Infohazard.HyperNav

2.3.0

Generated by Doxygen 1.9.5

1 HyperNav Documentation **1**

1.1 Table of Contents . . . 1
1.2 Introduction . . . 2
1.3 Documentation and Support . . . 3
1.4 License . . . 3
1.5 Installation . . . 3
1.5.1 Prerequisites . . . 3
1.5.2 Asset Store . . . 3
1.5.3 Package Manager (Git URL or Submodule) . . . 3
1.6 Setup . . . 3
1.6.1 General Setup . . . 3
1.6.2 SRP Setup . . . 4
1.7 Demos . . . 4
1.7.1 HyperNavDemo (Volumetric Pathfinding) . . . 4
1.7.2 HyperNavAvoidanceDemo (Pathfinding Avoidance) . . . 4
1.7.3 HyperNavAvoidanceScaleDemo (Many-Agent Standalone Avoidance) . . . 4
1.7.4 HyperNavFloatingOriginDemo (Floating Origin System) . . . 4
1.7.5 HyperNavDynamicLinkBakingDemo (Runtime External Link Generation) . . . 4
1.7.6 HyperNavRuntimeBakingDemo (Runtime Data Generation) . . . 5
1.7.7 HyperNavManualLinkDemo (Manual Toggle-able Links) . . . 5
1.7.8 HyperNavSurfaceDemo (Omnidirectional Surface Pathfinding) . . . 5
1.7.9 HyperNavSurfaceAndVolumeDemo (Hybrid Walking/Flying Character) . . . 5
1.7.10 HyperNavLayersDemo (Navigation Layers) . . . 5
1.8 Features Guide . . . 5
1.8.1 Volume and Surface Setup . . . 5
1.8.2 External Links . . . 6
1.8.3 Pathfinder Setup . . . 6
1.8.4 Agent Setup . . . 6
1.8.5 Avoidance . . . 7
1.8.6 Navigation Layers . . . 8
1.8.7 Moving Volumes and Floating Origin . . . 8

2 Changelog **9**

2.1 [2.3.0] - 2026-8-24 . . . 9
2.1.1 Added . . . 9
2.2 [2.2.0] - 2025-8-14 . . . 9
2.2.1 Added . . . 9
2.2.2 Changed . . . 9
2.2.3 Fixed . . . 9
2.3 [2.1.0] - 2025-4-5 . . . 9
2.3.1 Added . . . 9
2.3.2 Changed . . . 10

2.3.3 Fixed . . . 10
2.4 [2.0.0] - 2025-2-11 . . . 10
2.4.1 Added . . . 10
2.4.2 Changed . . . 10
2.4.3 Fixed . . . 11
2.5 [1.1.6] - 2023-6-21 . . . 11
2.5.1 Fixed . . . 11
2.6 [1.1.5] - 2023-3-10 . . . 11
2.6.1 Fixed . . . 11
2.7 [1.1.4] - 2023-2-24 . . . 11
2.7.1 Added . . . 11
2.8 [1.1.3] - 2023-1-19 . . . 11
2.8.1 Added . . . 11
2.8.2 Fixed . . . 11
2.9 [1.1.2] - 2023-1-18 . . . 12
2.9.1 Fixed . . . 12
2.10 [1.1.1] - 2022-12-14 . . . 12
2.10.1 Fixed . . . 12
2.11 [1.1.0] - 2022-11-22 . . . 12
2.11.1 Added . . . 12
2.11.2 Changed . . . 12
2.11.3 Fixed . . . 12
2.12 [1.0.0] - 2022-11-08 . . . 13
2.12.1 Added . . . 13

3 Hierarchical Index **13**
3.1 Class Hierarchy . . . 13

4 Class Index **16**
4.1 Class List . . . 16

5 Namespace Documentation **21**
5.1 Infohazard Namespace Reference . . . 21
5.2 Infohazard.HyperNav Namespace Reference . . . 21
5.2.1 Enumeration Type Documentation . . . 23
5.2.2 Function Documentation . . . 24
5.3 Infohazard.HyperNav.Editor Namespace Reference . . . 24
5.4 Infohazard.HyperNav.Jobs Namespace Reference . . . 25
5.5 Infohazard.HyperNav.Jobs.Baking Namespace Reference . . . 25
5.6 Infohazard.HyperNav.Jobs.Baking.Surface Namespace Reference . . . 25
5.7 Infohazard.HyperNav.Jobs.Baking.Volume Namespace Reference . . . 26
5.8 Infohazard.HyperNav.Jobs.Utility Namespace Reference . . . 26
5.8.1 Enumeration Type Documentation . . . 27

5.9 Infohazard.HyperNav.Settings Namespace Reference . . . 27
5.10 Infohazard.HyperNav.Tests Namespace Reference . . . 27
5.11 Infohazard.HyperNav.Tests.UnitTests Namespace Reference . . . 27

6 Class Documentation **27**

6.1 Infohazard.HyperNav.Avoidance Class Reference . . . 27
6.1.1 Detailed Description . . . 28
6.1.2 Property Documentation . . . 28
6.2 Infohazard.HyperNav.AvoidanceAgent Class Reference . . . 28
6.2.1 Detailed Description . . . 29
6.2.2 Member Function Documentation . . . 29
6.2.3 Property Documentation . . . 30
6.2.4 Event Documentation . . . 31
6.3 Infohazard.HyperNav.Jobs.AvoidanceJob Struct Reference . . . 31
6.3.1 Detailed Description . . . 32
6.3.2 Member Function Documentation . . . 32
6.3.3 Member Data Documentation . . . 33
6.4 Infohazard.HyperNav.AvoidanceManager Class Reference . . . 34
6.4.1 Detailed Description . . . 35
6.4.2 Member Function Documentation . . . 35
6.4.3 Property Documentation . . . 35
6.5 Infohazard.HyperNav.AvoidanceObstacleBase Class Reference . . . 36
6.5.1 Detailed Description . . . 37
6.5.2 Member Function Documentation . . . 37
6.5.3 Property Documentation . . . 37
6.6 Infohazard.HyperNav.Jobs.Baking.Surface.CalculateVertexAnglesJob Struct Reference . . . 38
6.6.1 Detailed Description . . . 38
6.7 Infohazard.HyperNav.Edge Struct Reference . . . 38
6.7.1 Detailed Description . . . 39
6.7.2 Constructor & Destructor Documentation . . . 39
6.7.3 Member Function Documentation . . . 40
6.7.4 Member Data Documentation . . . 42
6.7.5 Property Documentation . . . 42
6.8 Infohazard.HyperNav.Editor.EditorNavAreaBakingUtility Class Reference . . . 42
6.8.1 Detailed Description . . . 43
6.8.2 Member Function Documentation . . . 43
6.8.3 Member Data Documentation . . . 45
6.8.4 Event Documentation . . . 45
6.9 Infohazard.HyperNav.Editor.ExternalLinkEditorUtility Class Reference . . . 46
6.9.1 Detailed Description . . . 46
6.9.2 Member Function Documentation . . . 46
6.10 Infohazard.HyperNav.Jobs.Utility.Fast3DArray Struct Reference . . . 47

6.10.1 Detailed Description . . . 48
6.10.2 Constructor & Destructor Documentation . . . 48
6.10.3 Member Function Documentation . . . 48
6.10.4 Member Data Documentation . . . 49
6.10.5 Property Documentation . . . 50
6.11 Infohazard.HyperNav.IAvoidanceAgent Interface Reference . . . 50
6.11.1 Detailed Description . . . 51
6.11.2 Member Function Documentation . . . 51
6.11.3 Property Documentation . . . 51
6.12 Infohazard.HyperNav.IAvoidanceObstacle Interface Reference . . . 52
6.12.1 Detailed Description . . . 52
6.12.2 Property Documentation . . . 53
6.13 Infohazard.HyperNav.Jobs.Utility.IntList2 Struct Reference . . . 53
6.13.1 Detailed Description . . . 54
6.13.2 Member Function Documentation . . . 54
6.13.3 Member Data Documentation . . . 55
6.13.4 Property Documentation . . . 55
6.14 Infohazard.HyperNav.Jobs.Baking.InvertIndexArrayJob Struct Reference . . . 56
6.14.1 Detailed Description . . . 56
6.15 Infohazard.HyperNav.ISurfaceUprightDirectionHandler Interface Reference . . . 56
6.15.1 Detailed Description . . . 56
6.15.2 Member Function Documentation . . . 56
6.16 Infohazard.HyperNav.Jobs.Utility.MarchingCubesCavityTables Class Reference . . . 57
6.16.1 Detailed Description . . . 57
6.16.2 Member Data Documentation . . . 57
6.17 Infohazard.HyperNav.Jobs.Utility.MarchingCubesTables Class Reference . . . 58
6.17.1 Detailed Description . . . 58
6.17.2 Member Function Documentation . . . 59
6.17.3 Member Data Documentation . . . 59
6.18 Infohazard.HyperNav.Jobs.NativeAvoidanceObstacleData Struct Reference . . . 61
6.18.1 Detailed Description . . . 61
6.18.2 Member Data Documentation . . . 61
6.19 Infohazard.HyperNav.Jobs.Utility.NativeBounds Struct Reference . . . 62
6.19.1 Detailed Description . . . 63
6.19.2 Constructor & Destructor Documentation . . . 63
6.19.3 Member Data Documentation . . . 63
6.20 Infohazard.HyperNav.Jobs.Utility.NativeMathUtility Class Reference . . . 64
6.20.1 Detailed Description . . . 64
6.20.2 Member Function Documentation . . . 64
6.21 Infohazard.HyperNav.Jobs.Utility.NativeNavExternalLinkData Struct Reference . . . 69
6.21.1 Detailed Description . . . 70
6.21.2 Constructor & Destructor Documentation . . . 70

6.21.3 Member Data Documentation . . . 70
6.22 Infohazard.HyperNav.NativeNavSurfaceDataPointers Struct Reference . . . 72
6.22.1 Detailed Description . . . 72
6.22.2 Member Function Documentation . . . 72
6.23 Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData Struct Reference . . . 72
6.23.1 Detailed Description . . . 73
6.23.2 Constructor & Destructor Documentation . . . 73
6.23.3 Member Data Documentation . . . 74
6.24 Infohazard.HyperNav.NativeNavVolumeDataPointers Struct Reference . . . 77
6.24.1 Detailed Description . . . 77
6.24.2 Member Function Documentation . . . 77
6.25 Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeInternalLinkData Struct Reference . . . 77
6.25.1 Detailed Description . . . 78
6.25.2 Constructor & Destructor Documentation . . . 78
6.25.3 Member Data Documentation . . . 78
6.26 Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeRegionData Struct Reference . . . 79
6.26.1 Detailed Description . . . 79
6.26.2 Constructor & Destructor Documentation . . . 79
6.26.3 Member Data Documentation . . . 80
6.27 Infohazard.HyperNav.Jobs.Utility.NativeNavWaypoint Struct Reference . . . 81
6.27.1 Detailed Description . . . 81
6.27.2 Constructor & Destructor Documentation . . . 81
6.27.3 Member Data Documentation . . . 82
6.28 Infohazard.HyperNav.Jobs.NativePlane Struct Reference . . . 82
6.28.1 Detailed Description . . . 83
6.28.2 Constructor & Destructor Documentation . . . 83
6.28.3 Member Data Documentation . . . 84
6.28.4 Property Documentation . . . 84
6.29 Infohazard.HyperNav.Jobs.NativeRay Struct Reference . . . 84
6.29.1 Detailed Description . . . 84
6.29.2 Member Data Documentation . . . 84
6.30 Infohazard.HyperNav.Jobs.NativeRaycastElement Struct Reference . . . 85
6.30.1 Detailed Description . . . 85
6.30.2 Member Data Documentation . . . 85
6.31 Infohazard.HyperNav.NavAgent Class Reference . . . 86
6.31.1 Detailed Description . . . 89
6.31.2 Member Function Documentation . . . 89
6.31.3 Property Documentation . . . 92
6.31.4 Event Documentation . . . 98
6.32 Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings > Class Template Reference . . . 98
6.32.1 Detailed Description . . . 100

6.32.2 Member Function Documentation . . . 100
6.32.3 Member Data Documentation . . . 102
6.32.4 Property Documentation . . . 102
6.33 Infohazard.HyperNav.NavAreaBakeProgress Struct Reference . . . 103
6.33.1 Detailed Description . . . 103
6.34 Infohazard.HyperNav.NavAreaBakingUtility Class Reference . . . 103
6.34.1 Detailed Description . . . 104
6.34.2 Member Function Documentation . . . 104
6.35 Infohazard.HyperNav.NavAreaBase Class Reference . . . 104
6.35.1 Detailed Description . . . 106
6.35.2 Member Function Documentation . . . 106
6.35.3 Property Documentation . . . 107
6.36 Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf > Class Template Reference . . . 110
6.36.1 Detailed Description . . . 111
6.36.2 Member Function Documentation . . . 111
6.36.3 Property Documentation . . . 112
6.37 Infohazard.HyperNav.Settings.NavAreaBaseSettingsAsset< TData > Class Template Reference . . . 113
6.37.1 Detailed Description . . . 113
6.37.2 Property Documentation . . . 114
6.38 Infohazard.HyperNav.NavAreaExternalLinkUpdate Class Reference . . . 114
6.38.1 Detailed Description . . . 114
6.38.2 Member Function Documentation . . . 114
6.39 Infohazard.HyperNav.Editor.NavAreaMigrator Class Reference . . . 115
6.39.1 Detailed Description . . . 115
6.40 Infohazard.HyperNav.Editor.NavEditorUtility Class Reference . . . 115
6.40.1 Detailed Description . . . 115
6.40.2 Member Function Documentation . . . 116
6.41 Infohazard.HyperNav.NavExternalLinkData Class Reference . . . 117
6.41.1 Detailed Description . . . 117
6.41.2 Constructor & Destructor Documentation . . . 117
6.41.3 Member Function Documentation . . . 118
6.41.4 Property Documentation . . . 118
6.42 Infohazard.HyperNav.NavInternalLinkData Class Reference . . . 119
6.42.1 Detailed Description . . . 120
6.42.2 Constructor & Destructor Documentation . . . 120
6.42.3 Member Function Documentation . . . 122
6.42.4 Property Documentation . . . 122
6.43 Infohazard.HyperNav.NavLayer Struct Reference . . . 123
6.43.1 Detailed Description . . . 123
6.43.2 Member Function Documentation . . . 123
6.44 Infohazard.HyperNav.NavLayerMask Struct Reference . . . 126
6.44.1 Detailed Description . . . 127

6.44.2 Member Function Documentation . . . 127
6.44.3 Member Data Documentation . . . 131
6.44.4 Property Documentation . . . 132
6.45 Infohazard.HyperNav.Jobs.NavMultiRaycastJob Struct Reference . . . 132
6.45.1 Detailed Description . . . 132
6.45.2 Member Function Documentation . . . 132
6.45.3 Member Data Documentation . . . 133
6.46 Infohazard.HyperNav.NavPath Class Reference . . . 133
6.46.1 Detailed Description . . . 134
6.46.2 Member Function Documentation . . . 134
6.46.3 Property Documentation . . . 134
6.47 Infohazard.HyperNav.NavPathfinder Class Reference . . . 135
6.47.1 Detailed Description . . . 136
6.47.2 Member Function Documentation . . . 136
6.47.3 Property Documentation . . . 138
6.48 Infohazard.HyperNav.Jobs.NavPathJob Struct Reference . . . 139
6.48.1 Detailed Description . . . 140
6.48.2 Member Function Documentation . . . 140
6.48.3 Member Data Documentation . . . 140
6.49 Infohazard.HyperNav.Jobs.NavRaycastJob Struct Reference . . . 142
6.49.1 Detailed Description . . . 142
6.49.2 Member Function Documentation . . . 142
6.49.3 Member Data Documentation . . . 142
6.50 Infohazard.HyperNav.NavRegionData Class Reference . . . 143
6.50.1 Detailed Description . . . 144
6.50.2 Constructor & Destructor Documentation . . . 144
6.50.3 Member Function Documentation . . . 145
6.50.4 Property Documentation . . . 145
6.51 Infohazard.HyperNav.Jobs.Utility.NavSampleResult Struct Reference . . . 146
6.51.1 Detailed Description . . . 147
6.51.2 Constructor & Destructor Documentation . . . 147
6.51.3 Member Data Documentation . . . 147
6.51.4 Property Documentation . . . 148
6.52 Infohazard.HyperNav.NavSurfaceBakeHandler Class Reference . . . 149
6.52.1 Detailed Description . . . 149
6.52.2 Constructor & Destructor Documentation . . . 149
6.53 Infohazard.HyperNav.NavSurfaceInternalLinkData Class Reference . . . 150
6.53.1 Detailed Description . . . 150
6.53.2 Constructor & Destructor Documentation . . . 150
6.53.3 Member Function Documentation . . . 151
6.53.4 Property Documentation . . . 151
6.54 Infohazard.HyperNav.NavSurfaceRegionData Class Reference . . . 152

6.54.1 Detailed Description . . . 153
6.54.2 Constructor & Destructor Documentation . . . 153
6.54.3 Member Function Documentation . . . 154
6.54.4 Property Documentation . . . 154
6.55 Infohazard.HyperNav.Settings.NavSurfaceSettings Class Reference . . . 155
6.55.1 Detailed Description . . . 156
6.55.2 Member Function Documentation . . . 156
6.55.3 Property Documentation . . . 157
6.56 Infohazard.HyperNav.Settings.NavSurfaceSettingsAsset Class Reference . . . 159
6.56.1 Detailed Description . . . 159
6.57 Infohazard.HyperNav.NavSurfaceUpdate Class Reference . . . 159
6.57.1 Detailed Description . . . 160
6.57.2 Member Function Documentation . . . 160
6.58 Infohazard.HyperNav.NavVolume Class Reference . . . 160
6.58.1 Detailed Description . . . 161
6.58.2 Member Function Documentation . . . 161
6.58.3 Property Documentation . . . 162
6.59 Infohazard.HyperNav.NavVolumeBakeHandler Class Reference . . . 162
6.59.1 Detailed Description . . . 162
6.59.2 Constructor & Destructor Documentation . . . 162
6.60 Infohazard.HyperNav.NavVolumeData Class Reference . . . 163
6.60.1 Detailed Description . . . 163
6.60.2 Member Function Documentation . . . 164
6.60.3 Property Documentation . . . 165
6.61 Infohazard.HyperNav.Editor.NavVolumeEditor Class Reference . . . 166
6.61.1 Detailed Description . . . 166
6.62 Infohazard.HyperNav.Settings.NavVolumeSettings Class Reference . . . 166
6.62.1 Detailed Description . . . 167
6.62.2 Member Function Documentation . . . 167
6.62.3 Property Documentation . . . 167
6.63 Infohazard.HyperNav.Settings.NavVolumeSettingsAsset Class Reference . . . 168
6.63.1 Detailed Description . . . 168
6.64 Infohazard.HyperNav.NavVolumeUpdate Class Reference . . . 168
6.64.1 Detailed Description . . . 168
6.64.2 Member Function Documentation . . . 168
6.65 Infohazard.HyperNav.NavWaypoint Struct Reference . . . 169
6.65.1 Detailed Description . . . 169
6.65.2 Property Documentation . . . 169
6.66 Infohazard.HyperNav.NavAgent.PropNames Class Reference . . . 170
6.66.1 Detailed Description . . . 170
6.67 Infohazard.HyperNav.NavAreaBase.PropNames Class Reference . . . 170
6.67.1 Detailed Description . . . 170

6.68 Infohazard.HyperNav.NavPathfinder.PropNames Class Reference . . . 171
6.68.1 Detailed Description . . . 171
6.69 Infohazard.HyperNav.NavSurface.PropNames Class Reference . . . 171
6.69.1 Detailed Description . . . 171
6.70 Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf >.PropNames Class Reference . . . 171
6.70.1 Detailed Description . . . 171
6.71 Infohazard.HyperNav.Settings.NavAreaBaseSettingsAsset< TData >.PropNames Class Reference 171
6.71.1 Detailed Description . . . 172
6.72 Infohazard.HyperNav.Settings.NavSurfaceSettings.PropNames Class Reference . . . 172
6.72.1 Detailed Description . . . 172
6.73 Infohazard.HyperNav.Settings.NavVolumeSettings.PropNames Class Reference . . . 172
6.73.1 Detailed Description . . . 172
6.74 Infohazard.HyperNav.SimpleNavAgentMover.PropNames Class Reference . . . 172
6.74.1 Detailed Description . . . 172
6.75 Infohazard.HyperNav.Jobs.Utility.ReadOnlyArrayPointer< T > Struct Template Reference . . . 172
6.75.1 Detailed Description . . . 172
6.76 Infohazard.HyperNav.Jobs.Baking.Surface.RemoveJaggedTrianglesJob Struct Reference . . . 173
6.76.1 Detailed Description . . . 173
6.77 Infohazard.HyperNav.RigidbodyAvoidanceObstacle Class Reference . . . 173
6.77.1 Detailed Description . . . 174
6.77.2 Member Function Documentation . . . 174
6.77.3 Property Documentation . . . 174
6.78 Infohazard.HyperNav.SimpleAvoidanceObstacle Class Reference . . . 174
6.78.1 Detailed Description . . . 175
6.78.2 Member Function Documentation . . . 175
6.78.3 Property Documentation . . . 175
6.79 Infohazard.HyperNav.SimpleNavAgentMover Class Reference . . . 175
6.79.1 Detailed Description . . . 178
6.79.2 Member Enumeration Documentation . . . 178
6.79.3 Member Function Documentation . . . 178
6.79.4 Member Data Documentation . . . 180
6.79.5 Property Documentation . . . 181
6.80 Infohazard.HyperNav.SplineNavAgent Class Reference . . . 184
6.80.1 Detailed Description . . . 186
6.80.2 Member Function Documentation . . . 186
6.80.3 Property Documentation . . . 187
6.81 Infohazard.HyperNav.SplinePath Struct Reference . . . 190
6.81.1 Detailed Description . . . 191
6.81.2 Constructor & Destructor Documentation . . . 191
6.81.3 Member Function Documentation . . . 191
6.81.4 Property Documentation . . . 195
6.82 Infohazard.HyperNav.SplinePoint Struct Reference . . . 196

6.82.1 Detailed Description . . . 196
6.82.2 Member Data Documentation . . . 196
6.83 Infohazard.HyperNav.Jobs.SplineProjectJob Struct Reference . . . 197
6.83.1 Detailed Description . . . 197
6.83.2 Member Function Documentation . . . 198
6.83.3 Member Data Documentation . . . 198
6.84 Infohazard.HyperNav.SurfaceUprightDirectionHandlerWithOverrides Class Reference . . . 199
6.84.1 Detailed Description . . . 199
6.84.2 Member Function Documentation . . . 199
6.85 Infohazard.HyperNav.Triangle Struct Reference . . . 200
6.85.1 Detailed Description . . . 201
6.85.2 Constructor & Destructor Documentation . . . 201
6.85.3 Member Function Documentation . . . 201
6.85.4 Member Data Documentation . . . 203
6.85.5 Property Documentation . . . 203
6.86 Infohazard.HyperNav.Jobs.Utility.TriangleRegionIndices Struct Reference . . . 204
6.86.1 Detailed Description . . . 204
6.87 Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T > Struct Template Reference . . . 204
6.87.1 Detailed Description . . . 205
6.87.2 Constructor & Destructor Documentation . . . 205
6.87.3 Member Function Documentation . . . 206
6.87.4 Member Data Documentation . . . 207
6.87.5 Property Documentation . . . 207
6.88 Infohazard.HyperNav.Jobs.Utility.UnsafeConcurrentQueue< T > Struct Template Reference . . . 208
6.88.1 Detailed Description . . . 208
6.89 Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T > Struct Template Reference . . . 208
6.89.1 Detailed Description . . . 209
6.89.2 Constructor & Destructor Documentation . . . 209
6.89.3 Member Function Documentation . . . 210
6.89.4 Property Documentation . . . 213

1 HyperNav Documentation

1.1 Table of Contents

- HyperNav Documentation
  - Table of Contents
  - Introduction
  - Documentation and Support
  - License
  - Installation

    - Prerequisites
    - Asset Store
    - Package Manager (Git URL or Submodule)
  - Setup
    - General Setup
    - SRP Setup
  - Demos
    - HyperNavDemo (Volumetric Pathfinding)
    - HyperNavAvoidanceDemo (Pathfinding Avoidance)
    - HyperNavAvoidanceScaleDemo (Many-Agent Standalone Avoidance)
    - HyperNavFloatingOriginDemo (Floating Origin System)
    - HyperNavDynamicLinkBakingDemo (Runtime External Link Generation)
    - HyperNavRuntimeBakingDemo (Runtime Data Generation)
    - HyperNavManualLinkDemo (Manual Toggle-able Links)
    - HyperNavSurfaceDemo (Omnidirectional Surface Pathfinding)
    - HyperNavSurfaceAndVolumeDemo (Hybrid Walking/Flying Character)
    - HyperNavLayersDemo (Navigation Layers)
  - Features Guide
    - Volume and Surface Setup
      - Baking
      - Runtime Baking
    - External Links
      - Shared Settings
      - Manual Links
      - Runtime Link Generation
    - Pathfinder Setup
    - Agent Setup
      - Agent-Driven Movement
      - Spline Agent Setup
    - Avoidance
      - AvoidanceManager Setup
      - AvoidanceAgent Setup
      - AvoidanceAgent With NavAgent
      - AvoidanceAgent Usage
      - AvoidanceObstacle Setup
    - Navigation Layers
      - Layer Setup
      - Layer Usage
      - Layer Scripting
    - Moving Volumes and Floating Origin
      - Moving Volume Limitations

1.2 Introduction

HyperNav is a navmesh-like system for implementing volumetric and omnidirectional navigation and 3D obstacle avoidance. Unlike a navmesh, which is bound to walkable surfaces, HyperNav creates volumes in which characters can navigate on all three axes, and surfaces on which characters can navigate with any upwards direction. I developed it for Astral Horizon, an FPS with 6-DOF gameplay.

1.3 Documentation and Support

API Docs

Tutorial Playlist

Discord

1.4 License

HyperNav uses the standard Unity Asset Store per-seat license for tools.

1.5 Installation

1.5.1 Prerequisites

HyperNav depends on the Infohazard.Core library, which you can get from the Asset Store or as a Package Manager package from Github. Regardless of how you install it, you should do so before importing HyperNav. You can see more information about the core library here.

HyperNav also requires UniTask, which can be installed as a package manager git package or a .unitypackage file.

Finally HyperNav has two additional package manager dependencies: burst and collections. However, these will automatically be installed when you import it, so no need to do anything here.

1.5.2 Asset Store

The main way to install HyperNav is through the asset store. Just install it as you would any other asset, and make sure you allow Unity to update package manager dependencies.

1.5.3 Package Manager (Git URL or Submodule)

Because HyperNav is a paid asset, the Github repository is not open source. However, if you wish to contribute to HyperNav and have purchased a seat, please feel free to email me and I can look into getting you read access to the repository.

1.6 Setup

1.6.1 General Setup

The only setup required beyond installation is to add references to the Infohazard.HyperNav assembly if you are using an assembly definition. If you are using the default assemblies (such as Assembly-CSharp), nothing is needed here.

1.6.2 SRP Setup

If you are using a scriptable render pipeline (URP, HDRP, etc) and wish to run the demos, you will need to upgrade the materials using your render pipeline's material upgrade system. The materials you'll need to upgrade are in:

- `Assets/Plugins/Infohazard/Demos/Infohazard.HyperNav/Materials`
- `Assets/Plugins/Infohazard/Demos/Shared Demo Assets/Materials`
- `Assets/StarterAssets/ThirdPersonController/Character/Materials`

1.7 Demos

The following demo scenes are provided to cover the various features of HyperNav. They are found in `Assets/↩ Plugins/Infohazard/Demos/Infohazard.HyperNav/Scenes`.

1.7.1 HyperNavDemo (Volumetric Pathfinding)

This scene demonstrates how to set up NavVolumes and a NavPathfinder, and use NavAgent and SplineNavAgent to find paths. Run the demo and click anywhere on the right side of the screen to set a destination, and use the WASD keys to rotate the camera on the left side of the screen.

1.7.2 HyperNavAvoidanceDemo (Pathfinding Avoidance)

This scene demonstrates how you can use avoidance along with NavAgents to avoid agents getting stuck in a narrow maze. Try turning off avoidance by setting the agents' avoidance weights to zero to see the difference it makes.

1.7.3 HyperNavAvoidanceScaleDemo (Many-Agent Standalone Avoidance)

This scene deomonstrates how you can use avoidance without a NavAgent. It also shows that the avoidance system performs well even with a huge number of agents all avoiding one another.

1.7.4 HyperNavFloatingOriginDemo (Floating Origin System)

This scene deomonstrates how you can use HyperNav with a floating origin system. This is useful in very large scenes where you need to shift all objects such that the player remains close to the origin.

1.7.5 HyperNavDynamicLinkBakingDemo (Runtime External Link Generation)

This scene demonstrates how you can generate volume external links at runtime in order to support a runtime modular building system. This requires having one volume per module, but is preferable to baking the entire volume data at runtime (it is much, much faster).

1.7.6 HyperNavRuntimeBakingDemo (Runtime Data Generation)

This scene demonsstrates how you can generate the entire volume data at runtime in order to support fully procedurally generated maps. Note that while runtime generation is supported, it is still quite performance intensive. I do not recommend generating data during gameplay; rather you should try to do it during a loading screen if possible (or better yet, use modular volumes and simply generate the links at runtime).

1.7.7 HyperNavManualLinkDemo (Manual Toggle-able Links)

This scene shows how you can create manual external links that can be turned on or off. This is useful, for example, if you have doors that can be locked and should only be considered navigable when they are unlocked. Manual external links can also connect different regions in the same volume, rather than only separate volumes.

1.7.8 HyperNavSurfaceDemo (Omnidirectional Surface Pathfinding)

This scene shows how you can use the NavSurface component to enable pathfinding on complex omnidirectional surfaces. The agent behaves as a normal walking character, but it can walk on surfaces pointing in any direction. Instead of being constrained to a single upright direction as in a traditional NavMesh, a NavSurface allows arbitrary upright directions anywhere on the surface.

1.7.9 HyperNavSurfaceAndVolumeDemo (Hybrid Walking/Flying Character)

Inspired by Astral Horizon, this scene shows how you can have a character able to both walk on surfaces and fly in volumes, and transition dynamically between them. The volume and surface take up the same area and are connected by external links.

1.7.10 HyperNavLayersDemo (Navigation Layers)

This scene shows how to use NavLayers to create separate nagivation bakes for agents of different sizes. The scene features a narrow gap that a smaller character can navigate through but a larger character cannot.

1.8 Features Guide

1.8.1 Volume and Surface Setup

The first step to setup navigation in your scene is to create a navigable area (NavSurface, NavVolume, or both), using one of the menu items:

- `Tools > Infohazard > Create > Nav Volume` - create NavVolume for volumetric navigation.
- `Tools > Infohazard > Create > Nav Surface` - create NavSurface for omnidirectional surface-based navigation.
- `Tools > Infohazard > Create > Nav Volume + Surface` - create a single area for both types of navigation.

1.8.1.1 Baking

Once your parameters are configured, you can bake the nav area! Just hit the "Bake" button and wait for it to finish (it should be fairly quick). If the baking is taking a long time, consider breaking your area up into multiple smaller areas. By default, when baking is done, you should see the baked results rendered as a blue mesh in the scene view when the area is selected.

1.8.1.2 Runtime Baking

To regernate area data at runtime, use `NavVolumeUpdate.GenerateVolumeData` or `NavSurfaceUpdate.GenerateSurfaceData` with the area you wish to update. This operation is quite performance intensive, so it is not recommended to do it during gameplay. Instead, try to do it during a loading screen if possible. Additionally, you will probably only want to generate one area at a time, as generating multiple areas will cause thread starvation and tank the framerate.

1.8.2 External Links

If you are using multiple areas (including using a volume + surface), you most likely will want agents to be able to find paths between them. This is accomplished using external links. Simply use the "Generate External Links" button after all areas are baked, or, for convenience, "Generate All External Links" to generate links on all loaded areas. Once the external links are generated, you can visualize them in each area's visualization settings.

Note that if you re-bake an area, you will need to regenerate not only its external links but the external links of all of its neighbors. If you don't do this, you will get links leading to the wrong region, and potentially paths that cross through impassible areas. The "Generate All External Links" button comes in handy here.

1.8.2.1 Shared Settings

By default, the settings for each NavArea are specific to each instance. However, by creating a settings asset, you can share settings between multiple areas. To do so, go into the inspector for an area, and click Create Shared Settings. This will copy the current settings on the area into a new settings asset which can then be assigned to other areas of the same type. You can also clone the asset to make variants. Currently, overriding individual settings is not supported, so if you need to change a setting for a single area, you will need to create a new shared settings asset.

1.8.2.2 Manual Links

In addition to the auto-generated links, you can create manual external links. These links can be toggled on or off at runtime, and can connect different regions in the same volume, rather than only separate volumes. To create a manual link, use the menu item `Tools > Infohazard > Create > Manual Nav Link`, and use the gizmos to position the endpoints of the link in the desired areas. The next time you generate the volumes' external links, the manual links will be included (note that generating the links is required for the manual links to be included in the pathfinding data).

1.8.2.3 Runtime Link Generation

To regenerate external links at runtime, use `NavAreaExternalLinkUpdate.GenerateExternalLinks` with the areas you wish to update.

1.8.3 Pathfinder Setup

Before you can start pathfinding, you also need to create a NavPathfinder, using the menu item Tools > Infohazard > Create > Nav Pathfinder. You can create a NavPathfinder in each scene that needs one, or you can create one that persists through level loading.

1.8.4 Agent Setup

The final step to start pathfinding is to set up agents. A NavAgent is the easiest way to start finding paths from code. Simply add the NavAgent to your GameObject, set its Destination, and read its DesiredVelocity.

1.8.4.1 Agent-Driven Movement A NavAgent simply provides a DesiredVelocity, so you'll need another script to actually move the object. You can either use your own movement code that reads DesiredVelocity, or use the provided SimpleNavAgentMover script to handle some common situations.

1.8.4.2 Spline Agent Setup If you wish to use the spline path system to achieve smoother navigation paths, simply replace the NavAgent with a SplineNavAgent. SplineNavAgent provides the same properties as NavAgent (as it is a subclass) with some additional properties. SplineNavAgent works on both volumes and surfaces, but the actual splines will only be used for volume-based navigation; on surfaces it will revert to the NavAgent behavior.

1.8.5 Avoidance

Whereas NavAgent and SplineNavAgent are used to plan paths between two points, avoidance is used to steer clear of obstacles that might obstruct that path. These obstacles might be moving objects or even other navigating agents, so they can't easily be baked into the area data. Instead, you should use the obstacle avoidance system.

1.8.5.1 AvoidanceManager Setup In order to use avoidance, there must be an AvoidanceManager present in your scene. To create an AvoidanceManager, use the menu item Tools > Infohazard > Create > Avoidance Manager.

1.8.5.2 AvoidanceAgent Setup If you want an object to avoid obstacles, you should add an AvoidanceAgent script. This script needs to know the maximum speed the agent can travel at, the agent's size, and several other properties.

1.8.5.3 AvoidanceAgent With NavAgent When you want to use avoidance with a NavAgent or SplineNav↩ Agent, the usage is simple. The NavAgent should have a button to create an AvoidanceAgent on the object and instantly connect it to the NavAgent. The NavAgent will handle the input and output velocity for the Avoidance↩ Agent, so you can simply read the NavAgent's DesiredVelocity which now includes avoidance. You will still need to configure the max speed on the AvoidanceAgent for it to work properly.

1.8.5.4 AvoidanceAgent Usage You can use an AvoidanceAgent without a NavAgent. For this use case, you must supply an input velocity (the velocity the agent wants to move in) and read its AvoidanceVelocity (the velocity it should move in to avoid obstacles). The input velocity is provided via a delegate, allowing you to connected it to whatever source you want.

1.8.5.5 AvoidanceObstacle Setup While AvoidanceAgents will automatically avoid each other, you may wish to add additional obstacles that are not agents. This can be done using AvoidanceObstacleBase or any of its subclasses (which determine the current velocity using different methods). Just add one of these scripts to an object and the obstacle is good to go:

- AvoidanceObstacleBase: Does not calculate a velocity, so it is assumed to be stationary. You can inherit from it to add your own velocity calculations.
- SimpleAvoidanceObstacle: Calculates its velocity by measuring delta position / delta time every frame.
- RigidbodyAvoidanceObstacle: Uses the current velocity of an attached Rigidbody component.

1.8.6 Navigation Layers

Nav layers can be used to restrict pathfinidng operations to certain groups of areas. Each area has one layer that it exists on, and each agent or pathfinding query has a mask of the layers it can traverse. Areas can also be restricted to only generating external links to certain other layers.

1.8.6.1 Layer Setup While not necessary, It will probably be beneficial to give custom names to your navigation layers. This is managed in a similar way to Unity's physics layers, with 32 layers that can be given custom names. If you don't do this, they will simply all be named Layer 0, Layer 1, etc.

To start naming layers, open the HyperNav project settings: `Tools > Infohazard > HyperNav Settings`. Under Nav Layer Names, name as many of the layers as you want (you do not need to name all of them). The layer names are stored in an asset under Resources (by default, `Assets/Infohazard.Core.Data/Resources/HyperNavSettings.asset`). You can move this file anywhere you want in your project as long as it is directly under a Resources folder.

1.8.6.2 Layer Usage You can use layers to restrict entire areas to a specific type of agent, such as a ceiling/wall that only characters with a wall-climbing ability can walk on. You can also cover the same space with multiple areas on different layers. This can be used, for example, to acommodate multiple agent sizes. When overlapping areas, I highly suggest disabling external links between them using the `ExternalLinkTargetLayers` to avoid creating a huge number of external links.

1.8.6.3 Layer Scripting You can reference layers and masks in your code by using the `NavLayer` and `NavLayerMask` types respectively. The best way to set these values is to assign them in the inspector, as a drop-down will be provided automatically similar to Unity's `LayerMask` type. If needed, you can also look up layers by name using `NavLayer.Get(string name)`. `NavLayer` can be converted to `NavLayerMask` using `NavLayer.AsMask` or `NavLayerMask.Combine`. `NavLayerMask` also supports all the bitwise operations you would expect from a mask type.

1.8.7 Moving Volumes and Floating Origin

You may need to occasionally move NavVolumes, either individually or all together. When this occurs, the native data for the volumes needs to be updated. For moving individual volumes, you can enable `AutoDetectMovement` on the volumes and the data updating will be handled automatically. For moving all volumes at once such as in a floating origin system, it will be more performant to use `NavVolume.UpdateAllTransforms()` after they are moved.

1.8.7.1 Moving Volume Limitations In both cases, any pathfinding operations currently underway will automatically be restarted. Any agents that already have paths, however, will need to be manually stopped by calling `Stop()` and the destination re-supplied in shifted space.

Moving volumes does update the external link positions, so two volumes with links to one another that move together will have the links updated correctly. However, if one volume moves relative to another volume that it shares links with, the link positions will stay relative to the originating volume. To account for this, you'll need to regenerate the links at runtime.

While individual moving volumes are supported, note that when a volume is moved, any calculating paths will be canceled. So if a volume is moving every frame, the paths will be continuously canceled before they can complete. You can get past this by putting the NavPathfinder in `Instantaneous` mode, but that will likely lead to stutters.

2 Changelog

All notable changes to this project will be documented in this file.

The format is based on Keep a Changelog, and this project adheres to Semantic Versioning.

2.1 [2.3.0] - 2026-8-24

2.1.1 Added

- New NavLayer feature allows restricting agents to only traveling on certain areas based on configurable layers.

2.2 [2.2.0] - 2025-8-14

2.2.1 Added

- Added backtracking option to enable agents to move to a previous point in their path if they can't reach their next waypoint.
- Added option to check surface normal when sampling.

2.2.2 Changed

- Baking multiple areas in the editor now happens sequentially rather than in parallel. This avoids monopolizing the CPU.

2.2.3 Fixed

- Fixed look SimpleNavAgentMover look vector not being normalized.
- Fixed an allocation bug that caused the wrong allocator to be used for path jobs.
- Fixed a bug where a spline agent could have a spline path with no points.
- Fixed a bug with frame count in NavPathfinder that caused TempJob allocations to be retained for too long.
- Fixed a possible bug where the number of physics queries in a bake step could exceed int.MaxValue.
- Fixed a bug where a path could be instantly marked as completed.
- Fixed manual links not working for surfaces.
- Fixed a major bug where external links would not be saved properly immediately after baking area data.

2.3 [2.1.0] - 2025-4-5

2.3.1 Added

- Ability to create an asset to share settings between multiple areas.

2.3.2 Changed

- The settings of each area are now stored in a serialized object, to simplify the implementation of settings assets. The settings are migrated from the old fields so existing areas shouldn't be broken.

2.3.3 Fixed

- Fixed area IDs not being calculated correctly in prefabs.
- Fixed incorrect error message when trying to bake a surface in custom upright mode with no custom handler.
- Fixed a build error.
- Fixed baking in prefab mode not correctly using prefab objects.
- Fixed a possible invalid memory access if multiple volumes were added with the same ID.

2.4 [2.0.0] - 2025-2-11

2.4.1 Added

- Support for runtime data generation.
  - Baking of volumes should be much faster at edit time as well.
  - Both the volume data itself and the external links can be generated at runtime.
  - Added new demo scenes for this functionality.
- Surface-based omnidirectional navigation.
  - A new type of area called a NavSurface is introduced.
  - This allows pathfinding for agents that walk on surfaces.
  - Supports standard gravity, omnidirectional normal-based gravity, and custom upward directions.
  - Integrates with volume-based navigation via external links.
  - A single pathfinding job can include both surface and volume-based navigation.
- Pre-made mover component.
  - A script is provided to get an agent up and running quickly, that can move towards or follow a target.
  - Handles some of the most common agent behaviors such as re-pathing when the target moves or after a certain time.
  - Supports Transform and Rigidbody-based movement.
  - Supports surface and volume-based navigation.
- Manual external links which can be turned on and off.
- Greatly improved runtime performance of sampling NavVolumes.
- Added a way to manually supply NavSampleResults to an agent when finding a path.
- Added step-by-step visualization so you can see the volume data being built at each step.

2.4.2 Changed

- Minimum Unity version changed to 2022.3.

2.4.3 Fixed

- Fixed some cases where an agent could be stuck between two volumes and continuously re-path to the previous one.

2.5 [1.1.6] - 2023-6-21

2.5.1 Fixed

- Fixed a case where SplinePath data was not disposed.

2.6 [1.1.5] - 2023-3-10

2.6.1 Fixed

- Better prefab support for baked volumes.
  - Can now bake a volume in prefab mode.
  - Volumes that are prefab instances update their ID but do not unset their data reference.
  - Volumes baked in a prefab are stored in the prefab folder.

2.7 [1.1.4] - 2023-2-24

2.7.1 Added

- Support for moving volumes and floating origin, including a new demo scene.
  - External link positions are now stored in local space, but previously generated world-space links will still function.
  - Individual volumes can auto-detect movement, or you can call NavVolume.UpdateAllTransforms for floating origin.

2.8 [1.1.3] - 2023-1-19

2.8.1 Added

- New baking option EnableMultiQuery which fixes bake issues when VoxelSize is larger than MaxAgentRadius.
  - EnableMultiQuery is enabled by default and should be kept on unless you have a good reason to disable it.
  - Added a new visualization option that will show you what the query looks like while baking.

2.8.2 Fixed

- Fixed a mesh simplification issue that led to "Did not find vertex to clip."
- Fixed a region concavity issue that led to infinite bake time.

2.9 [1.1.2] - 2023-1-18

2.9.1 Fixed

- Fixed errors due to volume native data not being correctly initialized when "Reload Scene" is disabled in the editor settings.
- Fixed errors due to volumes being loaded or unloaded while a path is in progress.

2.10 [1.1.1] - 2022-12-14

2.10.1 Fixed

- Fixed a NullReferenceException that occurred when baking multiple volumes that didn't have their data created yet.

2.11 [1.1.0] - 2022-11-22

2.11.1 Added

- New avoidance system.
  - Avoidance can be used either on its own or with a NavAgent or SplineNavAgent.
  - Any object can be an obstacle that is avoided by agents.
  - Agents avoid obstacles as well as each other.
  - Uses the ORCA algorithm, adapted from the RVO2-3D library from University of North Carolina, which is licensed under Apache 2.0.
- Added an option to help prevent spline paths from entering blocked regions by raycasting the tangents.
- Added a button to bake all active NavVolumes at once.
- SplineNavAgents can now get un-stuck by raycasting to see if they are stopped on blocking triangles.

2.11.2 Changed

- Volume identifier now depends on the scene GUID, so a save-as can no longer cause volumes in two scenes to have the same ID.
  - This causes all volume IDs to change. Upon opening a scene with existing volumes, a prompt to migrate the IDs will appear.
- SplinePath is now a struct and uses native data, no longer requiring managed memory allocations. This also means it's Burst-compatible.
- Improved performance of spline projecting by using Burst.

2.11.3 Fixed

- Fixed several cases where projecting on a spline path would be incorrect.

2.12 [1.0.0] - 2022-11-08

2.12.1 Added

- Initial release, all files and documentation added.

3 Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Infohazard.HyperNav.Avoidance 27

Infohazard.HyperNav.Jobs.AvoidanceJob 31

Infohazard.HyperNav.AvoidanceManager 34

Infohazard.HyperNav.Jobs.Baking.Surface.CalculateVertexAnglesJob 38

Infohazard.HyperNav.Edge 38

Infohazard.HyperNav.Editor.EditorNavAreaBakingUtility 42

Infohazard.HyperNav.Editor.ExternalLinkEditorUtility 46

Infohazard.HyperNav.Jobs.Utility.Fast3DArray 47

Infohazard.HyperNav.IAvoidanceObstacle 52

- **Infohazard.HyperNav.AvoidanceObstacleBase** 36
  - **Infohazard.HyperNav.AvoidanceAgent** 28
  - **Infohazard.HyperNav.RigidbodyAvoidanceObstacle** 173
  - **Infohazard.HyperNav.SimpleAvoidanceObstacle** 174
- **Infohazard.HyperNav.IAvoidanceAgent** 50
  - **Infohazard.HyperNav.AvoidanceAgent** 28

Infohazard.HyperNav.Jobs.Utility.IntList2 53

Infohazard.HyperNav.Jobs.Baking.InvertIndexArrayJob 56

Infohazard.HyperNav.ISurfaceUprightDirectionHandler 56

- **Infohazard.HyperNav.SurfaceUprightDirectionHandlerWithOverrides** 199

Infohazard.HyperNav.Jobs.Utility.MarchingCubesCavityTables 57

Infohazard.HyperNav.Jobs.Utility.MarchingCubesTables 58

Infohazard.HyperNav.Jobs.NativeAvoidanceObstacleData 61

Infohazard.HyperNav.Jobs.Utility.NativeBounds 62

Infohazard.HyperNav.Jobs.Utility.NativeMathUtility 64

Infohazard.HyperNav.Jobs.Utility.NativeNavExternalLinkData 69

Infohazard.HyperNav.NativeNavSurfaceDataPointers 72

Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData 72

Infohazard.HyperNav.NativeNavVolumeDataPointers 77

Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeInternalLinkData 77

Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeRegionData 79

Infohazard.HyperNav.Jobs.Utility.NativeNavWaypoint 81

Infohazard.HyperNav.Jobs.NativePlane 82

Infohazard.HyperNav.Jobs.NativeRay 84

Infohazard.HyperNav.Jobs.NativeRaycastElement 85

Infohazard.HyperNav.NavAgent 86

- Infohazard.HyperNav.SplineNavAgent 184

Infohazard.HyperNav.NavArea< NavSurface, NavSurfaceData, NativeNavSurfaceData, NativeNav↩ SurfaceDataPointers, NavSurfaceSettings > 98

Infohazard.HyperNav.NavArea< NavVolume, NavVolumeData, NativeNavVolumeData, NativeNav↩ VolumeDataPointers, NavVolumeSettings > 98

- Infohazard.HyperNav.NavVolume 160

Infohazard.HyperNav.NavAreaBakeProgress 103

Infohazard.HyperNav.NavAreaBakingUtility 103

Infohazard.HyperNav.NavAreaBase 104

- Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings > 98

Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf > 110

Infohazard.HyperNav.Settings.NavAreaBaseSettings< NavSurfaceSettings > 110

- Infohazard.HyperNav.Settings.NavSurfaceSettings 155

Infohazard.HyperNav.Settings.NavAreaBaseSettings< NavVolumeSettings > 110

- Infohazard.HyperNav.Settings.NavVolumeSettings 166

Infohazard.HyperNav.Settings.NavAreaBaseSettingsAsset< TData > 113

Infohazard.HyperNav.Settings.NavAreaBaseSettingsAsset< NavSurfaceSettings > 113

- Infohazard.HyperNav.Settings.NavSurfaceSettingsAsset 159

Infohazard.HyperNav.Settings.NavAreaBaseSettingsAsset< NavVolumeSettings > 113

- Infohazard.HyperNav.Settings.NavVolumeSettingsAsset 168

Infohazard.HyperNav.Settings.NavAreaBaseSettingsAsset< TSettings > 113

Infohazard.HyperNav.NavAreaExternalLinkUpdate 114

Infohazard.HyperNav.Editor.NavAreaMigrator 115

Infohazard.HyperNav.Editor.NavEditorUtility 115

Infohazard.HyperNav.NavExternalLinkData 117

Infohazard.HyperNav.NavInternalLinkData 119

Infohazard.HyperNav.NavLayer 123

Infohazard.HyperNav.NavLayerMask 126

Infohazard.HyperNav.Jobs.NavMultiRaycastJob 132

Infohazard.HyperNav.NavPath 133

Infohazard.HyperNav.NavPathfinder 135

Infohazard.HyperNav.Jobs.NavPathJob 139

Infohazard.HyperNav.Jobs.NavRaycastJob 142

Infohazard.HyperNav.NavRegionData 143

Infohazard.HyperNav.Jobs.Utility.NavSampleResult 146

Infohazard.HyperNav.NavSurfaceBakeHandler 149

Infohazard.HyperNav.NavSurfaceInternalLinkData 150

Infohazard.HyperNav.NavSurfaceRegionData 152

Infohazard.HyperNav.NavSurfaceUpdate 159

Infohazard.HyperNav.NavVolumeBakeHandler 162

Infohazard.HyperNav.NavVolumeData 163

Infohazard.HyperNav.Editor.NavVolumeEditor 166

Infohazard.HyperNav.NavVolumeUpdate 168

Infohazard.HyperNav.NavWaypoint 169

Infohazard.HyperNav.NavAgent.PropNames 170

Infohazard.HyperNav.NavAreaBase.PropNames 170

Infohazard.HyperNav.NavPathfinder.PropNames 171

Infohazard.HyperNav.NavSurface.PropNames 171

Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf >.PropNames 171

Infohazard.HyperNav.Settings.NavAreaBaseSettingsAsset< TData >.PropNames 171

Infohazard.HyperNav.Settings.NavSurfaceSettings.PropNames 172

Infohazard.HyperNav.Settings.NavVolumeSettings.PropNames 172

Infohazard.HyperNav.SimpleNavAgentMover.PropNames 172

Infohazard.HyperNav.Jobs.Utility.ReadOnlyArrayPointer< T > 172

Infohazard.HyperNav.Jobs.Utility.ReadOnlyArrayPointer< float4 > 172

Infohazard.HyperNav.Jobs.Utility.ReadOnlyArrayPointer< IntList2 > 172

Infohazard.HyperNav.Jobs.Baking.Surface.RemoveJaggedTrianglesJob 173

Infohazard.HyperNav.SimpleNavAgentMover 175

Infohazard.HyperNav.SplinePath 190

Infohazard.HyperNav.SplinePoint 196

Infohazard.HyperNav.Jobs.SplineProjectJob 197

Infohazard.HyperNav.Triangle 200

Infohazard.HyperNav.Jobs.Utility.TriangleRegionIndices 204

Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T > 204

Infohazard.HyperNav.Jobs.Utility.UnsafeArray< float4 > 204

Infohazard.HyperNav.Jobs.Utility.UnsafeArray< Infohazard.HyperNav.Jobs.NativePlane > 204

Infohazard.HyperNav.Jobs.Utility.UnsafeArray< Infohazard.HyperNav.Jobs.Utility.NativeNav↩ ExternalLinkData > 204

Infohazard.HyperNav.Jobs.Utility.UnsafeArray< Infohazard.HyperNav.Jobs.Utility.NativeNav↩ VolumeInternalLinkData > 204

Infohazard.HyperNav.Jobs.Utility.UnsafeArray< Infohazard.HyperNav.Jobs.Utility.NativeNav↩ VolumeRegionData > 204

Infohazard.HyperNav.Jobs.Utility.UnsafeArray< int > 204

Infohazard.HyperNav.Jobs.Utility.UnsafeArray< int2 > 204

Infohazard.HyperNav.Jobs.Utility.UnsafeArray< int3 > 204

Infohazard.HyperNav.Jobs.Utility.UnsafeArray< NativeNavSurfaceInternalLinkData > 204

Infohazard.HyperNav.Jobs.Utility.UnsafeArray< NativeNavSurfaceRegionData > 204

Infohazard.HyperNav.Jobs.Utility.UnsafeArray< UnsafeList< int > > 204

Infohazard.HyperNav.Jobs.Utility.UnsafeConcurrentQueue< T > 208

Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T > 208

Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< PendingPathNode > 208

4 Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Infohazard.HyperNav.Avoidance
Static container that keeps track of all avoidance agents and obstacles. 27

Infohazard.HyperNav.AvoidanceAgent
Base implementation of IAvoidanceAgent that should work in most scenarios. 28

Infohazard.HyperNav.Jobs.AvoidanceJob
Job that calculates the IAvoidanceAgent.AvoidanceVelocity of all IAvoidanceAgents. 31

Infohazard.HyperNav.AvoidanceManager
Handles calculating the avoidance velocities for all IAvoidanceAgents. 34

Infohazard.HyperNav.AvoidanceObstacleBase
Base class for obstacles that are not expected to perform avoidance, but are avoided by agents. 36

Infohazard.HyperNav.Jobs.Baking.Surface.CalculateVertexAnglesJob
Used to calculate the offset from each vertex to check for collisions. For a flat or convex vertex, the offset is zero; however, for a concave vertex, the collision check must be offset in order to avoid the collision check hitting the ground. 38

Infohazard.HyperNav.Edge
Represents the indices of an edge (two connected vertices) in an indexed mesh. 38

Infohazard.HyperNav.Editor.EditorNavAreaBakingUtility
Manages multiple NavAreaBaseBakeHandlers for baking NavAreaBases. 42

Infohazard.HyperNav.Editor.ExternalLinkEditorUtility
Utilities for generating the external links of NavVolumes. 46

Infohazard.HyperNav.Jobs.Utility.Fast3DArray
A data structure equivalent to a three-dimensional int array (int[,,]), but more efficient. 47

Infohazard.HyperNav.IAvoidanceAgent
Interface for objects that both can be avoided and themselves avoid other obstacles using the avoidance system. 50

Infohazard.HyperNav.IAvoidanceObstacle
Interface for objects that can be avoided using the avoidance system. 52

Infohazard.HyperNav.Jobs.Utility.IntList2
Serves as a list of ints with a max size of 2, to avoid allocations. 53

Infohazard.HyperNav.Jobs.Baking.InvertIndexArrayJob
Given an array of indices, creates an array where the value at each index is the index of that value in the input array. OutputArray is expected to be at least as large as the maximum value in InputArray plus one. 56

Infohazard.HyperNav.ISurfaceUprightDirectionHandler
Interface for custom logic to determine the upright direction of a surface. 56

Infohazard.HyperNav.Jobs.Utility.MarchingCubesCavityTables
Tables for determining which cubes and combinations of cubes from the Marching Cubes algorithm will produce concave results. 57

Infohazard.HyperNav.Jobs.Utility.MarchingCubesTables
Tables used in the Marching Cubes algorithm. 58

Infohazard.HyperNav.Jobs.NativeAvoidanceObstacleData
Represents one obstacle (which may be an agent) in the avoidance system. 61

Infohazard.HyperNav.Jobs.Utility.NativeBounds
A native-friendly of a bounding box. 62

Infohazard.HyperNav.Jobs.Utility.NativeMathUtility
Provides math operations that are compatible with Burst. 64

Infohazard.HyperNav.Jobs.Utility.NativeNavExternalLinkData
The native-friendly data representing a connection from one region to another region in another volume. 69

Infohazard.HyperNav.NativeNavSurfaceDataPointers
References to the NativeArrays allocated for a NativeNavSurfaceData. 72

Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData
The baked data of a NavVolume, converted to a form compatible with Burst. 72

Infohazard.HyperNav.NativeNavVolumeDataPointers
References to the NativeArrays allocated for a NativeNavVolumeData. 77

Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeInternalLinkData
The native-friendly data representing a connection from one region to another region in the same volume. 77

Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeRegionData
The native-friendly data representing a single region in a NavVolume. 79

Infohazard.HyperNav.Jobs.Utility.NativeNavWaypoint
A structure used by the navigation job to return the waypoints of a path. 81

Infohazard.HyperNav.Jobs.NativePlane
A plane constructed using native math types. 82

Infohazard.HyperNav.Jobs.NativeRay
A ray constructed using native math types. 84

Infohazard.HyperNav.Jobs.NativeRaycastElement
A single raycast in a NavMultiRaycastJob. 85

Infohazard.HyperNav.NavAgent
A script that can be used to calculate paths by any entity that needs to use HyperNav for navigation. 86

Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >
Generic base class for areas in which HyperNav pathfinding can occur. 98

Infohazard.HyperNav.NavAreaBakeProgress
Represents current bake state of a volume, including progress fraction and current operation display name. 103

Infohazard.HyperNav.NavAreaBakingUtility
Utility methods used for baking NavAreas in both the editor and at runtime. 103

Infohazard.HyperNav.NavAreaBase
Base class for areas in which HyperNav pathfinding can occur. 104

Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf >
Base settings data for NavAreas. 110

Infohazard.HyperNav.Settings.NavAreaBaseSettingsAsset< TData >
Base class for all NavArea settings assets. 113

Infohazard.HyperNav.NavAreaExternalLinkUpdate
Utility for updating external links for nav areas. 114

Infohazard.HyperNav.Editor.NavAreaMigrator
Handles migration of NavArea data to new versions. 115

Infohazard.HyperNav.Editor.NavEditorUtility
Utility functions used internally, but you can use them too, I mean I'm not your boss. 115

Infohazard.HyperNav.NavExternalLinkData
A connection from one region to another region in another volume. 117

Infohazard.HyperNav.NavInternalLinkData
A connection from one region to another region in the same volume. 119

Infohazard.HyperNav.NavLayer
A single NavLayer. May have the value 0 to 31, or -1 for invalid. 123

Infohazard.HyperNav.NavLayerMask
A mask of NavLayers with one bit for each possible layer. 126

Infohazard.HyperNav.Jobs.NavMultiRaycastJob
Job that performs multiple raycasts in one or more NavVolumes in parallel. 132

Infohazard.HyperNav.NavPath
A completed, valid path. 133

Infohazard.HyperNav.NavPathfinder
A script used to calculate HyperNav paths. 135

Infohazard.HyperNav.Jobs.NavPathJob
Job used to find a HyperNav path. 139

Infohazard.HyperNav.Jobs.NavRaycastJob
Job that performs a single raycast in a NavVolume. 142

Infohazard.HyperNav.NavRegionData
The serialized data representing a single region in a NavVolume. 143

Infohazard.HyperNav.Jobs.Utility.NavSampleResult
A native-friendly representation of a navigation query result. 146

Infohazard.HyperNav.NavSurfaceBakeHandler
Manages the baking process for a NavSurface. Cannot be reused. 149

Infohazard.HyperNav.NavSurfaceInternalLinkData
A connection from one region to another region in the same surface. 150

Infohazard.HyperNav.NavSurfaceRegionData
The serialized data representing a single region in a NavSurface. 152

Infohazard.HyperNav.Settings.NavSurfaceSettings
Settings for baking NavSurfaces. 155

Infohazard.HyperNav.Settings.NavSurfaceSettingsAsset
Settings asset for baking NavSurfaces. 159

Infohazard.HyperNav.NavSurfaceUpdate
Utility class for updating NavSurface data. 159

Infohazard.HyperNav.NavVolume
A volume of space in which HyperNav pathfinding can occur. 160

Infohazard.HyperNav.NavVolumeBakeHandler
Manages the baking process for a NavVolume. Cannot be reused. 162

Infohazard.HyperNav.NavVolumeData
The baked data of a NavVolume, saved as an asset. 163

Infohazard.HyperNav.Editor.NavVolumeEditor
Custom editor for Infohazard.HyperNav.NavVolume. 166

Infohazard.HyperNav.Settings.NavVolumeSettings
Settings for baking NavVolumes. 166

Infohazard.HyperNav.Settings.NavVolumeSettingsAsset
Settings asset for baking NavVolumes. 168

Infohazard.HyperNav.NavVolumeUpdate
Utility functions for updating NavVolume data. 168

Infohazard.HyperNav.NavWaypoint
A waypoint in a completed path. 169

Infohazard.HyperNav.NavAgent.PropNames
This is used to refer to the names of private fields in this class from a custom Editor. 170

Infohazard.HyperNav.NavAreaBase.PropNames
This is used to refer to the names of private fields in this class from a custom Editor. 170

Infohazard.HyperNav.NavPathfinder.PropNames
This is used to refer to the names of private fields in this class from a custom Editor. 171

Infohazard.HyperNav.NavSurface.PropNames
This is used to refer to the names of private fields in this class from a custom Editor. 171

Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf >.PropNames
This is used to refer to the names of private fields in this class from a custom Editor. 171

Infohazard.HyperNav.Settings.NavAreaBaseSettingsAsset< TData >.PropNames
This is used to refer to the names of private fields in this class from a custom Editor. 171

Infohazard.HyperNav.Settings.NavSurfaceSettings.PropNames
This is used to refer to the names of private fields in this class from a custom Editor. 172

Infohazard.HyperNav.Settings.NavVolumeSettings.PropNames
Serialized property names for NavVolumeSettings. 172

Infohazard.HyperNav.SimpleNavAgentMover.PropNames
This is used to refer to the names of private fields in this class from a custom Editor. 172

Infohazard.HyperNav.Jobs.Utility.ReadOnlyArrayPointer< T >
Reference over a native array, which does not allow the memory to be modified. 172

Infohazard.HyperNav.Jobs.Baking.Surface.RemoveJaggedTrianglesJob
Remove triangles that have a vertex that is not part of any other triangle. This cleans up the edges of the mesh, removing jagged edges. 173

Infohazard.HyperNav.RigidbodyAvoidanceObstacle
An IAvoidanceObstacle that gets its IAvoidanceObstacle.InputVelocity from a Rigidbody. 173

Infohazard.HyperNav.SimpleAvoidanceObstacle
An IAvoidanceObstacle that gets its IAvoidanceObstacle.InputVelocity by measuring its position/time delta. 174

Infohazard.HyperNav.SimpleNavAgentMover
A script that enables using a NavAgent without writing any code. 175

Infohazard.HyperNav.SplineNavAgent
A script that can be used to calculate smooth paths by any entity that needs to use HyperNav for navigation. 184

Infohazard.HyperNav.SplinePath
A spline specialized for path following, created with a NavPath. 190

Infohazard.HyperNav.SplinePoint
Represents a point on a spline and the segment that starts with it. 196

Infohazard.HyperNav.Jobs.SplineProjectJob
Job used to find the parameter along a spline that is nearest to the given point. 197

Infohazard.HyperNav.SurfaceUprightDirectionHandlerWithOverrides
Serves as an example of how to implement a custom upright direction handler for surfaces, including the related jobs. 199

Infohazard.HyperNav.Triangle
Represents the indices of a triangle (three vertices by a face) in an indexed mesh. 200

Infohazard.HyperNav.Jobs.Utility.TriangleRegionIndices
Serves as a Dictionary of ints with a max size of 2, to avoid allocations. This works because a triangle can be part of at most two regions. 204

Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T >
This is a simple wrapper for unmanaged memory which bypasses Unity's safety checks. This allows arrays to be nested in other arrays (or in structs contained in arrays). Note that you must keep a reference to the original NativeArray, or Unity will detect a memory leak. 204

Infohazard.HyperNav.Jobs.Utility.UnsafeConcurrentQueue< T >
A thread-safe queue for use in Burst jobs. 208

Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T >
An implementation of a Max Heap that can be used with jobs and Burst. 208

5 Namespace Documentation

5.1 Infohazard Namespace Reference

5.2 Infohazard.HyperNav Namespace Reference

Classes

- class Avoidance

  Static container that keeps track of all avoidance agents and obstacles.
- class AvoidanceAgent

  Base implementation of IAvoidanceAgent that should work in most scenarios.
- class AvoidanceManager

  Handles calculating the avoidance velocities for all IAvoidanceAgents.
- class AvoidanceObstacleBase

  Base class for obstacles that are not expected to perform avoidance, but are avoided by agents.
- struct Edge

Represents the indices of an edge (two connected vertices) in an indexed mesh.

- interface IAvoidanceAgent

  Interface for objects that both can be avoided and themselves avoid other obstacles using the avoidance system.
- interface IAvoidanceObstacle

  Interface for objects that can be avoided using the avoidance system.
- interface ISurfaceUprightDirectionHandler

  Interface for custom logic to determine the upright direction of a surface.
- struct NativeNavSurfaceDataPointers

  References to the NativeArrays allocated for a NativeNavSurfaceData.
- struct NativeNavVolumeDataPointers

  References to the NativeArrays allocated for a NativeNavVolumeData.
- class NavAgent

  A script that can be used to calculate paths by any entity that needs to use HyperNav for navigation.
- class NavArea

  Generic base class for areas in which HyperNav pathfinding can occur.
- struct NavAreaBakeProgress

  Represents current bake state of a volume, including progress fraction and current operation display name.
- class NavAreaBakingUtility

  Utility methods used for baking NavAreas in both the editor and at runtime.
- class NavAreaBase

  Base class for areas in which HyperNav pathfinding can occur.
- class NavAreaExternalLinkUpdate

  Utility for updating external links for nav areas.
- class NavExternalLinkData

  A connection from one region to another region in another volume.
- class NavInternalLinkData

  A connection from one region to another region in the same volume.
- struct NavLayer

  A single NavLayer. May have the value 0 to 31, or -1 for invalid.
- struct NavLayerMask

  A mask of NavLayers with one bit for each possible layer.
- class NavPath

  A completed, valid path.
- class NavPathfinder

  A script used to calculate HyperNav paths.
- class NavRegionData

  The serialized data representing a single region in a NavVolume.
- class NavSurfaceBakeHandler

  Manages the baking process for a NavSurface. Cannot be reused.
- class NavSurfaceInternalLinkData

  A connection from one region to another region in the same surface.
- class NavSurfaceRegionData

  The serialized data representing a single region in a NavSurface.
- class NavSurfaceUpdate

  Utility class for updating NavSurface data.
- class NavVolume

  A volume of space in which HyperNav pathfinding can occur.
- class NavVolumeBakeHandler

  Manages the baking process for a NavVolume. Cannot be reused.
- class NavVolumeData

  The baked data of a NavVolume, saved as an asset.

- class NavVolumeUpdate

  Utility functions for updating NavVolume data.
- struct NavWaypoint

  A waypoint in a completed path.
- class RigidbodyAvoidanceObstacle

  An IAvoidanceObstacle that gets its IAvoidanceObstacle.InputVelocity from a Rigidbody.
- class SimpleAvoidanceObstacle

  An IAvoidanceObstacle that gets its IAvoidanceObstacle.InputVelocity by measuring its position/time delta.
- class SimpleNavAgentMover

  A script that enables using a NavAgent without writing any code.
- class SplineNavAgent

  A script that can be used to calculate smooth paths by any entity that needs to use HyperNav for navigation.
- struct SplinePath

  A spline specialized for path following, created with a NavPath.
- struct SplinePoint

  Represents a point on a spline and the segment that starts with it.
- class SurfaceUprightDirectionHandlerWithOverrides

  Serves as an example of how to implement a custom upright direction handler for surfaces, including the related jobs.
- struct Triangle

  Represents the indices of a triangle (three vertices by a face) in an indexed mesh.

Enumerations

- enum AvoidanceManagerUpdateMode

  The update modes in which AvoidanceManager can operate.
- enum NavPathfindingMode

  The modes in which pathfinding can be executed.
- enum NavWaypointType

  Types of waypoint in a completed path.
- enum NavSurfaceVisualizationMode

  The various modes available to generate a preview mesh in the editor for visualization.
- enum NavVolumeVisualizationMode

  The various modes available to generate a preview mesh in the editor for visualization.
- enum NavAreaTypes

  Type of NavArea.

Functions

- delegate void HyperNavPathCallback (long id, NavPath path)

  A callback that receives a path when it is complete.

5.2.1 Enumeration Type Documentation

5.2.1.1 AvoidanceManagerUpdateMode `enum Infohazard.HyperNav.AvoidanceManagerUpdateMode`

The update modes in which AvoidanceManager can operate.

5.2.1.2 NavAreaTypes `enum Infohazard.HyperNav.NavAreaTypes`

Type of NavArea.

5.2.1.3 NavPathfindingMode `enum Infohazard.HyperNav.NavPathfindingMode`

The modes in which pathfinding can be executed.

5.2.1.4 NavSurfaceVisualizationMode `enum Infohazard.HyperNav.NavSurfaceVisualizationMode`

The various modes available to generate a preview mesh in the editor for visualization.

5.2.1.5 NavVolumeVisualizationMode `enum Infohazard.HyperNav.NavVolumeVisualizationMode`

The various modes available to generate a preview mesh in the editor for visualization.

5.2.1.6 NavWaypointType `enum Infohazard.HyperNav.NavWaypointType`

Types of waypoint in a completed path.

5.2.2 Function Documentation

5.2.2.1 HyperNavPathCallback() `delegate void Infohazard.HyperNav.HyperNavPathCallback (`

```
long id,
NavPath path )
```

A callback that receives a path when it is complete.

5.3 Infohazard.HyperNav.Editor Namespace Reference

Classes

- class EditorNavAreaBakingUtility

  Manages multiple NavAreaBaseBakeHandlers for baking NavAreaBases.
- class ExternalLinkEditorUtility

  Utilities for generating the external links of NavVolumes.
- class NavAreaMigrator

  Handles migration of NavArea data to new versions.
- class NavEditorUtility

  Utility functions used internally, but you can use them too, I mean I'm not your boss.
- class NavVolumeEditor

  Custom editor for Infohazard.HyperNav.NavVolume.

5.4 Infohazard.HyperNav.Jobs Namespace Reference

Classes

- struct AvoidanceJob

  Job that calculates the IAvoidanceAgent.AvoidanceVelocity of all IAvoidanceAgents.
- struct NativeAvoidanceObstacleData

  Represents one obstacle (which may be an agent) in the avoidance system.
- struct NativePlane

  A plane constructed using native math types.
- struct NativeRay

  A ray constructed using native math types.
- struct NativeRaycastElement

  A single raycast in a NavMultiRaycastJob.
- struct NavMultiRaycastJob

  Job that performs multiple raycasts in one or more NavVolumes in parallel.
- struct NavPathJob

  Job used to find a HyperNav path.
- struct NavRaycastJob

  Job that performs a single raycast in a NavVolume.
- struct SplineProjectJob

  Job used to find the parameter along a spline that is nearest to the given point.

5.5 Infohazard.HyperNav.Jobs.Baking Namespace Reference

Classes

- struct InvertIndexArrayJob

  Given an array of indices, creates an array where the value at each index is the index of that value in the input array. OutputArray is expected to be at least as large as the maximum value in InputArray plus one.

5.6 Infohazard.HyperNav.Jobs.Baking.Surface Namespace Reference

Classes

- struct CalculateVertexAnglesJob

  Used to calculate the offset from each vertex to check for collisions. For a flat or convex vertex, the offset is zero; however, for a concave vertex, the collision check must be offset in order to avoid the collision check hitting the ground.
- struct RemoveJaggedTrianglesJob

  Remove triangles that have a vertex that is not part of any other triangle. This cleans up the edges of the mesh, removing jagged edges.

5.7 Infohazard.HyperNav.Jobs.Baking.Volume Namespace Reference

5.8 Infohazard.HyperNav.Jobs.Utility Namespace Reference

Classes

- struct Fast3DArray

  A data structure equivalent to a three-dimensional int array (int[,,]), but more efficient.
- struct IntList2

  Serves as a list of ints with a max size of 2, to avoid allocations.
- class MarchingCubesCavityTables

  Tables for determining which cubes and combinations of cubes from the Marching Cubes algorithm will produce concave results.
- class MarchingCubesTables

  Tables used in the Marching Cubes algorithm.
- struct NativeBounds

  A native-friendly of a bounding box.
- class NativeMathUtility

  Provides math operations that are compatible with Burst.
- struct NativeNavExternalLinkData

  The native-friendly data representing a connection from one region to another region in another volume.
- struct NativeNavVolumeData

  The baked data of a NavVolume, converted to a form compatible with Burst.
- struct NativeNavVolumeInternalLinkData

  The native-friendly data representing a connection from one region to another region in the same volume.
- struct NativeNavVolumeRegionData

  The native-friendly data representing a single region in a NavVolume.
- struct NativeNavWaypoint

  A structure used by the navigation job to return the waypoints of a path.
- struct NavSampleResult

  A native-friendly representation of a navigation query result.
- struct ReadOnlyArrayPointer

  Reference over a native array, which does not allow the memory to be modified.
- struct TriangleRegionIndices

  Serves as a Dictionary of ints with a max size of 2, to avoid allocations. This works because a triangle can be part of at most two regions.
- struct UnsafeArray

  This is a simple wrapper for unmanaged memory which bypasses Unity's safety checks. This allows arrays to be nested in other arrays (or in structs contained in arrays). Note that you must keep a reference to the original Native↩ Array, or Unity will detect a memory leak.
- struct UnsafeConcurrentQueue

  A thread-safe queue for use in Burst jobs.
- struct UnsafeHeap

  An implementation of a Max Heap that can be used with jobs and Burst.

Enumerations

- enum NavSamplePriority

  How to prioritize results when sampling in both volumes and surfaces.
- enum NavPathState

  The state of a pathfinding request.

5.8.1 Enumeration Type Documentation

5.8.1.1 NavPathState `enum Infohazard.HyperNav.Jobs.Utility.NavPathState`

The state of a pathfinding request.

5.8.1.2 NavSamplePriority `enum Infohazard.HyperNav.Jobs.Utility.NavSamplePriority`

How to prioritize results when sampling in both volumes and surfaces.

5.9 Infohazard.HyperNav.Settings Namespace Reference

Classes

- class NavAreaBaseSettings

  Base settings data for NavAreas.
- class NavAreaBaseSettingsAsset

  Base class for all NavArea settings assets.
- class NavSurfaceSettings

  Settings for baking NavSurfaces.
- class NavSurfaceSettingsAsset

  Settings asset for baking NavSurfaces.
- class NavVolumeSettings

  Settings for baking NavVolumes.
- class NavVolumeSettingsAsset

  Settings asset for baking NavVolumes.

5.10 Infohazard.HyperNav.Tests Namespace Reference

5.11 Infohazard.HyperNav.Tests.UnitTests Namespace Reference

6 Class Documentation

6.1 Infohazard.HyperNav.Avoidance Class Reference

Static container that keeps track of all avoidance agents and obstacles.

Properties

- static List< IAvoidanceObstacle > AllObstacles = new List<IAvoidanceObstacle>() `[get]`

  All active obstacles (including agents).
- static List< IAvoidanceAgent > AllAgents = new List<IAvoidanceAgent>() `[get]`

  All active agents.

6.1.1 Detailed Description

Static container that keeps track of all avoidance agents and obstacles.

6.1.2 Property Documentation

6.1.2.1 AllAgents `List<IAvoidanceAgent> Infohazard.HyperNav.Avoidance.AllAgents = new List<IAvoidanceAgent>( [static], [get]`

All active agents.

6.1.2.2 AllObstacles `List<IAvoidanceObstacle> Infohazard.HyperNav.Avoidance.AllObstacles = new List<IAvoidanceObstacle>() [static], [get]`

All active obstacles (including agents).

The documentation for this class was generated from the following file:

- Runtime/Avoidance/Avoidance.cs

6.2 Infohazard.HyperNav.AvoidanceAgent Class Reference

Base implementation of IAvoidanceAgent that should work in most scenarios.

Public Member Functions

- void UpdateAvoidanceVelocity (Vector3 newAvoidance)

  Called by the system to update AvoidanceVelocity.

Parameters

newAvoidance	*New avoidance velocity*

Protected Member Functions

- override void OnEnable ()

  Resets desired velocity and adds self to list of all obstacles.
- override void OnDisable ()

  Removes self from list of all obstacles.

Properties

- virtual float AvoidanceWeight `[get, set]`
  How much effort the agent will take to avoid obstacles and other agents.
- virtual float AvoidancePadding `[get, set]`
  How much extra space to leave when avoiding obstacles.
- virtual bool DebugAvoidance `[get, set]`
  Whether to draw debugging information in the scene view.
- TagMask AvoidedTags `[get, set]`
  Tags of objects that the agent will try to avoid.
- Func< Vector3 > InputVelocityFunc `[get, set]`
  Function used to calculate InputVelocity.
- override Vector3 InputVelocity `[get]`
  The object's desired (or actual) velocity.
- virtual bool IsActive `[get, set]`
  Whether the agent should actively avoid obstacles. If false, will still behave as an obstacle.
- virtual Vector3 AvoidanceVelocity `[get]`
  The velocity the agent should have in order to avoid collisions with obstacles and other agents.
- virtual Vector3 NormalizedAvoidanceVelocity `[get]`
  Avoidance velocity divided by max speed, so it is in [0, 1] range.

Events

- Action< Vector3 > AvoidanceUpdated
  Invoked when avoidance is updated.

6.2.1 Detailed Description

Base implementation of IAvoidanceAgent that should work in most scenarios.

It gets its desired velocity from a delegate, so you can point it to whatever system you need.

6.2.2 Member Function Documentation

6.2.2.1 OnDisable() `override void Infohazard.HyperNav.AvoidanceAgent.OnDisable ( ) [protected], [virtual]`

Removes self from list of all obstacles.

Reimplemented from Infohazard.HyperNav.AvoidanceObstacleBase.

6.2.2.2 OnEnable() `override void Infohazard.HyperNav.AvoidanceAgent.OnEnable ( ) [protected], [virtual]`

Resets desired velocity and adds self to list of all obstacles.

Reimplemented from Infohazard.HyperNav.AvoidanceObstacleBase.

6.2.2.3 UpdateAvoidanceVelocity() `void Infohazard.HyperNav.AvoidanceAgent.UpdateAvoidance↩Velocity ( Vector3 newAvoidance )`

Called by the system to update AvoidanceVelocity.

Parameters

newAvoidance	New avoidance velocity

Implements Infohazard.HyperNav.IAvoidanceAgent.

6.2.3 Property Documentation

6.2.3.1 AvoidancePadding `virtual float Infohazard.HyperNav.AvoidanceAgent.AvoidancePadding [get], [set], [add]`

How much extra space to leave when avoiding obstacles.

Implements Infohazard.HyperNav.IAvoidanceAgent.

6.2.3.2 AvoidanceVelocity `virtual Vector3 Infohazard.HyperNav.AvoidanceAgent.AvoidanceVelocity [get]`

The velocity the agent should have in order to avoid collisions with obstacles and other agents.

Implements Infohazard.HyperNav.IAvoidanceAgent.

6.2.3.3 AvoidanceWeight `virtual float Infohazard.HyperNav.AvoidanceAgent.AvoidanceWeight [get], [set]`

How much effort the agent will take to avoid obstacles and other agents.

Implements Infohazard.HyperNav.IAvoidanceAgent.

6.2.3.4 AvoidedTags `TagMask Infohazard.HyperNav.AvoidanceAgent.AvoidedTags [get], [set]`

Tags of objects that the agent will try to avoid.

Implements Infohazard.HyperNav.IAvoidanceAgent.

6.2.3.5 DebugAvoidance `virtual bool Infohazard.HyperNav.AvoidanceAgent.DebugAvoidance [get], [set]`

Whether to draw debugging information in the scene view.

Implements Infohazard.HyperNav.IAvoidanceAgent.

6.2.3.6 InputVelocity `override Vector3 Infohazard.HyperNav.AvoidanceAgent.InputVelocity [get]`

The object's desired (or actual) velocity.

Implements Infohazard.HyperNav.IAvoidanceObstacle.

6.2.3.7 InputVelocityFunc `Func<Vector3> Infohazard.HyperNav.AvoidanceAgent.InputVelocityFunc [get], [set]`

Function used to calculate InputVelocity.

6.2.3.8 IsActive `virtual bool Infohazard.HyperNav.AvoidanceAgent.IsActive [get], [set]`

Whether the agent should actively avoid obstacles. If false, will still behave as an obstacle.

Implements Infohazard.HyperNav.IAvoidanceAgent.

6.2.3.9 NormalizedAvoidanceVelocity `virtual Vector3 Infohazard.HyperNav.AvoidanceAgent.NormalizedAvoidanceVelocity [get]`

Avoidance velocity divided by max speed, so it is in [0, 1] range.

6.2.4 Event Documentation

6.2.4.1 AvoidanceUpdated `Action<Vector3> Infohazard.HyperNav.AvoidanceAgent.AvoidanceUpdated`

Invoked when avoidance is updated.

The documentation for this class was generated from the following file:

- Runtime/Avoidance/AvoidanceAgent.cs

6.3 Infohazard.HyperNav.Jobs.AvoidanceJob Struct Reference

Job that calculates the IAvoidanceAgent.AvoidanceVelocity of all IAvoidanceAgents.

Public Member Functions

- void Execute (int index)

  Run on a single agent index.
- float3 OrcaAvoidance (int agentIndex, ref NativeArray< NativePlane > agentTempPlanes, ref NativeArray< NativePlane > agentTempProjPlanes)

  Perform avoidance calculation for an agent using the ORCA (optimal reciprocal collision avoidance) algorithm.

Public Attributes

- NativeArray< int > AgentIndices

  Indices in the Obstacles array that are agents that need updating.
- NativeArray< NativeAvoidanceObstacleData > Obstacles

  All obstacles in the world that agents must consider.
- int ObstacleCount

  Number of valid obstacles in the Obstacles array.
- int MaxObstaclesConsidered

  The maximum number of obstacles each agent will consider.
- float DeltaTime

  How much time has passed since the last avoidance update.
- float TimeHorizon

  How far in the future to look when considering avoidance.
- NativeArray< float3 > AvoidanceVelocities

  The calculated avoidance velocity for each agent in AgentIndices.
- NativeArray< NativePlane > TempPlanes

  Used to store the obstacle planes for performing linear programming.
- NativeArray< NativePlane > TempProjPlanes

  Used to store a subset of obstacle planes for linear programming in four dimensions.

6.3.1 Detailed Description

Job that calculates the IAvoidanceAgent.AvoidanceVelocity of all IAvoidanceAgents.

6.3.2 Member Function Documentation

6.3.2.1 Execute() `void Infohazard.HyperNav.Jobs.AvoidanceJob.Execute ( int index )`

Run on a single agent index.

Parameters

index	The index in the AgentIndices array.

6.3.2.2 OrcaAvoidance()

```
float3 Infohazard.HyperNav.Jobs.AvoidanceJob.OrcaAvoidance (
        int agentIndex,
        ref NativeArray< NativePlane > agentTempPlanes,
        ref NativeArray< NativePlane > agentTempProjPlanes )
```

Perform avoidance calculation for an agent using the ORCA (optimal reciprocal collision avoidance) algorithm.

Adapted from the RVO2-3D library from University of North Carolina, which is licensed under Apache 2.0.

Parameters

agentIndex	Index of the agent in the Obstacles array.
agentTempPlanes	Array to store temp planes for linear programming.
agentTempProjPlanes	Array to store a subset of temp planes for 4D linear programming.

Returns

The calculated best velocity for the agent.

6.3.3 Member Data Documentation

6.3.3.1 AgentIndices

```
NativeArray<int> Infohazard.HyperNav.Jobs.AvoidanceJob.AgentIndices
```

Indices in the Obstacles array that are agents that need updating.

6.3.3.2 AvoidanceVelocities

```
NativeArray<float3> Infohazard.HyperNav.Jobs.AvoidanceJob.Avoidance↩
Velocities
```

The calculated avoidance velocity for each agent in AgentIndices.

6.3.3.3 DeltaTime

```
float Infohazard.HyperNav.Jobs.AvoidanceJob.DeltaTime
```

How much time has passed since the last avoidance update.

6.3.3.4 MaxObstaclesConsidered

```
int Infohazard.HyperNav.Jobs.AvoidanceJob.MaxObstaclesConsidered
```

The maximum number of obstacles each agent will consider.

6.3.3.5 ObstacleCount `int Infohazard.HyperNav.Jobs.AvoidanceJob.ObstacleCount`

Number of valid obstacles in the Obstacles array.

6.3.3.6 Obstacles `NativeArray<NativeAvoidanceObstacleData> Infohazard.HyperNav.Jobs.Avoidance↩Job.Obstacles`

All obstacles in the world that agents must consider.

6.3.3.7 TempPlanes `NativeArray<NativePlane> Infohazard.HyperNav.Jobs.AvoidanceJob.TempPlanes`

Used to store the obstacle planes for performing linear programming.

6.3.3.8 TempProjPlanes `NativeArray<NativePlane> Infohazard.HyperNav.Jobs.AvoidanceJob.Temp↩ProjPlanes`

Used to store a subset of obstacle planes for linear programming in four dimensions.

6.3.3.9 TimeHorizon `float Infohazard.HyperNav.Jobs.AvoidanceJob.TimeHorizon`

How far in the future to look when considering avoidance.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/AvoidanceJob.cs

6.4 Infohazard.HyperNav.AvoidanceManager Class Reference

Handles calculating the avoidance velocities for all IAvoidanceAgents.

Public Member Functions

- virtual void UpdateAvoidance (float deltaTime)

  Update the avoidance of all agents. Can be called manually if UpdateMode is set to manual.

Protected Member Functions

- virtual void OnEnable ()

  Schedule coroutine when the manager is enabled.
- void OnDisable ()

  Dispose data structures when disabled. Coroutine will stop automatically.

Properties

- AvoidanceManagerUpdateMode UpdateMode `[get, set]`

  When to update the avoidance velocities of agents.
- float TimeHorizon `[get, set]`

  How far in the future to look when avoiding collisions.
- int MaxObstaclesConsidered `[get, set]`

  The maximum number of obstacles that can be considered by each agent for avoidance.

6.4.1 Detailed Description

Handles calculating the avoidance velocities for all IAvoidanceAgents.

There should only be one AvoidanceManager active at a time, as it will handle all agents together. This is done using the C# Job System and Burst compiler to calculate avoidance very quickly. If it is still not fast enough, try changing the TimeHorizon and/or MaxObstaclesConsidered values.

6.4.2 Member Function Documentation

6.4.2.1 OnDisable() `void Infohazard.HyperNav.AvoidanceManager.OnDisable ( ) [protected]`

Dispose data structures when disabled. Coroutine will stop automatically.

6.4.2.2 OnEnable() `virtual void Infohazard.HyperNav.AvoidanceManager.OnEnable ( ) [protected], [virtual]`

Schedule coroutine when the manager is enabled.

6.4.2.3 UpdateAvoidance() `virtual void Infohazard.HyperNav.AvoidanceManager.UpdateAvoidance ( float deltaTime ) [virtual]`

Update the avoidance of all agents. Can be called manually if UpdateMode is set to manual.

Parameters

deltaTime	Time delta since last call.

6.4.3 Property Documentation

6.4.3.1 MaxObstaclesConsidered `int Infohazard.HyperNav.AvoidanceManager.MaxObstaclesConsidered [get], [set]`

The maximum number of obstacles that can be considered by each agent for avoidance.

This value caps the number of avoidance calculations per agent. Generally it should be equal to the max number of obstacles you expect to be within TimeHorizon $*$ IAvoidanceObstacle.MaxSpeed of an agent.

6.4.3.2 TimeHorizon `float Infohazard.HyperNav.AvoidanceManager.TimeHorizon [get], [set]`

How far in the future to look when avoiding collisions.

A lower value reduces the number of calculations per agent, with the drawback of being able to plan less far ahead.

6.4.3.3 UpdateMode `AvoidanceManagerUpdateMode Infohazard.HyperNav.AvoidanceManager.UpdateMode [get], [set]`

When to update the avoidance velocities of agents.

The documentation for this class was generated from the following file:

- Runtime/Avoidance/AvoidanceManager.cs

6.5 Infohazard.HyperNav.AvoidanceObstacleBase Class Reference

Base class for obstacles that are not expected to perform avoidance, but are avoided by agents.

Protected Member Functions

- virtual void OnEnable ()

  Resets desired velocity and adds self to list of all obstacles.
- virtual void OnDisable ()

  Removes self from list of all obstacles.

Properties

- virtual float MaxSpeed `[get, set]`

  Maximum speed the object can travel at.
- virtual float Radius `[get, set]`

  Radius of the object from its position.
- virtual Vector3 Position `[get]`

  World-space position of the object.
- virtual Vector3 InputVelocity `[get]`

  The object's desired (or actual) velocity.
- TagMask TagMask `[get, private set]`

  Tag of the object for matching agents' IAvoidanceAgent.AvoidedTags.

6.5.1 Detailed Description

Base class for obstacles that are not expected to perform avoidance, but are avoided by agents.

Objects that do perform avoidance (which are obstacles as well) should inherit from AvoidanceAgent instead, which is a derived class of AvoidanceObstacleBase.

6.5.2 Member Function Documentation

6.5.2.1 OnDisable() `virtual void Infohazard.HyperNav.AvoidanceObstacleBase.OnDisable ( ) [protected], [virtual]`

Removes self from list of all obstacles.

Reimplemented in Infohazard.HyperNav.AvoidanceAgent.

6.5.2.2 OnEnable() `virtual void Infohazard.HyperNav.AvoidanceObstacleBase.OnEnable ( ) [protected], [virtual]`

Resets desired velocity and adds self to list of all obstacles.

Reimplemented in Infohazard.HyperNav.AvoidanceAgent, and Infohazard.HyperNav.SimpleAvoidanceObstacle.

6.5.3 Property Documentation

6.5.3.1 InputVelocity `virtual Vector3 Infohazard.HyperNav.AvoidanceObstacleBase.InputVelocity [get]`

The object's desired (or actual) velocity.

Implements Infohazard.HyperNav.IAvoidanceObstacle.

6.5.3.2 MaxSpeed `virtual float Infohazard.HyperNav.AvoidanceObstacleBase.MaxSpeed [get], [set]`

Maximum speed the object can travel at.

Implements Infohazard.HyperNav.IAvoidanceObstacle.

6.5.3.3 Position `virtual Vector3 Infohazard.HyperNav.AvoidanceObstacleBase.Position [get]`

World-space position of the object.

Implements Infohazard.HyperNav.IAvoidanceObstacle.

6.5.3.4 Radius `virtual float Infohazard.HyperNav.AvoidanceObstacleBase.Radius [get], [set]`

Radius of the object from its position.

Implements Infohazard.HyperNav.IAvoidanceObstacle.

6.5.3.5 TagMask `TagMask Infohazard.HyperNav.AvoidanceObstacleBase.TagMask [get], [private set]`

Tag of the object for matching agents' IAvoidanceAgent.AvoidedTags.

Implements Infohazard.HyperNav.IAvoidanceObstacle.

The documentation for this class was generated from the following file:

- Runtime/Avoidance/AvoidanceObstacleBase.cs

6.6 Infohazard.HyperNav.Jobs.Baking.Surface.CalculateVertexAnglesJob Struct Reference

Used to calculate the offset from each vertex to check for collisions. For a flat or convex vertex, the offset is zero; however, for a concave vertex, the collision check must be offset in order to avoid the collision check hitting the ground.

6.6.1 Detailed Description

Used to calculate the offset from each vertex to check for collisions. For a flat or convex vertex, the offset is zero; however, for a concave vertex, the collision check must be offset in order to avoid the collision check hitting the ground.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Baking/Surface/CalculateVertexAnglesJob.cs

6.7 Infohazard.HyperNav.Edge Struct Reference

Represents the indices of an edge (two connected vertices) in an indexed mesh.

Public Member Functions

- Edge (int vertex1, int vertex2)

  Construct a new Edge with the given indices.
- Edge (int2 vertices)

  Construct a new Edge with the given indices.
- int2 ToInt2 ()

  Convert to int2.
- override bool Equals (object obj)

  Compare to another object.
- bool Equals (Edge other)

  Compare to another Edge.
- override int GetHashCode ()

  Get integer for use with hash table.

Static Public Member Functions

- static bool operator== (Edge a, Edge b)

  Equality operator.
- static bool operator!= (Edge a, Edge b)

  Inequality operator.

Static Public Attributes

- static readonly Edge InvalidEdge = new() { _minVertex = -1, _maxVertex = -1 }

  Invalid Edge.

Properties

- int Vertex1 `[get]`

  First vertex index, which is the lower of the two.
- int Vertex2 `[get]`

  Second vertex index, which is the higher of the two.
- bool IsValid `[get]`

  Whether the Edge is valid.

6.7.1 Detailed Description

Represents the indices of an edge (two connected vertices) in an indexed mesh.

The same Edge will be created regardless of the order in which indices are fed to the constructor.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 Edge() [1/2]

```
Infohazard.HyperNav.Edge.Edge (
        int vertex1,
        int vertex2 )
```

Construct a new Edge with the given indices.

The order of the indices doesn't matter; the same Edge is constructed either way. The indices cannot be the same.

Parameters

vertex1	First vertex index.
vertex2	Second vertex index.

6.7.2.2 Edge() [2/2] Infohazard.HyperNav.Edge.Edge (int2 *vertices*)

Construct a new Edge with the given indices.

Parameters

vertices	Vertex indices.

6.7.3 Member Function Documentation

6.7.3.1 Equals() [1/2] bool Infohazard.HyperNav.Edge.Equals (Edge *other*)

Compare to another Edge.

Parameters

other	Edge to compare to.

Returns

Whether the two edges are equal.

6.7.3.2 Equals() [2/2] override bool Infohazard.HyperNav.Edge.Equals (object *obj*)

Compare to another object.

Parameters

obj	Object to compare to.

Returns

Whether the two objects are equal.

6.7.3.3 GetHashCode() `override int Infohazard.HyperNav.Edge.GetHashCode ( )`

Get integer for use with hash table.

Returns

Integer hash code.

6.7.3.4 operator"!=() `static bool Infohazard.HyperNav.Edge.operator!= ( Edge a, Edge b ) [static]`

Inequality operator.

Parameters

a	First edge to compare.
b	Second edge to compare.

Returns

Whether the edges are not equal.

6.7.3.5 operator==() `static bool Infohazard.HyperNav.Edge.operator== ( Edge a, Edge b ) [static]`

Equality operator.

Parameters

a	First edge to compare.
b	Second edge to compare.

Returns

Whether the edges are equal.

6.7.3.6 ToInt2() `int2 Infohazard.HyperNav.Edge.ToInt2 ( )`

Convert to int2.

Returns

Edge as int2.

6.7.4 Member Data Documentation

6.7.4.1 InvalidEdge `readonly Edge Infohazard.HyperNav.Edge.InvalidEdge = new() { _minVertex = -1, _maxVertex = -1 } [static]`

Invalid Edge.

6.7.5 Property Documentation

6.7.5.1 IsValid `bool Infohazard.HyperNav.Edge.IsValid [get]`

Whether the Edge is valid.

6.7.5.2 Vertex1 `int Infohazard.HyperNav.Edge.Vertex1 [get]`

First vertex index, which is the lower of the two.

6.7.5.3 Vertex2 `int Infohazard.HyperNav.Edge.Vertex2 [get]`

Second vertex index, which is the higher of the two.

The documentation for this struct was generated from the following file:

- Runtime/Utility/Edge.cs

6.8 Infohazard.HyperNav.Editor.EditorNavAreaBakingUtility Class Reference

Manages multiple NavAreaBaseBakeHandlers for baking NavAreaBases.

Static Public Member Functions

- static void GetOrCreateData (NavAreaBase area)

  Get the NavVolumeData for a given volume, or create and save the object if it doesn't exist yet.
- static UniTaskCompletionSource BakeDataWithCompletionSource (NavAreaBase area, bool sanity↩ Checks=false, bool profile=false, UniTask? dependency=null)

  Bake the NavVolumeData for a given volume.
- static async UniTask BakeDataAsync (NavAreaBase area, bool sanityChecks=false, bool profile=false, Uni↩ Task? dependency=null)

  Bake the NavVolumeData for a given volume.
- static void CancelBake (NavAreaBase area)

  Cancel an actively baking volume and clear out its data.
- static bool TryGetBakeProgress (NavAreaBase area, out NavAreaBakeProgress progress)

  Get the bake progress of a volume.
- static void ClearData (NavAreaBase area)

  Clear out the baked data of a volume (does not destroy or un-assign the actual data object).

Static Public Attributes

- static readonly Dictionary< NavAreaBase, NavAreaBaseBakeHandler > BakeHandlers = new()

  The coroutine for each volume currently being baked.

Events

- static Action< NavAreaBase > BakeProgressUpdated

  Invoked when the bake progress for a NavVolume changes.

6.8.1 Detailed Description

Manages multiple NavAreaBaseBakeHandlers for baking NavAreaBases.

6.8.2 Member Function Documentation

6.8.2.1 BakeDataAsync()

```
static async UniTask Infohazard.HyperNav.Editor.EditorNavAreaBaking↩
Utility.BakeDataAsync (
            NavAreaBase area,
            bool sanityChecks = false,
            bool profile = false,
            UniTask?  dependency = null )  [static]
```

Bake the NavVolumeData for a given volume.

Parameters

area	The area to bake.
sanityChecks	If true, run sanity checks after each step.
profile	If true, run the profiler during baking.
dependency	Optional task to wait for before baking.

6.8.2.2 BakeDataWithCompletionSource()

```
static UniTaskCompletionSource Infohazard.HyperNav.Editor.EditorNavAreaBakingUtility.BakeDataWithCompletionSource (
            NavAreaBase area,
            bool sanityChecks = false,
            bool profile = false,
            UniTask? dependency = null )  [static]
```

Bake the NavVolumeData for a given volume.

Parameters

area	The area to bake.
sanityChecks	If true, run sanity checks after each step.
profile	If true, run the profiler during baking.
dependency	Optional task to wait for before baking.

Returns

A completion source that will be set when the baking is done.

6.8.2.3 CancelBake()

```
static void Infohazard.HyperNav.Editor.EditorNavAreaBakingUtility.CancelBake (
            NavAreaBase area )  [static]
```

Cancel an actively baking volume and clear out its data.

Parameters

area	The volume being baked.

6.8.2.4 ClearData()

```
static void Infohazard.HyperNav.Editor.EditorNavAreaBakingUtility.ClearData (
            NavAreaBase area )  [static]
```

Clear out the baked data of a volume (does not destroy or un-assign the actual data object).

Parameters

area	

6.8.2.5 GetOrCreateData()

```
static void Infohazard.HyperNav.Editor.EditorNavAreaBakingUtility.↩
GetOrCreateData (
            NavAreaBase area )  [static]
```

Get the NavVolumeData for a given volume, or create and save the object if it doesn't exist yet.

Parameters

area	The NavAreaBase component.

6.8.2.6 TryGetBakeProgress()

```
static bool Infohazard.HyperNav.Editor.EditorNavAreaBaking↩
Utility.TryGetBakeProgress (
            NavAreaBase area,
            out NavAreaBakeProgress progress )  [static]
```

Get the bake progress of a volume.

Parameters

area	The volume to check.
progress	The current bake progress for the volume.

Returns

If the volume is currently being baked.

6.8.3 Member Data Documentation

6.8.3.1 BakeHandlers

```
readonly Dictionary<NavAreaBase, NavAreaBaseBakeHandler> Infohazard.↩
HyperNav.Editor.EditorNavAreaBakingUtility.BakeHandlers = new()  [static]
```

The coroutine for each volume currently being baked.

6.8.4 Event Documentation

6.8.4.1 BakeProgressUpdated

```
Action<NavAreaBase> Infohazard.HyperNav.Editor.EditorNavArea↩
BakingUtility.BakeProgressUpdated  [static]
```

Invoked when the bake progress for a NavVolume changes.

The documentation for this class was generated from the following file:

- Editor/EditorNavAreaBakingUtility.cs

6.9 Infohazard.HyperNav.Editor.ExternalLinkEditorUtility Class Reference

Utilities for generating the external links of NavVolumes.

Static Public Member Functions

- static void ClearExternalLinks (INavArea volume)

  Clear the external links for a volume.
- static NavAreaBase[ ] GetAllLoadedAreas ()

  Get all loaded NavAreaBases that are relevant to link baking. If in prefab mode, only returns areas in the prefab.
- static void GenerateAllExternalLinks ()

  Generate the external links for all loaded INavAreas.
- static void GenerateExternalLinks (INavArea area)

  Generate the external links for a specific INavArea.
- static void GenerateExternalLinks (IReadOnlyList< INavArea > areas)

  Generate the external links for a list of INavAreas.

6.9.1 Detailed Description

Utilities for generating the external links of NavVolumes.

6.9.2 Member Function Documentation

6.9.2.1 ClearExternalLinks() `static void Infohazard.HyperNav.Editor.ExternalLinkEditorUtility.ClearExternalLinks ( INavArea volume ) [static]`

Clear the external links for a volume.

6.9.2.2 GenerateAllExternalLinks() `static void Infohazard.HyperNav.Editor.ExternalLinkEditorUtility.GenerateAllExternalLinks ( ) [static]`

Generate the external links for all loaded INavAreas.

6.9.2.3 GenerateExternalLinks() [1/2] `static void Infohazard.HyperNav.Editor.ExternalLinkEditorUtility.GenerateExternalLinks ( INavArea area ) [static]`

Generate the external links for a specific INavArea.

Parameters

area	

6.9.2.4 GenerateExternalLinks() [2/2] static void Infohazard.HyperNav.Editor.ExternalLink↩ EditorUtility.GenerateExternalLinks (IReadOnlyList< INavArea > *areas*) [static]

Generate the external links for a list of INavAreas.

Parameters

areas	

6.9.2.5 GetAllLoadedAreas() static NavAreaBase[] Infohazard.HyperNav.Editor.ExternalLink↩ EditorUtility.GetAllLoadedAreas ( ) [static]

Get all loaded NavAreaBases that are relevant to link baking. If in prefab mode, only returns areas in the prefab.

Returns

The loaded areas.

The documentation for this class was generated from the following file:

- Editor/ExternalLinkEditorUtility.cs

6.10 Infohazard.HyperNav.Jobs.Utility.Fast3DArray Struct Reference

A data structure equivalent to a three-dimensional int array (int[„]), but more efficient.

Public Member Functions

- Fast3DArray (int sizeX, int sizeY, int sizeZ, Allocator allocator=Allocator.Persistent)
  Construct a new Fast3DArray with the given dimensions.
- bool IsOneOf (int x, int y, int z, int option1, int option2)
  Return true if the element at [x, y, z] is either option1 or option2.
- bool Contains (int item)
  Return true if the given item is contained at any position in the array.

Public Attributes

- readonly int SizeX

  Size of first dimension.
- readonly int SizeY

  Size of second dimension.
- readonly int SizeZ

  Size of third dimension.

Properties

- int this[int x, int y, int z] `[get, set]`

  Get or set the value at given coordinates.
- int this[int index] `[get, set]`

  Get or set the value at the given index.

6.10.1 Detailed Description

A data structure equivalent to a three-dimensional int array (int[„]), but more efficient.

It does not perform bounds checks, and uses an unsafe pointer to the data.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 Fast3DArray()

```
Infohazard.HyperNav.Jobs.Utility.Fast3DArray.Fast3DArray (
        int sizeX,
        int sizeY,
        int sizeZ,
        Allocator allocator = Allocator.Persistent )
```

Construct a new Fast3DArray with the given dimensions.

Parameters

sizeX	Size of first dimension.
sizeY	Size of second dimension.
sizeZ	Size of third dimension.
allocator	Allocator to use for the array.

6.10.3 Member Function Documentation

6.10.3.1 Contains() `bool Infohazard.HyperNav.Jobs.Utility.Fast3DArray.Contains ( int item )`

Return true if the given item is contained at any position in the array.

6.10.3.2 IsOneOf() `bool Infohazard.HyperNav.Jobs.Utility.Fast3DArray.IsOneOf ( int x, int y, int z, int option1, int option2 )`

Return true if the element at [x, y, z] is either option1 or option2.

Parameters

x	First coordinate.
y	Second coordinate.
z	Third coordinate.
option1	First option to check equality.
option2	Second option to check equality.

Returns

If the value at the given coordinates is equal to either option1 or option2.

6.10.4 Member Data Documentation

6.10.4.1 SizeX `readonly int Infohazard.HyperNav.Jobs.Utility.Fast3DArray.SizeX`

Size of first dimension.

6.10.4.2 SizeY `readonly int Infohazard.HyperNav.Jobs.Utility.Fast3DArray.SizeY`

Size of second dimension.

6.10.4.3 SizeZ `readonly int Infohazard.HyperNav.Jobs.Utility.Fast3DArray.SizeZ`

Size of third dimension.

6.10.5 Property Documentation

6.10.5.1 this[int index] `int Infohazard.HyperNav.Jobs.Utility.Fast3DArray.this[int index] [get], [set]`

Get or set the value at the given index.

6.10.5.2 this[int x, int y, int z] `int Infohazard.HyperNav.Jobs.Utility.Fast3DArray.this[int x, int y, int z] [get], [set]`

Get or set the value at given coordinates.

Parameters

x	First coordinate.
y	Second coordinate.
z	Third coordinate.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Utility/Fast3DArray.cs

6.11 Infohazard.HyperNav.IAvoidanceAgent Interface Reference

Interface for objects that both can be avoided and themselves avoid other obstacles using the avoidance system.

Public Member Functions

- void UpdateAvoidanceVelocity (Vector3 newAvoidance)

  Called by the system to update AvoidanceVelocity.

Properties

- float AvoidanceWeight `[get]`

  How much effort the agent will take to avoid obstacles and other agents.
- float AvoidancePadding `[get]`

  How much extra space to leave when avoiding obstacles.
- Vector3 AvoidanceVelocity `[get]`

  The velocity the agent should have in order to avoid collisions with obstacles and other agents.
- bool IsActive `[get]`

  Whether the agent should actively avoid obstacles. If false, will still behave as an obstacle.
- bool DebugAvoidance `[get]`

  Whether to draw debugging information in the scene view.
- TagMask AvoidedTags `[get]`

  Tags of objects that the agent will try to avoid.

6.11.1 Detailed Description

Interface for objects that both can be avoided and themselves avoid other obstacles using the avoidance system.

6.11.2 Member Function Documentation

6.11.2.1 UpdateAvoidanceVelocity() `void Infohazard.HyperNav.IAvoidanceAgent.UpdateAvoidanceVelocity ( Vector3 newAvoidance )`

Called by the system to update AvoidanceVelocity.

Parameters

newAvoidance	New avoidance velocity

Implemented in Infohazard.HyperNav.AvoidanceAgent.

6.11.3 Property Documentation

6.11.3.1 AvoidancePadding `float Infohazard.HyperNav.IAvoidanceAgent.AvoidancePadding [get]`

How much extra space to leave when avoiding obstacles.

Implemented in Infohazard.HyperNav.AvoidanceAgent.

6.11.3.2 AvoidanceVelocity `Vector3 Infohazard.HyperNav.IAvoidanceAgent.AvoidanceVelocity [get]`

The velocity the agent should have in order to avoid collisions with obstacles and other agents.

Implemented in Infohazard.HyperNav.AvoidanceAgent.

6.11.3.3 AvoidanceWeight `float Infohazard.HyperNav.IAvoidanceAgent.AvoidanceWeight [get]`

How much effort the agent will take to avoid obstacles and other agents.

Implemented in Infohazard.HyperNav.AvoidanceAgent.

6.11.3.4 AvoidedTags `TagMask Infohazard.HyperNav.IAvoidanceAgent.AvoidedTags [get]`

Tags of objects that the agent will try to avoid.

Implemented in Infohazard.HyperNav.AvoidanceAgent.

6.11.3.5 DebugAvoidance `bool Infohazard.HyperNav.IAvoidanceAgent.DebugAvoidance [get]`

Whether to draw debugging information in the scene view.

Implemented in Infohazard.HyperNav.AvoidanceAgent.

6.11.3.6 IsActive `bool Infohazard.HyperNav.IAvoidanceAgent.IsActive [get]`

Whether the agent should actively avoid obstacles. If false, will still behave as an obstacle.

Implemented in Infohazard.HyperNav.AvoidanceAgent.

The documentation for this interface was generated from the following file:

- Runtime/Avoidance/IAvoidanceAgent.cs

6.12 Infohazard.HyperNav.IAvoidanceObstacle Interface Reference

Interface for objects that can be avoided using the avoidance system.

Properties

- Vector3 Position `[get]`
  World-space position of the object.
- Vector3 InputVelocity `[get]`
  The object's desired (or actual) velocity.
- float MaxSpeed `[get]`
  Maximum speed the object can travel at.
- float Radius `[get]`
  Radius of the object from its position.
- TagMask TagMask `[get]`
  Tag of the object for matching agents' IAvoidanceAgent.AvoidedTags.

6.12.1 Detailed Description

Interface for objects that can be avoided using the avoidance system.

6.12.2 Property Documentation

6.12.2.1 InputVelocity `Vector3 Infohazard.HyperNav.IAvoidanceObstacle.InputVelocity [get]`

The object's desired (or actual) velocity.

Implemented in Infohazard.HyperNav.AvoidanceAgent, Infohazard.HyperNav.AvoidanceObstacleBase, Infohazard.HyperNav.Rigidbod
and Infohazard.HyperNav.SimpleAvoidanceObstacle.

6.12.2.2 MaxSpeed `float Infohazard.HyperNav.IAvoidanceObstacle.MaxSpeed [get]`

Maximum speed the object can travel at.

Implemented in Infohazard.HyperNav.AvoidanceObstacleBase.

6.12.2.3 Position `Vector3 Infohazard.HyperNav.IAvoidanceObstacle.Position [get]`

World-space position of the object.

Implemented in Infohazard.HyperNav.AvoidanceObstacleBase.

6.12.2.4 Radius `float Infohazard.HyperNav.IAvoidanceObstacle.Radius [get]`

Radius of the object from its position.

Implemented in Infohazard.HyperNav.AvoidanceObstacleBase.

6.12.2.5 TagMask `TagMask Infohazard.HyperNav.IAvoidanceObstacle.TagMask [get]`

Tag of the object for matching agents' IAvoidanceAgent.AvoidedTags.

Implemented in Infohazard.HyperNav.AvoidanceObstacleBase.

The documentation for this interface was generated from the following file:

- Runtime/Avoidance/IAvoidanceObstacle.cs

6.13 Infohazard.HyperNav.Jobs.Utility.IntList2 Struct Reference

Serves as a list of ints with a max size of 2, to avoid allocations.

Public Member Functions

- void Add (int value)

  Add a value to the list.
- bool Contains (int value)

  Check if the list contains the given value.

Static Public Attributes

- const int MaxCount = 2

  The maximum number of elements the list can hold.

Properties

- int Count `[get, private set]`

  The number of elements in the list.
- int this[int index] `[get, set]`

  Get or set the value at the given index.

6.13.1 Detailed Description

Serves as a list of ints with a max size of 2, to avoid allocations.

Used to store which regions each vertex is part of, and which other vertices it's connected to.

6.13.2 Member Function Documentation

6.13.2.1 Add() `void Infohazard.HyperNav.Jobs.Utility.IntList2.Add ( int value )`

Add a value to the list.

Parameters

value	The value to add.

Exceptions

IndexOutOfRangeException	If the list is full.

6.13.2.2 Contains() `bool Infohazard.HyperNav.Jobs.Utility.IntList2.Contains ( int value )`

Check if the list contains the given value.

Parameters

value	The value to check.

Returns

Whether the value was found.

6.13.3 Member Data Documentation

6.13.3.1 MaxCount `const int Infohazard.HyperNav.Jobs.Utility.IntList2.MaxCount = 2 [static]`

The maximum number of elements the list can hold.

6.13.4 Property Documentation

6.13.4.1 Count `int Infohazard.HyperNav.Jobs.Utility.IntList2.Count [get], [private set]`

The number of elements in the list.

6.13.4.2 this[int index] `int Infohazard.HyperNav.Jobs.Utility.IntList2.this[int index] [get], [set]`

Get or set the value at the given index.

Parameters

index	Index to get or set.

Exceptions

IndexOutOfRangeException	If the index is out of range.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Utility/IntList2.cs

6.14 Infohazard.HyperNav.Jobs.Baking.InvertIndexArrayJob Struct Reference

Given an array of indices, creates an array where the value at each index is the index of that value in the input array. OutputArray is expected to be at least as large as the maximum value in InputArray plus one.

6.14.1 Detailed Description

Given an array of indices, creates an array where the value at each index is the index of that value in the input array. OutputArray is expected to be at least as large as the maximum value in InputArray plus one.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Baking/InvertIndexArrayJob.cs

6.15 Infohazard.HyperNav.ISurfaceUprightDirectionHandler Interface Reference

Interface for custom logic to determine the upright direction of a surface.

Public Member Functions

- UniTask< NavSurfaceBakeData > CalculateUprightDirections (NavSurfaceBakeData bakeData, NativeArray< RaycastCommand > queries, int resultsPerQuery, NativeArray< RaycastHit > results, NativeCancellationToken cancellationToken)

  Calculate the upright directions for each triangle of the surface, and store the results in NavSurfaceBakeData.NavSurfaceBakeData.MeshInfo.BakingNavSurfaceMeshInfo.TriangleUprightWorldDirections.

6.15.1 Detailed Description

Interface for custom logic to determine the upright direction of a surface.

6.15.2 Member Function Documentation

6.15.2.1 CalculateUprightDirections()

```
UniTask< NavSurfaceBakeData > Infohazard.HyperNav.ISurfaceUprightDirectionHandler.CalculateUprightDirections (
            NavSurfaceBakeData bakeData,
            NativeArray< RaycastCommand > queries,
            int resultsPerQuery,
            NativeArray< RaycastHit > results,
            NativeCancellationToken cancellationToken )
```

Calculate the upright directions for each triangle of the surface, and store the results in NavSurfaceBakeData.NavSurfaceBakeData.MeshInfo.BakingNavSurfaceMeshInfo.TriangleUprightWorldDirections.

This should be done using a parallel job if possible. Any triangles that fail custom checks should be added to NavSurfaceBakeData.NavSurfaceBakeData.FilterData's NavSurfaceFilterData.TrianglesToSplit or NavSurfaceFilterData.TrianglesToRemove.

Parameters

bakeData	The in-progress data for the surface.
cancellationToken	Cancellation token for the operation.
queries	The raycast queries that were performed.
resultsPerQuery	How many hits in the results array correspond to each query.
results	The results of the raycast queries.

Returns

The updated bake data (can be the same as the input if only referenced values are updated).

Implemented in Infohazard.HyperNav.SurfaceUprightDirectionHandlerWithOverrides.

The documentation for this interface was generated from the following file:

- Runtime/NavSurface.cs

6.16 Infohazard.HyperNav.Jobs.Utility.MarchingCubesCavityTables Class Reference

Tables for determining which cubes and combinations of cubes from the Marching Cubes algorithm will produce concave results.

Static Public Attributes

- static readonly CubesWithInternalCavityTable CubesWithInternalCavityTable

  This table maps a cube index (the same index used in MarchingCubesTables.TriTable to a boolean, which indicates whether that cube contains internal concavities.
- static readonly CubeConcaveNeighborsTable CubeConcaveNeighborsTable

  This table maps a cube index to another table, which maps a direction index to a list of other cube indices, which, when they are adjacent in that direction, will create a concavity.

6.16.1 Detailed Description

Tables for determining which cubes and combinations of cubes from the Marching Cubes algorithm will produce concave results.

This file is auto-generated but should never change.

6.16.2 Member Data Documentation

6.16.2.1 CubeConcaveNeighborsTable `readonly CubeConcaveNeighborsTable Infohazard.HyperNav.↩ Jobs.Utility.MarchingCubesCavityTables.CubeConcaveNeighborsTable [static]`

This table maps a cube index to another table, which maps a direction index to a list of other cube indices, which, when they are adjacent in that direction, will create a concavity.

index1 = cube index
index2 = direction index (use directions array)
index3 = other cube index that is concave when in that direction

6.16.2.2 CubesWithInternalCavityTable `readonly CubesWithInternalCavityTable Infohazard.Hyper↩ Nav.Jobs.Utility.MarchingCubesCavityTables.CubesWithInternalCavityTable [static]`

This table maps a cube index (the same index used in MarchingCubesTables.TriTable to a boolean, which indicates whether that cube contains internal concavities.

If a cube contains internal cavities, it will always be concave no matter what cubes it is adjacent to.

The documentation for this class was generated from the following file:

- Runtime/Jobs/Utility/MarchingCubesCavityTables.cs

6.17 Infohazard.HyperNav.Jobs.Utility.MarchingCubesTables Class Reference

Tables used in the Marching Cubes algorithm.

Static Public Member Functions

- static int3 GetPositiveDirection (int index)

  Easy way to loop through positive directions in the order X, Y, Z.

Static Public Attributes

- static readonly TriTable TriTable

  This table maps one of 256 8-bit cube IDs to a list of triangles.
- static readonly EdgeToVertexIndexTable EdgeToVertexIndexTable

  This table maps one of 12 edge indices to the indices of the two vertices that it connects. The second vertex is always greater than the first on one axis.
- static readonly VertexIndexToPositionTable VertexIndexToPositionTable

  This table maps one of 8 vertex indices to a local position in a cube.
- static readonly AcrossCenterEdges AcrossCenterEdges

  A list of the edge indices that cross the center of the cube.
- static readonly DirectionToVerticesOnSideTable DirectionToVerticesOnSideA

  This table maps the index of a direction in GetPositiveDirection to a list of vertices on the positive side of that direction.
- static readonly DirectionToVerticesOnSideTable DirectionToVerticesOnSideB

  This table maps the index of a direction in GetPositiveDirection to a list of vertices on the negative side of that direction.

6.17.1 Detailed Description

Tables used in the Marching Cubes algorithm.

6.17.2 Member Function Documentation

6.17.2.1 GetPositiveDirection() `static int3 Infohazard.HyperNav.Jobs.Utility.MarchingCubes↩ Tables.GetPositiveDirection (`

```
            int index )  [static]
```

Easy way to loop through positive directions in the order X, Y, Z.

6.17.3 Member Data Documentation

6.17.3.1 AcrossCenterEdges `readonly AcrossCenterEdges Infohazard.HyperNav.Jobs.Utility.↩ MarchingCubesTables.AcrossCenterEdges  [static]`

Initial value:

```
= new AcrossCenterEdges {
            [0] = 6,
            [1] = 7,
            [2] = 4,
            [3] = 5,
            [4] = 2,
            [5] = 3,
            [6] = 0,
            [7] = 1,
            [8] = 10,
            [9] = 11,
            [10] = 8,
            [11] = 9,
        }
```

A list of the edge indices that cross the center of the cube.

6.17.3.2 DirectionToVerticesOnSideA `readonly DirectionToVerticesOnSideTable Infohazard.Hyper↩ Nav.Jobs.Utility.MarchingCubesTables.DirectionToVerticesOnSideA  [static]`

Initial value:

```
=
            new DirectionToVerticesOnSideTable {
                [0] = new(0, 3, 7, 4),
                [1] = new(0, 1, 2, 3),
                [2] = new(0, 1, 5, 4),
            }
```

This table maps the index of a direction in GetPositiveDirection to a list of vertices on the positive side of that direction.

6.17.3.3 DirectionToVerticesOnSideB

`readonly DirectionToVerticesOnSideTable Infohazard.HyperNav.Jobs.Utility.MarchingCubesTables.DirectionToVerticesOnSideB [static]`

Initial value:

```
=
            new DirectionToVerticesOnSideTable {
                [0] = new(1, 2, 6, 5),
                [1] = new(4, 5, 6, 7),
                [2] = new(2, 3, 7, 6),
            }
```

This table maps the index of a direction in GetPositiveDirection to a list of vertices on the negative side of that direction.

6.17.3.4 EdgeToVertexIndexTable

`readonly EdgeToVertexIndexTable Infohazard.HyperNav.Jobs.Utility.MarchingCubesTables.EdgeToVertexIndexTable [static]`

Initial value:

```
= new EdgeToVertexIndexTable {
            [0] = new(0, 1),
            [1] = new(1, 2),
            [2] = new(3, 2),
            [3] = new(0, 3),
            [4] = new(4, 5),
            [5] = new(5, 6),
            [6] = new(7, 6),
            [7] = new(4, 7),
            [8] = new(0, 4),
            [9] = new(1, 5),
            [10] = new(2, 6),
            [11] = new(3, 7),
        }
```

This table maps one of 12 edge indices to the indices of the two vertices that it connects. The second vertex is always greater than the first on one axis.

6.17.3.5 TriTable

`readonly TriTable Infohazard.HyperNav.Jobs.Utility.MarchingCubesTables.TriTable [static]`

This table maps one of 256 8-bit cube IDs to a list of triangles.

Each bit of the cube ID represents one of 8 corners, which is either on or off. The order of these corners is that of the VertexIndexToPositionTable array. The output byte[] is a list of edge indices (each three edge indices makes a triangle). Each edge index is an index in EdgeToVertexIndexTable, giving two vertices to take the midpoint of in order to get one point of a triangle.

6.17.3.6 VertexIndexToPositionTable

`readonly VertexIndexToPositionTable Infohazard.HyperNav.Jobs.Utility.MarchingCubesTables.VertexIndexToPositionTable [static]`

Initial value:

```
= new VertexIndexToPositionTable {
            [0] = new(0, 0, 0),
            [1] = new(1, 0, 0),
            [2] = new(1, 0, 1),
            [3] = new(0, 0, 1),
            [4] = new(0, 1, 0),
            [5] = new(1, 1, 0),
            [6] = new(1, 1, 1),
            [7] = new(0, 1, 1),
        }
```

This table maps one of 8 vertex indices to a local position in a cube.

The documentation for this class was generated from the following file:

- Runtime/Jobs/Utility/MarchingCubesTables.cs

6.18 Infohazard.HyperNav.Jobs.NativeAvoidanceObstacleData Struct Reference

Represents one obstacle (which may be an agent) in the avoidance system.

Public Attributes

- float3 Position

  Position of the obstacle.
- float3 InputVelocity

  Velocity of the obstacle (or desired velocity if it is an agent).
- float Radius

  Radius of the obstacle.
- float Padding

  If an agent, extra padding to give when avoiding obstacles.
- float Avoidance

  If an agent, its contribution weight to avoidance. If not an agent, zero.
- float Speed

  The maximum speed that this obstacle can move at.
- long TagMask

  The tag mask of this obstacle.
- long AvoidedTags

  If an agent, the tag masks this agent will avoid.
- bool Debug

  If an agent, whether to draw debug lines when calculating avoidance.

6.18.1 Detailed Description

Represents one obstacle (which may be an agent) in the avoidance system.

6.18.2 Member Data Documentation

6.18.2.1 Avoidance `float Infohazard.HyperNav.Jobs.NativeAvoidanceObstacleData.Avoidance`

If an agent, its contribution weight to avoidance. If not an agent, zero.

6.18.2.2 AvoidedTags `long Infohazard.HyperNav.Jobs.NativeAvoidanceObstacleData.AvoidedTags`

If an agent, the tag masks this agent will avoid.

6.18.2.3 Debug `bool Infohazard.HyperNav.Jobs.NativeAvoidanceObstacleData.Debug`

If an agent, whether to draw debug lines when calculating avoidance.

6.18.2.4 InputVelocity `float3 Infohazard.HyperNav.Jobs.NativeAvoidanceObstacleData.Input↩ Velocity`

Velocity of the obstacle (or desired velocity if it is an agent).

6.18.2.5 Padding `float Infohazard.HyperNav.Jobs.NativeAvoidanceObstacleData.Padding`

If an agent, extra padding to give when avoiding obstacles.

6.18.2.6 Position `float3 Infohazard.HyperNav.Jobs.NativeAvoidanceObstacleData.Position`

Position of the obstacle.

6.18.2.7 Radius `float Infohazard.HyperNav.Jobs.NativeAvoidanceObstacleData.Radius`

Radius of the obstacle.

6.18.2.8 Speed `float Infohazard.HyperNav.Jobs.NativeAvoidanceObstacleData.Speed`

The maximum speed that this obstacle can move at.

6.18.2.9 TagMask `long Infohazard.HyperNav.Jobs.NativeAvoidanceObstacleData.TagMask`

The tag mask of this obstacle.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/NativeAvoidanceData.cs

6.19 Infohazard.HyperNav.Jobs.Utility.NativeBounds Struct Reference

A native-friendly of a bounding box.

Public Member Functions

- NativeBounds (float4 center, float4 extents)

  Initialize a new NativeBounds with the given data.

Public Attributes

- float4 Center

  Center of the bounds.
- float4 Extents

  Extents of the bounds (half of its size).

6.19.1 Detailed Description

A native-friendly of a bounding box.

6.19.2 Constructor & Destructor Documentation

6.19.2.1 NativeBounds() `Infohazard.HyperNav.Jobs.Utility.NativeBounds.NativeBounds (`

```
float4 center,
float4 extents )
```

Initialize a new NativeBounds with the given data.

Parameters

center	Center of the bounds.
extents	Extents of the bounds (half of its size).

6.19.3 Member Data Documentation

6.19.3.1 Center `float4 Infohazard.HyperNav.Jobs.Utility.NativeBounds.Center`

Center of the bounds.

6.19.3.2 Extents `float4 Infohazard.HyperNav.Jobs.Utility.NativeBounds.Extents`

Extents of the bounds (half of its size).

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Utility/NavVolumeNativeData.cs

6.20 Infohazard.HyperNav.Jobs.Utility.NativeMathUtility Class Reference

Provides math operations that are compatible with Burst.

Static Public Member Functions

- static float4 ProjectOnPlane (in float4 vector, in float3 normal)

  Project a vector onto a the plane defined by a normal.
- static bool GetNearestPointOnSegment (in float4 v1, in float4 v2, in float4 point, out float4 pointOnSegment)

  Find the point on a bounded line segment where it is nearest to a position, and return whether that point is in the segment's bounds.
- static bool GetNearestPointOnTriangle (in float4 v1, in float4 v2, in float4 v3, in float4 point, out float4 point↩ OnTriangle)

  Find the point on a triangle where it is nearest to a position, and return whether that point is in the triangle's bounds.
- static float4 GetNearestPointOnTriangleIncludingBounds (in float4 v1, in float4 v2, in float4 v3, in float4 point)

  Find the point on a triangle (including its bounds) where it is nearest to a position.
- static bool IsPointInsideBound (in float4 v1, in float4 v2, in float3 normal, in float4 point)

  Returns true if a given point is on the inner side (defined by a given normal) of a segment.
- static bool DoesSegmentIntersectTriangle (in float4 v1, in float4 v2, in float4 v3, in float4 s1, in float4 s2, out float t)

  Raycast a line segment against a triangle, and return whether they intersect.
- static unsafe bool NavVolumeRaycast (float4 start, float4 end, bool earlyReturn, in NativeNavVolumeData volume, out float t)

  Cast a ray against the blocking triangles of the volume, and return the nearest hit.
- static bool NavSurfacePathCheck (int region1, int region2, float4 pos1, float4 pos2, float angleLimitDot, in NativeNavSurfaceData surface)

  Check a path through the surface to see if the line is passable. This is significantly more complex than a volume raycast, because it must consider the surface's walkable geometry rather than just blocking triangles.
- static float4 GetPerpendicularVector (float4 vector)

  Returns an arbitrary vector that is perpendicular to the given vector.
- static bool IsOutOfBounds (in int3 boundSize, in int3 n)

  Simple bounds check for a 3D coordinate.
- static int GetDirectionIndex (float3 normal, float epsilon=0.001f)

  Get the direction index for the given plane normal. This returns a value in the range [0, 26]. It assumes the only directions are the six cardinal directions, and the diagonals combining two or three of those directions.

6.20.1 Detailed Description

Provides math operations that are compatible with Burst.

Managed side versions are available in the Infohazard.Core library under MathUtility.

6.20.2 Member Function Documentation

6.20.2.1 DoesSegmentIntersectTriangle() `static bool Infohazard.HyperNav.Jobs.Utility.Native↩ MathUtility.DoesSegmentIntersectTriangle (`

```
            in float4 v1,
            in float4 v2,
            in float4 v3,
            in float4 s1,
            in float4 s2,
            out float t )  [static]
```

Raycast a line segment against a triangle, and return whether they intersect.

Parameters

v1	The first triangle point.
v2	The second triangle point.
v3	The third triangle point.
s1	The start of the segment.
s2	The end of the segment.
t	The point along the input segment where it intersects the triangle, or -1.

Returns

Whether the segment intersects the triangle.

6.20.2.2 GetDirectionIndex()

```
static int Infohazard.HyperNav.Jobs.Utility.NativeMathUtility.GetDirectionIndex (
            float3 normal,
            float epsilon = 0.001f )  [static]
```

Get the direction index for the given plane normal. This returns a value in the range [0, 26]. It assumes the only directions are the six cardinal directions, and the diagonals combining two or three of those directions.

Parameters

normal	Plane normal, which does not need to be normalized.
epsilon	Epsilon for comparing normal components.

Returns

Direction index for given normal.

6.20.2.3 GetNearestPointOnSegment()

```
static bool Infohazard.HyperNav.Jobs.Utility.NativeMathUtility.GetNearestPointOnSegment (
            in float4 v1,
            in float4 v2,
            in float4 point,
            out float4 pointOnSegment )  [static]
```

Find the point on a bounded line segment where it is nearest to a position, and return whether that point is in the segment's bounds.

Does not return points on the ends of the segment. If the nearest point on the segment's line is outside the segment, will fail and not return a valid point.

Parameters

v1	The start of the segment.
v2	The end of the segment.
point	The point to search for.
pointOnSegment	The point on the segment closest to the input point.

Returns

Whether the nearest point is within the segment's bounds.

6.20.2.4 GetNearestPointOnTriangle()

```
static bool Infohazard.HyperNav.Jobs.Utility.NativeMathUtility.GetNearestPointOnTriangle (
            in float4 v1,
            in float4 v2,
            in float4 v3,
            in float4 point,
            out float4 pointOnTriangle )  [static]
```

Find the point on a triangle where it is nearest to a position, and return whether that point is in the triangle's bounds.

Does not return points on the edge of the triangle. If the nearest point on the triangle's plane is outside the triangle, will fail and not return a valid point.

Parameters

v1	The first triangle point.
v2	The second triangle point.
v3	The third triangle point.
point	The point to search for.
pointOnTriangle	The point on the triangle closest to the input point.

Returns

Whether the nearest point is within the triangle's bounds.

6.20.2.5 GetNearestPointOnTriangleIncludingBounds()

```
static float4 Infohazard.HyperNav.Jobs.Utility.NativeMathUtility.GetNearestPointOnTriangleIncludingBounds (
            in float4 v1,
            in float4 v2,
            in float4 v3,
            in float4 point )  [static]
```

Find the point on a triangle (including its bounds) where it is nearest to a position.

If nearest point is on the triangle's bounds, that point will be returned, unlike GetNearestPointOnTriangle.

Parameters

v1	The first triangle point.
v2	The second triangle point.
v3	The third triangle point.
point	The point to search for.

Returns

The nearest point on the triangle to the given point.

6.20.2.6 GetPerpendicularVector()

```
static float4 Infohazard.HyperNav.Jobs.Utility.NativeMathUtility.GetPerpendicularVector (
            float4 vector ) [static]
```

Returns an arbitrary vector that is perpendicular to the given vector.

Parameters

vector	Input vector.

Returns

A perpendicular vector.

6.20.2.7 IsOutOfBounds()

```
static bool Infohazard.HyperNav.Jobs.Utility.NativeMathUtility.IsOutOfBounds (
            in int3 boundSize,
            in int3 n ) [static]
```

Simple bounds check for a 3D coordinate.

Parameters

boundSize	The size of the bounds.
n	The coordinate to check.

Returns

Whether the coordinate is outside the bounds (equal to size is considered outside).

6.20.2.8 IsPointInsideBound()

```
static bool Infohazard.HyperNav.Jobs.Utility.NativeMathUtility.IsPointInsideBound (
            in float4 v1,
            in float4 v2,
            in float3 normal,
            in float4 point ) [static]
```

Returns true if a given point is on the inner side (defined by a given normal) of a segment.

Parameters

v1	The start of the segment.
v2	The end of the segment.
normal	The normal, defining which side is inside.
point	The point to search for.

Returns

Whether the point is on the inner side.

6.20.2.9 NavSurfacePathCheck()

```
static bool Infohazard.HyperNav.Jobs.Utility.NativeMathUtility.NavSurfacePathCheck (
            int region1,
            int region2,
            float4 pos1,
            float4 pos2,
            float angleLimitDot,
            in NativeNavSurfaceData surface )  [static]
```

Check a path through the surface to see if the line is passable. This is significantly more complex than a volume raycast, because it must consider the surface's walkable geometry rather than just blocking triangles.

Parameters

region1	The start region of the path.
region2	The end region of the path.
pos1	The start position of the path in local space of the surface.
pos2	The end position of the path in local space of the surface.
angleLimitDot	All regions along the path must be within this angle of the start.
surface	The surface to check.

Returns

True if the path is directly passable, otherwise false.

6.20.2.10 NavVolumeRaycast()

```
static unsafe bool Infohazard.HyperNav.Jobs.Utility.NativeMathUtility.NavVolumeRaycast (
            float4 start,
            float4 end,
            bool earlyReturn,
            in NativeNavVolumeData volume,
            out float t )  [static]
```

Cast a ray against the blocking triangles of the volume, and return the nearest hit.

Parameters

start	The position to start the query at in local space of the volume.
end	The position to end the query at in local space of the volume.
earlyReturn	If true, will return true as soon as any triangle is hit, not necessarily giving you the closest hit point.
volume	The volume in which to raycast.
t	If the query hits a triangle, the ratio between start and end at which the hit occurred.

Returns

Whether a triangle was hit.

6.20.2.11 ProjectOnPlane()

```
static float4 Infohazard.HyperNav.Jobs.Utility.NativeMathUtility.↩
ProjectOnPlane (
            in float4 vector,
            in float3 normal )  [static]
```

Project a vector onto a the plane defined by a normal.

Parameters

vector	The vector to project.
normal	The normal of the plane.

Returns

The projected vector.

The documentation for this class was generated from the following file:

- Runtime/Jobs/Utility/NativeMathUtility.cs

6.21 Infohazard.HyperNav.Jobs.Utility.NativeNavExternalLinkData Struct Reference

The native-friendly data representing a connection from one region to another region in another volume.

Public Member Functions

- NativeNavExternalLinkData (long toArea, NavAreaTypes toAreaType, int toRegion, float4 fromPosition, float4 toPosition, long manualLinkID)

  Initialize a new NativeNavExternalLinkData with the given data.

Public Attributes

- readonly long ToArea

  The ID of the connected area.
- readonly NavAreaTypes ToAreaType

  The type of the connected area.
- readonly int ToRegion

  The ID of the connected region.
- readonly float4 FromPosition

  The position at which the connection originates (local space of the 'from' volume).
- readonly float4 ToPosition

  The position at which the connection ends (local space of the 'from' volume).
- readonly float InternalCost

  The distance from FromPosition to ToPosition.
- readonly long ManualLinkID

  The unique ID of the ManualNavLink that created this link, or 0.

6.21.1 Detailed Description

The native-friendly data representing a connection from one region to another region in another volume.

6.21.2 Constructor & Destructor Documentation

6.21.2.1 NativeNavExternalLinkData() `Infohazard.HyperNav.Jobs.Utility.NativeNavExternalLink↩Data.NativeNavExternalLinkData (`

```
            long toArea,
            NavAreaTypes toAreaType,
            int toRegion,
            float4 fromPosition,
            float4 toPosition,
            long manualLinkID )
```

Initialize a new NativeNavExternalLinkData with the given data.

Parameters

toArea	The ID of the connected volume.
toAreaType	The type of the connected volume.
toRegion	The ID of the connected region.
fromPosition	The position at which the connection originates.
toPosition	The position at which the connection ends.
manualLinkID	The unique ID of the ManualNavLink that created this link, or 0.

6.21.3 Member Data Documentation

6.21.3.1 FromPosition

`readonly float4 Infohazard.HyperNav.Jobs.Utility.NativeNavExternalLinkData.FromPosition`

The position at which the connection originates (local space of the 'from' volume).

6.21.3.2 InternalCost

`readonly float Infohazard.HyperNav.Jobs.Utility.NativeNavExternalLinkData.InternalCost`

The distance from FromPosition to ToPosition.

6.21.3.3 ManualLinkID

`readonly long Infohazard.HyperNav.Jobs.Utility.NativeNavExternalLinkData.ManualLinkID`

The unique ID of the ManualNavLink that created this link, or 0.

6.21.3.4 ToArea

`readonly long Infohazard.HyperNav.Jobs.Utility.NativeNavExternalLinkData.ToArea`

The ID of the connected area.

6.21.3.5 ToAreaType

`readonly NavAreaTypes Infohazard.HyperNav.Jobs.Utility.NativeNavExternalLinkData.ToAreaType`

The type of the connected area.

6.21.3.6 ToPosition

`readonly float4 Infohazard.HyperNav.Jobs.Utility.NativeNavExternalLinkData.ToPosition`

The position at which the connection ends (local space of the 'from' volume).

6.21.3.7 ToRegion

`readonly int Infohazard.HyperNav.Jobs.Utility.NativeNavExternalLinkData.ToRegion`

The ID of the connected region.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Utility/NavVolumeNativeData.cs

6.22 Infohazard.HyperNav.NativeNavSurfaceDataPointers Struct Reference

References to the NativeArrays allocated for a NativeNavSurfaceData.

Public Member Functions

- void Dispose ()

  Dispose all of the native array references.

6.22.1 Detailed Description

References to the NativeArrays allocated for a NativeNavSurfaceData.

In the NativeNavSurfaceData itself, these arrays are kept as pointers, which cannot be used to deallocate the arrays under Unity's safe memory system. In order to play nicely with that system, the original references must be kept and disposed.

6.22.2 Member Function Documentation

6.22.2.1 Dispose() `void Infohazard.HyperNav.NativeNavSurfaceDataPointers.Dispose ( )`

Dispose all of the native array references.

The documentation for this struct was generated from the following file:

- Runtime/NavSurfaceData.cs

6.23 Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData Struct Reference

The baked data of a NavVolume, converted to a form compatible with Burst.

Public Member Functions

- NativeNavVolumeData (long id, float4x4 transform, float4x4 inverseTransform, NativeBounds bounds, NavLayer layer, UnsafeArray< float4 > vertices, UnsafeArray< NativeNavVolumeRegionData > regions, UnsafeArray< int > triangleIndices, int blockingTriangleIndexCount, UnsafeArray< NativePlane > boundPlanes, UnsafeArray< NativeNavVolumeInternalLinkData > internalLinks, UnsafeArray< NativeNavExternalLinkData > externalLinks, UnsafeArray< int > linkVertices, UnsafeArray< int2 > linkEdges, UnsafeArray< int3 > linkTriangles)

  Initialize a new NativeNavVolumeData with the given data.

Public Attributes

- readonly long ID

  ID of the volume.
- readonly float4x4 Transform

  Transform matrix of the volume.
- readonly float4x4 InverseTransform

  Inverse transform matrix of the volume.
- readonly NativeBounds Bounds

  Bounds of the volume in local space.
- readonly NavLayer Layer

  The layer this area exists on.
- readonly UnsafeArray< float4 > Vertices

  The vertex positions of all of the volume's regions, in local space.
- readonly UnsafeArray< NativeNavVolumeRegionData > Regions

  The regions of the volume.
- readonly UnsafeArray< int > TriangleIndices

  The vertex indices of triangles.
- readonly int BlockingTriangleIndexCount

  The number of indices in the TriangleIndices array that represent blocking triangles.
- readonly UnsafeArray< NativePlane > BoundPlanes

  The bound planes of all of the volume's regions.
- readonly UnsafeArray< NativeNavVolumeInternalLinkData > InternalLinks

  The internal links of all of the volume's regions.
- readonly UnsafeArray< NativeNavExternalLinkData > ExternalLinks

  The external links of all of the volume's regions.
- readonly UnsafeArray< int > LinkVertices

  The shared vertices of all of the volume's internal links.
- readonly UnsafeArray< int2 > LinkEdges

  The shared edges of all of the volume's internal links.
- readonly UnsafeArray< int3 > LinkTriangles

  The shared triangles of all of the volume's internal links.

6.23.1 Detailed Description

The baked data of a NavVolume, converted to a form compatible with Burst.

6.23.2 Constructor & Destructor Documentation

6.23.2.1 NativeNavVolumeData() `Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.NativeNavVolumeData (`

```
long id,
float4x4 transform,
float4x4 inverseTransform,
NativeBounds bounds,
NavLayer layer,
UnsafeArray< float4 > vertices,
UnsafeArray< NativeNavVolumeRegionData > regions,
UnsafeArray< int > triangleIndices,
int blockingTriangleIndexCount,
UnsafeArray< NativePlane > boundPlanes,
UnsafeArray< NativeNavVolumeInternalLinkData > internalLinks,
UnsafeArray< NativeNavExternalLinkData > externalLinks,
UnsafeArray< int > linkVertices,
UnsafeArray< int2 > linkEdges,
UnsafeArray< int3 > linkTriangles )
```

Initialize a new NativeNavVolumeData with the given data.

Parameters

id	ID of the volume.
transform	Transform matrix of the volume.
inverseTransform	Inverse transform matrix of the volume.
bounds	Bounds of the volume in local space.
layer	The layer this area exists on.
vertices	The vertex positions of all of the volume's regions.
regions	The regions of the volume.
triangleIndices	The indices of triangles.
blockingTriangleIndexCount	The number of indices in the TriangleIndices array that represent blocking triangles.
boundPlanes	The bound planes of all of the volume's regions.
internalLinks	The internal links of all of the volume's regions.
externalLinks	The external links of all of the volume's regions.
linkVertices	The shared vertices of all of the volume's internal links.
linkEdges	The shared edges of all of the volume's internal links.
linkTriangles	The shared triangles of all of the volume's internal links.

6.23.3 Member Data Documentation

6.23.3.1 BlockingTriangleIndexCount `readonly int Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.BlockingTriangleIndexCount`

The number of indices in the TriangleIndices array that represent blocking triangles.

6.23.3.2 BoundPlanes

`readonly UnsafeArray<NativePlane> Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.BoundPlanes`

The bound planes of all of the volume's regions.

6.23.3.3 Bounds

`readonly NativeBounds Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.Bounds`

Bounds of the volume in local space.

6.23.3.4 ExternalLinks

`readonly UnsafeArray<NativeNavExternalLinkData> Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.ExternalLinks`

The external links of all of the volume's regions.

6.23.3.5 ID

`readonly long Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.ID`

ID of the volume.

6.23.3.6 InternalLinks

`readonly UnsafeArray<NativeNavVolumeInternalLinkData> Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.InternalLinks`

The internal links of all of the volume's regions.

6.23.3.7 InverseTransform

`readonly float4x4 Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.InverseTransform`

Inverse transform matrix of the volume.

6.23.3.8 Layer

`readonly NavLayer Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.Layer`

The layer this area exists on.

6.23.3.9 LinkEdges

`readonly UnsafeArray<int2> Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.LinkEdges`

The shared edges of all of the volume's internal links.

6.23.3.10 LinkTriangles

`readonly UnsafeArray<int3> Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.LinkTriangles`

The shared triangles of all of the volume's internal links.

6.23.3.11 LinkVertices

`readonly UnsafeArray<int> Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.LinkVertices`

The shared vertices of all of the volume's internal links.

6.23.3.12 Regions

`readonly UnsafeArray<NativeNavVolumeRegionData> Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.Regions`

The regions of the volume.

6.23.3.13 Transform

`readonly float4x4 Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.Transform`

Transform matrix of the volume.

6.23.3.14 TriangleIndices

`readonly UnsafeArray<int> Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.TriangleIndices`

The vertex indices of triangles.

6.23.3.15 Vertices

`readonly UnsafeArray<float4> Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeData.Vertices`

The vertex positions of all of the volume's regions, in local space.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Utility/NavVolumeNativeData.cs

6.24 Infohazard.HyperNav.NativeNavVolumeDataPointers Struct Reference

References to the NativeArrays allocated for a NativeNavVolumeData.

Public Member Functions

- void Dispose ()

  Dispose all of the native array references.

6.24.1 Detailed Description

References to the NativeArrays allocated for a NativeNavVolumeData.

In the NativeNavVolumeData itself, these arrays are kept as pointers, which cannot be used to deallocate the arrays under Unity's safe memory system. In order to play nicely with that system, the original references must be kept and disposed.

6.24.2 Member Function Documentation

6.24.2.1 Dispose() `void Infohazard.HyperNav.NativeNavVolumeDataPointers.Dispose ( )`

Dispose all of the native array references.

The documentation for this struct was generated from the following file:

- Runtime/NavVolumeData.cs

6.25 Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeInternalLinkData Struct Reference

The native-friendly data representing a connection from one region to another region in the same volume.

Public Member Functions

- NativeNavVolumeInternalLinkData (int toRegion, SerializableRange vertexRange, SerializableRange edge↩ Range, SerializableRange triangleRange)

  Initialize a new NativeNavInternalLinkData with the given data.

Public Attributes

- readonly int ToRegion

  The ID of the connected region.
- readonly SerializableRange VertexRange

  The range of the link's vertices in the volume's NativeNavVolumeData.LinkVertices list.
- readonly SerializableRange EdgeRange

  The range of the link's edges in the volume's NativeNavVolumeData.LinkEdges list.
- readonly SerializableRange TriangleRange

  The range of the link's triangles in the volume's NativeNavVolumeData.LinkTriangles list.

6.25.1 Detailed Description

The native-friendly data representing a connection from one region to another region in the same volume.

6.25.2 Constructor & Destructor Documentation

6.25.2.1 NativeNavVolumeInternalLinkData() `Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeInternalLinkData.NativeNavVolumeInternalLinkData (`

```
            int toRegion,
            SerializableRange vertexRange,
            SerializableRange edgeRange,
            SerializableRange triangleRange )
```

Initialize a new NativeNavInternalLinkData with the given data.

Parameters

toRegion	The ID of the connected region.
vertexRange	The range of the link's vertices.
edgeRange	The range of the link's edges.
triangleRange	The range of the link's triangles.

6.25.3 Member Data Documentation

6.25.3.1 EdgeRange `readonly SerializableRange Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeInternalLinkData.EdgeRange`

The range of the link's edges in the volume's NativeNavVolumeData.LinkEdges list.

6.25.3.2 ToRegion `readonly int Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeInternalLinkData.ToRegion`

The ID of the connected region.

6.25.3.3 TriangleRange `readonly SerializableRange Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeInternalLinkData.TriangleRange`

The range of the link's triangles in the volume's NativeNavVolumeData.LinkTriangles list.

6.25.3.4 VertexRange `readonly SerializableRange Infohazard.HyperNav.Jobs.Utility.NativeNav↩ VolumeInternalLinkData.VertexRange`

The range of the link's vertices in the volume's NativeNavVolumeData.LinkVertices list.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Utility/NavVolumeNativeData.cs

6.26 Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeRegionData Struct Reference

The native-friendly data representing a single region in a NavVolume.

Public Member Functions

- NativeNavVolumeRegionData (int id, NativeBounds bounds, SerializableRange triangleIndexRange, SerializableRange boundPlaneRange, SerializableRange internalLinkRange, SerializableRange external↩ LinkRange)

  Initialize a new NativeNavRegionData with the given data.

Public Attributes

- readonly int ID

  The ID of the region.
- readonly NativeBounds Bounds

  The bounds of the region in local space of the volume.
- readonly SerializableRange TriangleIndexRange

  The range of the region's triangles in the volume's NativeNavVolumeData.TriangleIndices list.
- readonly SerializableRange BoundPlaneRange

  The range of the region's bound planes in the volume's NativeNavVolumeData.BoundPlanes list.
- readonly SerializableRange InternalLinkRange

  The range of the region's internal links in the volume's NativeNavVolumeData.InternalLinks list.
- readonly SerializableRange ExternalLinkRange

  The range of the region's externals link in the volume's NativeNavVolumeData.ExternalLinks list.

6.26.1 Detailed Description

The native-friendly data representing a single region in a NavVolume.

6.26.2 Constructor & Destructor Documentation

6.26.2.1 NativeNavVolumeRegionData() `Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeRegion↩ Data.NativeNavVolumeRegionData (`

```
            int id,
            NativeBounds bounds,
            SerializableRange triangleIndexRange,
            SerializableRange boundPlaneRange,
            SerializableRange internalLinkRange,
            SerializableRange externalLinkRange )
```

Initialize a new NativeNavRegionData with the given data.

Parameters

id	The ID of the region.
bounds	The bounds of the region in local space of the volume.
triangleIndexRange	The range of the region's triangle indices.
boundPlaneRange	The range of the region's bound planes.
internalLinkRange	The range of the region's internal links.
externalLinkRange	The range of the region's external links.

6.26.3 Member Data Documentation

6.26.3.1 BoundPlaneRange `readonly SerializableRange Infohazard.HyperNav.Jobs.Utility.↩ NativeNavVolumeRegionData.BoundPlaneRange`

The range of the region's bound planes in the volume's NativeNavVolumeData.BoundPlanes list.

6.26.3.2 Bounds `readonly NativeBounds Infohazard.HyperNav.Jobs.Utility.NativeNavVolume↩ RegionData.Bounds`

The bounds of the region in local space of the volume.

6.26.3.3 ExternalLinkRange `readonly SerializableRange Infohazard.HyperNav.Jobs.Utility.↩ NativeNavVolumeRegionData.ExternalLinkRange`

The range of the region's externals link in the volume's NativeNavVolumeData.ExternalLinks list.

6.26.3.4 ID `readonly int Infohazard.HyperNav.Jobs.Utility.NativeNavVolumeRegionData.ID`

The ID of the region.

6.26.3.5 InternalLinkRange `readonly SerializableRange Infohazard.HyperNav.Jobs.Utility.Native↩ NavVolumeRegionData.InternalLinkRange`

The range of the region's internal links in the volume's NativeNavVolumeData.InternalLinks list.

6.26.3.6 TriangleIndexRange `readonly SerializableRange Infohazard.HyperNav.Jobs.Utility.↩ NativeNavVolumeRegionData.TriangleIndexRange`

The range of the region's triangles in the volume's NativeNavVolumeData.TriangleIndices list.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Utility/NavVolumeNativeData.cs

6.27 Infohazard.HyperNav.Jobs.Utility.NativeNavWaypoint Struct Reference

A structure used by the navigation job to return the waypoints of a path.

Public Member Functions

- NativeNavWaypoint (float4 position, float4 up, NavWaypointType type, long areaID, int region)

  Initialize a new NativeNavWaypoint with the given data.

Public Attributes

- readonly float4 Position

  Position of the waypoint in world space.
- readonly float4 Up

  Upward direction of the waypoint.
- readonly NavWaypointType Type

  Type of the waypoint in relation to the containing volume.
- readonly long AreaID

  Identifier of the NavArea that contains this waypoint, or -1.
- readonly int Region

  Identifier of the region that contains this waypoint, or -1.

6.27.1 Detailed Description

A structure used by the navigation job to return the waypoints of a path.

6.27.2 Constructor & Destructor Documentation

6.27.2.1 NativeNavWaypoint() `Infohazard.HyperNav.Jobs.Utility.NativeNavWaypoint.NativeNav↩ Waypoint (`

```
            float4 position,
            float4 up,
            NavWaypointType type,
            long areaID,
            int region )
```

Initialize a new NativeNavWaypoint with the given data.

Parameters

position	Position of the waypoint in world space.
up	Upward direction of the waypoint.
type	Type of the waypoint in relation to the containing volume.
areaID	Identifier of the NavVolume that contains this waypoint, or -1.
region	Identifier of the region that contains this waypoint, or -1.

6.27.3 Member Data Documentation

6.27.3.1 AreaID `readonly long Infohazard.HyperNav.Jobs.Utility.NativeNavWaypoint.AreaID`

Identifier of the NavArea that contains this waypoint, or -1.

6.27.3.2 Position `readonly float4 Infohazard.HyperNav.Jobs.Utility.NativeNavWaypoint.Position`

Position of the waypoint in world space.

6.27.3.3 Region `readonly int Infohazard.HyperNav.Jobs.Utility.NativeNavWaypoint.Region`

Identifier of the region that contains this waypoint, or -1.

6.27.3.4 Type `readonly NavWaypointType Infohazard.HyperNav.Jobs.Utility.NativeNavWaypoint.Type`

Type of the waypoint in relation to the containing volume.

6.27.3.5 Up `readonly float4 Infohazard.HyperNav.Jobs.Utility.NativeNavWaypoint.Up`

Upward direction of the waypoint.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Utility/NavVolumeNativeData.cs

6.28 Infohazard.HyperNav.Jobs.NativePlane Struct Reference

A plane constructed using native math types.

Public Member Functions

- NativePlane (float4 normal, float4 point)

  Construct a new NativePlane, calculating the distance based on any point in the plane.
- NativePlane (float3 normal, float3 point)

  Construct a new NativePlane, calculating the distance based on any point in the plane.

Public Attributes

- float4 NormalDistance

  Normal of the plane (xyz, normalized) and distance from the origin (w).

Properties

- float4 Center `[get]`

  Nearest point to the origin on the plane.

6.28.1 Detailed Description

A plane constructed using native math types.

6.28.2 Constructor & Destructor Documentation

6.28.2.1 NativePlane() [1/2]

```
Infohazard.HyperNav.Jobs.NativePlane.NativePlane (
        float4 normal,
        float4 point )
```

Construct a new NativePlane, calculating the distance based on any point in the plane.

Parameters

normal	Normal of the plane, which should be normalized.
point	Any point on the plane.

6.28.2.2 NativePlane() [2/2]

```
Infohazard.HyperNav.Jobs.NativePlane.NativePlane (
        float3 normal,
        float3 point )
```

Construct a new NativePlane, calculating the distance based on any point in the plane.

Parameters

normal	Normal of the plane, which should be normalized.
point	Any point on the plane.

6.28.3 Member Data Documentation

6.28.3.1 NormalDistance `float4 Infohazard.HyperNav.Jobs.NativePlane.NormalDistance`

Normal of the plane (xyz, normalized) and distance from the origin (w).

6.28.4 Property Documentation

6.28.4.1 Center `float4 Infohazard.HyperNav.Jobs.NativePlane.Center [get]`

Nearest point to the origin on the plane.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/NativeAvoidanceData.cs

6.29 Infohazard.HyperNav.Jobs.NativeRay Struct Reference

A ray constructed using native math types.

Public Attributes

- float4 Origin
  Origin of the ray.
- float4 Direction
  Direction of the ray, which should be normalized.

6.29.1 Detailed Description

A ray constructed using native math types.

6.29.2 Member Data Documentation

6.29.2.1 Direction `float4 Infohazard.HyperNav.Jobs.NativeRay.Direction`

Direction of the ray, which should be normalized.

6.29.2.2 Origin `float4 Infohazard.HyperNav.Jobs.NativeRay.Origin`

Origin of the ray.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/NativeAvoidanceData.cs

6.30 Infohazard.HyperNav.Jobs.NativeRaycastElement Struct Reference

A single raycast in a NavMultiRaycastJob.

Public Attributes

- long VolumeID
  Volume to raycast in.
- float OutDistance
  Where the hit distance (or -1 if no hit) of the raycast is written.
- float4 Start
  Start point of the segment.
- float4 End
  End point of the segment.

6.30.1 Detailed Description

A single raycast in a NavMultiRaycastJob.

6.30.2 Member Data Documentation

6.30.2.1 End `float4 Infohazard.HyperNav.Jobs.NativeRaycastElement.End`

End point of the segment.

6.30.2.2 OutDistance `float Infohazard.HyperNav.Jobs.NativeRaycastElement.OutDistance`

Where the hit distance (or -1 if no hit) of the raycast is written.

6.30.2.3 Start `float4 Infohazard.HyperNav.Jobs.NativeRaycastElement.Start`

Start point of the segment.

6.30.2.4 VolumeID `long Infohazard.HyperNav.Jobs.NativeRaycastElement.VolumeID`

Volume to raycast in.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/NavRaycastJob.cs

6.31 Infohazard.HyperNav.NavAgent Class Reference

A script that can be used to calculate paths by any entity that needs to use HyperNav for navigation.

Classes

- class PropNames

  This is used to refer to the names of private fields in this class from a custom Editor.

Public Member Functions

- virtual void Stop (bool abortPaths)

  Stop following the current path, and optionally cancel all path requests.
- virtual UpdatePathResult SetDestinationByNavHit (NavSampleResult fromHit, NavSampleResult toHit)

  Request a new path between two NavSampleResults.
- virtual UpdatePathResult UpdatePath (Vector3 destination)

  Request a new path from the current position to the desired destination. This is equivalent to setting Destination, but returns a result.

Protected Member Functions

- virtual void Awake ()

  Sets the AvoidanceAgent.Infohazard.HyperNav.AvoidanceAgent.InputVelocityFunc.
- virtual void OnEnable ()

  Resets MeasuredVelocity and sets Arrived to true.
- virtual void OnDisable ()

  Stops all pathfinding and cancels path requests.
- virtual void Update ()

  Updates measured velocity and current index in path.
- virtual void OnDrawGizmos ()

  Draws the current path as a sequence of debug lines if DebugPath is true.
- virtual Vector3 CalculateDesiredNavigationVelocity ()

  Calculate the velocity the agent wants to move in, in the range [0, 1].
- virtual void UpdateMeasuredVelocity ()

  Update the value of MeasuredVelocity, which is used to determine StoppingDistance.
- virtual void UpdatePathIndex (bool isInitial)

  Update the current path index, which is used to determine NextWaypointPosition.
- virtual void OnPathReady (long id, NavPath path)

  Callback that is received when a pathfinding request completes, which should start moving along that path.

Properties

- float Acceptance [get, set]

  How close the agent must get to a destination before it is considered to have arrived.
- float AccelerationEstimate [get, set]

  This should be set to the maximum acceleration of your agent.
- Rigidbody Rigidbody [get, set]

  Optional rigidbody to use for measuring velocity.
- float SampleRadius [get, set]

  The radius to search when finding the nearest NavVolume.
- NavAreaTypes AreaTypeMask [get, set]

  The types of areas that the agent can traverse.
- NavLayerMask AreaLayerMask [get, set]

  The layers of areas that the agent can traverse.
- NavSamplePriority StartSamplingPriority [get, set]

  How to prioritize results for the destination position when sampling in both volumes and surfaces.
- NavSamplePriority DestSamplingPriority [get, set]

  How to prioritize results for the destination position when sampling in both volumes and surfaces.
- Transform VolumeSamplingTransform [get, set]

  The transform to use for sampling the agent's position in volumes.
- Transform SurfaceSamplingTransform [get, set]

  The transform to use for sampling the agent's position on surfaces.
- bool CheckSurfaceUpVector [get, set]

  Whether to check that the surface up vector is within a certain dot product of the sampling up vector. The reference vector will be the up vector of either SurfaceSamplingTransform if it is set, or the agent's transform otherwise.
- float MinSurfaceUpVectorDot [get, set]

  The minimum dot product between the surface up vector and the sampling up vector to consider a surface valid.
- float CostToChangeToVolume [get, set]

  Additional cost to changing from a surface to a volume.
- float CostToChangeToSurface [get, set]

  Additional cost to changing from a volume to a surface.
- float VolumeCostMultiplier [get, set]

  Multiplier for the cost of traversing volume areas.
- float SurfaceCostMultiplier [get, set]

  Multiplier for the cost of traversing surface areas.
- NavPathfindingParams PathfindingParams [get, set]

  Pathfinding parameters as a combined struct.
- float DesiredSpeedRatio [get, set]

  The desired fraction of the maximum speed to travel at.
- bool DebugPath [get, set]

  Whether to draw a debug line in the scene view showing the agent's current path.
- bool KeepPathWhileCalculating [get, set]

  Whether to keep following the current path while waiting for a new path to finish calculating.
- AvoidanceAgent AvoidanceAgent [get, set]

  AvoidanceAgent that this agent uses for avoidance (can be null).
- bool ControlAvoidanceIsActive [get, set]

  If true, the Infohazard.HyperNav.AvoidanceAgent.IsActive state of the AvoidanceAgent is set based on whether there is a current valid path.
- bool CheckSkippingWithPhysicsQuery [get, set]

  Whether to check during path following if the agent should skip to the next waypoint.
- bool CheckBacktrackWithPhysicsQuery [get, set]

Whether to check during path following if the agent should go back to the previous waypoint.

- LayerMask SkippingCheckLayerMask `[get, set]`

  Layer mask to use for physics queries to check for obstacles when skipping waypoints.
- Collider SkippingCheckCollider `[get, set]`

  Collider to determine the query parameters for sweep testing.
- float SkippingCheckColliderPadding `[get, set]`

  Padding to reduce the size of the collider used for skipping waypoint checks.
- bool SkippingCheckStaticOnly `[get, set]`

  Whether to only consider static colliders when checking for skipping waypoints.
- bool BacktrackCheckStaticOnly `[get, set]`

  Whether to only consider static colliders when checking for backtracking waypoints.
- bool IsPathPending `[get]`

  Whether a path is currently in the process of being calculated for this agent.
- virtual float StoppingDistance `[get]`

  The distance that it will take the agent to come to a stop from its current velocity, determined using the AccelerationEstimate.
- Vector3 NextWaypointPosition `[get]`

  The current path waypoint that the agent is trying to move towards.
- Vector3 PositionForVolume `[get]`

  The position to use for sampling volumes.
- Vector3 PositionForSurface `[get]`

  The position to use for sampling surfaces.
- NavWaypoint? NextWaypoint `[get]`

  The current path waypoint that the agent is trying to move towards.
- NavWaypoint? PreviousWaypoint `[get]`

  The previous waypoint in the path.
- bool Arrived `[get, private set]`

  True if the agent has no active or pending path.
- float RemainingDistance `[get, protected set]`

  The remaining distance to the destination.
- float RemainingDistanceToNextWaypoint `[get, protected set]`

  The remaining distance to the next waypoint.
- float DistanceFromStart `[get, protected set]`

  The distance the agent has traveled along the path.
- Vector3 Destination `[get, set]`

  Get or set the agent's destination (the position it is trying to get to).
- Vector3 MeasuredVelocity `[get, protected set]`

  Velocity of the agent measured as delta position / delta time over the last frame, which is used to determine stopping distance.
- NavPath CurrentPath `[get, set]`

  The current path that the agent is following.
- float AvoidanceMaxSpeed `[get, set]`

  Maximum speed possible by this agent when avoiding obstacles.
- NavAgentWaypointPredicate AdvancePredicate `[get, set]`

  An optional condition to check if the agent can advance past a given waypoint in the path.
- NavAgentWaypointPredicate BacktrackPredicate `[get, set]`

  An optional condition to check if the agent can backtrack to the previous waypoint in the path. Will only be used if a physics query determines the current waypoint is unreachable.

Events

- Action PathReady

  Invoked when the agent finds a path to the destination.
- Action PathFailed

  Invoked when the agent fails to find a path to the destination.
- Action< NavPath > CurrentPathChanged

  Invoked when the agent's path is updated.
- Action< int > PathIndexChanged

  Invoked when the agent's path index changes.

6.31.1 Detailed Description

A script that can be used to calculate paths by any entity that needs to use HyperNav for navigation.

While a NavAgent is not necessary to use HyperNav, it makes pathfinding easier. The NavAgent does not impose any restrictions on how movement occurs, nor does it actually perform any movement. It simply provides a desired movement velocity, which other scripts on the object are responsible for using however they need.

The agent can have one active path (the path it is currently following), but can have multiple pending paths (paths in the process of being calculated by a NavPathfinder).

If you desire smoother movement then what the NavAgent provides, see SplineNavAgent.

6.31.2 Member Function Documentation

6.31.2.1 Awake() `virtual void Infohazard.HyperNav.NavAgent.Awake ( ) [protected], [virtual]`

Sets the AvoidanceAgent.Infohazard.HyperNav.AvoidanceAgent.InputVelocityFunc.

6.31.2.2 CalculateDesiredNavigationVelocity() `virtual Vector3 Infohazard.HyperNav.NavAgent.↩ CalculateDesiredNavigationVelocity ( ) [protected], [virtual]`

Calculate the velocity the agent wants to move in, in the range [0, 1].

Reimplemented in Infohazard.HyperNav.SplineNavAgent.

6.31.2.3 OnDisable() `virtual void Infohazard.HyperNav.NavAgent.OnDisable ( ) [protected], [virtual]`

Stops all pathfinding and cancels path requests.

Reimplemented in Infohazard.HyperNav.SplineNavAgent.

6.31.2.4 OnDrawGizmos() `virtual void Infohazard.HyperNav.NavAgent.OnDrawGizmos ( ) [protected], [virtual]`

Draws the current path as a sequence of debug lines if DebugPath is true.

Reimplemented in Infohazard.HyperNav.SplineNavAgent.

6.31.2.5 OnEnable() `virtual void Infohazard.HyperNav.NavAgent.OnEnable ( ) [protected], [virtual]`

Resets MeasuredVelocity and sets Arrived to true.

6.31.2.6 OnPathReady() `virtual void Infohazard.HyperNav.NavAgent.OnPathReady ( long id, NavPath path ) [protected], [virtual]`

Callback that is received when a pathfinding request completes, which should start moving along that path.

Parameters

id	The id of the path request.
path	The completed path, which is null if no path was found.

6.31.2.7 SetDestinationByNavHit() `virtual UpdatePathResult Infohazard.HyperNav.NavAgent.SetDestinationByNavHit ( NavSampleResult fromHit, NavSampleResult toHit ) [virtual]`

Request a new path between two NavSampleResults.

This method is similar to setting Destination. However, you can use this if you want to supply your own query results. For example, you can run a single NavSampleJob for many pathfinding calls.

Parameters

fromHit	Hit at the agent's current position.
toHit	Hit at the destination.

Returns

The result of the path update.

6.31.2.8 Stop() `virtual void Infohazard.HyperNav.NavAgent.Stop ( bool abortPaths ) [virtual]`

Stop following the current path, and optionally cancel all path requests.

Parameters

abortPaths	Whether to cancel pending path requests.

Reimplemented in Infohazard.HyperNav.SplineNavAgent.

6.31.2.9 Update() `virtual void Infohazard.HyperNav.NavAgent.Update ( ) [protected], [virtual]`

Updates measured velocity and current index in path.

Reimplemented in Infohazard.HyperNav.SplineNavAgent.

6.31.2.10 UpdateMeasuredVelocity() `virtual void Infohazard.HyperNav.NavAgent.UpdateMeasuredVelocity ( ) [protected], [virtual]`

Update the value of MeasuredVelocity, which is used to determine StoppingDistance.

6.31.2.11 UpdatePath() `virtual UpdatePathResult Infohazard.HyperNav.NavAgent.UpdatePath ( Vector3 destination ) [virtual]`

Request a new path from the current position to the desired destination. This is equivalent to setting Destination, but returns a result.

It is usually not necessary to call this yourself, as it is called when setting Destination. However, if the agent gets stuck or pushed off course, you may wish to use this to get a new path.

Parameters

destination	The position to move towards.

Returns

The result of the path update.

6.31.2.12 UpdatePathIndex() `virtual void Infohazard.HyperNav.NavAgent.UpdatePathIndex ( bool isInitial ) [protected], [virtual]`

Update the current path index, which is used to determine NextWaypointPosition.

6.31.3 Property Documentation

6.31.3.1 AccelerationEstimate `float Infohazard.HyperNav.NavAgent.AccelerationEstimate [get], [set]`

This should be set to the maximum acceleration of your agent.

This is used to determine when the agent needs to start slowing down when approaching its destination.

6.31.3.2 Acceptance `float Infohazard.HyperNav.NavAgent.Acceptance [get], [set]`

How close the agent must get to a destination before it is considered to have arrived.

Note that setting acceptance too low may prevent the agent from ever stopping, but setting it to high can make the agent stop too far from the destination.

6.31.3.3 AdvancePredicate `NavAgentWaypointPredicate Infohazard.HyperNav.NavAgent.Advance↩ Predicate [get], [set]`

An optional condition to check if the agent can advance past a given waypoint in the path.

A multicast delegate is not supported.

6.31.3.4 AreaLayerMask `NavLayerMask Infohazard.HyperNav.NavAgent.AreaLayerMask [get], [set]`

The layers of areas that the agent can traverse.

6.31.3.5 AreaTypeMask `NavAreaTypes Infohazard.HyperNav.NavAgent.AreaTypeMask [get], [set]`

The types of areas that the agent can traverse.

6.31.3.6 Arrived `bool Infohazard.HyperNav.NavAgent.Arrived [get], [private set]`

True if the agent has no active or pending path.

6.31.3.7 AvoidanceAgent `AvoidanceAgent Infohazard.HyperNav.NavAgent.AvoidanceAgent [get], [set]`

AvoidanceAgent that this agent uses for avoidance (can be null).

6.31.3.8 AvoidanceMaxSpeed `float Infohazard.HyperNav.NavAgent.AvoidanceMaxSpeed [get], [set]`

Maximum speed possible by this agent when avoiding obstacles.

6.31.3.9 BacktrackCheckStaticOnly `bool Infohazard.HyperNav.NavAgent.BacktrackCheckStaticOnly [get], [set]`

Whether to only consider static colliders when checking for backtracking waypoints.

6.31.3.10 BacktrackPredicate `NavAgentWaypointPredicate Infohazard.HyperNav.NavAgent.BacktrackPredicate [get], [set]`

An optional condition to check if the agent can backtrack to the previous waypoint in the path. Will only be used if a physics query determines the current waypoint is unreachable.

6.31.3.11 CheckBacktrackWithPhysicsQuery `bool Infohazard.HyperNav.NavAgent.CheckBacktrackWithPhysicsQuery [get], [set]`

Whether to check during path following if the agent should go back to the previous waypoint.

6.31.3.12 CheckSkippingWithPhysicsQuery `bool Infohazard.HyperNav.NavAgent.CheckSkippingWithPhysicsQuery [get], [set]`

Whether to check during path following if the agent should skip to the next waypoint.

6.31.3.13 CheckSurfaceUpVector `bool Infohazard.HyperNav.NavAgent.CheckSurfaceUpVector [get], [set]`

Whether to check that the surface up vector is within a certain dot product of the sampling up vector. The reference vector will be the up vector of either SurfaceSamplingTransform if it is set, or the agent's transform otherwise.

6.31.3.14 ControlAvoidanceIsActive `bool Infohazard.HyperNav.NavAgent.ControlAvoidanceIsActive [get], [set]`

If true, the Infohazard.HyperNav.AvoidanceAgent.IsActive state of the AvoidanceAgent is set based on whether there is a current valid path.

6.31.3.15 CostToChangeToSurface `float Infohazard.HyperNav.NavAgent.CostToChangeToSurface [get], [set]`

Additional cost to changing from a volume to a surface.

6.31.3.16 CostToChangeToVolume `float Infohazard.HyperNav.NavAgent.CostToChangeToVolume [get], [set]`

Additional cost to changing from a surface to a volume.

6.31.3.17 CurrentPath `NavPath Infohazard.HyperNav.NavAgent.CurrentPath [get], [set]`

The current path that the agent is following.

6.31.3.18 DebugPath `bool Infohazard.HyperNav.NavAgent.DebugPath [get], [set]`

Whether to draw a debug line in the scene view showing the agent's current path.

6.31.3.19 DesiredSpeedRatio `float Infohazard.HyperNav.NavAgent.DesiredSpeedRatio [get], [set]`

The desired fraction of the maximum speed to travel at.

6.31.3.20 Destination `Vector3 Infohazard.HyperNav.NavAgent.Destination [get], [set]`

Get or set the agent's destination (the position it is trying to get to).

If set within the _acceptance radius of the current position, will abort all movement.

6.31.3.21 DestSamplingPriority `NavSamplePriority Infohazard.HyperNav.NavAgent.DestSampling↩ Priority [get], [set]`

How to prioritize results for the destination position when sampling in both volumes and surfaces.

6.31.3.22 DistanceFromStart `float Infohazard.HyperNav.NavAgent.DistanceFromStart [get], [protected set]`

The distance the agent has traveled along the path.

6.31.3.23 IsPathPending `bool Infohazard.HyperNav.NavAgent.IsPathPending [get]`

Whether a path is currently in the process of being calculated for this agent.

6.31.3.24 KeepPathWhileCalculating `bool Infohazard.HyperNav.NavAgent.KeepPathWhileCalculating [get], [set]`

Whether to keep following the current path while waiting for a new path to finish calculating.

If true, there can be two pending paths at the same time - the most and least recently requested ones. This ensures that even when the agent is receiving pathfinding requests faster than they can be calculated, they will still finish and the agent will not be deadlocked and unable to ever complete a path.

6.31.3.25 MeasuredVelocity `Vector3 Infohazard.HyperNav.NavAgent.MeasuredVelocity [get], [protected set]`

Velocity of the agent measured as delta position / delta time over the last frame, which is used to determine stopping distance.

This value is calculated in UpdateMeasuredVelocity. You can override that method to implement your own logic for calculating velocity.

6.31.3.26 MinSurfaceUpVectorDot `float Infohazard.HyperNav.NavAgent.MinSurfaceUpVectorDot [get], [set]`

The minimum dot product between the surface up vector and the sampling up vector to consider a surface valid.

6.31.3.27 NextWaypoint `NavWaypoint? Infohazard.HyperNav.NavAgent.NextWaypoint [get]`

The current path waypoint that the agent is trying to move towards.

6.31.3.28 NextWaypointPosition `Vector3 Infohazard.HyperNav.NavAgent.NextWaypointPosition [get]`

The current path waypoint that the agent is trying to move towards.

If there is no active path, will return the agent's current position.

6.31.3.29 PathfindingParams `NavPathfindingParams Infohazard.HyperNav.NavAgent.PathfindingParams [get], [set]`

Pathfinding parameters as a combined struct.

6.31.3.30 PositionForSurface `Vector3 Infohazard.HyperNav.NavAgent.PositionForSurface [get]`

The position to use for sampling surfaces.

6.31.3.31 PositionForVolume `Vector3 Infohazard.HyperNav.NavAgent.PositionForVolume [get]`

The position to use for sampling volumes.

6.31.3.32 PreviousWaypoint `NavWaypoint? Infohazard.HyperNav.NavAgent.PreviousWaypoint [get]`

The previous waypoint in the path.

6.31.3.33 RemainingDistance `float Infohazard.HyperNav.NavAgent.RemainingDistance [get], [protected set]`

The remaining distance to the destination.

6.31.3.34 RemainingDistanceToNextWaypoint `float Infohazard.HyperNav.NavAgent.RemainingDistanceToNextWaypoint [get], [protected set]`

The remaining distance to the next waypoint.

6.31.3.35 Rigidbody `Rigidbody Infohazard.HyperNav.NavAgent.Rigidbody [get], [set]`

Optional rigidbody to use for measuring velocity.

6.31.3.36 SampleRadius `float Infohazard.HyperNav.NavAgent.SampleRadius [get], [set]`

The radius to search when finding the nearest NavVolume.

6.31.3.37 SkippingCheckCollider `Collider Infohazard.HyperNav.NavAgent.SkippingCheckCollider [get], [set]`

Collider to determine the query parameters for sweep testing.

The layer does not matter and it does not need to be enabled. However, it must be of type Box, Capsule, or Sphere, as Unity does not support Mesh casts.

6.31.3.38 SkippingCheckColliderPadding

`float Infohazard.HyperNav.NavAgent.SkippingCheckColliderPadding [get], [set], [add]`

Padding to reduce the size of the collider used for skipping waypoint checks.

6.31.3.39 SkippingCheckLayerMask

`LayerMask Infohazard.HyperNav.NavAgent.SkippingCheckLayerMask [get], [set]`

Layer mask to use for physics queries to check for obstacles when skipping waypoints.

6.31.3.40 SkippingCheckStaticOnly

`bool Infohazard.HyperNav.NavAgent.SkippingCheckStaticOnly [get], [set]`

Whether to only consider static colliders when checking for skipping waypoints.

6.31.3.41 StartSamplingPriority

`NavSamplePriority Infohazard.HyperNav.NavAgent.StartSamplingPriority [get], [set]`

How to prioritize results for the destination position when sampling in both volumes and surfaces.

6.31.3.42 StoppingDistance

`virtual float Infohazard.HyperNav.NavAgent.StoppingDistance [get]`

The distance that it will take the agent to come to a stop from its current velocity, determined using the AccelerationEstimate.

6.31.3.43 SurfaceCostMultiplier

`float Infohazard.HyperNav.NavAgent.SurfaceCostMultiplier [get], [set]`

Multiplier for the cost of traversing surface areas.

6.31.3.44 SurfaceSamplingTransform

`Transform Infohazard.HyperNav.NavAgent.SurfaceSamplingTransform [get], [set]`

The transform to use for sampling the agent's position on surfaces.

6.31.3.45 VolumeCostMultiplier `float Infohazard.HyperNav.NavAgent.VolumeCostMultiplier [get], [set]`

Multiplier for the cost of traversing volume areas.

6.31.3.46 VolumeSamplingTransform `Transform Infohazard.HyperNav.NavAgent.VolumeSampling↩ Transform [get], [set]`

The transform to use for sampling the agent's position in volumes.

6.31.4 Event Documentation

6.31.4.1 CurrentPathChanged `Action<NavPath> Infohazard.HyperNav.NavAgent.CurrentPathChanged`

Invoked when the agent's path is updated.

6.31.4.2 PathFailed `Action Infohazard.HyperNav.NavAgent.PathFailed`

Invoked when the agent fails to find a path to the destination.

6.31.4.3 PathIndexChanged `Action<int> Infohazard.HyperNav.NavAgent.PathIndexChanged`

Invoked when the agent's path index changes.

6.31.4.4 PathReady `Action Infohazard.HyperNav.NavAgent.PathReady`

Invoked when the agent finds a path to the destination.

The documentation for this class was generated from the following file:

- Runtime/NavAgent.cs

6.32 Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings > Class Template Reference

Generic base class for areas in which HyperNav pathfinding can occur.

Public Member Functions

- override void Register ()

  Register this volume in the Instances dictionary and perform initialization. This can be used in edit mode to create the native data. It does not need to be called at runtime.
- override void Deregister (bool destroyData)

  Remove this volume from the Instances dictionary, and optionally destroy its native data.
- void UpdateNativeData (in TNativeData newData, in TPointers newPointers, bool updateSerializedAreaData, bool updateSerializedLinkData, IReadOnlyList< string > externalLinkScenePaths=null)

  Update native data for the nav area, optionally referencing new arrays.

Static Public Member Functions

- static void UpdateAllTransforms ()

  Update the native data on all loaded NavAreas. Note, this must be called on each area type class (NavSurface and NavVolume) separately.

Static Public Attributes

- static NativeParallelHashMap< long, TNativeData > NativeDataMap

  Data for all loaded volumes in the format used by jobs.

Protected Member Functions

- override void Reset ()

  Reset certain properties to the first NavArea on the object.

Properties

- TData Data `[get, set]`
- ref readonly TNativeData NativeData `[get]`

  The native data for the volume. Will not be initialized until the volume is registered.
- ref readonly TPointers DataStructurePointers `[get]`

  The internal pointers to the native data for the volume.
- TSettings InstanceSettings `[get, set]`

  Settings specific to this area instance. On set, SharedSettings will be set to null.
- NavAreaBaseSettingsAsset< TSettings > SharedSettings `[get, set]`

  Shared settings asset for all areas of this type. If not null, InstanceSettings will be ignored.
- TSettings Settings `[get]`

  The settings object in use by this area, whether shared or instance. The returned object, while mutable, should not be modified directly as it may reference a ScriptableObject.
- override NavAreaBaseSettings BaseSettings `[get]`
- override bool IsNativeDataCreated `[get]`
- override NavLayer Layer `[get, set]`
- static IReadOnlyDictionary< long, TArea > Instances `[get]`

  All currently loaded volumes.

6.32.1 Detailed Description

Generic base class for areas in which HyperNav pathfinding can occur.

Type Constraints

TArea **:** ***NavArea***
TArea **:** ***TArea***
TArea **:** ***TData***
TArea **:** ***TNativeData***
TArea **:** ***TPointers***
TArea **:** ***TSettings***
TData **:** ***ScriptableObject***
TData **:** ***INavAreaData***
TData **:** ***TNativeData***
TData **:** ***TPointers***
TNativeData **:** ***unmanaged***
TNativeData **:** ***IDisposable***
TNativeData **:** ***INativeNavAreaData***
TPointers **:** ***struct***
TPointers **:** ***IDisposable***
TPointers **:** ***INativeNavAreaDataPointers***
TSettings **:** ***NavAreaBaseSettings*<*TSettings*>**
TSettings **:** ***new()***

6.32.2 Member Function Documentation

6.32.2.1 Deregister() override void Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >.Deregister (bool *destroyData*) [virtual]

Remove this volume from the Instances dictionary, and optionally destroy its native data.

Parameters

destroyData	Whether to free the memory used for native data.

Implements Infohazard.HyperNav.NavAreaBase.

6.32.2.2 Register() override void Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >.Register ( ) [virtual]

Register this volume in the Instances dictionary and perform initialization. This can be used in edit mode to create the native data. It does not need to be called at runtime.

Implements Infohazard.HyperNav.NavAreaBase.

6.32.2.3 Reset()

```
override void Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >.Reset ( )  [protected], [virtual]
```

Reset certain properties to the first NavArea on the object.

Reimplemented from Infohazard.HyperNav.NavAreaBase.

6.32.2.4 UpdateAllTransforms()

```
static void Infohazard.HyperNav.NavArea< TArea, TData, TNative↩
Data, TPointers, TSettings >.UpdateAllTransforms ( )  [static]
```

Update the native data on all loaded NavAreas. Note, this must be called on each area type class (NavSurface and NavVolume) separately.

Use this after moving all areas when AutoDetectMovement is disabled.

6.32.2.5 UpdateNativeData()

```
void Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >.UpdateNativeData (
            in TNativeData newData,
            in TPointers newPointers,
            bool updateSerializedAreaData,
            bool updateSerializedLinkData,
            IReadOnlyList< string > externalLinkScenePaths = null )
```

Update native data for the nav area, optionally referencing new arrays.

This does NOT automatically dispose the old arrays, as they may be shared with the new data. If you are not reusing the old arrays, you should dispose them manually.

Parameters

newData	New native data.
newPointers	New pointers to arrays.
updateSerializedAreaData	Whether to update the serialized area ScriptableObject as well. Note that this will incur a major performance cost because it will allocate managed memory. If this is used at runtime, it will copy the ScriptableObject to avoid modifying assets.
updateSerializedLinkData	Whether to update the serialized external link data as well. Note that this will incur a performance cost because it will allocate managed memory.
externalLinkScenePaths	If updating serialized link data, should contain the scene paths for each external link in the newData.ExternalLinks array.

6.32.3 Member Data Documentation

6.32.3.1 NativeDataMap `NativeParallelHashMap<long, TNativeData> Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >.NativeDataMap [static]`

Data for all loaded volumes in the format used by jobs.

6.32.4 Property Documentation

6.32.4.1 BaseSettings `override NavAreaBaseSettings Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >.BaseSettings [get]`

6.32.4.2 Data `TData Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >.Data [get], [set]`

6.32.4.3 DataStructurePointers `ref readonly TPointers Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >.DataStructurePointers [get]`

The internal pointers to the native data for the volume.

6.32.4.4 Instances `IReadOnlyDictionary<long, TArea> Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >.Instances [static], [get]`

All currently loaded volumes.

6.32.4.5 InstanceSettings `TSettings Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >.InstanceSettings [get], [set]`

Settings specific to this area instance. On set, SharedSettings will be set to null.

6.32.4.6 IsNativeDataCreated `override bool Infohazard.HyperNav.NavArea< TArea, TData, TNative↩ Data, TPointers, TSettings >.IsNativeDataCreated [get]`

6.32.4.7 Layer

`override NavLayer Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >.Layer [get], [set]`

6.32.4.8 NativeData

`ref readonly TNativeData Infohazard.HyperNav.NavArea< TArea, TData, TNative↩ Data, TPointers, TSettings >.NativeData [get]`

The native data for the volume. Will not be initialized until the volume is registered.

6.32.4.9 Settings

`TSettings Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >.Settings [get]`

The settings object in use by this area, whether shared or instance. The returned object, while mutable, should not be modified directly as it may reference a ScriptableObject.

6.32.4.10 SharedSettings

`NavAreaBaseSettingsAsset<TSettings> Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >.SharedSettings [get], [set]`

Shared settings asset for all areas of this type. If not null, InstanceSettings will be ignored.

The documentation for this class was generated from the following file:

- Runtime/NavArea.cs

6.33 Infohazard.HyperNav.NavAreaBakeProgress Struct Reference

Represents current bake state of a volume, including progress fraction and current operation display name.

6.33.1 Detailed Description

Represents current bake state of a volume, including progress fraction and current operation display name.

The documentation for this struct was generated from the following file:

- Runtime/Utility/NavAreaBaseBakeHandler.cs

6.34 Infohazard.HyperNav.NavAreaBakingUtility Class Reference

Utility methods used for baking NavAreas in both the editor and at runtime.

Static Public Member Functions

- static NativeHashSet< int > GetStaticCollidersForBaking ()

  Get the set of instance IDs for all static colliders relevant to baking. In the editor, if editing a prefab, only colliders in the prefab are considered. Otherwise, all colliders in the scene are considered.

6.34.1 Detailed Description

Utility methods used for baking NavAreas in both the editor and at runtime.

6.34.2 Member Function Documentation

6.34.2.1 GetStaticCollidersForBaking() `static NativeHashSet< int > Infohazard.HyperNav.NavAreaBakingUtility.GetStaticCollidersForBaking ( ) [static]`

Get the set of instance IDs for all static colliders relevant to baking. In the editor, if editing a prefab, only colliders in the prefab are considered. Otherwise, all colliders in the scene are considered.

Returns

A set of instance IDs for all static colliders relevant to baking.

The documentation for this class was generated from the following file:

- Runtime/Utility/NavAreaBakingUtility.cs

6.35 Infohazard.HyperNav.NavAreaBase Class Reference

Base class for areas in which HyperNav pathfinding can occur.

Classes

- class PropNames

  This is used to refer to the names of private fields in this class from a custom Editor.

Public Member Functions

- void UpdateTransform ()

  Update the native data of this NavArea.

Protected Member Functions

- virtual void OnEnable ()

  Register this area in the Instances dictionary and perform initialization.
- virtual void OnDisable ()

  Remove this area from the Instances dictionary.
- virtual void Reset ()

  Reset certain properties to the first NavArea on the object.
- virtual void OnDestroy ()

  Dispose native-side data for this area.
- virtual void Update ()

  Update UniqueID in editor, and check movement.

Properties

- Bounds Bounds `[get, set]`

  The boundaries of the area.
- abstract NavLayer Layer `[get, set]`

  The layer this area exists on.
- INavAreaData INavArea. Data `[get]`
- Transform Transform `[get]`
- IReadOnlyList< NavExternalLinkData > ExternalLinks `[get]`

  The external links for this area.
- IReadOnlyList< SerializableRange > ExternalLinkRanges `[get]`

  The ranges of external links for this area (one element per region).
- ulong DataVersionForExternalLinks `[get, protected set]`

  The version of the data when the external links were last updated.
- long InstanceID `[get]`

  The unique ID for this area to identify it in pathfinding jobs and serialized data.
- bool RandomInstanceID `[get, set]`

  Whether to generate a random instance ID on awake (use if instantiating dynamically).
- bool AutoDetectMovement `[get, set]`

  Whether to automatically update native data if the area moves.
- abstract NavAreaBaseSettings BaseSettings `[get]`

  Base settings for the area, whether shared or instance. The returned object, while mutable, should not be modified directly as it may reference a ScriptableObject.
- abstract Type SettingsAssetType `[get]`

  Type of settings asset for this area (should inherit from NavAreaBaseSettings).
- bool UseStartLocations `[get, set]`

  Whether only regions connected to certain locations are considered valid.
- IReadOnlyList< Vector3 > StartLocations `[get, set]`

  If _useStartLocations is true, which start locations to use.
- int VisualizationSoloRegion `[get, set]`

  If set, only this region will be included in the visualization mesh.
- bool VisualizeNeighbors `[get, set]`

  Whether to show the connections of a selected region in the scene view.
- int VisualizeNeighborsRegion `[get, set]`

  If _visualizeNeighbors is true, which region to visualize in the scene view.
- bool VisualizeExternalLinks `[get, set]`

  Whether to show the external links.
- bool ShowVertexNumbers `[get, set]`

Whether to show the vertex numbers of the preview mesh in the scene view (for debugging).

- float ShowVertexNumbersRange `[get, set]`

  Max distance from the camera at which vertex numbers will be shown.

- bool VisualizeRegionBounds `[get, set]`

  Whether to visualize the bounds of each region in the scene view.

- bool VisualizeVoxelQueries `[get, set]`

  Whether to visualize the queries that are performed for a voxel when baking.

- Mesh PreviewMesh `[get, set]`

  (Editor Only) Preview mesh that is rendered to visualize the area.

- Material[ ] PreviewMaterials `[get, set]`

  (Editor Only) List of materials to use for drawing the preview mesh.

- abstract bool IsNativeDataCreated `[get]`

  Returns whether the native data for this area has been created.

6.35.1 Detailed Description

Base class for areas in which HyperNav pathfinding can occur.

6.35.2 Member Function Documentation

6.35.2.1 OnDestroy() `virtual void Infohazard.HyperNav.NavAreaBase.OnDestroy ( ) [protected], [virtual]`

Dispose native-side data for this area.

6.35.2.2 OnDisable() `virtual void Infohazard.HyperNav.NavAreaBase.OnDisable ( ) [protected], [virtual]`

Remove this area from the Instances dictionary.

6.35.2.3 OnEnable() `virtual void Infohazard.HyperNav.NavAreaBase.OnEnable ( ) [protected], [virtual]`

Register this area in the Instances dictionary and perform initialization.

6.35.2.4 Reset() `virtual void Infohazard.HyperNav.NavAreaBase.Reset ( ) [protected], [virtual]`

Reset certain properties to the first NavArea on the object.

Reimplemented in Infohazard.HyperNav.NavArea< TArea, TData, TNativeData, TPointers, TSettings >.

6.35.2.5 Update() `virtual void Infohazard.HyperNav.NavAreaBase.Update ( ) [protected], [virtual]`

Update UniqueID in editor, and check movement.

6.35.2.6 UpdateTransform() `void Infohazard.HyperNav.NavAreaBase.UpdateTransform ( )`

Update the native data of this NavArea.

This is called automatically if AutoDetectMovement is enabled.

6.35.3 Property Documentation

6.35.3.1 AutoDetectMovement `bool Infohazard.HyperNav.NavAreaBase.AutoDetectMovement [get], [set]`

Whether to automatically update native data if the area moves.

Note that if this is true and the area moves every frame, pathfinding will never be able to occur.

6.35.3.2 BaseSettings `abstract NavAreaBaseSettings Infohazard.HyperNav.NavAreaBase.BaseSettings [get]`

Base settings for the area, whether shared or instance. The returned object, while mutable, should not be modified directly as it may reference a ScriptableObject.

6.35.3.3 Bounds `Bounds Infohazard.HyperNav.NavAreaBase.Bounds [get], [set]`

The boundaries of the area.

This cannot be set while the game is running.

6.35.3.4 Data `INavAreaData INavArea. Infohazard.HyperNav.NavAreaBase.Data [get], [private]`

6.35.3.5 DataVersionForExternalLinks `ulong Infohazard.HyperNav.NavAreaBase.DataVersionForExternalLinks [get], [protected set]`

The version of the data when the external links were last updated.

6.35.3.6 ExternalLinkRanges

`IReadOnlyList<SerializableRange> Infohazard.HyperNav.NavArea↩ Base.ExternalLinkRanges [get]`

The ranges of external links for this area (one element per region).

6.35.3.7 ExternalLinks

`IReadOnlyList<NavExternalLinkData> Infohazard.HyperNav.NavAreaBase.↩ ExternalLinks [get]`

The external links for this area.

6.35.3.8 InstanceID

`long Infohazard.HyperNav.NavAreaBase.InstanceID [get]`

The unique ID for this area to identify it in pathfinding jobs and serialized data.

6.35.3.9 IsNativeDataCreated

`abstract bool Infohazard.HyperNav.NavAreaBase.IsNativeDataCreated [get]`

Returns whether the native data for this area has been created.

6.35.3.10 Layer

`abstract NavLayer Infohazard.HyperNav.NavAreaBase.Layer [get], [set]`

The layer this area exists on.

6.35.3.11 PreviewMaterials

`Material [] Infohazard.HyperNav.NavAreaBase.PreviewMaterials [get], [set]`

(Editor Only) List of materials to use for drawing the preview mesh.

These should be references to assets so that they don't need to be destroyed.

6.35.3.12 PreviewMesh

`Mesh Infohazard.HyperNav.NavAreaBase.PreviewMesh [get], [set]`

(Editor Only) Preview mesh that is rendered to visualize the area.

When set, the old mesh will be destroyed. The supplied mesh will have its HideFlags set to HideAndDontSave, meaning it will not be saved with the scene, but will not be destroyed except manually here. Therefore it is important that this value be set to null when the NavArea is destroyed, in order to avoid leaking memory.

6.35.3.13 RandomInstanceID

`bool Infohazard.HyperNav.NavAreaBase.RandomInstanceID [get], [set]`

Whether to generate a random instance ID on awake (use if instantiating dynamically).

6.35.3.14 SettingsAssetType

`abstract Type Infohazard.HyperNav.NavAreaBase.SettingsAssetType [get]`

Type of settings asset for this area (should inherit from NavAreaBaseSettings).

6.35.3.15 ShowVertexNumbers

`bool Infohazard.HyperNav.NavAreaBase.ShowVertexNumbers [get], [set]`

Whether to show the vertex numbers of the preview mesh in the scene view (for debugging).

6.35.3.16 ShowVertexNumbersRange

`float Infohazard.HyperNav.NavAreaBase.ShowVertexNumbersRange [get], [set]`

Max distance from the camera at which vertex numbers will be shown.

6.35.3.17 StartLocations

`IReadOnlyList<Vector3> Infohazard.HyperNav.NavAreaBase.StartLocations [get], [set]`

If _useStartLocations is true, which start locations to use.

6.35.3.18 Transform

`Transform Infohazard.HyperNav.NavAreaBase.Transform [get]`

6.35.3.19 UseStartLocations

`bool Infohazard.HyperNav.NavAreaBase.UseStartLocations [get], [set]`

Whether only regions connected to certain locations are considered valid.

This can be used to exclude certain regions from an area, such as regions that are outside reachable area.

6.35.3.20 VisualizationSoloRegion

`int Infohazard.HyperNav.NavAreaBase.VisualizationSoloRegion [get], [set]`

If set, only this region will be included in the visualization mesh.

6.35.3.21 VisualizeExternalLinks `bool Infohazard.HyperNav.NavAreaBase.VisualizeExternalLinks [get], [set]`

Whether to show the external links.

6.35.3.22 VisualizeNeighbors `bool Infohazard.HyperNav.NavAreaBase.VisualizeNeighbors [get], [set]`

Whether to show the connections of a selected region in the scene view.

6.35.3.23 VisualizeNeighborsRegion `int Infohazard.HyperNav.NavAreaBase.VisualizeNeighbors↩ Region [get], [set]`

If _visualizeNeighbors is true, which region to visualize in the scene view.

6.35.3.24 VisualizeRegionBounds `bool Infohazard.HyperNav.NavAreaBase.VisualizeRegionBounds [get], [set]`

Whether to visualize the bounds of each region in the scene view.

6.35.3.25 VisualizeVoxelQueries `bool Infohazard.HyperNav.NavAreaBase.VisualizeVoxelQueries [get], [set]`

Whether to visualize the queries that are performed for a voxel when baking.

The documentation for this class was generated from the following file:

- Runtime/NavArea.cs

6.36 Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf > Class Template Reference

Base settings data for NavAreas.

Classes

- class PropNames

  This is used to refer to the names of private fields in this class from a custom Editor.

Public Member Functions

- virtual TSelf Clone ()

  Clone this settings object.

Properties

- float VoxelSize `[get, set]`

  The voxel size of this area, which determines the precision but also baking cost.
- float MaxAgentRadius `[get, set]`

  The maximum size of agents using this area.
- NavLayer Layer `[get, set]`

  Layer this area exists in.
- float MaxExternalLinkDistanceToVolume `[get, set]`

  The maximum distance that external links can extend to other volumes.
- float MaxExternalLinkDistanceToSurface `[get, set]`

  The maximum distance that external links can extend to other surfaces.
- NavLayerMask ExternalLinkTargetLayers `[get, set]`

  Layers that external links from this area can connect to.
- LayerMask BlockingLayers `[get, set]`

  Which layers are considered impassible for pathfinding.
- bool StaticOnly `[get, set]`

  Whether only static objects should be included in the baked data.

6.36.1 Detailed Description

Base settings data for NavAreas.

Extension with self-cloning capabilities.

Template Parameters

TSelf	Type of self.

Type Constraints

TSelf* : *NavAreaBaseSettings

TSelf* : *new()

6.36.2 Member Function Documentation

6.36.2.1 Clone() `virtual TSelf Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf >.Clone ( ) [virtual]`

Clone this settings object.

Returns

A clone of this settings object.

Reimplemented in Infohazard.HyperNav.Settings.NavSurfaceSettings, and Infohazard.HyperNav.Settings.NavVolumeSettings.

6.36.3 Property Documentation

6.36.3.1 BlockingLayers `LayerMask Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf >.BlockingLayers [get], [set]`

Which layers are considered impassible for pathfinding.

6.36.3.2 ExternalLinkTargetLayers `NavLayerMask Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf >.ExternalLinkTargetLayers [get], [set]`

Layers that external links from this area can connect to.

6.36.3.3 Layer `NavLayer Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf >.Layer [get], [set]`

Layer this area exists in.

6.36.3.4 MaxAgentRadius `float Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf >.MaxAgentRadius [get], [set]`

The maximum size of agents using this area.

6.36.3.5 MaxExternalLinkDistanceToSurface `float Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf >.MaxExternalLinkDistanceToSurface [get], [set]`

The maximum distance that external links can extend to other surfaces.

6.36.3.6 MaxExternalLinkDistanceToVolume `float Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf >.MaxExternalLinkDistanceToVolume [get], [set]`

The maximum distance that external links can extend to other volumes.

6.36.3.7 StaticOnly `bool Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf >.StaticOnly [get], [set]`

Whether only static objects should be included in the baked data.

6.36.3.8 VoxelSize `float Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf >.VoxelSize [get], [set]`

The voxel size of this area, which determines the precision but also baking cost.

The documentation for this class was generated from the following file:

- Runtime/Settings/NavAreaBaseSettings.cs

6.37 Infohazard.HyperNav.Settings.NavAreaBaseSettingsAsset< TData > Class Template Reference

Base class for all NavArea settings assets.

Classes

- class PropNames

  This is used to refer to the names of private fields in this class from a custom Editor.

Properties

- TData Data `[get]`

  The settings data.

6.37.1 Detailed Description

Base class for all NavArea settings assets.

Generic base class for NavArea settings assets.

Template Parameters

TData	Type of the settings data.

Type Constraints

TData* : *NavAreaBaseSettings

6.37.2 Property Documentation

6.37.2.1 Data `TData Infohazard.HyperNav.Settings.NavAreaBaseSettingsAsset< TData >.Data [get]`

The settings data.

The documentation for this class was generated from the following file:

- Runtime/Settings/NavAreaBaseSettings.cs

6.38 Infohazard.HyperNav.NavAreaExternalLinkUpdate Class Reference

Utility for updating external links for nav areas.

Static Public Member Functions

- static async UniTask GenerateExternalLinks (IReadOnlyList< INavArea > areas, bool updateSerializedData, bool keepLinksToUnloadedScenes, int maxCompletionFrames=3)

  Generate external links for the given areas. Can be used at runtime to support dynamically created links.

6.38.1 Detailed Description

Utility for updating external links for nav areas.

6.38.2 Member Function Documentation

6.38.2.1 GenerateExternalLinks() `static async UniTask Infohazard.HyperNav.NavAreaExternalLinkUpdate.GenerateExternalLinks (`

```
IReadOnlyList< INavArea > areas,
bool updateSerializedData,
bool keepLinksToUnloadedScenes,
int maxCompletionFrames = 3 )  [static]
```

Generate external links for the given areas. Can be used at runtime to support dynamically created links.

Parameters

areas	Areas to generate external links for. Regardless of this parameter, all loaded areas will be considered as destinations for external links.
updateSerializedData	Whether to update serialized data for the areas. If this is false, GC allocations will be avoided, but the data will not be saved.
keepLinksToUnloadedScenes	If true, links to areas in unloaded scenes will be kept.
maxCompletionFrames	Maximum number of frames to wait for job completion. -1 for no limit.

Exceptions

OperationCanceledException	Thrown if the job was cancelled due to data changing.

The documentation for this class was generated from the following file:

- Runtime/Utility/NavAreaExternalLinkUpdate.cs

6.39 Infohazard.HyperNav.Editor.NavAreaMigrator Class Reference

Handles migration of NavArea data to new versions.

6.39.1 Detailed Description

Handles migration of NavArea data to new versions.

The documentation for this class was generated from the following file:

- Editor/NavAreaMigrator.cs

6.40 Infohazard.HyperNav.Editor.NavEditorUtility Class Reference

Utility functions used internally, but you can use them too, I mean I'm not your boss.

Static Public Member Functions

- static void ExportPreviewMesh (Mesh mesh)

  Export a mesh as an OBJ file.
- static void CreateVolume ()

  Menu item to create a new NavVolume.
- static void CreateSurface ()

  Menu item to create a new NavSurface.
- static void CreateVolumeAndSurface ()

  Menu item to create a new NavVolume + NavSurface object.
- static void CreatePathfinder ()

  Menu item to create a new NavPathfinder.
- static void CreateAvoidanceManager ()

  Menu item to create a new AvoidanceManager.
- static void CreateManualLink ()

  Menu item to create a new ManualNavLink.

6.40.1 Detailed Description

Utility functions used internally, but you can use them too, I mean I'm not your boss.

6.40.2 Member Function Documentation

6.40.2.1 CreateAvoidanceManager() `static void Infohazard.HyperNav.Editor.NavEditorUtility.↩ CreateAvoidanceManager ( ) [static]`

Menu item to create a new AvoidanceManager.

6.40.2.2 CreateManualLink() `static void Infohazard.HyperNav.Editor.NavEditorUtility.Create↩ ManualLink ( ) [static]`

Menu item to create a new ManualNavLink.

6.40.2.3 CreatePathfinder() `static void Infohazard.HyperNav.Editor.NavEditorUtility.Create↩ Pathfinder ( ) [static]`

Menu item to create a new NavPathfinder.

6.40.2.4 CreateSurface() `static void Infohazard.HyperNav.Editor.NavEditorUtility.CreateSurface ( ) [static]`

Menu item to create a new NavSurface.

6.40.2.5 CreateVolume() `static void Infohazard.HyperNav.Editor.NavEditorUtility.CreateVolume ( ) [static]`

Menu item to create a new NavVolume.

6.40.2.6 CreateVolumeAndSurface() `static void Infohazard.HyperNav.Editor.NavEditorUtility.↩ CreateVolumeAndSurface ( ) [static]`

Menu item to create a new NavVolume + NavSurface object.

6.40.2.7 ExportPreviewMesh() `static void Infohazard.HyperNav.Editor.NavEditorUtility.Export↩ PreviewMesh ( Mesh mesh ) [static]`

Export a mesh as an OBJ file.

This is the most basic export possible, and should not be used for actual art. It is only used to more closely inspect a preview mesh in a modeling application. The material names will be included in the OBJ, but the MTL file is not created. Normals are also not included.

Parameters

mesh	The mesh to export.

The documentation for this class was generated from the following file:

- Editor/NavEditorUtility.cs

6.41 Infohazard.HyperNav.NavExternalLinkData Class Reference

A connection from one region to another region in another volume.

Public Member Functions

- NavExternalLinkData (in NativeNavExternalLinkData nativeData, string connectedScenePath)
  Create a new NavExternalLinkData from the given native data.
- NativeNavExternalLinkData ToNativeData ()
  Convert to a native representation.

Properties

- long ConnectedAreaID `[get]`
  The NavVolume.InstanceID of the connected volume.
- NavAreaTypes ConnectedAreaType `[get]`
  The type of the connected area.
- int ConnectedRegionID `[get]`
  The ID of the connected region.
- Vector3 FromPosition `[get]`
  The position at which the connection originates (local space).
- Vector3 ToPosition `[get]`
  The position at which the connection ends (local space).
- string ConnectedScenePath `[get]`
  Scene path of the connected volume.
- long ManualLinkID `[get]`
  The unique ID of the ManualNavLink that created this link, or 0.

6.41.1 Detailed Description

A connection from one region to another region in another volume.

6.41.2 Constructor & Destructor Documentation

6.41.2.1 NavExternalLinkData() `Infohazard.HyperNav.NavExternalLinkData.NavExternalLinkData (`

```
in NativeNavExternalLinkData nativeData,
string connectedScenePath )
```

Create a new NavExternalLinkData from the given native data.

Parameters

nativeData	Native data to copy.
connectedScenePath	Scene path of the connected volume.

Returns

The created NavExternalLinkData.

6.41.3 Member Function Documentation

6.41.3.1 ToNativeData() `NativeNavExternalLinkData Infohazard.HyperNav.NavExternalLinkData.ToNativeData ( )`

Convert to a native representation.

Returns

Created native data.

6.41.4 Property Documentation

6.41.4.1 ConnectedAreaID `long Infohazard.HyperNav.NavExternalLinkData.ConnectedAreaID [get]`

The NavVolume.InstanceID of the connected volume.

6.41.4.2 ConnectedAreaType `NavAreaTypes Infohazard.HyperNav.NavExternalLinkData.ConnectedAreaType [get]`

The type of the connected area.

6.41.4.3 ConnectedRegionID `int Infohazard.HyperNav.NavExternalLinkData.ConnectedRegionID [get]`

The ID of the connected region.

6.41.4.4 ConnectedScenePath `string Infohazard.HyperNav.NavExternalLinkData.ConnectedScenePath [get]`

Scene path of the connected volume.

6.41.4.5 FromPosition `Vector3 Infohazard.HyperNav.NavExternalLinkData.FromPosition [get]`

The position at which the connection originates (local space).

6.41.4.6 ManualLinkID `long Infohazard.HyperNav.NavExternalLinkData.ManualLinkID [get]`

The unique ID of the ManualNavLink that created this link, or 0.

6.41.4.7 ToPosition `Vector3 Infohazard.HyperNav.NavExternalLinkData.ToPosition [get]`

The position at which the connection ends (local space).

The documentation for this class was generated from the following file:

- Runtime/NavExternalLinkData.cs

6.42 Infohazard.HyperNav.NavInternalLinkData Class Reference

A connection from one region to another region in the same volume.

Public Member Functions

- NavInternalLinkData ()
  Default constructor for serialization.
- NavInternalLinkData (int connectedRegionID, SerializableRange vertexRange, SerializableRange edge↩ Range, SerializableRange triangleRange)
  Create a new NavInternalLinkData with the given properties.
- NavInternalLinkData (in NativeNavVolumeInternalLinkData data)
  Create a new NavInternalLinkData with the given native data.
- NativeNavVolumeInternalLinkData ToNativeData ()
  Convert to a native representation.

Properties

- int ConnectedRegionID `[get]`

  The ID of the connected region.
- SerializableRange VertexRange `[get]`

  Range of the link's indices in the volume's NavVolumeData.LinkVertices array.
- SerializableRange EdgeRange `[get]`

  Range of the link's indices in the volume's NavVolumeData.LinkEdges array.
- SerializableRange TriangleRange `[get]`

  Range of the link's indices in the volume's NavVolumeData.LinkTriangles array.

6.42.1 Detailed Description

A connection from one region to another region in the same volume.

6.42.2 Constructor & Destructor Documentation

6.42.2.1 NavInternalLinkData() [1/3] Infohazard.HyperNav.NavInternalLinkData.NavInternalLinkData ( )

Default constructor for serialization.

6.42.2.2 NavInternalLinkData() [2/3] Infohazard.HyperNav.NavInternalLinkData.NavInternalLinkData (

```
int connectedRegionID,
SerializableRange vertexRange,
SerializableRange edgeRange,
SerializableRange triangleRange )
```

Create a new NavInternalLinkData with the given properties.

Parameters

connectedRegionID	ID of the connected region.
vertexRange	Range of the link's indices.
edgeRange	Range of the link's indices.
triangleRange	Range of the link's indices.

6.42.2.3 NavInternalLinkData() [3/3] Infohazard.HyperNav.NavInternalLinkData.NavInternalLinkData (

```
in NativeNavVolumeInternalLinkData data )
```

Create a new NavInternalLinkData with the given native data.

Parameters

data	Native data to copy.

6.42.3 Member Function Documentation

6.42.3.1 ToNativeData() `NativeNavVolumeInternalLinkData Infohazard.HyperNav.NavInternalLink↩ Data.ToNativeData ( )`

Convert to a native representation.

Returns

Created native data.

6.42.4 Property Documentation

6.42.4.1 ConnectedRegionID `int Infohazard.HyperNav.NavInternalLinkData.ConnectedRegionID [get]`

The ID of the connected region.

6.42.4.2 EdgeRange `SerializableRange Infohazard.HyperNav.NavInternalLinkData.EdgeRange [get]`

Range of the link's indices in the volume's NavVolumeData.LinkEdges array.

6.42.4.3 TriangleRange `SerializableRange Infohazard.HyperNav.NavInternalLinkData.TriangleRange [get]`

Range of the link's indices in the volume's NavVolumeData.LinkTriangles array.

6.42.4.4 VertexRange `SerializableRange Infohazard.HyperNav.NavInternalLinkData.VertexRange [get]`

Range of the link's indices in the volume's NavVolumeData.LinkVertices array.

The documentation for this class was generated from the following file:

- Runtime/NavVolumeData.cs

6.43 Infohazard.HyperNav.NavLayer Struct Reference

A single NavLayer. May have the value 0 to 31, or -1 for invalid.

Public Member Functions

- readonly NavLayerMask AsMask ()

  Get a mask that represents this single layer.
- readonly string GetName ()

  Get the name of this layer from the settings.
- readonly override string ToString ()
- readonly bool Equals (NavLayer other)
- readonly override bool Equals (object obj)
- readonly override int GetHashCode ()

Static Public Member Functions

- static NavLayer Get (string name)

  Get the NavLayer corresponding to the given name in the settings.
- static bool operator== (NavLayer layer1, NavLayer layer2)

  Equality check.
- static bool operator!= (NavLayer layer1, NavLayer layer2)

  Inequality check.
- static bool TryGet (string name, out NavLayer layer)

  Try to get the NavLayer corresponding to the given name in the settings.

6.43.1 Detailed Description

A single NavLayer. May have the value 0 to 31, or -1 for invalid.

6.43.2 Member Function Documentation

6.43.2.1 AsMask() `readonly NavLayerMask Infohazard.HyperNav.NavLayer.AsMask ( )`

Get a mask that represents this single layer.

Returns

A mask that represents this layer.

6.43.2.2 Equals() [1/2] `readonly bool Infohazard.HyperNav.NavLayer.Equals ( NavLayer other )`

6.43.2.3 Equals() [2/2] readonly override bool Infohazard.HyperNav.NavLayer.Equals (object *obj*)

6.43.2.4 Get() static NavLayer Infohazard.HyperNav.NavLayer.Get (string *name*) [static]

Get the NavLayer corresponding to the given name in the settings.

Parameters

name	The name of the layer.

Returns

The layer with that name, or None if invalid.

6.43.2.5 GetHashCode() `readonly override int Infohazard.HyperNav.NavLayer.GetHashCode ( )`

6.43.2.6 GetName() `readonly string Infohazard.HyperNav.NavLayer.GetName ( )`

Get the name of this layer from the settings.

Returns

The name of the layer, or an empty string if it is invalid.

6.43.2.7 operator"!=() `static bool Infohazard.HyperNav.NavLayer.operator!= ( NavLayer layer1, NavLayer layer2 ) [static]`

Inequality check.

6.43.2.8 operator==() `static bool Infohazard.HyperNav.NavLayer.operator== ( NavLayer layer1, NavLayer layer2 ) [static]`

Equality check.

6.43.2.9 ToString() `readonly override string Infohazard.HyperNav.NavLayer.ToString ( )`

6.43.2.10 TryGet() `static bool Infohazard.HyperNav.NavLayer.TryGet ( string name, out NavLayer layer ) [static]`

Try to get the NavLayer corresponding to the given name in the settings.

Parameters

name	The name of the layer.
layer	The layer with that name, or None if invalid.

Returns

Whether the layer exists.

The documentation for this struct was generated from the following file:

- Runtime/NavLayer.cs

6.44 Infohazard.HyperNav.NavLayerMask Struct Reference

A mask of NavLayers with one bit for each possible layer.

Public Member Functions

- readonly bool Contains (int layer)

  Returns true if the mask contains the given layer index.
- readonly Enumerator GetEnumerator ()

  Non-allocating enumerator over this mask.
- IEnumerator< NavLayer > IEnumerable< NavLayer >. GetEnumerator ()
- IEnumerator IEnumerable. GetEnumerator ()
- readonly override string ToString ()
- readonly bool Equals (NavLayerMask other)
- readonly override bool Equals (object obj)
- readonly override int GetHashCode ()

Static Public Member Functions

- static NavLayerMask Combine (NavLayer layer1, NavLayer layer2)

  Get a mask representing the combination of two layers.
- static NavLayerMask Combine (NavLayer layer1, NavLayer layer2, NavLayer layer3)

  Get a mask representing the combination of three layers.
- static NavLayerMask Combine (NavLayer layer1, NavLayer layer2, NavLayer layer3, NavLayer layer4)

  Get a mask representing the combination of four layers.
- static bool operator== (NavLayerMask layer1, NavLayerMask layer2)

  Equality check.
- static bool operator!= (NavLayerMask layer1, NavLayerMask layer2)

  Inequality check.
- static implicit operator NavLayerMask (uint value)

  Convert from uint.
- static implicit operator NavLayerMask (int value)

  Convert from int.
- static implicit operator uint (NavLayerMask i)

  Convert to uint.

- static implicit operator int (NavLayerMask i)

  Convert to int.
- static NavLayerMask operator| (NavLayerMask lhs, NavLayerMask rhs)

  Apply bitwise OR operator to two NavLayerMasks.
- static NavLayerMask operator| (NavLayerMask lhs, uint rhs)

  Apply bitwise OR operator to a NavLayerMask and an unsigned integer.
- static uint operator| (uint lhs, NavLayerMask rhs)

  Apply bitwise OR operator to an unsigned integer and a NavLayerMask.
- static NavLayerMask operator& (NavLayerMask lhs, NavLayerMask rhs)

  Apply bitwise AND operator to two NavLayerMasks.
- static NavLayerMask operator& (NavLayerMask lhs, uint rhs)

  Apply bitwise AND operator to a NavLayerMask and an unsigned integer.
- static uint operator& (uint lhs, NavLayerMask rhs)

  Apply bitwise AND operator to an unsigned integer and a NavLayerMask.
- static NavLayerMask operator^ (NavLayerMask lhs, NavLayerMask rhs)

  Apply bitwise XOR operator to two NavLayerMasks.
- static NavLayerMask operator^ (NavLayerMask lhs, uint rhs)

  Apply bitwise XOR operator to a NavLayerMask and an unsigned integer.
- static uint operator^ (uint lhs, NavLayerMask rhs)

  Apply bitwise XOR operator to an unsigned integer and a NavLayerMask.
- static NavLayerMask operator~ (NavLayerMask lhs)

  Invert a NavLayerMask.

Static Public Attributes

- static readonly NavLayerMask None = new(0)

  A mask that represents no layers.
- static readonly NavLayerMask All = new(0xFFFFFFFF)

  A mask that represents all possible layers.

Properties

- readonly bool IsEmpty `[get]`

  Whether this mask is empty.

6.44.1 Detailed Description

A mask of NavLayers with one bit for each possible layer.

6.44.2 Member Function Documentation

6.44.2.1 Combine() [1/3]

```
static NavLayerMask Infohazard.HyperNav.NavLayerMask.Combine (
        NavLayer layer1,
        NavLayer layer2 )  [static]
```

Get a mask representing the combination of two layers.

6.44.2.2 Combine() [2/3] static NavLayerMask Infohazard.HyperNav.NavLayerMask.Combine (NavLayer *layer1*, NavLayer *layer2*, NavLayer *layer3*) [static]

Get a mask representing the combination of three layers.

6.44.2.3 Combine() [3/3] static NavLayerMask Infohazard.HyperNav.NavLayerMask.Combine (NavLayer *layer1*, NavLayer *layer2*, NavLayer *layer3*, NavLayer *layer4*) [static]

Get a mask representing the combination of four layers.

6.44.2.4 Contains() readonly bool Infohazard.HyperNav.NavLayerMask.Contains (int *layer*)

Returns true if the mask contains the given layer index.

Parameters

layer	The index of the layer to check.

Returns

Whether the mask contains that value.

6.44.2.5 Equals() [1/2] readonly bool Infohazard.HyperNav.NavLayerMask.Equals (NavLayerMask *other*)

6.44.2.6 Equals() [2/2] readonly override bool Infohazard.HyperNav.NavLayerMask.Equals (object *obj*)

6.44.2.7 GetEnumerator() [1/3] readonly Enumerator Infohazard.HyperNav.NavLayerMask.GetEnumerator ( )

Non-allocating enumerator over this mask.

6.44.2.8 GetEnumerator() [2/3]

IEnumerator< NavLayer > IEnumerable< NavLayer >. Infohazard.HyperNav.NavLayerMask.GetEnumerator ( )

6.44.2.9 GetEnumerator() [3/3]

IEnumerator IEnumerable. Infohazard.HyperNav.NavLayerMask.GetEnumerator ( )

6.44.2.10 GetHashCode()

readonly override int Infohazard.HyperNav.NavLayerMask.GetHashCode ( )

6.44.2.11 operator int()

static implicit Infohazard.HyperNav.NavLayerMask.operator int (NavLayerMask *i*) [static]

Convert to int.

6.44.2.12 operator NavLayerMask() [1/2]

static implicit Infohazard.HyperNav.NavLayerMask.operator NavLayerMask (int *value*) [static]

Convert from int.

6.44.2.13 operator NavLayerMask() [2/2]

static implicit Infohazard.HyperNav.NavLayerMask.operator NavLayerMask (uint *value*) [static]

Convert from uint.

6.44.2.14 operator uint()

static implicit Infohazard.HyperNav.NavLayerMask.operator uint (NavLayerMask *i*) [static]

Convert to uint.

6.44.2.15 operator"!=()

static bool Infohazard.HyperNav.NavLayerMask.operator!= (NavLayerMask *layer1*, NavLayerMask *layer2*) [static]

Inequality check.

6.44.2.16 operator&() [1/3] static NavLayerMask Infohazard.HyperNav.NavLayerMask.operator& (NavLayerMask *lhs*, NavLayerMask *rhs*) [static]

Apply bitwise AND operator to two NavLayerMasks.

6.44.2.17 operator&() [2/3] static NavLayerMask Infohazard.HyperNav.NavLayerMask.operator& (NavLayerMask *lhs*, uint *rhs*) [static]

Apply bitwise AND operator to a NavLayerMask and an unsigned integer.

6.44.2.18 operator&() [3/3] static uint Infohazard.HyperNav.NavLayerMask.operator& (uint *lhs*, NavLayerMask *rhs*) [static]

Apply bitwise AND operator to an unsigned integer and a NavLayerMask.

6.44.2.19 operator==() static bool Infohazard.HyperNav.NavLayerMask.operator== (NavLayerMask *layer1*, NavLayerMask *layer2*) [static]

Equality check.

6.44.2.20 operator^() [1/3] static NavLayerMask Infohazard.HyperNav.NavLayerMask.operator^ (NavLayerMask *lhs*, NavLayerMask *rhs*) [static]

Apply bitwise XOR operator to two NavLayerMasks.

6.44.2.21 operator^() [2/3] static NavLayerMask Infohazard.HyperNav.NavLayerMask.operator^ (NavLayerMask *lhs*, uint *rhs*) [static]

Apply bitwise XOR operator to a NavLayerMask and an unsigned integer.

6.44.2.22 operator^() [3/3]

```
static uint Infohazard.HyperNav.NavLayerMask.operator^ (
    uint lhs,
    NavLayerMask rhs )  [static]
```

Apply bitwise XOR operator to an unsigned integer and a NavLayerMask.

6.44.2.23 operator"|() [1/3]

```
static NavLayerMask Infohazard.HyperNav.NavLayerMask.operator| (
    NavLayerMask lhs,
    NavLayerMask rhs )  [static]
```

Apply bitwise OR operator to two NavLayerMasks.

6.44.2.24 operator"|() [2/3]

```
static NavLayerMask Infohazard.HyperNav.NavLayerMask.operator| (
    NavLayerMask lhs,
    uint rhs )  [static]
```

Apply bitwise OR operator to a NavLayerMask and an unsigned integer.

6.44.2.25 operator"|() [3/3]

```
static uint Infohazard.HyperNav.NavLayerMask.operator| (
    uint lhs,
    NavLayerMask rhs )  [static]
```

Apply bitwise OR operator to an unsigned integer and a NavLayerMask.

6.44.2.26 operator~()

```
static NavLayerMask Infohazard.HyperNav.NavLayerMask.operator~ (
    NavLayerMask lhs )  [static]
```

Invert a NavLayerMask.

6.44.2.27 ToString()

```
readonly override string Infohazard.HyperNav.NavLayerMask.ToString ( )
```

6.44.3 Member Data Documentation

6.44.3.1 All

```
readonly NavLayerMask Infohazard.HyperNav.NavLayerMask.All = new(0xFFFFFFFF)  [static]
```

A mask that represents all possible layers.

6.44.3.2 None `readonly NavLayerMask Infohazard.HyperNav.NavLayerMask.None = new(0) [static]`

A mask that represents no layers.

6.44.4 Property Documentation

6.44.4.1 IsEmpty `readonly bool Infohazard.HyperNav.NavLayerMask.IsEmpty [get]`

Whether this mask is empty.

The documentation for this struct was generated from the following file:

- Runtime/NavLayer.cs

6.45 Infohazard.HyperNav.Jobs.NavMultiRaycastJob Struct Reference

Job that performs multiple raycasts in one or more NavVolumes in parallel.

Public Member Functions

- void Execute (int index)

  Execute the job for the raycast at index.

Public Attributes

- NativeParallelHashMap< long, NativeNavVolumeData > Volumes

  All loaded volume data.
- NativeArray< NativeRaycastElement > Raycasts

  The raycasts to perform (results are stored in each element's NativeRaycastElement.OutDistance.

6.45.1 Detailed Description

Job that performs multiple raycasts in one or more NavVolumes in parallel.

6.45.2 Member Function Documentation

6.45.2.1 Execute() `void Infohazard.HyperNav.Jobs.NavMultiRaycastJob.Execute ( int index )`

Execute the job for the raycast at index.

Parameters

index	Index of the raycast to execute.

6.45.3 Member Data Documentation

6.45.3.1 Raycasts `NativeArray<NativeRaycastElement> Infohazard.HyperNav.Jobs.NavMultiRaycastJob.Raycasts`

The raycasts to perform (results are stored in each element's NativeRaycastElement.OutDistance.

6.45.3.2 Volumes `NativeParallelHashMap<long, NativeNavVolumeData> Infohazard.HyperNav.Jobs.NavMultiRaycastJob.Volumes`

All loaded volume data.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/NavRaycastJob.cs

6.46 Infohazard.HyperNav.NavPath Class Reference

A completed, valid path.

Public Member Functions

- void Dispose ()

  Dispose the path, returning it to an object pool.

Properties

- long ID = -1 `[get, set]`

  ID of the path.
- bool HasBeenDisposed = true `[get, set]`

  Whether the path has been disposed.
- Vector3 StartPos `[get, set]`

  The position where the path originates from.
- Vector3 EndPos `[get, set]`

  The destination of the path.
- NavSampleResult StartHit `[get, set]`

  The navigation query result at the start of the path.
- NavSampleResult EndHit `[get, set]`

  The navigation query result at the end of the path.
- NavPathfinder Pathfinder `[get, set]`

  The NavPathfinder that was used to calculate the path.
- List< NavWaypoint > InternalWaypoints = new List<NavWaypoint>() `[get]`

  Mutable list of waypoints of the path.
- IReadOnlyList< NavWaypoint > Waypoints `[get]`

  List of waypoints of the path.

6.46.1 Detailed Description

A completed, valid path.

6.46.2 Member Function Documentation

6.46.2.1 Dispose() `void Infohazard.HyperNav.NavPath.Dispose ( )`

Dispose the path, returning it to an object pool.

6.46.3 Property Documentation

6.46.3.1 EndHit `NavSampleResult Infohazard.HyperNav.NavPath.EndHit [get], [set]`

The navigation query result at the end of the path.

6.46.3.2 EndPos `Vector3 Infohazard.HyperNav.NavPath.EndPos [get], [set]`

The destination of the path.

6.46.3.3 HasBeenDisposed `bool Infohazard.HyperNav.NavPath.HasBeenDisposed = true [get], [set]`

Whether the path has been disposed.

6.46.3.4 ID `long Infohazard.HyperNav.NavPath.ID = -1 [get], [set]`

ID of the path.

6.46.3.5 InternalWaypoints `List<NavWaypoint> Infohazard.HyperNav.NavPath.InternalWaypoints = new List<NavWaypoint>() [get], [package]`

Mutable list of waypoints of the path.

6.46.3.6 Pathfinder `NavPathfinder Infohazard.HyperNav.NavPath.Pathfinder [get], [set], [package]`

The NavPathfinder that was used to calculate the path.

6.46.3.7 StartHit `NavSampleResult Infohazard.HyperNav.NavPath.StartHit [get], [set]`

The navigation query result at the start of the path.

6.46.3.8 StartPos `Vector3 Infohazard.HyperNav.NavPath.StartPos [get], [set]`

The position where the path originates from.

6.46.3.9 Waypoints `IReadOnlyList<NavWaypoint> Infohazard.HyperNav.NavPath.Waypoints [get]`

List of waypoints of the path.

The documentation for this class was generated from the following file:

- Runtime/NavPathfinder.cs

6.47 Infohazard.HyperNav.NavPathfinder Class Reference

A script used to calculate HyperNav paths.

Classes

- class PropNames

  This is used to refer to the names of private fields in this class from a custom Editor.

Public Member Functions

- long FindPath (NavSampleResult startHit, NavSampleResult endHit, Vector3 startPos, Vector3 endPos, NavPathfindingParams pathParams, HyperNavPathCallback receiver)

  Find a path between two already-calculated nav query results, and invoke the receiver when it is completed.
- void CancelPath (long id, bool logError=true)

  Cancel a pending path with the given ID.

Protected Member Functions

- virtual void OnEnable ()

  If IsMainInstance is true, set MainInstance or log an error if it is already set.
- virtual void OnDisable ()

  Dispose the pools of pending paths and completed paths, and all memory allocated for pathfinding jobs.
- virtual void Update ()

  If mode is JobThread, check job completion. If mode is MainThreadAsynchronous, perform pathfinding work.

Package Functions

- void DisposePath (NavPath path)

  Called by NavPath.Dispose.

Properties

- bool IsMainInstance `[get, set]`

  Whether to set NavPathfinder.Instance to this instance.
- NavPathfindingMode PathfindingMode `[get, set]`

  The mode to use for calculating paths.
- int MaxConcurrentJobs `[get, set]`

  (JobThread Mode ONLY) Maximum number of pathfinding jobs that can be actively running at once.
- int MaxCompletionFrames `[get, set]`

  (JobThread Mode ONLY) Maximum number of frames a job can take before the main thread must wait for it.
- int PathTighteningIterations `[get, set]`

  Number of times to apply the path tightening operation to attempt to shorten the path.
- static NavPathfinder MainInstance `[get, private set]`

  The main instance, which should be used in most situations.

Private Member Functions

- void LateUpdate ()

  Move paths from the pending queue and start executing them.

6.47.1 Detailed Description

A script used to calculate HyperNav paths.

Can be used as a singleton, or you can have more than one if needed.

6.47.2 Member Function Documentation

6.47.2.1 CancelPath()

```
void Infohazard.HyperNav.NavPathfinder.CancelPath (
        long id,
        bool logError = true )
```

Cancel a pending path with the given ID.

If the mode is set to JobThread and the requested path is already executing, the actual work thread cannot be cancelled. However, this will still remove the receiver, so no matter what that will not be called for the path.

Parameters

id	The path ID to cancel.
logError	Whether to log an error if the path is not running.

6.47.2.2 DisposePath() `void Infohazard.HyperNav.NavPathfinder.DisposePath (`

```
NavPath path ) [package]
```

Called by NavPath.Dispose.

Parameters

path	The path to dispose.

6.47.2.3 FindPath() `long Infohazard.HyperNav.NavPathfinder.FindPath (`

```
NavSampleResult startHit,
NavSampleResult endHit,
Vector3 startPos,
Vector3 endPos,
NavPathfindingParams pathParams,
HyperNavPathCallback receiver )
```

Find a path between two already-calculated nav query results, and invoke the receiver when it is completed.

If pathfinding cannot occur, for example because there are no volumes, this method will return -1 and the receiver will not be invoked. If no path can be found, the receiver will be invoked with a null Path argument.

Parameters

startHit	Query result for the start of the path.
endHit	Query result for the end of the path.
startPos	Start position for the path.
endPos	Destination for the path.
pathParams	Parameters for pathfinding.
receiver	Callback to receive the path when it has been calculated.

Returns

The ID of the pending path, or -1 if pathfinding cannot occur.

6.47.2.4 LateUpdate() `void Infohazard.HyperNav.NavPathfinder.LateUpdate ( ) [private]`

Move paths from the pending queue and start executing them.

6.47.2.5 OnDisable() `virtual void Infohazard.HyperNav.NavPathfinder.OnDisable ( ) [protected], [virtual]`

Dispose the pools of pending paths and completed paths, and all memory allocated for pathfinding jobs.

6.47.2.6 OnEnable() `virtual void Infohazard.HyperNav.NavPathfinder.OnEnable ( ) [protected], [virtual]`

If IsMainInstance is true, set MainInstance or log an error if it is already set.

6.47.2.7 Update() `virtual void Infohazard.HyperNav.NavPathfinder.Update ( ) [protected], [virtual]`

If mode is JobThread, check job completion. If mode is MainThreadAsynchronous, perform pathfinding work.

6.47.3 Property Documentation

6.47.3.1 IsMainInstance `bool Infohazard.HyperNav.NavPathfinder.IsMainInstance [get], [set]`

Whether to set NavPathfinder.Instance to this instance.

This cannot be set while the game is running.

6.47.3.2 MainInstance `NavPathfinder Infohazard.HyperNav.NavPathfinder.MainInstance [static], [get], [private set]`

The main instance, which should be used in most situations.

If you need more than one NavPathfinder, you can use direct references to other instances with IsMainInstance set to false.

6.47.3.3 MaxCompletionFrames `int Infohazard.HyperNav.NavPathfinder.MaxCompletionFrames [get], [set]`

(JobThread Mode ONLY) Maximum number of frames a job can take before the main thread must wait for it.

Unity imposes a limit of 3 frames for faster TempJob allocations, so increasing this beyond 3 will slightly decrease memory performance. If a job takes longer than this and is forced to block the main thread, a warning will be logged showing you how long it blocked the main thread for. This cannot be set while any paths are executing.

6.47.3.4 MaxConcurrentJobs `int Infohazard.HyperNav.NavPathfinder.MaxConcurrentJobs [get], [set]`

(JobThread Mode ONLY) Maximum number of pathfinding jobs that can be actively running at once.

Requests beyond this number are queued. You should keep this number fairly low, as each job has the potential to take up a CPU thread.

6.47.3.5 PathfindingMode `NavPathfindingMode Infohazard.HyperNav.NavPathfinder.PathfindingMode [get], [set]`

The mode to use for calculating paths.

This cannot be set while the game is running.

6.47.3.6 PathTighteningIterations `int Infohazard.HyperNav.NavPathfinder.PathTighteningIterations [get], [set]`

Number of times to apply the path tightening operation to attempt to shorten the path.

The documentation for this class was generated from the following file:

- Runtime/NavPathfinder.cs

6.48 Infohazard.HyperNav.Jobs.NavPathJob Struct Reference

Job used to find a HyperNav path.

Public Member Functions

- void Execute ()

  Execute the pathfinding operation all the way through.

Public Attributes

- NativeParallelHashMap< long, NativeNavVolumeData > Volumes

  Map containing all loaded NavVolumes, keyed by their instance ID>
- NativeParallelHashMap< long, NativeNavSurfaceData > Surfaces

  Map containing all loaded NavSurfaces, keyed by their instance ID>
- NativeHashMap< long, bool > ManualLinksEnabled

  Map indicating whether manual links (keyed by their instance ID) are enabled.
- float4 StartPosition

  Position where the path starts (world space).
- NavSampleResult StartHit

  Nav query result where the path starts.
- NavSampleResult EndHit

  Nav query result where the path ends.
- NavAreaTypes AllowedAreaTypes

  Types of areas that the path can traverse.
- NavLayerMask AllowedLayers

  Layers that the path can traverse.
- float CostToChangeToVolume

  Allows adding an additional cost to changing from a surface to a volume.
- float CostToChangeToSurface

  Allows adding an additional cost to changing from a volume to a surface.
- float VolumeCostMultiplier

  Multiplier for the cost of traversing volume areas.
- float SurfaceCostMultiplier

  Multiplier for the cost of traversing surface areas.
- int PathTighteningIterations

  Number of times to apply the path tightening operation to attempt to shorten the path.
- NativeList< NativeNavWaypoint > OutPathWaypoints

  Used to return the result path (as a list of waypoints) back to managed code.

6.48.1 Detailed Description

Job used to find a HyperNav path.

6.48.2 Member Function Documentation

6.48.2.1 Execute() `void Infohazard.HyperNav.Jobs.NavPathJob.Execute ( )`

Execute the pathfinding operation all the way through.

6.48.3 Member Data Documentation

6.48.3.1 AllowedAreaTypes `NavAreaTypes Infohazard.HyperNav.Jobs.NavPathJob.AllowedAreaTypes`

Types of areas that the path can traverse.

6.48.3.2 AllowedLayers `NavLayerMask Infohazard.HyperNav.Jobs.NavPathJob.AllowedLayers`

Layers that the path can traverse.

6.48.3.3 CostToChangeToSurface `float Infohazard.HyperNav.Jobs.NavPathJob.CostToChangeTo↩Surface`

Allows adding an additional cost to changing from a volume to a surface.

6.48.3.4 CostToChangeToVolume `float Infohazard.HyperNav.Jobs.NavPathJob.CostToChangeToVolume`

Allows adding an additional cost to changing from a surface to a volume.

6.48.3.5 EndHit `NavSampleResult Infohazard.HyperNav.Jobs.NavPathJob.EndHit`

Nav query result where the path ends.

6.48.3.6 ManualLinksEnabled

`NativeHashMap<long, bool> Infohazard.HyperNav.Jobs.NavPathJob.↩ ManualLinksEnabled`

Map indicating whether manual links (keyed by their instance ID) are enabled.

6.48.3.7 OutPathWaypoints

`NativeList<NativeNavWaypoint> Infohazard.HyperNav.Jobs.NavPath↩ Job.OutPathWaypoints`

Used to return the result path (as a list of waypoints) back to managed code.

6.48.3.8 PathTighteningIterations

`int Infohazard.HyperNav.Jobs.NavPathJob.PathTighteningIterations`

Number of times to apply the path tightening operation to attempt to shorten the path.

6.48.3.9 StartHit

`NavSampleResult Infohazard.HyperNav.Jobs.NavPathJob.StartHit`

Nav query result where the path starts.

6.48.3.10 StartPosition

`float4 Infohazard.HyperNav.Jobs.NavPathJob.StartPosition`

Position where the path starts (world space).

6.48.3.11 SurfaceCostMultiplier

`float Infohazard.HyperNav.Jobs.NavPathJob.SurfaceCostMultiplier`

Multiplier for the cost of traversing surface areas.

6.48.3.12 Surfaces

`NativeParallelHashMap<long, NativeNavSurfaceData> Infohazard.HyperNav.↩ Jobs.NavPathJob.Surfaces`

Map containing all loaded NavSurfaces, keyed by their instance ID>

6.48.3.13 VolumeCostMultiplier

`float Infohazard.HyperNav.Jobs.NavPathJob.VolumeCostMultiplier`

Multiplier for the cost of traversing volume areas.

6.48.3.14 Volumes `NativeParallelHashMap<long, NativeNavVolumeData> Infohazard.HyperNav.↩ Jobs.NavPathJob.Volumes`

Map containing all loaded NavVolumes, keyed by their instance ID>

The documentation for this struct was generated from the following file:

- Runtime/Jobs/NavPathJob.cs

6.49 Infohazard.HyperNav.Jobs.NavRaycastJob Struct Reference

Job that performs a single raycast in a NavVolume.

Public Member Functions

- void Execute ()

  Execute the raycast.

Public Attributes

- NativeNavVolumeData Volume

  Volume to raycast in.
- float4 Start

  Start point of the segment.
- float4 End

  End point of the segment.
- NativeArray< float > OutDistance

  A single-element array where the hit distance (or -1 if no hit) is written.

6.49.1 Detailed Description

Job that performs a single raycast in a NavVolume.

6.49.2 Member Function Documentation

6.49.2.1 Execute() `void Infohazard.HyperNav.Jobs.NavRaycastJob.Execute ( )`

Execute the raycast.

6.49.3 Member Data Documentation

6.49.3.1 End `float4 Infohazard.HyperNav.Jobs.NavRaycastJob.End`

End point of the segment.

6.49.3.2 OutDistance `NativeArray<float> Infohazard.HyperNav.Jobs.NavRaycastJob.OutDistance`

A single-element array where the hit distance (or -1 if no hit) is written.

6.49.3.3 Start `float4 Infohazard.HyperNav.Jobs.NavRaycastJob.Start`

Start point of the segment.

6.49.3.4 Volume `NativeNavVolumeData Infohazard.HyperNav.Jobs.NavRaycastJob.Volume`

Volume to raycast in.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/NavRaycastJob.cs

6.50 Infohazard.HyperNav.NavRegionData Class Reference

The serialized data representing a single region in a NavVolume.

Public Member Functions

- NavRegionData ()

  Default constructor for serialization.
- NavRegionData (int id, Bounds bounds, SerializableRange triangleIndexRange, SerializableRange boundPlaneRange, SerializableRange internalLinkRange)

  Create a new NavRegionData with the given properties.
- NavRegionData (in NativeNavVolumeRegionData data)

  Create a new NavRegionData with the given native data.
- NativeNavVolumeRegionData ToNativeData (SerializableRange externalLinkRange)

  Convert to a native representation.

Properties

- int ID `[get]`

  The ID of the region.
- Bounds Bounds `[get]`

  The bounds of the region in local space of the volume.
- SerializableRange TriangleIndexRange `[get]`

  Range of the region's triangle indices in the volume's index array.
- SerializableRange BoundPlaneRange `[get]`

  Range of bound planes of this region in the volume's bound planes array.
- SerializableRange InternalLinkRange `[get]`

  Range of links between this region and other regions in volume's internal link array.

6.50.1 Detailed Description

The serialized data representing a single region in a NavVolume.

6.50.2 Constructor & Destructor Documentation

6.50.2.1 NavRegionData() `[1/3]` Infohazard.HyperNav.NavRegionData.NavRegionData ( )

Default constructor for serialization.

6.50.2.2 NavRegionData() `[2/3]`

```
Infohazard.HyperNav.NavRegionData.NavRegionData (
    int id,
    Bounds bounds,
    SerializableRange triangleIndexRange,
    SerializableRange boundPlaneRange,
    SerializableRange internalLinkRange )
```

Create a new NavRegionData with the given properties.

Parameters

id	ID of the region.
bounds	Bounds of the region.
triangleIndexRange	Range of the region's triangle indices.
boundPlaneRange	Range of bound planes of this region.
internalLinkRange	Range of links between this region and other regions.

6.50.2.3 NavRegionData() `[3/3]`

```
Infohazard.HyperNav.NavRegionData.NavRegionData (
    in NativeNavVolumeRegionData data )
```

Create a new NavRegionData with the given native data.

Parameters

data	Native data to copy.

6.50.3 Member Function Documentation

6.50.3.1 ToNativeData() `NativeNavVolumeRegionData Infohazard.HyperNav.NavRegionData.ToNativeData (`

`SerializableRange externalLinkRange )`

Convert to a native representation.

Parameters

externalLinkRange	External links, which are included in the native data but not the serialized data.

Returns

Created native data.

6.50.4 Property Documentation

6.50.4.1 BoundPlaneRange `SerializableRange Infohazard.HyperNav.NavRegionData.BoundPlaneRange [get]`

Range of bound planes of this region in the volume's bound planes array.

6.50.4.2 Bounds `Bounds Infohazard.HyperNav.NavRegionData.Bounds [get]`

The bounds of the region in local space of the volume.

6.50.4.3 ID `int Infohazard.HyperNav.NavRegionData.ID [get]`

The ID of the region.

6.50.4.4 InternalLinkRange `SerializableRange Infohazard.HyperNav.NavRegionData.InternalLink↩ Range [get]`

Range of links between this region and other regions in volume's internal link array.

6.50.4.5 TriangleIndexRange `SerializableRange Infohazard.HyperNav.NavRegionData.Triangle↩ IndexRange [get]`

Range of the region's triangle indices in the volume's index array.

The documentation for this class was generated from the following file:

- Runtime/NavVolumeData.cs

6.51 Infohazard.HyperNav.Jobs.Utility.NavSampleResult Struct Reference

A native-friendly representation of a navigation query result.

Public Member Functions

- NavSampleResult (long areaID, NavAreaTypes type, NavLayer layer, int region, bool isOnEdge, float4 position, float4 up, float distance)

  Initialize a new NativeNavHit with the given data.

Public Attributes

- readonly long AreaID

  ID of the volume that was hit.
- readonly NavAreaTypes Type

  Type of area that was hit.
- readonly NavLayer Layer

  The layer of the area that was hit.
- readonly int Region

  ID of the region that was hit.
- readonly bool IsOnEdge

  Whether the result point was on the edge of the region.
- readonly float4 Position

  Position of the hit.
- readonly float4 Up

  Up vector at the hit point. Zero for volumes.
- readonly float Distance

  Distance from the sample position to the hit point.

Properties

- NavVolume Volume [get]

  Get the NavVolume instance that was hit. Not available in Burst. If the result type is not NavAreaTypes.Volume, this will return null.
- NavSurface Surface [get]

  Get the NavSurface instance that was hit. Not available in Burst. If the result type is not NavAreaTypes.Surface, this will return null.
- NavAreaBase Area [get]

  Get the NavAreaBase instance that was hit. Not available in Burst.

6.51.1 Detailed Description

A native-friendly representation of a navigation query result.

6.51.2 Constructor & Destructor Documentation

6.51.2.1 NavSampleResult()

```
Infohazard.HyperNav.Jobs.Utility.NavSampleResult.NavSampleResult (
    long areaID,
    NavAreaTypes type,
    NavLayer layer,
    int region,
    bool isOnEdge,
    float4 position,
    float4 up,
    float distance )
```

Initialize a new NativeNavHit with the given data.

Parameters

areaID	ID of the volume that was hit.
type	Type of area that was hit.
layer	Layer of the area that was hit.
region	ID of the region that was hit.
isOnEdge	Whether the result point was on the edge of the region.
position	Position of the hit.
up	Up vector at the hit point. Zero for volumes.
distance	Distance from the sample position to the hit point.

6.51.3 Member Data Documentation

6.51.3.1 AreaID `readonly long Infohazard.HyperNav.Jobs.Utility.NavSampleResult.AreaID`

ID of the volume that was hit.

6.51.3.2 Distance `readonly float Infohazard.HyperNav.Jobs.Utility.NavSampleResult.Distance`

Distance from the sample position to the hit point.

6.51.3.3 IsOnEdge `readonly bool Infohazard.HyperNav.Jobs.Utility.NavSampleResult.IsOnEdge`

Whether the result point was on the edge of the region.

6.51.3.4 Layer `readonly NavLayer Infohazard.HyperNav.Jobs.Utility.NavSampleResult.Layer`

The layer of the area that was hit.

6.51.3.5 Position `readonly float4 Infohazard.HyperNav.Jobs.Utility.NavSampleResult.Position`

Position of the hit.

6.51.3.6 Region `readonly int Infohazard.HyperNav.Jobs.Utility.NavSampleResult.Region`

ID of the region that was hit.

6.51.3.7 Type `readonly NavAreaTypes Infohazard.HyperNav.Jobs.Utility.NavSampleResult.Type`

Type of area that was hit.

6.51.3.8 Up `readonly float4 Infohazard.HyperNav.Jobs.Utility.NavSampleResult.Up`

Up vector at the hit point. Zero for volumes.

6.51.4 Property Documentation

6.51.4.1 Area `NavAreaBase Infohazard.HyperNav.Jobs.Utility.NavSampleResult.Area [get]`

Get the NavAreaBase instance that was hit. Not available in Burst.

This property performs a dictionary lookup, so the result should be cached.

6.51.4.2 Surface `NavSurface Infohazard.HyperNav.Jobs.Utility.NavSampleResult.Surface [get]`

Get the NavSurface instance that was hit. Not available in Burst. If the result type is not NavAreaTypes.Surface, this will return null.

This property performs a dictionary lookup, so the result should be cached.

6.51.4.3 Volume `NavVolume Infohazard.HyperNav.Jobs.Utility.NavSampleResult.Volume [get]`

Get the NavVolume instance that was hit. Not available in Burst. If the result type is not NavAreaTypes.Volume, this will return null.

This property performs a dictionary lookup, so the result should be cached.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Utility/NavSampleResult.cs

6.52 Infohazard.HyperNav.NavSurfaceBakeHandler Class Reference

Manages the baking process for a NavSurface. Cannot be reused.

Public Member Functions

- NavSurfaceBakeHandler (NavSurface surface, bool sanityChecks, bool updateSerializedData, NavSurfaceVisualizationMode? visualizationMode=null)

  Create a new bake handler for a volume.

6.52.1 Detailed Description

Manages the baking process for a NavSurface. Cannot be reused.

6.52.2 Constructor & Destructor Documentation

6.52.2.1 NavSurfaceBakeHandler() `Infohazard.HyperNav.NavSurfaceBakeHandler.NavSurfaceBakeHandler (`

```
NavSurface surface,
bool sanityChecks,
bool updateSerializedData,
NavSurfaceVisualizationMode?  visualizationMode = null )
```

Create a new bake handler for a volume.

Parameters

surface	Surface to bake.
sanityChecks	Whether to run sanity checks to catch baking issues.
updateSerializedData	Whether to update the serialized data after baking.
visualizationMode	Visualization mode to use during baking.

The documentation for this class was generated from the following file:

- Runtime/Utility/NavSurfaceBakeHandler.cs

6.53 Infohazard.HyperNav.NavSurfaceInternalLinkData Class Reference

A connection from one region to another region in the same surface.

Public Member Functions

- NavSurfaceInternalLinkData ()

  Default constructor for serialization.
- NavSurfaceInternalLinkData (int connectedRegionID, SerializableRange vertexRange, SerializableRange edgeRange)

  Create a new NavSurfaceInternalLinkData with the given properties.
- NavSurfaceInternalLinkData (in NativeNavSurfaceInternalLinkData data)

  Create a new NavSurfaceInternalLinkData with the given native data.
- NativeNavSurfaceInternalLinkData ToNativeData ()

  Convert to a native representation.

Properties

- int ConnectedRegionID `[get]`

  The ID of the connected region.
- SerializableRange VertexRange `[get]`

  Range of the link's indices in the surface's NavSurfaceData.LinkVertices array.
- SerializableRange EdgeRange `[get]`

  Range of the link's indices in the surface's NavSurfaceData.LinkEdges array.

6.53.1 Detailed Description

A connection from one region to another region in the same surface.

6.53.2 Constructor & Destructor Documentation

6.53.2.1 NavSurfaceInternalLinkData() [1/3] Infohazard.HyperNav.NavSurfaceInternalLinkData.Nav↩ SurfaceInternalLinkData ( )

Default constructor for serialization.

6.53.2.2 NavSurfaceInternalLinkData() [2/3] Infohazard.HyperNav.NavSurfaceInternalLinkData.Nav↩ SurfaceInternalLinkData (

```
int connectedRegionID,
SerializableRange vertexRange,
SerializableRange edgeRange )
```

Create a new NavSurfaceInternalLinkData with the given properties.

Parameters

connectedRegionID	ID of the connected region.
vertexRange	Range of the link's indices.
edgeRange	Range of the link's indices.

6.53.2.3 NavSurfaceInternalLinkData() [3/3] Infohazard.HyperNav.NavSurfaceInternalLinkData.Nav↩ SurfaceInternalLinkData (

```
in NativeNavSurfaceInternalLinkData data )
```

Create a new NavSurfaceInternalLinkData with the given native data.

Parameters

data	Native data to copy.

6.53.3 Member Function Documentation

6.53.3.1 ToNativeData() NativeNavSurfaceInternalLinkData Infohazard.HyperNav.NavSurfaceInternal↩ LinkData.ToNativeData ( )

Convert to a native representation.

Returns

Created native data.

6.53.4 Property Documentation

6.53.4.1 ConnectedRegionID `int Infohazard.HyperNav.NavSurfaceInternalLinkData.ConnectedRegionID [get]`

The ID of the connected region.

6.53.4.2 EdgeRange `SerializableRange Infohazard.HyperNav.NavSurfaceInternalLinkData.EdgeRange [get]`

Range of the link's indices in the surface's NavSurfaceData.LinkEdges array.

6.53.4.3 VertexRange `SerializableRange Infohazard.HyperNav.NavSurfaceInternalLinkData.VertexRange [get]`

Range of the link's indices in the surface's NavSurfaceData.LinkVertices array.

The documentation for this class was generated from the following file:

- Runtime/NavSurfaceData.cs

6.54 Infohazard.HyperNav.NavSurfaceRegionData Class Reference

The serialized data representing a single region in a NavSurface.

Public Member Functions

- NavSurfaceRegionData ()

  Default constructor for serialization.
- NavSurfaceRegionData (int id, Bounds bounds, SerializableRange triangleIndexRange, SerializableRange internalLinkRange, Vector3 upDirection, Vector3 normalVector, int islandIndex)

  Create a new NavRegionData with the given properties.
- NavSurfaceRegionData (in NativeNavSurfaceRegionData data)

  Create a new NavRegionData with the given native data.
- NativeNavSurfaceRegionData ToNativeData (SerializableRange externalLinkRange)

  Convert to a native representation.

Properties

- int ID `[get]`

  The ID of the region.
- Bounds Bounds `[get]`

  The bounds of the region in local space of the surface.
- SerializableRange TriangleIndexRange `[get]`

  Range of the region's triangle indices in the surface's index array.
- SerializableRange InternalLinkRange `[get]`

  Range of links between this region and other regions in surface's internal link array.
- Vector3 UpDirection `[get]`

  The upward direction for the region.
- Vector3 NormalVector `[get]`

  The normal vector for the region, which may be different from the up direction depending on the upward direction mode. If the normal vector is not set, it defaults to the up direction.
- int IslandIndex `[get]`

  The index of the island this region belongs to. Surface regions in the same island are connected.

6.54.1 Detailed Description

The serialized data representing a single region in a NavSurface.

6.54.2 Constructor & Destructor Documentation

6.54.2.1 NavSurfaceRegionData() [1/3] Infohazard.HyperNav.NavSurfaceRegionData.NavSurface↩ RegionData ( )

Default constructor for serialization.

6.54.2.2 NavSurfaceRegionData() [2/3] Infohazard.HyperNav.NavSurfaceRegionData.NavSurface↩ RegionData (

```
int id,
Bounds bounds,
SerializableRange triangleIndexRange,
SerializableRange internalLinkRange,
Vector3 upDirection,
Vector3 normalVector,
int islandIndex )
```

Create a new NavRegionData with the given properties.

Parameters

id	ID of the region.
bounds	Bounds of the region.
triangleIndexRange	Range of the region's triangle indices.
internalLinkRange	Range of links between this region and other regions.
upDirection	The upward direction for the region.
normalVector	The normal vector for the region.
islandIndex	Index of the island this region belongs to.

6.54.2.3 NavSurfaceRegionData() [3/3] Infohazard.HyperNav.NavSurfaceRegionData.NavSurface↩ RegionData (

```
in NativeNavSurfaceRegionData data )
```

Create a new NavRegionData with the given native data.

Parameters

data	Native data to copy.

6.54.3 Member Function Documentation

6.54.3.1 ToNativeData() NativeNavSurfaceRegionData Infohazard.HyperNav.NavSurfaceRegionData.↩ ToNativeData (SerializableRange *externalLinkRange*)

Convert to a native representation.

Parameters

externalLinkRange	External links, which are included in the native data but not the serialized data.

Returns

Created native data.

6.54.4 Property Documentation

6.54.4.1 Bounds Bounds Infohazard.HyperNav.NavSurfaceRegionData.Bounds [get]

The bounds of the region in local space of the surface.

6.54.4.2 ID int Infohazard.HyperNav.NavSurfaceRegionData.ID [get]

The ID of the region.

6.54.4.3 InternalLinkRange SerializableRange Infohazard.HyperNav.NavSurfaceRegionData.Internal↩ LinkRange [get]

Range of links between this region and other regions in surface's internal link array.

6.54.4.4 IslandIndex int Infohazard.HyperNav.NavSurfaceRegionData.IslandIndex [get]

The index of the island this region belongs to. Surface regions in the same island are connected.

6.54.4.5 NormalVector `Vector3 Infohazard.HyperNav.NavSurfaceRegionData.NormalVector [get]`

The normal vector for the region, which may be different from the up direction depending on the upward direction mode. If the normal vector is not set, it defaults to the up direction.

6.54.4.6 TriangleIndexRange `SerializableRange Infohazard.HyperNav.NavSurfaceRegionData.↩ TriangleIndexRange [get]`

Range of the region's triangle indices in the surface's index array.

6.54.4.7 UpDirection `Vector3 Infohazard.HyperNav.NavSurfaceRegionData.UpDirection [get]`

The upward direction for the region.

The documentation for this class was generated from the following file:

- Runtime/NavSurfaceData.cs

6.55 Infohazard.HyperNav.Settings.NavSurfaceSettings Class Reference

Settings for baking NavSurfaces.

Classes

- class PropNames

  This is used to refer to the names of private fields in this class from a custom Editor.

Public Member Functions

- override NavSurfaceSettings Clone ()

  Clone this settings object.

  Returns

  A clone of this settings object.

Properties

- float MaxAgentHeight `[get, set]`

  The maximum height of agents walking on this surface.
- float ErosionDistance `[get, set]`

  The minimum distance from edge of the surface to the edge of walkable ground.
- LayerMask WalkableLayers `[get, set]`

  Layers to consider as walkable.
- NavSurfaceUprightDirectionMode UprightDirectionMode `[get, set]`

  How to determine the 'upright' direction for walking on the surface.
- Vector3 FixedUprightDirection `[get, set]`

  Fixed direction to use as the upright direction. Only used if UprightDirectionMode is set to NavSurfaceUpright↩ DirectionMode.FixedWorldDirection or NavSurfaceUprightDirectionMode.FixedLocalDirection.
- float SlopeAngleLimit `[get, set]`

  Angle limit in degrees between hit normal and upright direction. Only used if UprightDirectionMode is set to Nav↩ SurfaceUprightDirectionMode.FixedWorldDirection or NavSurfaceUprightDirectionMode.FixedLocalDirection.
- ISurfaceUprightDirectionHandler CustomUprightDirectionHandler `[get, set]`

  Custom handler to determine the upright direction. Only used if UprightDirectionMode is set to NavSurfaceUpright↩ DirectionMode.Custom.
- float MaxAngleBetweenUpDirectionsWithinTriangle `[get, set]`

  The ground normals at all vertices of a triangle must be within this angle (in degrees) of each other. Otherwise, the triangle is discarded.
- float MaxAngleBetweenUpDirectionsBetweenTriangles `[get, set]`

  The ground normals of triangles at a vertex must be within this angle (in degrees) of each other. Otherwise, the vertex will be split into groups.
- int MaxTriangleDivisions `[get, set]`

  Maximum number of times to divide a triangle lying on a corner to try triangles on flat surfaces.
- float MinIslandSurfaceArea `[get, set]`

  If greater than zero, islands with less surface area will be discarded.
- float DecimationThreshold `[get, set]`

  Threshold for decimation - a higher threshold means more aggressive decimation.
- float BoundaryTriangleClippingThreshold `[get, set]`

  Threshold for removing bound vertices in decimation - a higher threshold means more aggressive decimation.
- float MinTriangleAreaFraction `[get, set]`

  The minimum area of a triangle as a fraction of the square of the voxel size. Smaller triangles are merged or discarded.
- float MaxSurfaceVolumeApproachAngle `[get, set]`

  The maximum angle in degrees between the surface normal and the approach direction from volumes to that surface, or that surface to volumes.

6.55.1 Detailed Description

Settings for baking NavSurfaces.

6.55.2 Member Function Documentation

6.55.2.1 Clone() override NavSurfaceSettings Infohazard.HyperNav.Settings.NavSurfaceSettings.↩ Clone ( ) [virtual]

Clone this settings object.

Returns

A clone of this settings object.

Reimplemented from Infohazard.HyperNav.Settings.NavAreaBaseSettings< NavSurfaceSettings >.

6.55.3 Property Documentation

6.55.3.1 BoundaryTriangleClippingThreshold float Infohazard.HyperNav.Settings.NavSurface↩ Settings.BoundaryTriangleClippingThreshold [get], [set]

Threshold for removing bound vertices in decimation - a higher threshold means more aggressive decimation.

6.55.3.2 CustomUprightDirectionHandler ISurfaceUprightDirectionHandler Infohazard.HyperNav.↩ Settings.NavSurfaceSettings.CustomUprightDirectionHandler [get], [set]

Custom handler to determine the upright direction. Only used if UprightDirectionMode is set to NavSurfaceUpright↩ DirectionMode.Custom.

6.55.3.3 DecimationThreshold float Infohazard.HyperNav.Settings.NavSurfaceSettings.Decimation↩ Threshold [get], [set]

Threshold for decimation - a higher threshold means more aggressive decimation.

6.55.3.4 ErosionDistance float Infohazard.HyperNav.Settings.NavSurfaceSettings.ErosionDistance [get], [set]

The minimum distance from edge of the surface to the edge of walkable ground.

6.55.3.5 FixedUprightDirection Vector3 Infohazard.HyperNav.Settings.NavSurfaceSettings.Fixed↩ UprightDirection [get], [set]

Fixed direction to use as the upright direction. Only used if UprightDirectionMode is set to NavSurfaceUpright↩ DirectionMode.FixedWorldDirection or NavSurfaceUprightDirectionMode.FixedLocalDirection.

6.55.3.6 MaxAgentHeight `float Infohazard.HyperNav.Settings.NavSurfaceSettings.MaxAgentHeight [get], [set]`

The maximum height of agents walking on this surface.

6.55.3.7 MaxAngleBetweenUpDirectionsBetweenTriangles `float Infohazard.HyperNav.Settings.NavSurfaceSettings.MaxAngleBetweenUpDirectionsBetweenTriangles [get], [set]`

The ground normals of triangles at a vertex must be within this angle (in degrees) of each other. Otherwise, the vertex will be split into groups.

6.55.3.8 MaxAngleBetweenUpDirectionsWithinTriangle `float Infohazard.HyperNav.Settings.NavSurfaceSettings.MaxAngleBetweenUpDirectionsWithinTriangle [get], [set]`

The ground normals at all vertices of a triangle must be within this angle (in degrees) of each other. Otherwise, the triangle is discarded.

6.55.3.9 MaxSurfaceVolumeApproachAngle `float Infohazard.HyperNav.Settings.NavSurfaceSettings.MaxSurfaceVolumeApproachAngle [get], [set]`

The maximum angle in degrees between the surface normal and the approach direction from volumes to that surface, or that surface to volumes.

6.55.3.10 MaxTriangleDivisions `int Infohazard.HyperNav.Settings.NavSurfaceSettings.MaxTriangleDivisions [get], [set]`

Maximum number of times to divide a triangle lying on a corner to try triangles on flat surfaces.

6.55.3.11 MinIslandSurfaceArea `float Infohazard.HyperNav.Settings.NavSurfaceSettings.MinIslandSurfaceArea [get], [set]`

If greater than zero, islands with less surface area will be discarded.

6.55.3.12 MinTriangleAreaFraction `float Infohazard.HyperNav.Settings.NavSurfaceSettings.MinTriangleAreaFraction [get], [set]`

The minimum area of a triangle as a fraction of the square of the voxel size. Smaller triangles are merged or discarded.

6.55.3.13 SlopeAngleLimit `float Infohazard.HyperNav.Settings.NavSurfaceSettings.SlopeAngleLimit [get], [set]`

Angle limit in degrees between hit normal and upright direction. Only used if UprightDirectionMode is set to NavSurfaceUprightDirectionMode.FixedWorldDirection or NavSurfaceUprightDirectionMode.FixedLocalDirection.

6.55.3.14 UprightDirectionMode `NavSurfaceUprightDirectionMode Infohazard.HyperNav.Settings.NavSurfaceSettings.UprightDirectionMode [get], [set]`

How to determine the 'upright' direction for walking on the surface.

6.55.3.15 WalkableLayers `LayerMask Infohazard.HyperNav.Settings.NavSurfaceSettings.WalkableLayers [get], [set]`

Layers to consider as walkable.

The documentation for this class was generated from the following file:

- Runtime/Settings/NavSurfaceSettings.cs

6.56 Infohazard.HyperNav.Settings.NavSurfaceSettingsAsset Class Reference

Settings asset for baking NavSurfaces.

Additional Inherited Members

6.56.1 Detailed Description

Settings asset for baking NavSurfaces.

The documentation for this class was generated from the following file:

- Runtime/Settings/NavSurfaceSettingsAsset.cs

6.57 Infohazard.HyperNav.NavSurfaceUpdate Class Reference

Utility class for updating NavSurface data.

Static Public Member Functions

- static async UniTask GenerateSurfaceData (NavSurface surface, bool updateSerializedData, NativeCancellationToken cancellationToken, NavSurfaceUpdateStepCallback stepCallback=null)

  Generate surface data for a NavSurface. Can be used at runtime to support dynamic baking. Note that runtime baking is a performance intensive operation and should occur during a loading screen if possible.

6.57.1 Detailed Description

Utility class for updating NavSurface data.

6.57.2 Member Function Documentation

6.57.2.1 GenerateSurfaceData() `static async UniTask Infohazard.HyperNav.NavSurfaceUpdate.↩ GenerateSurfaceData (`

```
            NavSurface surface,
            bool updateSerializedData,
            NativeCancellationToken cancellationToken,
            NavSurfaceUpdateStepCallback stepCallback = null )  [static]
```

Generate surface data for a NavSurface. Can be used at runtime to support dynamic baking. Note that runtime baking is a performance intensive operation and should occur during a loading screen if possible.

Parameters

surface	The surface to update.
updateSerializedData	Whether to update the serialized data on the surface.
cancellationToken	Cancellation token for the operation.
stepCallback	Optional callback that will be invoked at each step of the update process.

The documentation for this class was generated from the following file:

- Runtime/Utility/NavSurfaceUpdate.cs

6.58 Infohazard.HyperNav.NavVolume Class Reference

A volume of space in which HyperNav pathfinding can occur.

Public Member Functions

- bool Raycast (Vector3 start, Vector3 end, out float hit)

  Cast a ray against the blocking triangles of the volume, and return the nearest hit.
- bool Raycast (Vector3 start, Vector3 end)

  Cast a ray against the blocking triangles of the volume, and return if there was a hit.

Properties

- NavVolumeVisualizationMode VisualizationMode `[get, set]`

  Stage at which to visualize the volume bake process in the scene view.

Additional Inherited Members

6.58.1 Detailed Description

A volume of space in which HyperNav pathfinding can occur.

Each NavVolume is divided into convex regions that form pathfinding nodes. A volume's regions can have connections to each other, and to regions of other volumes. The information in a NavVolume must be baked in the editor - it cannot be calculated at runtime (for now).

6.58.2 Member Function Documentation

6.58.2.1 Raycast() [1/2]

```
bool Infohazard.HyperNav.NavVolume.Raycast (
        Vector3 start,
        Vector3 end )
```

Cast a ray against the blocking triangles of the volume, and return if there was a hit.

More performant than Raycast(Vector3, Vector3, out float) if you don't need the hit position, because it can return as soon as any hit is detected.

Parameters

start	The position (in world space) to start the query at.
end	The position (in world space) to end the query at.

Returns

Whether a triangle was hit.

6.58.2.2 Raycast() [2/2]

```
bool Infohazard.HyperNav.NavVolume.Raycast (
        Vector3 start,
        Vector3 end,
        out float hit )
```

Cast a ray against the blocking triangles of the volume, and return the nearest hit.

Parameters

start	The position (in world space) to start the query at.
end	The position (in world space) to end the query at.
hit	If the query hits a triangle, the ratio between start and end at which the hit occurred.

Returns

Whether a triangle was hit.

6.58.3 Property Documentation

6.58.3.1 VisualizationMode `NavVolumeVisualizationMode Infohazard.HyperNav.NavVolume.VisualizationMode [get], [set]`

Stage at which to visualize the volume bake process in the scene view.

The documentation for this class was generated from the following file:

- Runtime/NavVolume.cs

6.59 Infohazard.HyperNav.NavVolumeBakeHandler Class Reference

Manages the baking process for a NavVolume. Cannot be reused.

Public Member Functions

- NavVolumeBakeHandler (NavVolume volume, bool sanityChecks, bool updateSerializedData, NavVolumeVisualizationMode? visualizationMode=null)

  Create a new bake handler for a volume.

6.59.1 Detailed Description

Manages the baking process for a NavVolume. Cannot be reused.

6.59.2 Constructor & Destructor Documentation

6.59.2.1 NavVolumeBakeHandler() `Infohazard.HyperNav.NavVolumeBakeHandler.NavVolumeBakeHandler`

```
(
            NavVolume volume,
            bool sanityChecks,
            bool updateSerializedData,
            NavVolumeVisualizationMode?  visualizationMode = null )
```

Create a new bake handler for a volume.

Parameters

volume	Volume to bake.
sanityChecks	Whether to run sanity checks to catch baking issues.
updateSerializedData	Whether to update the serialized data after baking.
visualizationMode	The visualization mode to use during baking.

The documentation for this class was generated from the following file:

- Runtime/Utility/NavVolumeBakeHandler.cs

6.60 Infohazard.HyperNav.NavVolumeData Class Reference

The baked data of a NavVolume, saved as an asset.

Public Member Functions

- void UpdateData (Vector3[ ] vertices, NavRegionData[ ] regions, int[ ] triangleIndices, int blockingTriangleIndexCount, NavInternalLinkData[ ] internalLinks, NativePlane[ ] boundPlanes, int[ ] linkVertices, Edge[ ] linkEdges, Triangle[ ] linkTriangles)

  Update the data manually.
- void Clear ()

  Clear the properties of this NavVolumeData.
- void UpdateFromNativeData (in NativeNavVolumeData data)

  Update this volume data from the given native data.
- bool ToNativeData (INavArea area, out NativeNavVolumeData data, out NativeNavVolumeDataPointers pointers)

  Convert this NavVolumeData to the native format so that it can be used by jobs.

Properties

- IReadOnlyList< Vector3 > Vertices `[get]`

  The vertex positions of all of the volume's regions, in local space.
- IReadOnlyList< NavRegionData > Regions `[get]`

  The regions of the volume.
- IReadOnlyList< int > TriangleIndices `[get]`

  The vertex indices of triangles.
- int BlockingTriangleIndexCount `[get]`

  The number of indices in the TriangleIndices array that represent blocking triangles.
- IReadOnlyList< NativePlane > BoundPlanes `[get]`

  The bound planes of all of the volume's regions.
- IReadOnlyList< NavInternalLinkData > InternalLinks `[get]`

  The internal links of all of the volume's regions.
- IReadOnlyList< int > LinkVertices `[get]`

  The shared vertices of all of the volume's internal links.
- IReadOnlyList< Edge > LinkEdges `[get]`

  The shared edges of all of the volume's internal links.
- IReadOnlyList< Triangle > LinkTriangles `[get]`

  The shared triangles of all of the volume's internal links.
- ulong Version `[get]`

  The version of the data, which increments when it is baked.

6.60.1 Detailed Description

The baked data of a NavVolume, saved as an asset.

6.60.2 Member Function Documentation

6.60.2.1 Clear() `void Infohazard.HyperNav.NavVolumeData.Clear ( )`

Clear the properties of this NavVolumeData.

6.60.2.2 ToNativeData() `bool Infohazard.HyperNav.NavVolumeData.ToNativeData (`

```
INavArea area,
out NativeNavVolumeData data,
out NativeNavVolumeDataPointers pointers )
```

Convert this NavVolumeData to the native format so that it can be used by jobs.

Parameters

area	Area that owns the data.
data	Created native data.
pointers	Created data structure references (must be kept in order to deallocate).

6.60.2.3 UpdateData() `void Infohazard.HyperNav.NavVolumeData.UpdateData (`

```
Vector3[] vertices,
NavRegionData[] regions,
int[] triangleIndices,
int blockingTriangleIndexCount,
NavInternalLinkData[] internalLinks,
NativePlane[] boundPlanes,
int[] linkVertices,
Edge[] linkEdges,
Triangle[] linkTriangles )
```

Update the data manually.

Parameters

vertices	The vertex positions of all of the volume's regions, in local space.
regions	The regions of the volume.
triangleIndices	The vertex indices of triangles.
blockingTriangleIndexCount	The number of indices that represent blocking triangles.
internalLinks	The internal links of all of the volume's regions.
boundPlanes	The bound planes of all of the volume's regions.
linkVertices	The shared vertices of all of the volume's internal links.
linkEdges	The shared edges of all of the volume's internal links.
linkTriangles	The shared triangles of all of the volume's internal links.

6.60.2.4 UpdateFromNativeData() `void Infohazard.HyperNav.NavVolumeData.UpdateFromNativeData ( in NativeNavVolumeData data )`

Update this volume data from the given native data.

6.60.3 Property Documentation

6.60.3.1 BlockingTriangleIndexCount `int Infohazard.HyperNav.NavVolumeData.BlockingTriangleIndexCount [get]`

The number of indices in the TriangleIndices array that represent blocking triangles.

6.60.3.2 BoundPlanes `IReadOnlyList<NativePlane> Infohazard.HyperNav.NavVolumeData.BoundPlanes [get]`

The bound planes of all of the volume's regions.

6.60.3.3 InternalLinks `IReadOnlyList<NavInternalLinkData> Infohazard.HyperNav.NavVolumeData.InternalLinks [get]`

The internal links of all of the volume's regions.

6.60.3.4 LinkEdges `IReadOnlyList<Edge> Infohazard.HyperNav.NavVolumeData.LinkEdges [get]`

The shared edges of all of the volume's internal links.

6.60.3.5 LinkTriangles `IReadOnlyList<Triangle> Infohazard.HyperNav.NavVolumeData.LinkTriangles [get]`

The shared triangles of all of the volume's internal links.

6.60.3.6 LinkVertices `IReadOnlyList<int> Infohazard.HyperNav.NavVolumeData.LinkVertices [get]`

The shared vertices of all of the volume's internal links.

6.60.3.7 Regions `IReadOnlyList<NavRegionData> Infohazard.HyperNav.NavVolumeData.Regions [get]`

The regions of the volume.

6.60.3.8 TriangleIndices `IReadOnlyList<int> Infohazard.HyperNav.NavVolumeData.TriangleIndices [get]`

The vertex indices of triangles.

6.60.3.9 Version `ulong Infohazard.HyperNav.NavVolumeData.Version [get]`

The version of the data, which increments when it is baked.

6.60.3.10 Vertices `IReadOnlyList<Vector3> Infohazard.HyperNav.NavVolumeData.Vertices [get]`

The vertex positions of all of the volume's regions, in local space.

The documentation for this class was generated from the following file:

- Runtime/NavVolumeData.cs

6.61 Infohazard.HyperNav.Editor.NavVolumeEditor Class Reference

Custom editor for Infohazard.HyperNav.NavVolume.

6.61.1 Detailed Description

Custom editor for Infohazard.HyperNav.NavVolume.

The documentation for this class was generated from the following file:

- Editor/NavVolumeEditor.cs

6.62 Infohazard.HyperNav.Settings.NavVolumeSettings Class Reference

Settings for baking NavVolumes.

Classes

- class PropNames

  Serialized property names for NavVolumeSettings.

Public Member Functions

- override NavVolumeSettings Clone ()

  Clone this settings object.

  Returns

  A clone of this settings object.

Properties

- bool EnableMultiQuery `[get, set]`

  Whether to enable multiple physics queries per voxel to get a more accurate result.

6.62.1 Detailed Description

Settings for baking NavVolumes.

6.62.2 Member Function Documentation

6.62.2.1 Clone() `override NavVolumeSettings Infohazard.HyperNav.Settings.NavVolumeSettings.Clone ( ) [virtual]`

Clone this settings object.

Returns

A clone of this settings object.

Reimplemented from Infohazard.HyperNav.Settings.NavAreaBaseSettings< NavVolumeSettings >.

6.62.3 Property Documentation

6.62.3.1 EnableMultiQuery `bool Infohazard.HyperNav.Settings.NavVolumeSettings.EnableMultiQuery [get], [set]`

Whether to enable multiple physics queries per voxel to get a more accurate result.

The documentation for this class was generated from the following file:

- Runtime/Settings/NavVolumeSettings.cs

6.63 Infohazard.HyperNav.Settings.NavVolumeSettingsAsset Class Reference

Settings asset for baking NavVolumes.

Additional Inherited Members

6.63.1 Detailed Description

Settings asset for baking NavVolumes.

The documentation for this class was generated from the following file:

- Runtime/Settings/NavVolumeSettingsAsset.cs

6.64 Infohazard.HyperNav.NavVolumeUpdate Class Reference

Utility functions for updating NavVolume data.

Static Public Member Functions

- static async UniTask GenerateVolumeData (NavVolume volume, bool updateSerializedData, Native↩ CancellationToken cancellationToken, NavVolumeUpdateStepCallback stepCallback=null)

  Generate the data for a NavVolume. Can be used at runtime to support dynamic baking. Note that runtime baking is a performance intensive operation and should occur during a loading screen if possible.

6.64.1 Detailed Description

Utility functions for updating NavVolume data.

6.64.2 Member Function Documentation

6.64.2.1 GenerateVolumeData() `static async UniTask Infohazard.HyperNav.NavVolumeUpdate.↩ GenerateVolumeData (`

```
            NavVolume volume,
            bool updateSerializedData,
            NativeCancellationToken cancellationToken,
            NavVolumeUpdateStepCallback stepCallback = null )  [static]
```

Generate the data for a NavVolume. Can be used at runtime to support dynamic baking. Note that runtime baking is a performance intensive operation and should occur during a loading screen if possible.

Parameters

volume	The NavVolume to update.
updateSerializedData	Whether to update the serialized data on the surface.
cancellationToken	Cancellation token for the operation.
stepCallback	Optional callback that will be invoked at each step of the update process.

The documentation for this class was generated from the following file:

- Runtime/Utility/NavVolumeUpdate.cs

6.65 Infohazard.HyperNav.NavWaypoint Struct Reference

A waypoint in a completed path.

Properties

- NavWaypointType Type `[get, set]`

  The type of waypoint in relation to its containing volume.
- Vector3 Position `[get, set]`

  The position of the waypoint in world space.
- Vector3 Up `[get, set]`

  The up vector for a character standing at the waypoint. For waypoints inside volumes, this is zero.
- float AccumulatedDistance `[get, set]`

  The accumulated distance from the start of the path to this waypoint.
- long AreaID `[get, set]`

  Identifier of the NavArea that contains this waypoint, or -1.
- int Region `[get, set]`

  Identifier of the region that contains this waypoint.

6.65.1 Detailed Description

A waypoint in a completed path.

6.65.2 Property Documentation

6.65.2.1 AccumulatedDistance `float Infohazard.HyperNav.NavWaypoint.AccumulatedDistance [get], [set]`

The accumulated distance from the start of the path to this waypoint.

6.65.2.2 AreaID `long Infohazard.HyperNav.NavWaypoint.AreaID [get], [set]`

Identifier of the NavArea that contains this waypoint, or -1.

6.65.2.3 Position `Vector3 Infohazard.HyperNav.NavWaypoint.Position [get], [set]`

The position of the waypoint in world space.

6.65.2.4 Region `int Infohazard.HyperNav.NavWaypoint.Region [get], [set]`

Identifier of the region that contains this waypoint.

6.65.2.5 Type `NavWaypointType Infohazard.HyperNav.NavWaypoint.Type [get], [set]`

The type of waypoint in relation to its containing volume.

6.65.2.6 Up `Vector3 Infohazard.HyperNav.NavWaypoint.Up [get], [set]`

The up vector for a character standing at the waypoint. For waypoints inside volumes, this is zero.

The documentation for this struct was generated from the following file:

- Runtime/NavPathfinder.cs

6.66 Infohazard.HyperNav.NavAgent.PropNames Class Reference

This is used to refer to the names of private fields in this class from a custom Editor.

6.66.1 Detailed Description

This is used to refer to the names of private fields in this class from a custom Editor.

The documentation for this class was generated from the following file:

- Runtime/NavAgent.cs

6.67 Infohazard.HyperNav.NavAreaBase.PropNames Class Reference

This is used to refer to the names of private fields in this class from a custom Editor.

6.67.1 Detailed Description

This is used to refer to the names of private fields in this class from a custom Editor.

The documentation for this class was generated from the following file:

- Runtime/NavArea.cs

6.68 Infohazard.HyperNav.NavPathfinder.PropNames Class Reference

This is used to refer to the names of private fields in this class from a custom Editor.

6.68.1 Detailed Description

This is used to refer to the names of private fields in this class from a custom Editor.

The documentation for this class was generated from the following file:

- Runtime/NavPathfinder.cs

6.69 Infohazard.HyperNav.NavSurface.PropNames Class Reference

This is used to refer to the names of private fields in this class from a custom Editor.

6.69.1 Detailed Description

This is used to refer to the names of private fields in this class from a custom Editor.

The documentation for this class was generated from the following file:

- Runtime/NavSurface.cs

6.70 Infohazard.HyperNav.Settings.NavAreaBaseSettings< TSelf >.PropNames Class Reference

This is used to refer to the names of private fields in this class from a custom Editor.

6.70.1 Detailed Description

This is used to refer to the names of private fields in this class from a custom Editor.

The documentation for this class was generated from the following file:

- Runtime/Settings/NavAreaBaseSettings.cs

6.71 Infohazard.HyperNav.Settings.NavAreaBaseSettingsAsset< TData >.PropNames Class Reference

This is used to refer to the names of private fields in this class from a custom Editor.

6.71.1 Detailed Description

This is used to refer to the names of private fields in this class from a custom Editor.

The documentation for this class was generated from the following file:

- Runtime/Settings/NavAreaBaseSettings.cs

6.72 Infohazard.HyperNav.Settings.NavSurfaceSettings.PropNames Class Reference

This is used to refer to the names of private fields in this class from a custom Editor.

6.72.1 Detailed Description

This is used to refer to the names of private fields in this class from a custom Editor.

The documentation for this class was generated from the following file:

- Runtime/Settings/NavSurfaceSettings.cs

6.73 Infohazard.HyperNav.Settings.NavVolumeSettings.PropNames Class Reference

Serialized property names for NavVolumeSettings.

6.73.1 Detailed Description

Serialized property names for NavVolumeSettings.

The documentation for this class was generated from the following file:

- Runtime/Settings/NavVolumeSettings.cs

6.74 Infohazard.HyperNav.SimpleNavAgentMover.PropNames Class Reference

This is used to refer to the names of private fields in this class from a custom Editor.

6.74.1 Detailed Description

This is used to refer to the names of private fields in this class from a custom Editor.

The documentation for this class was generated from the following file:

- Runtime/SimpleNavAgentMover.cs

6.75 Infohazard.HyperNav.Jobs.Utility.ReadOnlyArrayPointer< T > Struct Template Reference

Reference over a native array, which does not allow the memory to be modified.

6.75.1 Detailed Description

Reference over a native array, which does not allow the memory to be modified.

Template Parameters

T	Element type.

Type Constraints

T* : *unmanaged

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Utility/ReadOnlyArrayPointer.cs

6.76 Infohazard.HyperNav.Jobs.Baking.Surface.RemoveJaggedTrianglesJob Struct Reference

Remove triangles that have a vertex that is not part of any other triangle. This cleans up the edges of the mesh, removing jagged edges.

6.76.1 Detailed Description

Remove triangles that have a vertex that is not part of any other triangle. This cleans up the edges of the mesh, removing jagged edges.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Baking/Surface/RemoveJaggedTrianglesJob.cs

6.77 Infohazard.HyperNav.RigidbodyAvoidanceObstacle Class Reference

An IAvoidanceObstacle that gets its IAvoidanceObstacle.InputVelocity from a Rigidbody.

Public Member Functions

- virtual void Reset ()
  Set _rigidbody.

Properties

- Rigidbody Rigidbody `[get, set]`
  Rigidbody to get the velocity from.
- override Vector3 InputVelocity `[get]`
  The object's desired (or actual) velocity.

Additional Inherited Members

6.77.1 Detailed Description

An IAvoidanceObstacle that gets its IAvoidanceObstacle.InputVelocity from a Rigidbody.

6.77.2 Member Function Documentation

6.77.2.1 Reset() `virtual void Infohazard.HyperNav.RigidbodyAvoidanceObstacle.Reset ( ) [virtual]`

Set _rigidbody.

6.77.3 Property Documentation

6.77.3.1 InputVelocity `override Vector3 Infohazard.HyperNav.RigidbodyAvoidanceObstacle.InputVelocity [get]`

The object's desired (or actual) velocity.

Implements Infohazard.HyperNav.IAvoidanceObstacle.

6.77.3.2 Rigidbody `Rigidbody Infohazard.HyperNav.RigidbodyAvoidanceObstacle.Rigidbody [get], [set]`

Rigidbody to get the velocity from.

The documentation for this class was generated from the following file:

- Runtime/Avoidance/RigidbodyAvoidanceObstacle.cs

6.78 Infohazard.HyperNav.SimpleAvoidanceObstacle Class Reference

An IAvoidanceObstacle that gets its IAvoidanceObstacle.InputVelocity by measuring its position/time delta.

Protected Member Functions

- override void OnEnable ()

  Resets desired velocity and adds self to list of all obstacles.
- virtual void LateUpdate ()

  Update computed velocity.

Properties

- override Vector3 InputVelocity `[get]`
  The object's desired (or actual) velocity.

6.78.1 Detailed Description

An IAvoidanceObstacle that gets its IAvoidanceObstacle.InputVelocity by measuring its position/time delta.

6.78.2 Member Function Documentation

6.78.2.1 LateUpdate() `virtual void Infohazard.HyperNav.SimpleAvoidanceObstacle.LateUpdate ( ) [protected], [virtual]`

Update computed velocity.

6.78.2.2 OnEnable() `override void Infohazard.HyperNav.SimpleAvoidanceObstacle.OnEnable ( ) [protected], [virtual]`

Resets desired velocity and adds self to list of all obstacles.

Reimplemented from Infohazard.HyperNav.AvoidanceObstacleBase.

6.78.3 Property Documentation

6.78.3.1 InputVelocity `override Vector3 Infohazard.HyperNav.SimpleAvoidanceObstacle.InputVelocity [get]`

The object's desired (or actual) velocity.

Implements Infohazard.HyperNav.IAvoidanceObstacle.

The documentation for this class was generated from the following file:

- Runtime/Avoidance/SimpleAvoidanceObstacle.cs

6.79 Infohazard.HyperNav.SimpleNavAgentMover Class Reference

A script that enables using a NavAgent without writing any code.

Classes

- class PropNames

  This is used to refer to the names of private fields in this class from a custom Editor.

Public Types

- enum RotateModeType

  Used to specify the modes for rotating the agent.

Protected Member Functions

- void OnEnable ()

  Reset state so next Update will calculate a new path.
- virtual void Update ()

  Update the path if needed, and move the agent if not using a Rigidbody.
- virtual void FixedUpdate ()

  Move the agent if using a Rigidbody.
- virtual void Reset ()

  Get references to components.
- virtual void UpdatePath ()

  Calculate a new path to DestinationTransform.
- virtual bool CheckRepath ()

  Check if a new path should be calculated.
- virtual void UpdateMovementTransform ()

  Update the agent's position, rotation, and velocity based on the desired velocity.
- virtual void UpdateMovementRigidbody ()

  Update the Rigidbody's velocity and rotation based on the desired velocity.
- virtual Quaternion GetNewRotation (Quaternion current, Vector3 velocity, float deltaTime)

  Get the updated rotation based on the desired velocity and time delta.
- virtual Quaternion InterpolateRotation (Quaternion from, Quaternion to, float delta)

  Interpolate between two rotations using either RotateTowards or Slerp.

Protected Attributes

- PassiveTimer RepathTimer

  Timer for repathing at intervals.
- Vector3 KinematicVelocity

  Current velocity. Not used if a Rigidbody is used.
- Vector3 LastDestination

  Last destination set.

Properties

- NavAgent NavAgent `[get, set]`

  The NavAgent that calculates the path.
- Rigidbody Rigidbody `[get, set]`

  If set, movement will be handled by this Rigidbody.
- Transform DestinationTransform `[get, set]`

  If set, the agent will move to this destination.
- float SampleRadius `[get, set]`

  The radius to sample around the agent for volumes.
- bool EnableRepathing `[get, set]`

  Enable automatically calculating a new path to the destination.
- bool RepathAtInterval `[get, set]`

  If enabled, a new path will be calculated at a set interval.
- float RepathInterval `[get, set]`

  Interval to calculate a new path.
- bool RepathOnDestinationTransformMove `[get, set]`

  If enabled, a new path will be calculated when the destination moves.
- float RepathDistanceThreshold `[get, set]`

  Distance threshold to repath when the destination moves.
- bool RepathOnReachEnd `[get, set]`

  Enable to repath when the agent nears the end of its path.
- float RepathOnReachEndDistance `[get, set]`

  Distance from the end of the path to repath.
- float MaxSpeedInVolume `[get, set]`

  The maximum speed the agent can move in units/second when in a volume.
- float AccelerationInVolume `[get, set]`

  The acceleration of the agent in units/second^2 when in a volume.
- float MaxSpeedOnSurface `[get, set]`

  The maximum speed the agent can move in units/second when on a surface.
- float AccelerationOnSurface `[get, set]`

  The acceleration of the agent in units/second^2 when on a surface.
- RotateModeType RotateMode `[get, set]`

  How the agent's rotation is determined.
- float RotationSpeed `[get, set]`

  The speed at which the agent rotates in degrees/second.
- bool RotateBySlerp `[get, set]`

  If enabled, rotation will be done with Slerp instead of RotateTowards.
- float DistanceToRotateToSurfaceUp `[get, set]`

  Once within this distance of a surface while flying, the agent will rotate to align with the surface normal.
- bool KeepAlignedToGroundNormal `[get, set]`

  When moving on a surface, keep the agent aligned to the ground normal.
- bool KeepOnGround `[get, set]`

  When moving on a surface, keep the agent on the ground by raycasting and moving if necessary.
- LayerMask GroundCheckLayerMask `[get, set]`

  Layer mask to use for ground check.
- float GroundCheckQueryRadius `[get, set]`

  Radius to use for ground check.
- float GroundCheckDistance `[get, set]`

  Distance to check for ground.
- float GroundCheckNormalAngle `[get, set]`

  When aligning to ground, ground normal must be within this angle of up vector.
- float DesiredOffsetToGround `[get, set]`

  Desired vertical offset from the center of this agent to the ground.

Private Member Functions

- void OnDisable ()

  Unsubscribe from events.

6.79.1 Detailed Description

A script that enables using a NavAgent without writing any code.

It is completely optional to use this script, and for more advanced uses you will likely need to write your own code to handle the movement.

6.79.2 Member Enumeration Documentation

6.79.2.1 RotateModeType `enum Infohazard.HyperNav.SimpleNavAgentMover.RotateModeType`

Used to specify the modes for rotating the agent.

6.79.3 Member Function Documentation

6.79.3.1 CheckRepath() `virtual bool Infohazard.HyperNav.SimpleNavAgentMover.CheckRepath ( ) [protected], [virtual]`

Check if a new path should be calculated.

Returns

True if a new path should be calculated.

6.79.3.2 FixedUpdate() `virtual void Infohazard.HyperNav.SimpleNavAgentMover.FixedUpdate ( ) [protected], [virtual]`

Move the agent if using a Rigidbody.

6.79.3.3 GetNewRotation() `virtual Quaternion Infohazard.HyperNav.SimpleNavAgentMover.GetNewRotation ( Quaternion current, Vector3 velocity, float deltaTime ) [protected], [virtual]`

Get the updated rotation based on the desired velocity and time delta.

Parameters

current	Current rotation.
velocity	Current velocity.
deltaTime	Time since last rotation update.

Returns

New rotation to use.

6.79.3.4 InterpolateRotation()

```
virtual Quaternion Infohazard.HyperNav.SimpleNavAgentMover.↩
InterpolateRotation (
            Quaternion from,
            Quaternion to,
            float delta )  [protected], [virtual]
```

Interpolate between two rotations using either RotateTowards or Slerp.

Parameters

from	Source rotation.
to	Target rotation.
delta	Delta value (meaning depends on interpolation method).

Returns

Interpolated rotation.

6.79.3.5 OnDisable()

```
void Infohazard.HyperNav.SimpleNavAgentMover.OnDisable ( )  [private]
```

Unsubscribe from events.

6.79.3.6 OnEnable()

```
void Infohazard.HyperNav.SimpleNavAgentMover.OnEnable ( )  [protected]
```

Reset state so next Update will calculate a new path.

6.79.3.7 Reset()

```
virtual void Infohazard.HyperNav.SimpleNavAgentMover.Reset ( )  [protected],
[virtual]
```

Get references to components.

6.79.3.8 Update() virtual void Infohazard.HyperNav.SimpleNavAgentMover.Update ( ) [protected], [virtual]

Update the path if needed, and move the agent if not using a Rigidbody.

6.79.3.9 UpdateMovementRigidbody() virtual void Infohazard.HyperNav.SimpleNavAgentMover.↩ UpdateMovementRigidbody ( ) [protected], [virtual]

Update the Rigidbody's velocity and rotation based on the desired velocity.

Should only be called in FixedUpdate and only if using a Rigidbody.

6.79.3.10 UpdateMovementTransform() virtual void Infohazard.HyperNav.SimpleNavAgentMover.↩ UpdateMovementTransform ( ) [protected], [virtual]

Update the agent's position, rotation, and velocity based on the desired velocity.

Should only be called in Update and only if not using a Rigidbody.

6.79.3.11 UpdatePath() virtual void Infohazard.HyperNav.SimpleNavAgentMover.UpdatePath ( ) [protected], [virtual]

Calculate a new path to DestinationTransform.

6.79.4 Member Data Documentation

6.79.4.1 KinematicVelocity Vector3 Infohazard.HyperNav.SimpleNavAgentMover.KinematicVelocity [protected]

Current velocity. Not used if a Rigidbody is used.

6.79.4.2 LastDestination Vector3 Infohazard.HyperNav.SimpleNavAgentMover.LastDestination [protected]

Last destination set.

6.79.4.3 RepathTimer PassiveTimer Infohazard.HyperNav.SimpleNavAgentMover.RepathTimer [protected]

Timer for repathing at intervals.

6.79.5 Property Documentation

6.79.5.1 AccelerationInVolume

`float Infohazard.HyperNav.SimpleNavAgentMover.AccelerationInVolume [get], [set]`

The acceleration of the agent in units/second^2 when in a volume.

6.79.5.2 AccelerationOnSurface

`float Infohazard.HyperNav.SimpleNavAgentMover.AccelerationOnSurface [get], [set]`

The acceleration of the agent in units/second^2 when on a surface.

6.79.5.3 DesiredOffsetToGround

`float Infohazard.HyperNav.SimpleNavAgentMover.DesiredOffsetToGround [get], [set]`

Desired vertical offset from the center of this agent to the ground.

6.79.5.4 DestinationTransform

`Transform Infohazard.HyperNav.SimpleNavAgentMover.DestinationTransform [get], [set]`

If set, the agent will move to this destination.

6.79.5.5 DistanceToRotateToSurfaceUp

`float Infohazard.HyperNav.SimpleNavAgentMover.DistanceToRotateToSurfaceUp [get], [set]`

Once within this distance of a surface while flying, the agent will rotate to align with the surface normal.

6.79.5.6 EnableRepathing

`bool Infohazard.HyperNav.SimpleNavAgentMover.EnableRepathing [get], [set]`

Enable automatically calculating a new path to the destination.

6.79.5.7 GroundCheckDistance

`float Infohazard.HyperNav.SimpleNavAgentMover.GroundCheckDistance [get], [set]`

Distance to check for ground.

6.79.5.8 GroundCheckLayerMask

`LayerMask Infohazard.HyperNav.SimpleNavAgentMover.GroundCheckLayerMask [get], [set]`

Layer mask to use for ground check.

6.79.5.9 GroundCheckNormalAngle

`float Infohazard.HyperNav.SimpleNavAgentMover.GroundCheckNormalAngle [get], [set]`

When aligning to ground, ground normal must be within this angle of up vector.

6.79.5.10 GroundCheckQueryRadius

`float Infohazard.HyperNav.SimpleNavAgentMover.GroundCheckQueryRadius [get], [set]`

Radius to use for ground check.

6.79.5.11 KeepAlignedToGroundNormal

`bool Infohazard.HyperNav.SimpleNavAgentMover.KeepAlignedToGroundNormal [get], [set]`

When moving on a surface, keep the agent aligned to the ground normal.

6.79.5.12 KeepOnGround

`bool Infohazard.HyperNav.SimpleNavAgentMover.KeepOnGround [get], [set]`

When moving on a surface, keep the agent on the ground by raycasting and moving if necessary.

6.79.5.13 MaxSpeedInVolume

`float Infohazard.HyperNav.SimpleNavAgentMover.MaxSpeedInVolume [get], [set]`

The maximum speed the agent can move in units/second when in a volume.

6.79.5.14 MaxSpeedOnSurface

`float Infohazard.HyperNav.SimpleNavAgentMover.MaxSpeedOnSurface [get], [set]`

The maximum speed the agent can move in units/second when on a surface.

6.79.5.15 NavAgent `NavAgent Infohazard.HyperNav.SimpleNavAgentMover.NavAgent [get], [set]`

The NavAgent that calculates the path.

6.79.5.16 RepathAtInterval `bool Infohazard.HyperNav.SimpleNavAgentMover.RepathAtInterval [get], [set]`

If enabled, a new path will be calculated at a set interval.

6.79.5.17 RepathDistanceThreshold `float Infohazard.HyperNav.SimpleNavAgentMover.RepathDistanceThreshold [get], [set]`

Distance threshold to repath when the destination moves.

6.79.5.18 RepathInterval `float Infohazard.HyperNav.SimpleNavAgentMover.RepathInterval [get], [set]`

Interval to calculate a new path.

6.79.5.19 RepathOnDestinationTransformMove `bool Infohazard.HyperNav.SimpleNavAgentMover.RepathOnDestinationTransformMove [get], [set]`

If enabled, a new path will be calculated when the destination moves.

6.79.5.20 RepathOnReachEnd `bool Infohazard.HyperNav.SimpleNavAgentMover.RepathOnReachEnd [get], [set]`

Enable to repath when the agent nears the end of its path.

6.79.5.21 RepathOnReachEndDistance `float Infohazard.HyperNav.SimpleNavAgentMover.RepathOnReachEndDistance [get], [set]`

Distance from the end of the path to repath.

6.79.5.22 Rigidbody `Rigidbody Infohazard.HyperNav.SimpleNavAgentMover.Rigidbody [get], [set]`

If set, movement will be handled by this Rigidbody.

6.79.5.23 RotateBySlerp `bool Infohazard.HyperNav.SimpleNavAgentMover.RotateBySlerp [get], [set]`

If enabled, rotation will be done with Slerp instead of RotateTowards.

6.79.5.24 RotateMode `RotateModeType Infohazard.HyperNav.SimpleNavAgentMover.RotateMode [get], [set]`

How the agent's rotation is determined.

6.79.5.25 RotationSpeed `float Infohazard.HyperNav.SimpleNavAgentMover.RotationSpeed [get], [set]`

The speed at which the agent rotates in degrees/second.

6.79.5.26 SampleRadius `float Infohazard.HyperNav.SimpleNavAgentMover.SampleRadius [get], [set]`

The radius to sample around the agent for volumes.

The documentation for this class was generated from the following file:

- Runtime/SimpleNavAgentMover.cs

6.80 Infohazard.HyperNav.SplineNavAgent Class Reference

A script that can be used to calculate smooth paths by any entity that needs to use HyperNav for navigation.

Public Member Functions

- override void Stop (bool abortPaths)

  Stop following the current path, and optionally cancel all path requests.

Parameters

abortPaths	*Whether to cancel pending path requests.*

Protected Member Functions

- override void Update ()

  Updates measured velocity and finds the nearest point on the spline.
- override void OnDrawGizmos ()

  Draws the current spline if NavAgent.DebugPath is true.
- override Vector3 CalculateDesiredNavigationVelocity ()

  Calculate the velocity the agent wants to move in, in the range [0, 1].
- override void OnDisable ()

  Stops all pathfinding and cancels path requests.

Properties

- float TangentScale `[get, set]`

  Scale to apply to spline tangents (lower values make the spline less curvy).
- bool RaycastTangents `[get, set]`

  Whether to shorten tangents by raycasting to ensure they don't penetrate blocked areas.
- int DistanceSamplesPerSegment `[get, set]`

  How many samples to take per segment of the spline when mapping the distance.
- int DebugPointCount `[get, set]`

  If NavAgent.DebugPath is enabled, how many points to use to draw the curve.
- float MaxAlignmentVelocityDistance `[get, set]`

  At what distance from the spline the agent will have all its desired velocity devoted to returning.
- float CurvatureSampleDistance `[get, set]`

  Delta-T to use when sampling curvature (should be quite small).
- float CurvatureOfMaxSlowdown `[get, set]`

  At what curvature value is the agent at its max curvature slowdown.
- float MaxCurvatureSlowdown `[get, set]`

  The multiplier on desired tangent velocity when at the max curvature value.
- bool DebugProjectOnSpline `[get, set]`

  Whether to draw debug lines when projecting on the spline.
- float BlockedDetectionDistance `[get, set]`

  Distance in front of the agent to check to see if it needs to avoid level geometry.
- float BlockedDetectionBackDistance `[get, set]`

  Distance behind the agent to check to see if it needs to avoid level geometry.
- float BlockedDetectionMinSplineDistance `[get, set]`

  How far the agent must be from the spline to check for blocking level geometry.
- bool BlockedDetectionPhysicsQuery `[get, set]`

  Whether to use physics queries to detect blocking level geometry.
- Rigidbody BlockedDetectionPhysicsQueryRigidbody `[get, set]`

  Rigidbody to use for physics queries to detect blocking level geometry.
- bool IsOnSpline `[get, private set]`

  Whether the agent is currently traveling along a spline.
- SplinePath SplinePath `[get]`

  The spline that the agent is currently following.
- float CurrentSplineParameter `[get]`

  The spline parameter value the agent is nearest to on the spline.
- float CurrentSplineDistance `[get]`

  The distance along the spline the agent is nearest to.
- float MaxSplineDistance `[get]`

  The length of the agent's current spline path.
- float RemainingDistanceAfterSpline `[get, private set]`

  The remaining distance along the path after the end of the current spline.
- int PathIndexOfSplineEnd `[get, private set]`

  The index of the end of the current spline in the current path.

Additional Inherited Members

6.80.1 Detailed Description

A script that can be used to calculate smooth paths by any entity that needs to use HyperNav for navigation.

A SplineNavAgent works just like (and is a subclass of) NavAgent. However, a SplineNavAgent feeds the path waypoints into a spline function to get a smoother path. The spline-based movement is only active during volume-based navigation; surface navigation will use the default point-based path. In order to follow the spline, the SplineNavAgent creates its NavAgent.DesiredVelocity based on two factors:

- Tangent: The direction the spline is pointing nearest the agent.
- Alignment: The direction from the agent to the nearest point on the spline.f

The agent will increase the influence of the alignment velocity the further it gets from the spline, in order to prevent drifting too far off. Additionally, the agent can slow down its desired tangent velocity on high-curvature regions, in order to take the curves more slowly.

6.80.2 Member Function Documentation

6.80.2.1 CalculateDesiredNavigationVelocity() `override Vector3 Infohazard.HyperNav.SplineNav↩ Agent.CalculateDesiredNavigationVelocity ( ) [protected], [virtual]`

Calculate the velocity the agent wants to move in, in the range [0, 1].

Reimplemented from Infohazard.HyperNav.NavAgent.

6.80.2.2 OnDisable() `override void Infohazard.HyperNav.SplineNavAgent.OnDisable ( ) [protected], [virtual]`

Stops all pathfinding and cancels path requests.

Reimplemented from Infohazard.HyperNav.NavAgent.

6.80.2.3 OnDrawGizmos() `override void Infohazard.HyperNav.SplineNavAgent.OnDrawGizmos ( ) [protected], [virtual]`

Draws the current spline if NavAgent.DebugPath is true.

The spline will be drawn with DebugPointCount points.

Reimplemented from Infohazard.HyperNav.NavAgent.

6.80.2.4 Stop() `override void Infohazard.HyperNav.SplineNavAgent.Stop ( bool abortPaths ) [virtual]`

Stop following the current path, and optionally cancel all path requests.

Parameters

abortPaths	Whether to cancel pending path requests.

Reimplemented from Infohazard.HyperNav.NavAgent.

6.80.2.5 Update() `override void Infohazard.HyperNav.SplineNavAgent.Update ( ) [protected], [virtual]`

Updates measured velocity and finds the nearest point on the spline.

Reimplemented from Infohazard.HyperNav.NavAgent.

6.80.3 Property Documentation

6.80.3.1 BlockedDetectionBackDistance `float Infohazard.HyperNav.SplineNavAgent.BlockedDetection↩ BackDistance [get], [set]`

Distance behind the agent to check to see if it needs to avoid level geometry.

6.80.3.2 BlockedDetectionDistance `float Infohazard.HyperNav.SplineNavAgent.BlockedDetection↩ Distance [get], [set]`

Distance in front of the agent to check to see if it needs to avoid level geometry.

6.80.3.3 BlockedDetectionMinSplineDistance `float Infohazard.HyperNav.SplineNavAgent.Blocked↩ DetectionMinSplineDistance [get], [set]`

How far the agent must be from the spline to check for blocking level geometry.

6.80.3.4 BlockedDetectionPhysicsQuery `bool Infohazard.HyperNav.SplineNavAgent.BlockedDetection↩ PhysicsQuery [get], [set]`

Whether to use physics queries to detect blocking level geometry.

6.80.3.5 BlockedDetectionPhysicsQueryRigidbody

`Rigidbody Infohazard.HyperNav.SplineNavAgent.BlockedDetectionPhysicsQueryRigidbody [get], [set]`

Rigidbody to use for physics queries to detect blocking level geometry.

6.80.3.6 CurrentSplineDistance

`float Infohazard.HyperNav.SplineNavAgent.CurrentSplineDistance [get]`

The distance along the spline the agent is nearest to.

6.80.3.7 CurrentSplineParameter

`float Infohazard.HyperNav.SplineNavAgent.CurrentSplineParameter [get]`

The spline parameter value the agent is nearest to on the spline.

6.80.3.8 CurvatureOfMaxSlowdown

`float Infohazard.HyperNav.SplineNavAgent.CurvatureOfMaxSlowdown [get], [set]`

At what curvature value is the agent at its max curvature slowdown.

6.80.3.9 CurvatureSampleDistance

`float Infohazard.HyperNav.SplineNavAgent.CurvatureSampleDistance [get], [set]`

Delta-T to use when sampling curvature (should be quite small).

6.80.3.10 DebugPointCount

`int Infohazard.HyperNav.SplineNavAgent.DebugPointCount [get], [set]`

If NavAgent.DebugPath is enabled, how many points to use to draw the curve.

6.80.3.11 DebugProjectOnSpline

`bool Infohazard.HyperNav.SplineNavAgent.DebugProjectOnSpline [get], [set]`

Whether to draw debug lines when projecting on the spline.

6.80.3.12 DistanceSamplesPerSegment

`int Infohazard.HyperNav.SplineNavAgent.DistanceSamplesPerSegment [get], [set]`

How many samples to take per segment of the spline when mapping the distance.

6.80.3.13 IsOnSpline

`bool Infohazard.HyperNav.SplineNavAgent.IsOnSpline [get], [private set]`

Whether the agent is currently traveling along a spline.

6.80.3.14 MaxAlignmentVelocityDistance

`float Infohazard.HyperNav.SplineNavAgent.MaxAlignmentVelocityDistance [get], [set]`

At what distance from the spline the agent will have all its desired velocity devoted to returning.

6.80.3.15 MaxCurvatureSlowdown

`float Infohazard.HyperNav.SplineNavAgent.MaxCurvatureSlowdown [get], [set]`

The multiplier on desired tangent velocity when at the max curvature value.

6.80.3.16 MaxSplineDistance

`float Infohazard.HyperNav.SplineNavAgent.MaxSplineDistance [get]`

The length of the agent's current spline path.

6.80.3.17 PathIndexOfSplineEnd

`int Infohazard.HyperNav.SplineNavAgent.PathIndexOfSplineEnd [get], [private set]`

The index of the end of the current spline in the current path.

6.80.3.18 RaycastTangents

`bool Infohazard.HyperNav.SplineNavAgent.RaycastTangents [get], [set]`

Whether to shorten tangents by raycasting to ensure they don't penetrate blocked areas.

6.80.3.19 RemainingDistanceAfterSpline `float Infohazard.HyperNav.SplineNavAgent.RemainingDistanceAfterSpline [get], [private set]`

The remaining distance along the path after the end of the current spline.

6.80.3.20 SplinePath `SplinePath Infohazard.HyperNav.SplineNavAgent.SplinePath [get]`

The spline that the agent is currently following.

6.80.3.21 TangentScale `float Infohazard.HyperNav.SplineNavAgent.TangentScale [get], [set]`

Scale to apply to spline tangents (lower values make the spline less curvy).

The documentation for this class was generated from the following file:

- Runtime/Spline/SplineNavAgent.cs

6.81 Infohazard.HyperNav.SplinePath Struct Reference

A spline specialized for path following, created with a NavPath.

Public Member Functions

- SplinePath (NavPath path, int start, int end, float tangentScale, int sampleCount, bool raycastTangents)
  Create a new SplinePath with the given path.
- void Initialize (NavPath path, int start, int end, float tangentScale, int sampleCount, bool raycastTangents)
  Re-initialize an existing SplinePath with the given path.
- void Dispose ()
  Dispose arrays allocated for this spline path.
- float GetDistance (float parameter)
  Get the distance along the spline for a given parameter value.
- float GetParameter (float distance)
  Get the parameter value for a given distance along the spline.
- Vector3 GetControlPosition (int index)
  Get the position of a given control point.
- Vector3 GetControlTangent (int index)
  Get the tangent of a given control point.
- NavVolume GetVolume (float parameter)
  Get the NavVolume that contains the given parameter value on the spline.
- Vector3 GetPosition (float parameter)
  Get the position at a given parameter value.
- Vector3 GetTangent (float parameter)
  Get the tangent at a given parameter value.
- Vector3 GetCurvature (float parameter, float offset=0.01f)
  Sample the curvature at a given parameter value.
- float ProjectPosition (Vector3 position, int iterations=5, bool debug=false)
  Approximate the parameter value of the position along the spline nearest to the given position.

Properties

- float Length `[get, private set]`

  Length of the spline in world units.
- int PointCount `[get, private set]`

  Number of control points on the spline.
- bool IsCreated `[get, private set]`

  Whether an actual spline has been constructed.
- NativeArray< SplinePoint > ControlPoints `[get]`

  List of all the control points of the spline.

6.81.1 Detailed Description

A spline specialized for path following, created with a NavPath.

Unlike most spline tools, the tangents in this spline are calculated automatically.

This spline implementation uses two coordinate spaces: parameter and distance. Distance ranges from zero to the length of the spline, and values are distributed (approximately) evenly. Parameter ranges from zero to one and is the actual value supplied to the spline function, but values are not distributed evenly.

6.81.2 Constructor & Destructor Documentation

6.81.2.1 SplinePath()

```
Infohazard.HyperNav.SplinePath.SplinePath (
        NavPath path,
        int start,
        int end,
        float tangentScale,
        int sampleCount,
        bool raycastTangents )
```

Create a new SplinePath with the given path.

Parameters

path	The input navigation path.
start	Waypoint index of the start of the path.
end	Waypoint index of the end of the path.
tangentScale	Scale to apply to spline tangents (lower values make the spline less curvy).
sampleCount	How many samples to take per segment of the spline when mapping the distance.
raycastTangents	Whether to shorten tangents by raycasting against NavVolume blocking triangles.

6.81.3 Member Function Documentation

6.81.3.1 Dispose() `void Infohazard.HyperNav.SplinePath.Dispose ( )`

Dispose arrays allocated for this spline path.

6.81.3.2 GetControlPosition() `Vector3 Infohazard.HyperNav.SplinePath.GetControlPosition ( int index )`

Get the position of a given control point.

Parameters

index	Control point index.

Returns

Position of that control point, in world space.

6.81.3.3 GetControlTangent() `Vector3 Infohazard.HyperNav.SplinePath.GetControlTangent ( int index )`

Get the tangent of a given control point.

Parameters

index	Control point index.

Returns

Tangent of that control point, in world space.

6.81.3.4 GetCurvature() `Vector3 Infohazard.HyperNav.SplinePath.GetCurvature ( float parameter, float offset = 0.01f )`

Sample the curvature at a given parameter value.

Unlike GetPosition and GetTangent, this does not return an exact value.

Parameters

parameter	The parameter value in range [0, 1].
offset	Offset distance to sample derivative of tangent function.

Returns

The sampled curvature value (use magnitude to get scalar curvature).

6.81.3.5 GetDistance() `float Infohazard.HyperNav.SplinePath.GetDistance ( float parameter )`

Get the distance along the spline for a given parameter value.

Parameters

parameter	The parameter value in range [0, 1].

Returns

The distance value in range [0, Length].

6.81.3.6 GetParameter() `float Infohazard.HyperNav.SplinePath.GetParameter ( float distance )`

Get the parameter value for a given distance along the spline.

Parameters

distance	The distance value in range [0, Length].

Returns

The parameter value in range [0, 1].

6.81.3.7 GetPosition() `Vector3 Infohazard.HyperNav.SplinePath.GetPosition ( float parameter )`

Get the position at a given parameter value.

Parameters

parameter	The parameter value in range [0, 1].

Returns

Position along the spline, in world space.

6.81.3.8 GetTangent() `Vector3 Infohazard.HyperNav.SplinePath.GetTangent (`

`float parameter )`

Get the tangent at a given parameter value.

Parameters

parameter	The parameter value in range [0, 1].

Returns

Tangent at that position, in world space.

6.81.3.9 GetVolume() `NavVolume Infohazard.HyperNav.SplinePath.GetVolume (`

`float parameter )`

Get the NavVolume that contains the given parameter value on the spline.

Parameters

parameter	Input parameter value.

Returns

The containing NavVolume.

6.81.3.10 Initialize() `void Infohazard.HyperNav.SplinePath.Initialize (`

```
NavPath path,
int start,
int end,
float tangentScale,
int sampleCount,
bool raycastTangents )
```

Re-initialize an existing SplinePath with the given path.

Parameters

path	The input navigation path.
start	Waypoint index of the start of the path.
end	Waypoint index of the end of the path.
tangentScale	Scale to apply to spline tangents (lower values make the spline less curvy).
sampleCount	How many samples to take per segment of the spline when mapping the distance.
raycastTangents	Whether to shorten tangents by raycasting against NavVolume blocking triangles.

6.81.3.11 ProjectPosition() `float Infohazard.HyperNav.SplinePath.ProjectPosition (`

```
Vector3 position,
int iterations = 5,
bool debug = false )
```

Approximate the parameter value of the position along the spline nearest to the given position.

This uses Newton's method. Increasing the iteration count increases both accuracy and cost.

Parameters

position	Position to project.
iterations	Number of Newton's method iterations.
debug	Whether to draw debug lines showing Newton's method iterations.

Returns

The approximate parameter along the spline in range [0, 1].

6.81.4 Property Documentation

6.81.4.1 ControlPoints `NativeArray<SplinePoint> Infohazard.HyperNav.SplinePath.ControlPoints [get]`

List of all the control points of the spline.

6.81.4.2 IsCreated `bool Infohazard.HyperNav.SplinePath.IsCreated [get], [private set]`

Whether an actual spline has been constructed.

6.81.4.3 Length `float Infohazard.HyperNav.SplinePath.Length [get], [private set]`

Length of the spline in world units.

6.81.4.4 PointCount `int Infohazard.HyperNav.SplinePath.PointCount [get], [private set]`

Number of control points on the spline.

The documentation for this struct was generated from the following file:

- Runtime/Spline/SplinePath.cs

6.82 Infohazard.HyperNav.SplinePoint Struct Reference

Represents a point on a spline and the segment that starts with it.

Public Attributes

- float4 Position

  Position of the control point.
- float4 Tangent

  Tangent of the control point.
- float4x4 PositionMatrix

  Matrix to multiply by a time vector along the segment to get a position.
- float4x4 TangentMatrix

  Matrix to multiply by a time vector along the segment to get a tangent.
- long FromVolume

  Volume that leads to the control point.
- long ToVolume

  Volume that the control point leads to.

6.82.1 Detailed Description

Represents a point on a spline and the segment that starts with it.

6.82.2 Member Data Documentation

6.82.2.1 FromVolume `long Infohazard.HyperNav.SplinePoint.FromVolume`

Volume that leads to the control point.

6.82.2.2 Position `float4 Infohazard.HyperNav.SplinePoint.Position`

Position of the control point.

6.82.2.3 PositionMatrix `float4x4 Infohazard.HyperNav.SplinePoint.PositionMatrix`

Matrix to multiply by a time vector along the segment to get a position.

6.82.2.4 Tangent `float4 Infohazard.HyperNav.SplinePoint.Tangent`

Tangent of the control point.

6.82.2.5 TangentMatrix `float4x4 Infohazard.HyperNav.SplinePoint.TangentMatrix`

Matrix to multiply by a time vector along the segment to get a tangent.

6.82.2.6 ToVolume `long Infohazard.HyperNav.SplinePoint.ToVolume`

Volume that the control point leads to.

The documentation for this struct was generated from the following file:

- Runtime/Spline/SplinePath.cs

6.83 Infohazard.HyperNav.Jobs.SplineProjectJob Struct Reference

Job used to find the parameter along a spline that is nearest to the given point.

Public Member Functions

- void Execute ()
  Execute the job.

Public Attributes

- SplinePath Spline
  Spline to query.
- float4 Position
  Position to find nearest parameter to.
- int Iterations
  How many iterations of Newton's Method to perform.
- bool DebugProjection
  Whether to draw debug lines showing each iteration of Newton's Method.
- NativeArray< float > OutPosition
  Where the calculated nearest parameter value is written.

6.83.1 Detailed Description

Job used to find the parameter along a spline that is nearest to the given point.

6.83.2 Member Function Documentation

6.83.2.1 Execute() `void Infohazard.HyperNav.Jobs.SplineProjectJob.Execute ( )`

Execute the job.

6.83.3 Member Data Documentation

6.83.3.1 DebugProjection `bool Infohazard.HyperNav.Jobs.SplineProjectJob.DebugProjection`

Whether to draw debug lines showing each iteration of Newton's Method.

6.83.3.2 Iterations `int Infohazard.HyperNav.Jobs.SplineProjectJob.Iterations`

How many iterations of Newton's Method to perform.

6.83.3.3 OutPosition `NativeArray<float> Infohazard.HyperNav.Jobs.SplineProjectJob.OutPosition`

Where the calculated nearest parameter value is written.

6.83.3.4 Position `float4 Infohazard.HyperNav.Jobs.SplineProjectJob.Position`

Position to find nearest parameter to.

6.83.3.5 Spline `SplinePath Infohazard.HyperNav.Jobs.SplineProjectJob.Spline`

Spline to query.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/SplineProjectJob.cs

6.84 Infohazard.HyperNav.SurfaceUprightDirectionHandlerWithOverrides Class Reference

Serves as an example of how to implement a custom upright direction handler for surfaces, including the related jobs.

Public Member Functions

- async UniTask< NavSurfaceBakeData > CalculateUprightDirections (NavSurfaceBakeData bakeData, NativeArray< RaycastCommand > queries, int resultsPerQuery, NativeArray< RaycastHit > results, NativeCancellationToken cancellationToken)

  Calculate the upright directions for each triangle of the surface, and store the results in NavSurfaceBakeData.Nav↩ SurfaceBakeData.MeshInfo.BakingNavSurfaceMeshInfo.TriangleUprightWorldDirections.

6.84.1 Detailed Description

Serves as an example of how to implement a custom upright direction handler for surfaces, including the related jobs.

6.84.2 Member Function Documentation

6.84.2.1 CalculateUprightDirections()

```
async UniTask< NavSurfaceBakeData > Infohazard.HyperNav.↩
SurfaceUprightDirectionHandlerWithOverrides.CalculateUprightDirections (
            NavSurfaceBakeData bakeData,
            NativeArray< RaycastCommand > queries,
            int resultsPerQuery,
            NativeArray< RaycastHit > results,
            NativeCancellationToken cancellationToken )
```

Calculate the upright directions for each triangle of the surface, and store the results in NavSurfaceBakeData.Nav↩ SurfaceBakeData.MeshInfo.BakingNavSurfaceMeshInfo.TriangleUprightWorldDirections.

This should be done using a parallel job if possible. Any triangles that fail custom checks should be added to NavSurfaceBakeData.NavSurfaceBakeData.FilterData's NavSurfaceFilterData.TrianglesToSplit or NavSurface↩ FilterData.TrianglesToRemove.

Parameters

bakeData	The in-progress data for the surface.
cancellationToken	Cancellation token for the operation.
queries	The raycast queries that were performed.
resultsPerQuery	How many hits in the results array correspond to each query.
results	The results of the raycast queries.

Returns

The updated bake data (can be the same as the input if only referenced values are updated).

Implements Infohazard.HyperNav.ISurfaceUprightDirectionHandler.

The documentation for this class was generated from the following file:

- Runtime/Utility/SurfaceNormalUprightDirectionHandlerWithOverrides.cs

6.85 Infohazard.HyperNav.Triangle Struct Reference

Represents the indices of a triangle (three vertices by a face) in an indexed mesh.

Public Member Functions

- Triangle (int vertex1, int vertex2, int vertex3)

  Construct a new Triangle with the given indices.
- Triangle (int3 vertices)

  Construct a new Triangle with the given indices.
- bool Contains (int vertex)

  Returns whether the triangle contains the given vertex.
- int3 ToInt3 ()

  Convert to int3.
- override bool Equals (object obj)

  Compare to another object.
- bool Equals (Triangle other)

  Compare to another Triangle.
- override int GetHashCode ()

  Get integer for use with hash table.

Static Public Attributes

- static readonly Triangle InvalidTriangle = new() { _minVertex = -1, _midVertex = -1, _maxVertex = -1 }

  Invalid Triangle.

Properties

- int Vertex1 `[get]`

  First vertex index, which is the lower of the three.
- int Vertex2 `[get]`

  Second vertex index, which is the middle of the three.
- int Vertex3 `[get]`

  Third vertex index, which is the larger of the three.
- bool IsValid `[get]`

  Whether the Triangle is valid.

6.85.1 Detailed Description

Represents the indices of a triangle (three vertices by a face) in an indexed mesh.

The same Triangle will be created regardless of the order in which indices are fed to the constructor.

6.85.2 Constructor & Destructor Documentation

6.85.2.1 Triangle() [1/2] Infohazard.HyperNav.Triangle.Triangle (

```
int vertex1,
int vertex2,
int vertex3 )
```

Construct a new Triangle with the given indices.

The order of the indices doesn't matter; the same Triangle is constructed either way. No two of the indices can be the same.

Parameters

vertex1	First vertex index.
vertex2	Second vertex index.
vertex3	Third vertex index.

6.85.2.2 Triangle() [2/2] Infohazard.HyperNav.Triangle.Triangle (

```
int3 vertices )
```

Construct a new Triangle with the given indices.

Parameters

vertices	Vertex indices.

6.85.3 Member Function Documentation

6.85.3.1 Contains() bool Infohazard.HyperNav.Triangle.Contains (

```
int vertex )
```

Returns whether the triangle contains the given vertex.

Parameters

vertex	Vertex index to check.

Returns

Whether the triangle contains the vertex.

6.85.3.2 Equals() [1/2] override bool Infohazard.HyperNav.Triangle.Equals (object *obj*)

Compare to another object.

Parameters

obj	Object to compare to.

Returns

Whether the two objects are equal.

6.85.3.3 Equals() [2/2] bool Infohazard.HyperNav.Triangle.Equals (Triangle *other*)

Compare to another Triangle.

Parameters

other	Triangle to compare to.

Returns

Whether the two triangles are equal.

6.85.3.4 GetHashCode() override int Infohazard.HyperNav.Triangle.GetHashCode ( )

Get integer for use with hash table.

Returns

Integer hash code.

6.85.3.5 ToInt3() `int3 Infohazard.HyperNav.Triangle.ToInt3 ( )`

Convert to int3.

Returns

Triangle as int3.

6.85.4 Member Data Documentation

6.85.4.1 InvalidTriangle `readonly Triangle Infohazard.HyperNav.Triangle.InvalidTriangle = new() { _minVertex = -1, _midVertex = -1, _maxVertex = -1 } [static]`

Invalid Triangle.

6.85.5 Property Documentation

6.85.5.1 IsValid `bool Infohazard.HyperNav.Triangle.IsValid [get]`

Whether the Triangle is valid.

6.85.5.2 Vertex1 `int Infohazard.HyperNav.Triangle.Vertex1 [get]`

First vertex index, which is the lower of the three.

6.85.5.3 Vertex2 `int Infohazard.HyperNav.Triangle.Vertex2 [get]`

Second vertex index, which is the middle of the three.

6.85.5.4 Vertex3 `int Infohazard.HyperNav.Triangle.Vertex3 [get]`

Third vertex index, which is the larger of the three.

The documentation for this struct was generated from the following file:

- Runtime/Utility/Triangle.cs

6.86 Infohazard.HyperNav.Jobs.Utility.TriangleRegionIndices Struct Reference

Serves as a Dictionary of ints with a max size of 2, to avoid allocations. This works because a triangle can be part of at most two regions.

6.86.1 Detailed Description

Serves as a Dictionary of ints with a max size of 2, to avoid allocations. This works because a triangle can be part of at most two regions.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Utility/NavAreaBakingStructures.cs

6.87 Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T > Struct Template Reference

This is a simple wrapper for unmanaged memory which bypasses Unity's safety checks. This allows arrays to be nested in other arrays (or in structs contained in arrays). Note that you must keep a reference to the original NativeArray, or Unity will detect a memory leak.

Public Member Functions

- unsafe UnsafeArray (int length, Allocator allocator, bool clearMemory=true)

  Allocate a new array of the given length.
- unsafe UnsafeArray (T *ptr, int length, Allocator allocator=Allocator.None)

  Wrap an existing pre-allocated memory block.
- UnsafeArray (IntPtr ptr, int length, Allocator allocator=Allocator.None)

  Wrap an existing pre-allocated memory block.
- unsafe void Dispose ()

  Free the memory if it has been allocated directly.
- Enumerator GetEnumerator ()

  Get an enumerator for the array.

Static Public Member Functions

- static unsafe UnsafeArray< T > ToPointer (in NativeArray< T > array)

  Create a pointer to the given NativeArray.

Public Attributes

- readonly int Length

  Length of the array.
- readonly IntPtr Pointer

  Pointer to the start of the array.
- readonly Allocator Allocator

  Allocator used to allocate the array, or None if it was created from a NativeArray.

Properties

- bool IsNull [get]

  Returns whether the pointer is null.
- int MemorySize [get]

  Returns the size of the memory block in bytes.
- unsafe ref T this[int index] [get]

  Get a reference to the element at the given index (can be used to set values as well).

6.87.1 Detailed Description

This is a simple wrapper for unmanaged memory which bypasses Unity's safety checks. This allows arrays to be nested in other arrays (or in structs contained in arrays). Note that you must keep a reference to the original NativeArray, or Unity will detect a memory leak.

Template Parameters

T	Element type of the array.

Type Constraints

T* : *unmanaged

6.87.2 Constructor & Destructor Documentation

6.87.2.1 UnsafeArray() [1/3]

```
unsafe Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T >.UnsafeArray
(
            int length,
            Allocator allocator,
            bool clearMemory = true )
```

Allocate a new array of the given length.

Parameters

length	Number of items to hold.
allocator	Allocator to use.
clearMemory	Whether to clear the memory.

Returns

The created array pointer.

6.87.2.2 UnsafeArray() [2/3] unsafe Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T >.UnsafeArray (

```
            T * ptr,
            int length,
            Allocator allocator = Allocator.None )
```

Wrap an existing pre-allocated memory block.

Parameters

ptr	Pointer to the memory block.
length	Length of the memory block.
allocator	Allocator to use for deallocation. None means this object won't free the memory.

6.87.2.3 UnsafeArray() [3/3] Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T >.UnsafeArray (

```
            IntPtr ptr,
            int length,
            Allocator allocator = Allocator.None )
```

Wrap an existing pre-allocated memory block.

Parameters

ptr	Pointer to the memory block.
length	Length of the memory block.
allocator	Allocator to use for deallocation. None means this object won't free the memory.

6.87.3 Member Function Documentation

6.87.3.1 Dispose() unsafe void Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T >.Dispose ( )

Free the memory if it has been allocated directly.

If this pointer is wrapping a NativeArray, this does nothing.

6.87.3.2 GetEnumerator() Enumerator Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T >.Get↩ Enumerator ( )

Get an enumerator for the array.

6.87.3.3 ToPointer() static unsafe UnsafeArray< T > Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T >.ToPointer (

```
            in NativeArray< T > array )  [static]
```

Create a pointer to the given NativeArray.

Parameters

array	Array to create a pointer to.

Returns

The created pointer.

6.87.4 Member Data Documentation

6.87.4.1 Allocator `readonly Allocator Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T >.Allocator`

Allocator used to allocate the array, or None if it was created from a NativeArray.

6.87.4.2 Length `readonly int Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T >.Length`

Length of the array.

6.87.4.3 Pointer `readonly IntPtr Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T >.Pointer`

Pointer to the start of the array.

6.87.5 Property Documentation

6.87.5.1 IsNull `bool Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T >.IsNull [get]`

Returns whether the pointer is null.

6.87.5.2 MemorySize `int Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T >.MemorySize [get]`

Returns the size of the memory block in bytes.

6.87.5.3 this[int index] `unsafe ref T Infohazard.HyperNav.Jobs.Utility.UnsafeArray< T >.this[int index] [get]`

Get a reference to the element at the given index (can be used to set values as well).

Parameters

index	The index.

Exceptions

InvalidOperationException	(Dev Only) If underlying array is not set.
IndexOutOfRangeException	(Dev Only) If index is outside the bounds of the array.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Utility/UnsafeArray.cs

6.88 Infohazard.HyperNav.Jobs.Utility.UnsafeConcurrentQueue< T > Struct Template Reference

A thread-safe queue for use in Burst jobs.

6.88.1 Detailed Description

A thread-safe queue for use in Burst jobs.

Template Parameters

T	Type of item.

Type Constraints

T* : *unmanaged

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Utility/UnsafeConcurrentQueue.cs

6.89 Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T > Struct Template Reference

An implementation of a Max Heap that can be used with jobs and Burst.

Public Member Functions

- UnsafeHeap (int initialCapacity, Allocator allocator)

  Create a new NativeHeap.
- void Dispose ()

  Free the memory used by the heap.

- void Clear ()

  Remove all items from the heap.
- void Add (T item, float priority)

  Add an item to the heap with the given priority.
- void Update (T item, float newPriority, bool replace=false, T replaceWith=default)

  Change the priority of an item in the heap, and optionally replace it with a new item.
- bool TryUpdate (T item, float newPriority, bool replace=false, T replaceWith=default)

  Change the priority of an item in the heap, and optionally replace it with a new item.
- bool TryRemove (T item)

  Remove an item from the heap.
- bool TryPeek (out T value)

  Get the element at the top of the heap without removing it.
- bool TryPeek (out T value, out float priority)

  Get the element at the top of the heap without removing it.
- bool TryRemove (out T value)

  Remove the element at the top of the heap and return it.
- bool TryRemove (out T value, out float priority)

  Remove the element at the top of the heap and return it.

Properties

- int Count `[get]`

  Current number of items in the heap.
- bool IsCreated `[get]`

  Whether the heap has been allocated.

6.89.1 Detailed Description

An implementation of a Max Heap that can be used with jobs and Burst.

Template Parameters

T	Element type of the heap.

Type Constraints

T* : *unmanaged
***T* : *IEquatable*<*T*>**

6.89.2 Constructor & Destructor Documentation

6.89.2.1 UnsafeHeap()

```
Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T >.UnsafeHeap (
    int initialCapacity,
    Allocator allocator )
```

Create a new NativeHeap.

Parameters

initialCapacity	The initial capacity to allocate.
allocator	The allocator to use.

6.89.3 Member Function Documentation

6.89.3.1 Add() `void Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T >.Add ( T item, float priority )`

Add an item to the heap with the given priority.

Parameters

item	Item to add.
priority	The item's priority.

6.89.3.2 Clear() `void Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T >.Clear ( )`

Remove all items from the heap.

6.89.3.3 Dispose() `void Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T >.Dispose ( )`

Free the memory used by the heap.

6.89.3.4 TryPeek() [1/2] `bool Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T >.TryPeek ( out T value )`

Get the element at the top of the heap without removing it.

Parameters

value	The value at the top of the heap.

Returns

Whether the heap had an item to get.

6.89.3.5 TryPeek() [2/2]

```
bool Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T >.TryPeek (
        out T value,
        out float priority )
```

Get the element at the top of the heap without removing it.

Parameters

value	The value at the top of the heap.
priority	The element's priority.

Returns

Whether the heap had an item to get.

6.89.3.6 TryRemove() [1/3]

```
bool Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T >.TryRemove (
        out T value )
```

Remove the element at the top of the heap and return it.

Parameters

value	The value that was at the top of the heap.

Returns

Whether the heap had an item to remove.

6.89.3.7 TryRemove() [2/3]

```
bool Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T >.TryRemove (
        out T value,
        out float priority )
```

Remove the element at the top of the heap and return it.

Parameters

value	The value that was at the top of the heap.
priority	The element's priority.

Returns

Whether the heap had an item to remove.

6.89.3.8 TryRemove() [3/3] bool Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T >.TryRemove (T *item*)

Remove an item from the heap.

Parameters

item	The item to remove.

Returns

Whether the item was successfully removed.

6.89.3.9 TryUpdate() bool Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T >.TryUpdate (T *item*, float *newPriority*, bool *replace* = *false*, T *replaceWith* = *default*)

Change the priority of an item in the heap, and optionally replace it with a new item.

Parameters

item	The item to change.
newPriority	The new priority to use.
replace	Whether to replace the item as well.
replaceWith	What item to replace it with.

Returns

Whether the item was successfully updated.

6.89.3.10 Update() void Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T >.Update (T *item*, float *newPriority*, bool *replace* = *false*, T *replaceWith* = *default*)

Change the priority of an item in the heap, and optionally replace it with a new item.

Parameters

item	The item to change.
newPriority	The new priority to use.
replace	Whether to replace the item as well.
replaceWith	What item to replace it with.

Exceptions

ArgumentException	Thrown when the item is not in the heap.

6.89.4 Property Documentation

6.89.4.1 Count `int Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T >.Count [get]`

Current number of items in the heap.

6.89.4.2 IsCreated `bool Infohazard.HyperNav.Jobs.Utility.UnsafeHeap< T >.IsCreated [get]`

Whether the heap has been allocated.

The documentation for this struct was generated from the following file:

- Runtime/Jobs/Utility/UnsafeHeap.cs