Business and IT alignment with the Value Map and SEAM: Implementation and First Solutions in the Context of a Start-up

Semester project
LAMS Laboratory

Sami Perrin
Gil Regev
Alain Wegmann
Blagovesta Kostova

EPFL
June 2017

Introduction Semester Project

Mobile gaming industry is at a very narrow intersection between engineering and art. Here we take a look at it through the prism and with the tools of online IT services. This allows us to start with a precise canvas and a set of tools and metrics to understand how to continuously improve a game in a startup environment.

Cheese Cat is a startup founded in 2016 by Sami Perrin and Guillaume Pedrazzini. The first game it launched is based on an innovative concept, redistributing with users a share of the ads revenues generated in an arcade game through cryptocurrencies. This semester project aims at helping CheeseCat to have a good understanding of its target market, start a seed round of funding and develops its internal organisation to be able to launch a game targeted at a wider audience in 2018.

CheeseCat's first and second products revenue streams are based on advertisement while its third product aims at testing the free-to-play game model in which the main functionalities of the game are free and some premium features are paid.

We begin by modeling its ecosystem and trying to understand the relations it has with its partners and adopters. From that point we try to understand what value is created and captured by which stakeholder. We propose three solution to improve the company organisation. One of which is the analytical solution, which is addressed in Part II. Finally in the Appendix, we discuss the report structure, we discuss on the main adopter of a service financed by advertising and we address the question of selecting an appropriate governance system.

Part I: CheeseCat Strategy

I.1 Introduction CheeseCat Strategy

To determine an appropriate strategy framework, CheeseCat used the strategy palette [1]. This framework aims at simplifying the selection of a strategic framework that match their environment.

It proposes a matrix across three dimensions, predictability, malleability, harshness. In mobile gaming the environment is not predictable which is evident in regard to last ten years or so during which almost one year over two a new player consistently managed to take a significant market share. And it is not malleable at the current scale of the company. The conditions are relatively harsh for a startup in that environment but that doesn't fit the definition that they propose which applies to an existing player that failed at adapting its way of doing business to maintain sustainability.

Thus the appropriate strategic quadrant for CheeseCat is *adaptative.* Its core idea is that a company in that segment should be fast as advantages are short lived. It should continuously experiment and identify new advantages rapidly.

In software development, a methodology is particularly appropriate to sustain a fast pace while experimenting, the lean methodology. [2]
The lean methodology aims at shortening the time required from startups to release a viable product. It was developed by Eric Ries in 2008. It takes its roots in adapting lean management methods to the fast paced universe of startups.
It is a scientific approach in the sense that one of its main principle is to progress through validated learning. That is to setup goals in term of measurable metrics, emit hypothesis on the effect that a future feature could have on a metric. Finally implementing the feature and tracking the results.

Doing so iteratively until its services can address the market effectively is the short term goal of CheeseCat.

Two more reasons suggests that this approach is rightly calibrated specifically in the mobile gaming segment.
The first one is the space of features that can be added to a game is extremely large. It ranges from adding social features to changing the game logic, adding more existing content such as levels, badges and so on, modifying some visual elements, defining or redefining the game story, etc… The lean methodology provides a framework to navigate this large decision space while converging to its goal.
The second one is that its resource is sparse. As it has a small team and limited funding it has to determine and implement in the shortest time span possible a minimal subset of the possible features to address the market effectively.

Thus CheeseCat plans an experimental phase which spans one year to one year and a half during which it periodically releases small games in which it tries each time a different subset of features. It analyses each features and rates them according to various metrics and their build time. At the end of this period, CheeseCat plans to build a bigger game in term of content, in which will be implemented the retained features.

This strategy will allow to invest more significantly in the promotion of the future game while having a good confidence that the game will be able to reach its targeted revenue.

This strategy should allow investors to invest in the marketing of the next game while having a good confidence that the game will reach its targeted revenue.

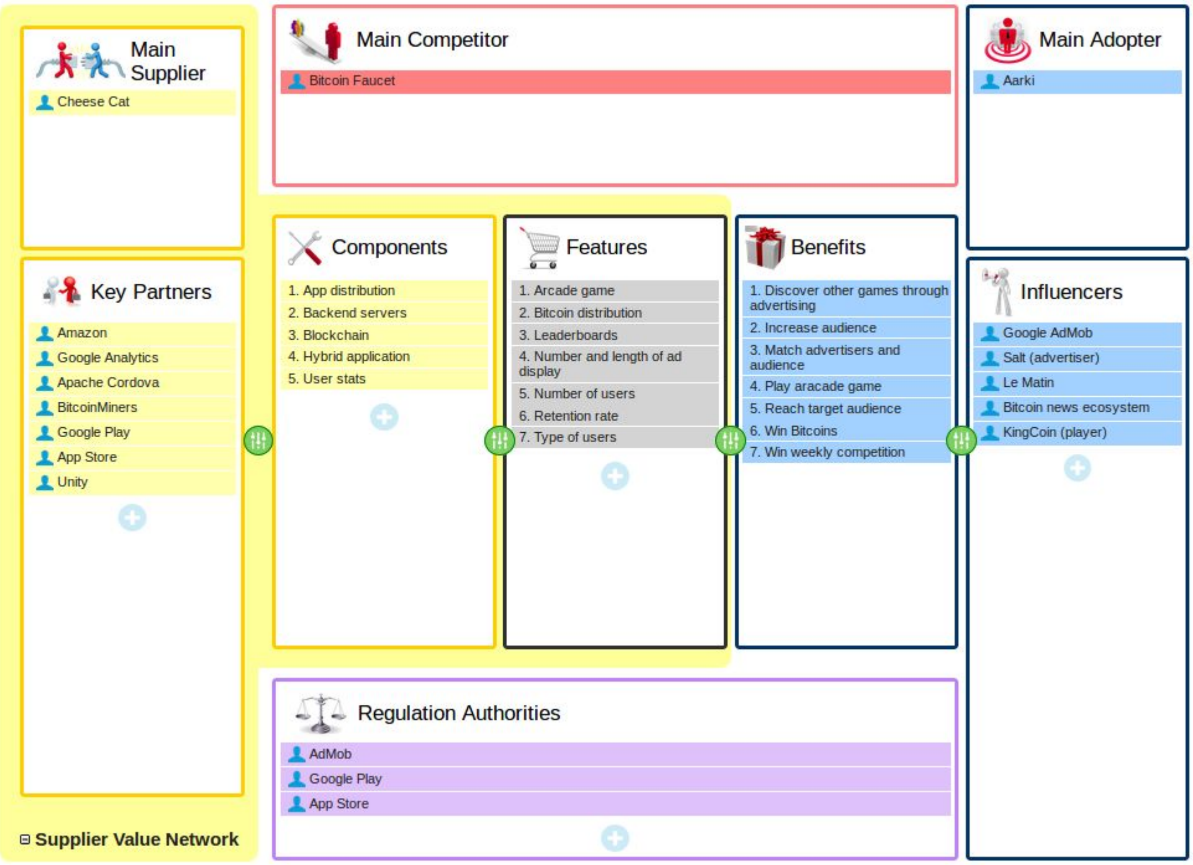
I.2 CheeseCat - Ecosystem Definition

In this section we use the a modeling framework, the *Value Map,* developed at the LAMS laboratory to explore the company’s ecosystem.[3] The revenue stream of CheeseCat games comes from two stakeholders. In the two first games, they come from the advertisers and in its third game from the players. It’s thus not evident to determine who is the main adopter of their products and on which relationship the company should dedicate its focus.

In this context the value map provides a framework to conceptualize and analyze the value creation and capture processes. As stated in the previous chapter the malleability of its environment is low, the models generated with the Value Map will allow us to formally sustain this claim but at the same time will allow us to determine which relationships are most valuable for CheeseCat and on which relations it can actually exert an influence.

Advertiser as Main Adopter

Here we model the ecosystem of the enterprise with the usual viewpoint with the adopter being the client, the advertiser. We define all the key partners that allow the service to exist. The service is internally defined in term of its components and externally in term of its feature.
All the models are produced with the help of the online tool TradeYourMind [4]. From the possibilities it offers we use here two of the main functionalities, the Service Model and the Alignment.



Main Supplier
- Cheese Cat

Main Competitor
- Bitcoin Faucet

Main Adopter
- Aarki

Key Partners
- Amazon
- Google Analytics
- Apache Cordova
- BitcoinMiners
- Google Play
- App Store
- Unity

Components
1. App distribution
2. Backend servers
3. Blockchain
4. Hybrid application
5. User stats

Features
1. Arcade game
2. Bitcoin distribution
3. Leaderboards
4. Number and length of ad display
5. Number of users
6. Retention rate
7. Type of users

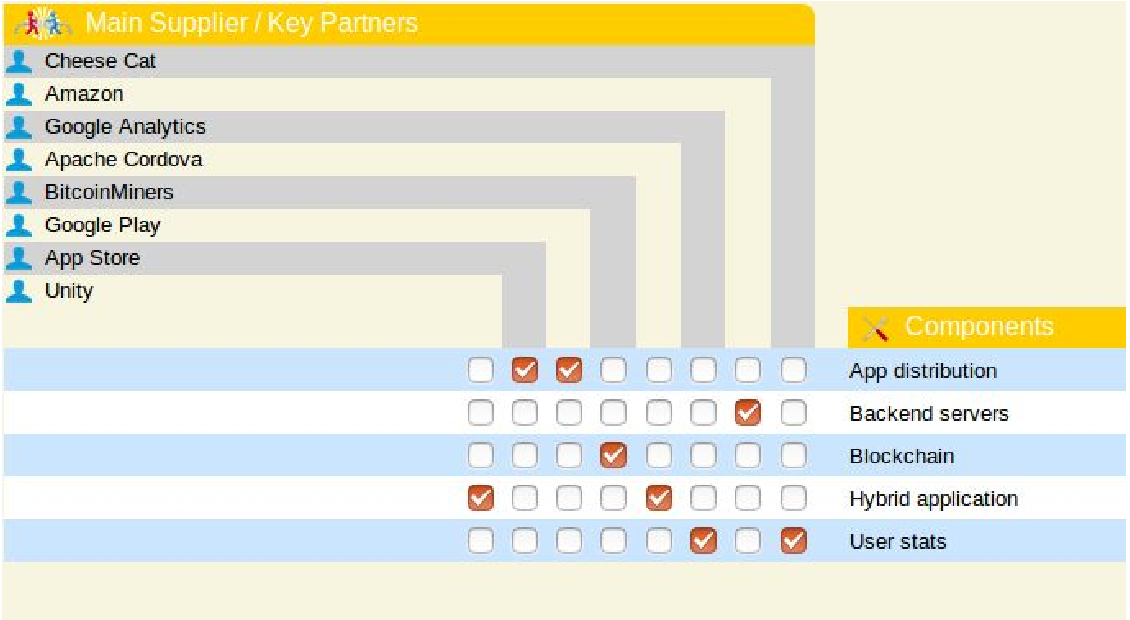
Benefits
1. Discover other games through advertising
2. Increase audience
3. Match advertisers and audience
4. Play aracade game
5. Reach target audience
6. Win Bitcoins
7. Win weekly competition

Influencers
- Google AdMob
- Salt (advertiser)
- Le Matin
- Bitcoin news ecosystem
- KingCoin (player)

Regulation Authorities
- AdMob
- Google Play
- App Store

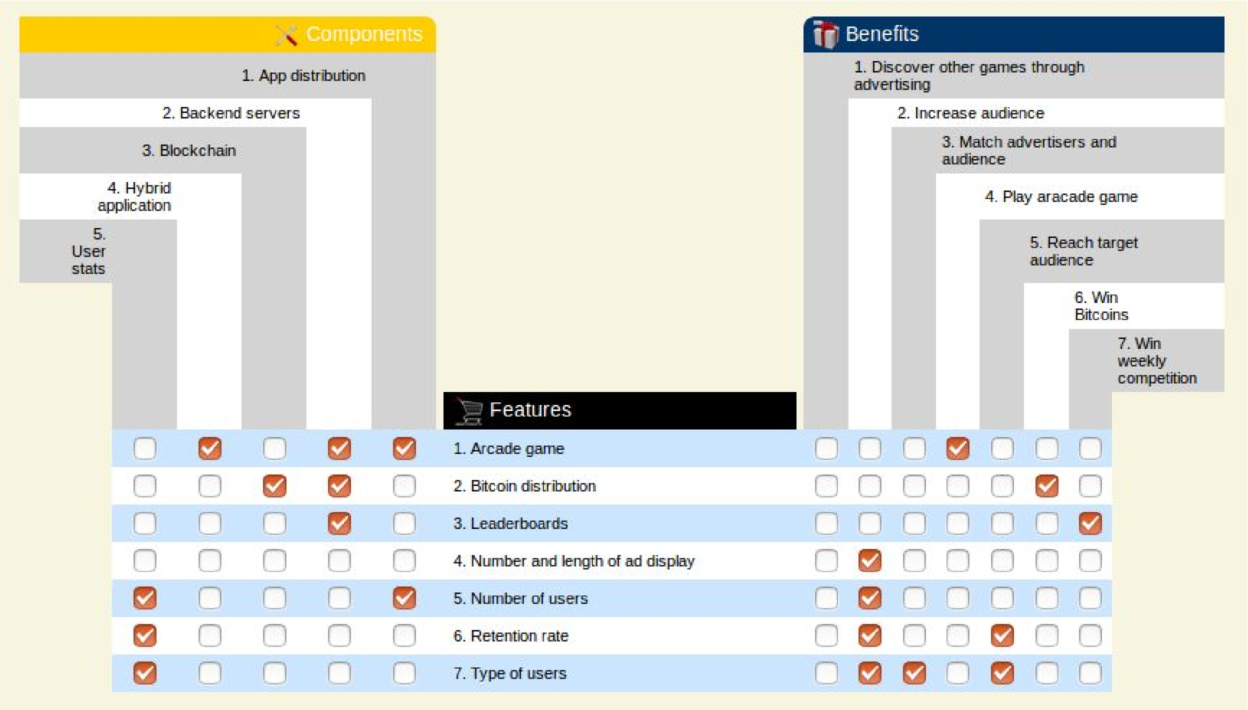
⊟ Supplier Value Network

The Service Model
With Aarki the advertisement company as the main adopter of CheeseCat’s services and KingCoin one of the player as the main influencer of Aarki



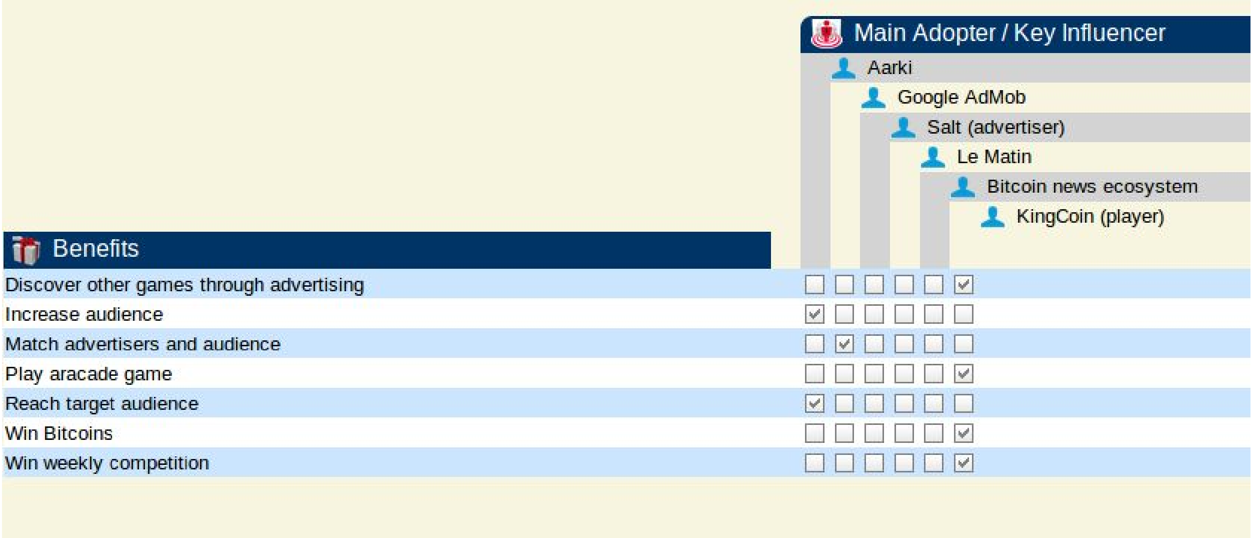
Unity	App Store	Google Play	BitcoinMiners	Apache Cordova	Google Analytics	Amazon	Cheese Cat	Components
☐	☑	☑	☐	☐	☐	☐	☐	App distribution
☐	☐	☐	☐	☐	☐	☑	☐	Backend servers
☐	☐	☐	☑	☐	☐	☐	☐	Blockchain
☑	☐	☐	☐	☑	☐	☐	☐	Hybrid application
☐	☐	☐	☐	☐	☑	☐	☑	User stats

The Value Map - Supplier-Partner / Components



5. User stats	4. Hybrid application	3. Blockchain	2. Backend servers	1. App distribution	Features	1. Discover other games through advertising	2. Increase audience	3. Match advertisers and audience	4. Play aracade game	5. Reach target audience	6. Win Bitcoins	7. Win weekly competition
☐	☑	☐	☑	☑	1. Arcade game	☐	☐	☐	☑	☐	☐	☐
☐	☐	☑	☑	☐	2. Bitcoin distribution	☐	☐	☐	☐	☐	☑	☐
☐	☐	☐	☑	☐	3. Leaderboards	☐	☐	☐	☐	☐	☐	☑
☐	☐	☐	☐	☐	4. Number and length of ad display	☐	☑	☐	☐	☐	☐	☐
☑	☐	☐	☐	☑	5. Number of users	☐	☑	☐	☐	☐	☐	☐
☑	☐	☐	☐	☐	6. Retention rate	☐	☑	☐	☐	☑	☐	☐
☑	☐	☐	☐	☐	7. Type of users	☐	☑	☑	☐	☑	☐	☐

Alignment - Components/Features/Benefits
We see that CheeseCat relies strongly on the backend servers and User analytics



Main Adopter / Key Influencer

Benefits	Aarki	Google AdMob	Salt (advertiser)	Le Matin	Bitcoin news ecosystem	KingCoin (player)
Discover other games through advertising	□	□	□	□	□	☑
Increase audience	☑	□	□	□	□	□
Match advertisers and audience	□	☑	□	□	□	□
Play aracade game	□	□	□	□	□	☑
Reach target audience	☑	□	□	□	□	□
Win Bitcoins	□	□	□	□	□	☑
Win weekly competition	□	□	□	□	□	☑

Alignment - Benefits/Adopters-Influencers

We see that the stakeholder that benefits the most from CheeseCat services is the player

Captured value creation

This perspective enables us to understand the monetary benefit that the service provider capture. Indeed, the advertisement model online is based on the number of users and their demographics. Having the gamer as an influencer of the advertiser thus makes sense in that context.

We see that our main adopter has mainly two benefits from the product usage, increasing its audience and reaching its target audience. From the perspective of CheeseCat, four features have an influence over those benefits. The number and length of ads display, the number of users, the retention rate of the users (if Aarki has a client that wants to maintain a given campaign over time on a specific set of users it would be possible if those users used the app at regular intervals), and the type of users.

Limitation of this model

The first limitation comes from the nature of the features that have an impact on Aarki's benefits. Out of the four features only one is directly at the reach of CheeseCat, the number and length of ads display.

The latter is to be manipulated with caution, if too much ads are displayed the user could feel overwhelmed.

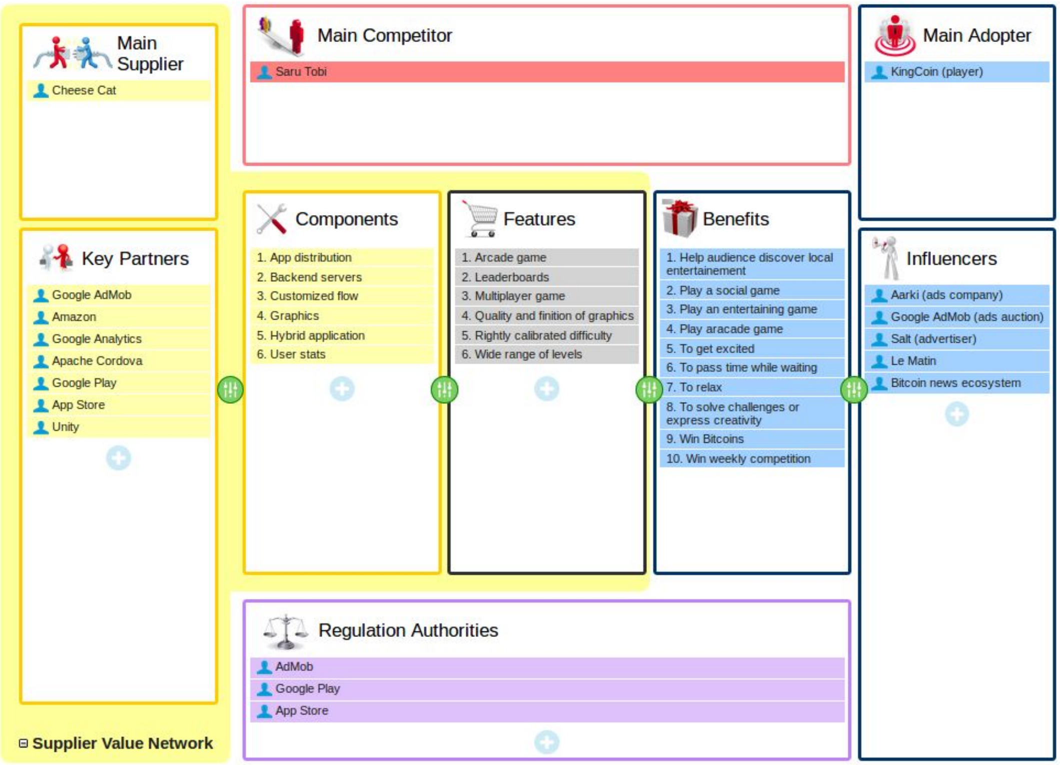
Over the type of users it has a very small influence; it can design games more oriented for a target audience but even in that case there is no guarantee that this audience would be the main one, and since any type of audience can be monetized it seems that the wider the audience the better and restricting the games to a niche might be detrimental to that last point.

Finally the number of users and their retention rate are extremely important points but they come naturally through improving the service in the direction of the player instead of the advertiser. In that sense it seems that they are secondary features.
Thus, in the perspective of this value map and specifically in trying to understand the process of value capture of the service provider we reach a two steps process.
First the value that the service provider must capture is non monetary. It’s the time invested on the app by the gamer. Then it can convert this value to a monetary one with the advertiser.

Thus the current model suggest that we could maybe capture more details and have a more accurate view of the the company’s value creation and capture processes if we modeled the value map with the player as the main adopter.

Gamer as Main Adopter

To remedy the limitations of the previous model we try a new abstraction with the gamer as the main adopter. Obviously that leaves us with the problem of explaining what role does the advertiser have. We propose an approach for modeling the online advertising system by placing the advertiser as a partner. This requires a paradigm shift from the traditional approach in which the relationship is based on a exchange of a nonmonetary good from its end and a monetary one from the enterprise, to an inverted relationship in term of monetary / nonmonetary exchange. But this model allows for a finer level of comprehension of their mutually beneficial cooperation.



Main Supplier
- Cheese Cat

Key Partners
- Google AdMob
- Amazon
- Google Analytics
- Apache Cordova
- Google Play
- App Store
- Unity

⊟ Supplier Value Network

Main Competitor
- Saru Tobi

Components
1. App distribution
2. Backend servers
3. Customized flow
4. Graphics
5. Hybrid application
6. User stats

Features
1. Arcade game
2. Leaderboards
3. Multiplayer game
4. Quality and finition of graphics
5. Rightly calibrated difficulty
6. Wide range of levels

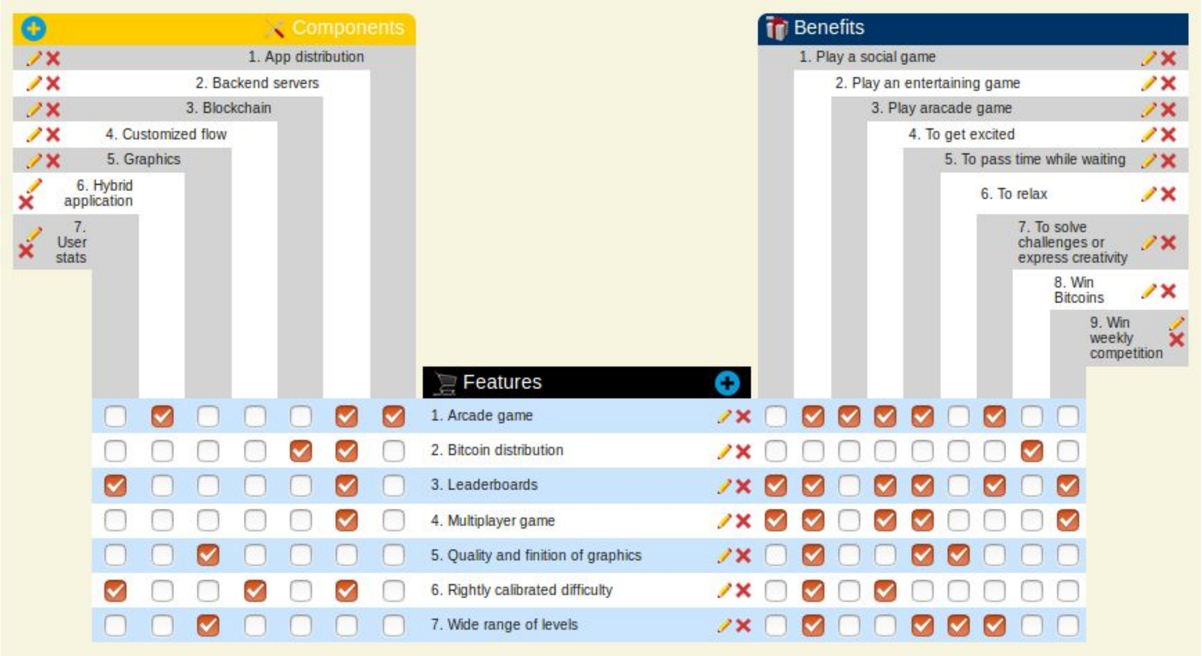
Benefits
1. Help audience discover local entertainement
2. Play a social game
3. Play an entertaining game
4. Play aracade game
5. To get excited
6. To pass time while waiting
7. To relax
8. To solve challenges or express creativity
9. Win Bitcoins
10. Win weekly competition

Main Adopter
- KingCoin (player)

Influencers
- Aarki (ads company)
- Google AdMob (ads auction)
- Salt (advertiser)
- Le Matin
- Bitcoin news ecosystem

Regulation Authorities
- AdMob
- Google Play
- App Store

Service Model



Components							Features	Benefits								
7. User stats	6. Hybrid application	5. Graphics	4. Customized flow	3. Blockchain	2. Backend servers	1. App distribution		1. Play a social game	2. Play an entertaining game	3. Play aracade game	4. To get excited	5. To pass time while waiting	6. To relax	7. To solve challenges or express creativity	8. Win Bitcoins	9. Win weekly competition
☐	☑	☐	☐	☐	☑	☑	1. Arcade game	☐	☑	☑	☑	☑	☐	☑	☐	☐
☐	☐	☐	☐	☑	☑	☐	2. Bitcoin distribution	☐	☐	☐	☐	☐	☐	☐	☑	☐
☑	☐	☐	☐	☐	☑	☐	3. Leaderboards	☑	☑	☐	☑	☑	☐	☑	☐	☑
☐	☐	☐	☐	☐	☑	☐	4. Multiplayer game	☑	☑	☐	☑	☑	☐	☐	☐	☑
☐	☐	☑	☐	☐	☐	☐	5. Quality and finition of graphics	☐	☑	☐	☐	☑	☑	☐	☐	☐
☑	☐	☐	☑	☐	☑	☐	6. Rightly calibrated difficulty	☐	☑	☐	☑	☐	☐	☐	☐	☐
☐	☐	☑	☐	☐	☐	☐	7. Wide range of levels	☐	☑	☐	☐	☑	☑	☑	☐	☐

Value Map - Alignment - Components/Features/Benefits

Captured value creation

We can see that this abstraction highlights far more efficiently the benefits that the player has from playing the game. It also shows features on which CheeseCat can actively work to improve those benefits. It also makes clear that the main value created is in the game features.

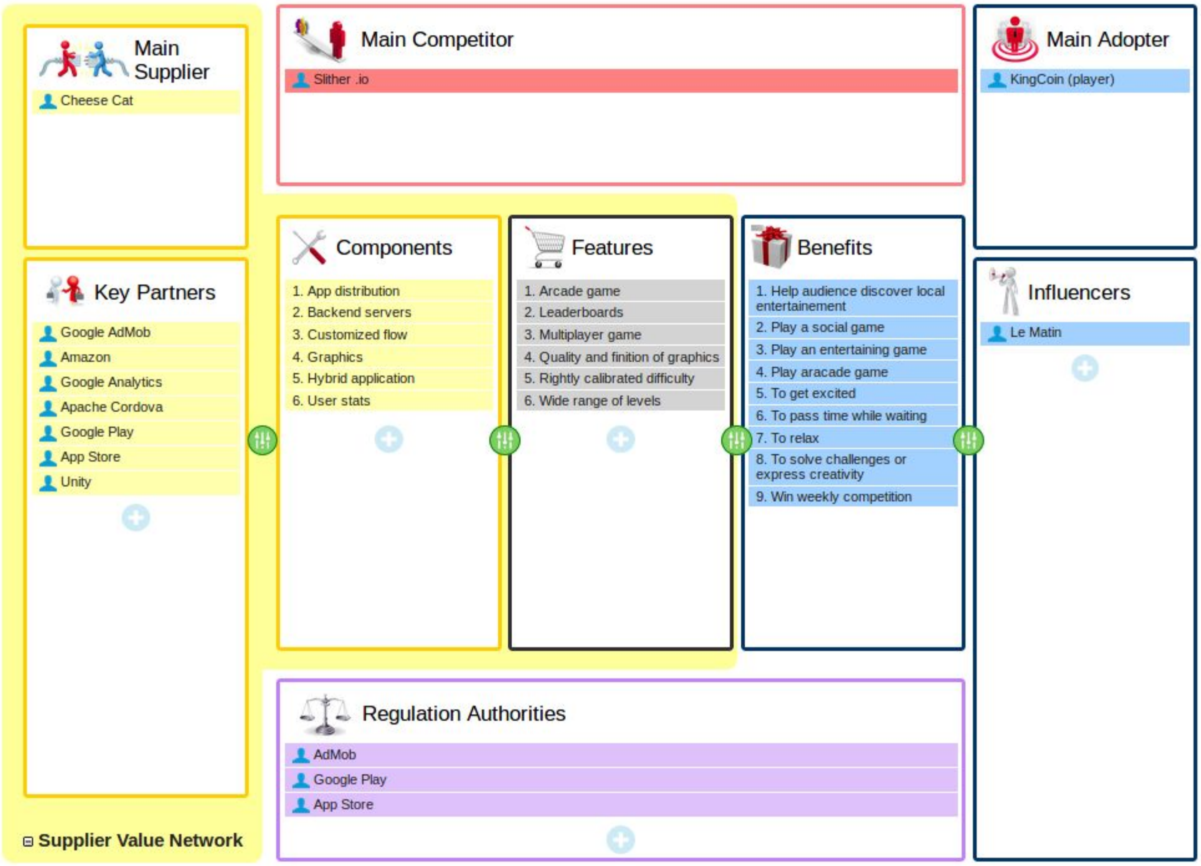
Limitation of this model

This model those not explicitly expose how CheeseCat can capture a monetary benefit from the value it create. As we mentioned earlier it is a two steps process, in a first time the company must be able to provide for a game that the player is disposed to play for a certain duration and then this time can be monetized through the advertiser.

We conclude that the two maps must be read jointly to fully represent the process of value creation and capture of the company.

In-app purchases

We design a third model in which the advertiser is no longer present and the player is the customer. In the future CheeseCat aims at offering paid games in its products as well. Thus, it searches to know if there is a conflict between the two revenue model and if the third model is applied to a new product, whether it would have to significantly change its features or at least its focus on the different aspects of the service.



Main Supplier
- Cheese Cat

Key Partners
- Google AdMob
- Amazon
- Google Analytics
- Apache Cordova
- Google Play
- App Store
- Unity

Main Competitor
- Slither .io

Components
1. App distribution
2. Backend servers
3. Customized flow
4. Graphics
5. Hybrid application
6. User stats

Features
1. Arcade game
2. Leaderboards
3. Multiplayer game
4. Quality and finition of graphics
5. Rightly calibrated difficulty
6. Wide range of levels

Benefits
1. Help audience discover local entertainement
2. Play a social game
3. Play an entertaining game
4. Play aracade game
5. To get excited
6. To pass time while waiting
7. To relax
8. To solve challenges or express creativity
9. Win weekly competition

Main Adopter
- KingCoin (player)

Influencers
- Le Matin

Regulation Authorities
- AdMob
- Google Play
- App Store

⊟ Supplier Value Network

Service Model

Captured value creation

As we can see the model almost does not change compared with the previous one (the alignment remains identical, we do not display it). The major change is the CheeseCat's value capture process.

Having in mind that the first process shown, (in which the value captured by CheeseCat is through the monetization of user time), is mechanical, (ie, the reporting is automated and CheeseCat cannot negotiate the prices) and that providing for an entertaining game is the determining factor in CheeseCat revenue stream. We can safely assume that having the player as the main adopter and customer does not significantly change the process of value creation and capture of CheeseCat and that having both types of products, and even some hybrid form should not lead to a conflict.

I.3 CheeseCat - Organisation Analysis, Issues and Solution

I.3.1 Analysis

CheeseCat aims at offering a smooth gaming experience to its users and faces many challenges to achieve this results. It is a small organisation currently composed of 5 people to answer all of the aspects of this service. It uses fifteen IT systems that provide services both internally and externally. Most of those services are outsourced. They can be grouped in several categories. Those services and their users are listed in the following table. The internal services such as the HR, accounting and cash-flow management while extremely important to any organisation, are relatively straightforward in a small organisation such as CheeseCat. They will require adjustments and resource allocation while the company growth. CheeseCat will manage this future evolution by a continuous usage of SEAM. In this paper, for clarity, those aspects are not discussed.

Mission	
Mobile Games BitcoinBandit Two Kids Very Mad Race	CheeseCat's products, or with a service oriented POV, the applications where the gaming services is provided
Social Networks /Com. Channels Mailchimp / Facebook / Twitter /Instagram	Community Management, User feedbacks and promotion.
Administration	
Trello	Web-based project management application
Google Drive	Internal shared storage
Google Analytics	Web Analytics service
Gogs	Code-base, bug tracking infrastructure
Slack	Internal messaging and coordination system
Infrastructure	
Production Servers	Backend servers for the games online features
Developpement Servers	Backend server for testing and building. Also used to host Gogs
Publisher stores App Store / Play Store	Games distribution channels

These services support its two core missions: creating games and promoting them.

Many business processes are taking place in the company. The main one in term of resource allocation is delivering new features to the application. As this process encompass many activities that are present in other processes, (ie. feature selection is also used while designing a new game, the collection of user feedback from social network is also used while promoting the game...) it makes for a representative process of CheeseCat's activity. We thus describe, analyse and model this process.

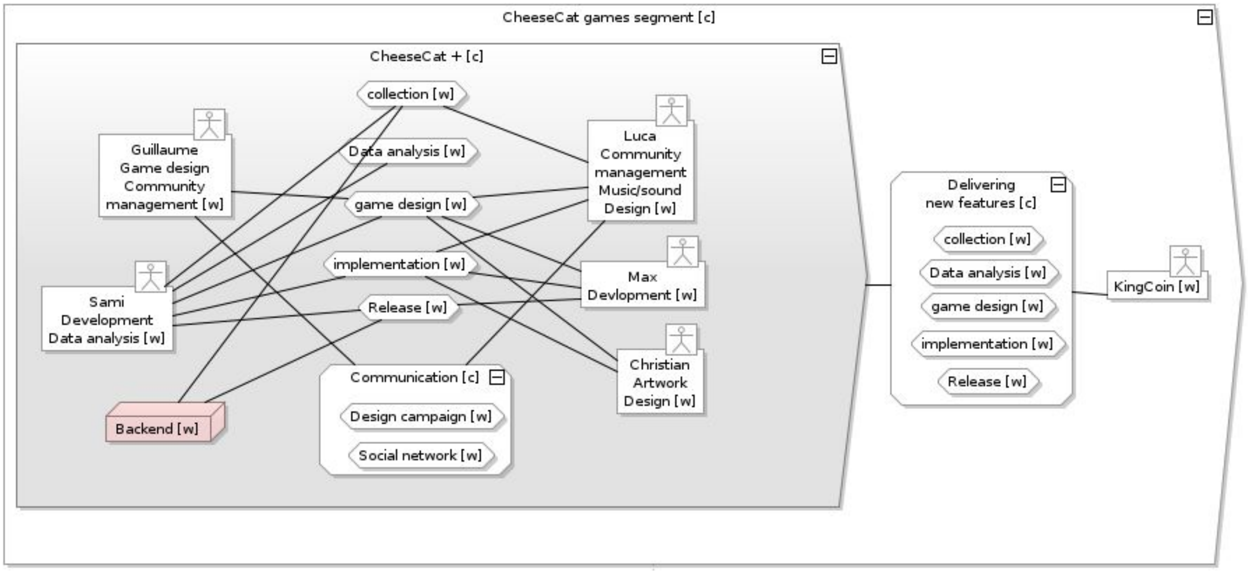
Collection:
The first step in the feature selection is to collect user data. Two main sources exists for that. The first sources are the social channels through which users can send their feedbacks which are qualitative sources. The user send its feedback through Facebook for instance, then the community manager determine if the feedback is useful for 1) fixing an unnoticed bug, in which case he synthesize the user message and submit a new bug in the Gogs bug tracker
2) improving the game via some new feature, in which case the feedback is submitted in its raw format in a backlog Trello board.
The second source is the analytics. This is a purely quantitative source. The action of the players in the game are anonymously collected and sent to Google Analytics.

Selection:
Each monday morning the whole team meets and discuss to determine the features to be implemented during the week. The qualitative feedbacks are used and periodically a software developer has conducted a factor analysis to help determine the priorities. Then a brainstorming is conducted and the priority and length of implementation of each features are discussed. And finally the selected features are added to Gogs. If any user feedback was left unanswered in the previous step a response is written at that moment.

Implementation:
The features are then implemented during the week and the four aspects, the design, the drawings, the music and the development are coordinated via Slack and files are shared via Google Drive for music and design and Gogs for the code-base.

Release and communication
By the end of the week the update is ready and published in the stores. An update message is added to the release and depending on the interest of the features for the community a message is sent via the communication channels. A mail is sent to the mailing list via MailChimp.

We use the SeamCAD [5] tool to build the SEAM models that conceptualizes this business process. We use the company view notation template. [6] The next figure depict the AS-IS SEAM model.



SEAM Model AS-IS of the delivering new features process

I.3.2 Issues & Solutions

While the current process is generally working well for CheeseCat, some aspects can be improved and more importantly we identified multiple points that would prevent this system to be sustainable while the user base and human resources are growing.
Some of the problems are:

- Some roles are not clearly defined. For instance the community management is done in part in group, and for the most part by the community manager that sometimes need to wait on the the developers
- The data analysis is manually conducted at recurrent intervals by the developers, so there is no clear data analysis requirements and after an update the tracking of features impact is not straightforward. The manual analysis conducted provides with some prediction on which features the team should work on to improve some metrics but when a recent analysis is not readily available the team relies on its intuitions; The number of propositions emitted during the brainstorming is large, and filtering, selecting and scheduling them is complicated.
- The responsibilities are not clearly defined. All of the unusual events are handled during group meetings, which prevent a fast response time.
- The monday meetings cover all aspects of the games and the company strategy including the aspects that are interesting only for a subset of the roles. For instance the developers talk about technical aspects which are not interesting for the rest of the team, the community manager talks about what illustration would best be used in his communication which is an exchange interesting for the community manager and the designers only. Thus, while interesting for the whole team, the actual structure of those meetings would make them too long and not very productive in case of a HR growth.

I.3.3 Selected Solution

The first solution is represented by this project. As the enterprise is a system in continuous evolution, we are able to monitor and prepare for significant changes by reflecting them in SEAM. [7]

For instance, a first warning that seems clear from the AS-IS model is that CheeseCat is already using a significant number of disparate IT systems relatively to its size, and that its growth will require to integrate them. While having that in mind will allow to monitor carefully when an integration will start to be required, implementing an integrated system now would probably represent a fatal overhead.
Thus we focus on change that:

1. Improve the main business process now
2. Allow for growth in the near future

Thus the second selected solution consists in clearly defining a set of roles and their associated responsibilities. [8] This way even if the roles that we define outnumber the people currently working, the responsibilities will be clear for each team members, the monday meetings might be alleviated of some concerns and when hiring new people it will be clear what tasks to assign and how to integrate them in the different business processes. We propose that solution in the next chapter and discuss governance in Appendix 1.4.

The third selected solution is to design an analytical tool to complete existing analytical solution. An important part of the process is to select features to be designed according to some order of importance. This process is partly done with manual data analysis at some recurrent interval and partly rely on the intuition of the team. We propose to automate a part of this process to have current data analysis available all the time. As the intuition of the team is not something that we would want to limit, we propose this solution not to replace the brainstorming but to help prioritize. Data analysis also often help to uncover some patterns in utilization that neither qualitative assessment nor brainstorming could. The definition of the requirements and the details of its implementation are presented in Part II.

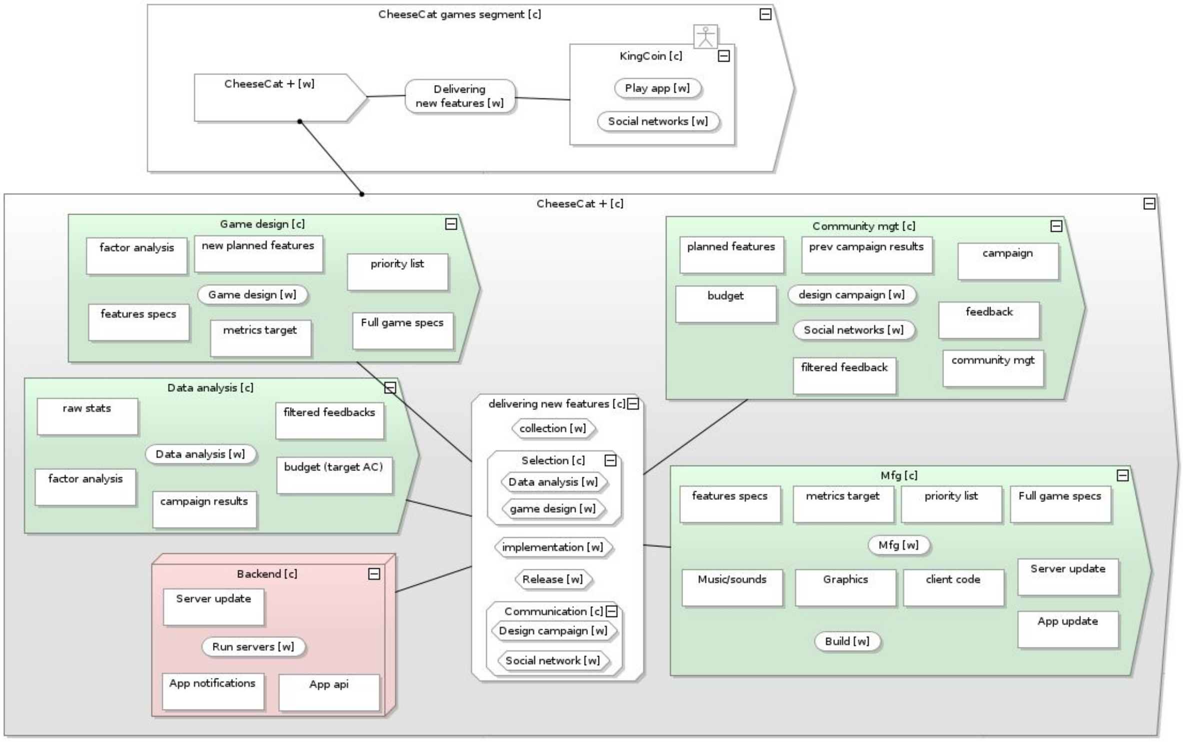
I.4 Roles and Responsibilities

We propose to create the role Game Design, which is separated from the actual implementation of the game features. This role is active in the corresponding business unit. The basic responsibility of this role is to assure the general creative direction while taking into account marketing data, community feedback and results from data analysis. The second role we propose is the Data Analysis as in its current form it’s only an action belonging to the development role. Again, we create the corresponding business unit for this role. This allows the company for two things, in the short term the first phase of data analysis, the exploratory data analysis will be automated (cf Part II). In a second term it will evolve in a business unit with data analysts providing the game design with manual analysis based on the game (and will be

extended to external source, ie. trend detection). We also group the roles artwork, design, music and development into a manufacturing business unit.
We define the responsibilities of the different roles in the following table.

Role	Responsibilities
Community Management	In charge of promoting the game to the community via social networks and various communication channels. If needed it requires additional material from the designers. In charge of collecting user feedback via the different communication channels to synthesize them and to submit them to the game design. In charge of providing an adequate response to the user in the shortest possible time.
Game Design	In charge of the general artistic direction. It must collect feedback from all the available sources and decide on feature implementation and schedule.
Design	In charge of producing the graphical elements according to the graphical chart, the feature list and priority. In charge of providing the community management with additional material.
Artwork	In charge of producing the artworks according to the feature list and priority. In charge of providing the community management with additional material.
Music / sound	In charge of producing the music and sounds elements according to the feature list and priority.
Devloppement	In charge of implementing the features and integrating the design, artwork and acoustic materials according to the feature list and priority. In charge of publishing frontend and backend updates to production. In charge of updating the data collection from the front-end when required.
Data Analysis	In charge of conducting data analysis from available sources. In charge of synthesizing the results and submit it to the game design. In charge of continuously tracking the evolution of key metrics and frontend updates impact on those. In charge of providing a target acquisition cost for various user segments to community management.

With those solutions we are now able to design the TO-BE model of the company. We omitted the role as their location in the business units is straightforward and their addition would limit readability.



SEAM Model TO-BE of the delivering new features process

I.5 Conclusion

In this paper we first identify the main stakeholders in CheeseCat ecosystem. With this results we are able to more precisely determine the process with which the company creates and capture value. We use SEAM to model the main value creation process. We identify problems with the current situation and propose three solutions to remedy in the present and prevent problems from arising. The first one is the continuous modeling of the company to capture changes in requirements and to identify early potential problems. The second is a clear definition of the roles and their associated responsibilities. The third is an IT system that aims at filling a gap in the current process.

I.6 References

[1] M. Reeves, K. Haanaes, J. Sinha, *Navigating the Dozens of Different Strategy Options*, Harvard Business Revue, 2015

[2] E. Ries, *Creating the Lean Startup*, Inc. 33 (8): 56–63, 2011

[3] A. Golnam, V. Viswanthan, C. I. Moser and A. Wegmann. Value Map: *A Diagnostic Framework to Improve Value Creation and Capture in Service Systems*. The 2nd International Symposium on Business Modeling and Software Design, Noordwijkerhout, the Netherlands, 2013.

[4] http://www.tradeyourmind.com

[5] L.-S. Lê and A. Wegmann. SeamCAD: *Object-Oriented Modeling Tool for Hierarchical Systems in Enterprise Architecture*. 39th IEEE Hawaii International Conference on System Sciences, Kauai, Hawaii, USA, 2006. http://lams.epfl.ch/seamcad/

[6] A. Wegmann, G. Regev, I. Rychkova, L.-S. Lê and J. D. De La Cruz et al. *Business-IT Alignment with SEAM for Enterprise Architecture*. The 11th IEEE International EDOC Conference (EDOC 2007), Annapolis, Maryland, 2007.

[7] A. Wegmann. *On the Systemic Enterprise Architecture Methodology (SEAM)*. International Conference on Enterprise Information Systems 2003 (ICEIS 2003). Citeseer, 2003.

[8] A. Ruppen, N. Clément, P. Balabko and A. Wegmann. *The Value of Roles in Modeling Business Processes*. Workshops on Business Process Modeling, Development and Support (BPMDS), Riga, 2004.

Part II: Analytical IT Solution

II.1 Introduction Analytical Solution

Analytics tools allow to collect data on the game usage and users demographics. Thus those are tools that allow for quantitative evidence collection. There are various tools that allows to collect usual data as the number of users, number of session per user, session duration and so on. They allow to have information not only on a subset of users but on all users, so they answer the problem of the sampling methodology to use.

II.2 Business Requirements for Analytical Solution

The analytics should answer multiple needs of the company. [9] We list here the four basics requirements and detail them after:

- They should provide the basic health indicators
- They should allow to understand the game audience
- They should allow to predict some behaviour
- They should allow to track updates impact on behaviour

Basic Health Indicators
As the first chapter mentioned there are two main revenue stream for CheeseCat depending on the product. The advertisement and the in-app purchases. In the case on the advertisement the revenue is determined by the number of click on the ads and the number of displayed ads. Thus the basic financial indicators includes the number of daily user, the session duration and the retention rate.

Audience
The designer should be able to have some information about their specifics of the audience. In particular they should have demographics information such as the countries in which the game is played, the age and gender of the users, the type of devices they use… The tool should also provide some behaviour information such as the referrer the propensity to share…

Prediction
On top of that with the advance of data analysis a solution should be able to give some indication on correlations with some behaviours and events, typically the basic health indicator. Obviously a lot of external events that are typically out-of-reach of the company decision can make up for it. For instance it could be that simply the weather could be an explanatory factor for a purchase decision. But there might also be factors that are within the direct reach of the development team that allows to improve the product. As we saw within the lean startup methodology the development team has the ability to test a feature and the effect it has on the game on a short time span. Nonetheless in a game the range of feature that can be developed is extremely large. Thus even within that setup the choice of which feature to implement

remains a critical question. Obviously one can only make an hypothesis prior to testing its effect. But we might find preexisting indication that this is a direction in which to further dig.

Impact Tracking

Once features have been rolled out, the company should be able to track their impact on their corresponding metric.

II.3 Implementation of Analytical Solution - Analysis, Issues and Solutions

II.3.1 Analysis

There are already existing and widely used tools to conduct basic web data analytics. Thus the requirements will only cover aspects that other providers do not already offer.

In particular the most used web analytics platform is Google Analytics. [10] It already cover three of the four main business requirements, basic health indicators, audience and impact tracking.

We thus focus on the remaining point *prediction*.

Motivation

Shortly after the launch of the first game a first experiment was made about the retention rate. Specifically we wanted to find factors to predict whether users would play 100 times or more. (We used the number of levels, because we focused more on the number of game played than the retention rate at that time, we found later that our conclusions are still relevant, as one would expect the two numbers are significantly correlated).

We used available data from the database:

- The average score per level
- The score at the 1, 2, …, 10th trial
- The min / avg / max fps (frame per second)
- The number of level played in the first session
- Whether the user had entered his wallet address
- Whether the user had reached level 2 in his first session
- The ratio of time spent with the controller pressed in the first 10 trials

We used the random forest as classifier and estimated the predictors importance by permutation with Matlab.

We determined that three predictors were by one order of magnitude more important. In order:

- The average fps
- The average score
- Whether the user entered his wallet address

The first one, the fps seems obvious retrospectively, but since we all had good devices we didn't realize that the game could have poor performances. In particular with user in Brazil, India and Iraq with devices we limited graphical capabilities. It was clear that under 45 fps the game was frustrating and that user gave up playing in those conditions. The first decision was thus, from the developer team to improve the game performances.
Then with the two other factors the question had a less direct consequence. Thus, a brainstorming was setup with the graphic designer and the game designer and the decision was taken to implement two more features.

1) On the the user's app first launch a tutorial had to be shown to the user to explain the game concept and shortly how to install a wallet
2) The coins were positioned slightly higher in order to make the game easier

With those three features we were successfully able to augment the 7-day retention rate by 3%.

Usage scenarios
The system should store every possible relevant informations about the application experience. It should help finding correlations between those various attributes. A complete data analysis still remains a human task as there are various steps required to find a good model depending on the data. Basically those tasks consists of comprehend the input data set, selecting an appropriate classifier and selecting the parameters in order to have good model.
The first step, the exploratory data analysis, is often achieved through data visualization. [11] Plotting data against each other, displaying an histogram of the data are routinely part of the analysis.

II.3.2 Issues & Solution

We want to automate those tasks as much as possible by providing a tool that can extract data from the database and display them in various graph through a human interface.

For instance in the previous scenario, it was determined that the average fps was a major cause of churn. The use of a sophisticated machine learning method, random forest helped in discovering that but was not necessary. Simply plotting the various data point was largely sufficient to determine that under a certain threshold the churn rate was nearly 100%.

Other usage scenarios includes

- Plotting the time of various maps in an histogram
- Plotting the score against the number of share (determine if a user is more eager to share the app if his score increases)
- Plotting the number of badges against the number of purchases
- Plotting the time spent in the character menu against the purchases

II.3.3 Selected Solution

Thus the solution should collect as much as possible from the user behaviour.
From that the two main data visualization tools routinely used in exploratory data analysis, the histogram and the scatter plot should be implemented and the results displayed in a user friendly front-end.

II.4 Functional Requirements

II.4.1 Front-End Collection

The statistics collection should be fast and easy to implement on the front end. As the tool designed here does not aim at replacing existing analytics suit but complement them it would be easier to reuse existing code. Specifically, a call is made to Google Analytics at each screen opening and for all events. A screen opening is a call that takes no arguments and an event calls takes five arguments described in the next section.
Thus we will use the proxy pattern and design a function that take the same number of argument and then forward the call to the various analytics tools. So that almost no modifications is required on the front end.

II.4.2 Storage

The statistics should be collected live through existing infrastructure. The storage method is thus the same database as the game.
Each click or choice of the user should be stored.
To distinguish between various action in the game we label an action with five values:

- Screen, to distinguish in which screen of the app the action was made
- Category, whether it is a button, a list, a checkbox...
- Action: the name of the input, (ie. BACK, PROFILE,...)
- Label: An optional text field to distinguish between actions
- Value: an optional numerical value

The access to this have two properties:

- It should be possible to aggregate data from this list, for instance the number of share is derived from the number of entries labeled share
- It should be able to join various table, as for instance being able to plot the number of badges which are stored in the user table against the number of share which are obtained from the aforementioned stat table

II.4.3 Back-End

Again the backend should reuse available resources and require a light design. The game currently serves requests through an http server designed in Node.js.
This http server should serve the end-user backend designed in html and the http routes for the subsequent api calls it makes to query the various relation. The data should be send in a light format, namely JSON, because it's easily parsable in the browser and the rest of the api is written with this data format.
It has to serve the request about what table and what fields can be aggregated. It has to make the sql queries for the histogram and scatter plot and aggregate the results in buckets in the case of histogram.

II.4.4 Front-End Visualisation

The frontend is written in HTML for the general structure of the display, the queries are made in javascript and the visualisation is also made with js using the canvas and 2d context offered by modern web browsers.

II.5 References

[9] H. Chen, R. Chiang, V. Storey, *BUSINESS INTELLIGENCE AND ANALYTICS: FROM BIG DATA TO BIG IMPACT*, MIS Quarterly Vol. 36 No. 4/December 2012
[10] https://www.google.com/analytics
[11] J. Tukey, *Exploratory Data Analysis*, Pearson. ISBN 978-0201076165, 1977

Conclusion Semester Project

In this semester project we presented three solutions to improve the main business process of CheeseCat. More importantly this project lay the foundations for the iterative use of modeling the company with the Value Map and SEAM. As a complex system, managing its evolution and being able to readily react to changes is one of the main key to success.

Appendix 1 : CheeseCat Strategy - Key Learning Points

1.1 Project and Report Structure

As the format of this semester project is in its third version, it was very interesting to learn how to structure it. The main problem of the two first versions was that they mixed the analysis of the company structure with the tool solution design.
As the goal of this semester project is twofold, first understanding how the modelisation of a company ecosystem with the Value Map and its internal organisation with SEAM helps to pinpoint processes that are suboptimal and that could be improved with a modification of the structure or the processes. From that perspective we can design IT systems and their requirements that align with the company core missions.
It is important to emphasize that the models in themselves are analysis and communication tools.
While the first formats were interesting for stakeholders with an interest in business processes and IT systems it was basically not reusable in any other situations.
While the strengths of the second format is two fold.
First, in clearly separating what knowledges are useful from a business standpoint from the implementation details which are more useful from an engineering standpoint.
Secondly, it follows from the first strength that the first part will be extremely useful for CheeseCat in the future. It will be able to maintain internally updated models of its ecosystem and internal organisation and will be able to adjust its structure whenever they are no longer aligned. It will also be able to reuse the first part (with few modifications) with external actors such as investors which are typically interested in verifying that the company's internal structure is actually able to support its mission.

1.2 Discussion on who is the Main Adopter

One of the most interesting aspect from the first two products is the twofold process to create and capture value. The value map helps in determining who are the stakeholder and what are their relations. In order to get a meaningful representation of the business process, it is required to determine who is the main adopter: for which stakeholder it is the most important that the company creates value. The usual approach is to select the main adopter as the most important customer. Thus, in the first model, the main adopter is the advertiser. Then the main business process should consist in creating value for this stakeholder. The relationship between CheeseCat and the advertiser is entirely automated and the created value is the number of user it can reach through our game and the ads display time. But CheeseCat cannot directly create that value. It should produce games that are interesting enough for user to spend time on them.

Thus basically all the effort of the company are directed to another stakeholder. From that standpoint it also makes sense to represent the user as the main adopter.

It should be noted that both representations are correct and simply display more or less details on different relationships.

We found extremely interesting to be able to analyze the company from both standpoint and in particular while as mentioned in the first part, the company size is not important enough to be able to negotiate the basis of its relationship with the advertiser, it is still a key element in CheeseCat's business process.

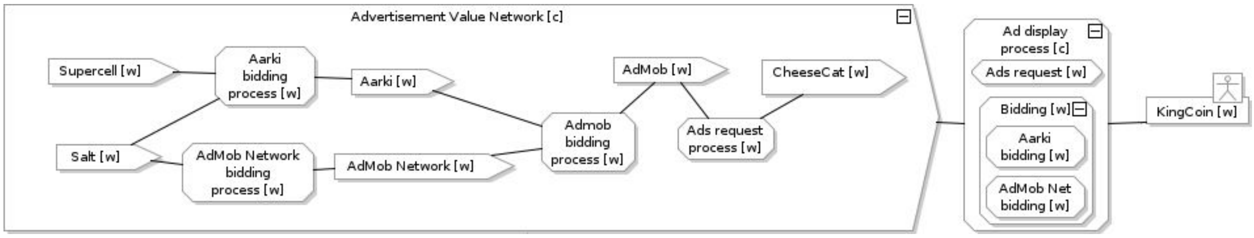
While most of its focus now is dedicated to the user, it will probably not always be the case and maintaining an updated Value Map will allow to not miss opportunities.
This open multiple possibilities for the future, for instance to keep tracking new offer from advertisers, as CheeseCat cannot directly negotiate with bigger structures, it can leverage competition between them by adding multiple provider and selecting the highest paying; another possibility is that in the future would be that CheeseCat designs a game for a segment which is more often targeted by advertisers...

1.3 Discussion on Advertisement Value Network

To get a better comprehension of CheeseCat ecosystem it's interesting to model the full advertisement ecosystem.

In this paper to improve comprehension we only referred to the advertisement value network as *the advertiser*.
CheeseCat is actually serving ads from multiple sources through one auction platform AdMob. Admob allows for any publisher (advertiser network) to submit and for developer to select ads publishers. By default the only publisher in AdMob is the AdMob Network, in which case the only ads served are through this network. When selecting multiple networks in Admob, the process becomes more interesting for CheeseCat. When a display is available in the game (typically at the end of a level), an ad request is sent to AdMob, the request is then forwarded to all the networks that then act as bidders. They propose a price, prices that fell under a threshold selected by CheeseCat are discarded and if any bid remains, the highest price is selected and the corresponding ad is sent for display.
The same process comes into play between the ads networks and their clients. For a given request, if the user's demographic match the client selection a bid is placed for the display and the highest paying wins the display. The bidder can bid on a cost per click (CPC) basis or on a cost per mille impressions (CPM) basis. The cost per mille is converted in a cost per click with the historical click through rate of the application in order to allow both types of bids to compete for the same display.
The following SEAM Model shows the *display ad* process across the advertisement value network. It contains two Ad Networks (CheeseCat have seven networks, but for the clarity we

limit to two), AdMob and Aarki, and two clients, Salt, a telecommunication company and Supercell a mobile game company.



SEAM model of the advertisement segment

The segment contain the advertisement value network and the app end user

1.4 Discussion on Management and Management Style

As CheeseCat is a young organisation of five people, that they all entered almost at the beginning, it's gouvernance system is kept simple, all the decisions are discussed and agreed unanimously. The current system works well for CheeseCat as the individual composing its team are highly motivated and the company strategy is kept simple.
That system while effective in the current setup cannot be maintained as the company grow. A simple counting argument shows that the time spent in discussing could grow quadratically with the number people.
Nonetheless, what the current structure shows is that the inputs from all the employees is valuable for the company. Thus, CheeseCat would like to maintain a system in which all the voices can be heard.

One such governance system is sociocracy.[12][13] It is organized around four main principles:

1. All the decisions are made by consent, a decision is taken when all substantial objections have been properly addressed
2. The organisation is composed of circles
3. Each circle elects a representative that also join the circle at the next level
4. Election, as a decision, is also made by consent

Another system of governance based on sociocracy is Holacracy. [14] It is similar to sociocracy as it is also based on circles.
It adds one main building block which is at the core of its structure, the *role.* A role is defined by its purpose, domains and accountabilities but not by the person that is assigned that role. Thus, a person can be assigned multiple roles.

CheeseCat organisation in it's current form can be assimilated to a Holacracy with only one level and one circle. The specific definition of roles and their responsibilities in Part I, allows CheeseCat to implement this governance system.

With the help of the iterative approach in using the Value Map and SEAM models, CheeseCat will be able to define new roles, responsibilities and circles when its structure will require changes.

Appendix 2 : Analytical IT Solution

2.1 Implementation Backend

The backend is implemented while reusing the existing backend infrastructure. The backend is built with Node.js [15] using the Javascript language. The backend serves multiple HTTP routes. The encryption is made at the proxy level, using Nginx [16] which runs on the same machine, proxying HTTPS requests to the HTTP backend.

It serves two types of routes, the API routes, which allow the front-end to make custom queries and the static routes which serve the static HTML, CSS and Javascript files composing the frontend. All the routes are written on top of a parsing and routing library, Express. [17]

The static files are cached in memory to prevent IO overheads (in this specific case, as the tool is mostly used internally the overhead of reading the files to disk would be marginal, but as the caching mechanism was already built for routes requiring a higher throughput - the game API - it is simply reused).

The dynamic routes are kept as simple as possible, delegating most of the work to the DBMS which is already optimised to work on large datasets. Thus, the routes are in charge of cleaning the inputs, storing the SQL queries, calling the DBMS with the queries along with the cleaned inputs, parsing the results and sending back the results to the frontend. In the specific case of the histogram it is also in charge of binning the results.

In the histogram case there are three dynamic routes, one to get the existing tables, one to get the column of numeric type for a given table and one to actually query the content of the selected table/column pair. The tables and rows are cached but not hardcoded, the backend actually has to query the database to discover them. This is a change from the common practice as it allows for two main advantages:

1. as the project is quickly evolving, database migrations are quite common and this system can thus adapt to a new schemas with zero overhead for the development team
2. this makes the system generic and reusable in other projects.

In the scatter plot there are two routes, one which answers with all the rows that can be aggregated with respect to the user table (as previously) and one to query the database for the content of the aggregation.

One other route allows non-admin but registered users to insert events into the database.

2.2 Implementation Database

The database implementation is two fold. As the game already have a database in a Postgresql [18] DBMS with usual tables such as user and sessions, and game related ones such as races and scores, we can already extract important informations from them. But to answer the specifications, we miss behaviour data.

We thus need to change the current DB schemas, while taking into account the front-end requirement that specifies that we must introduce minimum changes to the way data is collected. To collect an event in Google Analytics, one call is done in the front-end with four parameters. Three Strings: EventCategory, EventAction, EventLabel and one Long: value. Google Analytics tracks as well the screen in which the call was made by calling the underlying API. Thus we add a Table to the database following the same schemas. We also add a column userId, to allow aggregation. We add an index per column to allow fast sort and range selection. As described higher six SQL queries are designed, five SELECT for each GET route and one INSERT for the POST route.

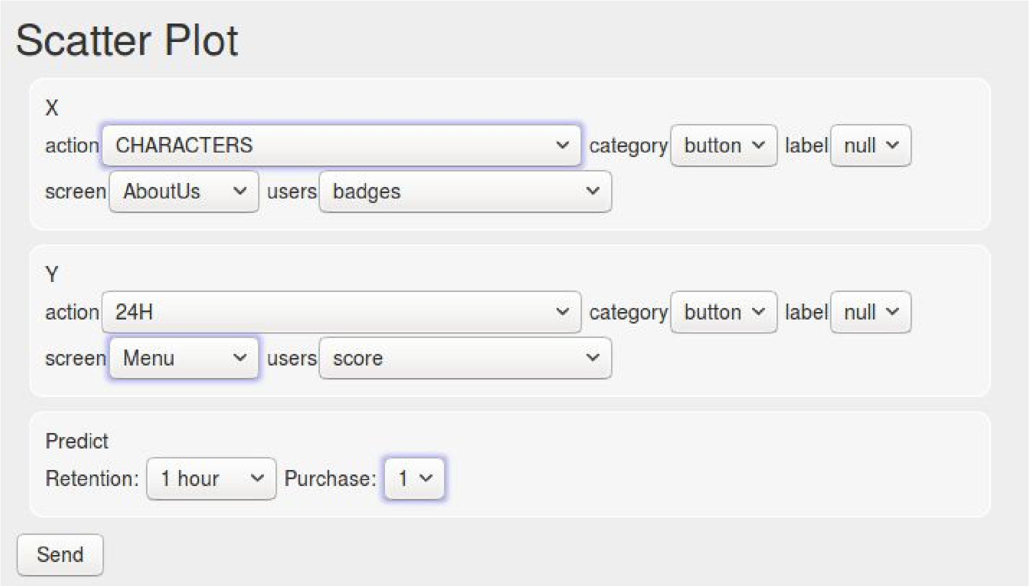
2.3 Implementation Frontend - collection

The collection part is the easiest. It is implemented in the game which uses the Unity Game Engine. [19] We design a proxy class that implements the functions we use in the Google Analytics object and substitute the latter with the former. As mentioned we maintain Google Analytics usage and thus the class should call both the LogEvent function from the Google Analytics object and the call to our servers. The call is a simple HTTPS request with JSON as the body of the request and the authentication mechanism is unchanged from the rest of the api.

2.4 Implementation Frontend - visualisation

The visualisation frontend is written in HTML, Javascript and CSS. Two libraries are used for the display, Bootstrap [20] for styling the inputs and Chart.js [21] for the plots. The calls are made to the backend with Ajax.

The histogram is made using box plots and the scatter plot is made by creating two data sets, one for each category and displaying them in two colors.



Scatter Plot

X

action CHARACTERS category button label null

screen AboutUs users badges

Y

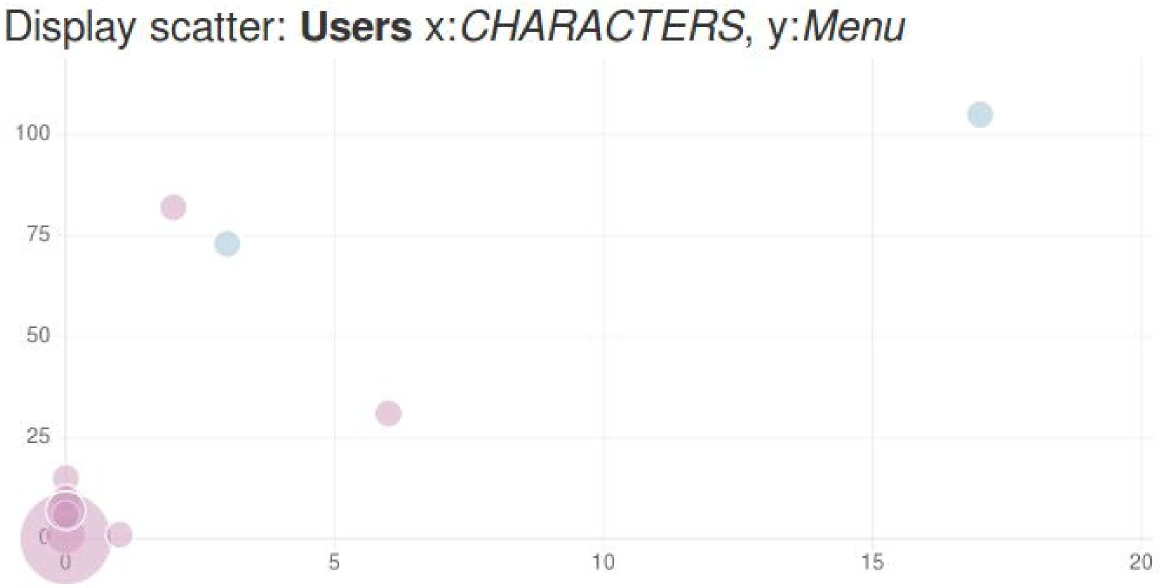
action 24H category button label null

screen Menu users score

Predict

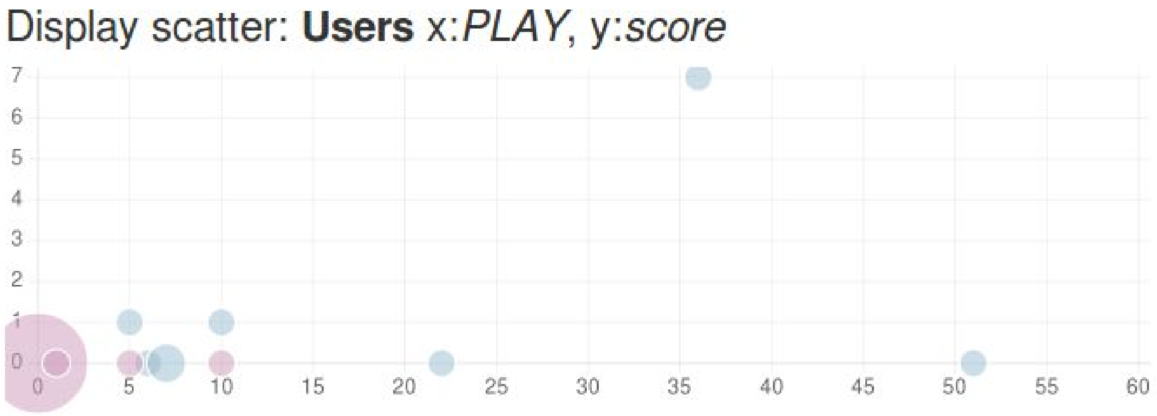
Retention: 1 hour Purchase: 1

A screen capture of the feature selection for the scatter plot



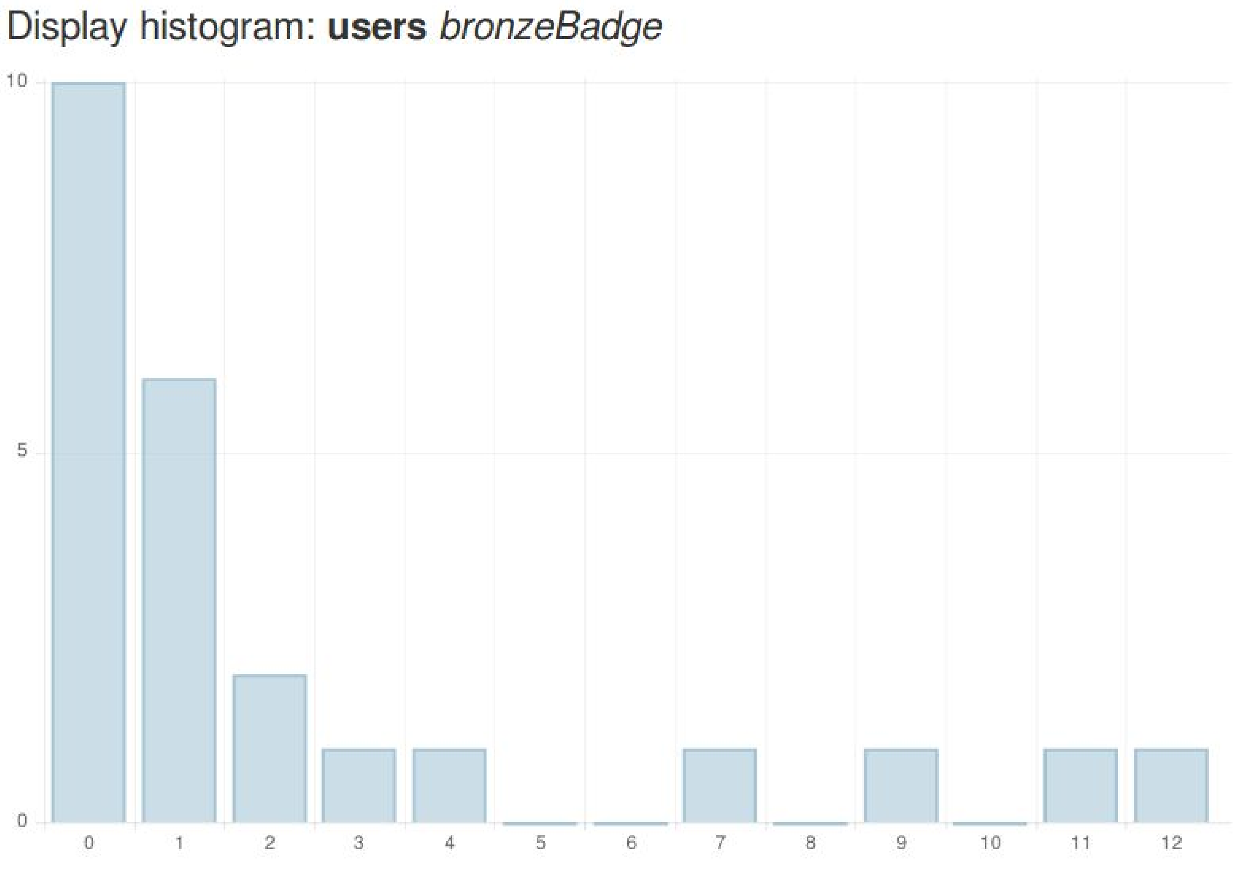
A screen capture of a scatter plot

The number of time a user went into the character screen is plotted against the number of time he went into the main menu. In blue users that made one or more purchase in red users that made no purchase.



A screen capture of a scatter plot

The number of time a user clicked in the play button is plotted against his score.
In blue users that were retained an hour or more, less in red.



A screen capture of an histogram

On the horizontal axis the number of silver badges,
on the vertical axis the number of users with that number of badge.

Appendix 3: References

[12] J. Buck, S. Villines,*Sociocracy: A Deeper Democracy. Washington DC: Sociocracy*, Info Press. pp. 31–50. ISBN 978-0979282706, 2007
[13] http://www.sociocracy.info/
[14] http://www.holacracy.org/constitution
[15] https://nodejs.org/en/
[16] https://nginx.org/en/
[17] https://expressjs.com/
[18] https://www.postgresql.org/
[19] https://unity3d.com/
[20] https://getbootstrap.com/
[21] http://www.chartjs.org/