, а также выявление наиболее оптимальных решений по улучшению эффективности выполнения пользовательских сценариев. Проблемы взаимодействия пользователя и программного продукта могут поставить под вопрос успех вашего бизнес-проекта. Несоответствие мировым трендам по качеству графического предоставления информации приводит к снижению пользовательской активности и, соответственно, влияет на монетизацию интересных современных проектов. Графический интерфейс и архитектурно-информационная подача системы должны способствовать тому, чтобы пользователь быстро и эффективно, а главное интуитивно, осуществлял целевые операции.

Тестирование usability. Задачи и цели

Целью проведения в совокупности GUI и Usability тестирования является выявление необходимой доработки дизайна и/или функциональности существующего продукта, подтверждение или опровержение определенного интерфейсного решения. Позволяет выявить архитектурно-концептуальные проблемы вашего продукта, будь то юзабилити тестирование сайта, мобильного или веб-приложения или корпоративной системы.

Основными задачами, которые ставятся перед проведением итерации указанного типа тестирования, являются:

* анализ основных бизнес-процессов и пользовательских сценариев;
* локализация ошибок интерфейса и макетов дизайна;
* корректировка текстового наполнения страниц переходов и выбора условий;
* исправление навигационных элементов в пользу удобства и интуитвности интерфейса;
* выработка рекомендаций по улучшению представления информации и функциональности;
* внедрение изменений по сопровождению пользователя по шагам;
* применение типового общепринятого стиля подсказок и проч.

Испытания, основанные на различных сертифицированных методиках, определяют «слабые места» взаимодействия пользователей с функциями продукта, после чего формируются рекомендации по усовершенствованию последнего.

Оценка удобства пользования. Программа и методика usability тестирования разрабатывается, исходя из специфики каждого проекта: начиная от применения usability guidelines, современных эвристик и подходов, и заканчивая привлечением респондентов ваших реальных целевых групп и проведением маркетинговых экспериментов.

Критериями оценки становятся:

* Функциональность (удобство использования, процессы и бизнес-логика приложения).
* Дизайн (размещение информации, взаиморасположение элементов и простота их восприятия).
* Эффективность (результативность использования приложения пользователем, отсутствие избыточных операций).
* Производительность (скорость загрузки страниц интерфейса).
* Обучаемость пользователя (возможность работы с приложением на интуитивном уровне).
* Навигация (дизайн навигационных элементов и их налаженное функционирование).
* Целостность интерфейса (единая терминология, единый вид элементов интерфейса, стандартный способ работы общепринятых функций).
* Обратная связь с пользователем.
* Ставка на объективность и результат

7.2 Результаты тестирования

Тестирование пользовательского интерфейса – user interface (UI) и логики программы (таблица 1).

Таблица 1 – Тестирование пользовательского интерфейса и логики программы.

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Тестируемый функционал | Ожидаемый результат | Результат тестирования |
| 1 | Копирование ссылки на игру щелчком мыши | Ссылка скопирована | + |
| 2 | Возможность расставлять и поворачивать корабли | Корабль поставлен и повёрнут | + |
| 3 | Подсказка пользователю о доступности или недоступности текущего размещения корабля | Показывает доступное место расположения | + |
| 4 | Возможность сделать выстрел | Выстрел произведён | + |
| 5 | Отображение результата выстрела «попал»/ «мимо» | Отображается результат выстрела | + |
| 6 | Информирование игрока, что вражеский корабль полностью уничтожен | Игрок информирован об уничтожении вражеского корабля | –  Нет указания, уничтожен ли полностью корабль или только повреждён |
| 7 | Работа таймера отсчёта времени хода игрока | Таймер работает | + |
| 8 | Передача хода другому игроку по истечении лимита времени таймера | Смена хода по истечению времени | + |
| 9 | Наличие и работоспособность кнопки «Сыграть ещё» | Кнопка «Сыграть ещё» работает | + |
| 10 | Единообразие дизайна и используемых шрифтов | Дизайн и шрифты соответствуют единому стилевому решению | + |
| 12 | Работающий QR-код | QR-код работает | + |

Тестирование кроссбраузерности и мультиплатформенности (таблица 2).

Так как игра является браузерной, было проведено тестирование в разных браузерах и на экранах с разным разрешением. Подразумевается, что игра на устройствах с разным разрешением экрана должна выглядеть одинаково и быть удобной для восприятия.

Для тестирования выбраны самые популярные браузеры Google Chrome, Opera, Yandex, Mozilla Firefox при различных разрешениях экрана. Тестирование проводилось на операционных системах Windows 7, Windows 8.1, Windows 10.

Проверялись следующие параметры:

* Элементы интерфейса (кнопки, поля, текстовая область, фон, панели должны выглядеть одинаково вне зависимости от браузера, шрифты должны выглядеть единообразно).
* Размер окна (все элементы возможно использовать при уменьшении размера окна до минимально возможного).
* DPI (все элементы должны верно отображаться и на экранах с высоким DPI, и на экранах с низким).
* Масштабирование (меняются ли элементы при увеличении или уменьшении масштаба отображения страницы в браузере).

Таблица 2 – Проверка на кроссбраузерность отображения элементов при различном разрешении экрана.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Браузер | Разрешение экрана (DPI) | | | |
| 1024×768 | 1280×1024 | 1366×768 | 1920×1080 |
| Google Chrome 71.x | + | + | + | + |
| Opera 57.x | + | + | + | + |
| Yandex 18.x | + | + | + | + |
| Mozilla Firefox 64.x | + | + | + | + |
| Internet Explorer 11.x | ± | ± | ± | ± |

«+» – полное соответствие всех страниц приложения в браузере,

«±» – основная функциональность работает, возможны багги функциональности/отображения.

«–» – браузер не поддерживается.

При тестировании было обнаружено, что в браузере Internet Explorer игровое поле отображается со сдвигом (рис.)

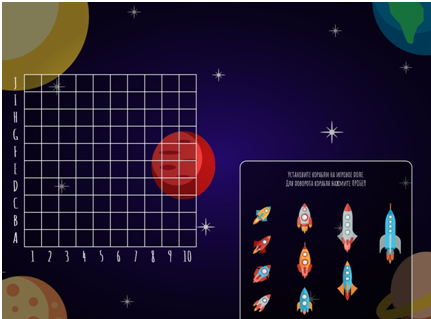


Рисунок 23 – Сдвиг отображении игрового поля при использовании браузера Internet Explorer.

При увеличении и уменьшении масштаба отображения страниц происходит изменение отображения элементов (рис. 24-25).

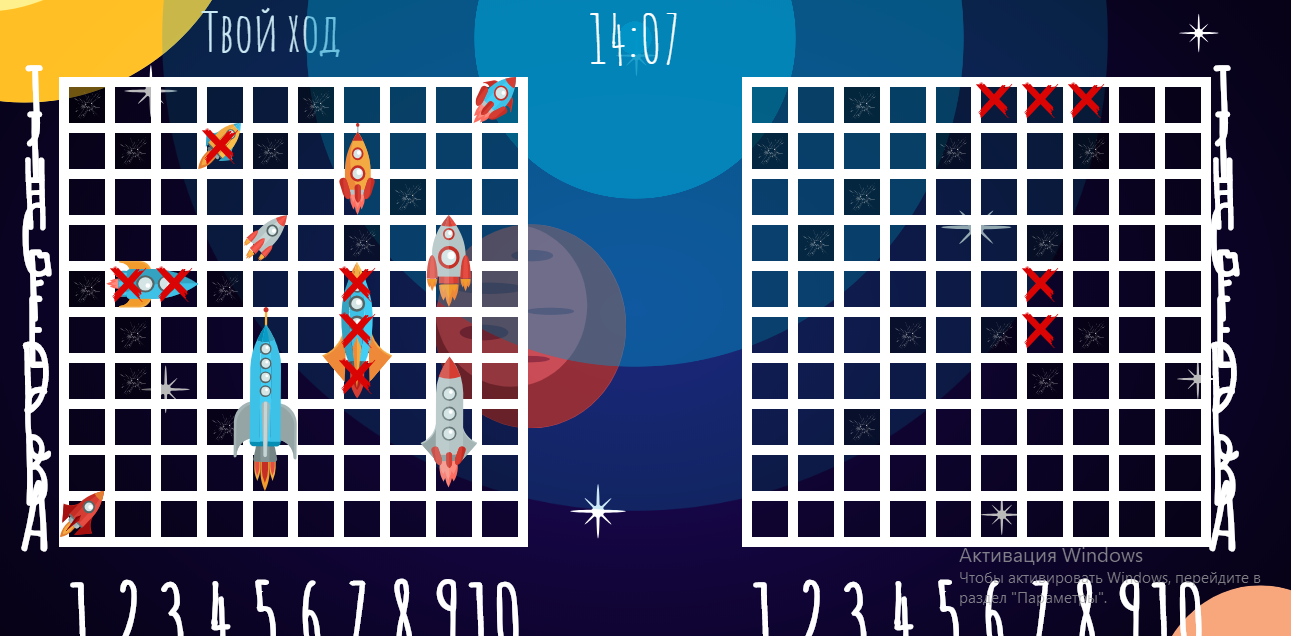


Рисунок 24 – Изменение отображения элементов при увеличении масштаба отображения страниц до 500%.

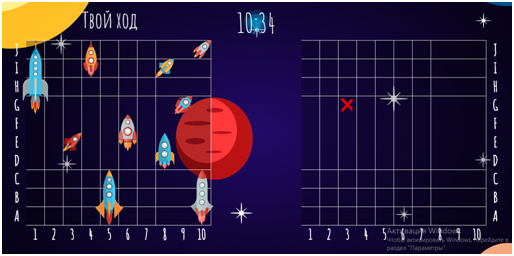


Рисунок 25 – Изменение отображения элементов при уменьшении масштаба отображения страниц до 50%.

При уменьшении размера окна происходит пропорциональное уменьшение игрового поля, но размер букв не уменьшается (рис. 26).

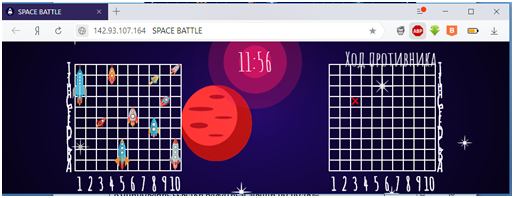


Рисунок 6 – Сбой при уменьшении размеров игрового окна.

Проверка корректности работы внешних девайсов.

Было проведено тестирование, в котором проверялась работа устройств ввода (манипулятор «мышь» и клавиатура) (таблица 3).

Таблица – Результаты тестирования работы устройств ввода

|  |  |
| --- | --- |
| Проверяемый функционал | Результат тестирования |
| Копирование ссылки нажатием мыши по полю | + |
| Поворот корабля помощью клавиатуры(пробел) | + |
| Расстановка корабля при помощи мыши | + |
| Воспроизведение выстрела при нажатии мышью по полю | + |
| Продолжение игры при нажатии мышью по кнопке «Сыграть ещё» | + |

Проверка орфографии и грамматики (таблица 4).

Таблица – Результаты тестирования орфографии и грамматики

|  |  |
| --- | --- |
| Проверяемые параметры | Результат тестирования |
| Орфография | + |
| Грамматика | + |

Тестирование звуковых эффектов (таблица 5).

Таблица – Результаты тестирования звуковых эффектов и фоновой музыки

|  |  |
| --- | --- |
| Звуковой эффект | Результат тестирования |
| Таймер перед запуском игры | + |
| Перемещение размещаемого корабля по игровому полю | + |
| Фиксирование корабля на игровом поле | + |
| Воспроизведение фоновой музыки | –  Фоновая музыка отсутствует |

Проверка корректности работы по сети

При кратковременном отсоединении и повторном подключении от интернета происходит сбой: наряду с таймером отсчёта времени хода появляется таймер начала игры.

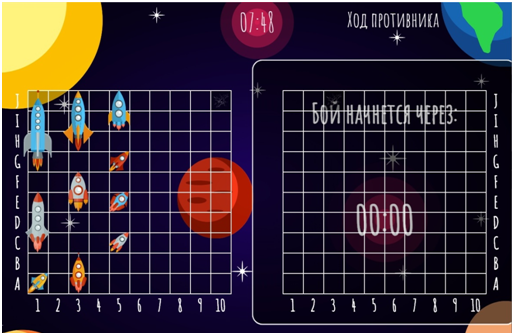


Рисунок 27 – Сбой при кратковременном отключении от интернета.

При изменении масштаба отображения одним игроком, меняется масштаб и у второго.

Таким образом, тестирование разработанного программного продукта показало его работоспособность. Было выявлено частичное несоответствие техническому заданию (отсутствие фоновой музыки). Обнаружены ряд багов, связанных с ошибками отображения при изменении масштаба отображения окон или изменения их размеров, а также обрыва интернет-соединения.

8 Документация

8.1 Руководство программиста

8.1.1 Назначение и условия применения программы

Программа является игрой для ПК, средством развлечения и предназначена для позитивного времяпровождения. Суть игры Морской бой состоит в том, что два человека по очереди вслепую выбирают координаты на карте противника (поле противника). Названная точка должна разгромить ракету или зацепить ее часть. Чем раньше игрок разрушит флот соперника, тем быстрее он одержит победу.

Ссылка на игру Морской бой находится в общем доступе на Игру в облачном хранилище. Необходимо открыть ссылку, скопировать ее в буфер обмена и отправить другу с которым хотим сыграть.

Для работы приложения необходимо подключение к интернету и любой браузер, более никаких дополнительных установок не требуется.

Минимальная конфигурация ПК (таблица 6):

Таблица 6 – Минимальная конфигурация ПК.

|  |  |  |
| --- | --- | --- |
| Операционная система | Рекомендуемые характеристики | Минимальные характеристики |
| Mac OS | Процессор Core i5 2-го поколения (от 2 ГГц), Core i5 3-го или 4-го поколения или аналогичные  Windows 7 и более поздние версии | Процессор Core i3 или аналогичные  Windows Vista и более поздние версии |
| Mac OS | Macbook Pro 2011, Macbook Air 2012 или более новые версии  Mac OS X 10.9 и более поздние версии | Macbook Pro 2010, Macbook Air 2011  Mac OS X 10.9 и более поздние версии |
| Chrome OS | Все модели 2014 года и более поздние; модели 2013 г. с процессором на ядре Haswell; Samsung Chromebook 2 | Неприменимо |
| Linux | Работа зависит от дистрибутивов, драйверов и среды рабочего стола |  |

8.1.2 Характеристика программы

Средняя длительность игры: 10 минут.

Быстродействие программы не зависит от качества ПК (при соответствии системным требованиям), то есть при повышении мощности компьютера быстродействие не увеличивается – игра стабильно работает на любых сборках системы.

Программа способна работать только с подключением к сети Интернет.

8.1.3 Обращение к программе

http://142.93.107.164/

8.1.4 Входные и выходные данные

Программа может принимать данные от пользователя только для настройки громкости звука и переключения оконного\полноэкранного режима работы. Остальные данные задаются фиксированно.

Программа предоставляет выходные данные пользователям посредствам визуального и звукового интерфейса.

8.2 Инструкция пользователя

Для запуска игры необходима ссылка в общем доступе на Игру, которая хранится в облачном хранилище.

Открываем ссылку, копируем в буфер обмена и отправляет другу с которым хотим сыграть.

В первую очередь необходимо скопировать ссылку в буфер обмена. Нужно кликнуть по ссылке в центре экрана, и она автоматически будет скопирована.

После прохождения по ссылке друга появляется второй экран, на котором необходимо расставить корабли.



Рисунок 28 – экран расстановки кораблей**.**

Слева находится поле в которое надо вставить корабли. Справа представлены сами корабли и обозначено их количество. Нужный корабль выбирается кликом по нему. Дальше нужно установить корабль на поле слева. При этом при наведении на поле будет отображаться контур корабля, привязанный к курсору и отображающий возможность установки в выбранном месте. В любой момент можно поменять корабль на любой другой кликнув на нём. При этом курсор изменится согласно выбранному кораблю. Если установка прошла корректно, счётчик у соответствующего типа корабля уменьшится на 1. Когда все корабли будут установлены, можно будет нажать на кнопку «Начать игру», при нажатии на которую произойдёт переход или на начало игры с изменением текущего экрана, или же на режим ожидания противника, который еще расставляет корабли.

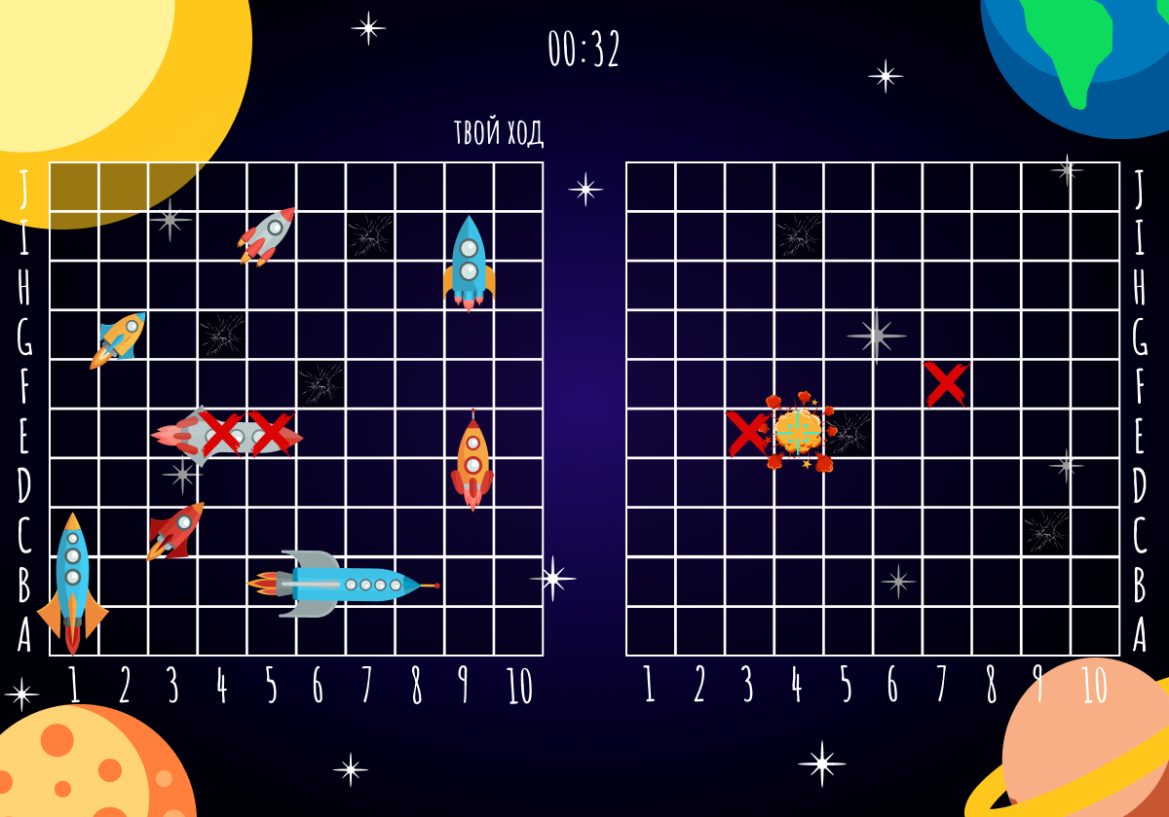


Рисунок 29 – **экран боя.**

Для первого хода игрок должен кликнуть по любой клетке правого поля. Ходы идут поочередно, вне зависимости от промаха или же попадания в корабль противника.

Игра ведётся до полного уничтожения кораблей одной из команд. По завершению игры выведется соответствующее сообщение и звуковое сопровождение.

Заключение

Целью данной работы была создание и разработка игры «Морской бой» с музыкальной подборкой и индивидуальным интерфейсом, для приятного времяпровождения.

В ходе создания курсового проекта было написано техническое задание и согласовано с преподавателем, реализован продукт, который разработан на мультипарадигменном языке программирования javaScript на платформе node.js. Используются также: язык гипертекстовой разметки страницы - html и каскадные таблицы стилей - css. Разработан и создан уникальный интерфейс в веб-приложении – Figma. Найдены и отредактированы в Adobe Audition звуковые эффекты и фоновое сопровождение для игры. Разработаны уникальные анимации в CSS. А также составлена вся необходимая документация для дальнейшего сопровождение ПО (документация разработчика, руководство по установке и руководство пользователя).

В результате проделанной работы было разработано приложение, реализующее неклассическую версию игры «морской бой» в космической тематике. Созданный продукт полностью выполняет все запланированные функциональные требования, такие как:

* возможность игры по сети;
* индивидуальный интерфейс в определенной тематике;
* звуковое сопровождение;
* визуализация интерфейса.

Поставленные цели достигнуты в срок, нареканий по работе программе на этапе проектного внедрения нет. Игру можно посмотреть по ссылке http://142.93.107.164/

Список использованных источников

1. Figma Vs. Sketch [Электронный ресурс]. — Режим доступа: https://goo.gl/W7qY3R (Дата обращения: 24.10.18);
2. You Are Going to Switch to Figma [Электронный ресурс]. — Режим доступа: https://goo.gl/QMnQrj (Дата обращения: 25.10.18);
3. Sketch или Figma [Электронный ресурс]. — Режим доступа: https://goo.gl/W7qY3R (Дата обращения: 24.10.18);
4. С. Чакон, Б. Штрауб. Git для профессионального программиста. — Санкт-Петербург, 2017. — 496 с. — ISBN 978-5-496-01763-3;
5. Итан Браун. Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript = Web Development with Node and Express / Итан Браун. — Санкт-Петербург, 2017. — 336 с. — ISBN 978-1-491-94930-6;
6. Описание продукта JetBrains PhpStorm [Электронный ресурс]. — Режим доступа: https://goo.gl/cPvi9n (Дата обращения: 15.12.2018);
7. Звук в играх: История и развитие [Электронный ресурс]. — Режим доступа: https://goo.gl/tddcku (Дата обращения: 04.12.18);
8. Playing with your mind: The psychology of sound in video games [Электронный ресурс]. — Режим доступа: https://goo.gl/DM9hA1 (Дата обращения: 06.12.18);
9. Тестирование. Фундаментальная теория [Электронный ресурс]. — Режим доступа: https://habr.com/post/279535/ (Дата обращения: 15.12.18);
10. Программы для 2D-анимации и мультипликации [Электронный ресурс]. — Режим доступа: https://goo.gl/tEM94g (Дата обращения: 28.11.18);
11. “Морской Бой” – Семь вариантов популярной игры мультипликации [Электронный ресурс]. — Режим доступа: https://goo.gl/F9AmnW (Дата обращения: 01.12.18).