Institutionen för datavetenskap
Department of Computer science

Examensarbete

Performance test and optimize in HTML5-based web game: a case study of Flappy Bird

by

Ping Liu

LIU-IDA/LITH-EX-A--15/001--SE

2015-11-18

Linköpings universitet

Linköpings universitet
Institutionen för datavetenskap

Examensarbete

Performance test and optimize in HTML5-based web game: a case study of Flappy Bird

by

Ping Liu

LIU-IDA/LITH-EX-A--15/001--SE

2015-11-18

Handledare: Min handledare
Examinator: Min examinator

Abstract

Abstract.tex

Acknowledgments

Acknowledgments.tex

Contents

List of Figures

List of Tables

1   Introduction

Since the HTML5 released and especially new element called Canvas, The percentage of web games in the whole video gaming market is increasing day by day. And what's more, the hot market return to the improvement of web game developing environment, such as open source web game engine and some useful web libraries. With Canvas and WebGL, the visual effect of games on website can be as good as that on the desktop application.

This thesis will take Flappy Bird as a case study, Flappy Bird was released in May 2013 and in early 2014 it got a amazing rise in popularity and became the most downloaded free game in the IOS App Store. Now, there are different versions realized on website and we will take hyspace's flappy as a case study. The reasons for taking this game are fellows: firstly, it is open source you can see the code and you can take white box test, secondly, this game itself is quite simple but the code itself is quite typical of misunderstanding and wrong use of graphics function.

1.1   Motivation

Although, a lot of plugins and open source game engine can help us build a simple game very fast, but with the misunderstanding of Canvas and WebGL and also the wrong use of these plugins, and actually nowadays in order to attract customers and developing interesting games, the size of application inscreases rapidly, these application hit the performance wall. It is easy to find a lot of resource focus on the optimizing performance of websites, but there is little resource focus on the optimization of web games, especially for HTML5-based web game. As you know, for web games, you need more graphical operati ons and the users are more sensitive about the delay of the page and really need a quick feedback if anything changes. So performance is more important to web games than web applications.

It is quite interesting to find out how to test performance of a web game and try to optimize it. And we will mainly focus on the web games based on HTML5 by using Canvas. In order to show the optimization procedure, flappy bird will be taken as a case study to introduce our way of optimization performance and through this study, in order to find some general suggestions that can help you build high performance web games.

1.2 Aim

The ambition of this thesis project is to find out something that can really help us build high performance web games. It is clearly that there is always something that can be improved to our project. Usually, the process would be like this: First try to test the game and then make some improvement, and the test it again. Through this optimization procedure, the performance of the game will be better and better.

Another ambition of this thesis is to consider it as a good example of how the optimization procedure works. And by this study,a list of suggestions that can be taken through generally web games.To be specify, there are several aims of this thesis project:

1. To give the procedure of optimizing performance of flappy bird as a case study

2. To give suggestions of how to build high performance to general web games.

1.3 Research questions

By the motivation and aim above, our research are mainly focus on two parts, one is the optimization procedure through the case study of flappy bird, and another is the general suggestions of web games with high performance.

According to our aim, we figure out several interesting research questions:

1. How to optimize a web game based on HTML5 Canvas?

   The most important performance metrics are computer performance and render performance, for computer performance, it means the JavaScript computation,which also including algorithmic complexity and memory usage, and for render performance, it means the time cost on changing the data from numbers to pixels on computer.

2. How the optimization procedure works during the case study of flappy bird?

3. What you should take care of when you develop a web game with high performance?

4. How to use Chrome to test performance of your code?

1.4 Delimitations

This thesis is mainly talk about web games that based on HTML5 Canvas, and for the performance test, it is mainly based on the Google Chrome browser.

2 Theory

This chapter is used to provide to some technique background and theory for this thesis, so it will first introduce different performance metrics on web application, and then focus on the render performance and compute performance, and then different web performance test tools and the process of optimization will be clearly identified.

2.1 performance metrics

For web application, performance is particular important to user experience since the application become more and more complexity and users become more and more picky.[7] Performance can a be very widely topic and hard to measure,in order to defined the exactly performance metrics of web application[8], we usually view it from three pillars: Network, Compute and Render[3].

Network performance is the base of every performance strategy, there is nothing we can do before receive the resource. It is hard and impossible to make sure the network will always be perfect as expected, what we can do are mostly about compress the files, reduce the request times.

Compute performance is always related to JavaScript computation for web application, which including algorithmic complexity, memory usage patterns, and unoptimized code. To optimize algorithm can be an eternal topic and always play a critical role not only for web application but also for all the procedure. Memory usage in browser is always misunderstanding and can easily cause memory leak. Although JavaScript will automatically collect garbage based on references[6], but if the code is written carelessly and come cycle, which means two objects are created and reference on another, it can cause problems.

Render performance means the time cost on the transformation from data to pixels on the screen. Except the network part, render cost most of the waiting time before user get the response from the website. Whenever users do something related to the display of web page, try to give the browser the least amount of work to update the pixels on the screen, which means you should reduce both render size and complexity of painting operations as much as possible. Reduce render size means you should just update the change part rather than the whole page, and reduce complexity of painting operations means you should choose the cheapest way to paint your elements.

This thesis will mainly focus on the compute and render performance.

2.2 Performance Test Tools

Different performance aspects will use different performance test tools. This thesis will just focus on the Chrome browser and will only use the tools provided by Chrome.Specifically, we will use just Profiles panel and Timeline panel in Chrome DevTools.

Timeline panel provides an overview of where time is spent loading up your web application such as how long it takes to process DOM events, render page layouts or paint elements to the screen which gives you a record for each even that happens, displayed in a "waterfall" graph.[9] It allows you to drill down into three separate facets that can help discover why your application is slow: Events, Frames and actual Memory usage.

Profilers show us which functions take the most time, there are three types of profileing: JavaScript CPU profile, CSS selector profile and Heap snapshot[10]. CPU profiles show where execution time is spent in your page's functions. There are two views top down and bottom up. The Top Down view shows an overall picture of the calling structure while the bottom up give a list of functions by the impact of performance.

2.3 Related Work

There are some papers related to the user experience of web application. Some resources identify the Web metrics for evaluating the reliability and maintainability of hypermedia applications[2]. Some resources just discuss the performance from a wide range and not provide how to optimize performance[4]. Some papers just give some ideas to optimize web application[11], Some books give advise of how to develop a high performance web game[1]. There do exists some good websites talk about what should be concerned when implement web games and how to optimize web games[12], but these articles are quite messy and not rigorous. And we don't find any papers focus on how to optimize performance of web games.

There are some papers give us the idea of how to test performance of a web application, for example Jiang Guangzhu and Jiang Shujuan give us a quick testing model of web performance based on testing flow[13], and Zhu Kunhua gives us some performance improvement strategy of web application[5].

3 Method

This chapter is used to provide the method of how to optimize performance of Flappy Bird, so the process of performance optimization(including performance testing, Analysis, Optimization) will be introduced first, then each part will be described carefully.Finally, according to the different metrics of performance, different performance optimization methods will be taken.

3.1 Performance Optimization Process

When optimize performance of an application, it is normal to start with the test and find the weakest part, and after optimize this part, test again and find the weakest part of the new version. After several times of this iteration, the performance will reach the requirement of users. Web game optimization is quite similar to this, in this thesis, there will be performance testing, analysis the testing result, then tuning the code, the only differences are the performance metrics, since this thesis will mainly focus on the compute and render performance.

Performance Testing

Chrome DevTools will be used in this part, first the game will be run on the Chrome browser and Chrome developer panels will be open. A record will be taken on the timeline panel, which will show four basic groups: Loading, Scripting, Rendering and Painting. What's more, all the events happened are captured during the recording, it is easy for you to find out where your application is spendind the most time. Besides, you can also choose the memory mode to see the memory used by your application over time. So the records of running the flappy bird will be restored and provided to the step of performance analysis.

Performance Analysis

After having the result of performance testing, it's time to analysis the charts and the code, but while analysing the data, the possibility of optimization on each part should also be considered. So the part which can probably be optimized will be chosen after the step of analysing.

Optimization

While the most probably part is found, the most important thing to do is refactoring the code, during this period, there are too many things we need to take care of and they are mainly based on our web knowledge. After change some code, testing the code immediately.

The most important rule during this process is focus on one part until it meets the requirement or until it reach the best you can do.

3.2 Different Performance Metrics

As we know, different performance metrics contribute to the different optimization process. It is quite clearly that compute performance is much more important than the render performance because nowadays scripting takes more than 50 percent of the whole web display time. Based on the process of performance optimization, the time spend on the computing will be consider first which means a lot of effect will be taken focus on that part, then render performance will be considered.

Actually, different web games can lead to different optimization, but the fact that try best to cut down the render times and use less compute power is the same.

4 Results

This chapter presents the results. Note that the results are presented factually, striving for objectivity as far as possible. The results shall not be analyzed, discussed or evaluated. This is left for the discussion chapter.

In case the method chapter has been divided into subheadings such as pre-study, implementation and evaluation, the result chapter should have the same sub-headings. This gives a clear structure and makes the chapter easier to write.

In case results are presented from a process (e.g. an implementation process), the main decisions made during the process must be clearly presented and justified. Normally, alternative attempts, etc, have already been described in the theory chapter, making it possible to refer to it as part of the justification.

5 Discussion

This chapter contains the following sub-headings.

5.1 Results

Are there anything in the results that stand out and need be analyzed and commented on? How do the results relate to the material covered in the theory chapter? What does the theory imply about the meaning of the results? For example, what does it mean that a certain system got a certain numeric value in a usability evaluation; how good or bad is it? Is there something in the results that is unexpected based on the literature review, or is everything as one would theoretically expect?

5.2 Method

This is where the applied method is discussed and criticized. Taking a self-critical stance to the method used is an important part of the scientific approach.

A study is rarely perfect. There are almost always things one could have done differently if the study could be repeated or with extra resources. Go through the most important limitations with your method and discuss potential consequences for the results. Connect back to the method theory presented in the theory chapter. Refer explicitly to relevant sources.

The discussion shall also demonstrate an awareness of methodological concepts such as replicability, reliability, and validity. The concept of replicability has already been discussed in the Method chapter (3). Reliability is a term for whether one can expect to get the same results if a study is repeated with the same method. A study with a high degree of reliability has a large probability of leading to similar results if repeated. The concept of validity is, somewhat simplified, concerned with whether a performed measurement actually measures what one thinks is being measured. A study with a high degree of validity thus has a high level of credibility. A discussion of these concepts must be transferred to the actual context of the study.

The method discussion shall also contain a paragraph of source criticism. This is where the authors' point of view on the use and selection of sources is described.

In certain contexts it may be the case that the most relevant information for the study is not to be found in scientific literature but rather with individual software developers and open

source projects. It must then be clearly stated that efforts have been made to gain access to this information, e.g. by direct communication with developers and/or through discussion forums, etc. Efforts must also be made to indicate the lack of relevant research literature. The precise manner of such investigations must be clearly specified in a method section. The paragraph on source criticism must critically discuss these approaches.

Usually however, there are always relevant related research. If not about the actual research questions, there is certainly important information about the domain under study.

5.3 The work in a wider context

There must be a section discussing ethical and societal aspects related to the work. This is important for the authors to demonstrate a professional maturity and also for achieving the education goals. If the work, for some reason, completely lacks a connection to ethical or societal aspects this must be explicitly stated and justified in the section Delimitations in the introduction chapter.

In the discussion chapter, one must explicitly refer to sources relevant to the discussion.

6   Conclusion

This chapter contains a summarization of the purpose and the research questions. To what extent has the aim been achieved, and what are the answers to the research questions?

The consequences for the target audience (and possibly for researchers and practitioners) must also be described. There should be a section on future work where ideas for continued work are described. If the conclusion chapter contains such a section, the ideas described therein must be concrete and well thought through.

Bibliography

[1]   Evan Burchard. *The Web Game Developer's Cookbook: Using JavaScript and HTML5 to Develop Games*. Addison-Wesley, 2013.

[2]   Sanjeev Dhawan and Rakesh Kumar. "Analyzing Performance of Web-Based Metrics for Evaluating Reliability and Maintainability of Hypermedia Applications". In: *3rd International Conference on Broadband Communications, Informatics and Biomedical Applications, BroadCom 2008, November 23, 2008 - November 26* (2008), pp. 376–383. DOI: `10.1109/BROADCOM.2008.73`.

[3]   *Html5 Feature Performance*. Dec. 2015. URL: `http://www.html5rocks.com/en/features/performance`.

[4]   Mike Jones Jay Bryant. *Pro HTML5 Performance*. Apress, 2012.

[5]   Zhu Kunhua, Fu Junhui, and Li Yancui. "Research the performance testing and performance improvement strategy in web application". In: *Education Technology and Computer (ICETC), 2010 2nd International Conference on* 2 (2010), pp. 328–332. DOI: `10.1109/ICETC.2010.5529374`.

[6]   Addy Osmani. *Writing Fast, Memory-Efficient JavaScript*. Dec. 2015. URL: `http://www.smashingmagazine.com/2012/11/writing-fast-memory-efficient-javascript/`.

[7]   Mark Crovella Paul Barford. "Measuring Web performance in the wide area". In: *ACM SIGMETRICS Performance Evaluation Review* 27 (1999).

[8]   Vern Paxson. "Towards a Framework for Defining Internet Performance Metrics". In: *Proceedings of INET '96, Montrea* (June 1996).

[9]   *Performance profiling with the Timeline*. Dec. 2015. URL: `https://developer.chrome.com/devtools/docs/timeline`.

[10]   *Profiling JavaScript Performance*. Dec. 2015. URL: `https://developer.chrome.com/devtools/docs/cpu-profiling`.

[11]   Hidam Kumarjit Singh. "Performance Metrics of a Customized Web Application Developed for Monitoring Sensor Data". In: (2015), pp. 157–162.

[12]   Agile Support Team. *Web applications Performance Testing Metrics*. 2013. URL: `http://www.agileload.com/agileload/blog/2013/02/18/web-applications-performance-testing-metrics`.

[13] K. Velmurugan and M.a. Maluk Mohamed. "An Empirical Performance Metrics Measurement and Analysis of Software Platforms for Implementation of Web Services". In: *2009 Eighth IEEE/ACIS International Conference on Computer and Information Science* (2009), pp. 914–920. DOI: 10.1109/ICIS.2009.194.

På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© Ping Liu