Sonat Game Studio
Bamboo Airway 265 Cầu giấy

Sonat tracking

27th December 2022

OVERVIEW

Sử dụng cho các đối tác thêm tracking cho các dự án hợp tác.

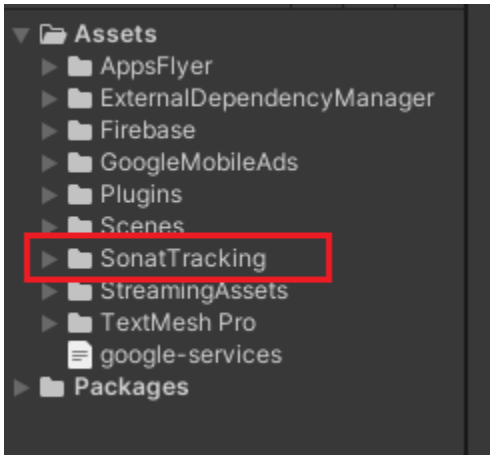
SPECIFICATIONS

1. Các plugins yêu cầu :
    - Firebase Analytics, Firebase RemoteConfig, Firebase Installation, Firebase Crashlytics
    - Appflyer https://github.com/AppsFlyerSDK/appsflyer-unity-plugin
    - Appflyer adrevenue connector https://github.com/AppsFlyerSDK/appsflyer-unity-adrevenue-generic-connector
    - Google Mobile Ads Unity (+ mediation network)
    - Facebook SDK

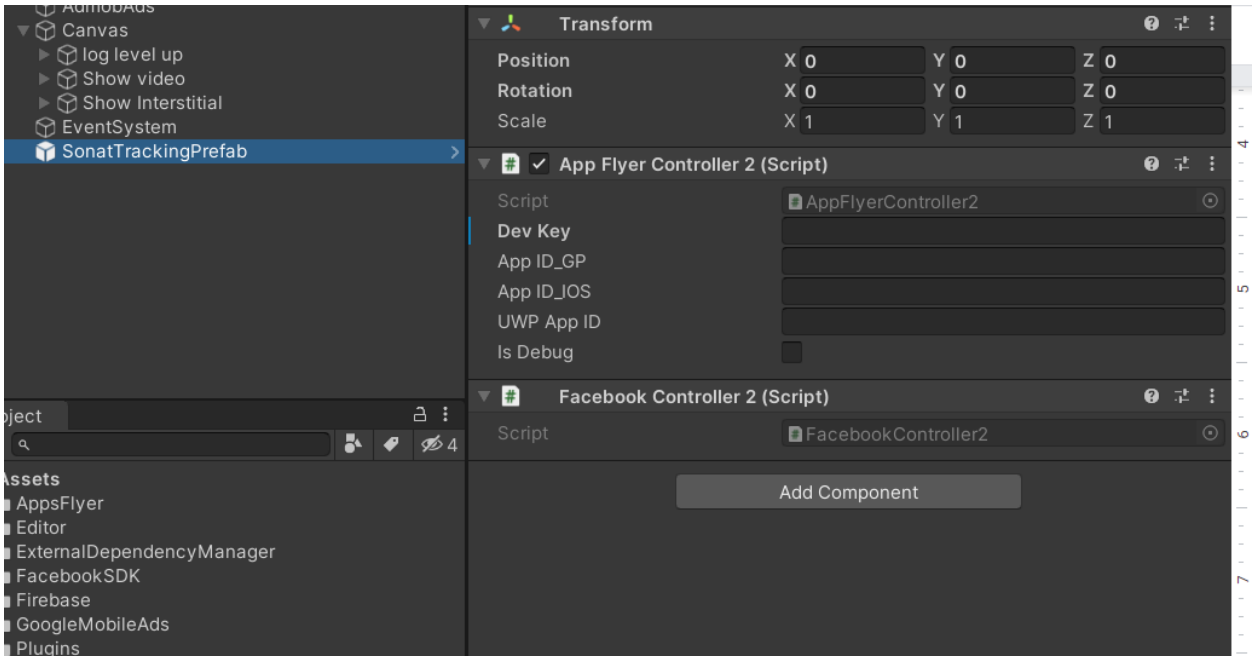
MILESTONES

Tích hợp:

1. **Thêm SonatTracking vào project**



2. **Thêm appflyer controller vào scene, sử dụng dev key của sonat, với game ios vui lòng thêm AppID_IOS**

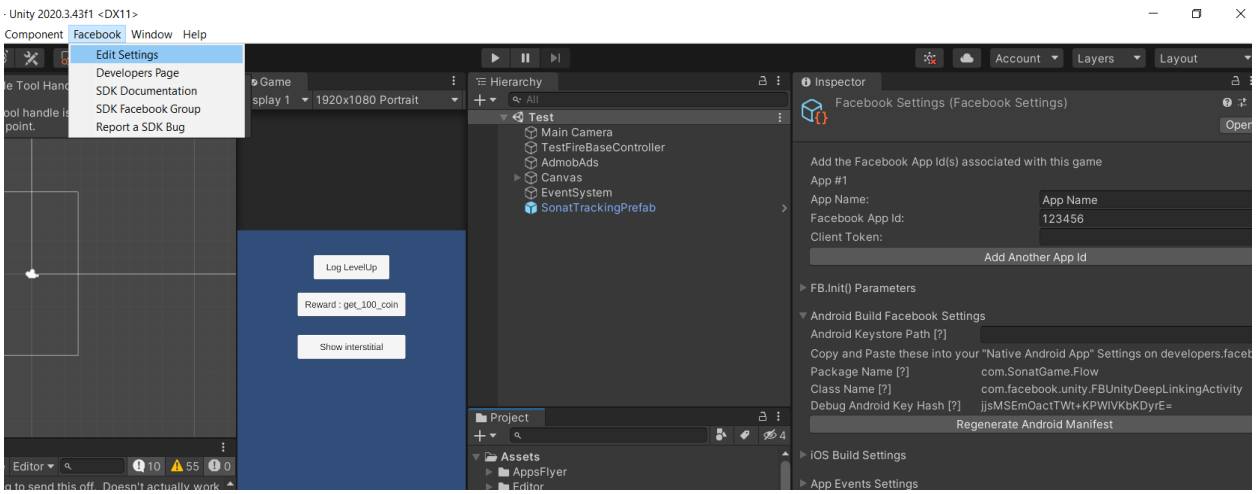


Chú ý nếu để ở scene loading vui lòng nhớ Dont Destroy Onload

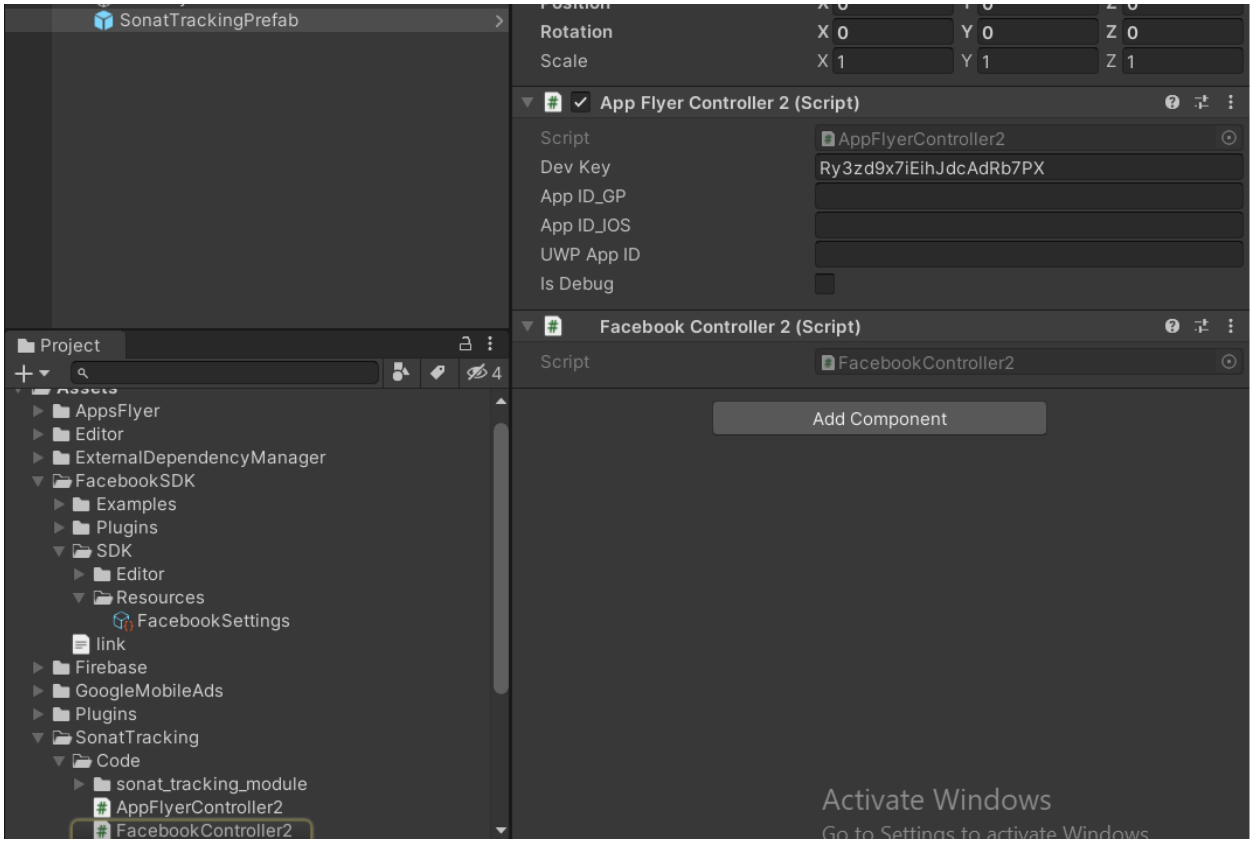
3. Thêm facebook sdk phục vụ quảng cáo facebook

Import facebook sd plugin và setting facebook trong Window/Facebook/Edit Settings

Thêm app id



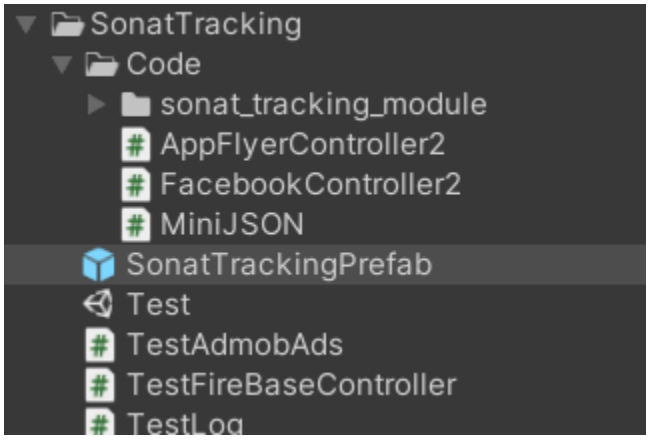
Thêm script Facebook Controller2 vào scene



4. **Nếu dự án đã sử dụng Firebase sẵn (nếu không có thể sử dụng theo mẫu** TestFirebaseController.cs), set thuộc tính SonatAnalyticTracker.FirebaseReady sau khi khởi tạo xong Firebase

```
Event function
protected  void Start()
{
    FirebaseApp.CheckAndFixDependenciesAsync().ContinueWithOnMainThread( continuation: task =>
    {
        dependencyStatus = task.Result;
        Debug.Log( message: "CheckAndFixDependenciesAsync " + dependencyStatus);
        if (dependencyStatus == DependencyStatus.Available)
        {
            SonatAnalyticTracker.FirebaseReady = true;
            InitializeFirebase();
        }
        else
        {
            Debug.LogError( message: "Could not resolve all Firebase dependencies: " + dependencyStatus);
        }
    });
}
```

Note : có thể sử dụng luôn SonatTrackingPrefab kéo vào scene khởi tạo



Tracking Revenue Admob

Nếu bạn đang sử dụng script xử lý admob khác vui lòng tích hợp thêm tracking như sau

1. **Thêm hàm HandlerPaidAdEvent vào script**

```
4 usages  ••• More
private void HandleAdPaidEvent(object sender, AdValueEventArgs args, AdType adType, string adapter,
    AdsPlatform platform)
{
    MobileAdsEventExecutor.ExecuteInUpdate(() =>
    {
        AdValue adValue = args.AdValue;
        SonatAnalyticTracker.LogRevenue(platform, adapter,
            revenue: adValue.Value / 1000000f, adValue.Precision.ToString(), adType.ToString(), adValue.CurrencyCode);
    });
}
```

Chú ý : không cần dùng hàm get network name nữa, adValue.Value phải của admob phải chia cho 1000000f

2. **Lưu lại các adapter khi load quảng cáo thành công**

```
private string bannerAdapter;
private string interstitialAdapter;
private string rewardedAdapter;
private string openAdsAdapter;
```

```
1 usage
private void HandleBannerAdLoaded(object sender, EventArgs args)
{
    MobileAdsEventExecutor.ExecuteInUpdate(() =>
    {
        _bannerState = AdsState.Loaded;
        bannerAdapter = _bannerView.GetResponseInfo().GetMediationAdapterClassName();
    });
}
```

Tương tự với các định dạng quảng cáo còn lại

3. **Thêm Callback gọi đến HandlerPaidAdEvent khi request quảng cáo**

```
_bannerView.OnPaidEvent += (sender, args) =>
{
    HandleAdPaidEvent(sender, args, AdType.banner, bannerAdapter, AdsPlatform.googleadmob);
};

// Create an empty ad request.
AdRequest request = new AdRequest.Builder().Build();
// Load the banner with the request.
_bannerView.LoadAd(request);
```

```
// Called when the ad click caused the user to leave the application.
_interstitial.OnPaidEvent += (sender, args) =>
{
    HandleAdPaidEvent(sender, args, AdType.interstital, interstitialAdapter, AdsPlatform.googleadmob);
};

// Create an empty ad request.
AdRequest request = new AdRequest.Builder().Build();
// Load the interstitial with the request.
_interstitial.LoadAd(request);
```

```
_rewardedAd.OnPaidEvent += (sender, args) =>
{
    HandleAdPaidEvent(sender, args, AdType.rewarded_video, rewardedAdapter, AdsPlatform.googleadmob);
};
_rewardedAd.OnAdOpening += HandleVideoAdOpeningEvent;
AdRequest request = new AdRequest.Builder()
    .Build();

_rewardedAd.LoadAd(request);
```

Chú ý không đc nhầm tham số, thêm khảo thêm TestAdmobAds.cs hoặc tự custom thêm để sử dụng.

Tracking Revenue MaxLovin

```
public void RequestRewardedAd(bool register = false)
{
    Debug.Log( message: "RequestRewardedAd");
    UIDebugLog.Log("duong : RequestRewardedAd");
    _videoState = AdsState.Requesting;

      if(_requestReward != null)
          StopCoroutine(_requestReward);
    MaxSdk.LoadRewardedAd(VideoId);

    if (register)
    {
        MaxSdkCallbacks.Rewarded.OnAdLoadedEvent += HandleRewardedAdLoaded;
        MaxSdkCallbacks.Rewarded.OnAdLoadFailedEvent += HandleOnRewardAdFailedToLoad;
        MaxSdkCallbacks.Rewarded.OnAdDisplayedEvent += HandleVideoAdOpeningEvent;
      MaxSdkCallbacks.Rewarded.OnAdClickedEvent += Handler;
        MaxSdkCallbacks.Rewarded.OnAdRevenuePaidEvent += HandleVideoAdPaidEvent;
        MaxSdkCallbacks.Rewarded.OnAdHiddenEvent += HandleRewardedAdClosed;
        MaxSdkCallbacks.Rewarded.OnAdDisplayFailedEvent += OnAdDisplayFailedEvent;
        MaxSdkCallbacks.Rewarded.OnAdReceivedRewardEvent += OnAdReceivedRewardEvent;
    }
}
```

```
1 usage
private void HandleBannerPaidEvent(string id, MaxSdkBase.AdInfo adInfo)
{
    SonatAnalyticTracker.LogFirebaseRevenue(AdsPlatform.max, bannerAdapter, adInfo.Revenue, adInfo.RevenuePrecision, adType: "banner");
    SonatAnalyticTracker.LogAppsFlyerAdRevenue(AdsPlatform.max,bannerAdapter, adInfo.Revenue, adType: "banner");
}
```

```
1 usage  ··· More
private void HandleVideoAdPaidEvent(string adUnitId, MaxSdkBase.AdInfo adInfo)
{
    SonatAnalyticTracker.LogFirebaseRevenue(AdsPlatform.max, rewardedAdapter, adInfo.Revenue, adInfo.RevenuePrecision, adType: "rewarded");
    SonatAnalyticTracker.LogAppsFlyerAdRevenue(AdsPlatform.max,rewardedAdapter, adInfo.Revenue, adType: "rewarded");
}
```

```
2 usages
private void HandleInterstitialAdPaidEvent(string id, MaxSdkBase.AdInfo adInfo)
{
    UIDebugLog.Log(nameof(HandleInterstitialAdPaidEvent) + " event received");
    SonatAnalyticTracker.LogFirebaseRevenue(AdsPlatform.max, interstitialAdapter, adInfo.Revenue, adInfo.RevenuePrecision, adType: "interstitial");
    SonatAnalyticTracker.LogAppsFlyerAdRevenue(AdsPlatform.max,interstitialAdapter, adInfo.Revenue, adType: "interstitial");
}
```

Tracking Behaviours

1. **Lưu lại placement trước khi show quảng cáo để phục vụ phân tích revenue**

```
Event handler  2 asset usages  2 usages
public void ShowRewarded(string placementName)
{
    SonatAnalyticTracker.RewardedLogName = placementName;
    ads.ShowVideoAds();
}

public void ShowInterstitial(string placementName)
{
    SonatAnalyticTracker.InterstitialLogName = placementName;
    ads.ShowInterstitial();
}
```

2. **Tracking**

Các logs mặc định đã được model sẵn trong file SonatLogs.cs. Khởi tạo đủ các thêm số cần thiết trước khi dùng hàm Post() để bắn log. Các yêu cầu log sẽ tùy từng dự án được mô tả trong một document khác.

VD :

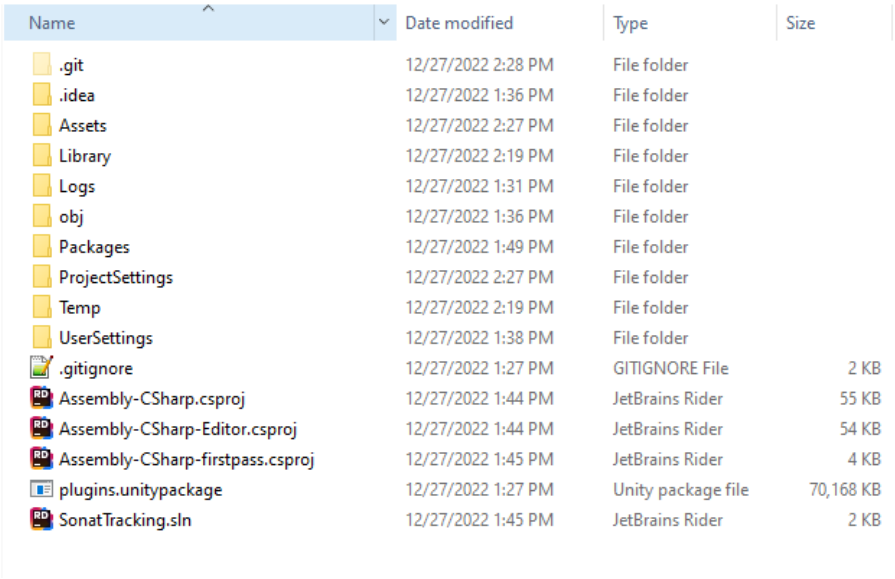
```
Event handler  1 asset usage  1 usage
public void TestLogLevelUp()
{
    var log = new SonatLogLevelUp()
    {
        level = "1",
        character = "bird_1"
    };
    log.Post();
}
```

Other

Tham khảo scene Test trong Assets/SonatTracking/Test

Import plugins.unitypackage để chạy thử project

SonatTracking



Name	Date modified	Type	Size
.git	12/27/2022 2:28 PM	File folder	
.idea	12/27/2022 1:36 PM	File folder	
Assets	12/27/2022 2:27 PM	File folder	
Library	12/27/2022 2:19 PM	File folder	
Logs	12/27/2022 1:31 PM	File folder	
obj	12/27/2022 1:36 PM	File folder	
Packages	12/27/2022 1:49 PM	File folder	
ProjectSettings	12/27/2022 2:27 PM	File folder	
Temp	12/27/2022 2:19 PM	File folder	
UserSettings	12/27/2022 1:38 PM	File folder	
.gitignore	12/27/2022 1:27 PM	GITIGNORE File	2 KB
Assembly-CSharp.csproj	12/27/2022 1:44 PM	JetBrains Rider	55 KB
Assembly-CSharp-Editor.csproj	12/27/2022 1:44 PM	JetBrains Rider	54 KB
Assembly-CSharp-firstpass.csproj	12/27/2022 1:45 PM	JetBrains Rider	4 KB
plugins.unitypackage	12/27/2022 1:27 PM	Unity package file	70,168 KB
SonatTracking.sln	12/27/2022 1:45 PM	JetBrains Rider	2 KB