



Photo by inagakijunya

【Unity】UNetからMirrorへの乗り換え入門

♡ 4



fuqunaga

2019/06/21 14:55

以前からobsoleteだったUNetがついにUnity2019で無くなりました。
と見せかけてパッケージ化して**PackageManager**で**Multiplayer HLAPI**として生きてます。これに気づかずにMirror乗り換えを結構まじめにやっちゃったのでせっかくなのでまとめて公開します！

* 検証バージョンはUnity2018.4 + Unet → Unity2019.1 + Mirror v3.7.3です。



Mirrorとは

vis2k/Mirror

A community replacement for Unity's abandoned UNET Networking
github.com

UNetの代替となるOSSのアセットで、[uMMORPG](#)でも使われている（というか開発者が同じ）**MMOスケールでもばっちり動くUNet**みたいな感じです。UNetなくなるの辛い！ってときには最も乗り換えコストが少ない選択肢だと思います。

使ってみた感想

そもそもMirrorに乗り換えて大丈夫なのかという点が気になる方も多いかと思いますが、触った感じ不可解な挙動もなく大丈夫そうです。つまづくところはだいたいUNetとの相違点で、後述の部分と少し被りますが公式の[Migration Guide](#)もあります。

なお、UNet未経験でMirrorから入るのはちょっとハードルが高そうです。UNetの概念的なものはMirrorのドキュメントだと少し説明が少なめなので[Unityのドキュメント](#)をざっと撫でてから始めるとよさそうです。

乗り換え



ほぼUNetと同等のインターフェースなのですが、より便利にしたり思想的に異なったりする部分があり、そのままでは動かないこともあります。ここでは主に自分が気づいた部分について紹介します。

おしながき

挙動の変更

- ・ PlayerがないとSpawnされない
- ・ DontDestroyOnLoadなやつはSpawnされない

コードの変更

- ・ NetworkSettingsAttribute廃止
- ・ SyncListStructがSyncListに
- ・ MessageTypeの代わりにSend<T>で型で区別する

ほとんど影響のない変更

- ・ デフォルトTCP通信（変更化）
- ・ Time系のサービス追加
- ・ 一部のシリアルイズ系の命令削除

* ロビーやマッチング周りはずっと使ってたのでノータッチです。

PlayerがないとSpawnされない



UNetにはプレイヤーの概念がありますが特に用意しなくても動作していました。しかし**Mirror**では**プレイヤーを作らないとクライアント側でオブジェクトがSpawnされません**。

これは次のようにクライアントプレイヤーをチェックして何を見せるかを判断するようになっているため、プレイヤーの位置や状態に応じた部分同期や不正なクライアントによるセキュリティリスクを考慮しているようです。

1. クライアントからサーバーにプレイヤー追加を通知
2. クライアントプレイヤーをチェックしてSpawnするオブジェクトの選定
3. 選定したオブジェクトをクライアント側でSpawn

別にプレイヤーいないんだけどーって場合はとりあえず空のプレイヤーを用意してやり過ごすのが良さそうです。

DontDestroyOnLoadなやつはSpawnされない

タイミングによっては消したはずのプレイヤーがDontDestroyOnLoadにいてこれを拾ってしまうためのようです。オフラインシーンになっても次オンラインシーンに復帰したら継続して・・・ってやつで結構使ってたので個人的になかなか痛い変更です。



メンバーをいじる形になってます。

```
[NetworkSettings(channel=1,sendInterval=0.05f)]
public class NetStreamer : NetworkBehaviour
{
    ...
}
```

replace it with:

```
public class NetStreamer : NetworkBehaviour
{
    void Start()
    {
        syncInterval = 0.05f;
    }
}
```

<https://vis2k.github.io/Mirror/General/Migration>

SyncListStructがSyncListに

UNetがnamespaceを考慮せずにSyncListStructを触ってしまうため（おそらく同居しているプロジェクトを想定して）名前を変えたかったようです。

また、SyncList<T>.Callbackにitemが渡るようになってます。

```
public delegate void SyncListChanged(Operation op, int itemIndex, T item)
```



ほかにも**SyncDictionary**などが増えていたりSyncList系はいい感じに発展しています。

MsgTypeの代わりにSend<T>で型で区別する

UNetでは独自のメッセージをやりとりするのにMsgTypeをほかと被らないように作ら必要があったのですが、これがメッセージの型から自動生成されるようになっていて不要になりました。便利！

こんなのが、

```
public class MyMsgType {  
    public static short Score = MsgType.Highest + 1;  
};  
  
~  
  
NetworkServer.SendToAll(MyMsgType.Score, msg);
```

<https://docs.unity3d.com/ja/current/Manual/UNetMessages.html>

こんなふうにはけるようになってます。

```
NetworkServer.SendToAll(msg);
```



<https://github.com/vis2k/Mirror/blob/11750a9e7a3f330e508642d1fca51173c3a4d5a8/Assets/Mirror/Runtime/NetworkServer.cs#L225>

デフォルトTCP通信(変更化)

UDPの不確実性だと辛すぎたようでかなり熱く語られています。

Why TCP by default and not UDP?

Documentation for Mirror, a high level networking library for
vis2k.github.io

ただトランスポート層は別ライブラリ扱いにされていてユーザーが自由に変えることができ選択肢も充実しています。

Transports

Documentation for Mirror, a high level networking library for
vis2k.github.io

Time系のサービス追加

UNetでは一足はやく消滅したNetwork.timeに変わるものが用意されています。またRTTが取れたり機能追加されています。

vls2k.github.io

一部のシリアライズ系の命令削除

次のメソッドがなくなっています。

```
NetworkWriter.SeekZero()
```

```
NetworkReader.readMessage<T>()
```

基本的にライブラリ内で使用されるものなので影響あるケースは少なそうですが、たまたま自分は使っていたので気づきました。

余談

UNetの頃からWeaverという名前に変なタイミングでエラーを吐くことがよくあったのですが今回Mirrorを見てこれがなにかわかりました。なんとビルドした実行ファイルを書き換えてるんですね。

Mono.CecilというコンパイルしたILをいじれるライブラリを使って、**Message**のシリアライザや前述の**MsgType**なしでも動くコードを生成しているようです。前者なんかはリフレクションで動的にでも出来そうですがもしかしたらパフォーマンスを気にしてるのかもしれない。



Mono.Cecilについてはこちらのサイトわかりやすいです。

既存の.NET アセンブリにMono.Cecilでコードを差し込む - ぶろじぇくと、みすじら。

www.misuzilla.org

これテストとか、いざってときにめっちゃ使えそうですね。

まとめ

自分が検証して気づいたUNetとMirrorの相違点を中心に書いてみました。乗り換えを検討してる方の参考になれば幸いです。

また、これからネットワーク対応を検討してる方にはUnityがフローチャートを出してるのでこちらも参考になるかもしれません。

Navigating Unity's multiplayer Netcode transition – Unity Blog

As many of you know, we put UNet into maintenance mode because
blogs.unity3d.com

#Unity

#技術メモ





noteで技術記事の実験中です。もしお役にたちましたらスキ、サポートなどのリアクションをもらえるとうれしいです

 サポートをする



fuqunaga

ゲーム／インタラクティブアートのプログラマー。Unityさんまい。
<https://fuquna.ga>



フォロー

オススメ記事


【WPF】TextBlockの見た目を変えないでURLをリンクする

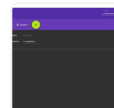
fuqunaga

【WPF】複数のDLLをまとめて1つのexeにする

fuqunaga

【WPF】PowerShellでリモートPCを操作する手順とGUIツール

 2 fuqunaga



【Unity】Package Managerに自作
パッケージを追加する

この記事が入っているマガジン



フォロー

Unity Technical Report



コメントを投稿するには、 ログイン または 会員登録 をする必要があります。

こちらもおすすめ

簡単な**Discord Bot**の作り方
(初心者向け)

♡ 44

 bami

...

アーキテクチャから見直した **Android 版 Zaim** の大規模...

♡ 16

 Hiroshi Hara

...

Android / iOS アプリの開発にクロスプラットフォームの...

♡ 84

 najeira

...

VRChat内で**AR**アプリの**UI/UX**を研究できるか検証してみた...


♡ 18

 SRETKS

...

Typeのデータを**Realm**から**Firestore**に移行しました

♡ 23

 かっくん

...

さよなら**Sketch**、こんにちは**Framer X**


♡ 104

 敢太

...

Goと**Dependency Injection**の現在

♡ 37

 Seiji Takahashi - timakin

...

Go 言語で **API** サーバーの開発をはじめのためのライブラ...

♡ 68


 maco


...

Kotlinでコードを自動生成する


2019.3.23-30のiOSアプリ開発ニュース








Cordovaをdisる人類全員に読んでほしい「Cordova つらい...




♡ 55

 榎原昌彦

...

Flutter感想

♡ 17

 mizchi

...

[クリエイターの方へ](#) [noteプレミアム](#) [note pro](#) [よくある質問・noteの使い方](#) [ノート](#) [マガジン](#) [ユーザー](#)
[ハッシュタグ](#) [プライバシー](#) [ご利用規約](#) [特商法表記](#) [クリエイターへのお問合せ](#) [noteカイゼン目安箱](#)
[クリエイターの推薦](#)