

MANUAL



REMOTE REPLAY RECORDER

For Unity

By RealitySims
v0.2, 2024-01-28

GIT: <https://github.com/RealitySims/rs-remote-replay>

INTRO

Remote Replay for Unity is a simple, lightweight system to record & analyze replays from Unity 2D games. You can integrate the system into your mobile games in just a few steps, and watch how your players play your game almost in real time.

HOW IT WORKS

Remote Replay records visual positions of sprites and camera for a period you can define (onboarding, challenging level or boss etc.). ***It does not record videos in the traditional sense (such as mp4).*** Once finished, it either caches the replay(if your player is offline) or connects to Firebase and sends the replay to Firebase storage. If the player is offline it simply tries to send their replay later.



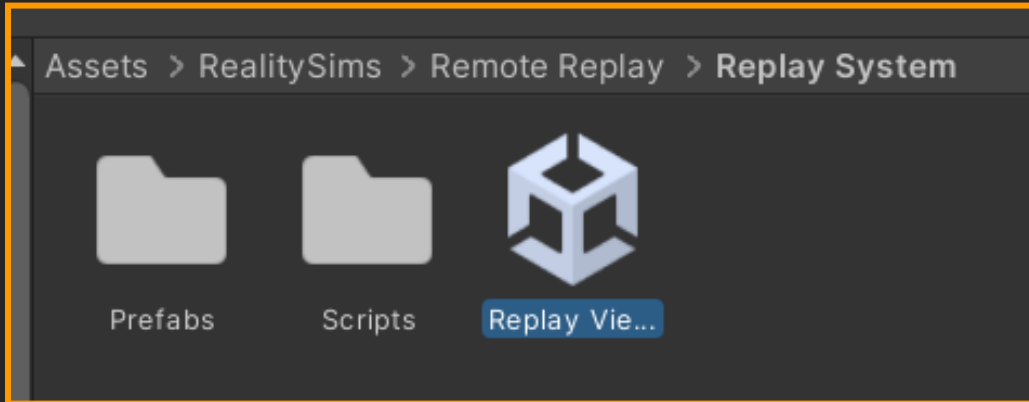
The replays are very lightweight (optimized data structure & gzip compression) - though it also depends on how you configure Remote Replay. For example, you can change the fps of recording or which data you record (rotation, Z axis etc.) which might affect the replay size considerably.

Later, when you want to watch your replays, check out Firebase storage for newly arrived replay files, copy & paste the filename into our replay player scene, and you're ready to watch.

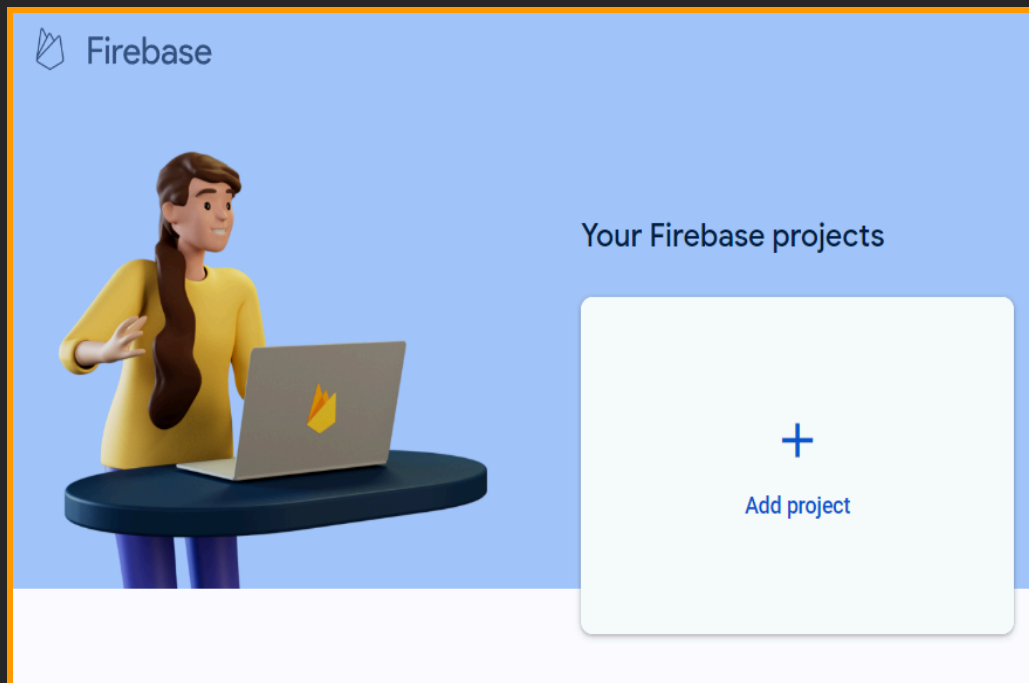


QUICKSTART

1. Add the package to your Unity Project, it will contain 2 folders:
 - One called *"Examples"* which currently contains one sample game.
 - Another folder called *"Replay System"* which also contains the replay player scene called *"Replay Viewer"*

















2. Create a new Firebase project at <https://console.firebase.google.com>



Please follow Google Manual (<https://firebase.google.com/docs/unity/setup>) if unsure how to finish setting up a Firebase project. This will include installing the Firebase SDK, in case it's not already a part of your project.

You will need these 3 Firebase modules for Remote Replay to work:

| | | | | | | |
|---|----------------------|------------|----|------------|------|------------------|
|  FirebaseAnalytics | Unity package file | 86 551 kB | Ne | 87 845 kB | 2 % | 19.10.2023 20:01 |
|  FirebaseAppCheck | Unity package file | 87 745 kB | Ne | 89 035 kB | 2 % | 19.10.2023 20:04 |
|  FirebaseAuth | Unity package file | 89 889 kB | Ne | 91 204 kB | 2 % | 19.10.2023 20:01 |
|  FirebaseCrashlytics | Unity package file | 85 784 kB | Ne | 87 071 kB | 2 % | 19.10.2023 20:01 |
|  FirebaseDatabase | Unity package file | 96 096 kB | Ne | 97 432 kB | 2 % | 19.10.2023 20:02 |
|  FirebaseDynamicLinks | Unity package file | 86 051 kB | Ne | 87 331 kB | 2 % | 19.10.2023 20:02 |
|  FirebaseFirestore | Unity package file | 105 922 kB | Ne | 107 330 kB | 2 % | 19.10.2023 20:04 |
|  FirebaseFunctions | Unity package file | 86 588 kB | Ne | 87 876 kB | 2 % | 19.10.2023 20:03 |
|  FirebaseInstallations | Unity package file | 86 162 kB | Ne | 87 438 kB | 2 % | 19.10.2023 20:03 |
|  FirebaseMessaging | Unity package file | 87 306 kB | Ne | 88 595 kB | 2 % | 19.10.2023 20:02 |
|  FirebaseRemoteConfig | Unity package file | 87 895 kB | Ne | 89 193 kB | 2 % | 19.10.2023 20:02 |
|  FirebaseStorage | Unity package file | 92 475 kB | Ne | 93 792 kB | 2 % | 19.10.2023 20:03 |
|  LICENSE | Soubor | 5 kB | Ne | 13 kB | 65 % | 19.10.2023 20:04 |
|  readme | Markdown Source File | 24 kB | Ne | 90 kB | 74 % | 19.10.2023 20:04 |

FirebaseAnalytics: For logging the “replay_uploaded” event, this is useful to get more details on the uploaded replays. If you're exporting analytics to BigQuery, you could filter replay names by country, version etc.

FirebaseAuth: This is mandatory, you need this to be able to use Firebase anonymous login.

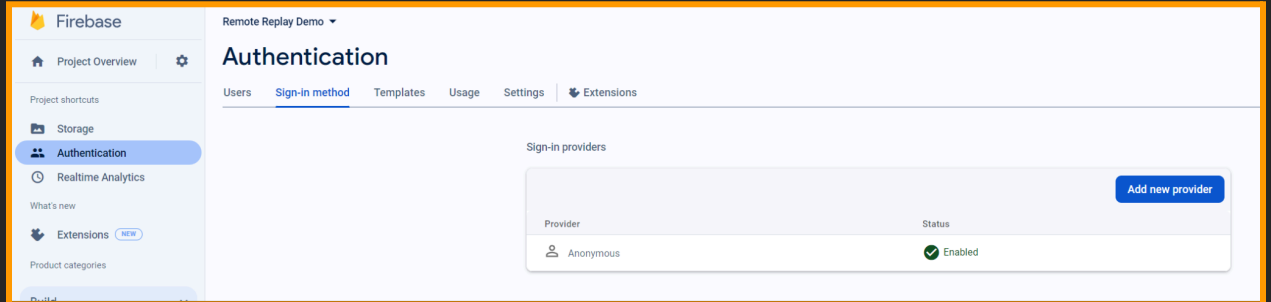
FirebaseStorage: This is also mandatory, all of the replay files are saved into Firebase Storage.

Don't forget to download **google-services.json** and put it into your Assets folder.

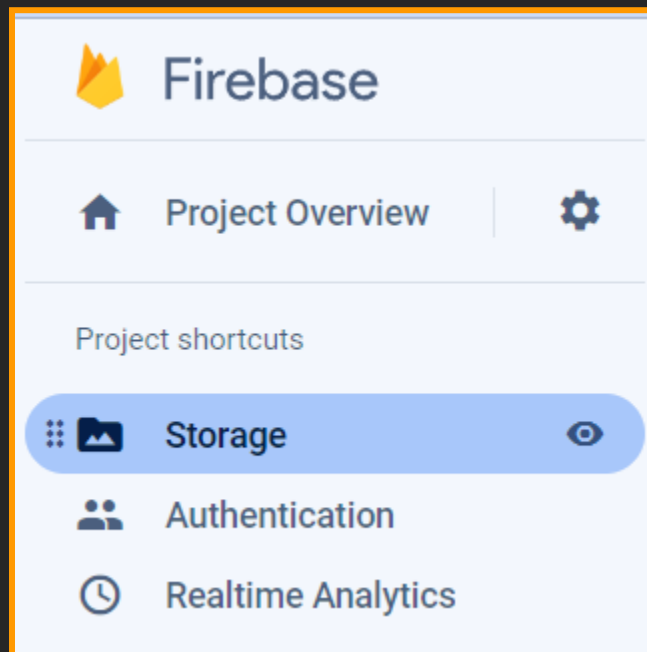
You might need to restart Unity, especially in case of subsequent Firebase errors complaining about malformed or missing google-services.json.



3. Enable Anonymous login in Firebase console, if not enabled yet.



4. Enable Storage in Firebase console, if not enabled yet. This is where your replays will be uploaded.



Here are the recommended Storage security rules, though of course you may adjust them to your liking if you're familiar with Firebase Storage:

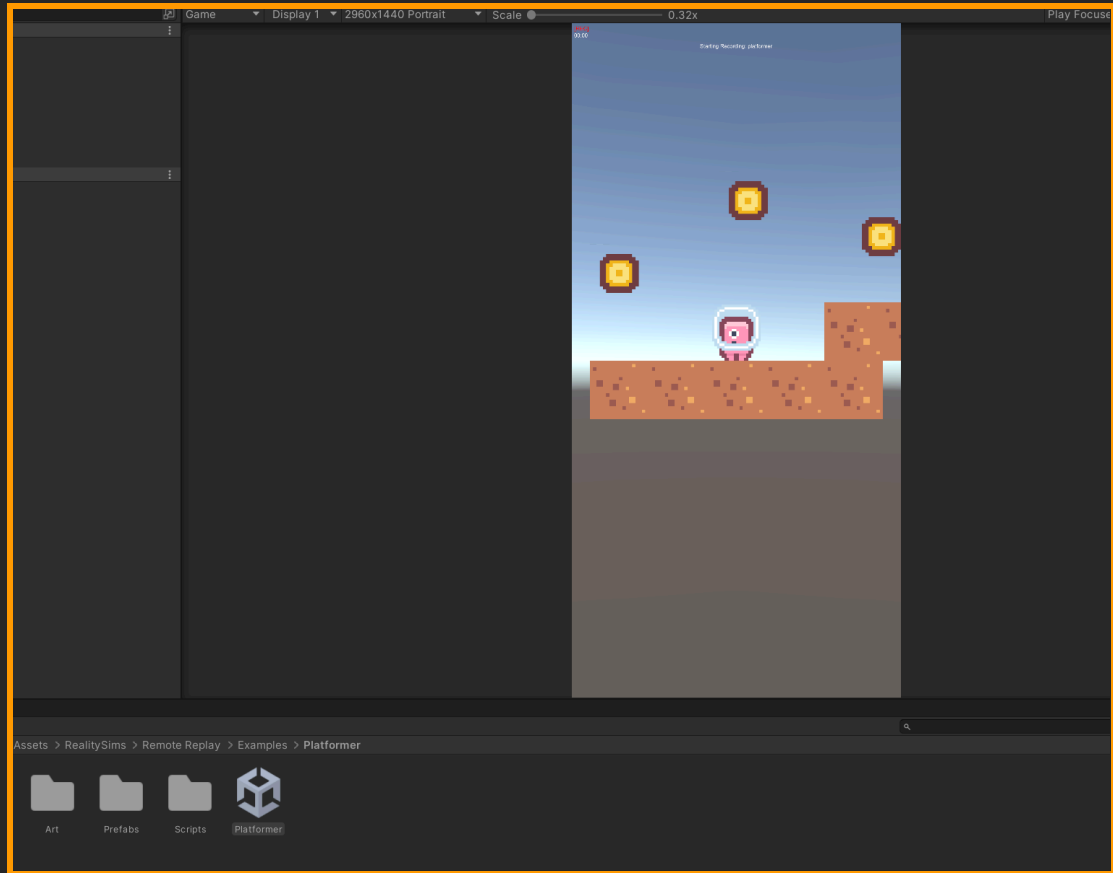
```
rules_version = '2';
service firebase.storage {
  match /b/{bucket}/o {
    match /replays/{document} {
      allow read;
      allow write: if request.auth != null &&
        document.matches(request.auth.uid + '_.+\\.replay');
    }
  }
}
```

5. Install Newtonsoft's JSON package, in case it's not already a part of your project. There is a good walkthrough at this URL:

<https://github.com/applejag/Newtonsoft.Json-for-Unity/wiki/Install-official-via-UPM>

6. Open the sample game scene (*Examples/Platformer*), play for 15 seconds. You should see a red indicator in the top left corner, confirming a replay is being recorded. After the recording finishes, you should see a message in the console confirming the replay has been recorded & uploaded to Firebase.





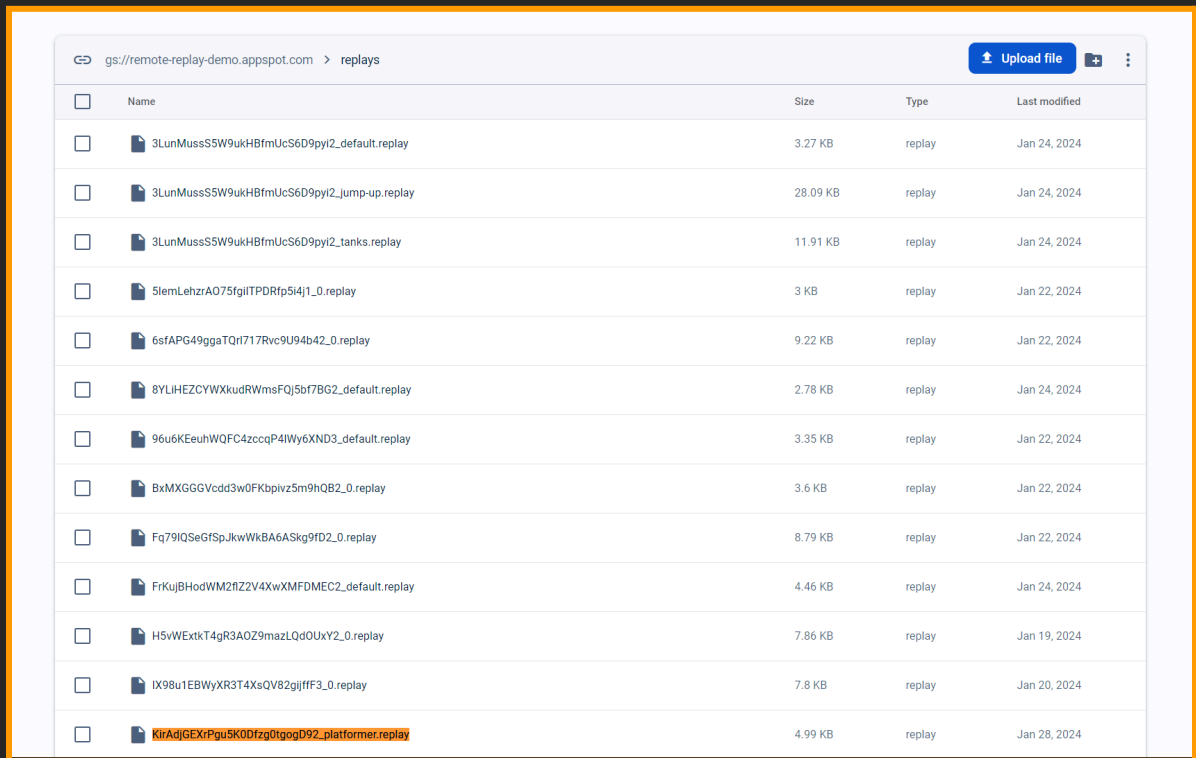
```

UnityEngine.Debug:Log (object)
[17:55:26] USE_AUTH_EMULATOR not set.
UnityEngine.Debug:Log (object)
[17:55:27] USE_AUTH_EMULATOR not set.
UnityEngine.Debug:Log (object)
[17:55:27] User signed in successfully: (KirAdjGEXrPgu5K0Dfzg0tgogD92)
UnityEngine.Debug:LogFormat (string,object[])
[17:55:41] ReplayRecorder: Replay Finished
UnityEngine.Debug:Log (object)
[17:55:41] ReplayRecorder: Attempting to upload replay: KirAdjGEXrPgu5K0Dfzg0tgogD92_platformer.replay
UnityEngine.Debug:Log (object)
[17:55:42] ReplayRecorder: Upload completed successfully
UnityEngine.Debug:Log (object)
[17:55:42] Replay Remotely Saved: KirAdjGEXrPgu5K0Dfzg0tgogD92_platformer.replay
UnityEngine.Debug:Log (object)

```

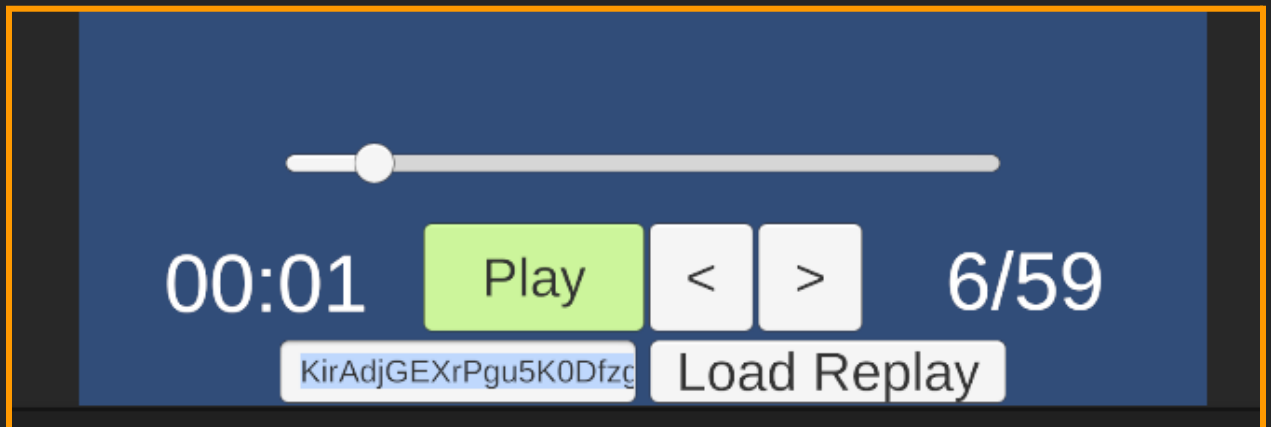


7. Copy & paste the replay filename either from Unity console, or from Firebase storage, which at this point should contain the new replay inside the “replays” folder.



| Name | Size | Type | Last modified |
|--|----------|--------|---------------|
| 3LunMussS5W9ukHBfmUcS6D9pyl2_default.replay | 3.27 KB | replay | Jan 24, 2024 |
| 3LunMussS5W9ukHBfmUcS6D9pyl2_jump-up.replay | 28.09 KB | replay | Jan 24, 2024 |
| 3LunMussS5W9ukHBfmUcS6D9pyl2_tanks.replay | 11.91 KB | replay | Jan 24, 2024 |
| 5lemLehzrA075gilTPDRfp5i4j1_0.replay | 3 KB | replay | Jan 22, 2024 |
| 6sfAPG49ggaTQr717Rvc9U94b42_0.replay | 9.22 KB | replay | Jan 22, 2024 |
| 8YLIHEZCYWXkudRWmsFQj5b7BG2_default.replay | 2.78 KB | replay | Jan 24, 2024 |
| 96u6KEuhWQFC4zccqP4IWY6XND3_default.replay | 3.35 KB | replay | Jan 22, 2024 |
| BxMXGGGVcdd3w0FKbplvz5m9hQB2_0.replay | 3.6 KB | replay | Jan 22, 2024 |
| Fq79IQSeGfSpJkwWkBA6ASkg9fD2_0.replay | 8.79 KB | replay | Jan 22, 2024 |
| FrKujBHodWM2fz2V4XwXMFDMEC2_default.replay | 4.46 KB | replay | Jan 24, 2024 |
| H5vWExtkT4gR3AOZ9mazLQdOUxY2_0.replay | 7.86 KB | replay | Jan 19, 2024 |
| IX98u1EBWyXR3T4XsQV8Zgjff3_0.replay | 7.8 KB | replay | Jan 20, 2024 |
| KirAdjGEXrPgu5K0Dfzg0tgogD92_platformer.replay | 4.99 KB | replay | Jan 28, 2024 |

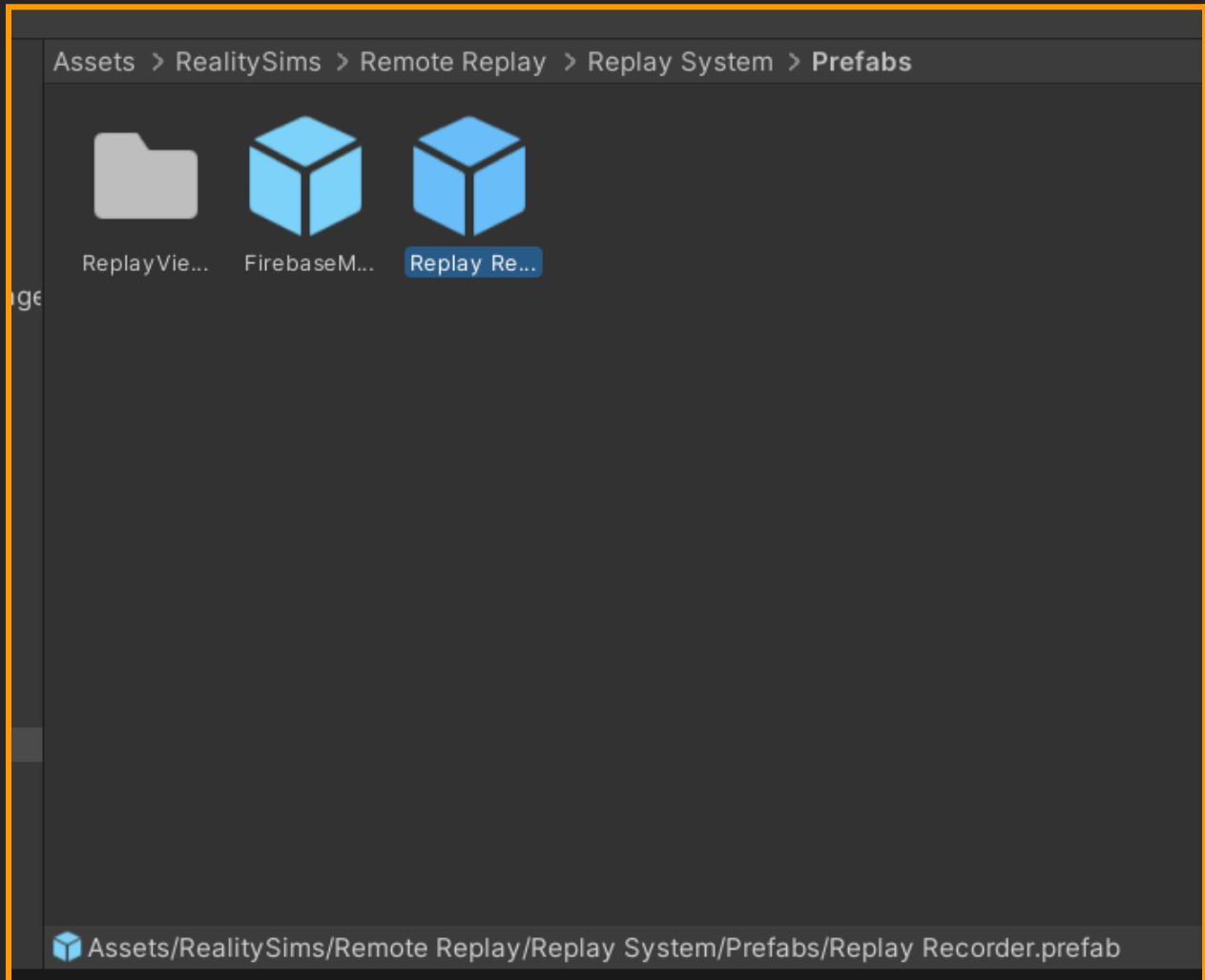
8. Launch the replay recorder scene in “Replay System/Replay Viewer.scene”, copy & paste the filename and watch the replay!



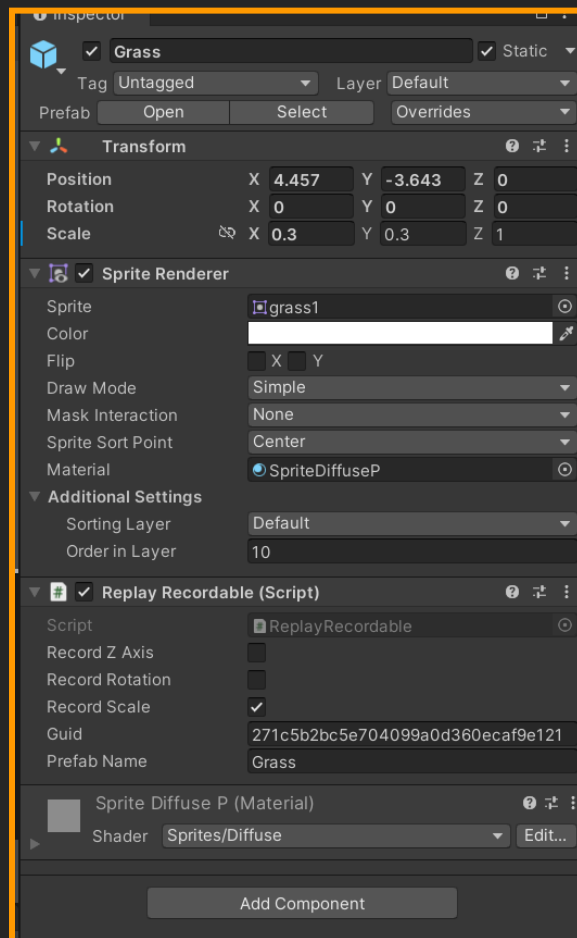
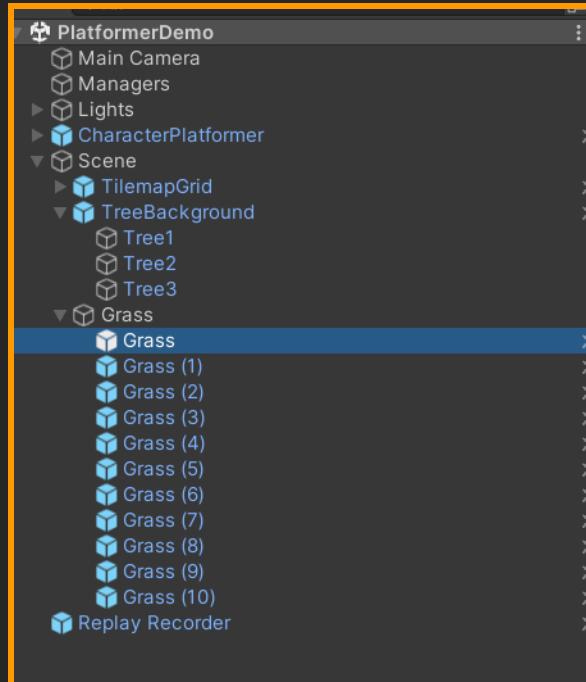
INTEGRATING REMOTE REPLAY INTO YOUR GAME

In order to integrate the replay system into your game, you need to do the following:

- 1) Drag the “**Replay Recorder**” from the Prefabs folder into your scene’s hierarchy.



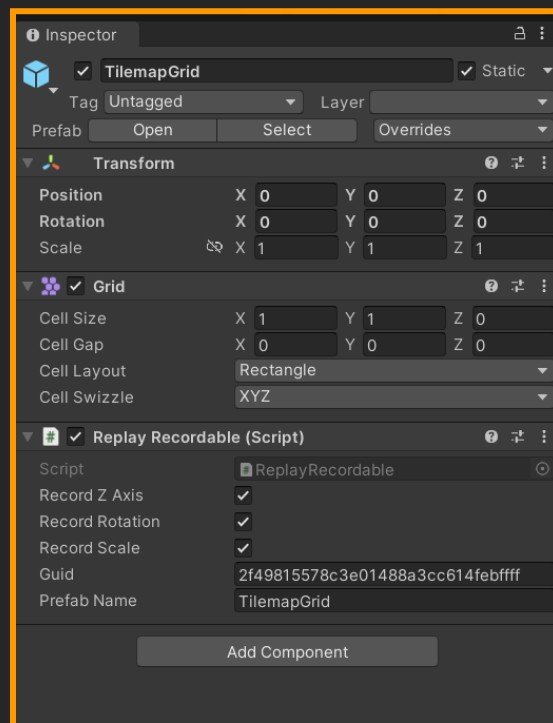
- 2) Anything you want to record needs to be a prefab, **and** it needs to have a “ReplayRecordable” script attached to it. The script needs to be attached “inside” of the prefab, not from the scene view, and it can reside on top of the prefab’s hierarchy, in case you’re using nested game objects inside the prefab.



You can tick / untick which properties you want to record, depending on your game. Currently we support recording Z axis, rotation, scale. Position is recorded automatically.

SUPPORTED RECORDABLES

Currently, we record anything with a transform in the replay. However, we only support viewing *Sprites* with the SpriteRenderer component, as well as *Tilemaps*. To record a tilemap, you can attach the recordable to the gameObject with Grid component, like this:

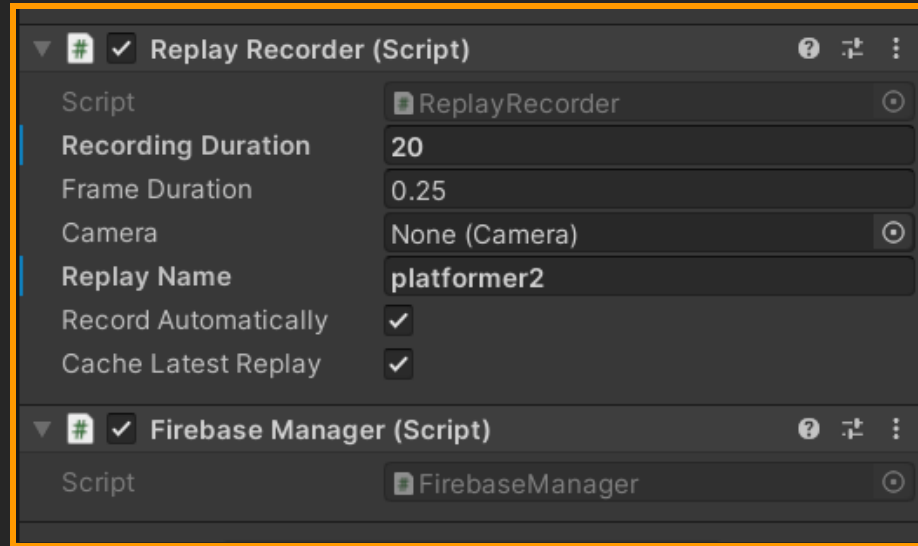


If something is not supported to be viewed but you're still recording it, you will see a filled circle with the name of the prefab as a stand in.

RECORDING FROM C# SCRIPT

By default, the Replay Recorder prefab records on Start automatically, so you don't need to do anything. However, you might want to record a replay at a particular game event later in the game.





To do that, untick “Record Automatically” which is ticked by default, as shown on the screenshot above. Then, use the following 2 methods on the Replay Recorder component to capture your replays from a C# script:

```
public void StartRecording(string replayName = null)
```

```
public void StopRecording()
```

Happy recording & replaying!

