



Draw.IO

Packages to support project development and live operation

by K.Kostenkov for Voodoo

Table of contents

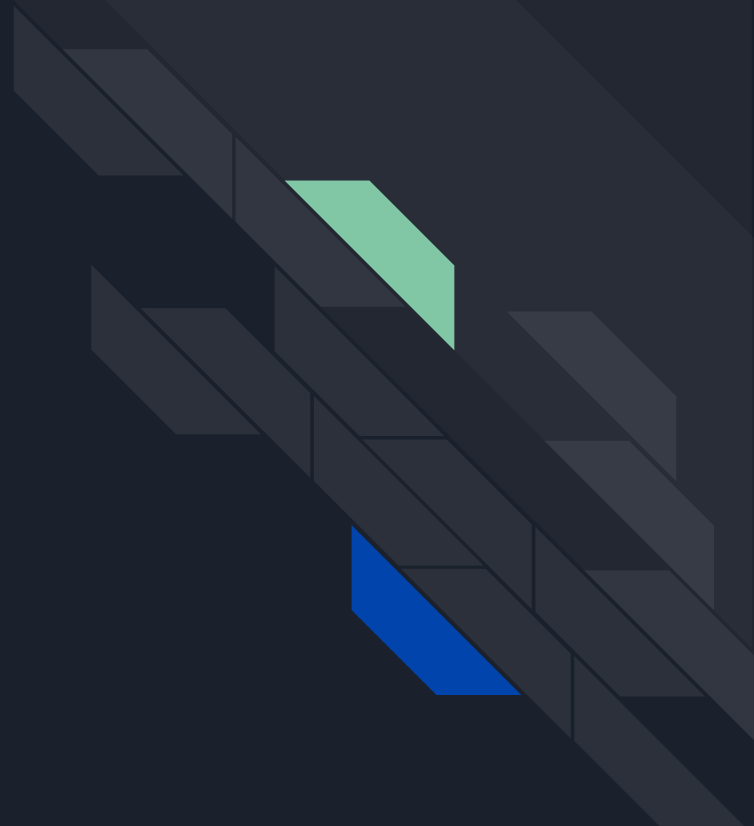
[Assignment](#)

[Philosophy and approach](#)

[Packages - Business and product](#)

[Packages - Stability](#)

[Packages - Development](#)





Assignment

Provide a list of external packages you will need to make Draw.IO game online / live operable

Interpreted as *“What might we need for the medium term development and operations plan?”*

Because for the basic functionality of “synchronising saves and receiving basic content for live ops” no external packages are needed.



Philosophy and approach

General approach is the less extra packages we bring on-board - the less unpredictable and lighter the project will be. However some long-lasting packages have proven to be stable and supported by their maintainers.

Some packages provided by vendors that leave us no option (Facebook SDK)

Packages mentioned here are not necessarily should end up in the project. Some of them are here just to show that I am aware of their presence and functionality.

If one vendor provides packages for several purposes - the preference goes to that vendor.





Packages - Business and Product

Analytics

[Google Analytics](#) (Firebase group)

A/B

[Firebase Remote Config](#) (remote config comes as a bonus)

LiveOps

Depending on the product roadmap and business needs. I prefer self-hosted server + CDN.

Localization

[I2 Localization](#) (if we don't mind per-seat pricing model) or [Unity localization](#)

Monetization (Ads/IAP?)

[Unity purchasing](#)
[AppLovin MaxSDK](#)



Packages - Stability

Crashes and prod monitoring

[Firestore Crashlytics](#) + breadcrumbs

[Unity Crash and Exception Reporting](#)

Logs

[Unity-Logs-Viewer](#)

[Serilog](#) + [Unity wrapper](#) Writes to file, structured logs, back end compatible

User reporting

[Unity User reporting](#)

User support

[Helpshift](#)



Packages - Development

Tweening

[DOTween](#)

Dependency injection

[Zenject](#)

Cheat menu

[DebugUISystem](#)

Asset management

[Addressables](#)

Async

[UniTask](#)

And a ton more including different UI support ones , asset usage finders, Odin inspector, R3, Fmod, Protobuf etc. Or even consider moving the project onto the ECS in the current gameplay won't make it in terms of performance. All depending on the situation and the necessity