

12. artisocレシピブック

A*探索アルゴリズムを使って、複数の目的地のうち最も近い場所へ移動しよう

本ドキュメントについてのご質問、『複雑系勉強会』のお問合せは、下記までご連絡ください。

(株)構造計画研究所
次世代事業開発部
MAS社会デザイン室 玉田

E-mail: tamada@kke.co.jp

A-star探索アルゴリズムを使って、 複数の目的地のうち最も近い場所へ移動しよう

- 複数の避難所の中から、最寄りの避難所へ避難したい。
- また、複数の出発地点を設定したい。

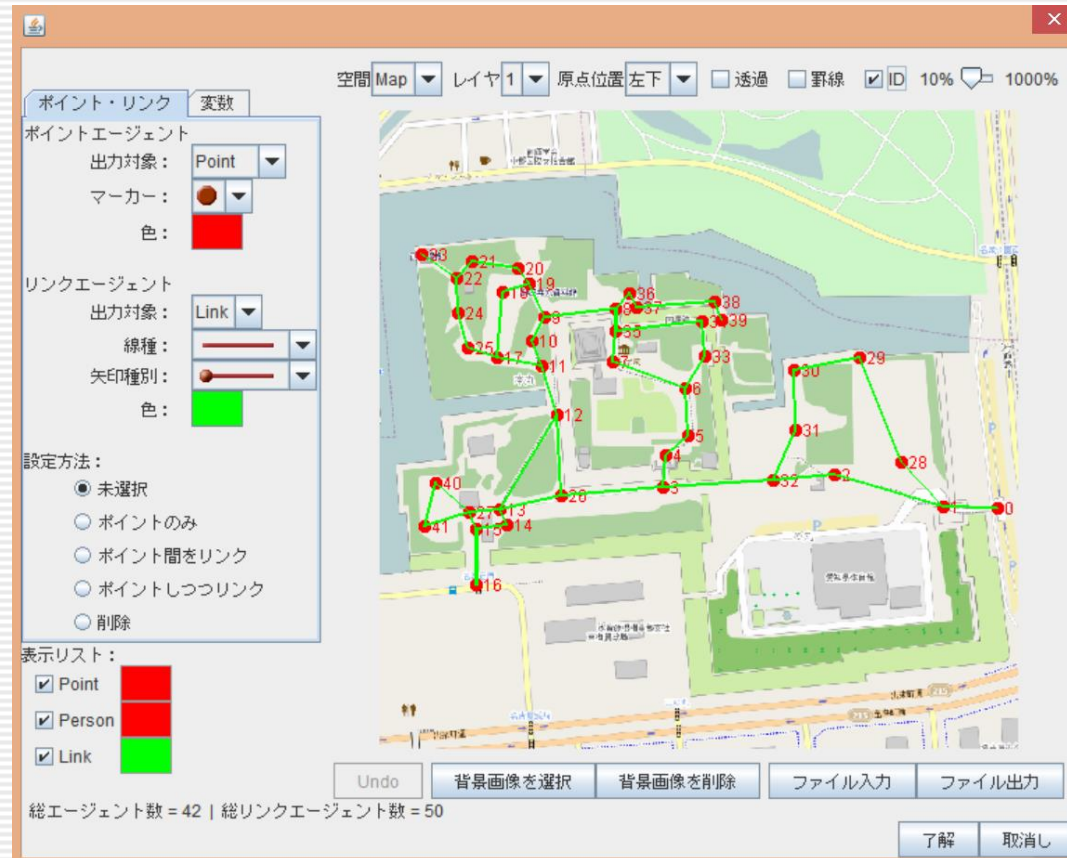
・歩行モデルの拡張

- ① 「11. artisocレシピブック」のおさらい
- ② 複数の出発地を設定し、目的地までの経路を設定
- ③ 歩行者の行動ルールを定義



①「11. artisocレシピブック」のおさらい

- A*探索アルゴリズムを使って、通れなくなった道路を迂回できるようになりました。
- これで避難経路が閉塞している場合、渋滞で迂回する必要がある場合をモデル化できるようになりました。



「11. artisocレシピブック」のモデルを元に、複数の出発地と目的地に対応したモデルに改造します。

②-1 複数の出発地を設定し、目的地までの経路を設定

■ 目的地を目指して、最短経路に沿って進みます。

● ツリーの「Point」を右クリックして、「ルールエディタ」を選択します。

Agt_Step{

Dim startPointIDArray **As** String, goalPointIDArray **As** String

Dim personAgt **As** Agt

Dim hitFlag **As** Boolean

Dim i **As** Integer

startPointIDArray = "0,12,26"

・・・出発地のPoint IDの配列を設定する

goalPointIDArray = "23,39,41"

・・・目的地のPoint IDの配列を設定する

hitFlag = False

For i=0 **To** CountToken(startPointIDArray) - 1

・・・出発地であるか判定する

If My.ID == CInt(GetToken(startPointIDArray, i)) **Then**

hitFlag = True

Break

End If

Next i

②-2 複数の出発地を設定し、目的地までの経路を設定

```
If (hitFlag == True And GetCountStep() Mod 10 == 1) And GetCountStep() < 50 Then  
    personAgt = CreateAgt(Universe.Map.Person)          ...出発地の場合はPersonを生成する  
    personAgt.X = My.X  
    personAgt.Y = My.Y  
    personAgt.RouteArray = @a_star_id_array(My.ID, goalPointIDArray)  
    personAgt.RouteCount = 1  
End If  
}
```

③-1 歩行者の行動ルールを定義

■ 経路に沿って進み、各Pointに到着すると、リンクの接続を確認します。

● ツリーの「Person」を右クリックして、「ルールエディタ」を選択します。

Agt_Step {

Dim startPointID **As** Integer, startPointAgt **As** Agt

Dim targetPointID **As** Integer, targetPointAgt **As** Agt

Dim distance **As** Double

If My.RouteCount < CountToken(My.RouteArray) **Then**

Do While(True)

targetPointID = CInt(GetToken(My.RouteArray, My.RouteCount))

targetPointAgt = Universe.Map.Point(targetPointID)

distance = Pursue(targetPointAgt, 1)

・・・経路に沿って進む

③-2 歩行者の行動ルールを定義

```
If distance > 0 Then
    My.RouteCount = My.RouteCount + 1
    If CountToken(My.RouteArray) > My.RouteCount Then
        startPointID = CInt(GetToken(My.RouteArray, My.RouteCount - 1))
        targetPointID = CInt(GetToken(My.RouteArray, My.RouteCount))
        If @check_link_id(startPointID, targetPointID) == False Then
            add_new_route()
        End If
    Else
        PrintLn("ゴールに到着！")
        Break
    End If
Else
    Break
End If
Loop
End If
}
```

...Pointに到着したとき

...リンクの接続を確認する
...リンクが接続していない場合は、新しい経路を探索する

③-3 歩行者の行動ルールを定義

```
Sub add_new_route()
```

```
{
```

```
    Dim startPointID As Integer, goalPointID As Integer
```

```
    Dim lastRoute As String, newRoute As String
```

```
    Dim i As Integer
```

```
    lastRoute = ""
```

```
    newRoute = ""
```

```
    For i=0 To My.RouteCount - 2
```

```
        lastRoute = lastRoute & GetToken(My.RouteArray, i) & ","
```

```
    Next i
```

```
    lastRoute = Left(lastRoute, Len(lastRoute) - 1)
```

・・・これまでにたどった経路を取得する

③-4 歩行者の行動ルールを定義

```
startPointID = CInt(GetToken(My.RouteArray, My.RouteCount - 1))  
goalPointID = CInt(GetToken(My.RouteArray, CountToken(My.RouteArray)-1))
```

```
If startPointID <> goalPointID Then
```

```
    newRoute = @a_star_id(startPointID, goalPointID)
```

```
    If CountToken(newRoute) == 0 Then
```

```
        PrintLn("目的地へ到達するルートがありません")
```

```
    End If
```

```
End If
```

```
If Len(newRoute) > 0 Then
```

```
    My.RouteArray = lastRoute & "," & newRoute
```

```
Else
```

```
    My.RouteArray = lastRoute
```

```
End If
```

```
}
```

・・・新しい経路を取得する

・・・どの経路をたどってもゴール
に到達できない場合はメッセー
ジ表示する

・・・これまで辿った経路と新し
い経路をを連結する