

Getting Started with Everyplay



Everyplay allows you to monetize and engage your users easily with video. The integration is simple, and getting started with the product takes only a few minutes.

About Everyplay Replays

Everyplay Replays allows your users to share their best gaming moments directly from your game with their friends and other players of the game.

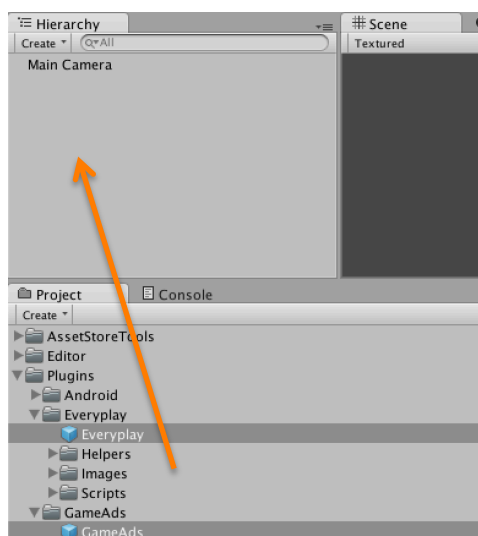
About Everyplay GameAds

Everyplay GameAds help you monetize your users through opt-in video ads for games that reward users for watching or interstitial video ads that can be muted and skipped. The best fill rates & eCPMs come from video advertising. Players love discovering new games to play alongside existing games.

Getting Started

Note: if you are getting an error “Could not create texture”, make sure your Build platform is set to iOS or Android.

To start using the products, start by dragging the GameAds and Everyplay prefabs from the Assets directory to your scene hierarchy.



After the prefabs have been added to your scene, click on each prefab to configure them. For configuring the prefabs, you need *game credentials* to use them. You can obtain them from our admin panels:

For Everyplay GameAds:

You need the *Game ID*, which is available after creating an account at <http://my.applifier.com/>

For Everyplay Replays

You need the *Client ID*, *Client Secret*, and *Redirect URI*, which are available after creating an account at <http://developers.everyplay.com/>

Device Support

Everyplay Replays

iOS

iOS version 5.0 or newer is required. Everyplay Replays runs the best on dual-core iOS devices (iPhone 4S or newer, iPad 2), but will run on older devices as well – performance varies by game and the best approach is to test this yourself and only disable Everyplay for certain devices if you do see a performance issue on low end devices.

Please note that when testing Everyplay, it will **not** run on the simulator.

Android

Android version 4.1 or newer is required. Everyplay generally works on most modern Android devices, but due to the device fragmentation, we cannot guarantee it to run on all devices. At first initialization, the SDK will collect information about the device being used and will poll from Everyplay servers the confirmation if that hardware can run Everyplay recording or not.

Everyplay GameAds

iOS

iOS version 5.0 or newer is required.

Android

Android version 2.2 or newer is required.

Controlling Gameplay Recording

Gameplay recording is controlled through methods provided by the *Everyplay.SharedInstance* object. The methods to control the playback are:

- `StartRecording()` – This will start the capture of the gameplay

- `PauseRecording()` – This will pause the capture of the gameplay
- `ResumeRecording()` – This will resume a paused recording
- `StopRecording()` – This will stop the current video capture, any subsequent calls to `StartRecording` will overwrite the previously generated video file.

Once the recording has been completed, you can prompt the user to share the recording by calling `ShowSharingModal()`. Alternatively, if you want the users to first have the ability to watch the replay prior to sharing it, you can call `PlayLastRecording()`. However, we suggest the use of the sharing modal, which also allows the user to re-watch the replay, but does not require them to view the video first.

That's it, it's this simple to get screen sharing accomplished!

For a full set of available functions, see the documentation at <https://developers.everyplay.com/doc/Everyplay-integration-to-Unity3d-game>

Controlling GameAds

Once GameAds is properly configured in your game, it will automatically check for ad inventory on startup. What you need to do is to create the proper delegates in a script to get notified of events in the GameAds SDK. These events notify you for example when campaigns have been loaded from the server and GameAds is ready to be shown to the user.

The two most important delegate functions are:

- `EveryplayGameAdsCampaignsAvailable()`; – This delegate is called when GameAds has successfully loaded campaigns for the current user. Prior to receiving this call, it's not advised to show the user anything pertaining to offers.
- `EveryplayGameAdsVideoCompleted(string rewardItemKey)`; – This delegate is called when the user has completed watching a video on GameAds, and should be rewarded an item. The `rewardItemKey` passed as a parameter is the key for the reward item that you should grant, that has been configured in the [GameAds Admin Panel](#)

The delegate methods can be registered via the `EveryplayGameAds.setVideoCompletedDelegate` and `EveryplayGameAds.setCampaignsAvailableDelegate` methods.

Full list of delegate methods is available in the online documentation at <https://developers.everyplay.com/doc/GameAds-Integration-Guide-for-Unity>

Showing GameAds Ads

When you've received information that there are ads available, you can then show GameAds to the user at the time of your choosing by calling `EveryplayGameAds.showImpact()`. Prior to calling this, it's always recommended that you check that everything is in order for GameAds to be shown using the `canShowImpact` and `canShowCampaigns` methods. This is especially useful if you allow the user to watch multiple videos in one session.

Example:

```
if (EveryplayGameAds.canShowCampaigns() &&  
    EveryplayGameAds.canShowImpact()) {  
    EveryplayGameAds.showImpact();  
}
```

For full integration instructions, refer to the online documentation at:
<https://developers.everyplay.com/doc/GameAds-Integration-Guide-for-Unity>

Controlling GameAds

If you are having issues with the products, feel free to send an email to our support at **support@everyplay.com**