

MOONGENE REALTIME MOBILE ANALYTICS

For games and apps

OVERVIEW

MoonGene real-time mobile analytical solution has been created by a team of game and web developers who were struggling to find a proper solution for games. Despite the fact that lion's share of profits on mobile market is generated by games there is no solution that is tailored for them. Most of the analytical solutions for mobile apps are too generic to be used in games out of the box. They usually answer questions that are not first on the list of producers and publishers.

MoonGene was created with games in mind, even though it can be used by regular apps as well. The goal is to maximize the average revenue per user, increase engagement and lifetime value of a player through a set of game-specific tools that analyze game's world balance and user's behavior.

The Objectives

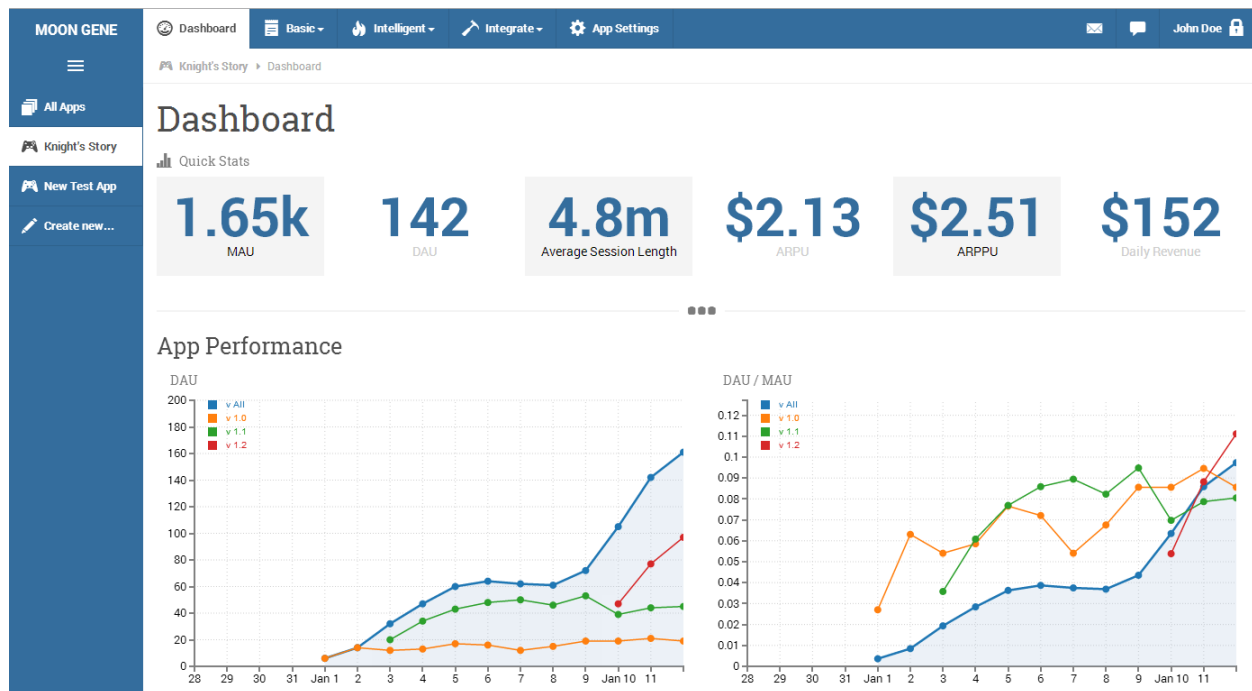
- Provide basic analytical functionality: DAU, MAU, geo, retention, etc.
- Provide behavior analysis
- Provide game economy and balance analysis
- Compute the most and least profitable user segments for further use during ad campaigns targeting
- Compute ROI for ad campaigns & networks

Technology

- Web Service is based on REST protocol
- Highly-concurrent and scalable architecture written in Scala
- Non-relational MongoDB database is used for storing data
- Client side integration through Android or iOS SDK, alternatively via direct HTTP REST calls
- Can be deployed in a public cloud (AWS, Azure, etc.) or on premises

WEB PORTAL

MoonGene Analytics is a web service and accessible through a web portal.



The web portal is used for:

- Analytics viewing
- Mobile apps management
- Managing payments and plans
- SDK integration into mobile apps

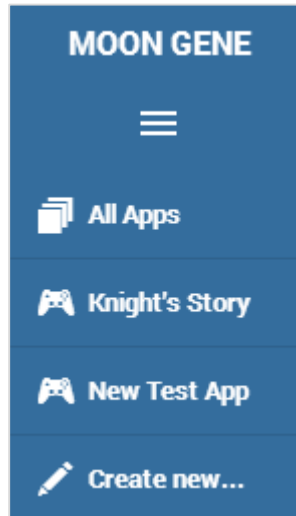
Accounts

New accounts can be easily created from the main front-end page. After providing basic information a user will be taken to the main dashboard from where he can start using the service.

The screenshot shows the 'Sign up' form in the MoonGene Analytics web portal. The form is titled 'Sign up' with a user icon. It contains four input fields: First Name (John), Last Name (Doe), Company (John Doe Inc.), and Email (john.doe@jdoe.com). There is also a Password field with a masked input. A blue button labeled 'Create account FREE' is positioned below the password field. Below the button is a link for 'Having troubles? Get Help'. At the bottom, there is a note: 'Please provide basic information to create an account.'

Apps

Every account has no limit on number of apps, so multiple apps can be created for different purposes, such as development, internal testing, closed beta-testing, release, etc. This will avoid mixture of data that is coming from different sources.



Apps are easily created by clicking on “Create new...” button on the left side panel.

Just app name is required to create an app and in a few seconds it is ready to be used. For every app there is token and ID that are generated. In case app token is compromised it can be changed in app settings page, but this will also stop collection of data from apps with previous token used, so has to be used with care and token should be kept secret in the app.

i Apps can be linked to multiple accounts, but there is just one user who has admin access to the app. This functionality is supported but not exposed in the web portal in the first release. Can be also transferred to a different account in case a company or app has been bought by another party.

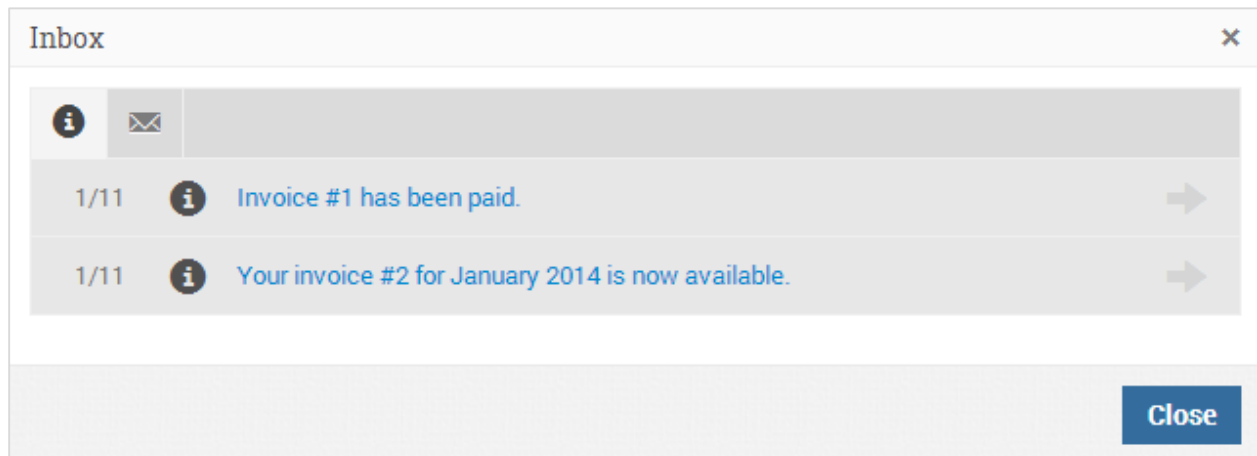
Notifications & Support



There is a built in support for notifications in the portal. Every time there is an important event a user gets notification. Currently those are mostly finance related, when a payment is made or invoice is available. But can be also used

for communication with support, etc.

Mailbox icon is used for viewing notifications & messages through a popup dialog.



Additionally support requests can be done using a Chat icon. They include a few pre-built topics for requests.


Customer Support

Topic

Billing

Message

Type your message ...

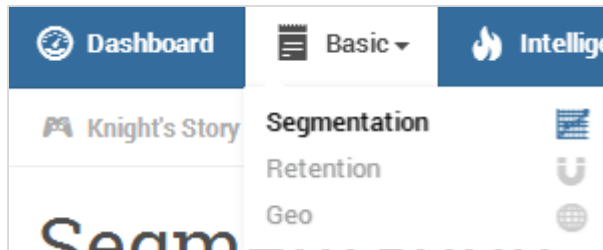
 Send

Technology

Web portal is built using:

- Play Framework, Scala
- MongoDB
- jQuery & Multiple JavaScript libraries
- D3JS.ORG library for graphs
- Uses REST protocol to request data
- Payments integration with Stripe.com service

BASIC ANALYTICS



Basic Analytics include features that are generic for all apps. There is no delay in processing events so data is always accurate and up to date.

i Funnels are supported and data is collected, but view is not implemented due to the fact that there are other game-specific features that allow doing funnel-like analysis.

Dashboard

Dashboard is the main page for all apps and updated in real-time. It includes multiple graphs with predefined dimensions.

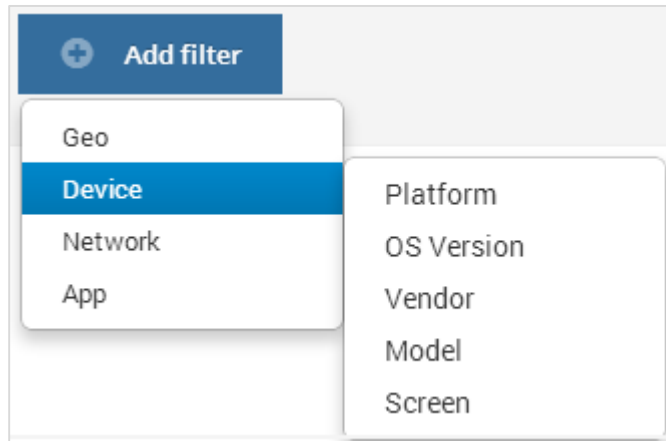


The following features are displayed on the dashboard:

- Quick stats, such as MAU (monthly active players), DAU (daily active players), Average Session Length, ARPU (average revenue per user), ARPPU (average revenue per paying user), Daily Revenue.
- Graphs of: DAU, DAU / MAU ratio, Sessions count, Average Session Length, Revenue, Payment Transactions
- Every graph displays per day values so a user can easily see increase or decrease in performance
- Every graph includes all versions of an app, that makes it possible to see whether performance of a new version is better than that of the previous one
- Every graph includes aggregated "All" version that is a sum of all game versions.

Segmentation

Segmentation allows analysis of cohorts by multiple dimensions.

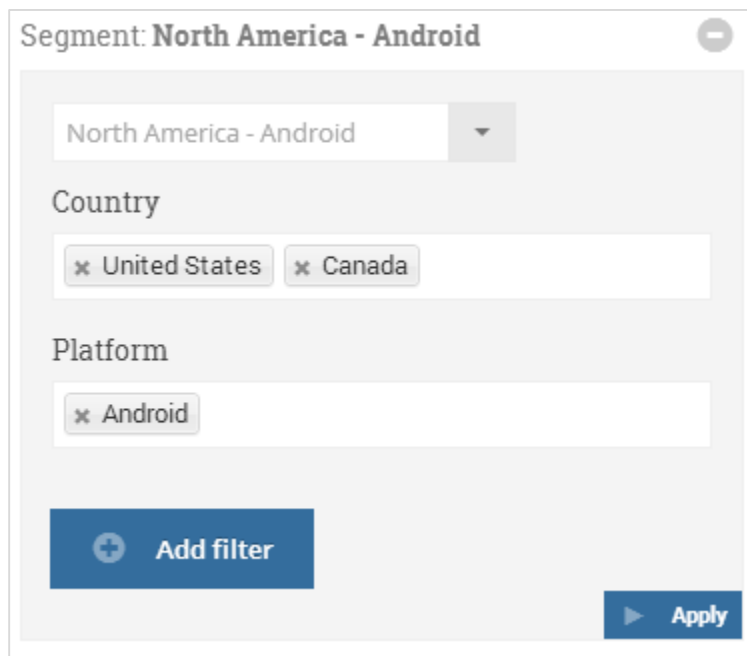


Currently supported dimensions:

- Geo: Country, City
- Device: Platform, OS Version, Vendor, Model, Screen
- Network: Carrier, Connection
- App: Version, User Type
- Marketing: Ad Network

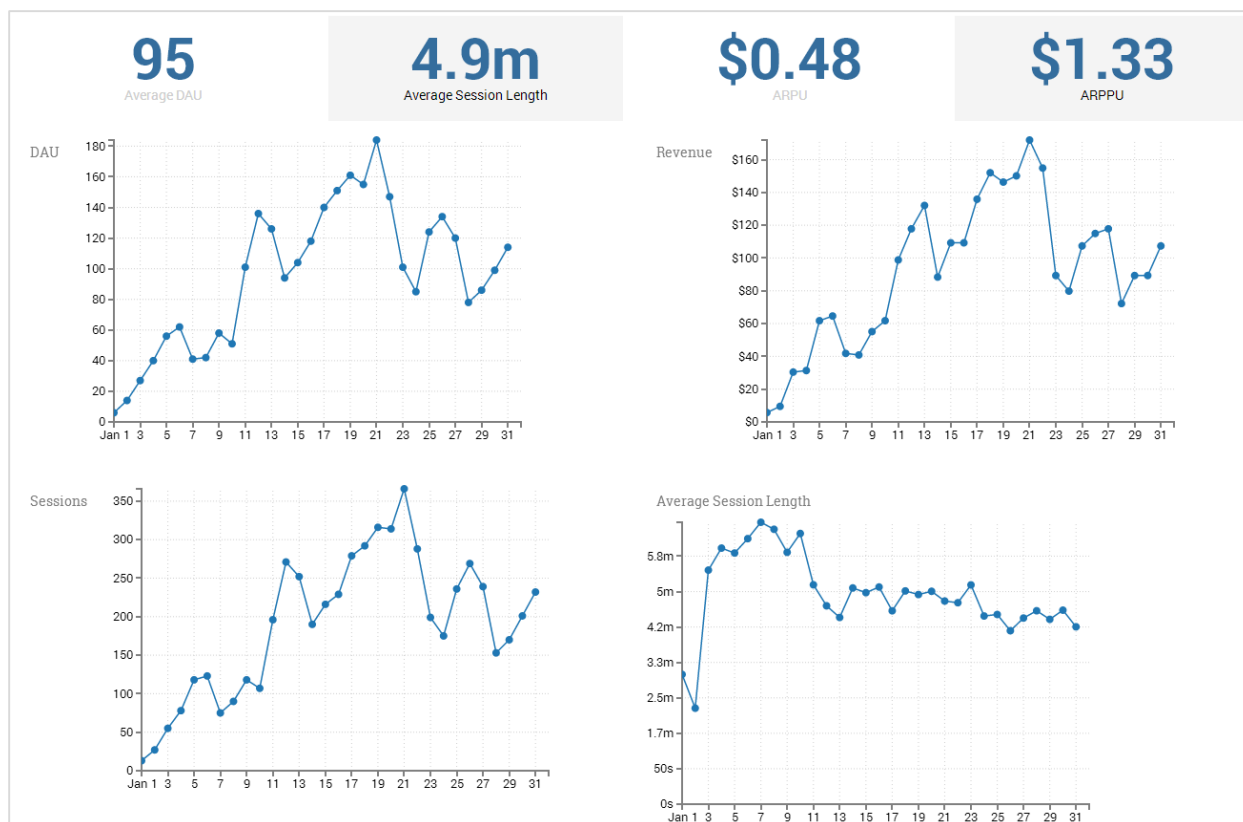


Marketing: Ad Network is supported but network IDs are currently not collected. Proof of concept is done, Ad Network Attribution will be available in the next release of the system.



User can create multiple segments and save them for later usage. Saved segments can be used across all features that support segmentation, for example Retention.

Once segment dimensions and time frame are defined graphs are drawn and quick stats calculated.



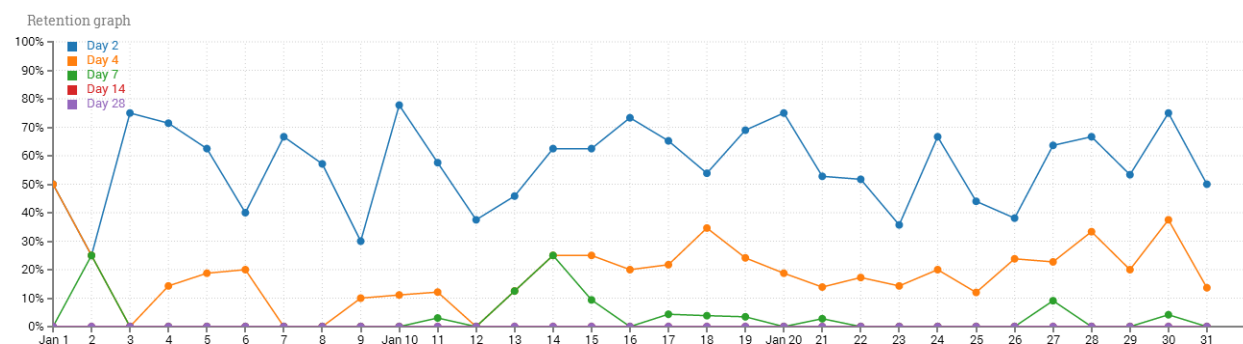
For every segment the following data is provided:

- Quick stats: Average DAU, Average Session Length, ARPU (average revenue per user), ARPPU (average revenue per paying user)
- Graphs: DAU (daily active users), Revenue, Sessions, Average Session Length

Retention

Retention is used to determine player's engagement. Just like in Segmentation a user can use dimensions and time frame to create a cohort.

For a mobile app and especially a game it is crucial to measure first month retention on certain days. Once segment and time-frame are defined a proper graph is built with evolution of retention rate.



In addition to retention graph a table is also computed that provides a table view for every day:

Date	Registration day	Day 2	Day 4	Day 7	Day 14	Day 28
Jan 1	100% 2	50% 1	50% 1	0% 0	0% 0	0% 0
Jan 2	100% 4	25% 1	25% 1	25% 1	0% 0	0% 0
Jan 3	100% 4	25% 2	0% 0	0% 0	0% 0	0% 0
Jan 4	100% 7	21% 2	14% 1	0% 0	0% 0	0% 0
Jan 5	100% 16	63% 10	19% 3	0% 0	0% 0	0% 0
Jan 6	100% 10	40% 4	20% 2	0% 0	0% 0	0% 0
Jan 7	100% 3	67% 2	0% 0	0% 0	0% 0	0% 0
Jan 8	100% 7	57% 4	0% 0	0% 0	0% 0	0% 0
Jan 9	100% 10	30% 3	10% 1	0% 0	0% 0	0% 0
Jan 10	100% 9	22% 2	11% 1	0% 0	0% 0	0% 0
Jan 11	100% 33	58% 19	12% 4	3% 1	0% 0	0% 0
Jan 12	100% 24	38% 9	0% 0	0% 0	0% 0	0% 0
Jan 13	100%	46%	13%	13%	0%	0%

Time Ranges have a few predefined options for user's convenience:

Time Range: Jan 1 - Jan 31 (31 days)

Past 3 days

Past 3 days

Past 7 days

Past 14 days

Past 28 days

Custom

Current selection: Jan 1 - Jan 31 (31 days)

January, 2014

Mo Tu We Th Fr Sa Su

30 31 1 2 3 4 5

6 7 8 9 10 11 12

13 14 15 16 17 18 19

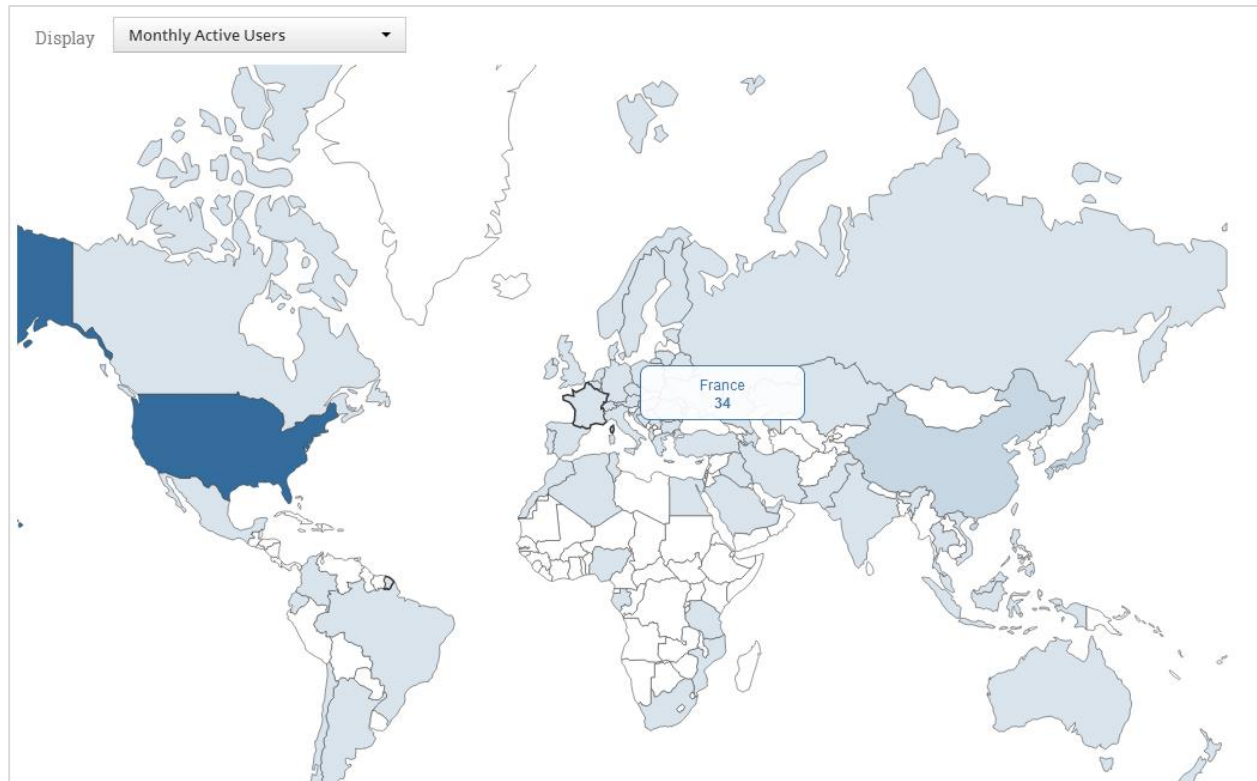
20 21 22 23 24 25 26

27 28 29 30 31 1 2

Apply

Geo

Geo data is computed for every new player. Data is aggregated in real-time and available in the portal.



The following dimensions are available for Geo view:

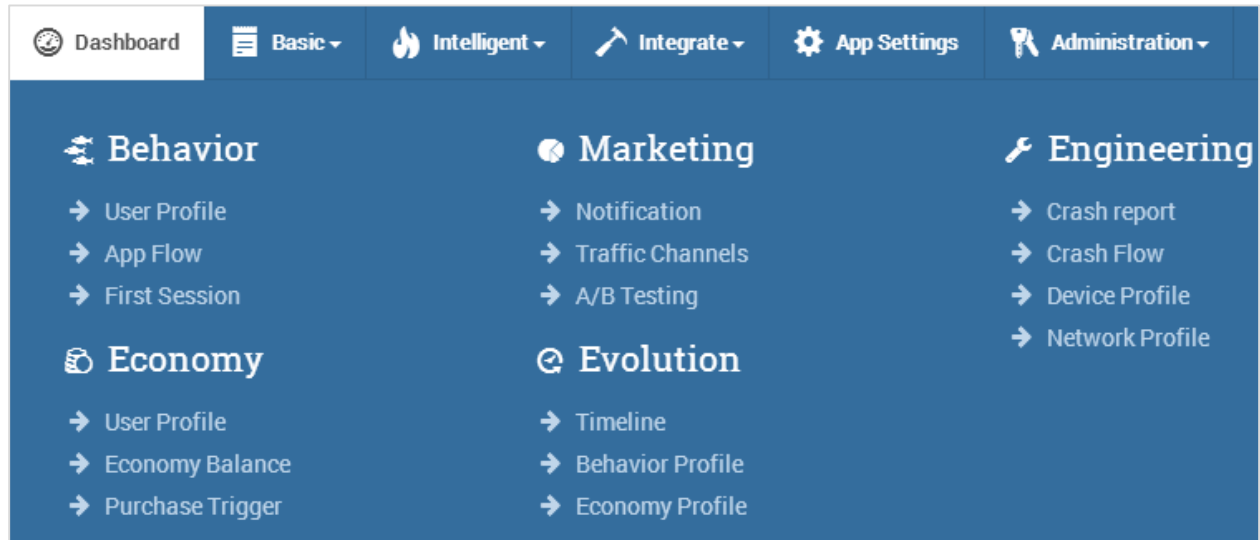
- MAU (Monthly Active Users)
- Average Session Length
- Sessions Per User
- Monthly Revenue
- ARPU (Average Revenue Per User)
- ARPPU (Average Revenue Per Paying User)

A table view with export possibilities is also provided for user's convenience:

Country	Monthly Active Users	Sessions	Total Length (seconds)	Average Session Length (seconds)	Sessions Per User	Revenue	ARPU / ARPPU
United States	610	2,693	779076 (21h 25m 36s)	289.30 (5m 49s)	4.41	\$1,273	\$2.09 / \$2.45
Japan	95	449	113971 (32h 40m 31s)	253.83 (4m 14s)	4.73	\$201	\$2.12 / \$2.72
United Kingdom	51	256	77746 (22h 36m 46s)	303.70 (5m 4s)	5.02	\$126	\$2.48 / \$2.87
Germany	49	206	68038 (19h 54m 58s)	330.28 (5m 30s)	4.20	\$105	\$2.13 / \$2.55
Korea	38	190	54739 (15h 12m 19s)	288.10 (5m 48s)	5	\$92.15	\$2.42 / \$2.79
France	34	134	40120 (11h 9m 40s)	299.40 (5m 59s)	3.94	\$52.25	\$1.54 / \$2.27
Canada	33	171	54766 (15h 13m 45s)	320.26 (5m 20s)	5.18	\$76.95	\$2.33 / \$2.65
Italy	22	79	20226 (6h 37m 6s)	256.03 (4m 16s)	3.59	\$44.65	\$2.03 / \$2.48
Russia	21	96	22549 (6h 16m 49s)	234.89 (4m 55s)	4.57	\$40.85	\$1.95 / \$2.15

INTELLIGENT ANALYTICS

Intelligent Part of the Analytical service is tailored specifically for games. It includes multiple features that are usually requested by game developers. After numerous conversations with different gaming companies features list was created and implemented for the first release.



i Marketing, Evolution and Engineering will be a part of second release of the system. Currently Behavior and Economy are the two major features that are supported.



Behavior

Behavior analysis is used by game developers to predict possible blockers in the game flow. Currently MoonGene provides three areas of such analysis:

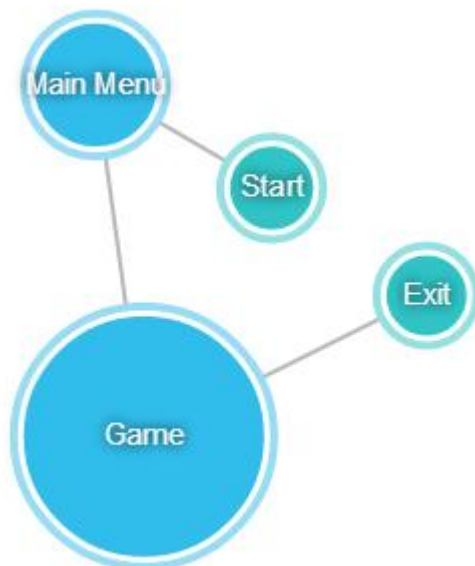
- **User Profile:** automatic detection of user's playing profile. For each version average session time and number of sessions per user is available.

Sessions / Version		
Version	Average Session Time	Average Sessions Per User
All	4.8m	4.5
1.0	2.7m	4.5
1.1	7.3m	4.6
1.2	3.7m	4.4

Additionally the most and least playing countries are detected:

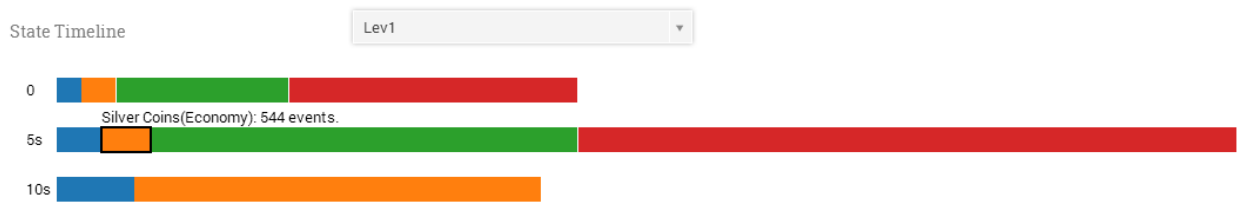
Behavior Stats	
Dimension	Value
Most Playing Country	 Jersey Daily play time: 3.0m Sessions per day: 0.4
Least Playing Country	 Hungary Daily play time: 6s Sessions per day: 0.1

- **AppFlow** is a complex analysis of the application flow. States and timelines on those states are tracked and then presented in a graphical way. Typically a flow is displayed as a graph of states and links between them. Every state has time spent and hits count stats that affect its size.



For every state multiple timelines are supported. Timeline is a special set of events that are grouped by certain parameters. For a game this can be “Level” for example. On a flow diagram clicking states reveals a list of timelines that were collected from the app.

Events are grouped by colors and displayed according to their trigger time. Such a way to present game events allows producers to analyze points in time where purchases happen or users decide to leave. There are multiple use cases and the more events are sent the richer picture of a timeline performance can be constructed.



- **First Session** is another big requirement from game developers for analytical systems. Knowing where players drop during the first session is crucial. Instead of building custom funnels, as most analytical system would recommend, we build a graph of events during the first session. Amount of people going through each path is correspondently displayed with a help of node size and line thickness. Unlike AppFlow section, in the First Session states are joined with events so that every transition is properly recorded and displayed.



It is up to a game developer to decide what events he considers important for the first session recording.

Additionally this data can be presented in user configured Funnels, though visual functionality is not yet implemented for this. If there will be a need this can be easily added and will be backward compatible with already collected data.

Economy

Economy analysis is used by game developers to detect the most and least paying user segments. The in game world balance is also used to track items accumulation during the game, items from sale campaigns accumulation and disposal, etc.

- **User Profile:** Just like in Behavior this profile is to describe typical behavior of a player but from economy point of view. Data is collected and analyzed automatically so a game developer can just see when a purchase event happens and how long it takes for a player to make it.

First Purchases		
From install till	Time	Sessions
First purchase	5.10 hours	1.82
Second purchase	16.21 hours	3.21
Third purchase	1.45 days	4.82

Most common	Item ID
First purchase	Gold Coins
Second purchase	Gold Coins
Third purchase	Gold Coins

Additionally most and least paying segments are calculated. This helps developers to understand where to invest their money and what is their best target audience for ad campaigns. Currently basic filters are supported, with developers requests list will be extended to cover needs. Every filter value is independent and a combination of those factors should yield the most profitable audience.

Most Paying Segment

Filter	Value	ARPPU
Country	United Arab Emirates	\$ 2.23
Platform	Android	\$ 2.52
OS Version	Android: 1.5	\$ 4.28
Device vendor	Panasonic	\$ 3.12
Traffic Source	Organic	\$

- Economy Balance:** It is a big challenge for a developer to balance the game. QA and focus tests allow developers to have some basic understanding of game learning curve and in-game wealth accumulation. However it is impossible to do once the game is live and gets frequent updates. Now it is possible to do with Economy Balance feature. Game developers just need to write one line of code to get this feature working.

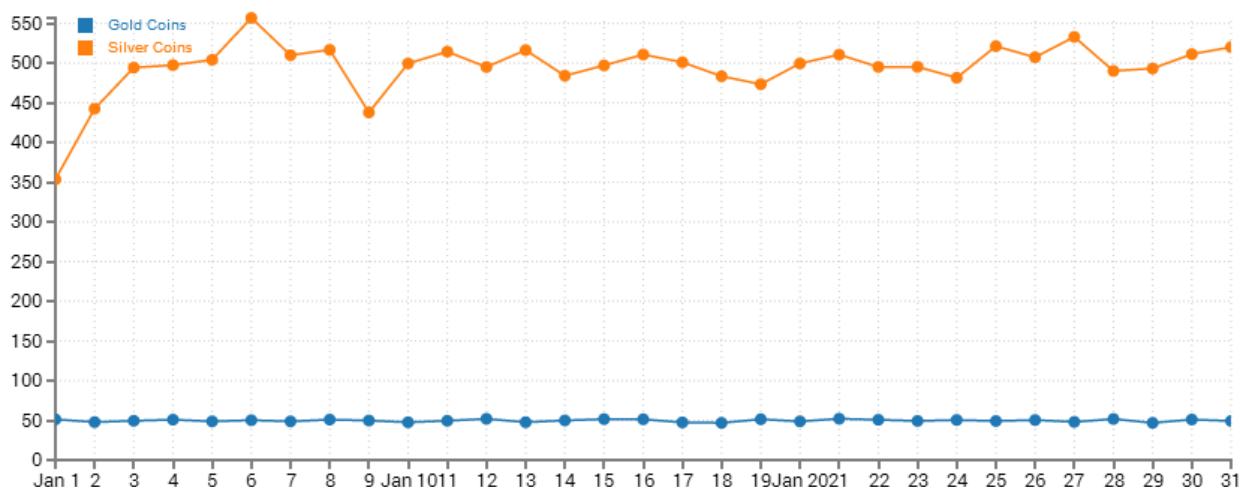
Items:

✕ Gold Coins
✕ Silver Coins

All the items that are sent by the game are accessible through a selector:

Once items are selected a graph is drawn. Even though it may seem that it stays steady most of the time it is affected a lot by sales and various campaigns. Making sure that world is not overstocked with items is now a simple task. Once items balance is back to normal values a new campaign can be done to increase revenue.

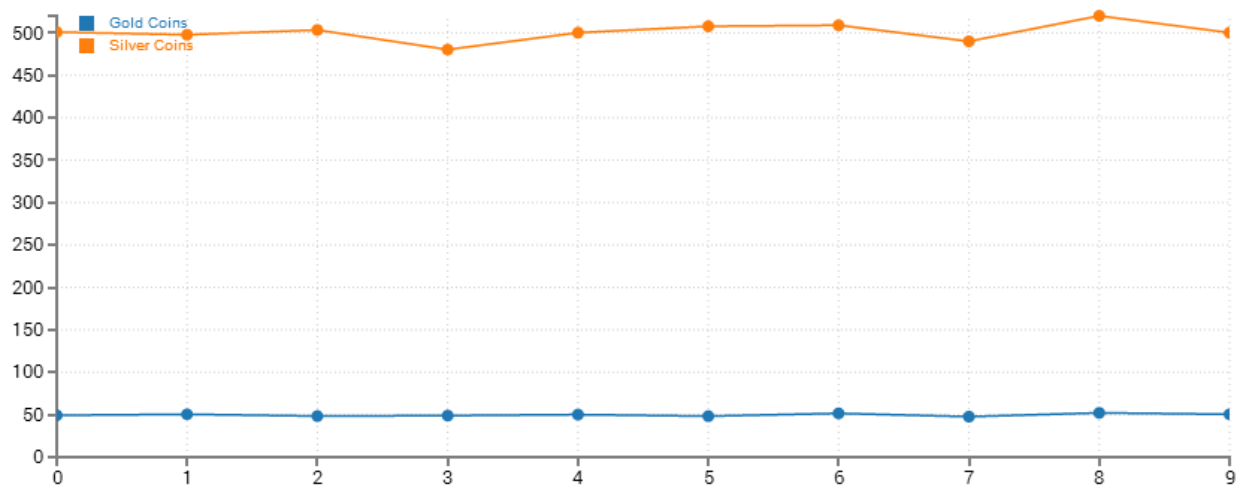
Items Balance



There is no limit on number of items to be tracked, so every time there is a special items that is available during sales, the developer can clearly see the increase and decrease with time of the stock. This makes it possible to have perfect timing for future campaigns.

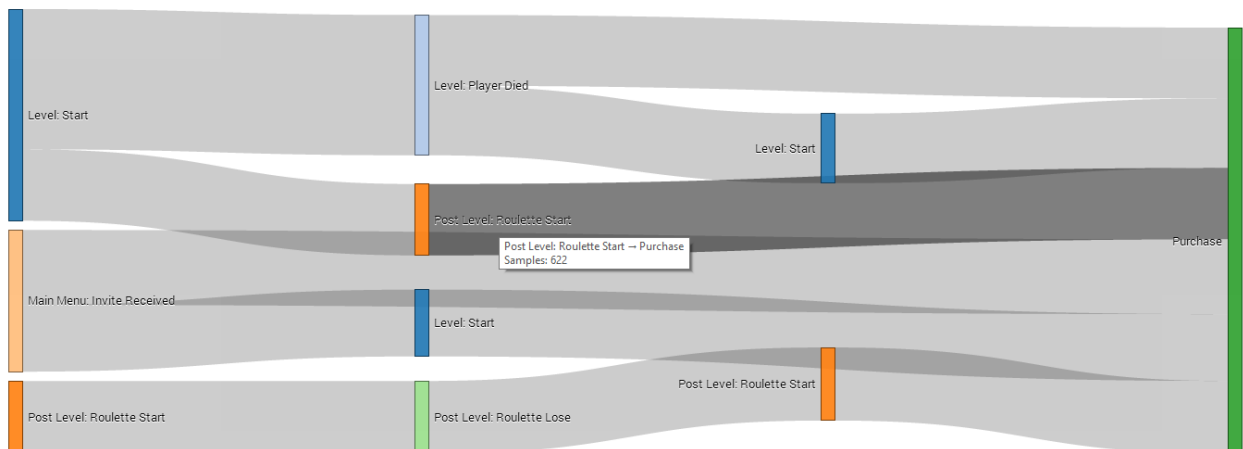
Another important aspect of economy balance is tracking accumulation of items \ wealth throughout the game. Game developer decides for himself the dimension that he wants to use for this feature, but typically this is player's experience level or game level. Making sure that there is no lack or overflow of items at various moments of the game progress is crucial for maximizing profits and keeping players engaged.

Balance progression



- **Purchase Trigger:** It is not clear what triggers purchase events, whether it's player's death or just curiosity to get further in the game. With Purchase Trigger feature, that works completely automatically, developers no longer need to worry about this. MoonGene Analytics creates a flow of pre-purchase events, which exposes the behavior of a user prior to purchases. Knowing such information allows a game developer to present various sales and offers at the right point in time. No need to bug a user when he won't buy a virtual item anyway.

Pre-Purchase Events Flow



Marketing

Marketing feature allows a game developer to link his players with ad networks. Knowing which networks provide the best audience is crucial for building a sustainable and well balanced marketing strategy.

 This feature is currently supported only as a proof of concept. Will be a part of the release 2.0.

- Notification: allows setting various notifications for different events.
- Traffic Channels: allows configuring various ad networks to link them with new players and be a part of user's behavior and economy profile calculations.
- A/B Testing: allows a game developer to do A\B testing in the game.


Evolution

Evolution helps developers to see changes that happened to their games throughout the time: evolution of user's behavior, economy from version to version.

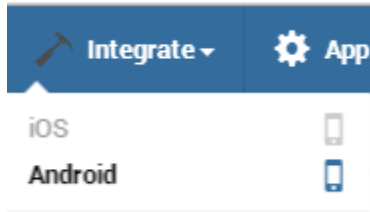
 This feature is currently supported only as a proof of concept. Will be a part of the release 2.0.

Engineering

Engineering section is aimed at helping developers to know crashes and various glitches that happened in a mobile app. Just like economy and custom user events engineering events are visible on timelines, first session graph, etc.

 This feature is currently supported only as a proof of concept. Will be a part of the release 2.0.

INTEGRATION



Integration is very straight-forward and user friendly. There is a built-in support for integration in the portal. Selecting the right OS will generate code snippets for game developers to avoid reading the documentation.

Integration is done with a help of Android and iOS SDKs. There are just a few functions that developers need to call in order to get the functionality working. Basic features are available after implementation of just the initialization call. For

Android it looks like:

```
mAnalytics = Analytics.getInstance(this, "521422cd883d3584960164d8", "L2SPRX");
```

Where app ID and token are passed.

If a game developer needs other features, he will just need to add a few more calls at various locations of the game code. For an experienced developer integration should take no longer than half an hour.

PERFORMANCE

Analytical solution was designed to be highly concurrent and scalable. Every component can be scaled independently and there are no bottlenecks.

Solution is cloud ready and can be deployed on any public cloud provider infrastructure.

Under load, full stack on a SINGLE machine: CPU: Intel i5 4200U, RAM: 8GB, SSD has a throughput of more than 1000 events per second or in monthly equivalent 2.5 billion events.