

はじめに

みなさん、ゲーム作っていますか？

Unityゲームエンジンの登場によってゲーム開発者人口は爆発的に増え、アプリストアには世界中のクリエイターのゲームがひしめき合っています。Unityを使っている開発者は、およそ550万人(2016年7月現在)と言われています。

Unityは「ゲーム開発の民主化」を掲げ、さまざまな機能アップデートを行ってきました。誰でもマルチプラットフォームで動作するゲームが作れること。誰でも美しい描画エフェクトが使えること。誰でも高度な経路探索AIが使えるようになること…それらの革新は、多くの個人・小規模クリエイターが、これまでは不可能だったゲームの表現を可能にしてくれました。

そして、今まさに発展しつつある分野が「ネットワーク機能」の民主化ではないかと考えています。これまでネットワーク機能は、リアルタイム・非リアルタイムの両方で、個人・小規模クリエイターがなかなか挑戦しづらい分野でした。

ネットワーク機能を実装するためには、まずは何がともあれ「サーバー側」の開発の知識が必要でした。自分でサーバーPCを稼働させるか、あるいは外部のサービスを使ってリモートに必要なミドルウェアをインストールし、コマンドラインで操作するといった、「ゲームづくりの面白さ」を作り込むこととはまた別の準備が必要だったのです。

しかし、そうした壁を解決するサービスが現れてきています。もちろん、プロフェッショナルの現場では独自のサーバー構築が適した場合がありますが、新しいサービスの登場によってこれまでネットワーク機能を搭載することが難しかった個人・小規模クリエイターや、Unityの初学者でも使うことができるようになってきています。

あなたのゲームにネットワーク機能を導入することで、新しいゲームシステムを構築できたり、プレイヤーが継続的に遊んでくれる仕組みを加えたりすることができるようになります。本書を通して、ゲームクリエイターのみなさんのアイデアが、より輝くようなお手伝いができると思っています。

2017年5月 著者を代表して 一條 貴彰

本書の構成

本書では、サーバーへのデータ保存と取得を中心とした「非リアルタイム系」のネットワーク機能と、アクションゲームのオンラインマルチプレイなどの「リアルタイム系」ネットワーク機能の2つのテーマを解説しています。

基本的には、小規模・個人ゲーム開発者でも作ることができるネットワーク機能の実装手法を紹介します。ネットワークゲーム機能をはじめて開発する人向けの書籍のため、何十万人も同時にアクセスするような大規模なソーシャルゲームやオンラインMMOの開発に適した内容にはなっておりませんので、ご了承ください。

また本書は、何らかのUnity初学者向け書籍を読んだことがある開発者か、実際にUnityでアプリ制作を行ったことがある開発者を対象としています。これからUnityを学びたいという方は、『Unity5 3D/2Dゲーム開発 実践入門ー作りながら覚えるスマートフォンゲーム制作』（ソシム刊）などの入門書を読んでおくことをお勧めします。

本書では、Unityの基本機能やUnityエディタの使い方、スクリプティングの基礎などに関しては解説していません。

本書は、大きく次の3つのPartと、序章、Appendixで構成されています。

●序章 Unityとネットワークゲームの概要

本書のテーマであるネットワークゲームの概要と、サンプルゲームの紹介、必要な開発環境とライセンスについて解説します。

●Part 1 非同期型ネットワーク機能活用ーNCMB編

mBaaSの1つである「ニフティクラウド mobile backend」(NCMB)を使って、ゲームに非リアルタイム系のネットワーク機能を付加する手法について解説します。

●Part 2 リアルタイム・ネットワークゲーム開発ーUNET基礎編

Unity標準のネットワークAPIである「UNET」(正式名: Unity Multiplayer)を使って、リアルタイム系のネットワーク機能の基礎を簡単なサンプルを作りながら、ステップバイステップで学んでいきます。

●Part 3 リアルタイム・ネットワークゲーム開発ーUNET実践編

本格的なネットワーク対戦型ゲームの実装例をもとに、ネットワークの遅延回避の方法など、リアルタイム・ネットワークゲーム制作での実践的なテクニックを紹介します。

●Appendix

iOSアプリ開発手順の詳細や、クラウド上でアプリをビルドできる「Unity Cloud

Build」の紹介、および本文では取り上げられなかったNCMBのその他の機能などを紹介します。

サンプルプログラムのダウンロード

本書に掲載したサンプルプログラムは、以下のWebページよりダウンロードできます。利用方法の詳細は、本文を参照してください。

- サンプルプログラムのダウンロードWebサイト

<http://www.socym.co.jp/book/1099>

サンプルプログラムの利用について

本書に掲載したサンプルプログラムは、Unityの学習のために作成したもので、実用を保証するものではありません。学習用途以外ではお使いいただけませんので、ご注意ください。なお、本書に掲載したプログラムの著作権は、すべて著者に帰属します。

また、サンプルデータに同梱された3Dモデルデータなどのアセットも、本書の学習用途でのみ利用できます。本書のアセットを別なゲームで使いたい場合は、本書の最終ページに記載された各権利者にご連絡ください。



もくじ

本書の構成	004
-------------	-----

序章 Unityとネットワークゲームの概要..... 016

0-1 統合ゲーム開発環境Unityのおさらい016

Unityゲームエンジンとは？	016
強力なマルチプラットフォーム対応	017
無償から利用可能	017
ゲーム開発で「成功すること」への支援	018
Unityの未来	019

0-2 ゲームのネットワーク機能とは？020

「ネットワークゲーム」の種類	020
非リアルタイム系のネットワーク機能を実装する方法とは？	021
リアルタイム通信対戦系ゲームを実現するには？	022

0-3 本書のサンプルゲームについて025

サンプルゲーム1：『剣士なんですぐ死んでしまうん？』	025
サンプルゲーム2：『Painters』	026

0-4 必要な開発環境とUNETのライセンス027

iOS端末向けの開発環境	027
Android端末向けの開発環境	028
UNETのライセンス体系	032

Part1 非同期型ネットワーク機能活用－NCMB編

1章 サンプルゲームの概要と構造036

1-1 「剣士なんですぐ死んでしまうん？」の概要と遊び方037

サンプルゲームを動作させるには	037
サンプルゲームの遊び方	037

1-2 サンプルゲームの構造040

ゲームのシーン構造	040
サンプルゲームで実装しているネットワーク機能	041

2章 NCMBの導入と基礎042

2-1 「ニフティクラウド mobile backend」とは043

「ニフティクラウド mobile backend」(NCMB)の機能と特徴	043
ネットワーク機能をゲームアプリにどう活用するか	046

2-2 NCMB Unity SDKの導入手順	048
NCMB Unity SDKをプロジェクトにインポートする	048
NCMBを動かすためのオブジェクトをシーン内に生成する	050
NCMBのアカウントを取得する	051
2-3 NCMBの管理画面	055
管理画面の構成	055
NCMBの「アプリ」という単位について	056
ダッシュボードで利用状況を確認	056
2-4 データ保存・取得の基礎	058
スクリプトの作成	058
NCMBの管理画面での確認	061
コールバックによる通信完了後の処理	062
保存したデータを取得する	063
条件をつけてデータを取得する	064

3章 ログイン機能の実装と活用 068

3-1 ゲームとログイン機能の関係	069
ログイン機能が必要な理由	069
スマホゲームでも実は「ログイン」している？	069
NCMBにおけるログインの実装	070
3-2 ログイン機能の実装	071
アプリからサインアップする	071
管理画面でプレイヤーの情報を確認する	073
アプリからログインする	074
会員管理にデータを保存する	075
ログインしているプレイヤーのデータを取得する	077
アプリでログアウトする	077
ログアウトボタンを作成する	077
ログアウト処理のタイミング	080
3-3 ログイン機能の応用とセーブデータの保存	081
サンプルゲームを動作させてみる	081
NCMBに関する通信処理のフローチャート	081
サインアップ処理の解説	083
ログイン処理の解説	087
セーブデータの保存	088
3-4 オートログイン	090
オートログインの概要	090
サンプルゲームでのオートログイン実装を確認する	090
2回目以降の起動	093
3-5 端末引き継ぎ機能	095
引き継ぎコードの発行	095
引き継ぎコードの入力	096
退会の処理の実装	097

3-6 デイリーボーナス	098
デイリーボーナスの実装の概要	098
デイリーボーナスの実装の詳細	099

4章 ネットワーク機能の拡張102

4-1 データストアのおさらい	103
データストア機能と会員管理との違い	103
データストアの「クラス」設計	103
クエリを使って特定のデータを取得する	104
検索条件の組み合わせ	105
4-2 ランキング機能	107
ランキング機能の概要	107
サンプルゲームのランキング機能について	107
ランキングに必要なデータを保存する	108
ランキングデータを取得し、表示する	109
4-3 フレンド機能	111
サンプルゲームのフレンド機能の概要	111
「オープンなプレイヤー情報のリスト」をデータストアに作成する	112
PlayerInfoListに自分の情報を保存する	113
自分のPlayerInfoを更新する	114
フレンド候補のプレイヤーリストを取得する	114
プレイヤーをフレンドに登録する	116
フレンドリストを表示する	117
プレイヤーをIDで検索する	118
4-4 非同期型マルチプレイのゲームシステム	120
サンプルゲームの特徴的なゲームシステム	120
お墓データの保存	122
お墓データの取得	124
自分のお墓のステータスを表示する	125
4-5 ゲームの設定を管理画面から変更する	128
サンプルゲームでのゲームの設定ファイル	128
データストアにゲーム設定用のクラスを作成する	128
ゲーム設定を読み込む仕組み	130
ゲーム設定ファイルを使う際の注意点	132

5章 画像ファイルの保存と取得134

5-1 ファイルストアへ画像ファイルを保存する	135
スクリーンショットの撮影	135
スクリーンショットの送信と確認	139
5-2 ファイルストアから画像ファイルを取得する	141
ファイル取得メソッドの追加	141

スクリーンショットを表示するUIパーツの作成	142
ファイルストアからファイルを外部へ公開する	143
ファイルストアの扱いについて	144
5-3 アプリ内でスクリーンショットを共有・閲覧する	145
サンプルゲームでのスクリーンショットの共有	145
スクリーンショットをファイルストアに保存する	145
データストア経由で画像ファイルを取得する	148
5-4 バナー画像を管理画面からアプリに配信する	150
画像ファイルを管理画面からアップロードする	150
バナー画像が更新されたことをアプリで検出する	150
6章 プッシュ通知の実装と活用	154
6-1 ゲームにプッシュ通知を組みこむ	155
プッシュ通知とは	155
プッシュ通信の仕組み	156
プッシュ通知が可能になるまでの道のり	157
6-2 プッシュ通知の準備－UnityとNCMB編	158
NCMBSettingsの設定	158
Bundle Identifierを決める	158
NCMB管理画面の設定	160
6-3 プッシュ通知の準備－iOS編	161
iOSにおけるプッシュ通知の概要	161
Apple DeveloperでのAppID設定	161
Apple DeveloperでのAPNs証明書発行	163
NCMBへのAPNs証明書登録	166
Xcode上での設定	166
6-4 プッシュ通知の準備－Android編	170
Androidにおけるプッシュ通知の概要	170
Google Firebaseからキーを取得する	170
NCMB管理画面とUnityでの設定	172
Manifestファイルの編集	173
6-5 管理画面からプッシュ通知を配信する	175
端末情報の登録	175
管理画面からプッシュ通知を配信する	176
うまく配信できない場合のトラブルシューティング	179
6-6 アプリからプッシュ通知を配信する	181
アプリからプッシュ通知を行う流れ	181
installationのObjectIdをデータストアに保存する	181

Part2 リアルタイム・ネットワークゲーム開発ーUNET基礎編

1章 Network Managerで超簡単ネットワークゲーム！ 186

1-1 ネットワークゲームとUNETの基礎	187
ネットワークゲームの構造	187
UNETのネットワーク関連機能	188
UNETとネットワークシステム構成	189
1-2 Network Managerを使ってシーンを作成する	191
基本は「Network Manager」	191
シーンを作成する	191
Network Managerを作る	192
シーンを実行してみよう	193
1-3 キャラクターをプレファブとして作成する	195
プレファブを作る	195
ネットワーク・コンポーネントを追加する	196
プレファブをSpawn設定する	197
1-4 キャラクターを操作するためのスクリプトを記述する	199
スクリプトを作成する	199
Build&Runでアプリケーションを作る	200
1-5 キャラクターを操作するスクリプトの詳細	203
ネットワーク利用クラスとNetworkBehaviour	203
メソッドの属性について	203
プレイヤーは1つじゃない？	203
サーバー&クライアント関連の属性	204
ローカルプレイヤーかどうかのチェック	205
コマンドとリモートアクション	206
なぜ、[Command]で動かすのか？	207
1-6 仕上げとまとめ	208
オフライン時のシーンを作成する	208
Network Managerの主な機能	210

2章 Network Managerを使ったプログラミング.....212

2-1 ステータスの同期	213
ステータスの同期の概要	213
Textにステータスを表示する	214
SyncVarにフックをつける	216
2-2 クライアント間での値の共有	219
staticフィールドによる値の共有	219
ClientRpc属性について	221
サーバー起動時に初期化する	221

2-3 NetworkManagerクラスを利用する	224
ネットワーク接続の基本機能を実装するサンプル	224
ホスト・サーバー・クライアントの起動	227
NetworkClientでの接続情報の取得	227
2-4 プレイヤーオブジェクトの操作	229
プレイヤーオブジェクトの生成場所の指定	229
プレイヤーの初期設定を行う	230
オブジェクトを発射する	231
Cubeプレファブを用意する	232
プレファブをNetwork ManagerでSpawnする	233
プレファブをSpawnする手順	235
2-5 ネットワークメッセージを利用する	237
ネットワークメッセージの概要	237
メッセージの送信	239
全NetworkClientにメッセージを送る	241
独自メッセージクラスを定義する	243
クライアント側でメッセージ登録する	247
3章 UNETをカスタマイズする	250
3-1 プレイヤーが集まる場所「ロビー」の使い方	251
ロビーとは？	251
ロビーに必要なもの	252
Network Lobby Managerの作成	252
シーンを登録する	253
ロビープレイヤーを作る	254
ロビーを使ってみる	256
3-2 NetworkManagerクラスを拡張する	258
NetworkManagerクラスのコールバックメソッド	258
コールバックの流れを整理する	260
CustomNetworkManagerクラスを作る	260
Network Managerのスクリプトを変更する	262
発生するコールバックを確認する	263
3-3 NetworkLobbyManagerクラスを拡張する	265
CustomNetworkLobbyManagerクラスを作る	265
NetworkLobbyManagerのコールバックの流れ	269
3-4 実行中のユーザーを調べるネットワークディスカバリー	271
ネットワークディスカバリーとは？	271
NetworkDiscoveryを利用する	272
NetworkDiscoveryの基本設定	274
NetworkDiscoveryクラスを拡張する	274
GUIを作成する	276
実行して動作を確認する	276
NetworkDiscoveryの基本メソッド	277
NetworkDiscoveryでプログラムを実行する	278
自動実行の仕組み	280

4章 Transport Layer APIとインターネットサービス (Unity Cloud Service)282

4-1 Transport Layer APIを使ったメッセージの送受信283

Transport Layer APIとは？	283
Transport Layerによる通信の手順	283
TLScriptクラスの作成	285
TLScriptコンポーネントを用意する	286
GUIを用意する	287
送信と受信の処理を作成する	288
メッセージの送信の仕組み	291
メッセージ受信の仕組み	292

4-2 インターネットサービス(Unity Cloud Services)を利用する294

インターネットサービスとは？	294
UnityからCloud Servicesにログインする	295
クラウドサービスを使ってゲームする	298
Match Makerによるネットワークアクセス	301
Match Makerを利用する	303
MatchScriptを作成する	304
MatchScriptコンポーネントを用意する	306
MatchScriptスクリプトの概要	307

Part3 リアルタイム・ネットワークゲーム開発ーUNET実践編

1章 ネットワークゲームの基本とサンプルゲームの概要 312

1-1 UNET (Unity Networking API)のおさらい313

UNET (Unity Networking API)とは？	313
クライアント／サーバーシステム	313
サーバーの機能	314
クライアントと権限	315
権限のないオブジェクト	316
なぜクライアント／サーバーにするのか	316
ホストの機能	317
スクリプトを記述する際の注意点	318

1-2 サンプルゲーム「Painters」の概要と遊び方319

ゲームの開始とロビーの設定	319
ゲームの遊び方	321
ネットワークゲームとしての特徴	322

2章 サンプルゲーム「Painters」の実装の概要324

2-1 ネットワークゲーム開発の基本325

Unityの公式ドキュメント	325
Network Managerの配置	326

シーンの再生	327
オブジェクトの配置	328
プレイヤーオブジェクト(プレファブ)の作成	328
スタート位置のNetworkStartPosition作成	329
作成したシーンをテストする	330
プレイヤーキャラクターの実装	330
2-2 「Painters」での実装	332
プレイヤー操作の実装の原則	332
オブジェクトの初期化	333
「Painters」におけるオブジェクトの初期化の実装	334
プレイヤーの操作処理	334
プレイヤーの移動操作	335
ペイント弾の発射操作	336
3章 ゲーム時間の共有	338
3-1 遅延処理の概要	339
非同期型ゲームと遅延の問題	339
同期のアルゴリズム	339
3-2 通信の設計と実装	342
実装の考え方	342
実装の詳細	342
一定間隔ごとに送信する	343
自分の時刻を返送する	344
OnStartClientにおける受信メソッドの登録	345
OnClientReceiveGameTimeの処理	346
4章 キャラクター移動の補正	348
4-1 キャラクター移動処理の概要	349
キャラクター移動の問題点	349
キャラクター移動処理の実装の概要	350
PainterNetworkTransformによる同期の基礎	351
4-2 キャラクター移動処理の実装	352
送信側の実装	352
受信側の実装	353
4-3 キャラクター移動処理の補完と予測	355
キャラクター位置の補完	355
キャラクター位置の予測	356
5章 ペイント弾の同期と背景の塗りつぶし	360
5-1 ペイント弾を同期する	361
ペイント弾の射出の実装	361
ラグによる不公平	362

不公平の解消方法	363
サンプルゲーム「Painters」での実装	364

5-2 背景の塗りつぶし366

ペイント状況の各クライアントでの再現	366
背景ペイントの仕組み	366
背景ペイントの同期	367

Appendix

A-1 iOSアプリ開発のための証明書発行372

iOS App Development証明書の作成	372
Provisioning Profileの作成	376

A-2 Unity Cloud Buildでアプリをビルドする384

Unity Cloud Buildを利用する前準備	384
Unity Collaborateへプロジェクトファイルを送信する	385
Unity Cloud Buildの設定	386
Xcodeで設定する項目をスクリプトで行う	389

A-3 NCMBのその他の機能391

SNS連携機能でログインの手間を減らす	391
位置情報検索機能を活用して「位置ゲー」を作る	392
スクリプト機能で「ガチャ」を実装する	393
管理画面上でデータ暗号化を行い、プライバシーを守る	393
データへのアクセス権限を管理する	394
会員管理でパスワードの再発行	394
さらにアドバンスなNCMBの使い方	396
NCMBの利用事例や最新情報を知る	397

A-4 マルチプレイネットワークゲーム用のミドルウェア「Photon」398

Photonの概要	398
UNETの特徴	399
Photonの特徴	400
UNETの課題	401
Photonの課題	402

索引403

著者紹介407

コラム一覧

●序章

Unity5.6.0でのAndroidアプリのビルド 031

●Part1

APIキーについて 054
文字列データの暗号化 067
オンライン必須か、オフラインOKか? 080
OSをまたぐセーブデータの引き継ぎと、法的な問題
..... 089
より良いIDとパスワードの自動生成とは 094
APIコール数の節約 106
NCMBの活用事例「データストア」編① 119
メッセージから不適切な言葉をフィルタする 127
NCMBの活用事例「データストア」編② 132
ファイルストア機能はAsset Bundleには利用できない
..... 135
ファイルストアの応用 144
端末のローカル領域にファイルを保存する 153
NCMB活用事例「プッシュ通知編」① 157
NCMB活用事例「プッシュ通知編」② 169
AndroidにおけるNCMB SDKとほかのサービスとの
併用 174

●Part2

HLAPIに用意されている各種のコンポーネント... 202
エラーの際に見直すポイント 236
「クライアント」か、「サーバー」か、それが問題だ!
..... 249
out修飾子について 292
ゲームとファイアウォール 300

●Part3

同期型と非同期型のネットワークゲーム 323
劣悪な通信環境の場合の対応 343
UNETの将来 347
通信量を減らすためには 370

●Appendix

キーチェーンアクセスに証明書が登録できない場合
..... 375
Bundle IDのワイルドカード指定について 378
GithubやBitbucketの設定 385
Windows環境だけでiOSアプリを開発できる? 390
プランの利用上限を超えた時に備える 395

本書では、非リアルタイム型ゲームのサンプルとして『剣士なんですすぐ死んでしまうん?』を、リアルタイム型のゲームサンプルとして『Painters』を用意しました。それぞれのゲームの遊び方や具体的な実装方法については、以降のPartで紹介していますので、ここではゲームの概要について触れておきます。

サンプルゲーム1：『剣士なんですすぐ死んでしまうん?』

『剣士なんですすぐ死んでしまうん?』は、横持ち型のアクションゲームです。操作はバーチャルパッド方式になります。ゾンビのさまよう墓場で剣士が果敢に戦う...という内容です。



図0-6 剣士とゾンビの対戦



図0-7 残念ながら、死んでしまった！

主人公が敵にやられるとその場所に自分のお墓ができます。お墓にはメッセージと「魔法効果」を設定できます。

メッセージや魔法効果、お墓の位置などの情報は、「ニフティクラウド mobile backend」を使ってサーバー側に保存されます。ほかのプレイヤーがゲームを開始すると、同じ位置にお墓が現れ、そのお墓を調べるとメッセージの表示と魔法効果を受ける仕組みになっています。

また、メニュー画面ではスコアランキングやフレンド機能などの、ソーシャルゲームでよくある機能が実装されています。



図0-8 メニュー画面でフレンドやランキングを閲覧できる

サンプルゲーム2：『Painters』

『Painters』は、LANやインターネットで相手を探して、対戦できるリアルタイム型のネットワークゲームです。操作はキーボードの矢印キーとスペースキーで行います。



図0-9 インターネットとLANで遊べる

スペースキーでペイント弾を発射して、多くのエリアを自分の色で塗ったほうが勝ちというゲームです。また、敵にぶつかって行ってダメージを与えることもできます。なおプレイ時間は、180秒です。



図0-10 Paintersのゲーム画面1



図0-11 Paintersのゲーム画面2