

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

Омский государственный университет
им. Ф.М. Достоевского

Кафедра «Логики и логического программирования»

КУРСОВАЯ РАБОТА
на тему: «Разработка модулей для web–системы обучения
программированию»

ВЫПОЛНИЛ:
Студент гр. ММС-902-о
очной формы обучения
Притужалов Евгений

НАУЧНЫЙ РУКОВОДИТЕЛЬ:
Шмаков С. В.

Содержание

1 Структура проекта «Тортуга»	5
2 Сокращение ссылки	6
3 Очистка экрана	8
4 Изменение толщины рисования	8
5 Изменение толщины рисования	9
Заключение	10
Литература	11
А	13

Компьютерное и программное обеспечение играет неограниченную роль в современном мире. С его помощью человечеству открылся новый уровень знаний, возможности развития и новые стремления. На сегодняшний день мы уже имеем немалые достижения, такие как мобильные телефоны, сеть World Wide Web, умная бытовая техника и так далее. Компьютерам доверяют все более и более ответственные «должности»: работа светофоров, военные «беспилотники», комплексы военно-стратегического назначения, работа авиатранспорта, системы поддержки жизнеобеспечения. С каждым днем микроэлектроника заполняет все новые и новые предметы, заставляя их работать на нас с большей пользой. По этой причине в настоящее время область информационных технологий играет лидирующую роль в обществе. Постоянно создаются и модифицируются информационные системы, что требует хорошей подготовки кадров в данной области, способных создавать новые впечатляющие технологии, которыми будут пользоваться миллиарды других людей. И эта тенденция будет только усиливаться. К примеру, в США в скором времени ожидается, что около миллиона вакансий программистов будут заполнены извне из-за недостатка профессиональных кадров внутри страны [1-2].

Для наиболее эффективного обеспечения такими кадрами необходимо с детства развивать мышление в данной области. Ведь как известно, с возрастом обучаемость падает, да и к тому же способности к программированию у всех различные, поэтому так важно заложить крепкий фундамент основ программирования в раннем возрасте, что бы в перспективе поднять планку программирования не только лучших учеников, но так же и средний уровень знаний [3].

Для обучения детей младшего и среднего возраста широко применяются графические исполнители. Программист при помощи команд управления неким роботом, который может передвигаться по плоскости и чертить на ней рисунки. Для этого используется простой язык программирования, включающий систему команд непосредственного управления роботом, т.е. команды перемещения и рисования, а также управляющие конструкции, которые позволяют организовать повторения, ветвления, выделить какие-то действия в подпрограмму или процедуру.

В качестве такого языка часто используется Logo[16]. В Омской физико-математической школе № 64 в 90-е годы прошлого века в течение долгого времени использовалась программа MSW Logo от Softronic's Home Page [17]. К сожалению развитие этого продукта остановилось, и даже интерфейс продукта выглядит несколько инородно в современных версиях OS Windows. На данный момент лицей № 64 уже давно не использует этот продукт. Сейчас там используется Scratch - проект Массачусетского технологического института [18].

ООО «Образование IT» одна из омских организаций, которая занимается обучением школьников программированию в рамках проекта «Школа программиста». Её не устроили известные графические исполнители, реализованные в виде десктопных приложений, в силу громоздкости интерфейса, сложности установки, ограниченности поддерживаемых платформ. С одной стороны, хотелось иметь продукт, который бы не требовал установки вовсе, или устанавливался бы очень легко, а так же поддерживался всеми популярными платформами: Windows, OS X, Linux. Выбор упал на web-приложение, ведь для его работы необходим только современный браузер, который с большой вероятностью используется на компьютере в учебных заведениях и дома у учеников. С другой стороны, хотелось получить удобство и гибкость в создании и поддержке методических материалов, а так же иметь независимость от наличия интернета, чтобы иметь возможность заниматься в летних выездных школах в оздоровительных лагерях, а так же в рядовых городских образовательных заведениях, где перебои с интернетом – вещь весьма обычная. Поэтому «Школу программиста» не устроили существующие, на данный момент, Web аналоги [4-6, 20-26], которые не дают удобной возможности организовывать библиотеку методических материалов, а так же, как правило, требуют наличие соединения с сервером. Поэтому было принято решение создать собственного графического исполнителя, который обладал бы всеми необходимыми свойствами: работал бы в браузере, позволял организовывать методические материалы, не был бы привязан к интернету. Так появился проект «Тортуга».

«Тортуга» уже использовался для обучения школьников в течение весны 2013 года. В рамках этого проекта передо мной стояла задача по расширению функциональных возможностей программы. Описанные в данной работе результаты на данный момент включены в релизную версию программы, которая использовалась преподавателями «Школы программиста» в летних выездных и при школьных лагерях в июне 2013 года.

Глава 1

Структура проекта «Тортуга»

Программа «Тортуга» является проектом компании «Образование IT», которая ставит перед собой цель обучение школьников основам программирования. Программа представляет собой Web–приложение, построенное на языке JavaScript, с использованием технологий HTML5 и CSS3. Благодаря применению данных технологий, нет необходимости в постоянном доступе к сети Интернет. Использовать «Тортугу» можно, скопировав на флешку или любой другой цифровой носитель страницу сайта, где расположено данное приложение.

Один из сценариев работы с «Тортугой» следующий. Преподаватель на странице `constructor.html` создает урок: формулирует тексты и название задачи. Информация об уроке особым образом кодируется и формируется ссылка с текстом этого урока, пройдя по которой, откроется страница `index.html` с текстом задания. Для выполнения задачи ученику потребуется создать объект Черепашка и, управляя ее действиями через команды, достигнуть поставленной задачи.

Часто учителям не удается заинтересовать школьников предметом лишь потому, что не могут найти хорошего задания. По этому в «Тортуге» любой человек может самостоятельно построить именно такой урок, который будет ему необходим, что важно для применения в школах. Вариант готового урока можно увидеть на рис. 1.

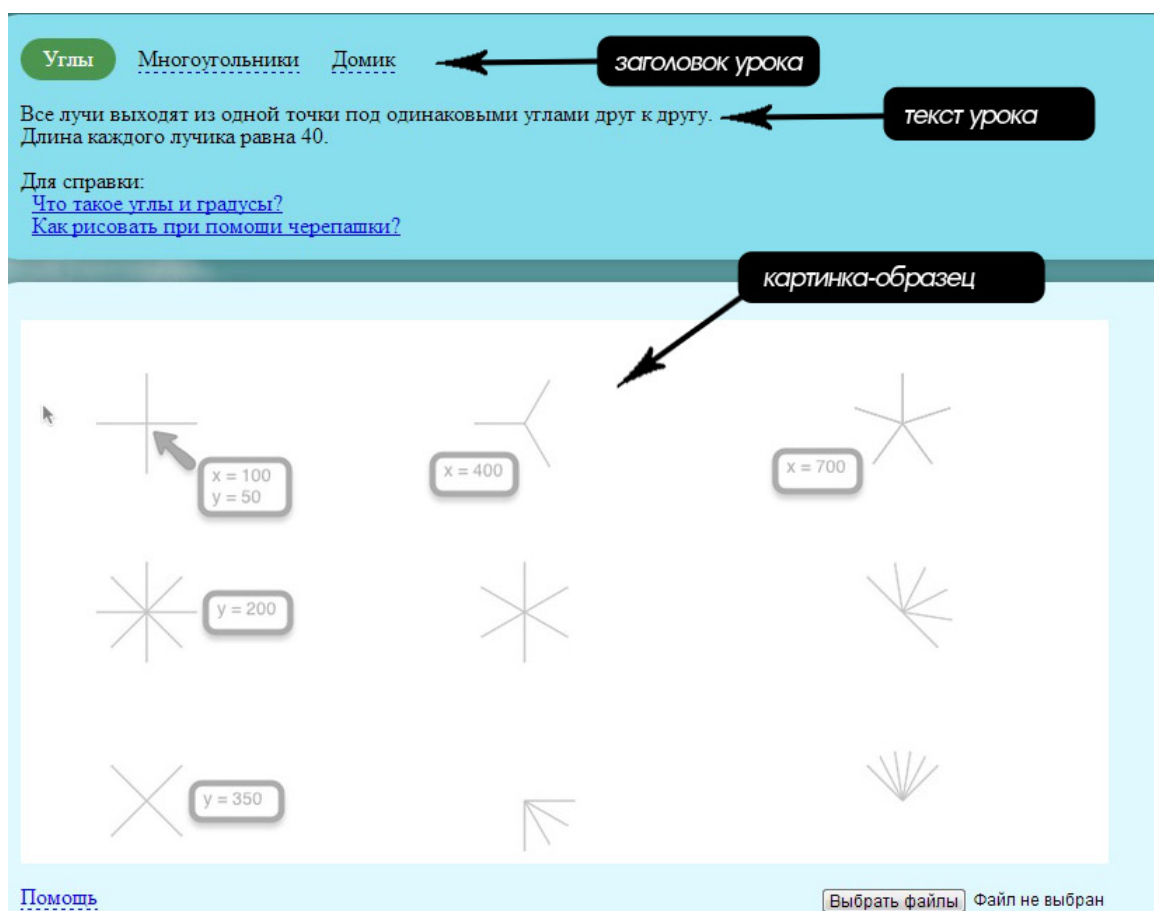


Рисунок 1.1: Пример урока.

Для создания урока на странице `constructor.html` в текстовое окно вводится текст урока и, при нажатии кнопки «Создать», содержимое текстового поля извлекается, сжимается, кодируется base64 и создается ссылка на урок.

Процесс разработки «Тортуги» ориентирован на то, чтобы программа работала в последних версиях Chrome и Яндекс браузер. Но на данный момент работает и в Firefox, Safari, IE 8+, Opera, но при возникновении проблем поддержка этих браузеров осуществляется по остаточному принципу.

Глава 2

Сокращение ссылки

Как было сказано выше, сценарий работы с «Тортугой» – это работа с подготовленными уроками, которые открываются по ссылке вида:

<http://tortuga/index.html?b4/wMsYxLAWL/+Ml5/HjEw> (1)

Урок содержит тексты задачи, ссылки на картинки–обазцы, поэтому ссылка получается довольно длинной, например:

<http://tortuga/index.html?7VcxTkMxDL2K9eeqB+gMGysTZTCJWywlTkhshIS4O04LN2AA5O3nx35+fk+OkvdNWQtth+2uDaraFvqF3EobMFkBK+kOUpNJSUltAGbuPDmxnIEK++6k7BlAbLO2fBSl2j2dJXHmbKJgCgWfvACQXsEJKp4FAQu/GO7hXoGEq6ND5fXx6kusu6O8GE+QNNVYBnqjkVhRuQlYKVhTu0KvIKelSl0wuXswEDr36qzaUS5NeDHdw83CRFMCHuZcrv2ywKA+6Jkk0/Dm/cdrK9a9HjkhbxZoTnIoLuVbJu/J4GRnRgVZlKDj8IWNPdy+JepKtrRGVpKSMnjknXOqCujyVH6aJxJlpRLLS+brHQE7wLa6eRaI2SaNNZubWURwSUSuyLzSlur+7Dwz1u47TZ28ed2eHiP0fxXvs6Rwss4ZsPCsDAs/JET1cPS4L70iZM1xjIs/A1j+bGLi2uMZFgYb4/wMsYxLAWL/+Ml5/HjEw>== (2)

Делиться ссылкой такого размера, отправлять по почте, размещать в соц–сетях, использовать в презентациях весьма неудобно. В связи с чем появилась необходимость укорачивания ссылки до 20 знаков и менее. Уже довольно давно существуют способы их сокращения при помощи сайтов–сокращателей таких как: *www.bitly.com*, *www.gg.gg*, *www.goo.gl* и др. Необходимо зайти на соответствующую страницу и после ввода в текстовое окно длинной ссылки, создастся короткая, вида:

<http://goo.gl/fbsS> (3)

при переходе по которой происходит перенаправление на длинную. Но такое ручное использование удобно только для единичных вариантов. Что бы создателю уроков не приходилось каждый раз выполнять все эти рутинные операции, возникла задача автоматизировать процесс сокращения ссылок. Для этого было решено воспользоваться API, которое предоставляет Google. Была найдена и использована сторонняя библиотека Jsonlib [27] (автор Девид Бай/David Bau). С ее помощью можно обмениваться запросами в формате JSON технологией AJAX не напрямую, а через сторонний сервер, который осуществляет обращения и возвращает ответ. В связи с чем были созданы функции *getShortenURL* и *parseShortenedResponse*. Одна из которых отправляет запрос с передачей длинной ссылки, а вторая, получив ответ в формате

JSON, и, распарсив, выводит на страницу `constructor.html` в текстовое окно. Если по какой-либо причине запрос не был осуществлен или ответ не был получен, на страницу выводится длинный URL (см. приложение А.1).

Глава 3

Очистка экрана

При выполнении урока ошибки в алгоритмах рисования неизбежны, и единственный способ очистки экрана - это обновление страницы. Но страница долго перегружается и приходится заново создавать черепаху с соответствующими настройками положения на экране, ее цветом и толщиной рисования, что отвлекает и создает неудобства. Решением проблемы была бы возможность очистки экрана без обновления страницы. Для этого в библиотеку функций была добавлена еще одна публичная функция *clearCanvas()*, очищающая прямоугольник размером с Canvas (см. приложение А.2). Помимо решения такой проблемы очистка экрана дает дополнительные возможности в программировании, например создание анимации.

Глава 4

Изменение толщины рисования

Дополнительное разнообразие достигается при возможности изменения толщины рисования черепахи. Для реализации этого функционала в объект, представляющий черепаху, добавлено свойство `width`, а в прототип (объект, представляющий свойства и методы, общие для всех черепах) была добавлена функция *setWidth*, внутри которой передаваемая в качестве параметра толщина присваивается конкретной черепахе. Так как доступ к графическому объекту Canvas доступен через графические методы, с помощью которых манипуляции

с черепахой отражаются на Canvas, то добавлена открытая функция *setWidth* (см.приложение А.3) [10].

Глава 5

Изменение толщины рисования

Во время использования созданных уроков возможно выявление неточностей, ошибок в орфографии и других недочетов, которые требуют исправления. Для этого функционал приложения был дополнен инструментом для редактирования уроков. Для его создания на страницу создания уроков *constructor.html* было добавлено текстовое поле *lessonInput*, в которое вставляется ссылка на уже имеющийся урок. При нажатии на кнопку «Изменить урок» данные из *lessonInput* передаются в свойство *givLessonArea* объекта Tortuga для вызова закрытой функции *updateArea*, которая, в свою очередь, извлекает текст из ссылки, разархивирует, парсит, и вставляет в текстовое окно в том виде, в котором он был введен. После чего текст урока можно править и далее, при нажатии кнопки «создать урок», формируется новая ссылка на исправленный урок. (см. приложение А.4) [14-15].

Заключение

Web-приложение «Тортуга» создано для обучения школьников основам программирования, и для достижения максимального результата этот процесс должен быть удобен как ученику, при выполнении заданий, так и учителю, при формировании урока. Длинные ссылки вида (2), которыми затруднительно делиться, использовать в презентациях, размещать в соц-сетях, отсутствие инструмента для редактирования готовых уроков, отсутствие функции очистки экрана являлись слабым местом приложения, требующие исправления.

Достигнутые результаты:

- Сокращение ссылок
- Очистка экрана `clearCanvas`
- Изменение толщины рисования
- Редактирование уроков

Литература

1. Руденко А. Е. Альманах современной науки и образования // Роль программирования в развитии современной цивилизации № 6. Тамбов: Грамота, 2009 год.
2. Выбор профессии – программист. URL: <http://fisdel.ru/faqs/2461-vybor-professii-programmist> (дата обращения 11.06.2013)
3. Руководители IT компаний призывают изучать программирование. URL: <http://www.3dnews.ru/news/642114/> (дата обращения 11.06.2013)
4. Blockly. Visual programming editor. URL: <https://code.google.com/p/blockly/> (дата обращения 06.06.2013)
5. Codecademy. URL: <http://www.codecademy.com/> (дата обращения 06.06.2013)
6. Microsoft Kodu. URL: <http://fuse.microsoft.com/projects/kodu> (дата обращения 06.06.2013)
7. Обмен данными для документов с разных доменов. URL: <http://javascript.ru/ajax/cross-domain-scripting> (дата обращения 03.06.2013)
8. XMLHttpRequest: описание, применение, частые проблемы. URL: <http://xmlhttprequest.ru/> (дата обращения 29.05.2013)
9. Современный учебник по JavaScript. URL: <http://learn.javascript.ru/> (дата обращения 13.06.2013)
10. Прототип объекта. URL: <http://javascript.ru/Object/prototype> (дата обращения 01.06.2013)
11. Дронов В. JavaScript народные советы. Санкт-Петербург: БХВ-Петербург, 2012 год.
12. URL Shortener API. URL: https://developers.google.com/url-shortener/v1/getting_started (дата обращения 06.06.2013)

13. Google APIs Client Library for JavaScript (Beta). URL: <https://developers.google.com/api-client-library/javascript/samples/samples> (дата обращения 07.06.2013)
14. Поиск элементов в DOM. URL: <http://javascript.ru/tutorial/dom/search> (дата обращения 07.06.2013)
15. Работа со строками. URL: <http://htmlweb.ru/java/string.php> (дата обращения 14.06.2013)
16. Шапошникова С.В. изучение языка программирования Logo в среде KТurtle: Курс для детей и подростков по программированию: лаборатория юного линуксоида, 2011 год.
17. Welcome to MSWLogo. URL: <http://www.softronix.com/logo.html> (дата обращения 16.06.2013)
18. Scratch. URL: <http://scratch.mit.edu/> (дата обращения 17.06.2013)
19. Стефанов С. JavaScript Шаблоны. Санкт-Петербург – Москва. O'REILLY, 2011 год.
20. MIT App Inventor. URL: <http://appinventor.mit.edu/> (дата обращения 17.06.2013)
21. Alice. URL: <http://www.alice.org/index.php> (дата обращения 17.06.2013)
22. Online Logo. URL: <http://www.transum.org/software/Logo/> (дата обращения 17.06.2013)
23. Logo Interpreter. URL: <http://www.calormen.com/logo/> (дата обращения 17.06.2013)
24. Logo Tortoise. URL: <http://andyhd.github.io/Logo-Tortoise/> (дата обращения 17.06.2013)
25. Tortue Logo. URL: <http://tortue-logo.fr/en/logo-turtle> (дата обращения 17.06.2013)
26. Papert. URL: <http://logo.twentygototen.org/EcsvFC4p> (дата обращения 17.06.2013)
27. JsonLib. URL: <http://call.jsonlib.com/> (дата обращения 17.06.2013)

Приложение А

А1. Сокращение ссылки

Отправляет запрос с передачей длинной ссылки:

```
var parseShortenedResponse = function(m, longUrl) {
    console.log(m, m.content);
    var url = null;
    try {
        url = JSON.parse(m.content).id;
        if (typeof url !== 'string') url = longUrl;
    } catch (e) {
        url = longUrl;
    }
    linkarea.innerHTML = "";
    var textinput = document.createElement("INPUT");
    textinput.type = "text";
    textinput.disabled = true;
    textinput.value = url;

    var link = document.createElement("A");
    link.href = url;
    link.innerHTML = "Try lesson";

    linkarea.appendChild(textinput);
    linkarea.appendChild(link);
    textinput.select();
}
```

Получает ответ в формате JSON и, распарсив, выводит на страницу constructor.html в текстовое окно.

```
var getShortenURL = function(url) {
    jsonlib.fetch({
```

```

        url: 'https://www.googleapis.com/urlshortener/v1/
        header: 'Content-Type: application/json',
        method: 'POST',
        data: JSON.stringify({longUrl: url})
    },
    function(m) {parseShortenedResponse(m, url)});
}

```

A2. Очистка экрана

Очищает прямоугольник, размером с Canvas

```

...
var clearCanvas;
Tortuga.initDrawing = function(canvas){
...
    clearCanvas = function() {
        ctx.clearRect(0, 0, canvas.width, canvas.height);
    }
}

```

A3. Изменение толщины рисования

В объект, представляющий черепаху, добавлено свойство width:

```

var Tortoise = function(xx, yy, color, width, tortoiseConta
{
    ...
    this.width = width || 1;
    ...
}

```

В прототип (объект, представляющий свойства и методы, общие для всех черепах) была добавлена функция setWidth:

```

var proto = {
    go : function(t, length){
        ...
        if(t.isDrawing){

```

```

        ...
        oldWidth = setWidth(t.width);
        ...
        setWidth(oldWidth);
    }
},
...
setWidth : function(t, w){t.width = w || t.width}
}

```

Добавлена публичная функция `setWidth` для отражения на Canvas действий черепахи:

```

Tortuga.initDrawing = function(canvas){
    ...
    setWidth = function(width){
        ctx.lineWidth = width;
    }
}

```

А4. Редактирование уроков

Формирование необходимых элементов для ввода и вывода информации:

```

<input type="text" id="lessonInput">
<button id="changebutton">Изменить урок</button><br>
<textarea id="area"></textarea><br>
<button id="createbutton">Создать урок</button><br>
<div id="linkarea"></div>

```

Передача данных из текстового поля `lessonInput` в объект Черепаха:

```

<script>
var init = function(){
    ...
    Tortuga.givLessonArea(document.getElementById("area"),
        document.getElementById("changebutton"),
        document.getElementById("lessonInput"));
}
</script>

```

Вызов закрытой функции `updateArea`:

```
Tortuga.givLessonArea = function(area, button, input){
    button.onclick = function(e){updateArea(area.value, input.value);}
}
```

Обработка полученных из lessonInput данных:

```
var updateArea = function (areaValue, inputValue){
    var t = Tortuga.ParamsUtil.getLessonTextFromGetUriValue(inputValue);
    var paramBegin = null;
    var paramAnd = null;
    var paramText = null;
    var resultText = "";

    paramBegin = t.indexOf(':');
    paramAnd = t.indexOf('&', paramBegin + 2);

    while (paramBegin > 0){
        paramText = t.substr(paramBegin + 2, paramAnd - paramBegin - 2);
        resultText = resultText + paramText + '\n\n';
        t = t.substr(paramAnd + 2);
        paramBegin = t.indexOf(':');
        paramAnd = t.indexOf('&', paramBegin + 2);
    }

    document.getElementById('area').value = resultText;
}
```

Извлечение текста урока из ссылки:

```
var getUriValue = function(u){
    var str = u;
    var vhozhd = str.indexOf('?');
    var result = str.substr(vhozhd + 1);
    return result;
}

var getLessonTextFromGetUriValue = function(u){
    return prezip_to_utf8(RawDeflate.inflate(atob(getUriValue(u))))
}
```