

FGL SDK for HTML5: Official Phaser Plugin

The official FGL Phaser Plugin makes it easier to use all the FGL SDK features in a Phaser project, in particular providing easy bindings for displaying sponsor branding, cross promotion and in-app purchase "unlock" buttons using native Phaser constructs.

Version 1.6 August 7th 2014

Note

This document offers Phaser Plugin specific implementation details as well as an overview of the methods and features available to the FGL Phaser plugin. You can also refer to the standard FGL HTML5 SDK documentation as it is fully compatible with Phaser, and many of the methods will be exactly the same as those documented here.

Getting Started

1. Start by dropping the included *FGLPhaser.js* file into the same directory as your game's *index.html*.
2. Add the external script reference to the FGL SDK to your *index.html*, after the `<script>` tag for Phaser. It should look like this:

```
<script src="js/phaser.min.js"></script>  
<script src="https://sites.mpstatic.com/html5/sdks/phaser/fgl.js"></script>
```

3. Add the Phaser Plugin `<script>` tag after the FGL SDK script reference. It should look like this:

```
<script src="js/phaser.min.js"></script>  
<script src="https://sites.mpstatic.com/html5/sdks/phaser/fgl.js"></script>  
<script src="FGLPhaser.js"></script>
```

4. You can now begin to add SDK functionality to your game's code. Note that `game.fgl` is the root FGL SDK object that will contain access to all of the required methods.

Displaying Ads

You can call a function to display a takeover ad at any time. The advert will be displayed on top of your game and will display a close button for the user.

We can control which ad networks run on your game, and we are constantly optimizing the inventory on your behalf to get the best possible return. The ads are player-friendly and can be immediately dismissed.

Call the following function when appropriate, for example when the user reaches the main menu, or between levels on your game:

```
game.fgl.showAd( );
```

Cross Promotion

To take advantage of our powerful cross-promotion system, you should implement a *"More Games"* button on your main menu.

When tapped, this button will display an overlay which will cross promote to other games in our network. By doing this, your game will obtain significant promotion at launch, when it needs it most.

To display a 'More Games' button, use the special `fglMoreGamesButton` type, similarly to how you would use a regular Phaser button:

```
fglMoreGamesButton(x, y, key, overFrame, outFrame, downFrame, upFrame, [group]);
```

```
// For example:
```

```
this.moreGamesButton = this.add.fglMoreGamesButton(240, 470, 'moreGamesButton', 'over', 'out', 'down');
```

(Continued on next page)

Cross Promotion (cont.)

Not all target platforms allow cross promotion, so the `fglMoreGamesButton` will automatically hide itself when necessary. If you would like to detect this case (to rearrange your other menu items when the more games button is hidden, for example):

```
if(game.fgl.crossPromotionEnabled){  
    // Cross promotion is enabled  
}
```

You can test your implementation by using the Test Session Options on the intro screen of the FGL SDK:

Test Session Options:

- ☒ Disable cross promotion
- ☐ Disable in app upgrade
- ☐ Game is premium mode
- ☐ Show sponsor branding

Premium Model

To most effectively monetize your game on the various platforms, we offer a simple set of features to allow us to sell a full or premium version of your game.

The system works by allowing you to check if the game is running in premium or free mode:

```
if(game.fgl.isPremium()){  
    // Allow premium content  
}
```

You should use this feature to restrict content or features when your game is running in free mode. This will help your game to upsell to premium. You do not need to disable/hide ads yourself as the SDK will automatically hide ads for you when the game is running in premium mode.

On many platforms your game will be unlockable to premium by a simple in-app purchase. You should take advantage of this by implementing an unlock button in various places within your game:

```
fglUpgradeButton(x, y, key, successCallback, failCallback, overFrame,  
outFrame, downFrame, upFrame, [group])
```

```
// For example:
```

```
this.upgradeButton = this.add.fglUpgradeButton(240, 550, 'upgradeButton',  
this.unlockSuccess, this.unlockFailed, 'over', 'out', 'down');
```

This button will automatically hide itself on markets where in-app purchases are not allowed. By using this method we can still sell a premium version of your game on markets which don't support in-app purchases, as well as distributing a free ad-supported version of your game.

(Continued on next page)

Premium Model (cont.)

If you would like to detect whether in-app purchases are allowed:

```
if(game.fgl.unlockEnabled){  
    // In-app unlock is enabled  
}
```

When tapped/clicked, your unlock button will take the player through the in-app purchase and eventually will trigger the callback functions you pass to the `fglUpgradeButton` constructor.

```
// On Success:  
var unlockSuccess = function() {  
    // Success! Unlock your game's premium content here.  
};  
  
// On Failure:  
var unlockFailed = function() {  
    // Make sure premium content is locked here  
    // Do not notify the user, that is handled by the SDK  
};  
  
this.upgradeButton = this.add.fglUpgradeButton(240, 550, 'upgradeButton',  
this.unlockSuccess, this.unlockFailed, 'over', 'out', 'down');
```

You can test your implementation by using the Test Session Options on the intro screen of the FGL SDK:

Test Session Options:

- ☐ Disable cross promotion
- ☒ Disable in app upgrade
- ☒ Game is premium mode
- ☐ Show sponsor branding

Publisher Branding

To further monetize your game, sometimes we may have the option to sell a "sponsorship branded" version of your game to a publisher, much like on the FGL Game Shop. If you would like your game to be able to make money in this way, then you must implement our simple publisher branding functionality.

You should display the image on your main menu, and in any other menus where it make sense. Publishers will be able to see where their branding will show up in your game before they purchase, so placing the branding somewhere non-prominent such as only in the credits menu may result in fewer sales.

To show publisher branding in your game, add the `fglPublisherBranding` button object, similarly to how you would add a regular Phaser button:

```
fglPublisherBranding(x, y, width, [group])
```

```
// Add some publisher branding
this.brandingButton = this.add.fglPublisherBranding(20, 470, 250);
```

The publisher branding will automatically hide in cases where publisher branding should not be displayed. You can detect this case if you need to:

```
if(game.fgl.brandingEnabled){
    // Publisher branding is enabled
}
```

(Continued on next page)

Publisher Branding (cont.)

When you implement publisher branding, you should see a test branding placeholder. This will help you to properly position the logo on your menu screens.

The image should always be 250px by 100px in size and will have a transparent background:



Example sponsor logo image

You can test your implementation by checking the "Show sponsor branding" option from the Test Session Options on the intro screen of the FGL SDK:

Test Session Options:

- ☐ Disable cross promotion
- ☐ Disable in app upgrade
- ☐ Game is premium mode
- ☒ Show sponsor branding

Scoreboards

Our generic leaderboard functions allow you to implement one simple set of leaderboard calls - we can then do all the work to make your HTML5 game use the best leaderboard system for each platform. This gives your game an advantage as many platforms will give additional promotion or exposure to games which integrate with their proprietary systems.

The following functions are available:

```
// Submit the score in response to a user pressing a submit button:  
game.fgl.submitScore(score);
```

```
// Show the scoreboard UI. This happens automatically after submitScore:  
game.fgl.displayScoreboard();
```

See the FGL HTML5 SDK documentation for further details.

Achievements

Like leaderboards, achievements can help boost the popularity and promotion of your game by allowing our system to tie in to the best solutions for each platform.

The following functions are available:

```
// Grant a named achievement to the player:  
game.fgl.grantAchievement(name);  
  
// Check if the player has previously been granted an achievement:  
boolean game.fgl.hasAchievement(name);
```

See the FGL HTML5 SDK documentation for further details.