

CS297 Report

JavaScript Game Engine for Mobile using HTML5

by

Nakul Vishwas Natu

Nakul.natu@gmail.com

Fall 2011

Advisor: Dr. Chris Pollett

San José State University

Department of Computer Science

One Washington Square

San Jose, CA 95192-0249

Ph: (408) 924-5145

<http://www.cs.sjsu.edu/faculty/pollett/>

Table of Contents

- 1. Introduction..... 3
- 2. Deliverable 1 4
- 3. Deliverable 2..... 6
- 4. Deliverable 3..... 8
- 5. Deliverable 4..... 10
- 6. Conclusion 12
- 7. References..... 13

1. Introduction

The goal of this project is to develop a mobile JavaScript game engine for mobile devices which uses HTML5. We will be able to create platform independent mobile games using this engine. To achieve game persistence, HTML5 local storage will be used. The created games will be stored on the desktop of the device in the form of a data url, and this can be clicked just like a native app. When the user launches the data url, the game will open in a browser but the user will feel he is in a native app. The data url contains the complete game. HTML5 has many restrictions on use of local storage, connecting it to the data url will be a really challenging aspect of this project. At the completion of CS298 I will also provide test games created using the engine to support my project.

There were total four tasks to be carried out in CS297. Creating a data url for game "Pong" and store the high scores, run test game created using "Crafty" a JavaScript game, developing HTML5 application using Local Storage, creating an application which will connect data URL to HTML5 Local Storage.

This CS297 report details these activities that were carried out to prepare for the final CS298 project. The four sections of the report contain the details about each task explaining what were the goals , experimentation carried out , learning and result. The last two sections contain the conclusion and the references.

2. Deliverable 1

Goal

To create a data url for game "Pong" and store the high scores.

Description

At the end of my project I have to deliver game engine , hence I had to first learn how game loop run in JavaScript. The purpose of this deliverable was to create a basic “Pong” game which should run on mobile device. Also I wanted to make sure if data url works on the device.

Data URL

Data URL is url which includes small data items. It behaves as if data is included externally. Some applications also need to embed media types into URL.

Syntax: data:[<mediatype>][;base64],<data>

The mediatype is a MIME type string, such as "image/jpeg" for a JPEG image file.

Default is : text/plain;charset=US-ASCII

";base64" means that the data is encoded as base64

As I am trying to develop a game engine, game produced by it should have a faster response and loading time. There are also lot of multimedia files involved in a game. Data URL consumes less bandwidth. If data url is used for transferring many small files takes less time than TCP. Also as the data is in line, it frees up a download connection for other content. It is possible to manage a multimedia page as a single file.

It has some disadvantages also , such as data is downloaded every time the containing documents are downloaded. Content must be re-encoded and re-embedded every time a change is made. Lack of error handling.

Touch Events

One of the paddle in the game is controlled by human and other is bot. I used the touchmove and touchstart event to control the paddle.

We can also set two variables which control movement of ball and speed of bot paddle. The game also displays a score. I used data urls for all the images present in the game. I also converted the game itself into a data url, so you can access it from anywhere.

Result

You have to move the paddle up and down using touch or mouse. To score against the opponent, you have to hit the ball past bot's paddle. You can see your score above the frame.



Click on Paddle to Start Game.

Figure 1 : Pong in JavaScript

3. Deliverable 2

Goal

The goal for this deliverable was to create a game using "Crafty" a JavaScript game engine.

Description

I have to build a JavaScript game engine at the end of my CS298 project. To do this it was necessary to examine existing game engine, get some experience in developing games using that. It was important to look into working of the game engine, and what facilities it should provide. I created a role playing game using the demo and documentation available.

“Crafty” JavaScript Game Engine

Crafty is a JavaScript game engine which helps to develop games for the browser. It has a modular design, which helps in creating reusable components and extensions. Rather than having a long hierarchy of inheritance, Crafty uses the Entity-Component paradigm or data oriented. Basically, every game object is an entity and every bit of functionality is a component. All you need to do to give your entities functionality is to assign some components.

The coding style is loosely based on DOM libraries in a sense that you can select entities by the components assigned and entities methods are chainable. It also has collision detection, sprite map support as well as Canvas support. It also have audio and video elements of HTML5. I chose “Crafty” because it is based on JavaScript and HTML5. The famous “Fruit Ninja” is based on Crafty.

Implementation of Sprite Map

Sprite map is basically collection of different images. If you want to get different images using different HTTP request , it will consume lot of time. Sprite Map reduces that time as it delivers all the images in one map. It also requires less code or markup language

Result

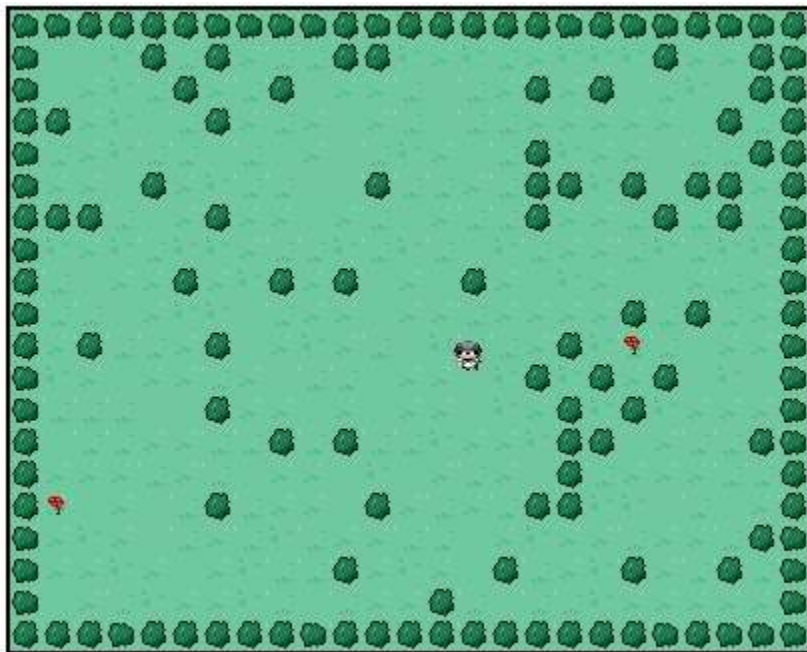


Figure 2 Flower Picker using Crafty

I created a small role playing game called Flower Picker using the sample sprite map provided with Crafty. I created a small park with bushes and flowers. Aim of the game is to pick all the flowers present. I also added some animation too like color and wind direction to flower. To simulate picking the flower , I used collision detection and also inserted a sound using Crafty's audio component. The game finishes when all flowers are picked. You can use arrow keys to move the girl. I tried to implement the touch events , but I was not successful. I will try to add listeners for these event in my own code to get it working.

4. Deliverable 3

Goal

The goal of this deliverable was to learn and use the HTML5 local storage.

Description

I am going to use HTML5 local storage for game persistence. This deliverable was designed to get acquainted with local storage and how it behave on the mobile platforms. I added high score functionality to my previous Flower Picker game using local storage. I also did some small changes in how my game will be displayed and played.

HTML5 Local Storage

From the W3C's website : Local Storage is designed for storage that spans multiple windows, and lasts beyond the current session. In particular, Web applications may wish to store megabytes of user data, such as entire user-authored documents or a user's mailbox, on the client side for performance reasons. Local storage store the data in terms of key-value pair.

The api made available to access HTML5 local storage is "localStorage". You can just access this api and see if your browser support local storage or not. It returns the storage area or the security exception.

Api provides two methods call setItem(key,value) which stores value for the specific key. While getItem(key) retrieves the value for key. W3C also provides another method called clear() which removes all the key-value pairs stored.

Timer

I have used Window's setTimeout method to implement the timer. It calls a function or evaluates an expression after a specified number of milliseconds.

Syntax : setTimeout(code,millisec,lang) where language is optional

Result

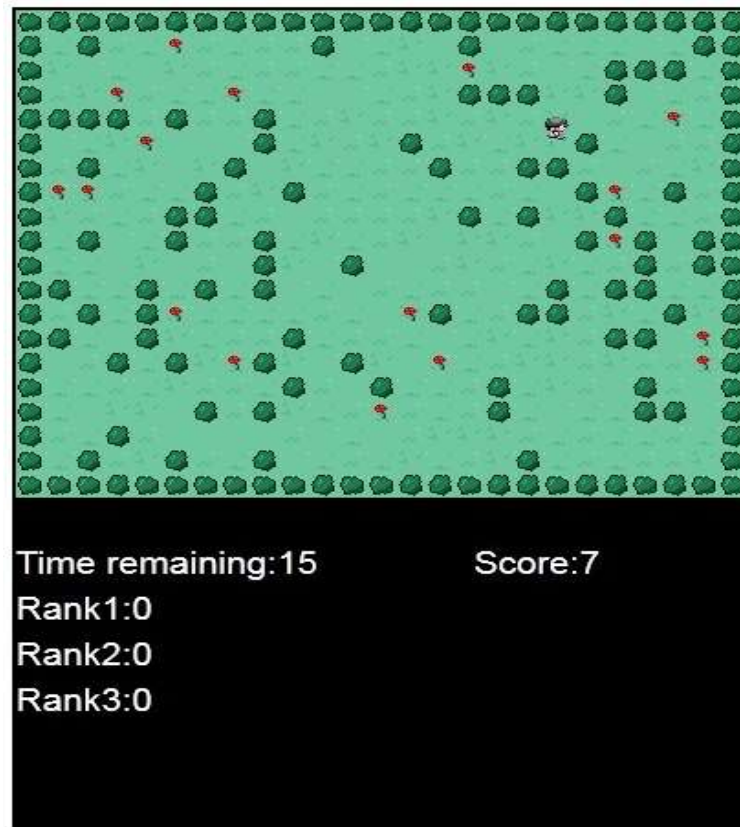


Figure 3 Flower Picker with High Scores

I added the timer of 30 seconds to the game. Now goal of the game is pick as many flowers you can in those 30 seconds. I display the remaining time and current user score. Game also displays current 3 highest scores. If player manages to score more than that he will be added to those ranks. I also added a functionality of different screen to the game such as loading , plating and ending screen using Crafty's scene capabilities.

5. Deliverable 4

Goal

This deliverables main goal was to create an application which will connect Data URL to HTML5 Local Storage

Description

I am going to use local storage for game persistence and data url for giving game native app feel. I already checked if both are working separately. This deliverable main purpose is to try and fit both of them together.

Small Test URL

I tested if data url work with HTML5 local storage in this deliverable. First I tried to convert the Flower Picker game created earlier into data URL as it has all the functionality. But it did not work, hence I thought of creating a tester small application to test if it works. Test application is a data url which just initializes local storage, sets the value and gets the value

```
window.onload=function()
{
    alert("hi");
    localStorage.setItem("highscore1",0);
    alert("stored");
    alert(localStorage.getItem("highscore1"));
};
```

BookMarklet

A bookmarklet is Unobtrusive JavaScript stored as the URL of a bookmark in a web browser or as a hyperlink on a web page. Bookmarklets are JavaScript functions which are directly inputted into browser address bar, and they are run without any http / ftp tags.

Bookmarklet used:

```
javascript:(function()  
  
  {  
  
    alert("hi");  
  
    localStorage.setItem("highscore1",0);  
  
    alert("stored");  
  
    alert(localStorage.getItem("highscore1"));  
  
  });
```

Results

When I carried out the experiment of test data url, different browsers gave me different results.

Firefox - Worked Chrome - Did not work Safari on iPhone - Did not work Android web browser - Did not work.

I also got different results for bookmarklet. It worked for Chrome and Safari - iPhone but failed on Firefox and Android web browser. We have to get a workaround working or find out some other way to give the game native app feel.

6. Conclusion

Purpose of CS297 was to check if the project is feasible or not. Each deliverable was designed to check different tasks involved in the project. First deliverable was to check if data urls work on device. The second was to see how game engine works. The third deliverable was to check html5 local storage and how it behaves. The last deliverable was designed to check the compatibility of data url with html5 local storage. As I have gone through the deliverables, and successfully completed each one, we can see that it is possible to have game engine in JavaScript for mobile devices which can have data url and html5 local storage for game persistence. But as we see in last task we have varying result for compatibility of data url and local storage. It is hard to get both work at the same time due to some security issues. We can solve that problem on iphone using bookmarklets. But on Android it still does not work. I have to find out a way to get the native app look to game created with my game engine. I still can use data urls as images , texts , sound or video.

7. References

[1] L. Masinter (1998). The "data" URL scheme. <http://www.ietf.org/rfc/rfc2397>

[2] Crafty game engine. <http://craftyjs.com/api/>

[3] Crafty Tutorial. <http://craftyjs.com/tutorials.php>

[4] Rick Rogers . Developing portable mobile web applications(2010). Linux Journal Volume 2010 Issue 197.

[5] W3C. Web Storage (2011). <http://dev.w3.org/html5/webstorage/>