

# Navigations Maps (QSF)

## AI [Navigation Maps](#)

Damit sich eine Einheit von einem Punkt auf der Karte zu einem anderen bewegen kann, verwendet QSF eine Pfadsuche (Routefinder).

[Navigation Maps](#) (auch AI-Maps oder KI-Maps genannt) sind das Fundament auf der die Pfadsuche basiert. In ihnen werden alle möglichen Pfade und Hindernisse eingetragen. Für verschiedene Einheiten und bewegte Objekte existieren verschiedene [Navigation Maps](#). So verwendet QSF das Street Network für zivile Fahrzeuge und Personen. Die Free Area wird für normale Einheiten verwendet, bei Fahrzeugen in Kombination mit dem Street Network. Helicopter View verwenden Helikopter, Diver Map Taucher und Water Streets Wasserfahrzeuge. Die Bridge Map wird für Brücken verwendet und ermöglicht Bewegung auf mehrere Ebenen. Außerdem gibt es spezielle [Events](#) die zusätzliche AI-Maps benötigen (z.B. "Water During Flood" in Hamburg).

### Inhaltsverzeichnis

- [1 AI Navigation Maps](#)
- [2 Aktualisierungen der Navigation Maps](#)
- [3 Brücken Editierung \(anspruchsvoll\)](#)

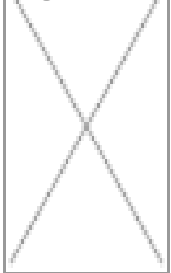
Alle [Navigation Maps](#) bestehen aus Pfaden, die eine bestimmte Breite besitzen. Auf diesen Pfaden können sich Einheiten bewegen. Hindernisse werden nur dadurch eingetragen, dass sich kein Pfad an dieser Stelle befindet. Nur statische Hindernisse sind für AI-Maps von Bedeutung. Im Editor werden Hindernisse mit Hilfe von Bullet Box Collision [Components](#) oder Blocker Polygonen mit Triangle Mesh Collision [Components](#) dargestellt. Nähere Informationen dazu in [Virtual Objects](#). Die Breite wird bei der Pfadsuche benutzt um herauszufinden, auf welchen Pfaden sich eine Einheit gerade noch bewegen kann.

Es gibt zwei Arten von [Navigation Maps](#): Direkt erstellte AI-Maps (wie z.B. das Street Network) und automatisch erstellte AI-Maps (wie z.B. die Free Area). Bei direkt editierte [Navigation Maps](#) werden einfach nur die im Editor erstellten Pfade direkt übertragen. Pfade werden mit dem Navigation Tile Tool erstellt und mit dem Street Network Tool überprüft und verbunden. Achtung: Editierfehler können hier automatisch zu falschen Pfaden, nicht benutzten Strassen und missachteten Hindernissen führen!

Automatisch erstellte [Navigation Maps](#) werden im Editor berechnet. Jedes Hindernis, das sich mit dem Bereich schneidet für den die AI-Map erstellt wurde, wird mit Pfaden umgangen. Mit Hilfe eines sogenannten [Voronoi Diagramm](#) wird jeder Pfad genau zwischen zwei Hindernisse gelegt. Auch hier muss beachtet werden, dass alle Hindernisse richtig erstellt wurden und nicht ungewollt Bereiche gesperrt werden. Damit der Bereich, in dem die AI-Map erstellt wird, genau bestimmt werden kann, muss auch die Ground Map korrekt vorliegen. In der Ground Map ist die Höhe eingetragen auf der sich Einheiten bewegen können sowie alle relevanten Ebenen (Wasser, Höhenstufen). Das Terrain wird immer in die Ground Map eingetragen, aber Objekte werden nur eingetragen wenn sie mit einer Walkable [Component](#) deklariert werden. Die normale Höhenstufe des Terrains und der meisten Strassen ist Walkable Level 0. Das Level 1 wird im Moment nur für Brücken verwendet.

## Aktualisierungen der [Navigation Maps](#)

image not found or type unknown



[Navigation Maps](#) werden mit dem [Navigation Map](#) Tool erstellt. Immer wenn Pfade, Hindernisse oder Kollisionen verändert werden, müssen alle [Navigation Maps](#) neu erstellt werden, um Änderungen zu übernehmen. Die [Navigation Maps](#) müssen in der richtigen Reihenfolge erstellt werden. (Tipp: Bei unserer Standardzusammenstellung die Navigationmaps einfach von unten nach oben durchrechnen).

Je nach Art müssen [Navigation Maps](#) mit Convert Street Network oder mit Cal erstellt werden:

- Direkt editierte [Navigation Maps](#) werden einfach nur mit Convert Street Network übertragen, Einstellungsparameter (außer Set) gibt es nicht. Das Übertragen aller Pfade benötigt eine kurze Zeit.
- Automatisch erstellte [Navigation Maps](#) werden mit Cal erstellt. Die benötigten Parameter müssen eingestellt sein. Die Berechnung kann mehrere Minuten benötigen.

Folgende Parameter sind von Bedeutung:

### **Corner1 und Corner2**

Definiert den Bereich in dem die [Navigation Map](#) erstellt wird. Nur innerhalb dieses Bereichs werden Pfade erstellt und Hindernisse beachtet. Die Map Border wird auch als Hindernis betrachtet. Die Höhe des Bereichs (der y-Wert der Koordinaten) wird, falls TestAtTerrainHeight aktiviert ist, mit Hilfe der Ground Map berechnet. Bei einem Bereich von -0,2 bis 2 Meter werden also alle Hindernisse bis zu 2 Meter über dem Boden eingetragen. Mit WalkableLevel wird festgelegt, welche Ebene in der Groundmap genutzt werden soll.

### **CellSize**

Bei der Berechnung mit Hilfe des Voronoi-Diagramms wird jedes Hindernis erst einmal in ein Grid eingetragen. Mit Cellsize können die Größe einer Zelle festlegen. Somit bestimmt dieser Wert die Auflösung unserer Hindernisse. Kleinere Details von Hindernissen gehen verloren.

### **LaneType**

Voreingestellte Werte sollten im Moment benutzt werden, dürfte für automatisch erstellte AI-Maps nicht von größerer Bedeutung sein.

### **MapId**

Identifikationsnummer für unsere [Navigation Map](#). Müsste automatisch korrekt erstellt werden. Achtung: Die Nummern der wichtigsten AI-Maps sind in [EMERGENCY 5](#) vorgegeben und dürfen nicht verändert werden. Für alle Einheiten und in der Spiellogik wird im Moment von den voreingestellten AI-Maps ausgegangen.

### **AvoidedCollisionFilter and RequiredCollisionFilter**

Einstellungen, welche Kollisionen benutzt oder ignoriert werden. Näheres dazu in Collision Flags.

### **WalkableLevel**

Wird benutzt um festzulegen, welches Walkable Level verwendet wird. Falls es eine darüberliegende Ebene gibt, die weniger als 1 Meter entfernt ist, wird der Bereich als Blocker behandelt. Dies wird benötigt, wenn mehrere Ebenen mit MixWithMapId vermischt werden. Falls dies nicht gewünscht wird muss das WalkableLevel auf -1 gesetzt.

### **MixWithMapId**

Wird benötigt um zwei KI-Maps von verschiedenen Ebene zu vermischen z.B. mit einer Bridge-Map. Somit wird ermöglicht, unter Brücken zu laufen und korrekte Pfade zu berechnen, die eine Einheit nicht von Brücken springen lässt. Für das Vermischen werden aber zusätzliche Editierungen benötigt. Spezielle Crossings müssen verwendet werden, um Pfade für die Verbindung einzutragen und Bridge [Components](#) müssen verwendet werden, damit Brücken erkannt werden. Um diese Funktion zu deaktivieren muss der Wert auf -1 gesetzt werden.

### **IncludeWater**

Falls Wasser auch mit eingetragen werden soll (z.B. bei der Diver Map), muss dieser Wert aktiviert werden. Ansonsten wird das Wasser komplett in der KI-Map blockiert. Achtung: Das Wasserpolygon wird unabhängig davon noch als Hindernis betrachtet und der Collision Filter muss richtig gesetzt werden.

### **TestAtTerrainHeight**

Falls aktiviert. wird das Terrain und die Walkables bei der Berechnung des Bereiches beachtet. Ansonsten wird der absolute Höhenwert der Karte benutzt.

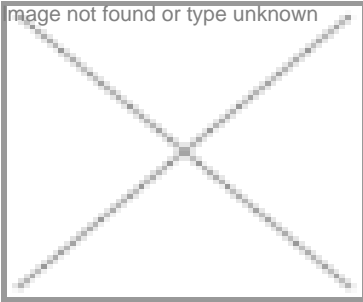
### **Show Obstacles, ShowEquidistantCells, mShowEquidistanceObstacleInfluence and DebugVisualizationTime**

Verschiedene Debug-Funktionen die ignoriert werden können. Eventuell ist Show Obstacles noch interessant, um alle Hindernisse anzuzeigen die eingetragen wurden.

### **Set mit seiner eigenen Leiste**

Das Street Network und die Free Area werden im Spiel verbunden, damit Einsatzfahrzeuge beide verwenden können. Deshalb sollte hier der Wert korrekt gesetzt werden. (1 bei Street Network, 0 bei Free Area)

Nach Abschluss der Berechnung der [Navigation Maps](#) muss die Einsatzkarte noch gespeichert werden. Dies sollte auf keinen Fall vergessen werden! Neue [Navigation Maps](#) können zwar auch erstellt werden und alte gelöscht werden, aber Anpassungen im Code müssten das Ganze meistens begleiten.



Die Breite der Pfade wird nicht dargestellt. Durch das Path Smoothing kann der Pfad die ganze Breite benutzen, er liegt also nicht direkt mittelt auf einem Pfad, auch wenn er ihn benutzt.

## Brücken Editierung (anspruchsvoll)

Damit Brücken und mehrere Ebenen von [Navigation Maps](#) korrekt funktionieren, müssen mehrere komplexe Editierungsschritte eingehalten werden! Für jede Brücke auf einem höherem Walkable Level müssen an jedem Übergang Pfade editiert werden. Dazu sollte eine Debug Box mit einer Street Crossing [Component](#) an dieser Stelle erstellt werden. Die Pfade können mit dem Navigation Tile Tool erstellt werden. Alle Pfade müssen sich genau zwischen dem Übergang der Walkables befinden (z.B. ein Knoten auf der Brücke mit Walkable 1 und einer auf der anderen Strasse mit Walkable 0). Die richtige Anzahl, Position und Breite der Pfade ist leider nicht immer einfach festzulegen. Da das "Path Smoothing" noch Probleme beim Übergang bereitet und Zickzack-Pfade entstehen können ist es wichtig, dass die Pfade an Positionen gelegt werden, die möglichst natürlichen Punkten beim Übergang entsprechen (z.B. ein Gehsteig). Die Pfade müssen breit genug sein, so dass alle Einheitentypen durch die Pfade durchpassen. Alle Übergangsstellen müssen durch Pfade abgedeckt werden, ansonsten wird der Bereich wie ein Hindernis betrachtet. Ohne einen einzigen Pfad ist eine Brücke auch nicht erreichbar und eine Einheit wird einen Endpunkt so nahe wie möglich an der Brücke suchen, außerdem wird die Brücke nicht beachtet, falls der kürzeste Weg zu einem anderen Ziel über die Brücke führen würde. Jeder Übergang sollte mit Personen und Fahrzeug-Einheiten getestet werden.

Der zweite Teil der Editierung betrifft das Erstellen von Bridge [Components](#) zum Identifizieren von Brücken. [EMERGENCY 5](#) hat zwei verschiedene Pfadsuchsysteme. Das erste Pfadsuchsystem funktioniert mit Hilfe der [Navigation Maps](#) und wird für das Erstellen des Pfades benutzt. Das zweite System wird verwendet um im Spiel dynamische Hindernisse (andere bewegte Objekte) zu umfahren und ist vollständig Grid basiert. Damit das zweite Pfadsuchsystem auch korrekt mit Brücken umgehen kann, müssen Brücken und erhöhte Ebenen mit WalkableLevel 1 oder höher an jedem Punkt identifizierbar sein. Durch Debug Boxen mit einer Bridge [Component](#) können wir jeder Brücke eine eindeutige Identifikationsnummer geben. Jeder Punkt auf der Brücke muss näher an einer Bridge [Component](#) von seiner Brücke liegen als zu einer Bridge [Component](#) einer anderen Brücke. Dies kann bei größeren Brücken bedeuten, dass mehrere Bridge [Components](#) verteilt werden, so dass jeder Teil abgedeckt wird. Außerdem muss unter jeder Brücke eine Kollisionsbox oder -polygon verhindern, dass eine Einheit mit Hilfe des zweiten Pfadsuchsystems von unten durch die Brücke bricht (der Abstand des Hindernisses zur Brücke muss aber trotzdem mindestens 0.2 Meter (falls der Bereich durch Corner1 und Corner2 zwischen -0.2 und 2 Metern liegt) betragen, damit kein Hindernis auf der Brücke eingetragen wird, nur unter der Brücke.

