

GameAnalytics



Your data tracking plan

# Designing your events

A Step-by-Step Guide for GameAnalytics

## Create a tracking plan

# Tracking data is never simple.

It takes plenty of effort to make sure you have valuable data downstream. If you want to get the most out of your data within GameAnalytics, you need a solid plan for what events, fields and dimensions you'll be tracking.

Creating this plan builds a bridge between the people analyzing the data and the ones collecting it. So make sure you involve both sides when going through these steps. It's also helpful to determine an owner to make sure the event structure, implementation and analysis doesn't get off track.

A tracking plan should clearly define the specific data points you want to collect (and how). It serves three purposes:

1. It helps you avoid collecting redundant or inaccurate data.
2. It'll act as a roadmap for how you'll put your processes in place.
3. It'll be a document to make sure you stay consistent if you need to add more tracking or if your team changes.

The more precisely you document your process, the more confidently people can analyze the data.



Your tracking plan will make sure everyone involved in the data analysis process is on the same page and working towards the same goals.

## Contents

# What's inside?

Five steps to get you from your questions to some answers.

- 1 Brainstorm your burning questions
- 2 Create your events
- 3 Define your dimensions and attributes
- 4 Plug in your game
- 5 Keep updating your plan

# Brainstorm your burning questions

Get everyone in a room and ask them: What questions do we have? Start by thinking about what you need to know about your players and why. It's also helpful to break down those burning questions into sub-questions.



Figuring out what questions you have helps you see what you need to measure (your metrics) and how you'll measure them (your events).

For example, you might come up with:

- What do I need to know about my players?
- Why are we collecting this data?
- What are we working towards?
- How long does it take to finish the tutorial?
- What stops people from buying our in-app purchases?
- How much are players willing to pay for X?
- What are the most (or least) used by active players: levels, items or guilds?
- What IAP packages convert free users into paying users?
- Do players see the weapon stashed in the corner?

# Create your events

Events are distinct actions that a player can perform in your game.

Struggling to start? We've already looked at the most common burning questions and created our [predefined event types](#).

If you use these, we'll automatically calculate the relevant key performance indicators (or KPIs) and add them to your dashboard.

Start with our predefined events, then match whatever you can to your burning questions. These events include:

- **Session start and end:** Triggers automatically.
- **Business:** Track in-app purchases and validates the receipts on our servers.
- **Resource:** Track virtual currencies – like gems or lives – and how you manage them.
- **Progression:** See how players progress through levels with Start, Fail and Complete.
- **Error:** Submit exception stack traces or custom error messages.
- **Ads:** See how players interact with ads in your game and monitor how they perform.
- **Impression:** Get impression data from ad networks.

## Step two



# Create your events

You can also [design your own events](#). In these custom events, you can have three event dimensions and one value. We recommend using the **"Category > Sub Category > Outcome or Action"** framework. This is a standard way of organizing data that focuses on the two main components: the object (or item) and the action taken on that object.

Only track events that actually answer your burning questions and consider keeping your designed events broad (but not necessarily generic) with clarifying information in the dimensions.



As a rule of thumb, we recommend keeping custom events to a maximum of 50 unique events. Otherwise, your event plan will become difficult to monitor, fix, and analyze.

Here are some examples:

Category: Weapon  
Sub Category: [Weapon Name]  
Action: Fired, Reloaded, Swapped

Category: Character  
Sub Category: [Character Name]  
Action: Created, Leveled up, Equipped item, Used ability

Category: Vehicle  
Sub Category: [Vehicle Name]  
Action: Accelerated, Braked, Steered, Used power-up

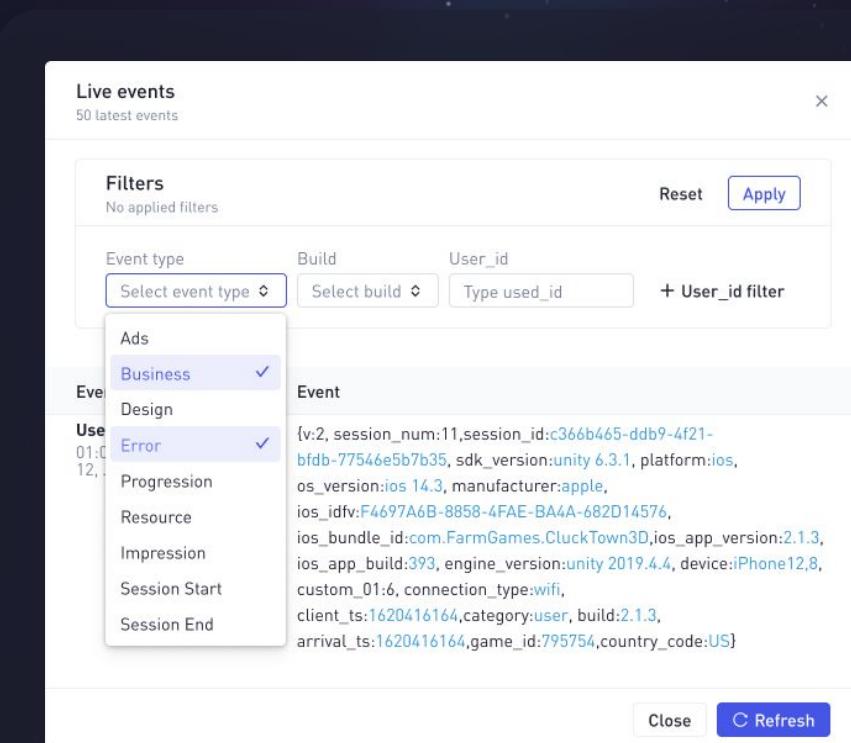
Category: Building  
Sub Category: [Building Name]  
Action: Built, Upgraded, Moved, Attacked

# Define your dimensions and attributes

Dimensions describe your event and the context around how it was triggered. You can have both global dimensions (which go across event types) or dimensions that only relate to a specific event type.

In GameAnalytics, **you can use three custom fields** to get this event context. These are available for analysis within our webtool.

**You also have 50 custom fields** for these descriptive dimensions. These are available for custom analysis with [our flexible data tools](#).



# Plug in your game

GameAnalytics offers a multitude of SDK's to integrate with your project:



Each of the SDKs have their own quirks when you're integrating them, so visit the [individual guide](#) for whichever one you'd like to use.

But generally, you'll need to:

1. [Create your game](#) in the GameAnalytics web-tool.
2. [Visit the documentation](#) for the SDK you want to integrate.
3. Download the files for the SDK to run (either from github or any available marketplace) and include them in your project.
4. Set up your Game Key and Secret Key.
5. Initialize the SDK.
6. Start sending events. (Our integration guides give examples of syntax for making events for each of the SDKs. [You can also learn more here.](#))

If you need a helping hand, [get in touch with our support team](#).

## Step five



# Keep updating your plan

You should always be updating your tracking plan, particularly when you add new features to your game. Reconsider your burning questions and go through the same process to determine your events and dimensions.



**One last thing**

# Watch out for common pitfalls

Before you get cracking, here are some common pitfalls we've spotted.

If you have any questions visit our documentation centre, or contact support. We're happy to help.

[Visit our docs](#)

[Contact support](#)

**Keep an eye out for:**

- Creating events that trigger twice, leading to duplicate (and misleading) data.
- Using multiple events for the same action.
- Using event names or parameter abbreviations that nobody understands.
- Putting your events in the wrong order.
- Not validating your schema properly.
- Going too granular with your event structure
- Adding events with no business or product purpose.
- Missing or using wrong identifiers.
- Not testing or validating your events on staging.
- Using the wrong parameter types.
- Changing parameter enums.