Rewarded Ads Integration Guide

SDK Version: 3.316 (v3)

This guide explains how to integrate and use **Rewarded Ads** using the Adinmo SDK. Rewarded Ads allow developers to grant in-game rewards to players in exchange for viewing an advertisement. This tutorial is intended for developers and test-case engineers integrating the feature into their own projects.

Prerequisites

Before you begin, ensure the following:

- You are using **Adinmo SDK v3.316 (v3)**.
- You have access to the **Adinmo Portal**.
- Your project already has Adinmo SDK successfully initialized and running.
- You are familiar with creating placements using the Adinmo SDK.

Step 1: Create a Rewarded Placement on the Adinmo Portal

1. Log in to the **Adinmo Portal**.
2. Create a **new placement** specifically configured for **Rewarded Ads**.
3. Take note of the following details:
    - **Placement Key**
    - **Game Key**
    - **Placement Name**

These identifiers are required to correctly configure the placement in your game.

Step 2: Share Placement Details with Adinmo

Send the following information to your **Adinmo point of contact**:

- Game Key
- Placement Key
- Placement Name

This step is required to ensure the rewarded placement is enabled and configured correctly on Adinmo's backend.

Step 3: Create the Placement in Your Game

1. In your game project, create a new placement by following the **"Create Your Placement"** tutorial provided in the SDK documentation.
2. If you are creating an **in-environment placement**, ensure that:
    - The **Mesh** property is set to **Quad16x9**.
3. Assign the **Placement Key** (obtained from the portal) to this placement.

At this stage, the rewarded placement should be visually and logically present in your scene.

Step 4: Subscribe to the Reward Event in Code

Rewarded Ads rely on event-based callbacks to notify your game when a reward should be granted.

1. Choose a **GameObject that persists for the lifetime of the game session** (e.g., a GameManager or similar singleton).

2. Add a script to this GameObject and subscribe to the rewarded event in `OnEnable`.

Event Subscription

```
private void OnEnable()
{
    AdinmoManager.OnGiveReward += HandleRewardGranted;
}
```

- `AdinmoManager.OnGiveReward` is triggered when a rewarded ad is successfully completed.
- The event passes an **integer parameter**, representing the reward amount.

> **Note:**
> For the current test implementation, the reward value is returned as an integer without requiring reward configuration on the portal. This gives developers full flexibility to define how the reward is interpreted and applied within the game. Portal-side reward configuration is not yet available.

Step 5: Unsubscribe from the Event

Always unsubscribe from the event in `OnDisable` to prevent unintended behavior.

```
private void OnDisable()
{
    AdinmoManager.OnGiveReward -= HandleRewardGranted;
}
```

Why This Is Important

Failing to unsubscribe from events can lead to:

- **Memory leaks**
- **Duplicate reward grants**

- **Null reference exceptions** when destroyed objects still receive callbacks
- **Unexpected game state changes** if multiple handlers are invoked unintentionally

Following proper subscription lifecycle management ensures stability and predictable behavior.

Step 6: Handle the Reward Logic

Implement a handler method to define how your game responds when the reward is granted.

```
private void HandleRewardGranted(int rewardAmount)
{
    // Apply reward logic here
}
```

Example Use Case

In the **Adinmo_Rewarded** demo scene:

- Successfully watching a rewarded ad grants the player **14 crops**.
- Checkout the CoinGranter.cs script for reference
- The `AddCoins` method adds coins to the player's gameplay state **Coins** by adding the specified count received from the ad.

You are free to tailor this logic to match your game design, whether that involves currency, power-ups, consumables, or progression boosts.

Summary

By following these steps, you can successfully integrate Rewarded Ads into your project using Adinmo SDK v3.316. The event-driven reward system provides flexibility while maintaining a clean separation between ad logic and game logic.

For further assistance or advanced configurations, please contact your Adinmo representative.