****

СОДЕРЖАНИЕ

[Введение 3](#_Toc511828771)

[1 Анализ предметной области 5](#_Toc511828772)

 [1.1Язык разметки страниц HTML 5](#_Toc511828773)

 [1.2 Каскадные таблицы стилей CSS 6](#_Toc511828774)

 [1.3 Концепция сайта 6](#_Toc511828775)

[2 Создание структуры сайта 7](#_Toc511828776)

 [2.1 Графическая составляющая сайта 9](#_Toc511828777)

 [2.2 Динамические объекты сайта 14](#_Toc511828778)

 [2.3 Программная реализация 17](#_Toc511828779)

[Заключение 19](#_Toc511828780)

[Список использованных источников 20](#_Toc511828781)

ВВЕДЕНИЕ

Интернет стал неотъемлемой частью нашей жизни. Способы его использования необычайно разнообразны: коммуникация, поиск информации, развлечения и многое другое. В нашем случае нас интересует возможность реализации в интернете web-приложения помогающего с выбором в компьютерной игре. Актуальность этой работы состоит в том что в наше время все стремительнее развиваются киберспортивные дисциплины с крупномасштабными турнирами, которые смотрят сотни тысяч людей по всему земному шару. Россия стала первой страной в мире, которая признала киберспорт официальным видом спорта. Это произошло 25 июля 2001 года по распоряжению предыдущего главы Госкомспорта России Рожкова Павла Алексеевича. В настоящее время в России как и во многих странах мира проводятся соревнования по компьютерным играм, например ВКСЛ (Всероссийская киберспортивная студенческая лига), в котором принимают участие сборные из многих российских вузов. Кроме в нашей стране имеют место и более крупные, международные турниры такие как EPICENTER в них соревнуются лучшие мировые команды, а победа в них гарантирует крупное денежно вознаграждение (1000000$ средний призовой фонд турниров класса major) и прямое приглашение на главный турнир года The International. И хоть на первый взгляд может показаться что в компьютерных играх для того чтобы показывать высокие результаты не нужно прикладывать много усилий, на самом деле это не так. Большинство игроков высокого уровня тренируются по 10 и более часов в сутки, на данном этапе игровая дисциплина перестает быть просто хобби, а во многих случаях уже и не приносит удовольствия [1].

 Еще один важный отличительный фактор некоторых онлайн-игр – это многообразие факторов влияющих на исход матча. Например в игре Dota 2, которую я рассматриваю игроки всегда играют на разных персонажах, которые обладают различными характеристиками, и конечно больший шанс на победу имеет та команда способности чьих персонажей нивелируют способности персонажей противника. И если в обычных играх каждый игрок выбирает себе персонажа, руководствуясь больше своими личными предпочтениями, то на турнирах перед игрой проводится “Игра капитанов”. В ней лидеры команд выбирают и блокирую персонажей, данный этап очень важен для развития игры и в большинстве случаев определяет исход матча, т.к уже на стадии выбора можно увидеть каким стелим будет играть команда. Но даже профессиональные игроки не всегда понимают преимущество выбора в некоторых ситуациях, не говоря уже о простых игроках, многие из которых тратят годы, для того чтобы научиться анализировать и предсказывать исход игры при выборе различных персонажей. Именно поэтому целью данного курсового проекта я поставил: разработку интернет-сайта, который подскажет наилучший выбор в различных игровых ситуациях.

1 Анализ предметной области

 Первый этап проектирования web-сайта подразумевает анализ предметной области. Нам необходимо определить комплекс задач, для реализации которых создаются данный сайт.

 Необходимо изучить структуру подобных проектов и целевую аудиторию. Все это поможет создать сайт удобный для потребителя, а удобство клиентов, один из самых важных факторов на рынке. На такой web-сайт пользователь вероятно вернется еще раз, а может и расскажет знакомым, что приведет в вашу организацию новых клиентов. Поэтому перед тем как приступать к проектированию, необходимо тщательно ознакомиться с информационной базой [2].

1.1Язык разметки страниц HTML

 Самым распространенным языком разметки является язык HTML. HTML-документ состоит из двух частей: текста - данных, составляющих содержимое документа, и тегов, необходимых для разметки страницы и управляющих отображением информации . Благодаря тегам, можно назначить, в каком виде будет представлено содержимое документа, какие компоненты будут исполнять роль ссылок, какие станут графическими или мультимедийными объектами .

 Графика и звуковые файлы, включенные в HTML-документ, хранятся в отдельных файлах. Браузеры переводят флаги разметки и располагают текст и графику на экране соответствующим образом [3].

 В большинстве случаев теги используются по два. Пара тегов состоит из открывающего <тег> и закрывающего </тег> тегов. Действие любого тега начинается с открывающего тега, и заканчивается при встрече соответствующего закрывающего тега. Часто пару тегов, называют контейнером, а часть текста внутри них — элементом.

1.2 Каскадные таблицы стилей CSS

 CSS используется для приданию сайту внешней привлекательности, он отвечает за шрифты, цвет, высоту и ширину строк, задний фон и многое другое. Сайт может существовать и без использования таблиц стилей, но тогда он будет выглядеть блекло и не интересно. На таком сайте пользователь вряд ли задержится, а это пагубно повлияет на продвижение. Поэтому большинство web-разработчиков уделяют большое внимание дизайну web-страницы. Цветовая расцветка не должна бросаться в глаза, но и не должна быть блеклой, шрифты должны быть читаемы и соответствовать стандартным размерам. Для решения всех этих задач нам понадобится CSS [4].

1.3 Концепция сайта

Сайт выполнен в темных расслабляющих цветах, что создаёт атмосферу уюта. Данная цветовая палитра позволяет пользователю максимально сконцентрировать внимание на центральной части странице в которой будет предложен выбор персонажа.

 Сайт имеет тип лендинг, т.е вся информационная функция сайта реализуется на одной странице, это наиболее удобный метод, который позволяет пользователю максимально быстро взаимодействовать с сайтом.

По левую сторону мы видим 5 интерактивных иконок, они предназначены для выбора персонажа противника.

При нажатии на одну из них в центре страницы появляется окно со всеми игровыми персонажами, при нажатии на одного из них будет предложены 18 героев имеющих максимальное преимущество по игре, сразу после выбора иконка персонажа заблокируется и выбрать снова его будет нельзя во избежание ошибок.

Выбирая новых персонажей, на странице будут интерактивно меняться рекомендуемые герои, это позволяет принимать решения максимально быстро и выгодно. Структура рекомендуемых героев построена, так что персонажи с большим преимуществом расположены левее, такое расположение наиболее привычно для пользователя.

2 Создание структуры сайта

 Для решения данной задачи будем использовать язык разметки html,

Технология этого способа создания Web-страницы такова:

В текстовом редакторе, например блокнот, создается файл Web-страницы, расширение которой .htm. После редактирования открываем файл в браузере, например Google Chrome. Для вызова редактора Блокнот с целью редактирования файла Web-страницы во время ее просмотра в Internet Explorer, используется пункт меню Вид, Источник или в виде HTML. После сохранения файла и выхода из Блокнота для просмотра отредактированной страницы надо нажать клавишу F5 или кнопку «Обновить» в панели инструментов Google Chrome.

 Ниже в листинге 1 приведена часть кода на языке HTML которая описывает 5 героев класса “Power”. Каждому блоку принадлежит свое изображение определенных параметров и функция которая вызывается при нажатии на это изображение.

<div class = "power">

 <h1>Сила</h1>

 <div class = "enemy1">

<div class = "abaddon" onclick="abaddonimba();">

<img src=" /img/heroes/abaddon/m\_icon.jpg"width="70" alt="abaddon">

</div>

<div class = "brewmaster" onclick="brewimba();">

<img src="/img/heroes/brewmaster/m\_icon.jpg" width="70" alt="brewmaster">

 </div>

 <div class = "clockwerk" onclick="clockimba();">

<img src="/img/heroes/clockwerk/m\_icon.jpg" width="70" alt="clockwerk">

 </div>

<div class = "earthshaker" onclick = "earth2imba();">

<imgsrc="/img/heroes/earthshaker/m\_icon.jpg" width="70" alt="earhtshaker">

 </div>

<div class = "kunkka" onclick = "kunkkaimba();">

<img src=" /img/heroes/kunkka/m\_icon.jpg" width="70" alt="kunkka">

 </div>

Листинг 1 – Код HTML

Как можно видеть на рисунке 1 используя HTML я создал структуру списка в котором участвую данные персонажи.



Рисунок 1 - Список персонажей

2.1 Графическая составляющая сайта

 Сайт может существовать и без каскадных таблиц стилей, но исключая их разработчик лишает сайт возможности выглядеть визуально привлекательно. А без этой характеристики сайт вряд ли будет удобен для пользователей. CSS — формальный язык описания внешнего вида документа написанного на языке html, также может применяться к любым XML документам [5].

 Для максимально удобного использования всех функций сайта я решил спрятать меню выбора, которое станет доступно при нажатии на кнопку “выбора персонажа”. Эту функцию, как можно увидеть из листинга 2, я реализовал, используя CSS.

#hidenpool{

 width: 991px;

 height: 600px;

 background: #18211c;

 opacity: 0.95;

 border: 1px solid black;

 text-align: center;

 display:none;

 top: -20px;

 left: 200px;

}

#hidenpool p{

 display: none;

}

#hidenpool:target {

 display: block;

 position: absolute;

}

#hidenpool:target + .escape{

 display: none;

}

#hidenpool:target ~ .escape{

 display: block;

}

.escape{

 position:absolute;

 display: none;

 top:510px;

 left: 1100px;

}

.hero img{

 -moz-transition: all 1s ease-out;

 -o-transition: all 1s ease-out;

 -webkit-transition: all 1s ease-out;

-webkit-box-shadow: -5px -4px 27px -12px rgba(214,3,3,0.3);

-moz-box-shadow: -5px -4px 27px -12px rgba(214,3,3,0.3);

box-shadow: -5px -4px 27px -12px rgba(214,3,3,0.3);

}

.hero img:hover{

 -webkit-transform: scale(1.1);

 -moz-transform: scale(1.1);

 -o-transform: scale(1.1);

 -webkit-box-shadow: -5px -1px 46px 7px rgba(0,0,0,0.75);

-moz-box-shadow: -5px -1px 46px 7px rgba(0,0,0,0.75);

box-shadow: -5px -1px 46px 7px rgba(0,0,0,0.75);

}

Листинг 2 – Код CSS

 Как мы видим, из рисунка 2, используя CSS, мне удалось скрыть меню выбора, это позволит сделать страницу более структурировано и сэкономить место, также данный способ добавит интерактивности на сайт, благодаря чему пользователь сам решает, что в данный момент ему нужно.



Рисунок 2 - Скрытое меню выбора

 Освободившееся пространство я использую для реализации списка выбора рекомендуемых персонажей, всего их будет 18. Структура списка такова, что наиболее подходящий персонаж будет всегда левее, я посчитал что данный метод будет наиболее удобен для восприятия. Т.к с детства нас учат читать слева на право, таким образом, человеческий глаз первым делом обращает внимание на объекты расположенные немного левее центра и от верхнего левого угла в нижний правый.

 На рисунке 3 мы видим информационные объекты на которые человек обращает внимание в первую очередь, красным отмечены места наиболее интенсивного внимания пользователя, объекты раскрашенные синим напротив отражают места куда человеческий глаз смотрит в последнюю очередь. Таким образам данное расположение кажется мне наиболее оптимальным и позволит пользователю максимально быстро определяться с выбором.



Рисунок 3 – Информационные острова

 Такие очаги называются информационными островами, поэтому наиболее важную информацию всегда следует помещать по пути человеческого взора [7].

Некоторые люди даже применяют навык чтения наискосок для более быстрого получения информации от источника. Не говоря уже о информационных ресурсах где нужно максимально приковывать внимание пользователя [6].

 Благодаря CSS после выбора персонажа, список скрывается и нам предлагают на выбор 18 героев, которых мы можем видеть на рисунке 4. Каждый раз, когда мы выбираем нового персонажа, сайт будет предлагать наиболее рациональный выбор.



Рисунок 4 - Рекомендации к выбору

В зависимости от того каких героев выбрал пользователь, сайт сопоставит данные и выведет наиболее подходящих персонажей в данной ситуации. В конечном итоге пользователь должен будет выбрать 5 героев, тогда сайт предоставит наиболее полную картину. Для предупреждения ошибок сайт блокирует возможность повторного выбора персонажа, для устранения возможных ошибок.

2.2 Динамические объекты сайта

 Для того чтобы на сайте появилась возможность выбора, замены и показа рекомендаций я использовал мультипарадигменный язык программирования JavaScript на нем написано порядка 95% всего кода данного сайта. Обычно JS используется как встраиваемый язык для web-страниц, по средствам которого web-страница становится более интерактивной [8].

 Ниже, в листинге 3, приведен фрагмент кода на языке JavaScript, который отвечает за реализацию функции “lycanimba” которая соответствует герою lycan.

function lycanimba() {

if (lycan1 == 0){

showhide();

meepo.points = meepo.points + 6.94;

phantom\_lancer.points = phantom\_lancer.points + 4.75;

bristleback.points = bristleback.points + 4.68;

naga\_siren.points = naga\_siren.points + 4.33;

timbersaw.points = timbersaw.points + 4.20;

visage.points = visage.points + 3.74;

axe.points = axe.points + 3.37;

broodmother.points = broodmother.points + 3.26;

bloodseeker.points = bloodseeker.points + 2.96;

spectre.points = spectre.points - 5.10;

zeus.points = zeus.points - 4.20;

pudge.points = pudge.points - 3.55;

clockwerk.points = clockwerk.points - 3.39;

silencer.points = silencer.points - 3.07;

nyx\_assassin.points = nyx\_assassin.points - 3.06;

centaur\_warrunner.points = centaur\_warrunner.points - 2.84;

doom.points = doom.points - 2.78 ;

heroes.sort(function(a, b) {

 return parseFloat(a.points) - parseFloat(b.points);

});

if (hero1 == 1) {

var

img = document.getElementById("hero111"); img.src = '/img/heroes/lycan/m\_icon.jpg';

hero1 = 2;

}

zamena();

lycan1 = 1;

return location.href = '#'

hero2 = 2;

}

if (hero3 == 1) {

var img = document.getElementById("hero333");

img.src = ' /img/heroes/magnus/m\_icon.jpg';

hero3 = 2;

}

if (hero4 == 1) {

var img = document.getElementById("hero444");

img.src = ' /img/heroes/magnus/m\_icon.jpg';

hero4 = 2;

}

if (hero5 == 1) {

var img = document.getElementById("hero555");

img.src = ' /img/heroes/magnus/m\_icon.jpg';

hero5 = 2;

}

zamena();

magnus1 = 1;

return location.href = '#'

}

Листинг 3 – Код Java Script

 Как мы видим из кода, мы задаем параметры зачисления ”очков преимущества” или наоборот отнимаем их, в зависимости от героя. Данная информация взята из официальной игровой статистики за определенный промежуток времени. После этого функция сортирует персонажей (от большего к меньшему) и вызывает другую функцию, которая выводит на экран рекомендуемых персонажей взятых на основе “очков преимущества”. После этого на место кнопки выбора персонажей помещается изображение выбранного героя и кнопка блокируется для предупреждения ошибок неоднозначного выбора. Подобного кода в приложении довольно много, т.к. я реализую подобные действия для каждого героя, которых в игре 115 штук. Кроме добавления “очков преимущества” также идет удалении их, для максимально точного подсчета результата. Это сделано для того чтобы минимизировать погрешность и на выходе пользователь смог увидеть наиболее подходящих персонажей.

2.3 Программная реализация

 Когда сайт готов к выпуску в сеть интернет, то ему необходимо выбрать доменное имя сайта и подобрать подходящий хостинг. Доменное — символьное имя, которое необходимо ввести в адресную строку браузера чтобы попасть на определенный сайт. Доменное имя необходимо чтобы сайт могли найти среди всех остальных, оно подобно вашему домашнему адресу, только в сети интернет [9]. Хостинг — услуга по размещению информации на сервере, находящимся в сети интернет.

 Для того чтобы сайт появился интернете я купил доменное имя на год стоимость данной услуги обошлась в 200р с доменом верхнего уровня .RU

Далее я арендовал хостинг на 3 месяца на другом портале.

 Так как, домен и хостинг я приобретал на разных ресурсах мне пришлось прописать DNS для того чтобы связать домен и хостинг. Воспользовавшись

Файловым менеджером в панели управления сайтом, я загрузил необходимые файлы для работы интернет страницы [10]. На рисунке 5 можно увидеть как расположены файлы на сервере, выглядят они подобно информации на локальном носителе, только в данном случае доступ к ним возможен с любого устройства по средствам сети интернет.



Рисунок 5 - Файлы на сервере

ЗАКЛЮЧЕНИЕ

Целью данного курсового проекта была разработка и создание структуры сайта для оптимизации выбора в онлайн-игре.

В рамках поставленной цели были решены следующие задачи:

1 Ознакомление со стратегиями разработки интернет-сайта и web-приложений.

2 Выбраны языки разработки: HTML, CSS, JS.

3 Разработан сценарий сайта.

4 Проведен анализ программ используемых для верстки и создания интернет-сайта

5 Разработан интерфейс сайта.

В результате выполнения курсового проекта с использованием современных Web-технологий был создан тематический интернет-сай для оптимизации выбора персонажей в компьютерной онлайн-игре, имеющий доменное имя smartpicker.ru

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Киберспорт как феномен // Scincepop – электронный научно-популярный журнал. – (Рус.). – URL: <https://sciencepop.ru/kibersport-kak-fenomen/> [14 апреля 2018].

2 Бабаев А. Создание сайтов / А. Бабаев. –М.: Питер, 2014. – 410 с.

3 Дунаев В. HTML, скрипты и стили (3-е издание) / В. Дунаев. –М.: Питер, 2012. –816 с.

4 Алексеев А.П. Введение в Web-дизайн / А.П. Алексеев. –М.: Солон-Пресс, 2008. – 192 с.

5 Круга С. Веб-Дизайн или "не заставляйте меня думать!" / C. Круга. –М.: [Символ-Плюс](http://www.ozon.ru/brand/856498/), 2008. – 224 с.

6 Прохоренок Н. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Н. Прохоренок. –М.: БХВ-Петербург, 2015. – 768 с.

7 Панфилов К. По ту сторону веб-страницы / К. Панфилов. –М.: ДМК Пресс, 2008. – 440 с.

8 Кроудер Д. Создание web-сайта для чайников / Д. Кроудер. –М.: Диалектика, 2009. – 352 с.

9 Русаков М. Как сделать сайт-визитку своими руками / М. Русаков. –М.: Интерет-издание, 2016. – 165 с.

1. Бердышев С.Н. Искусство оформления сайта / С.Н. Бердышев. –М.: Питер, 2010. – 148 с.