МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «ВИТЕБСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ П.М. МАШЕРОВА»

Факультет математики и информационных технологий

Кафедра информационных технологий и управления бизнесом

КУРСОВОЙ ПРОЕКТ

по дисциплине «Компьютерные системы и сети»

РАЗРАБОТКА EB-ИГРЫ «KAPPALAND»

НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ JAVASCRIPT

Борисенко Анастасия Владимировна,

курс 2, группа 24

Руководитель:

Шпаков Сергей Андреевич,

старший преподаватель кафедры информационных технологий и управления бизнесом

Витебск, 2021

Содержание

[Введение 3](#_Toc74313106)

[1. Разработка web-игры 5](#_Toc74313107)

[1.1. Варианты разработки web-игры 5](#_Toc74313108)

[1.2. JavaScript 6](#_Toc74313109)

[1.3. Компьютерные сети 7](#_Toc74313110)

[1.4. MongoDb 7](#_Toc74313111)

[2. Реализация приложения 9](#_Toc74313112)

[3. Интерфейс приложения 16](#_Toc74313113)

[Заключение 18](#_Toc74313114)

[Список использованных источников 19](#_Toc74313115)

# Введение

Браузерные игры представляют собой категорию онлайн игр, в которых Web-браузер выступает либо в роли операционной оболочки для игр, позволяя играть в игру без установки на компьютере дополнительного ПО, либо служит контейнером для дополнительной виртуальной машины, которая непосредственно выполняет код игры (JavaScript, Flash, Shockwave и аналогичные). Игры данного типа чаще всего являются казуальными играми, что связано с ограничениями на размер.[1]

Целью данной работы является создание MMORPG-игры «KappaLand» на языке программирования JavaScript (вспомогательными языками стали HTML, CSS) при помощи кроссплатформенного фреймворка Express.js.

KappaLand-представляет самой небольшую наработку, своеобразную мини-игру в пинбола.

Для достижения поставленной цели необходимо решить следующие задачи:

* выработать умение логически строить и последовательно излагать материал по выбранной теме;
* освоить и овладеть особенностями создания MMORPG-игра при помощи [фреймворк](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA)а Express.js в Web-приложениях.
* разработать модуль который работает с браузером, а также модуль клавиш для работы с клавиатурой.
* разработать графический интерфейс пользователя для удобства взаимодействия между пользователем и приложением. Ориентируясь на среднего пользователя, интерфейс должен быть простым и удобным;
* изучить и проработать литературу, раскрывающую основы языка программирования JavaScript а также – средства, главные принципы, методы программирования на языке JavaScript, описания стандартов кодирования на данному языку программирования.
* получить навыки разработки MMORPG-игры в веб-браузере с использованием [фреймворк](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA)а Express.js
* анализ основных базовых операций проверки регистрационных HTML форм и разработка нескольких базовых JavaScript функций для этих целей.

# Разработка web-игры

## 1.1. Варианты разработки web-игры

Создание любой игры — это продолжительный и трудоёмкий процесс, состоящий из самых разнообразных этапов, включающий в себя как технические, так и творческие моменты.

Перед созданием игры программисту нужно решить, по какому пути её разработки он хочет пойти.

Первый вариант – это однопользовательские браузерные игры. Наиболее распространёнными в классе однопользовательских игр являются игры на Adobe Flash. В последнее время сравнимые возможности становятся доступны без плагинов благодаря технологиям из группы HTML5. Ранее их место занимали Java-игры. Однопользовательские браузерные игры, как правило, просты и доступны для быстрого освоения. В большинстве, такие игры можно отнести к разряду казуальных.

Второй вариант – это многопользовательские браузерные игры. К этому классу можно отнести игры, имитирующие настольные игры, такие как шахматы, нарды, и карточные игры, монополия. В такие игры можно играть вдвоём (друг против друга) или с несколькими партнёрами. Браузер обеспечивает связь игроков между собой посредством игрового сервера. Игра может требовать одновременного присутствия всех участвующих в игровой партии игроков, или нет. В последнем случае ходы делаются каждым игроком по очереди в удобное им время и сохраняются на игровом сервере. В массово-многопользовательских браузерных играх сотни, тысячи, или десятки тысяч игроков взаимодействуют друг с другом. В большинстве игр этого класса интерфейс выполнен в виде обычных HTML-страниц с текстом и изображениями. Минимальная интерактивность (обычно различные таймеры и чат) обеспечивается с помощью JavaScript. Ряд игр использует технологию Flash, что до появления HTML5 позволяло сделать игру более привлекательной внешне, однако вносит в игровой процесс присущие флэш-приложениям ограничения и недостатки, такие как перекрытие стандартных функций браузера (переход вперёд и назад, обновление страницы), задержки, связанные с загрузкой флэш-роликов, требующих, как правило, больший (чем HTML и изображения) объём передаваемых в браузер данных.

При разработке приложения «KappaLand» мной был выбран второй вариант, поскольку он ближе к данной дисциплине и интереснее.

## 1.2. JavaScript

Сейчас браузеры дают JavaScript-разработчикам огромное количество возможностей для создания интересных сайтов. Раньше для этого использовался Flash — он был популярен, и на нём было создано бессчётное количество игр, плееров, необычных интерфейсов и так далее. Однако они уже не запустятся ни в одном современном браузере.[2]

Чем все же хорош JavaScript:

* на лету вставить в HTML-код любые [теги](https://netology.ru/glossariy/teg);
* задать внешний вид элементов через класс и атрибуты HTML;
* переместить любой элемент;
* запросить у пользователя данные;
* отправить запрос на сервер (технология AJAX).

JavaScript — подходящий язык для выполнения этого проекта. Он достаточно прост, но содержит все фундаментальные вещи: алгоритмы, объектно-ориентированную модель, структуры данных. Если традиционные языки для обучения — Pascal и Basic — несут мало практической пользы, то JavaScript — рабочая лошадка.

Начинать с JavaScript хорошо и потому, что синтаксически он похож на великий и ужасный язык С. Изучив JavaScript, получишь базовое представление обо всех «сиобразных» языках: С++, C#, Java, PHP. Они задают тренд в своих областях и весьма популярны, поэтому для новичка важно познакомиться с синтаксисом С.

Для примера, Google Maps и веб-клиент Gmail используют JavaScript. А если вы хотите написать игру — возьмите JS, HTML5.

## 1.3. Компьютерные сети

Создание компьютерный сетей вызвано потребностью совместного использования информации на удаленных друг от друга компьютерах.

Oсновное назначение компьютерных сетей — совместное использование ресурсов и осуществление связи как внутри одной организации, так и за ее пределами.  
Все сети делятся на три типа:

* одноранговые сети;
* сети на основе сервера;
* комбинированные сети.

В данного проекта будем использовать серверный способ.

**Компьютерная сеть —**это объединение нескольких компьютеров для совместного решения информационных, вычислительных, учебных и других задач.

Компьютерные сети и сетевые технологии обработки информации стали основой современных информационных систем. Компьютер следует рассматривать не как отдельное вычислительное устройство, а как «окно» в компьютерные сети, средство связи с сетевыми ресурсами и другими пользователями сети.[3]

В последние годы глобальный Интернет стал мировым явлением. Сеть, которая до недавнего времени использовалась лишь ограниченной группой ученых, государственных служащих и сотрудников образовательных учреждений в процессе их профессиональной деятельности, теперь доступна для крупных и малых предприятий и даже индивидуальных пользователей.[3]

## 1.4. MongoDb

MongoDb — это документо-ориентированная база данных. Т.е. каждая запись – это документ без жестко заданной схемы, который может содержать вложенные документы.

MongoDb хороша высокой скоростью записи/чтения, масштабированностью, но опять же сохранность и целостность данных не так хороша. Так, например, в Mongo есть отличная реализация репликации, которая довольно легко устанавливается и настраивается или же шардинг (возможность разнести данные по нескольким серверам), который тоже довольно прост в установке. В комплексе, можно получить систему распределенных вычислений с высокой отказоустойчивостью[4].

MongoDb использует JSON нотацию для хранения и управления документами, а также свой достаточно удобный язык запросов. Основные возможности данной СУБД:

* Документо-ориентированное хранилище (простая и мощная JSON-подобная схема данных);
* Достаточно гибкий язык для формирования запросов;
* Динамические запросы;
* Полная поддержка индексов;
* Профилирование запросов 5;
* Быстрые обновления «на месте» ;
* Эффективное хранение двоичных данных больших объёмов, напр., фото и видео;
* Журналирование операций, модифицирующих данные в БД;
* Поддержка отказоустойчивости и масштабируемости: асинхронная репликация, набор реплик и шардинг;
* Может работать в соответствии с парадигмой MapReduce.

# Реализация приложения

Для создания приложения необходимо написания компилируемого кода с использования инструментов:

* NodeJS
* ExpressJS
* MongoDB
* socketIO
* Heroku

Разбираем частично код, написанный мной:

Начнем с основных моментов это сам игрок и чем он будет стрелять (Листинг 2.1.) и (Листинг 2.2.)

Листинг 2.1. – Игрок

id: Math.random(),

x: posX + 25,

y: posY + 25,

playerId: playerId,

direction: direction,

speed: 10,

timer: 0,

toRemove: false,

Листинг 2.2. – Пуля

x: X\_STARTING\_POS,

y: Y\_STARTING\_POS,

id: id,

username: name,

points: points,

char: 'tyler1',

rightPress: false,

leftPress: false,

upPress: false,

downPress: false,

lastPosition: STARTING\_DIR,

speed: PLAYER\_SPEED

Для смены позиции игрока (Листинг 2.3.) и пули (Листинг 2.4) [5]

Листинг 2.3. –Позиция игрока

player.updatePosition = function () {

if (player.rightPress)

player.x += player.speed;

if (player.leftPress)

player.x -= player.speed;

if (player.upPress)

player.y -= player.speed;

if (player.downPress)

player.y += player.speed;

Листинг 2.4. –Позиция пили

bullet.updatePosition = function(){

if (bullet.direction === 'right')

bullet.x += bullet.speed;

else if (bullet.direction === 'left')

bullet.x -= bullet.speed;

else if (bullet.direction === 'up')

bullet.y -= bullet.speed;

else if (bullet.direction === 'down')

bullet.y += bullet.speed;

Момент стрельбы игроком воспользуемся (Листинг 2.5.)

Листинг 2.5. –Стрельба

player.shootBullet = function (){

var bullet = Bullet(player.id,player.x,player.y,player.lastPosition);

bulletList[bullet.id] = bullet;

Листинг 2.6. – Начальная позиция

const X\_STARTING\_POS = 100;

const Y\_STARTING\_POS = 100;

const PLAYER\_SPEED = 10;

const STARTING\_DIR = 'down';

const MONGO\_REPO = "Account";

Мной будет использована MongoDB, как база данных соответственно, а также использовался socket.IO (Рисунок 2.2) для взаимодействия в реальном времени между сервером и клиентом[5].

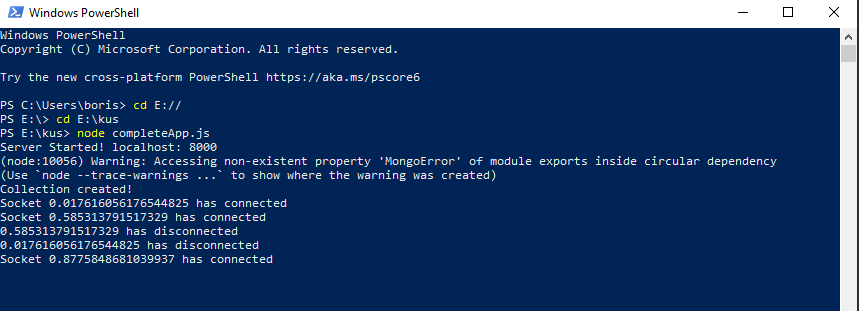


Рисунок 2.1. – Терминал-подключения

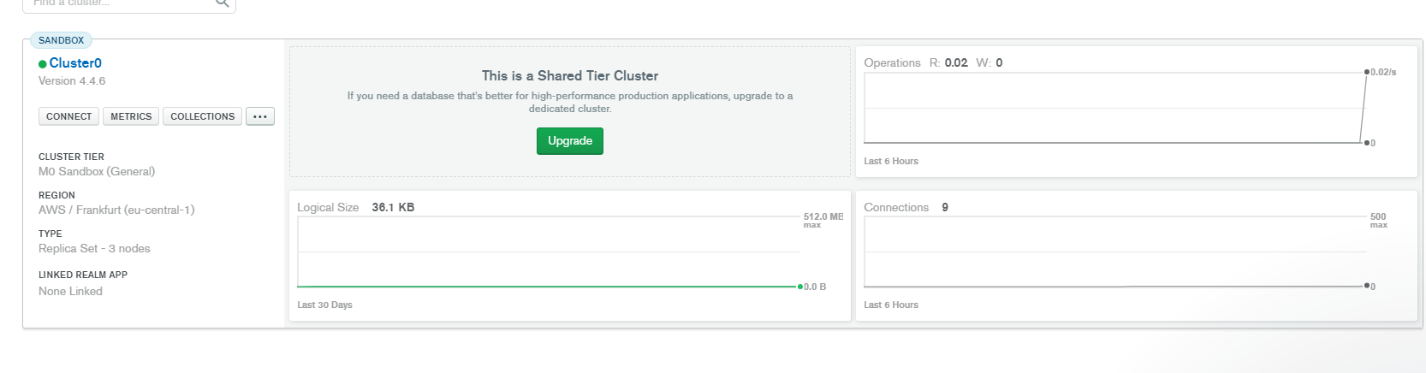


Рисунок 2.2. – MongoDB-проект

Приложение развертывается в Heroku с использованием базы данных mongoDB, размещенной в MongoDB Atlas, облачной службе баз данных(Рисунок 2.3).



Рисунок 2.3. – База

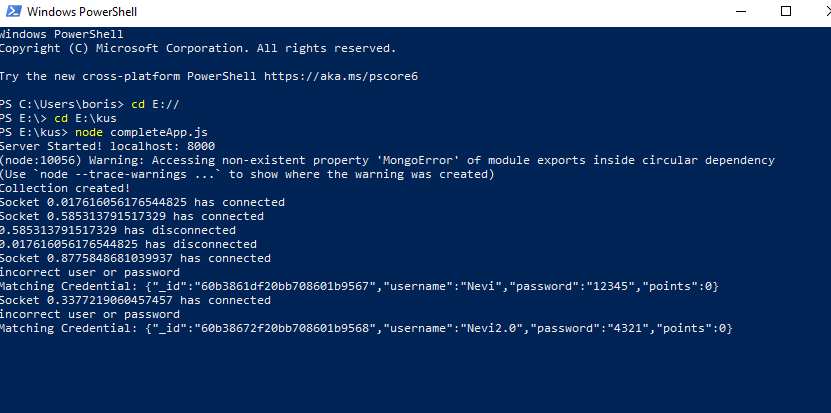


Рисунок 2.4. – Терминал-отладка

Игра требует, чтобы пользователь зарегистрировался с учетными данными(Листинг 2.8), которые будут храниться в базе данных. Их можно просмотреть не только в самой базе, но и через PowerShell (Рисунок 2.1 и, Рисунок 2.4) так как не происходит шифрования данных, мы можем наблюдать в Листинг 2.7.

Листинг 2.7. – Подключения к базе и фиксация админ позиции

var url = "mongodb+srv://admin:A12345@cluster0.e6gnp.mongodb.net/myFirstDatabase?retryWrites=true&w=majority";

Листинг 2.8. – Реализация ввода логина и пароля

function insertCredential(data) {

var account = {

username: data.username,

password: data.password,

points: 0

Интервал между игроком и пулей (Листинг 2.9) рассчитывается.

Листинг 2.9. – Расчет интервала

if (bullet.x > player.x && bullet.x < player.x + 50 && bullet.y > player.y && bullet.y < player.y + 60)

var player = Player(socket.id, name, points);

playerList[socket.id] = player;

Передвижения игроков происходит с помощью сокета, фишкой данной работы является глючноватые движение персонажа, которые происходят за счет клавиш (Листинг 2.10)

Листинг 2.10. –Сокет

socket.on('keyPress', function (data) {

if (data.inputId === 'right')

player.rightPress = data.state;

else if (data.inputId === 'left')

player.leftPress = data.state;

else if (data.inputId === 'up')

player.upPress = data.state;

else if (data.inputId === 'down')

player.downPress = data.state;

if (data.inputId === 'shoot' && playerList[socket.id] != null)

player.shootBullet();

else

player.lastPosition = data.inputId;

Так же в данном проекте реализован чат между игроками (Листинкг 2.11)

Листинг 2.11. –Данные для чата

var chatText = document.getElementById('chat-text');

var chatForm = document.getElementById('chat-form');

var chatInput = document.getElementById('chat-input');

Отрисовка поля для наших игроков (поле является статическим) (Листинг 2.12).

Листинг 2.12. – Поля

case 'down':

canvas.drawImage(playersImg, 0, 0, imgWidth, imgHeight, player.x, player.y, imgWidth, imgHeight);

break;

case 'up':

canvas.drawImage(playersImg, imgFrameIndex, 0, imgWidth, imgHeight, player.x, player.y, imgWidth, imgHeight);

break;

case 'left':

canvas.drawImage(playersImg, imgFrameIndex \* 2, 0, imgWidth, imgHeight, player.x, player.y, imgWidth, imgHeight);

break;

case 'right':

canvas.drawImage(playersImg, imgFrameIndex \* 3, 0, imgWidth, imgHeight, player.x, player.y, imgWidth, imgHeight);

break;

# Интерфейс приложения

В разделе представлены результаты выполнения основных функциональных частей web-приложения.

При запуске web-приложения нам предлагаю зарегистрироваться или зайти на уже существующею учётную запись (Рисунок 3.1.):

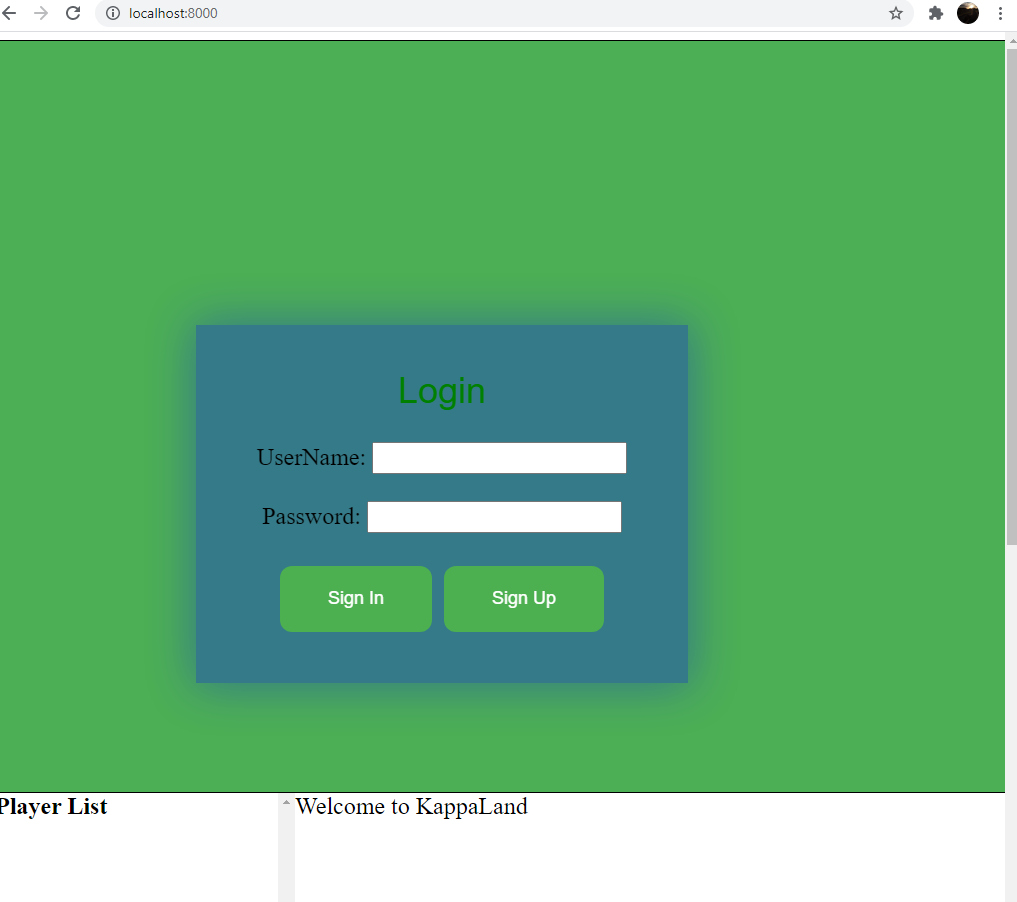


Рисунок 3.1. –Поля регистрации и входа

В этой игре реализована система очков. Зарабатывать очки можно бросая шары в других игроков и пытаясь получить максимальное количество. Так же есть реализация чата. Еще отличительным моментом данной игры является не финал, а организация спора между игроками на определенное количество очков, так как игра бесконечна (Рисунок 3.2):



Рисунок 3.2. – Игровое окно

# Заключение

В данном курсовом проекте был рассмотрен один из множества способов создания web-игры.

Разработка компьютерной игры «KappLand», созданной на языке программирования JavaScript, превратилась в увлекательное занятие, не требующее глубоких знаний технологий обработки изображения и базовых познаний в базах данных.

В мире существует множество платформ и утилит, упрощающих разработку компьютерных игр. Но, к сожалению, многие из них могут по-разному работать с разными операционными системами. Например, игра, написанная с использованием технологии Microsoft XNA, не будет работать на операционной системе класса Unix[5].

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Синтаксис языка JavaScript достаточно прост и интуитивно понятен, что понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При этом JavaScript является очень востребованным языком.

Научившись программировать на языке JavaScript, начинающий разработчик получит мощный толчок в индустрии web-разработки в разработки браузерных расширений, серверных приложений.

Проанализировав предметную область, можно сказать, что разработка web-игры «KappLand» на языке JavaScript актуальна. Цель работы достигнута, так как разработанное приложение соответствует поставленной цели.

# Список использованных источников

1*.*Компьютерная сеть [Электронный ресурс] <https://lfirmal.com/kursovaya-rabota-na-temu-kompyuternye-seti/> – Дата доступа:04.05.2021

2.Как написать игру на JavaScript [Электронный ресурс] <https://skillbox.ru/media/code/kak_napisat_igru_na_javascript/> – Дата доступа: 04.05.2021

3. Разработка ЛВС компьютерного клуба [Электронный ресурс] <https://www.yaneuch.ru/cat_22/razrabotka-lvs-kompjuternogo-kluba/> 8568.527081.page1.html – Дата доступа:04.05.2021

4.JavaScript: Подробное руководство (Definitive Guide), Давид Финнерман. Спб, 2007г. – Дата доступа:04.05.2021

5.Система управления сайтом — CMS Joomla [Электронный ресурс] //Система управления сайтом — CMS Joomla [Сайт]. URL: <http://x340.ru/soft/16-soft1.html> – Дата доступа:04.05.2021