Case Study of a Mobile HTML5 Multiplayer Game Portal

Leonardo V. de Carvalho Átila V. M. Moreira Ícaro V. M. Moreira
Marco T.C.F. Albuquerque Vicente V. Filho Geber L. Ramalho

Universidade Federal de Pernambuco, Centro de Informática, Brazil

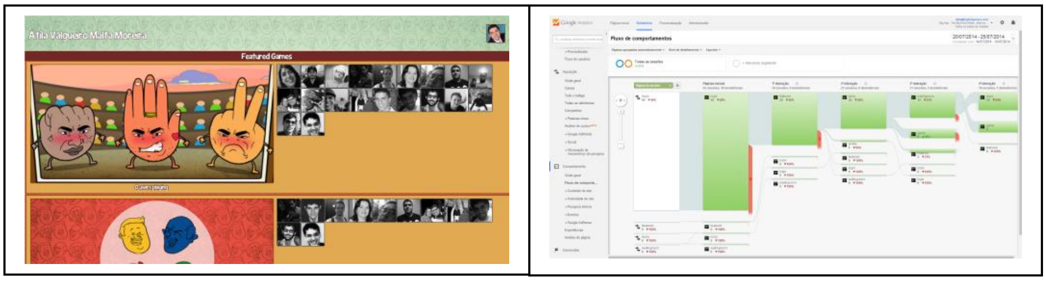


Figure 1: Portal page with game list on the left and analytics report on the right.

Abstract

This paper studies the design and development process of a mobile HTML5 game portal with focus on real-time competitive multiplayer games and a gamified environment to incentive competitiveness between players even outside the games. This project involved several complex elements, among which are the need for the portal to be compatible with most major mobile and desktop browsers and game optimizations which were required so that the games could be played with a good performance on mobile devices. Another important element is an accessible match-making service which enables developers to easily integrate their multiplayer games with the portal, relieving them from the need to create logic relative to the creation and management of matches, as well as victory and defeat logic. In order to conduct the case study, a total of four games with varying genres were developed and integrated with the match-making services, while several metrics regarding user behavior were collected. The analysis of these metrics allowed for a better understanding of how players engage with the portal and how to improve it..

Keywords: Development process and tools, Mobile computing

Authors' contact:
`{lvc,avmm,ivmm,mtcfa,vvf,glr}@cin.ufpe.br`

1. Introduction

The development of the portal detailed on this paper - called BigHub, which can be accessed on http://bighub me as of the writing of this paper - was motivated by two main factors: the stagnation of the major App Stores, on which the major players are already well established and have not changed over a long period of time [ThinkingGame 2014] and the continuous growth of the HTML5 game portals [Chris 2014;TreSensa 2014], which offer an alternative route for game developers to have their games seen and played.

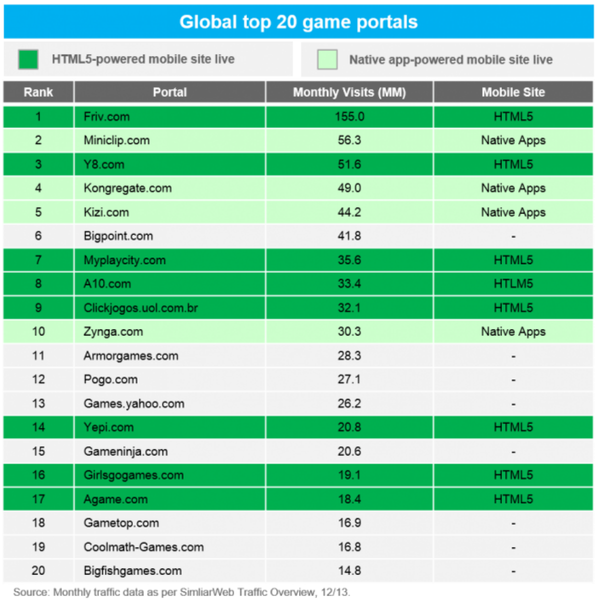
Many of the existing developers that try to publish their games on the existing app stores are suffering with this stagnation [Lototska 2014]: since the big players are settled and their products have a higher LTV(Lifetime Value) they are doing UA(User Acquisition) with high values than the LTV from small developers, which makes competition against such companies a considerably hard endeavor. Furthermore, the Apple Promotion System is unable to compete with many of the existing Ad Networks that the big players are using to promote their games [Dotson 2013].

Mobile Web is an interesting alternative to circumvent the UA costs, since it fits into the lower web advertising costs instead of app advertising costs [Grossberg 2014]. Moreover, mobile portals are relevant to the game industry and usually have more growth potential than a single game, specially since they are comprised of a usually considerable catalog of games. Table 1 shows the top 20 game portals - and the large amount of users that they have - and it's important to note that more than a half of those are mobile compatible.

HTML5 is a mobile web compatible technology that allows developers to deliver rich content to users through the mobile web, in other words, the users doesn't need to wait for the download of a big app in order to start using it, they only need to open it through their mobile browsers.

This paper presents the development of a social mobile gamified portal for games. The social aspect is defined as a network of social interactions and personal relationships [Oxford Dictionaries 2014], which for the scope of the portal means that players will be able to challenge their friends and play against them. The main subjects of this development process are described in the following 6 sections: Problems and Proposed

Solutions, which covers the technical and market issues and the team's approach to solve them; Development Methodology, that covers how the development team approached the project development; Software Architecture, infrastructure adopted in the development and deployment processes; Analysis, a general analysis of the final product; and Conclusion and Future Work.



Global top 20 game portals

🟩 HTML5-powered mobile site live

□ Native app-powered mobile site live

Rank	Portal	Monthly Visits (MM)	Mobile Site
1	Friv.com	155.0	HTML5
2	Miniclip.com	56.3	Native Apps
3	Y8.com	51.6	HTML5
4	Kongregate.com	49.0	Native Apps
5	Kizi.com	44.2	Native Apps
6	Bigpoint.com	41.8	-
7	Myplaycity.com	35.6	HTML5
8	A10.com	33.4	HTLM5
9	Clickjogos.uol.com.br	32.1	HTML5
10	Zynga.com	30.3	Native Apps
11	Armorgames.com	28.3	-
12	Pogo.com	27.1	-
13	Games.yahoo.com	26.2	-
14	Yepi.com	20.8	HTML5
15	Gameninja.com	20.6	-
16	Girlsgogames.com	19.1	HTML5
17	Agame.com	18.4	HTML5
18	Gametop.com	16.9	-
19	Coolmath-Games.com	16.8	-
20	Bigfishgames.com	14.8	-

Source: Monthly traffic data as per SimliarWeb Traffic Overview, 12/13.

Table 1: Global top 20 game portals[Grossberg 2014]

2. Problems and Proposed Solutions

According to Moreira et al.[2013], "The acquisition stage deals with how developers can reach out to users, and acquire players. It deals with features that make game social and viral. The retention stage concerns how to keep players around once they have been acquired. Specifically, it deals with features that make game sticky, or addictive, and is closely related to the gaming mechanics and dynamics presented in gamification. The final stage, monetization, looks at the features used in mobile games to generate revenue from their users. The features should provide incentives for players to pay for the virtual goods.", this project has as its main goals the acquisition and retention of the ARM funnel. The acquisition problem was mitigated as explained on the introduction, due to the reduction of the UA costs on the mobile web when compared to the costs for native apps.

The retention problem, on the other hand, demands changes on the project. The authors believe that retention on games must be treated as a cycle with 3 steps, Figure 2 illustrates this cycle: Action, Reward and Expansion. On the Action step, the user makes micro actions such as playing a game session, or visiting friends within the game. Every action results in a Reward, and after making Actions and obtaining enough Rewards, the player should make a major progress, which is called an Expansion. On the portal, the act of playing a match against another player is considered the Action - after every match, the player's avatar gains experience and after enough experience is earned, a level will be gained, with the possibility of also being awarded with a higher rank.

Social interaction is another tool to improve player engagement. With this purpose in mind, functions from the social network Facebook were used, specifically the like and share social actions which are shown to the player after a match is won. Since all the games on the portal are real-time multiplayer, this is an effective way to improve engagement and virality.

Besides the previously designed solutions aimed at dealing with some of the aspects of the ARM funnel, there are also some technological constraints associated with mobile web portals. Some of these identified issues were:

1. The need to run well on the majority of the mobile devices on the market, regardless of aspect ratio or resolution
2. The whole portal needs to be lightweight, so that it can be downloaded quickly over the mobile operator networks, which commonly do not offer very reliable services
3. Commonly a considerable delay is introduced on the messages exchanged between the game client and game servers, caused by the unreliability of the mobile operator networks

Aspect Ratio is defined as the geometry ratio between width and height. On mobile apps, when two devices with same aspect ratio have different sizes, the developer normally can adapt the application to both of them by using a relatively simple scale operation. However this problem is not so trivially solved when multiple aspects are involved. The team involved on this project has considerable previous experience with game development and had to deal with this problem on previous situations, and the solution was to use an approach similar to what is done on scaleform [scaleform 2014].

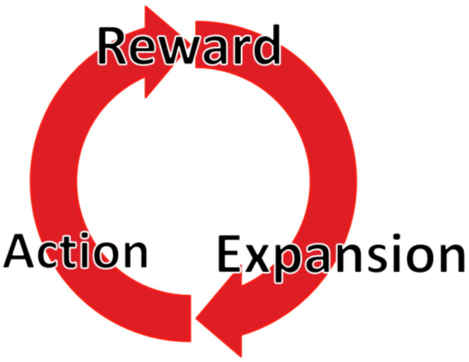


Figure 2: Cycle of retention in games.

Data optimization is a non-functional requirement on this type of portal. The team couldn't apply a definitive solution for this issue, but it was decided that frequent reduction and optimization of the images should be a common practice.

Delay is not the kind of issue that the team is able to solve, since it is a problem in the user network infrastructure, the solution was design all games so that they don't require an accurate synchronization between the players that are competing in a match.

3. Tools

On this section, an analysis regarding the tools and technologies used during the development process is made. This analysis is divided into:

1. Prototyping - tools used for prototyping and validation of the project
2. Management - tools used for team management and version control
3. Frontend - technologies and tools used in the implementation of the client-side logic, which includes adaptable interfaces and metrics tracking
4. Backend - technologies and tools used on the server-side logic, including communication with the client and a lobby system so that players can engage in matches
5. Games - technologies and tools used for creating multiplayer games

3.1 Prototyping

Before starting any step of the project development, all the project pages were prototyped using POP[Woomoo 2014], this step was essential to align the vision of the product with the entire team and also to narrow the scope, while consequently improving the project estimation and planning. The prototype is composed by the following pages: login, main, lobby, create room, game, social action, win, lose, draw and profile. Each one of these pages with transition flows implemented.

3.2 Management

As it will be see on the Development Methodology section, this project was developed in a distributed fashion with a very narrow development time. Given this situation, it was necessary to employ an agile and effective development cycle, as well as a straightforward task-control and version-control solutions.

For the team management, Trello [Trello 2014] was used. Trello allows the creation of entities named cards, these cards can be placed on several categories, such as tasks that have to be done, tasks that are being worked on or complete tasks. This made the creation of tasks and visualization of the project progress more accessible. All the tasks were planned by using a variation of Scrum, and the output of this planning were various task cards on Trello, each with their own priority level.

The version control technology that was chosen for this project was git [Git 2014], and for the repository hosting the Bitbucket [Atlassian 2014] service was used. The choice for git over other version control options was made because of how well it fits with a distributed development flow alongside an agile methodology [Spandel 2014]: the creation of branches allowed the team to work on several different features while not affecting existing features, and the product deployment also became a process straightforward enough that it could be made on a frequent basis.

These technologies in combination allowed a very agile and dynamic development environment, which resulted in quicker production and deployment of features while also enabling the team to respond quickly to issues reported in the portal.

3.3 Frontend

The team had no previous experience with frontend development, so it was needed to research, test and change technologies and tools with some frequency during the development process. In the first iteration, the team opted to use Phone.JS[DevExpress 2014] as the HTML5 framework and Angular.JS[AngularJs 2014] as the MVVM(Model-View-View-Model) framework. HTML5 frameworks help in the development of interface flows, layout and customization of the page. On the other hand, a MVVM framework helps in the development of a dynamic page with responsive ui and flexible changes, through dynamic binding and data update.

Phone.JS implements a native device interface for each mobile platform - namely iOS, Android, Blackberry and Windows Phone - and it was very promising at first glance, and Angular JS seemed like a good option given that it is maintained by the Google Team. However, as soon as the front-end developer noticed that the documentation and community for Knockout.JS[Knockoutjs 2014] with Phone.JS is more thorough and active than Angular.JS, so the change to Knockout was proposed to the team and it was accepted.

The team developed the first version of the portal with most features and with the integration of third-party games to validate it. But after some performance tests, it was noted that the average time to load the initial page over a mobile network operator was of 1.32 minutes, making the product unviable because the majority of users would drop out of the portal before it loaded. So the team made a simple performance test with other HTML5 frameworks and got several loading times, the best one being presented by jQueryMobile[jQuery 2014], which takes an average of

0.2 minutes to load a page, almost 5 times faster than Phone.JS. Moreover, the team agreed that would not be possible to create a gamified environment with the unstandardized native interface, which is associated with Phone.JS, so the HTML5 framework was changed to jQueryMobile.

After developing a page with jQueryMobile by using the default interface elements provided by the framework, the team needed to override most of the native components of jQueryMobile in order to achieve the desired look and feel for the portal. Figure 3 illustrates the design evolution through these tools changes and project milestones.

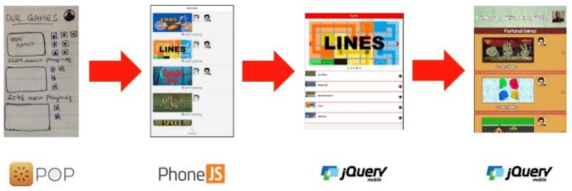


Figure 3: Interface Evolution during development.

Also, the frontend was responsible for tracking analytics data. The team had extensive experience with Google Analytics[Analytics 2014] on previous projects, so learning to use the Javascript version was the only challenge, but all the concepts and main metrics that could be extracted in order to understand the user behaviors were already known.

3.4 Backend

One of the major problems present in server infrastructures is scalability. In order to maintain the portal scalable, a cloud computing service was used, this way the computing power needed by the portal will grow alongside the project costs. The development team had previous experience with Google App Engine[AppEngine 2014], so it was the platform chosen for this project.

Regarding social integration, restFB[restFB 2014] was integrated on the backend to enable the use of viral actions. Since viral actions require several calls to the Facebook service, it would require too much bandwidth from the client to leave these calls on the client side, so this integration on the server made it possible to save a considerable amount of bandwidth usage on the client.

All communication between client and server were made through Jersey API[Oracle 2014], a RESTful API which makes all the servlet generation process transparent to the developer, thus allowing the developer to focus on the development of features.

The creation of game rooms was made by using the ChannelAPI[ChannelAPI 2014], a google app engine feature which makes it possible to open a socket which can be listened to by several users. The following protocol was used to match online users:

1. Player 1 creates a room in the lobby page: An associated channel is created and the user keeps listening to any event sent by the server.
2. Player 2 opens the lobby: A list of all channel waiting for challengers is shown.
3. Player 2 open Player 1's room: Player 2 joins Player 1's Channel and sends a message notifying that he is ready for the match.

After this process ends, the game between the two players is started.

3.5 Games

The nature of the games developed for the portal - easy to understand games with mechanics that allowed for quick sessions - led to the choice of lightweight technologies that offered a minimum engine footprint, both in terms of download size and performance. Initially Cocos2d-JS [Cocos2dJs 2014] was chosen as the game engine, but even though it's a very powerful and flexible engine, the development team chose the Phaser Engine [PhotonStorm 2014] - a lightweight HTML5 game engine with a quick and simple development pipeline. This choice was based mainly on the fact that its pipeline is very similar to that of the Flixel Engine [Flixel 2014], with which the team had previous experience.

Given that all the games had a considerable focus on the multiplayer competition, it was necessary to use a communication technology that worked well both on desktop computer browsers and most major mobile browsers [Khalaf 2014], while also offering minimum connection overhead. While initially the team chose Photon [ExitGames 2014] for the networking logic because of how well the middleware is established on the market, the service presented a lack of documentation for it's Javascript variant as of the writing of this paper. The team instead opted to use node.js [Joyent 2014], as it offered an efficient websocket[Fette 2011] connection through it's socket.io implementation. The game servers were hosted on the Heroku [Heroku 2014] web hosting service, which offers a quick and accessible pipeline from development to deployment, with an easily scalable service.

The communication protocol between the client on the browser and the server hosted on heroku is made through the establishment of a websocket connection between these two. Given the multiplayer nature of the games, the server stores several matches which are comprised of pairs of client-server connections. Once both connections of a match are severed, it is removed from the server.

Websocket connections allow for real-time communication between client and server, and the communication protocol that was used is comprised of message exchanges and responses: the client sends messages to the server with actions and an associated data package that can contain any information relevant to the game - such as player input or action - and the server processes how the client message affects the match. After the processing operation is done on the server, it sends a message back to the client, which will then interpret that message according to the game logic, causing the game to resume and allowing it to once more send a message to the server.

4. Development Methodology

This project used as methodology a set of good practices from SCRUM[Willian 2014] adapted for the team's limitations - namely a remote distributed workflow with a very narrow development time. All stakeholders brainstormed alternatives of development that would fit on their limitation/development styles and vision of process, the following production process was reached as a conclusion:

- Meeting/Planning on Wednesdays: the team always met personally on wednesdays to talk about what they had done and discuss about the next sprint.
- One-day Sprint Duration: Everyone in the team needs to do all the tasks assigned to them, integrate with the repository project and deploy a stable and tested version at the end of the day. Each member of the team is free to decide the time of the day that they will work and how they will work, but Saturday is the day that was chosen for the work to be done.
- Outsourcing: To improve the team focus in the software development and reduce the project management cost, all tasks non-related directly to programming were outsourced, its including art, game design and production.

The team also decided to use Trello[Trello 2014] as a project management tool, mainly because of its simplicity. Trello is commonly used for small startups during MVP and initial stage of development, this is exactly the scenario of this development team, so this tool fits very well with its demands. Figure 4 illustrates the development methodology.

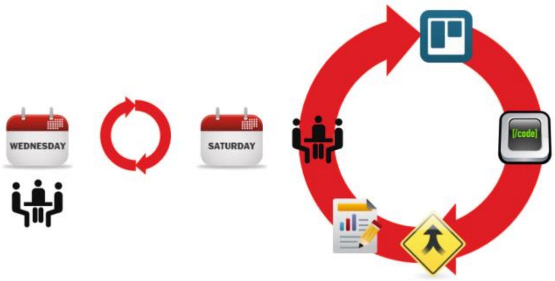


Figure 4: Diagram of the portal development process.

5. Software Architecture

The team adopted the standard three-tier architecture because all members have real project experience with it, also its simplicity and low coupling rate the were decisive in the choosing process. This architecture segments the project structure into three layers(Figure 5):

- Presentation Layer: Provides the application's user interface and the communication with the backend.
- Business Logic Layer: Implements all the business functionalities.
- Data Access Layer: Responsible for the datastore connection and all functionalities associated to it.

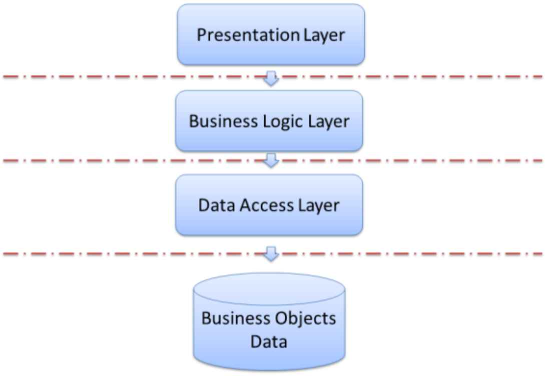


Figure 5: Diagram of Three-tier architecture.

It's important to highlight that the team has tested this same architecture with a previous project for dynamic adjustment of virtual goods prices on a game that reached a peak of 40,000 concurrent users and it didn't show any performance issues, so this architecture is simple and scalable enough for this project's scope and size.

6. Analysis

The goal of this section is to provide an overview of the portal, as well as an analysis of its usage metrics to evaluate it and identify possible problems on its usability.

6.1 The Portal

As of the time of writing, the portal can be accessed on the url http://bighub me and contains a specific set of functions wich is subject to change with future updates and improvements. On its current state, the portal presents a total of four games of several genres:

1. Jo-ken-po: an implementation of the classic Rock-Paper-Scissor game [Arneson 2013], on a best out of three matches dispute.

2. Simon: an implementation of the classic Simon game [Hasbro 2014]. The first player must choose a color to start the sequence and the next player has to repeat the sequence, if the sequence is correct the player can add another color to the sequence. The match continues until a player makes a mistake, making the other player the winner.
3. Tap Racing: a simulated racing game on which the player accelerates the car by tapping on the mobile device screen - or clicking on the mouse if playing on a desktop computer. Both players have 5 seconds to tap or click as much as possible, and once the time is over the player with more taps is the winner.
4. Pong Squash: a combination of the classic Pong game [Pong 2014] with the squash sport [Squash 2014]. The players play on separated rooms, and each one has a paddle and a ball. The goal is to bounce the ball against the wall opposite to the paddle as many times as possible without letting the ball get behind the paddle. The player that makes the ball bounce the most against the wall is the winner.

It's relevant to note that the original concept of some games for the portal, Pong Squash for instance, involved real-time mechanics. However, upon testing some of these mechanics over mobile networks, the games became unplayable due to the high latency and instability of the connection. The solution found was to define the constraint that all games in the portal should use asynchronous mechanics. With this constraint in place, issues with latency and slow connection no longer became relevant or affected the player experience.

The flow of use of the portal is a very straightforward one: after opening the page, the user can login by using a Facebook account. Once logged in, the user sees the list of featured games, which also contains the information of how many players are currently playing each game and which games are being played by the Facebook friends, as it can be seen on Figure 6.

On this screen, the user can click/tap on its own portrait to see its profile on the portal, or it can click/tap on any of the game images to enter a lobby. Once in the lobby, it's possible to see a list containing matches created by other players and join any one of them or to create a new match, which in turn can be joined by another player. After joining a match, the game will start and the player will play against its rival. Once the game is over, the player sees a screen informing of its defeat,victory or draw, and in the case of victory the player will get experience on the portal and a popup is shown which invites the player to boast its achievement by posting on Facebook.

After all these interactions, the player is redirected to the featured games page once again, where it can play other games.

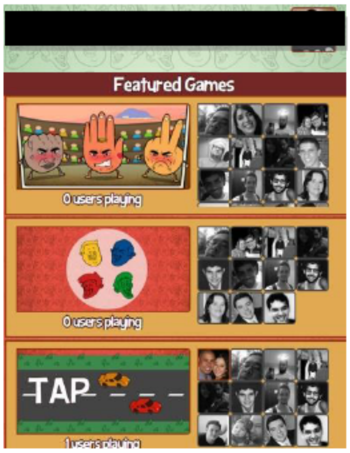

[redacted]

Figure 6: On the Featured Games page, the player can see a list of the games available.

6.2 Usage Analysis

In order to make the analysis, data was collected by using the Google Analytics service, but for the purpose of this study the analysis time frame was limited to a 30-day period starting on June 25 2014. Over this period of time a total of 445 sessions were registered, 65% of those were from smartphones, 30% from desktop computers and 5% from tablets. The high amount of smartphones is to be expected given that the website was designed mainly with a mobile focus in mind, but the high amount of desktop users when compared to the amount of tablet users is an unexpected development. The amount of tablet users itself is a surprising data, given that mobile devices in generals - smartphones and tablets - where the focus of the portal. Further investigation of this information must be made in order to better understand why the amount of tablet users is so small when compared to the other platforms.

The average duration of a session on the website is of 2 minutes and 38 seconds, which is a good amount considering the amount of content available. There is still room for improvement on this however, but this will be discussed in further detail on the Conclusion and Future Work section.

The portal also presents a bounce rate [Gabe 2007] of 42%, which is considered an average value when compared to industry benchmarks[Purushothaman 2010], but it's very close to be considered an above-average value, so this value is expected to improve with some minor tweaks to the portal in general.

Another important analysis is that of the usage funnels, which will provide some insight on where in the portal utilization the users are leaving the portal without completing some operations. Starting from a sample of 400 users that visited the website and saw the login screen for the first time, only 39.25% of those logged-in and saw the game list. This is a very alarming information and it shows that the login is a considerable attrition point in the portal interaction. In order to improve this it would be advised to provide initial access to the players without the need to login and offer additional functions or advantages to users after they log in using Facebook.

Of the remaining 157 users that logged-in and reached the Featured Games list, 71% decided to enter the game lobby screen. This is a reasonable amount, but still a considerable amount of users did not bother to enter a match even after they went through the login flow. Some causes for this might be the lack of games that draw the users' attention or the fact that no users were playing any games on the moment that those 29% of users that did not enter any game saw the list.

Once the 112 users of the sample entered the lobby, 77.7% of them went to the waiting room, where the players wait for someone to join their games, while 10% joined a game that was already available in the lobby. This shows that a considerable amount of the players are willing to play the games once they have found a title that draws their attention. The main issue that was noticed however is that 76% of the players who wait in the lobby for someone to join their game give up after some time and go back to the featured game screen, while only 18.4% of said players have someone join the game.

Upon an analysis of the full player flow, it's possible to notice that there is a considerable funnel from the moment that a user enters the site for the first time to the moment that it enters a game. From the 400 users that enter the portal, only 27 - or 6.75% - get to play a game session within four interactions or less. Changes to improve this metric will be further discussed in the Conclusion and Future Work section, but there are still other steps that need to be considerably improved, such as the login flow which is the cause for most of the users who leave the portal.

7. Conclusion and Future Work

In this work, the development of a high risk project with a distributed team that achieved a successful result was presented. This case study offers a detailed insight into the development process a HTML5 Mobile Game Portal and aims to help other development teams that plan to work on a similar software projects.

The portal underwent a soft launch [Katkoff 2012] period of one month, during which it was possible to obtain key metrics of how it's being used before the actual release. As the analysis has shown, there are still some key issues that need to be worked on, such as the amount of time that the users are spending on the portal and the high attrition rate of the login screen. In order to solve the first issue, more games are most likely needed, preferably of different genres. As for the second one, the most effective solution would most likely be to remove the mandatory login, while at the same time rewarding players that decide to log in.

Another very relevant issue is the low amount of players that effectively get to play a match, and to mitigate this issue, a bot system will be developed. So if a user waits more than a given amount of time in the lobby and no human player joins the game, a bot player will automatically join the game and play against the player. This change is expected to be very effective, as it will most likely greatly increase the amount of players who join a lobby and get to play a game.

In order to improve the player engagement, new features will be implemented, namely leaderboards and achievements. It's expected that the implementation of these features and their integration in the games will make the gamified features clearer for the players and increase their effectiveness.

One final change that is being planned is in the portal monetization. Currently there are no monetization elements present, but there are plans to add monetization through advertisements. This will require a considerable amount of work, given the need to provide a system for the advertisers to manage their ad campaigns and for the campaigns to be properly shown on the portal.

Acknowledgements

The authors would like to thank our sponsors Big Hut Games, Manifesto Games Studio and Capes, also Professor Silvio Meira, who was responsible for the Software Engineering classes that made this project possible.

References

THINKGAMING, 2014, Think Gaming [online], Available from http://thinkgaming.com/app-sales-data/[Accessed 26 July 2014].

OXFORD DICTIONARIES, 2014, Oxford Dictionaries [online], Available from http://www.oxforddictionaries.com/definition/english/social-network [Accessed 2 September 2014].

CHRIS, 2014, 1st Quarter 2014 Developer Sales Stats [online], fgl – the game industry's marketplace, Available from: https://www.fgl.com/blog/2014/04/1st-quarter-2014-developer-sales-stats/ [Accessed 26 July 2014].

TRESENSA, 2014, HTML5 AS ANOTHER AVENUE FOR INDIES [online], Game Sauce, Available from: http://startupbeat.com/2014/03/11/tresensa-qa-id3442/ [Accessed 26 July 2014].

LOTOTSKA, M., 2014, Forget apps—with a fresh $2M in Series A funding, TreSensa is betting big on mobile web games [online], Startup beat, Available from: http://gamesauce.org/news/2014/06/27/html5-as-another-avenue-for-indies/ [Accessed 26 July 2014].

Dotson, C., 2013, Stateside: Apple is powerless to thaw the App Store's big freeze [online], Pocket Gamer, Available from:http://www.pocketgamer.biz/stateside/54193/stateside-apple-is-powerless-to-thaw-the-app-stores-big-freeze/[Accessed 26 July 2014].

Grossberg, R., 2014. The Mobile Web Rises [online]. *Gamasutra,* Available from: http://www.gamasutra.com/blogs/RobGrossberg/20140205/210128/The_Mobile_Web_Rises.php[Accessed 26 July 2014].

Moreira, A.V.M., Filho V.V.,Barros, G.A.B., Carvalho, L.V., Albuquerque, M.T.C.F. and Ramalho, G.L., 2013. Which features matter to make a successful mobile game?. *SBC - Proceedings of SBGames,* 193-202.

Scaleform, 2014. Scaleform – Autodesk. Available from: http://gameware.autodesk.com/scaleform [Accessed 26 July 2014].

Woomooo, 2014. POP – Prototype On Paper. Available from: https://popapp.in/ [Accessed 26 July 2014].

Git, 2014. Git –fast-version-control. Available from: http://git-scm.com/ [Accessed 26 July 2014].

Atlassian, 2014, Atlassian - Bitbucket. Available from: https://bitbucket.org/ [Accessed 26 July 2014].

Spandel, D. and Kjellgren, J., 2014. Choosing between Git and Subversion - How does the choice affect software developers?. *Bachelor Thesis in Software Engineering*, Bleking Tekniska Högskola. Available from: http://www.bth.se/fou/cuppsats.nsf/all/dd7198cc1f202d45c1257d00004243fd/$file/BTH2014Spandel.pdf [Accessed 26 July 2014].

DevExpress, 2014. PhoneJs. Available from: http://js.devexpress.com/ [Accessed 26 July 2014].

AngularJs, 2014. AngularJs by Google. Available from: https://angularjs.org/ [Accessed 26 July 2014].

Knockoutjs, 2014. KnockoutJs. Available from: http://knockoutjs.com/ [Accessed 26 July 2014].

jQuery, 2014. jQuery Mobile. Available from: http://jquerymobile.com/ [Accessed 26 July 2014].

Analytics, G., 2014. Google Analytics. Available from: http://www.google.com/analytics/ [Accessed 26 July 2014].

AppEngine, G., 2014. Google AppEngine. Available from: https://appengine.google.com/ [Accessed 26 July 2014].

restFB, 2014. restFB. Available from: http://restfb.com/ [Accessed 26 July 2014].

Oracle, 2014. Jersey API. Available from: https://jersey.java net/ [Accessed 26 July 2014].

ChannelAPI, G., 2014. Channel Java API Overview. Available from: https://developers.google.com/appengine/docs/java/channel/ [Accessed 26 July 2014].

Cocos2dJs, 2014. Cocos2d Js. Available from: http://www.cocos2d-x.org/wiki/Cocos2d-JS [Accessed 26 July 2014].

PhotonStorm, 2014. Phaser.io . Available from: http://phaser.io/ [Accessed 26 July 2014].

Khalaf, S., 2014, Apps Solidify Leadership Six Years into the Mobile Revolution [online], Flurry Blog, Available from: http://www flurry.com/bid/109749/Apps-Solidify-Leadership-Six-Years-into-the-Mobile-Revolution#.U9RN9vldVCW [Accessed 26 July 2014].

ExitGames, 2014. Photon realtime. Available from: https://www.exitgames.com/en/Realtime [Accessed 26 July 2014].

Flixel, 2014. Flixel. Available from: http://flixel.org/ [Accessed 2 September 2014].

Joyent, 2014. nodeJs . Available from: http://nodejs.org/ [Accessed 26 July 2014].

Fette, I. and Melnikov, A., 2011. The WebSocket Protocol. *Internet Engineering Task Force,* 1-71.

Heroku, 2014. nodeJs . Available from: https://www heroku.com/ [Accessed 26 July 2014].

Williams, L., Brown, G., Meltzer, A. and Nagappan,N., 2011. Scrum+ engineering practices: Experiences of three microsoft teams. *Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on. IEEE, 2011. p. 463-471.*

Trello, 2014. Trello . Available from: https://trello.com/ [Accessed 26 July 2014].

Arneson, E., 2013, Rock Paper Scissors - The complete rules for the classic game Rock Paper Scissors [online], Available from: http://boardgames.about.com/od/partygame1/a/rps_rules.htm [Accessed 26 July 2014].

Hasbro, 2014, SIMON SWIPE GAME [online], AboutHome, Available from: http://www hasbro.com/en_US/shop/details.cfm?R=FA45E76C-5056-9047-F541-122FDDEF7288:en_US [Accessed 26 July 2014].

Pong, 2014, Pong Games [online], Available from: http://www.ponggame.org/ [Accessed 26 July 2014].

Squash, 2014, RULES OF THE GAME [online], Available from: http://www.squash.ca/en/rules-game/ [Accessed 26 July 2014].

Facebook, 2014, Facebook [online], Available from: https://www facebook.com/ [Accessed 26 July 2014].

Gabe, G., 2007, Bounce Rate and Exit Rate, What is the Difference and Why You Should Care [online], The Internet Marketing Dricer – Covering the latest in online marketing strategy, Available from: http://www hmtweb.com/blog/2007/08/bounce-rate-and-exit-rate-what-is html/[Accessed 26 July 2014].

PURUSHOTHAMAN, A., 2010, What Is A Good Analytics Bounce Rate? [online], TechWyse – Internet Marketing, Available from: http://www.techwyse.com/blog/website-analytics/improving-website-bounce-rates/[Accessed 26 July 2014].

KATKOFF, M., 2012, First month post-launch is crucial for monetization [online], Deconstructor of fun – because fun is serious business, Available from: http://www.deconstructoroffun.com/2012/07/first-month-post-launch-is-crucial-for html [Accessed 26 July 2014].

Big Hut Games. [Online]. Available: http://www.bighutgames.com[Accessed 26 July 2014].

Manifesto Games Studio. [Online]. Available:
http://www.manifestogames.com.br[Accessed 26 July 2014].

CAPES. [Online] Available: http://www.capes.gov.br/[Accessed 26 July 2014].