内容

[**１．作品概要** 2](#_Toc30493567)

[1.1 タイトル 2](#_Toc30493568)

[1.2 対応ハード 2](#_Toc30493569)

[1.3 製作人数 2](#_Toc30493570)

[1.4開発期間 2](#_Toc30493571)

[1.5開発環境 2](#_Toc30493572)

[1.6 使用言語 2](#_Toc30493573)

[**２．ナビゲーションメッシュの生成** 3](#_Toc30493574)

[**３．A\*経路探査とスムージング** 4](#_Toc30493575)

[**3.1 スムージングのアルゴリズム** 5](#_Toc30493576)

[**４．アニメーションブレンディング** 6](#_Toc30493577)

# **１．作品概要**

## 1.1 タイトル

Break Down Tank(ブレイクダウンタンク)

## 1.2 対応ハード

PC Windows10

## 1.3 製作人数

　１人

## 1.4開発期間

2019年8月～

## 1.5開発環境

エンジン

　 学校内製の簡易エンジン(DirectX11)

ツール

Microsoft VisualStudio2017

3DSMAX2018

FireAlpaca

Git

## 1.6 使用言語

C++

# **２．ナビゲーションメッシュの生成**

経路探査を行うためのナビゲーションメッシュをプログラムで自動生成しています。

３Dモデルの頂点バッファとインデックスバッファの情報をもとに地形データを生成します。

地形データと障害物で当たり判定をとって、衝突したポリゴンを除いて、残ったポリゴンをナビゲーションメッシュのセルとしてリストに登録しています。

当たり判定はBullet Physicsを使用しています。

図１(障害物判定を行わない場合のナビゲーションメッシュ)

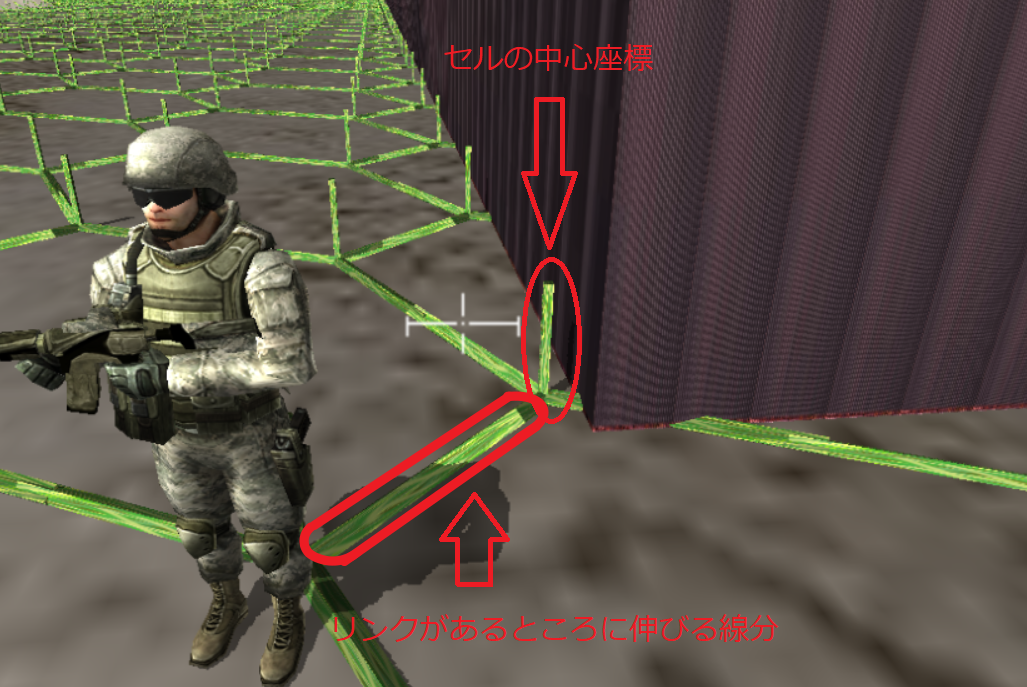


図２(障害物判定を行った場合のナビゲーションメッシュ)

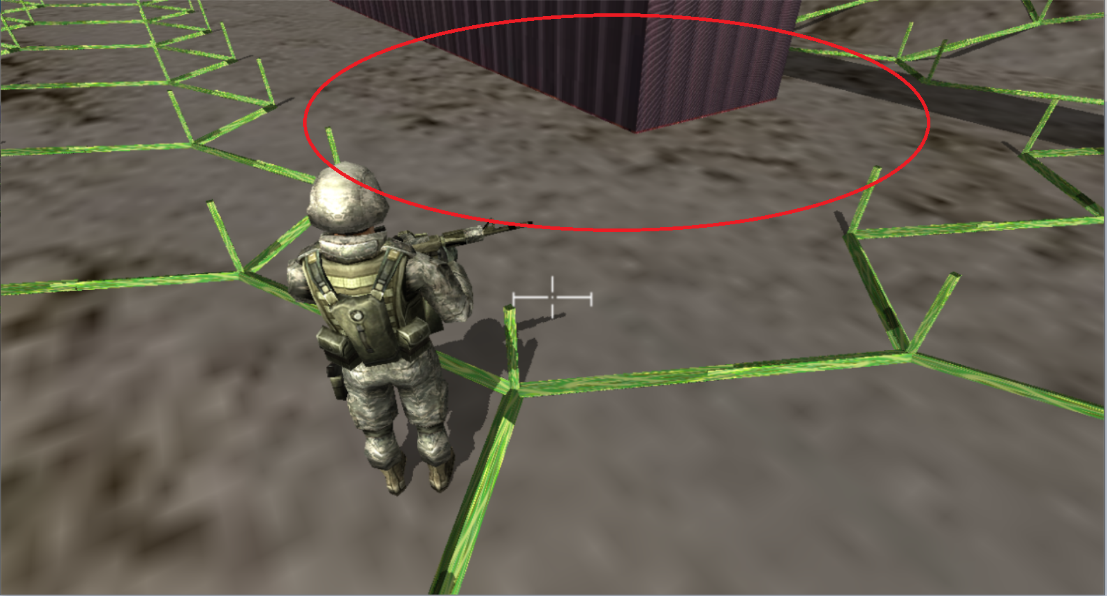
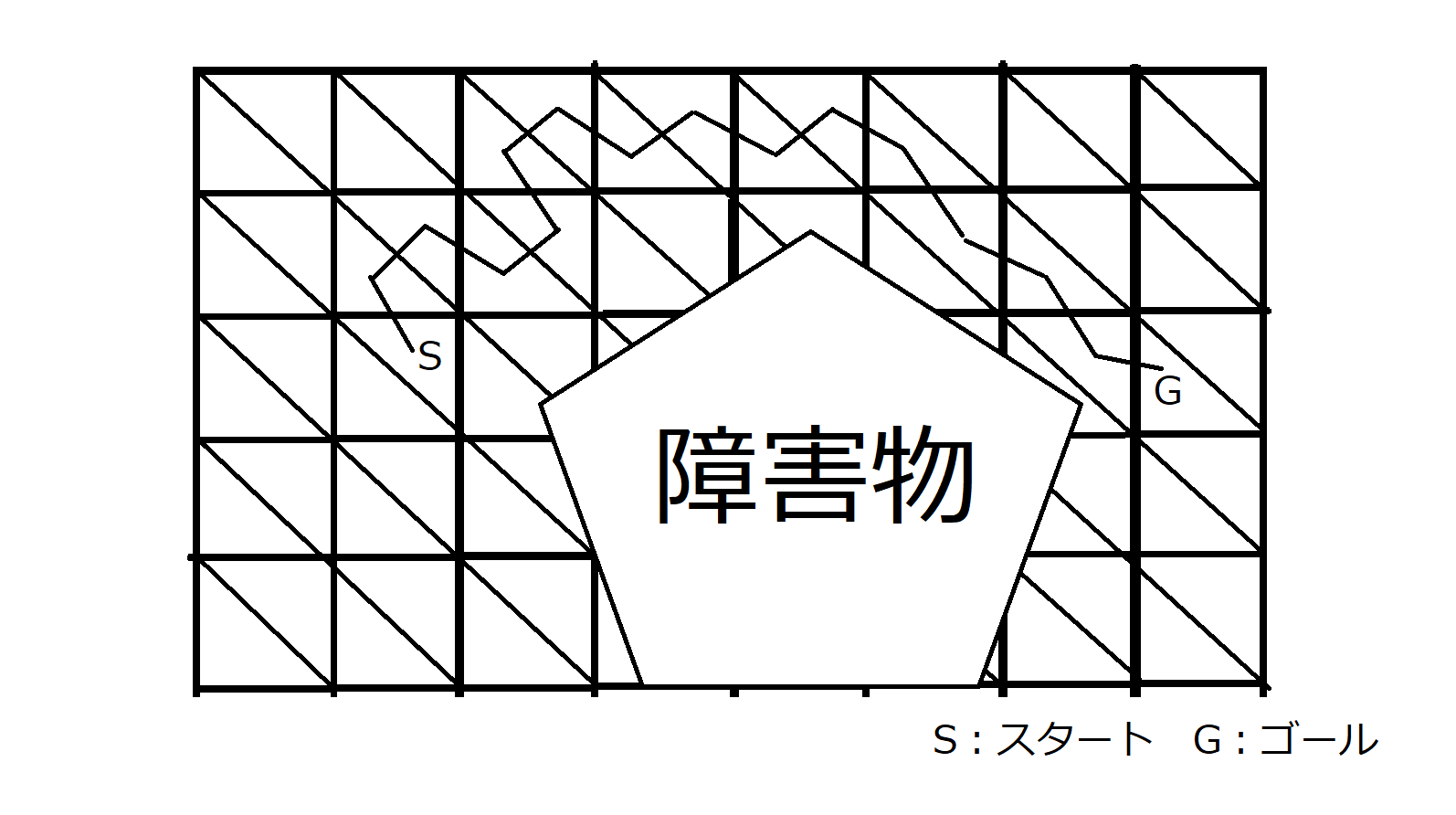


図2の壁付近のセルが除去されたことがわかります。

# **３．A\*経路探査とスムージング**

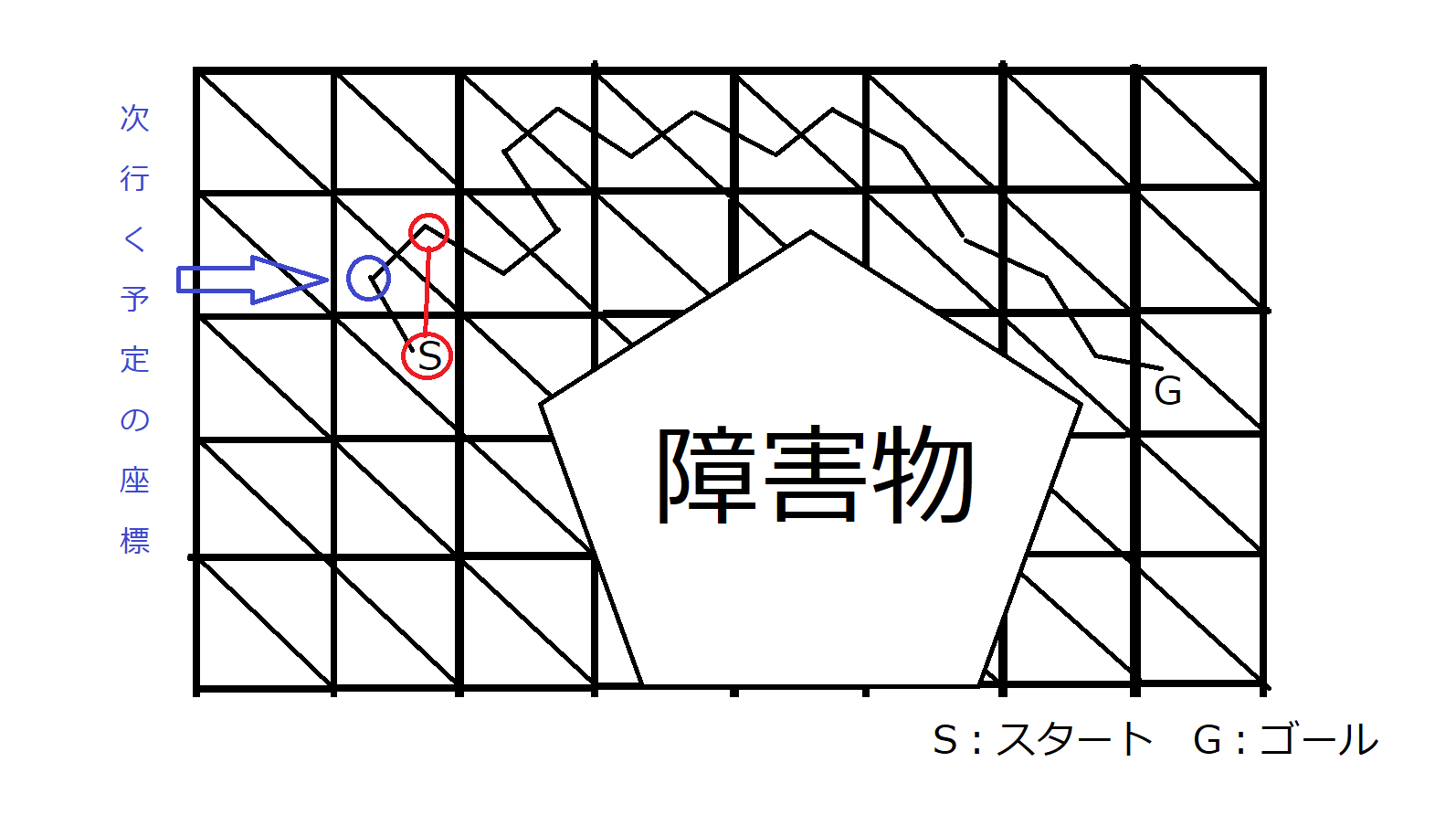
経路探査はA\*アルゴリズムを使用しています。ナビゲーションメッシュとA\*を利用して経路を作成すると、その経路はジグザグになってしまいます(図３参照)。

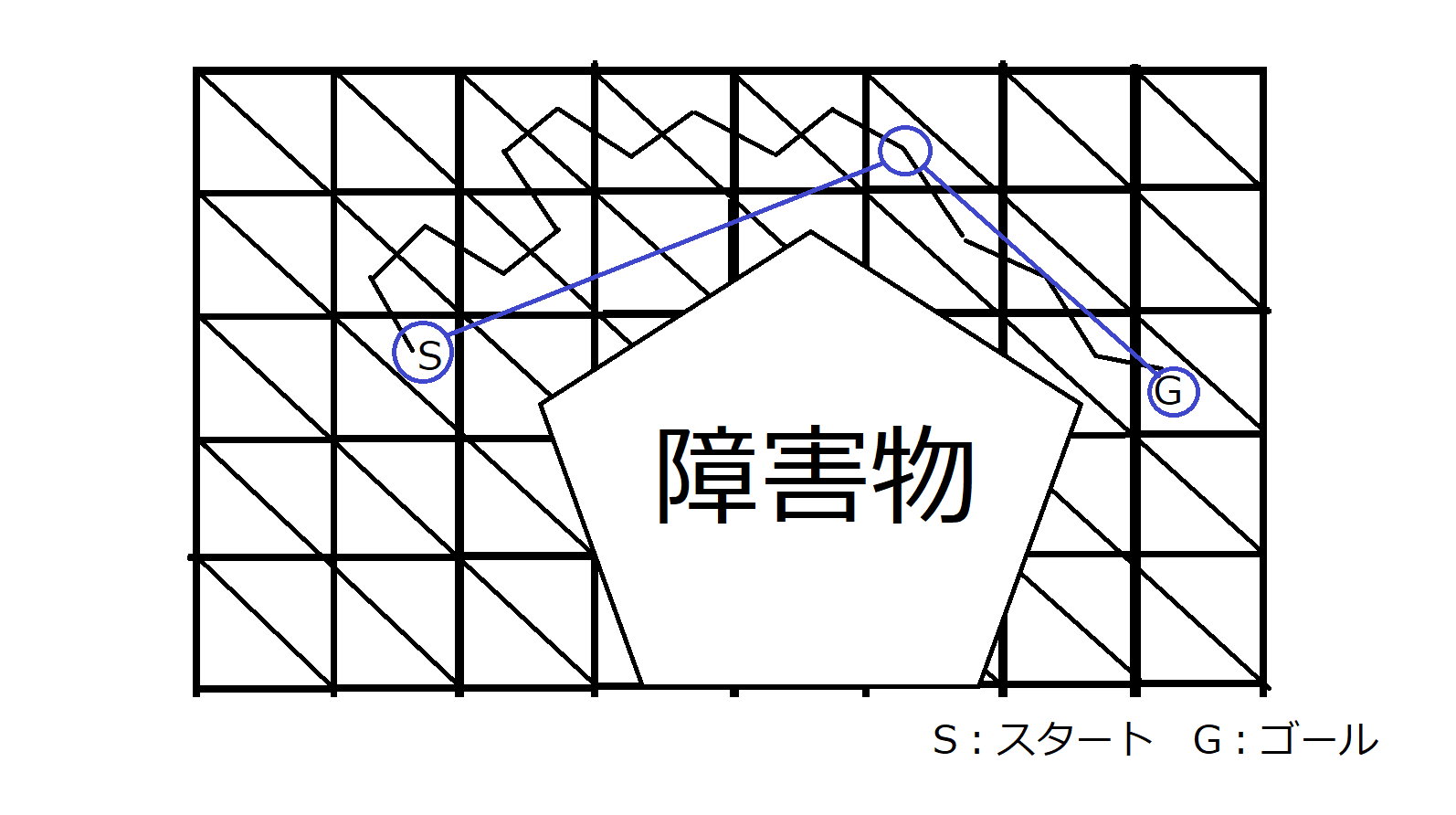
図３(経路のイメージ)

これを解消するために、経路のスムージングを行っています。

## **3.1 スムージングのアルゴリズム**

スムージングは今いる座標と次に進む予定の座標の次の座標の間(図４の赤い線分)で当たり判定をとって障害物がなければ次に進む予定の座標を進め、判定をとって当たるところまで繰り返し、当たったらループを抜け次行く予定の座標まで移動させたいものを進めることでスムージングをかけています。

図4

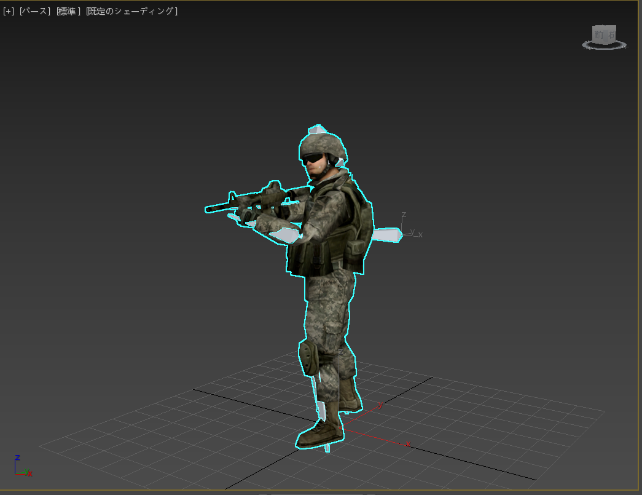
図５

この経路の場合だと最終的に通るルートは図５の青いラインになります。

# **４．アニメーションブレンディング**

学校内製の簡易エンジンにアニメーションを流す機能はもともとあったのですが、一つのアニメーションを流す機能しかなかったので、上半身と下半身で別のアニメーションを流せるように改造しました。

アニメーションを作るときにボーンに名前を付けられるので上半身のボーンには頭文字に「U」、下半身のボーンには「L」をつけることで、プログラムを書くときにボーンの名前を検索して、上半身だけ又は下半身だけにアニメーションを流せるようにしました。

これをすることで銃を構えた状態で下半身だけ歩きアニメーションだったり銃を撃ちながら歩いたり止まったりするときに違和感がなくなるようになりました。

上半身と下半身で別のアニメーションを適用することで構えながら歩くことができる。