

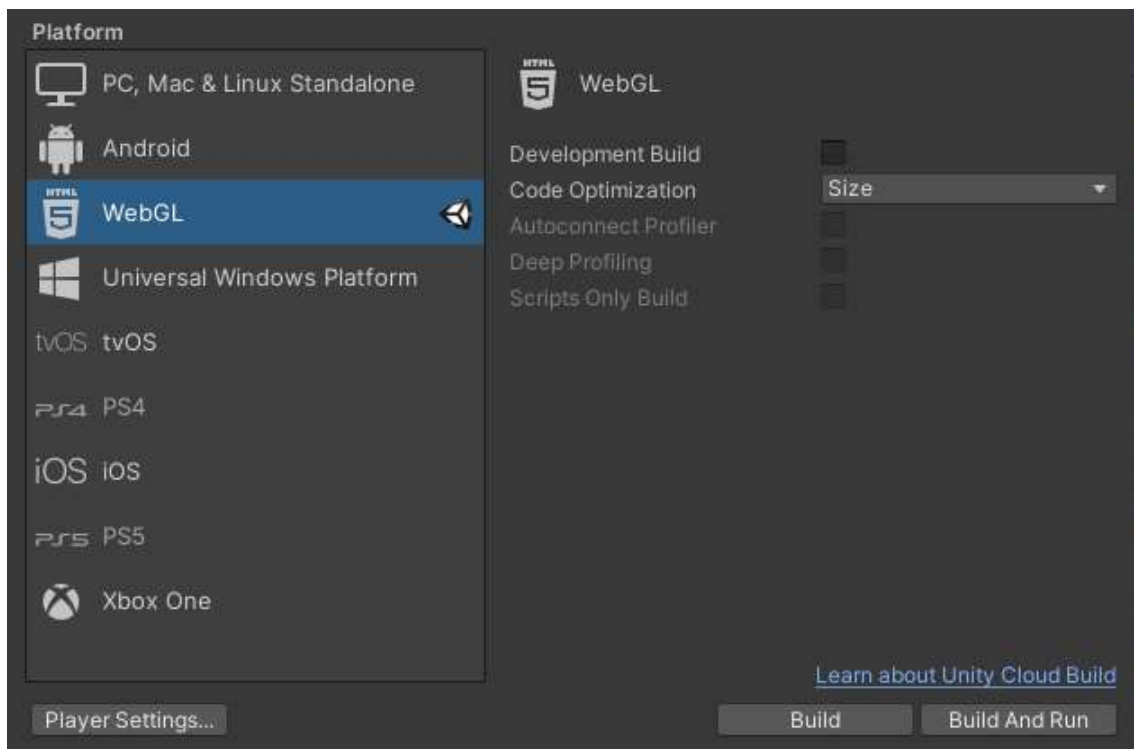


Hello and Welcome to the **ScoreMilk** platform!

Quick Start

1) Select “WebGL” as game Platform.

- Go to: File > Build Settings > Platform
- Select “WebGL”

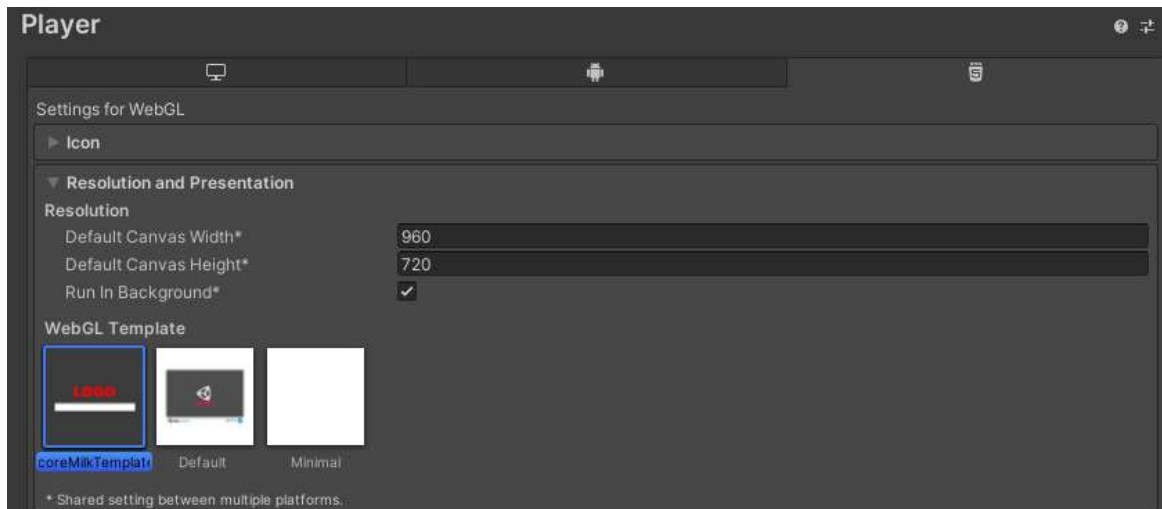


If you don't find this option or it is disabled, please install WebGL build support package in Unity Hub.



2) Set WebGL Template to "ScoreMilkTemplate"

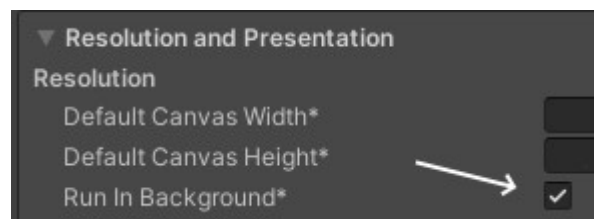
- Go to: Edit > Project Settings > Player > WebGL > Resolution and Presentation
- Select "ScoreMilkTemplate"



ScoreMilk Template should have been included in .unitypackage

3) Check "Run in Background" option

- Go to: Edit > Project Settings > Player > WebGL > Resolution and Presentation, if you are not there yet
- Check in "Run in Background" if it's not set in yet

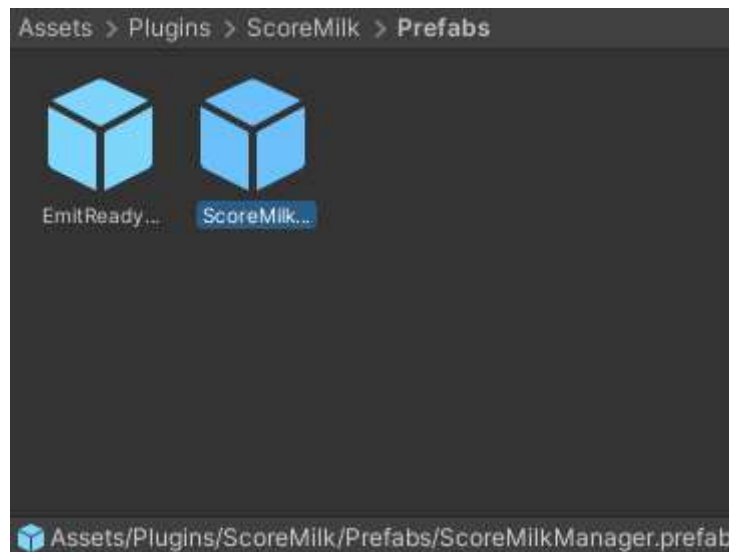


Option to set, width and height values not relevant for this quick start



4) Drag ScoreMilkManager Prefab to first scene

- Go to folder: Assets/Plugins/ScoreMilk/Prefabs
- Select and Drag “ScoreMilkManager” prefab to scene



Prefab can be in every scene. It is a singleton that automatically removed copies

5) Subscribe GameObjects to ScoreMilk Events

- You Subscribe events in a GameObject like this:

```
private void OnEnable() {  
    ScoreMilk.Connection.OnReceivedMatchmakingStart += OnMatchmaking;  
}  
  
private void OnDisable() {  
    ScoreMilk.Connection.OnReceivedMatchmakingStart -= OnMatchmaking;  
}
```

- You have to Subscribe/Unsubscribe to 4 events:

Event

```
ScoreMilk.Connection.OnReceivedMatchmakingStart  
ScoreMilk.Connection.OnReceivedToPractice  
ScoreMilk.Connection.OnReceivedCancelMatch  
ScoreMilk.Connection.OnReceivedStart  
  
ScoreMilk.Connection.OnReceivedWalletConnected  
ScoreMilk.Connection.OnReceivedWalletDisconnected
```

Trigger

```
Player pressed "play" button  
Player pressed "practice" button  
Match was cancelled for any reason  
Both players are ready and accepted the  
required crypto transaction  
Wallet connected  
Wallet disconnected
```

Notes:

- Unsubscribing is not strictly necessary, but it's considered good practice.
- In “Assets\Plugins\ScoreMilk\Scripts\Example” you can see “GameControllerExample.cs” script for a usage example and basic explanation



6) Emit messages to server accordingly to game and situation

- You Emit messages like this:

```
ScoreMilk.Connection.EmitAddScore(points);
```

- You must Emit 3 messages:

Message

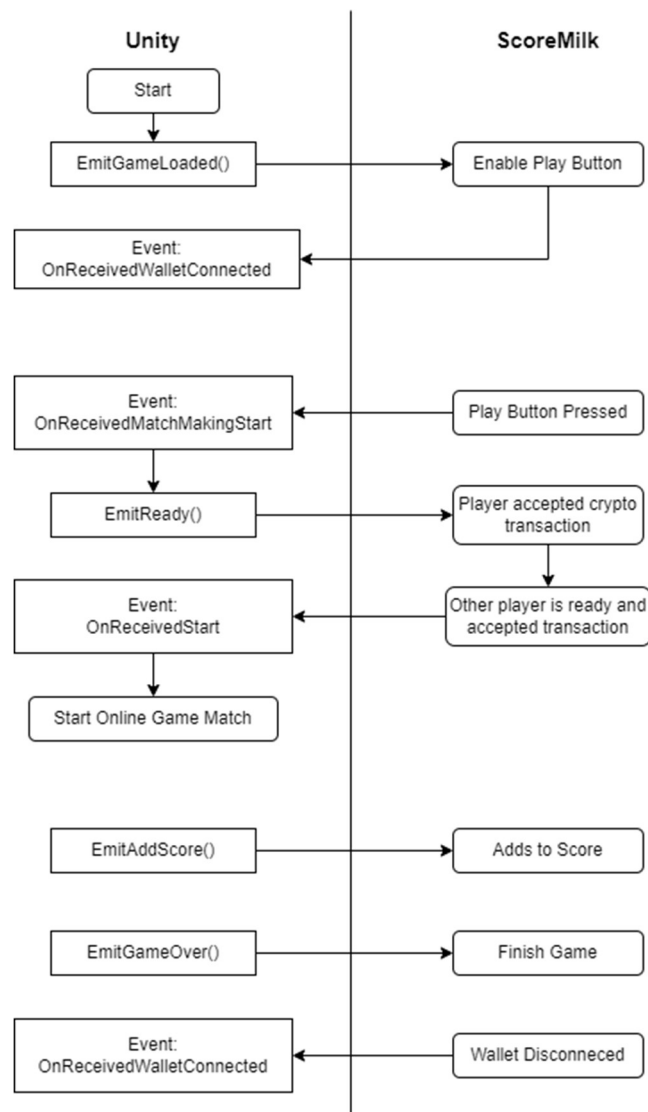
```
ScoreMilk.Connection.EmitReady();  
ScoreMilk.Connection.EmitAddScore(int points);  
ScoreMilk.Connection.EmitGameOver(int points);
```

Must emit when...

When online Scene is fully loaded
When player scored <points> points
When game has ended.

Notes:

- Location of code varies heavily accordingly with game
- Points can be positive or negative. Negative points decrease final value.
- Points emitted at Game Over must be the same as the sum of all points emitted previously.



Simplified flow of information between Unity game and ScoreMilk matchmaking server



7) You are done!

- Build the game and send us the built files!

Documentation

Events

ScoreMilk.Connection.OnReceivedMatchmakingStart

- Event triggered when Player pressed "play" button.
 - Game expected to start loading online scene or show “searching for match” message
-

ScoreMilk.Connection.OnReceivedToPractice

- Event triggered when Player pressed "play" button.
 - Game expected to load and play practice scene, or go to title screen if practice scene does not exist
-

ScoreMilk.Connection.OnReceivedCancelMatch

- Event triggered when match was cancelled for any reason
 - Game expected to return to title screen
-

ScoreMilk.Connection.OnReceivedStart

- Event triggered when both players are ready, sent their “EmitReady()” messages and accepted the required crypto transaction
 - Game expected to start match properly
-

ScoreMilk.Connection.OnReceivedWalletConnected

- Event triggered when player wallet is connected
 - Sends wallet id as string in event
-



ScoreMilk.Connection.OnReceivedWalletDisconnected

- Even triggered player wallet is disconnected
-

Methods

ScoreMilk.Connection.EmitReady();

- Must be emitted when game is ready to start an online match.
 - Game expected to emit after loading online scene, but before starting the match properly
-

ScoreMilk.Connection.EmitAddScore(int points);

- Must be emitted when player scored <points> points
-

ScoreMilk.Connection.EmitGameOver(int points);

- Must be emitted when game has ended
 - Game expected to send the final points count, and the value must coincide with all values sent
-



Example

See "GameControllerExample.cs" script for a usage example and basic explanation.

Community

Discord: <https://discord.gg/N2mEknPs>