



Unityはじめるよ ～NavMesh応用 2～

統合開発環境を内蔵したゲームエンジン
<http://japan.unity3d.com/>

※いろんな職業の方が見る資料なので説明を簡単にしている部分があります。正確には本来の意味と違いますが上記理由のためです。ご了承ください。
この資料内の一部の画像、一部の文章はUnity公式サイトから引用しています。

NavMeshおさらい

- 自動で移動するNPCが作れるよ。
- 障害物を自動で避けながら目標地点にキャラを移動させるなんてこともできるよ。

非常に少ない手順で上記のことが叶う
Unityの素敵な機能なのさ。

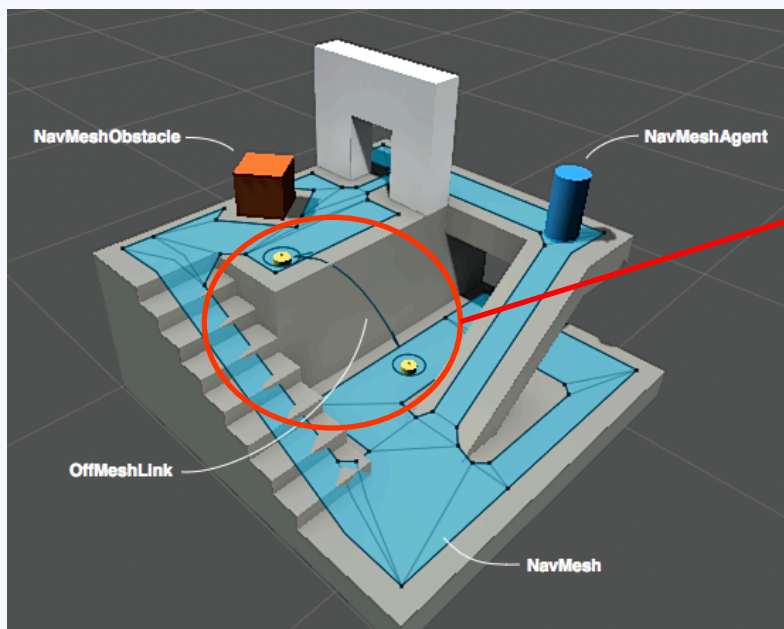
■現時点での問題点

- ・ オフメッシュリンク間の移動が不自然
スーッと移動しちゃってるね。
- ・ 自動でオフメッシュリンクを作ると、
飛び降り側は作成できるが、飛び乗り側が作成されない。

オフメッシュリンクのおさらい

オフメッシュリンク (Off-Mesh Link)

ナビメッシュだけでは越えられない段差や溝を超えるようにする仕組み



オフメッシュリンク

■ 解決方法

- ・ オフメッシュリンク間の移動が不自然
スーッと移動しちゃってるね。
➤ スクリプトで制御すればなんとかなる
- ・ 自動でオフメッシュリンクを作ると、
飛び降り側は作成できるが、飛び乗り側が作成されない。
➤ 飛び乗りは手動で作成する

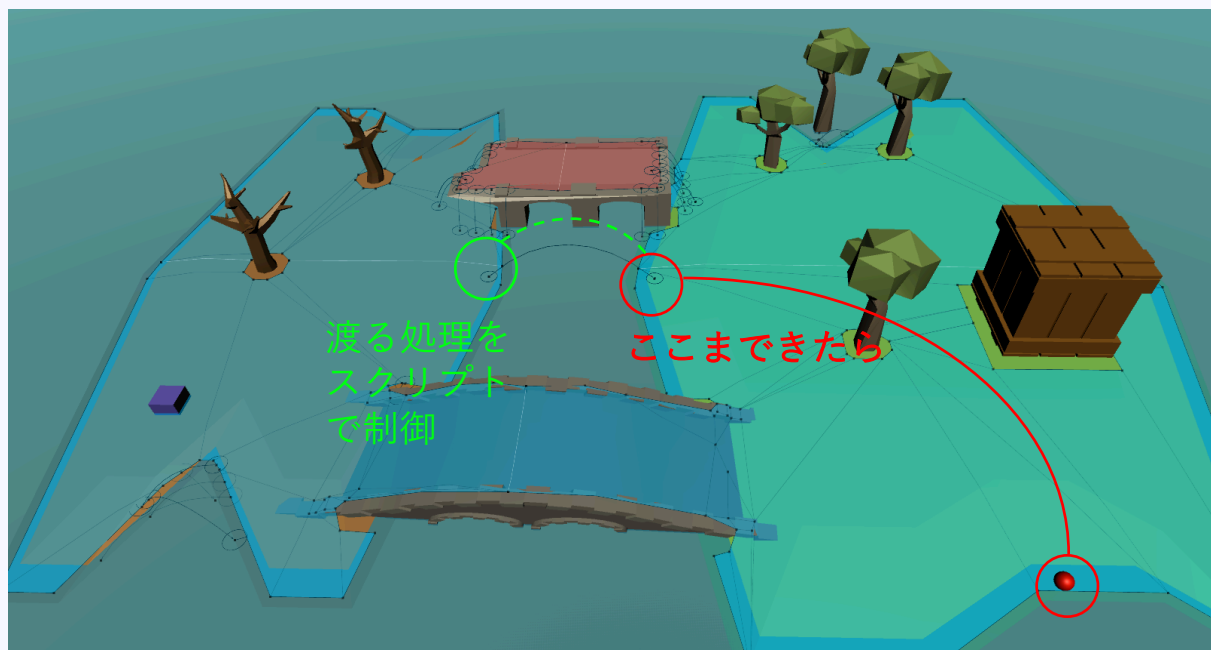
それでは順番に見ていきましょう

まずは

オフメッシュリンク間の移動が不自然
にスーッと移動しちゃってるね問題

NavMeshAgentをスクリプトで制御

キャラクターがオフメッシュリンクの始点にたどり着いたら、
終点までの移動をスクリプトで行うことで、
不自然な動きを解消することができる



ポイント

- 1、自動でオフメッシュリンクを超えないように NavMeshAgentの設定を変更する
- 2、NavMeshAgentがオフメッシュリンクにたどり着いたか調べる
- 3、スクリプトでオフメッシュリンクの終点まで移動
- 4、NavMeshAgentにオフメッシュリンクを超えたことを通知

1、自動でオフメッシュリンクを超えないように NavMeshAgentの設定を変更する

今回はStart関数で設定を変更しておきます。
NavMeshAgentの

bool autoTraverseOffMeshLink

というプロパティを**false**にすると
自動でオフメッシュリンクを超えなくなる。

例)

```
NavMeshAgent    m_navMeshAgent;    /// NavMeshAgent

void Start () {
    // オフメッシュリンクを自動で飛び越えないようにする
    m_navMeshAgent.autoTraverseOffMeshLink = false;
}
```

2、NavMeshAgentがオフメッシュリンクにたどり着いたか調べる

NavMeshAgentの

bool isOnOffMeshLink

というプロパティが**true**なら
オフメッシュリンクの始点にたどり着いていることになる

例)

```
// オフメッシュリンクに辿り着いているなら
if (m_navMeshAgent.isOnOffMeshLink) {
    // オフメッシュリンクを超える処理をここに書く
}
```

3、スクリプトでオフメッシュリンクの終点まで移動

スクリプトで移動させるためには、
始点と終点を知る必要がある。
NavMeshAgentの

currentOffMeshLinkData

に現在たどり着いているオフメッシュリンクの情報が入っている。

例)

// オフメッシュリンクの情報を取得

```
OffMeshLinkData data = m_navMeshAgent.currentOffMeshLinkData;
```

// 始点は data.startPos で取得可能

// 終点は data.endPos で取得可能

始点と終点の情報が取れたら、次は移動処理。

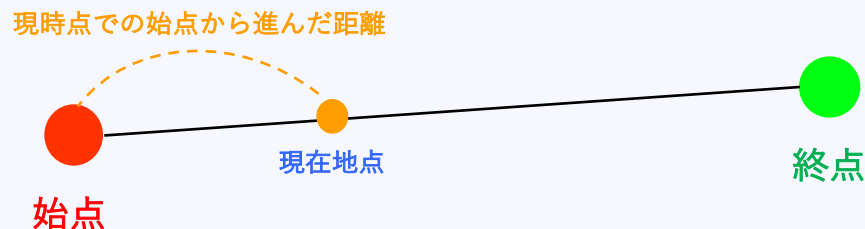
移動方法はゲームごとオリジナルの実装になるけど、始点から終点まで確実に移動させる方法として、

Vector3クラスのMoveTowards()関数

を使うと簡単かも。

```
public static Vector3 MoveTowards(Vector3 current, Vector3 target, float maxDistanceDelta);
```

引数に、始点、終点、現時点での始点から進んだ距離を渡すと、現在地点を返してくれる。



4、NavMeshAgentにオフメッシュリンクを超えたことを通知

NavMeshAgentの

CompleteOffMeshLink ()

という関数を呼ぶだけ。
この関数を呼ばないと、NavMeshAgentはいつまでたっても自動で移動を開始してくれません。

例)

```
// オフメッシュリンクの移動を完了させたことをAgentに通知  
m_navMeshAgent.CompleteOffMeshLink ();
```

```
using UnityEngine;
using System.Collections;
```

```
// サンプルなので、移動処理はきっちり作っていません。
```

```
public class DogController : MonoBehaviour {

    public GameObject TargetObject; // 目標位置
    public float Jump = 20.0f; // エージェントのジャンプ力
    NavMeshAgent m_navMeshAgent; // NavMeshAgent

    /// オフメッシュリンク移動制御用構造体
    struct OffMeshLinkMove {
        public bool isMoveOnTheOffMeshLink; // オフメッシュリンクを移動中か
        public float moveTime; // オフメッシュリンク間移動にかける時間
        public float nowDistance; // オフメッシュリンク間移動の現時点での移動距離
        public float distanceOfOffMeshLinks; // オフメッシュリンクの距離
        public float moveSpeed; // エージェントの実移動速度
        public Vector3 startPos; // オフメッシュリンクの始点
        public Vector3 endPos; // オフメッシュリンクの終点
    };
    OffMeshLinkMove m_offMeshLinkMove; // オフメッシュリンク移動制御用構造体

    // Use this for initialization
    void Start () {

        // NavMeshAgentコンポーネントを取得
        m_navMeshAgent = GetComponent<NavMeshAgent>();

        // オフメッシュリンクを自動で飛び越えないようにする
        m_navMeshAgent.autoTraverseOffMeshLink = false;

        // オフメッシュリンク移動制御用構造体を初期化
        InitOffMeshLinkMove ();
    }

    // Update is called once per frame
    void Update () {

        // NavMeshが準備できているなら
        if(m_navMeshAgent.pathStatus != NavMeshPathStatus.PathInvalid) {

            // NavMeshAgentに目的地をセット
            m_navMeshAgent.SetDestination(TargetObject.transform.position);

            // オフメッシュリンクに辿り着いているなら
            if (m_navMeshAgent.isOnOffMeshLink){

                // オフメッシュリンクの情報を取得
                OffMeshLinkData data = m_navMeshAgent.currentOffMeshLinkData;

                // オフメッシュリンク移動の初期化処理-----
                if (m_offMeshLinkMove.distanceOfOffMeshLinks == 0) {
                    // オフメッシュリンクの距離を求める
                    m_offMeshLinkMove.distanceOfOffMeshLinks = Vector3.Distance (data.startPos, data.endPos);

                    // 目的地の方向を向かせる
                    transform.LookAt (new Vector3 (data.endPos.x, transform.position.y, data.endPos.z));
                }
                // オフメッシュリンク移動の初期化処理-----

                // 移動処理-----
                // 現時点での移動距離を求める
                m_offMeshLinkMove.nowDistance += Time.deltaTime * m_offMeshLinkMove.distanceOfOffMeshLinks / m_offMeshLinkMove.moveTime;

                // 移動開始地点と目標地点を指定して移動 3つめの引数では現時点での移動開始地点からの移動距離を指定する
                Vector3 pos = Vector3.MoveTowards (data.startPos, data.endPos, m_offMeshLinkMove.nowDistance);

                // Sin波を使って放物線を描く
                float degree = (m_offMeshLinkMove.nowDistance / m_offMeshLinkMove.distanceOfOffMeshLinks * 180.0f);
                pos.y = data.startPos.y + (Mathf.Sin (Mathf.Deg2Rad * degree) * Jump);

                // 下降を始めた時に終点よりき防ぐ
                if (degree > 90.0f) {
                    if (pos.y < data.endPos.y) {
                        pos.y = data.endPos.y;
                    }
                }

                // 座標を反映
                transform.position = pos;
                // 移動処理-----

                // 移動完了判定-----
                if ((transform.position.x == data.endPos.x) && (transform.position.z == data.endPos.z)) {
                    m_navMeshAgent.CompleteOffMeshLink (); // オフメッシュリンクの移動を完了させたことをAgentに通知
                    InitOffMeshLinkMove (); // オフメッシュリンク移動制御用構造体を初期化
                }
                // 移動完了判定-----
            }
        }
    }
}
```

続き

```
// オフメッシュリンク移動制御用構造体を初期化
void InitOffMeshLinkMove () {
    m_offMeshLinkMove.isMoveOnTheOffMeshLink = false;
    m_offMeshLinkMove.moveTime = 1.0f;
    m_offMeshLinkMove.nowDistance = 0;
    m_offMeshLinkMove.distanceOfOffMeshLinks = 0;
    m_offMeshLinkMove.moveSpeed = 0;
}
```

プロジェクト一式は、
[http://monolizm.com/sab/src/NavMeshTest\(応用編2\)完成版.zip](http://monolizm.com/sab/src/NavMeshTest(応用編2)完成版.zip)

次は

自動でオフメッシュリンクを作ると、
飛び降り側は作成できるが、
飛び乗り側が作成されない問題

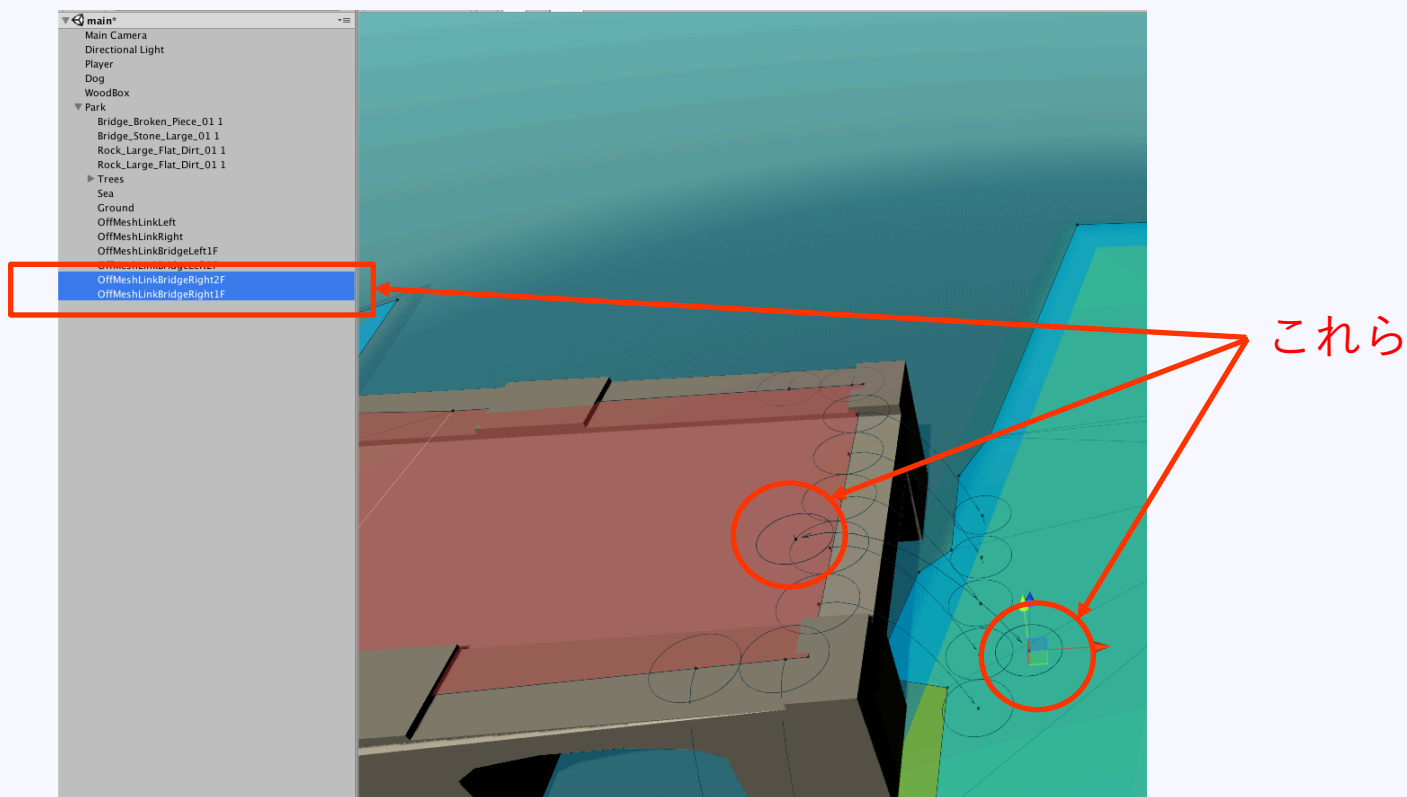
手動で作るしかない

自動でオフメッシュリンクを作成するGeneratedOffMeshLinksの

設定項目「**JumpDistance**」は

飛び乗る高さを表しているものではなく

飛び越える距離を表すプロパティでした。



前回のプレゼンで紹介したやつは、
手動で作ってありました・・・。
すっかり忘れてました・・・。
申し訳ないです<(_ _)>

■参考サイト

Unity公式マニュアル

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

Unityを使った3Dゲームの作り方

<http://gametukurikata.com/navigation/offmeshlink>

ご清聴ありがとうございました