



Unityはじめるよ ～NavMesh応用～

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ある部分があります。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。

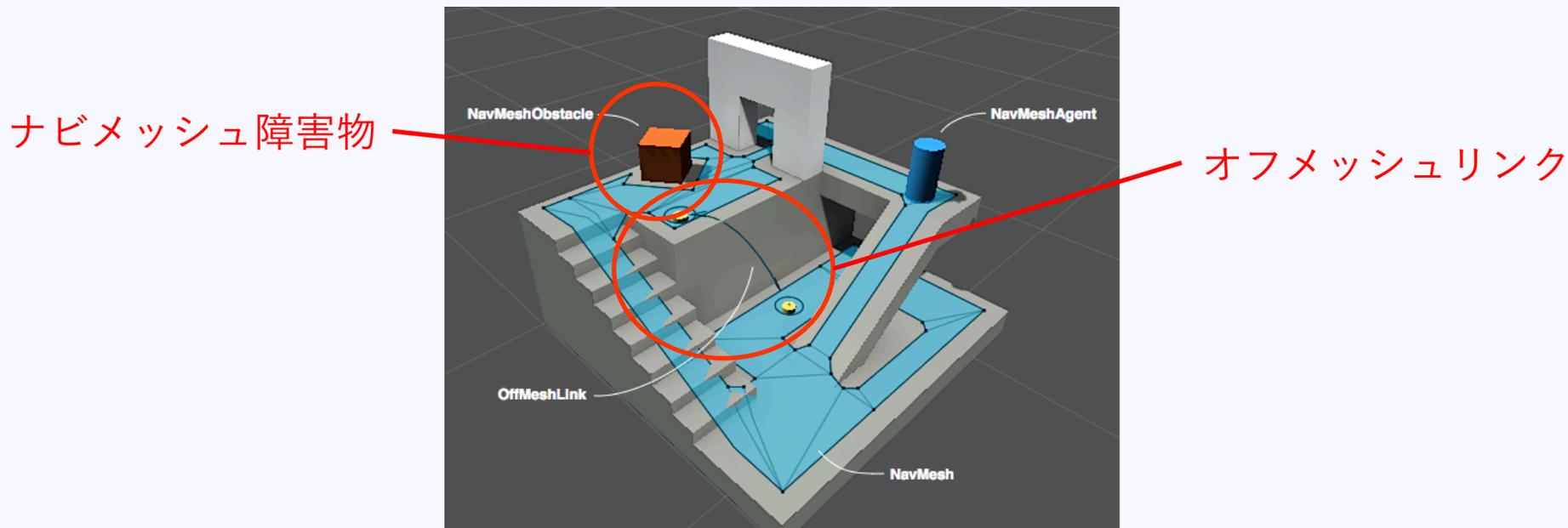
今回はオフメッシュリンクと ナビメッシュ障害物をやるよ

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

ナビメッシュ障害物 (NavMeshObstacle)

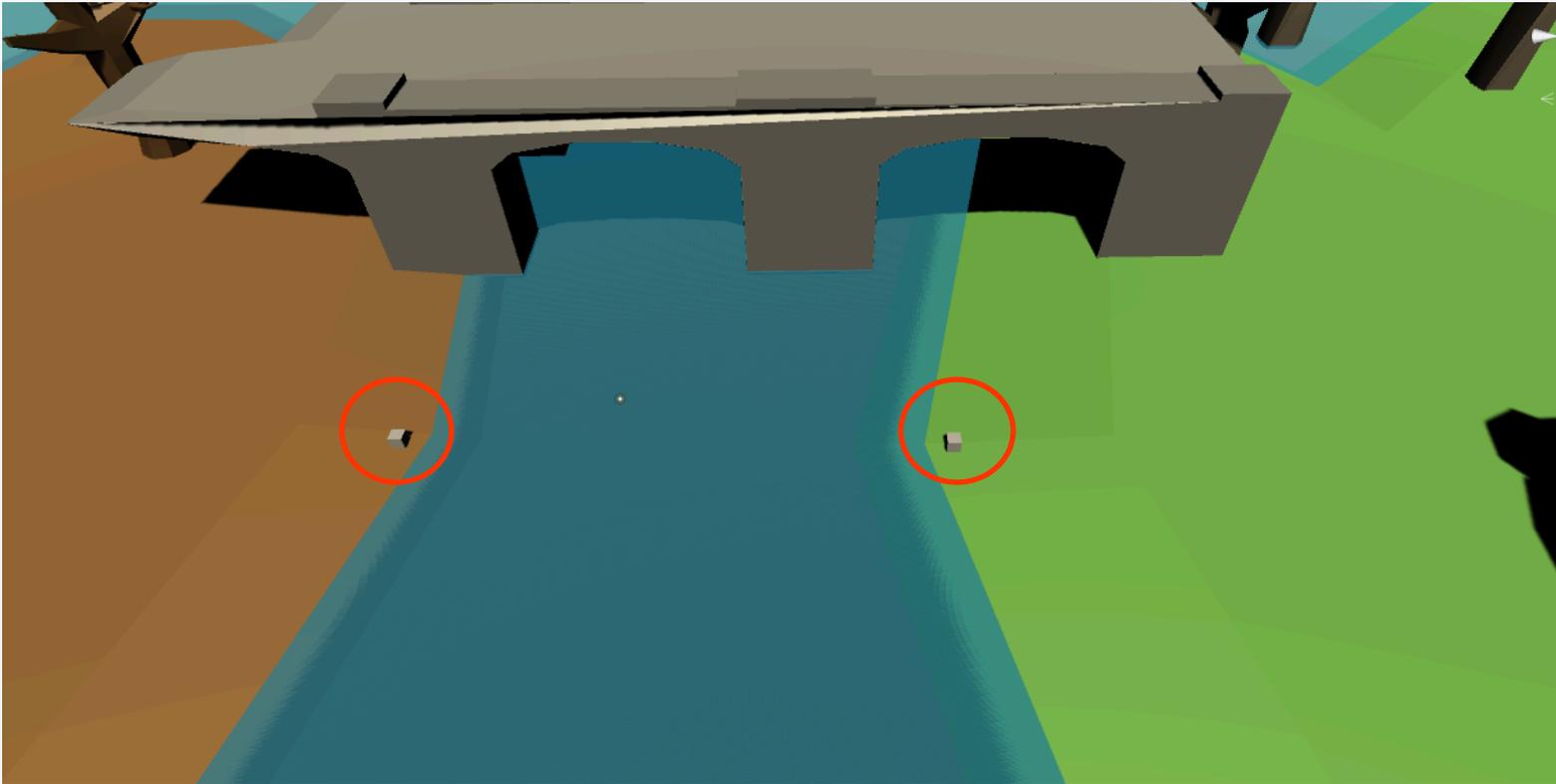
動く障害物を実現する仕組み



それじゃ順番に見ていきましょう

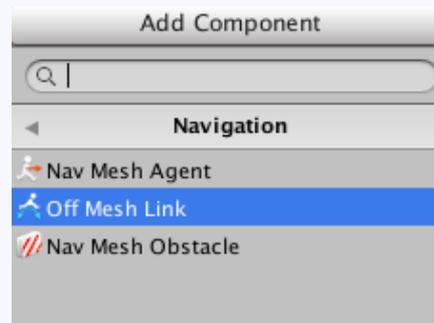
1. オフメッシュリンクを手動で作成
2. オフメッシュリンクを自動で作成
3. ナビメッシュ障害物
4. 移動コストについて
5. NavMeshAgent利用時の注意点

1. オフメッシュリンクを 手動で作成

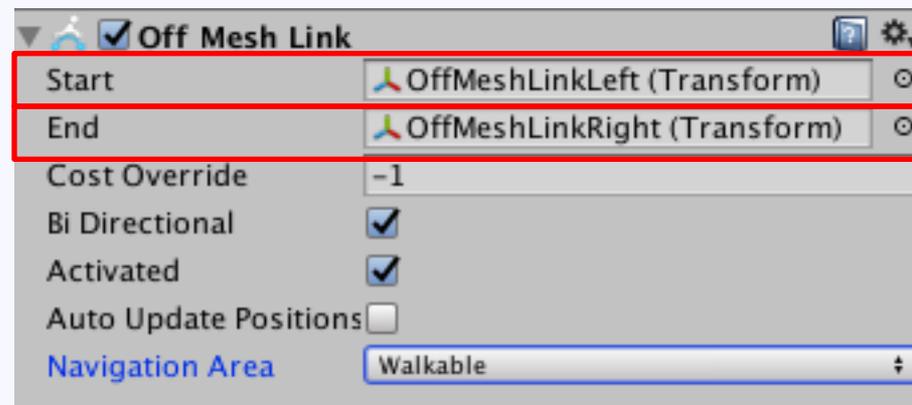


オフメッシュリンクを作りたいところに、
GameObjectを置く。
※わかりやすくする為にCubeにしてあるが
空のGmeObjectが良い

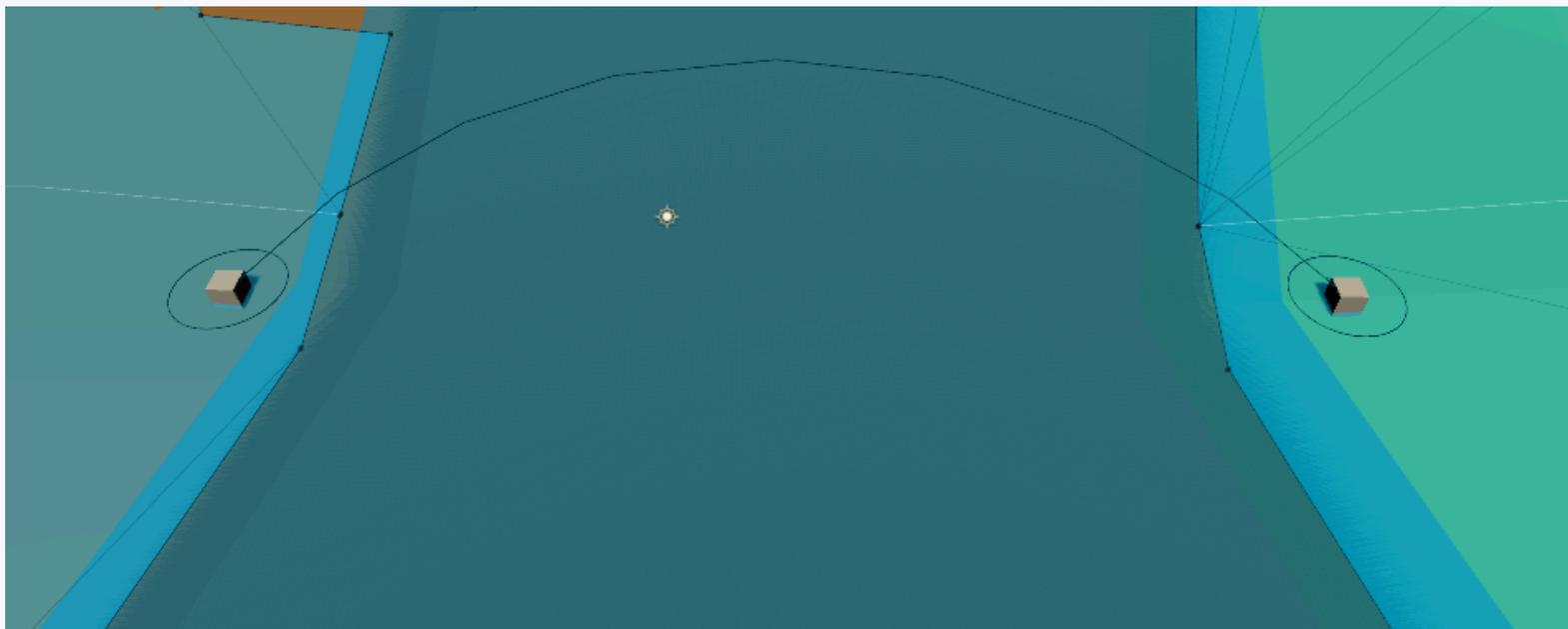
2つのGameObjectに**OffMeshLinkコンポーネント**を追加。



Startに自分のGameObject(Transform)、
Endにもう片方（移動先）のGameObject(Transform)
を指定する。



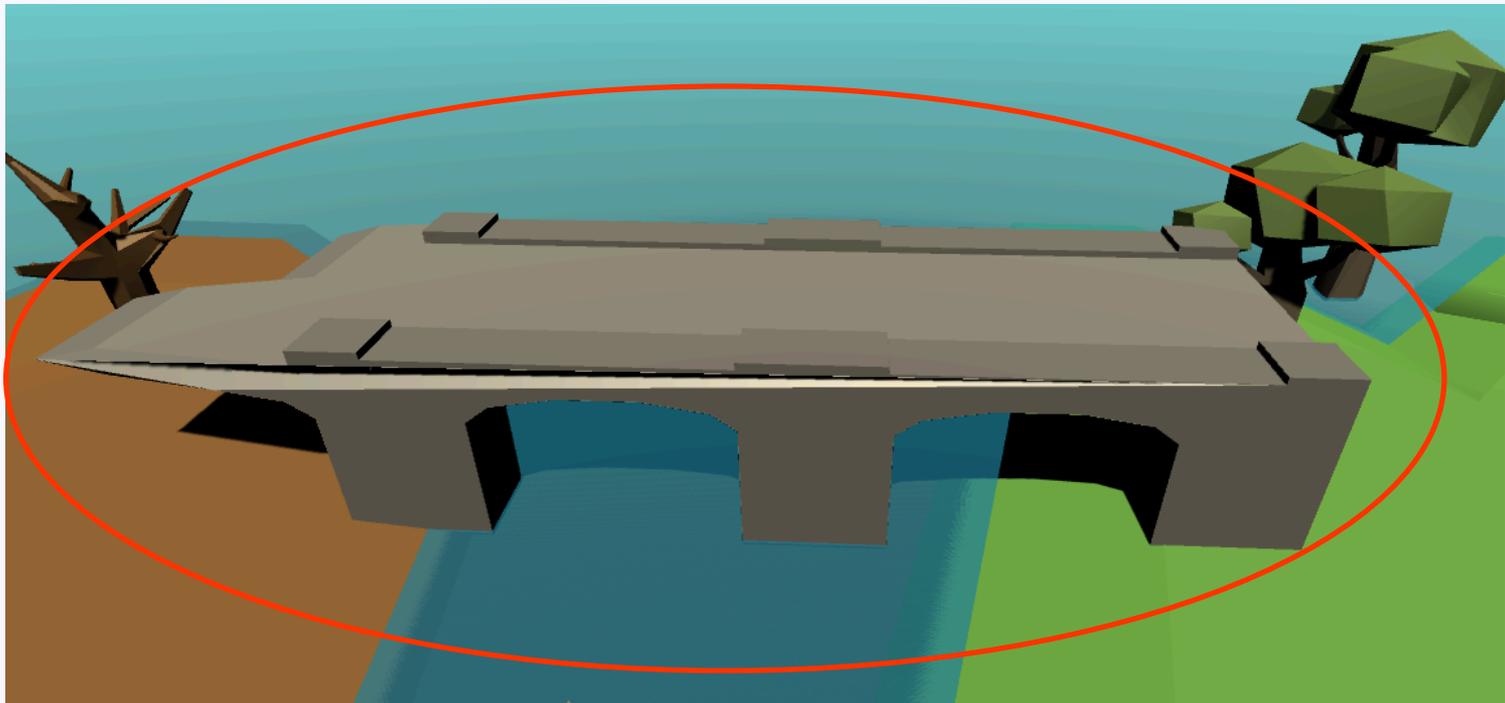
あとは、いつものように**Bake**するだけ！



線で結ばれ、オフメッシュリンクが作成された。

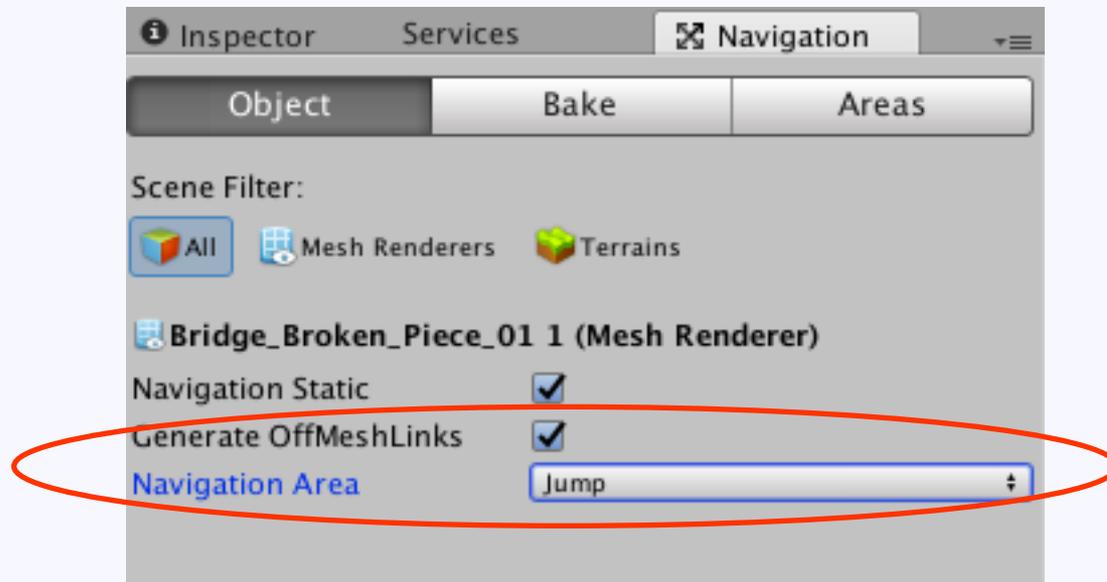
Unityで動作確認！

2. オフメッシュリンクを 自動で作成

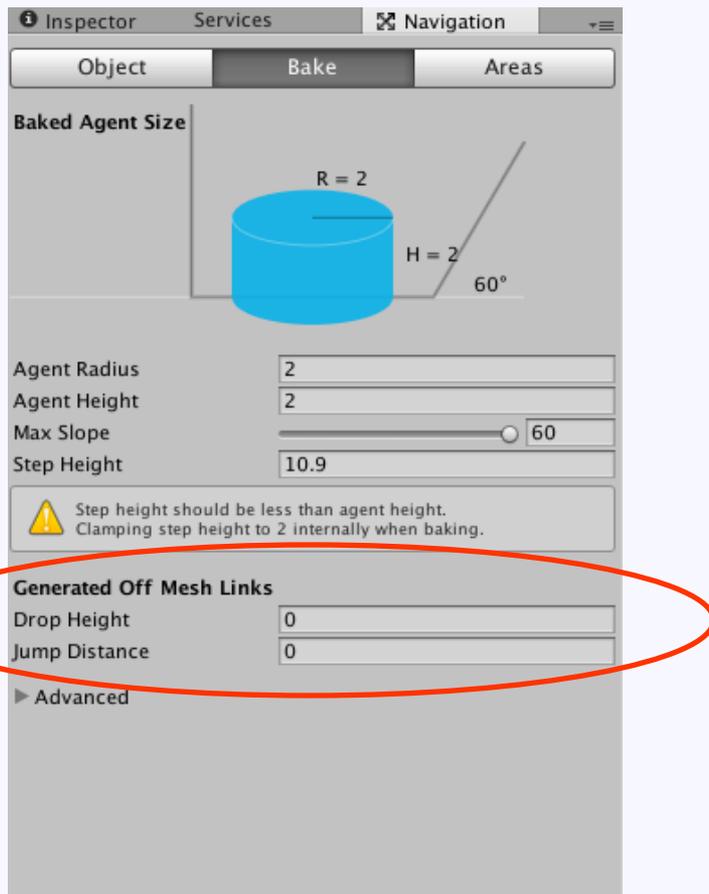


まずは、段差の激しい土台を用意する。
今回はここに飛び乗ったり、
ここから飛び降りたりできるようにする。

インスペクタービューで土台を選択し、NavigationウィンドウのObjectタブを開き、**GenerateOffMeshLinks**にチェックを入れ、**NavifgationArea**を**Jump**にしておく。



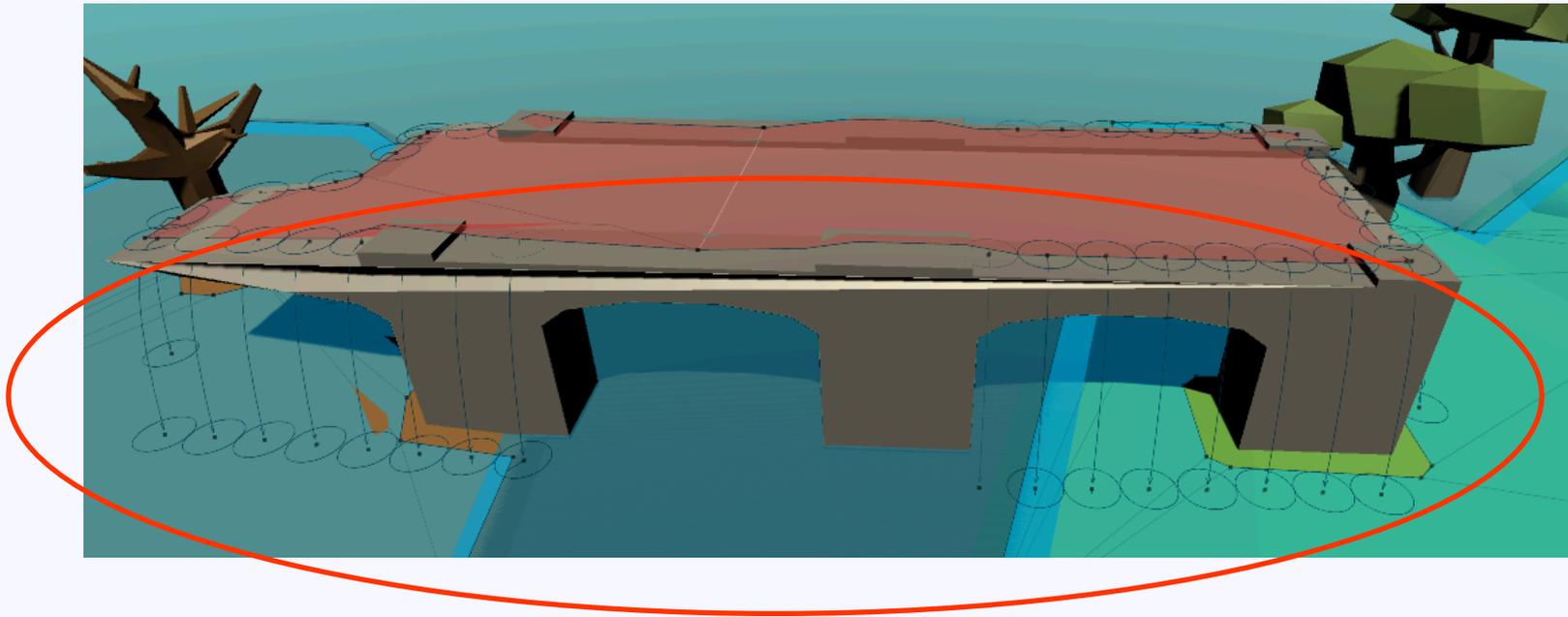
NavigationウィンドウのBakeタブを開き、Generated Off Mesh Links項目の、**Drop Height** と **Jump Distance** に値を入れBake。



Drop Heightは、
Agentが飛び降りれる高さ

Jump Distanceは、
Agentが飛び乗れる高さ

オフメッシュリンクがたくさん作成されたぜ！



Unityで動作確認！

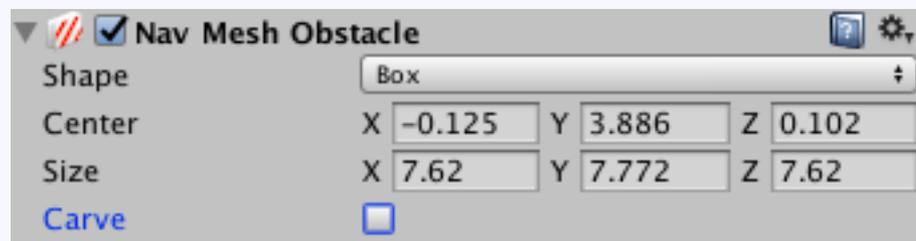
3.ナビメッシュ障害物

物理演算が働くGameObjectを
ナビメッシュ障害物にしてみる。

まずは、GameObjectを作成し、
Rigidbodyをアタッチ。



そこに、
NavMeshObstacleコンポーネントをアタッチ。



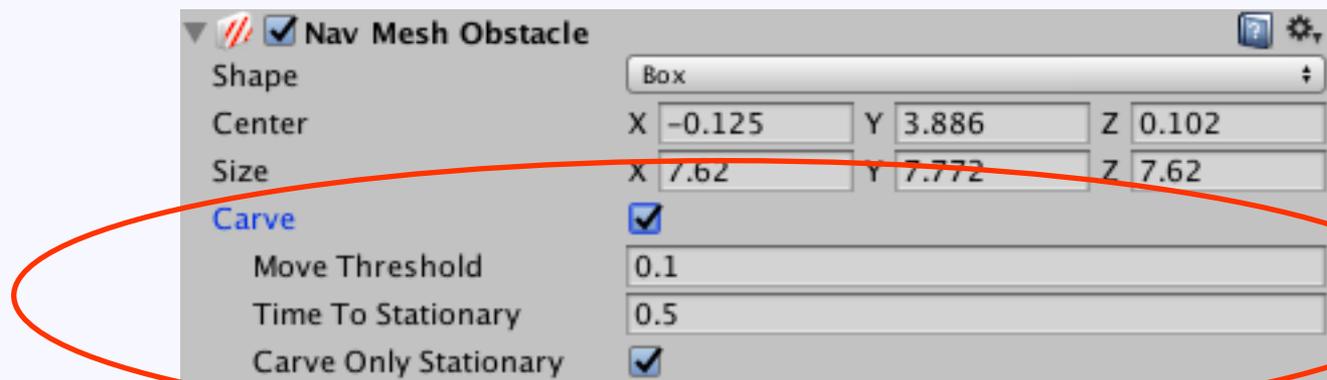
AddComponent > Navigation > NavMeshObstacle

Carveプロパティについて。

- ・ 車など、常に動くものはオフにしておく
- ・ 木箱など、普段は静止しているが、
外的要因で動くものはオンにしておく。

理由はマニュアルを見てもらうとして、簡単に説明すると、
処理速度の問題と、別ルートを算出する方法の違いがある。

今回は木箱を置いたので、Carveのチェックをオンに。



NavMeshObstacleコンポーネントのプロパティは表の通り

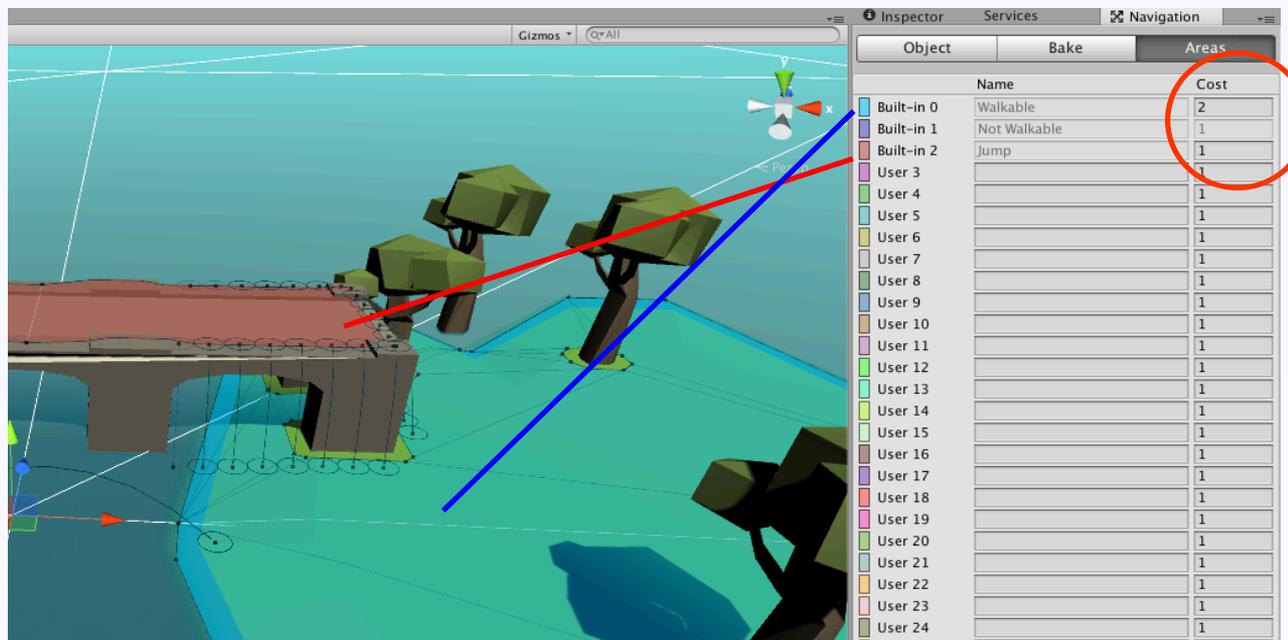
プロパティ	説明
Shape	障害物のジオメトリの形: <i>capsule</i> または <i>box</i>
<i>box</i> の場合	
Center	トランスフォームの位置に対する相対的な、ボックスの中心
Size	ボックスの大きさ
<i>capsule</i> の場合	
Center	トランスフォームの位置に対する相対的な、カプセルの中心
Radius	カプセルの半径
Height	カプセルの高さ
Carve	オンになっていると障害物によってナビメッシュ上に穴が作られます。
<i>Carving</i> がオンの場合	
Move Threshold	動いている穴が更新される距離のしきい値
Time To Stationary	障害物が静止物として扱われるまでの待ち時間
Carve Only Stationary	有効になっていると、障害物は静止時にのみ穴をつくります。

Unityで動作確認！

4. 移動コストについて

Agentは目標地点に向けて最速のルートを選ぶ。
つまり、距離が短くてもコストが高いルートは
選ばれない可能性もある。
ルート選択の計算式はシンプルで、
移動距離×コストとなる。

コストの設定は、
Navigationウィンドウの**Areasタブ**で設定可能。
NavigationArea(オフメッシュの色が違う)毎に設定可能。



5 .NavMeshAgent利用時の注意点

NavMeshAgentはGameObjectの移動を行う。

その為、GameObjectを移動させる機能を持った他のコンポーネントと同時に使うと、**競合が起こり正常な動作にならない**。

GameObjectを移動させる代表的なコンポーネント

- **Rigidbody**
同時に使う時はIsKinematicをオンにすること
- **Animator**
同時に使う時はRootMotionを使わない、または、移動処理の順番を一定にする。
(Agentが先かAnimatorが先か)

詳細は公式ドキュメントを参照

<https://docs.unity3d.com/ja/current/Manual/nav-MixingComponents.html>

■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
スーツ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

■まとめ

オフメッシュリンクを利用すれば、あらゆる地形をNPCが移動可能に！

ナビメッシュ障害物で、動く障害物も作成可能！

■参考サイト

Unity公式マニュアル

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

ウダサンコウボウ

http://udasankoubou.blogspot.jp/2014/01/unity_28.html

ご清聴ありがとうございました