

3D ゲームプログラミング II / ゲームプログラミング B

第 6 回 ネットワーク対応ゲーム

Unity のサイトにあるチュートリアル、マルチプレイヤーネットワーキングに基づき、ネットワーク対応ゲームを作成する。

Unity のトップページから、「学ぶ」→「チュートリアル」→「マルチプレイヤーネットワーキング」を選ぶ。

1. Introduction to a Simple Multiplayer Example から 18. Simple Game Summary まですを実行する。

ポイントと修正点

2. ネットワークマネージャー

- ・ ネットワーク対応ゲームには、NetworkManager コンポーネントが必要である。
- ・ NetworkManagerHUD は NetworkManager のユーザーインターフェイスを提供する。

3. Player プレハブのセットアップ

- ・ Player オブジェクトはプレハブ化しておく必要がある。
- ・ Player オブジェクトに NetworkIdentity コンポーネントを追加し、Local Player Authority を有効にしておく。

4. プレイヤープレハブを登録する

- ・ NetworkManager の PlayerPrefab に Player オブジェクトのプレハブを登録する。

6. クライアント上でプレイヤーの動きをテストする

- ・ アプリケーションを 2 つ同時に起動し、Host と Client とする。
- ・ Host は、Server と Client を 1 台の PC 上で実行する。

7. プレイヤーの動きをネットワーク化する

- ・ スクリプトには、名前空間 UnityEngine.Networking を追加する。
- ・ MonoBehaviour ではなく、NetworkBehaviour の派生クラスとする。
- ・ Player オブジェクトを同期させるためには、NetworkTransform コンポーネントを追加する。

9. ローカルプレイヤーの特定

- ・ 次の箇所を修正する。

(修正前) GetComponent().material.color = Color.blue;

(修正後) GetComponent<Renderer>().material.color = Color.blue;

11. マルチプレイヤーシューティングを追加する

- ・ 弾のプレハブに NetworkIdentity コンポーネントと NetworkTransform コンポーネントを追加する。
- ・ [Command]の付いたメソッドは、クライアントから呼び出され、サーバーで実行される。
- ・ [Command]属性のメソッド名は、Cmd から始まらなければならない。

12. プレイヤーの体力【シングルプレイヤー】

- ・ TakeDamage メソッドで、文字化けしていたら、次のように修正する。

(修正前) if (currentHealth <= 0)

(修正後) if (currentHealth <= 0)

- ・ OnCollisionEnter メソッドで、次のように修正する。

(修正前) var health = hit.GetComponent();

(修正後) var health = hit.GetComponent<Health>();

- ・ Background の RectTransform の PosX, PosY, PosZ を(0, 0, 0)にする。
- ・ Foreground の RectTransform の PosX, PosY, PosZ を(100, 0, 0)に、Pivot の x, y を(1, 0.5)にする。

13. プレイヤーの体力をネットワーク化する

- ・ 同期させる変数には、[SyncVar]を付ける。
- ・ [SyncVar]の付いた変数がサーバーで変更されると、クライアントでも変更される。
- ・ [SyncVar(hook = "メソッド名")]とすると、変数が変更されたときにメソッドが実行される。引数としてその変数の値が渡される。

14. 死と再生成

- ・ [ClientRpc]の付いたメソッドは、サーバーで呼び出され、クライアントで実行される。
- ・ [ClientRpc]属性のメソッド名は、Rpc から始まらなければならない。

15. 非プレイヤーオブジェクトをハンドルする

- ・ 文字化けしていたら、次のように修正する。

(修正前) for (int i=0; i <= numberOfEnemies; i++)

(修正後) for (int i=0; i < numberOfEnemies; i++)

17. 生成と再生成

- ・ **Player** オブジェクトが出現する位置は、**NetworkStartPosition** コンポーネントを持つ複数のオブジェクトの位置から選ばれる。