

# Omsi-Bau-Tagebuch

## Kreuzungen Teil 1

© 24.02.2010 Marcel Kuhnt

In diesem ersten Kreuzungs-Tutorial wird erklärt, wie man eine einfache T-Einmündung zweier einfacher Straßen ohne Radweg baut.

Grundsätzlich kann man sich natürlich ohne Probleme ein Luftbild hinterlegen – darauf wird im Rahmen dieses Tutorials aber aus urheberrechtlichen Gründen verzichtet. Bei einer so einfachen Kreuzung ist es auch nicht unbedingt nötig – es reicht, wenn man im Luftbild-Programm ein paar Maße nimmt.

Es wird bei diesem Tutorial davon ausgegangen, dass die Straßentypen, die in diese Kreuzung münden, bereits existieren. Wie man eigene Straßentypen erstellt, soll dann in einem separaten Tutorial erklärt werden.

Wichtig ist, dass man die Straßentypen, die ja üblicherweise im Spline-Format \*.sli vorliegen, in x-Dateien konvertiert. Hierfür wird das Programm „StreetCreator.exe“ verwendet, welches auch für das Erstellen von Straßensplines verwendet werden kann. StreetCreator erstellt dann aus dem Splineprofil ein 10m langes Segment und speichert es als x-Datei ab, sodass diese dann in Blender als Vorlage eingefügt werden kann, was gerade bei komplizierteren Straßen sehr hilfreich ist.

Bevor es losgeht, noch eine allgemeine Konvention, die wir bei unseren Straßen und Kreuzungen eingeführt haben: **alle Straßenoberflächen** liegen auf der Höhe **0,1m**, **alle Gehwege** auf der Höhe von **0,25m**, sodass also die Bordsteinhöhe 0,15m = 15cm entspricht. **Abgesenkte Bordsteine** liegen auf der Höhe **0,15m**, sodass der Unterschied zum normalen Bordstein 10cm, der Höhenunterschied zur Straße 5cm beträgt.

Fangen wir nun also an...

Diesem Tutorial liegt eine Blenderdatei „kreuz\_templ.blend“ bei (welche die Vorlagen der wichtigsten Einzelteile einer Kreuzung enthält) sowie eine Reihe von nützlichen Kreuzungstexturen, welche zusammen mit der Vorlage genutzt werden können. Diese bilden auch die Grundlagen für die Kreuzungen unserer Omsi-Strecke.

Die Vorlage sowie die Texturen sollten zunächst in einen neuen Projektordner für alle künftig zu bauenden Kreuzungen kopiert werden und die Blenderdatei dann geladen werden.

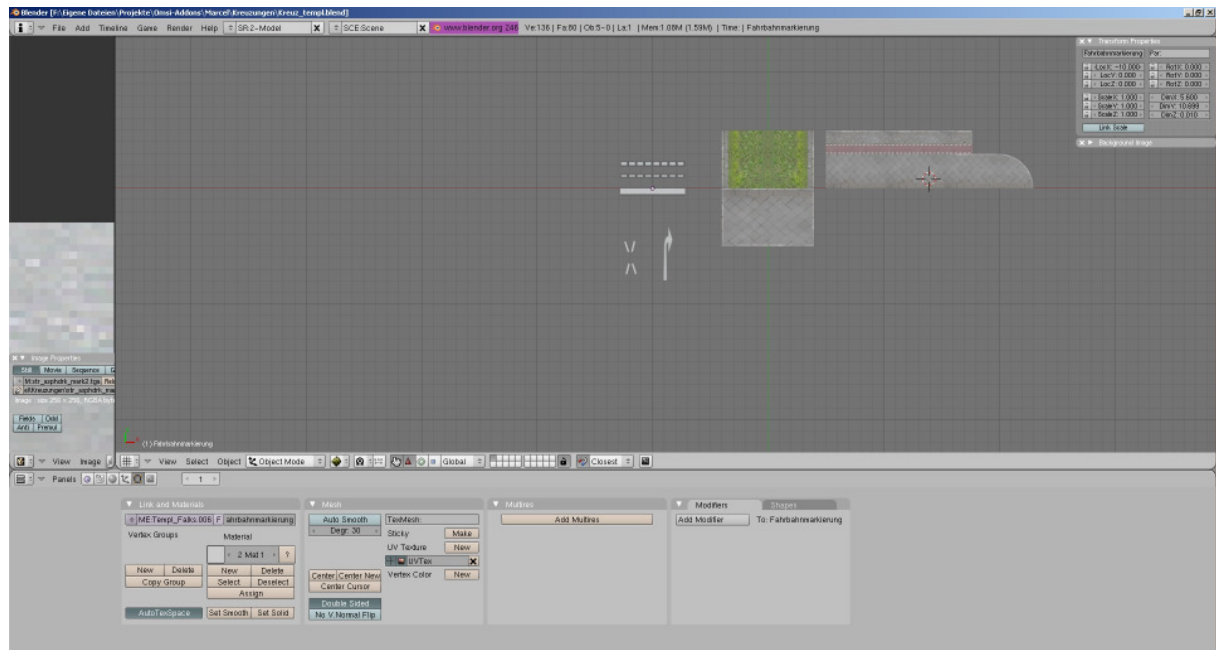
Sollten die Texturen dabei nicht korrekt geladen werden, so muss für jede nicht geladene, aber bereits in der List im Image-Editor (links) befindliche Textur folgender Vorgang durchgeführt werden:

- Aufrufen der Textur (Klick auf den Eintrag in der Liste)
- mit „N“ (während Mauszeiger über dem Image-Editor liegt) das Dialogfeld „Image Properties“ aufrufen, sofern nicht schon geschehen

- im zweiten Textfeld von oben (mit dem Ordnersymbol davor) ist der komplette Dateipfad der jeweiligen Textur einzugeben. Dieser setzt sich wie üblich aus dem Pfad des (neuen) Kreuzungs-Projektordner und dem Dateinamen der Textur zusammen, letzterer kann im Textfeld darüber abgelesen werden.

Da dieser Vorgang recht aufwändig ist, sollte, nachdem alle Texturen korrekt geladen wurden, die Vorlagen-Datei „kreuz\_templ.blend“ unter einem neuen Namen gespeichert und schreibgeschützt werden.

Wenn alles geklappt hat, sollte die Datei sich in etwa wie folgt präsentieren:



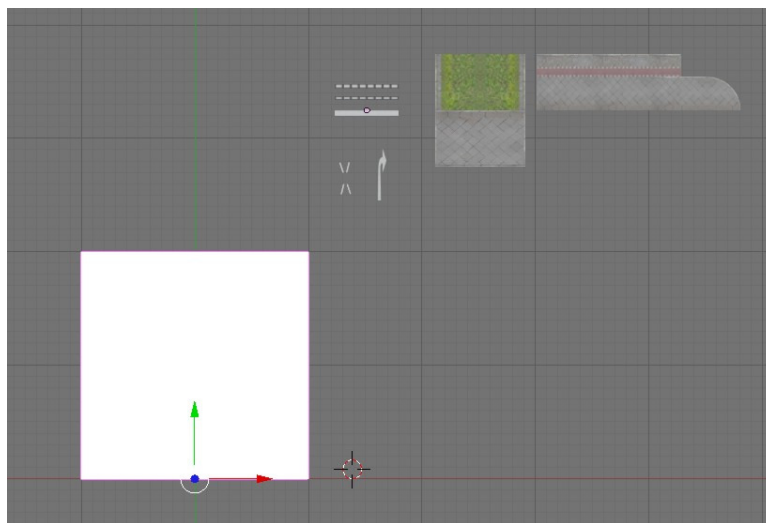
Die Vorlage wird als erstes mit dem Dateinamen der nun zu erstellenden Kreuzung gespeichert (damit die Vorlage vorhanden bleibt). Es handelt sich um die Einmündung des Päwesiner Wegs in die Seeburger Str. Gemäß unserer Konvention heißt die Einmündung deshalb „einm\_see\_paewesiner.blend“.

Die Vorlagen-Objekte sollten vor dem nächsten Schritt ein wenig zur Seite geräumt werden... ;-)

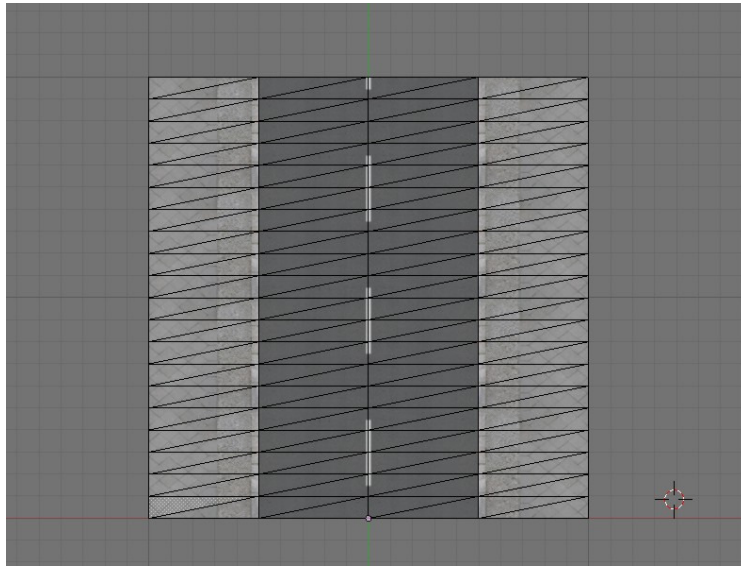
Als nächstes müssen die in das x-Format konvertierten Straßen importiert werden. Dies geschieht über den „File“ => „Import“ => „DirectX“ Befehl im Menü:



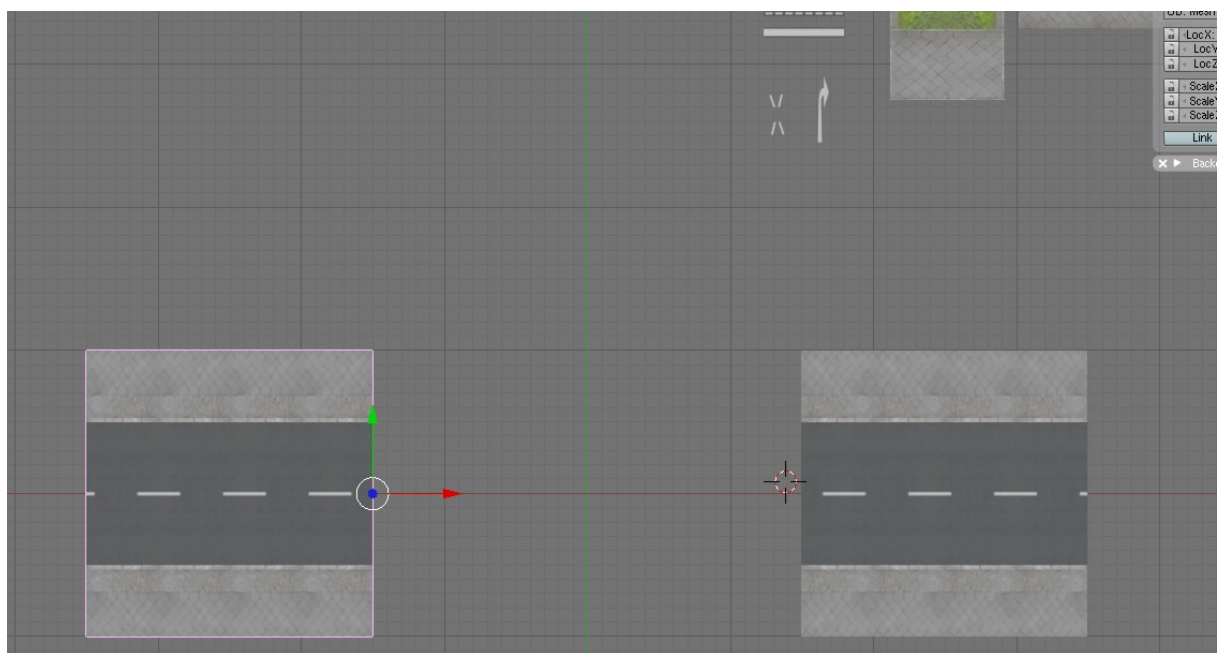
Die importierte Straße zeigt sich (leider!) völlig untexturiert:



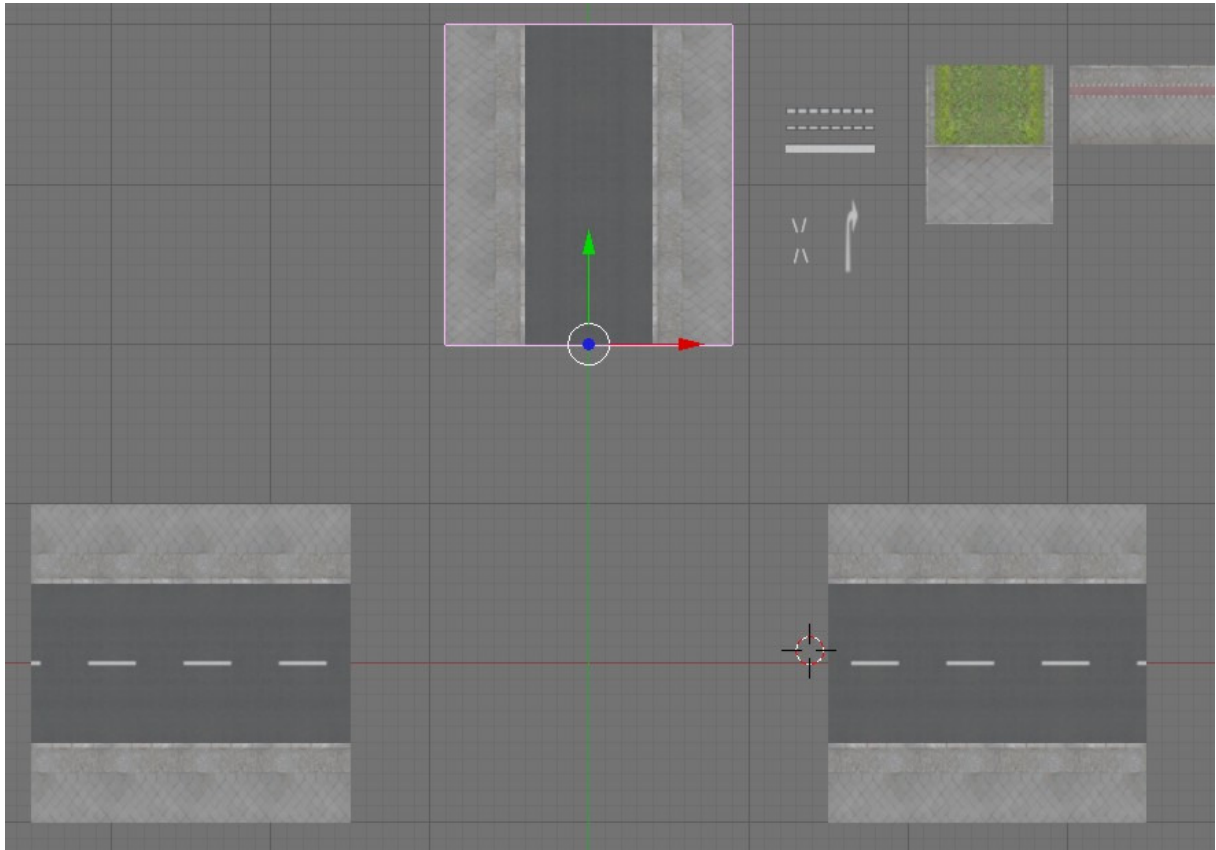
Dies muss manuell nachgeholt werden (die verwendeten Texturdateien können in der \*.sli-Datei nachgelesen werden!), sofern man es beim Erstellen der Kreuzung leichter haben möchte... das Texturieren geschieht ganz normal mit dem Image-Editor im Edit-Mode. Dann sollte es so aussehen:



Zurück im Object-Mode (Wichtig! Damit Objekt-Ursprung mitbewegt wird!) wird die Straße nun (entweder anhand des Luftbildes oder mittels Koordinateneingabe) so platziert wie sie später relativ zur Kreuzung im Map-Editor andockt werden soll. Da diese Straße in unserem Fall links und rechts andockt werden soll, habe ich das Objekt einfach einmal kopiert (anstatt zweimal importiert und texturiert):

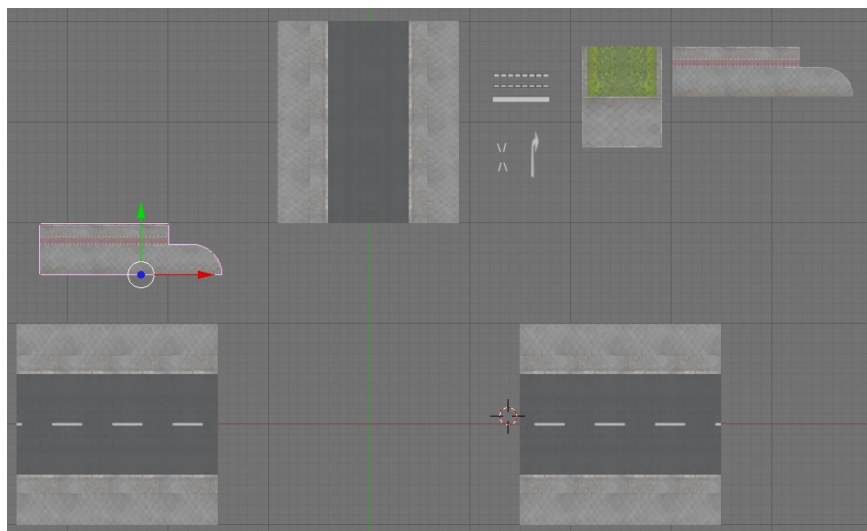


Dasselbe mit dem anderen Straßentyp, der beim Päwesiner Weg verwendet wird und einmündet:



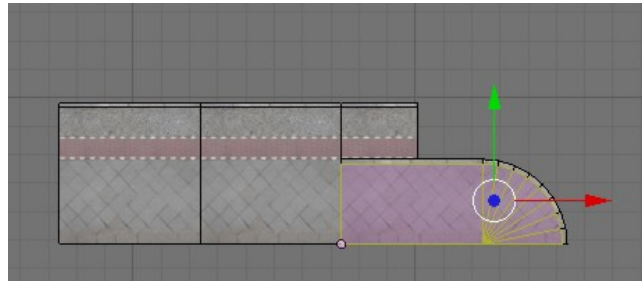
Die eigentliche Kreuzung erstelle ich aus zwei separaten Objekten: das eine Objekt besteht aus allen Bürgersteigen, das andere aus allen Straßenflächen. Später, beim Export, wird aber alles zusammen in eine x/o3d-Datei exportiert.

Wir beginnen, indem wir die Vorlage für die Bürgersteige erstmal komplett im Object-Mode kopieren (Shift+D) ...:

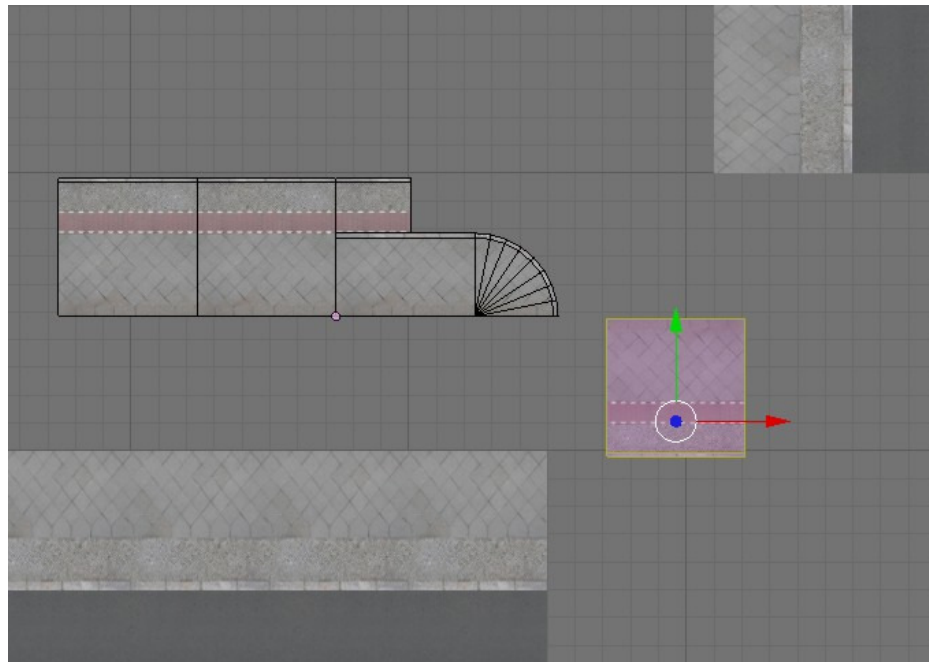


... und dann im Edit-Mode betrachten: der Bürgersteig ist so gestaltet, dass die Bordsteinkante und die Oberfläche getrennt sind. Das hat den Vorteil, dass die Bordsteinkante auch unregelmäßige Formen haben kann und die Oberfläche des Bürgersteigs dennoch gleichmäßig gemappt werden kann. Außerdem ist erkennbar, dass die Vorlage aus den wichtigsten drei Bürgersteigelementen besteht: dem „normalen“ Bürgersteig, der

Radwegschräge mit dahinterliegendem „Rest-Bürgersteig“ und dem 90°-Bogen für Ecken oder Mittelinseln u.Ä.



Zunächst wird aber nur ein normales Bürgersteigelement kopiert und um 180° gedreht. Hierbei ist folgendes zu beachten: die Funktion „occlude background Geometry“ muss **aus** sein, da andernfalls die Gefahr besteht, dass das (genau senkrecht stehende) Poly des Bordsteins nicht mitgenommen wird.

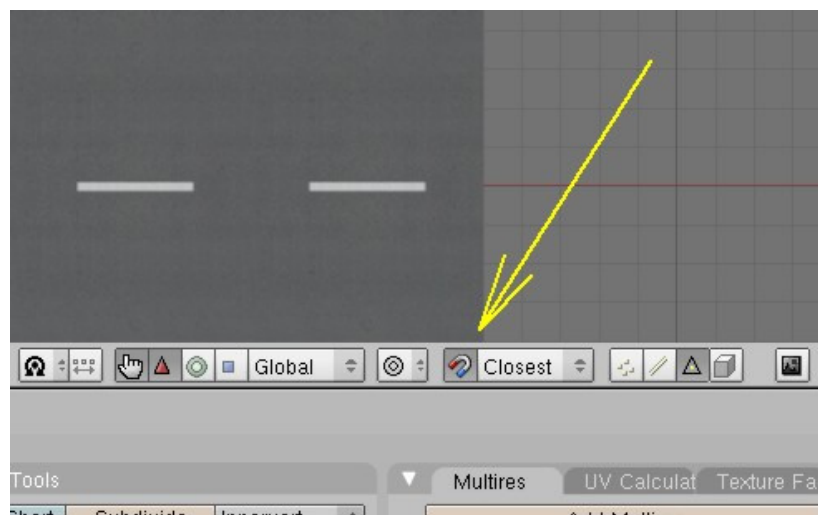


Weil die Textur geändert werden soll (str\_side1.bmp, ohne Radweg) und der andere Bürgersteigtyp über andere Texturkoordinaten verfügt, müssen diese angepasst werden:



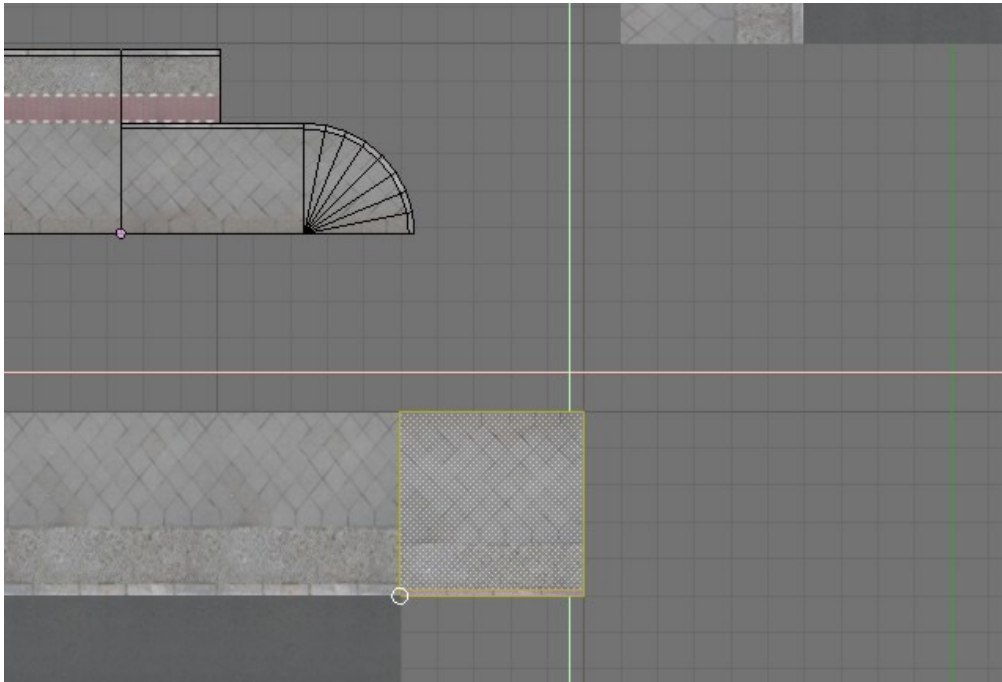


Als nächstes soll das Bürgersteigelement nahtlos an das bestehende Stück Straße angelegt werden (zur Erinnerung: die am Anfang bereitgelegten Straßenstücke dienen nur der Vorlage und werden später *nicht* Teil der Kreuzung sein!). Hierfür muss die „Snap“-Funktion an sein:

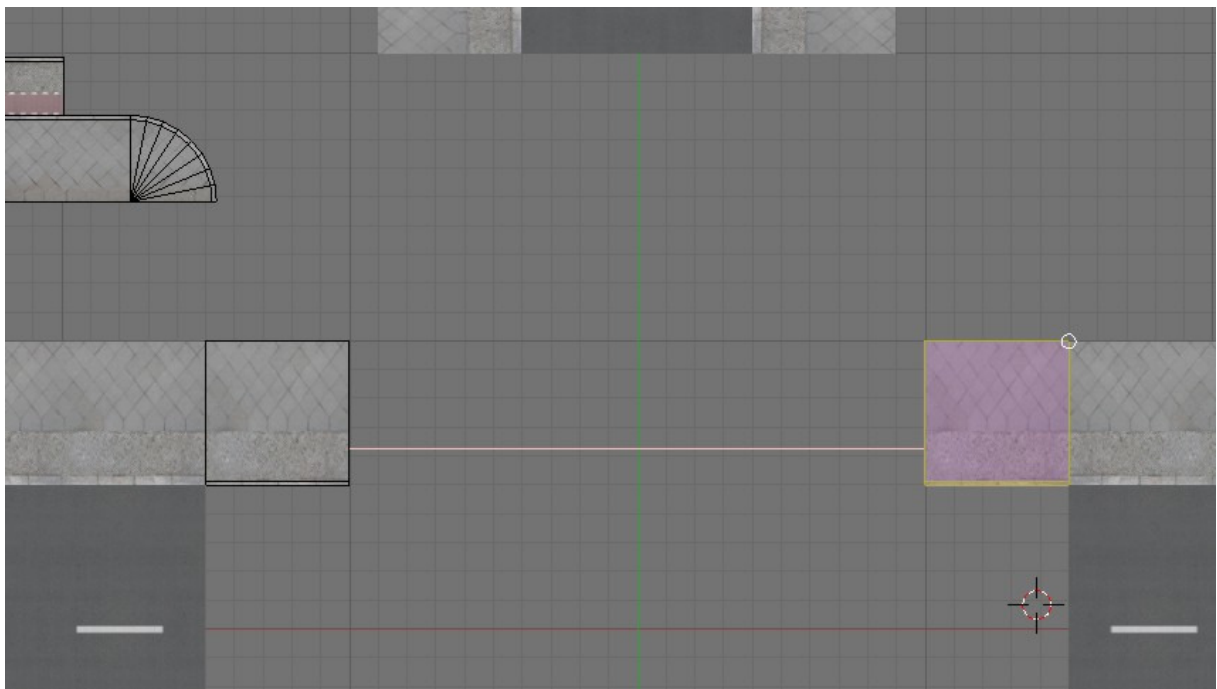


Es wird nun wie folgt vorgegangen: zuerst die Polys markieren, dann Taste G zum Bewegen, dann Shift+Z (Bewegung nur in der X-Y-Ebene, jedoch NICHT entlang der Z-Achse) und dann durch Drücken und Halten der Strg-Taste den Snap-Mode aktivieren und schließlich das Stück Bürgersteig so bewegen, dass es wie gewünscht einrastet und dann mit der Enter-Taste oder der linken Maustaste bestätigen (ohne die Strg-Taste loszulassen! Diese erst danach loslassen!)

Die Aktivierung der X-Y-Ebene ist deshalb nötig, damit beim Snap-Mode nicht die „falschen“ Punkte zusammengeführt werden (nämlich einmal die obere Bordsteinkante und einmal die untere) und auf jeden Fall verhindert wird, dass das Stück Bürgersteig in der Höhe verschoben wird (was keinen Sinn ergeben würde). Andernfalls kann dies nämlich passieren!

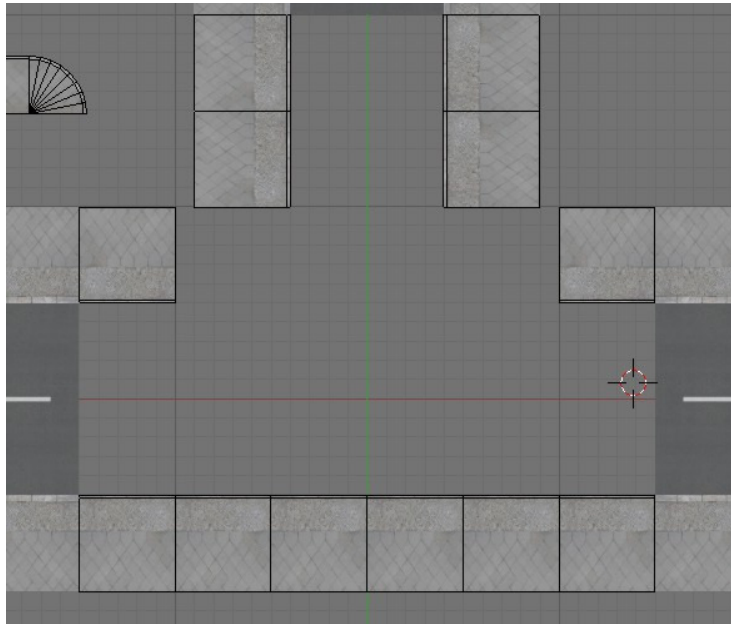


Auf diese Weise wird nun ein gleiches Stück Bürgersteig rechts gegenüber angebracht. In diesem Fall kann statt [Shift]+[Z] auch [X] gedrückt werden – da die Verschiebung ausschließlich entlang der X-Achse erfolgen soll:



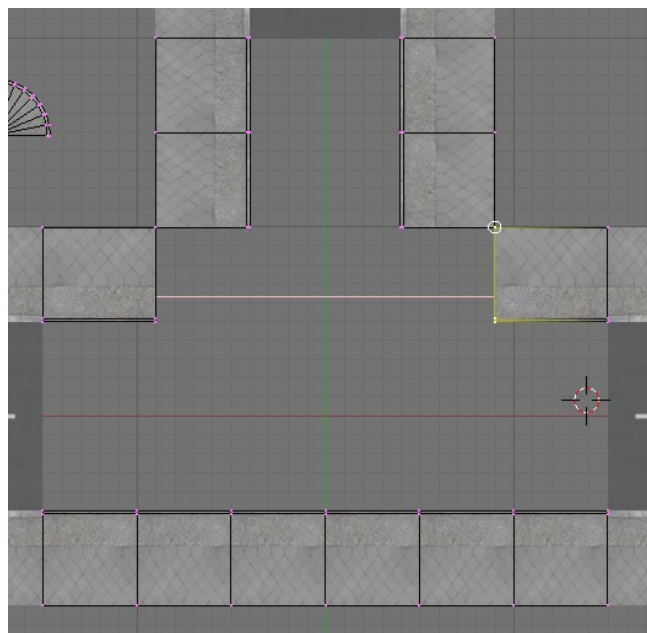
Nun werden die restlichen Bürgersteigteile auf gleiche Weise platziert, sodass das Ergebnis wie folgt aussieht. Hierbei ist zu beachten, dass das „Einrasten“ bei Blender manchmal von einer Seite nicht möglich ist – dann muss man ggf. das Objekt noch mal wieder etwas weiter weg bewegen und es erneut probieren.



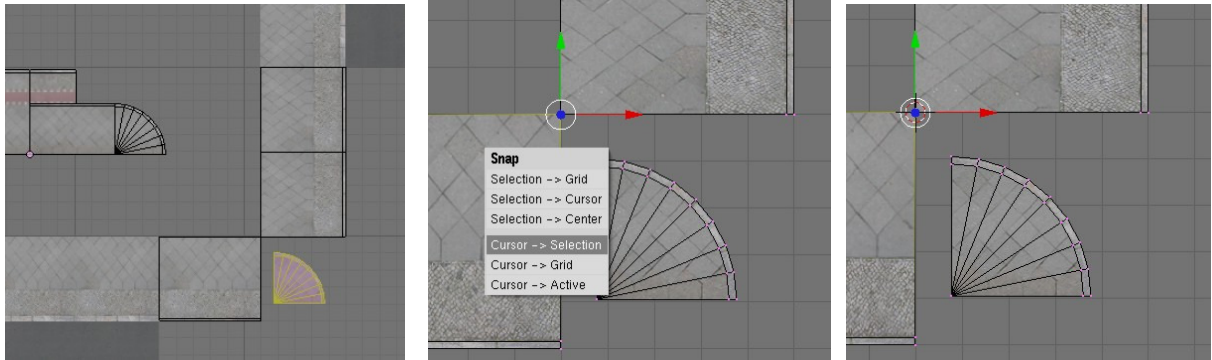


Immer wieder zwischendurch sollten die Dubletten der Punkte entfernt werden (alle Punkte im Vertex-Modus markieren, dann „W“ und im Kontextmenü „Remove Doubles“ anklicken).

Links und rechts fehlt, wie oben zusehen ist, jeweils 1m Bürgersteig – hier wird der Einfachheit halber lediglich das benachbarte Bürgersteigsegment etwas gestreckt (ebenfalls wird die Snap-Funktion verwendet):



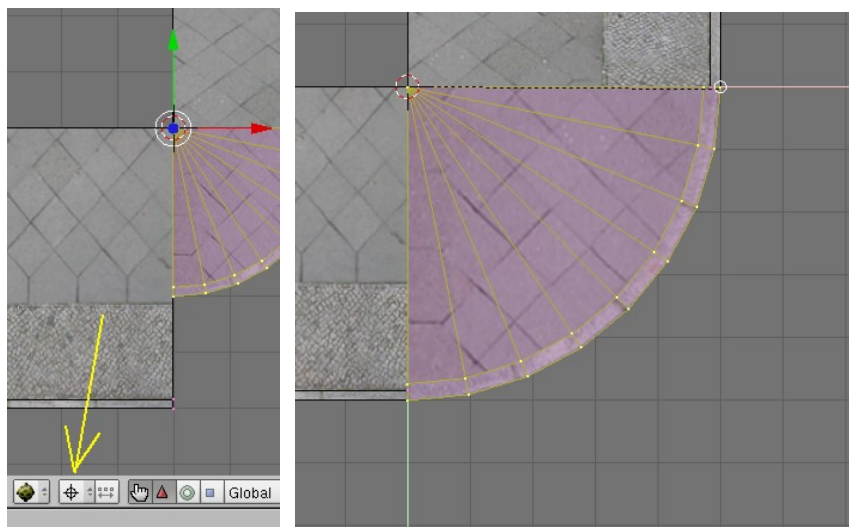
Kommen wir nun zu den Bürgersteig-Ecken, die ja eigentlich Bögen sind... zunächst einmal wird aus der Vorlage das Ecksegment inkl. Bordstein kopiert und zu unserer Baustelle verschoben. Nun kommt der Cursor ins Spiel (dieses schwarz/weiß/rote Fadenkreuz): Nachdem der Punkt (Vertex-Modus) in der Ecke des Bürgersteiges markiert wurde, wird das Kontextmenü mit Shift+S aufgerufen, dort wird dann „Cursor -> Selection“ gewählt. Der Cursor wird daraufhin exakt auf den ausgewählten Punkt gesetzt:



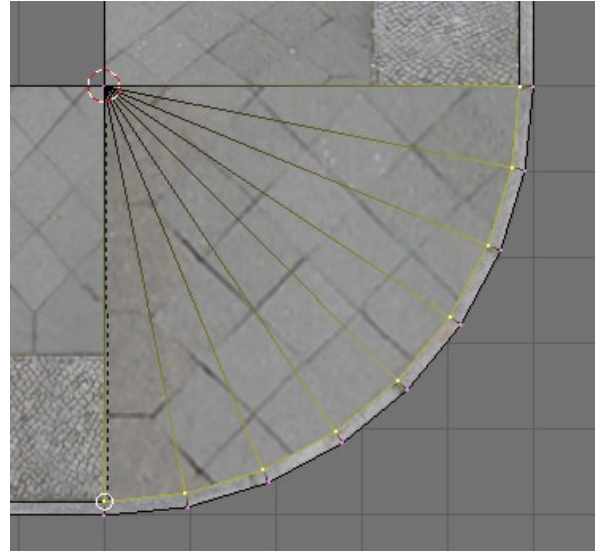
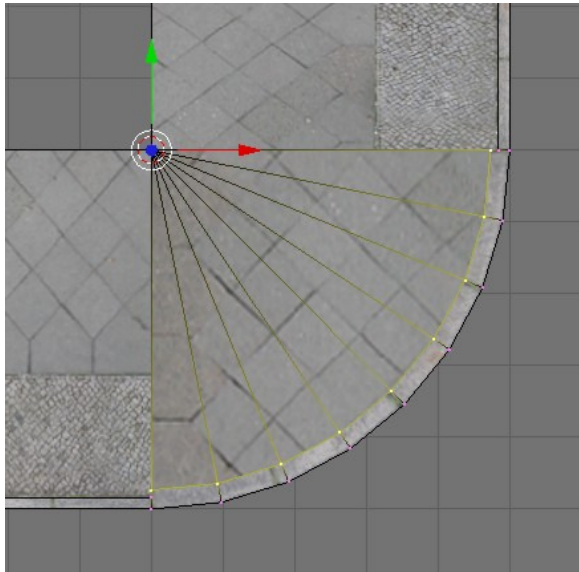
Nun wird das Viertelkreissegment um 90° gedreht (Erinnerung: im Rotationsmodus direkt die Zahl und ggf. „-“ eingeben sowie Enter drücken) und mit der Snapfunktion genau in die Ecke gesetzt.



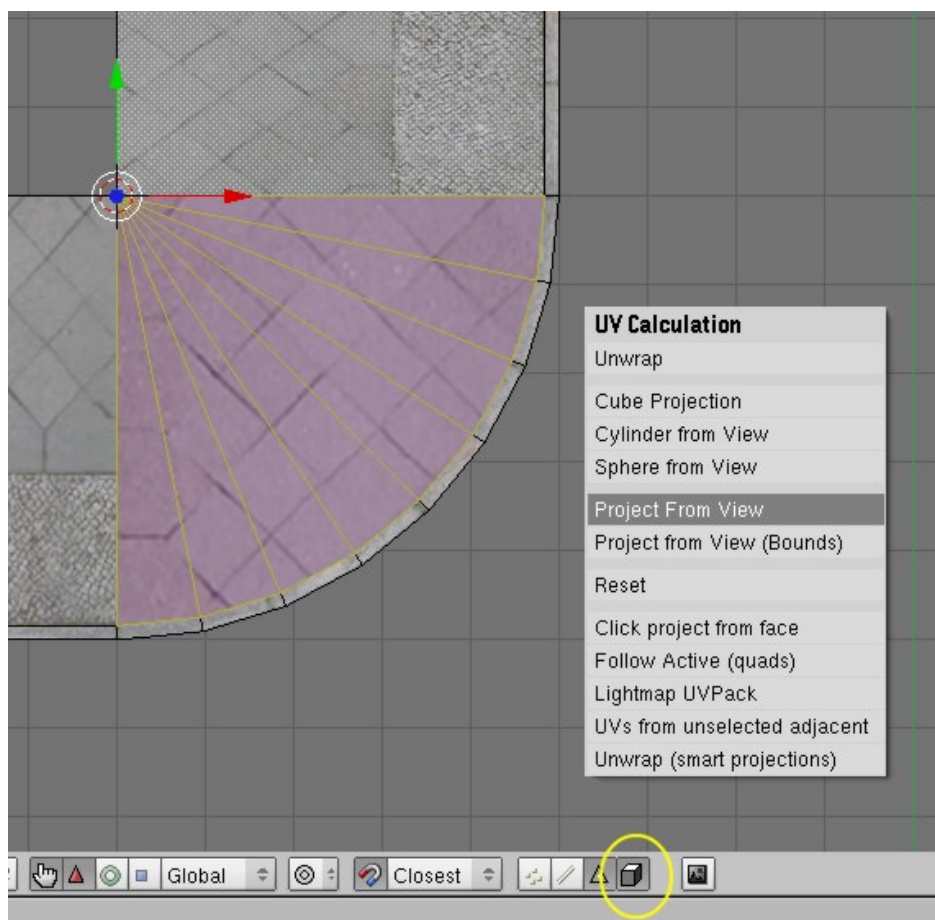
Nun wird der „Rotating/Scaling Pivot“, also der Ursprung der jeweiligen Transformationen auf den Cursor gesetzt, sodass im nächsten Schritt das gesamte Objekt **nur in der X-Y-Ebene** (durch Drücken von Shift+Z währenddessen) skaliert (Taste S) und mit der Snapfunktion passgenau eingerastet:



Da die Innenkante des Bordsteins damit an der falschen Stelle liegt, wird diese in einem weiteren Schritt getrennt noch mal skaliert – auf die gleiche Weise:



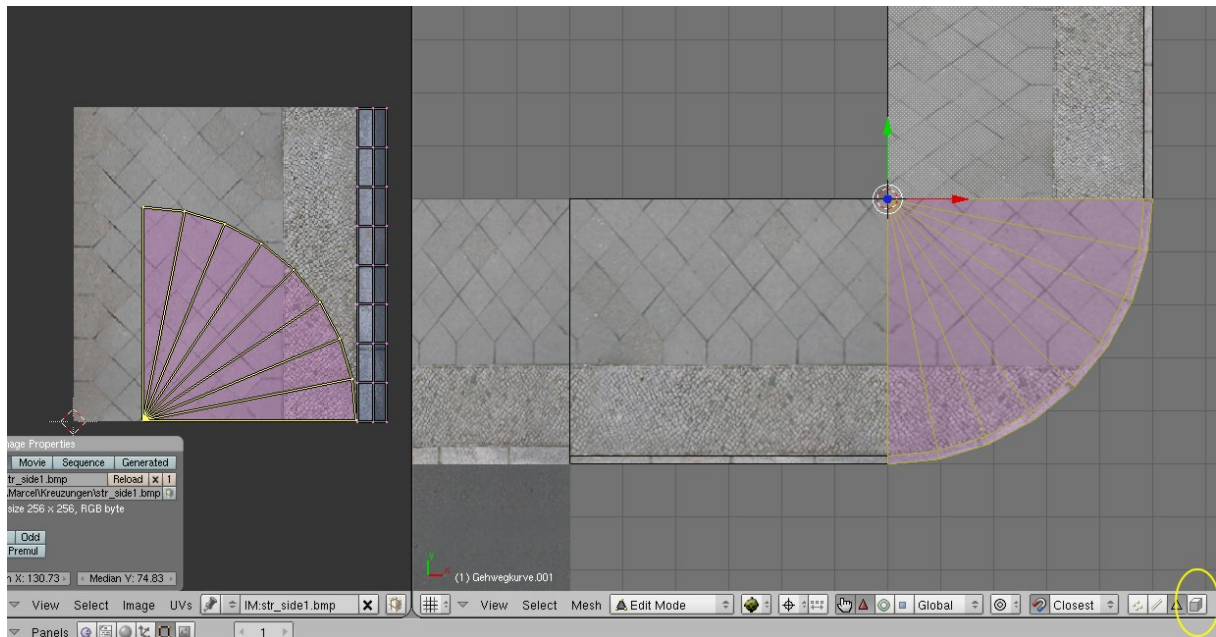
Nun muss lediglich die Texturierung angepasst werden. Dies geschieht zunächst nur für die Polygone der Oberfläche (hinterm Bordstein). Wenn diese selektiert werden sollen, schalte ich oft den Modus „Occlude Background Geometry“ an, da dann die Selektion in solchen Fällen einfacher ist. **Dann aber nicht vergessen, den Modus wieder auszuschalten!** Ansonsten verliert man, wie oben bereits erwähnt, u.U. die Seitenkanten des Bordsteins!



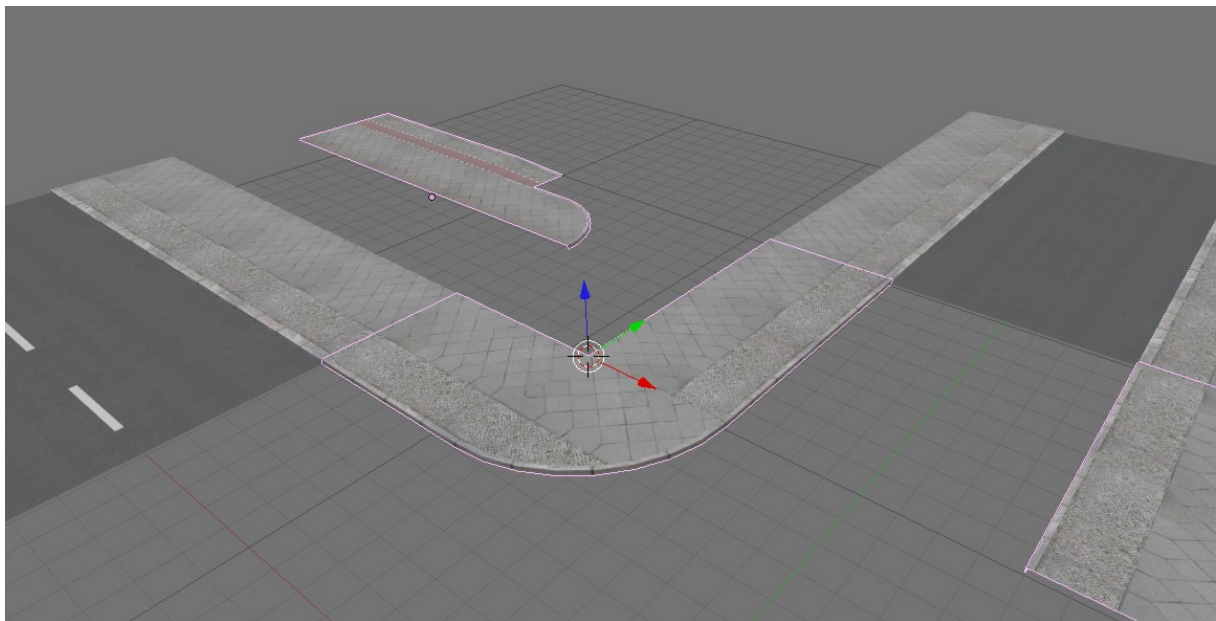
Schließlich wird die Textur noch auf dieselbe geändert wie die der anderen Bürgersteige (ursprünglich verwendet dieses Bogensegment die Textur „str\_gehweg03.bmp“, wohingegen



aber die Bürgersteigelemente – wie auch die anliegenden Straßen – die Bürgersteigtextur „str\_side1.bmp“ verwenden). Dadurch muss das Mapping noch mal angepasst werden:

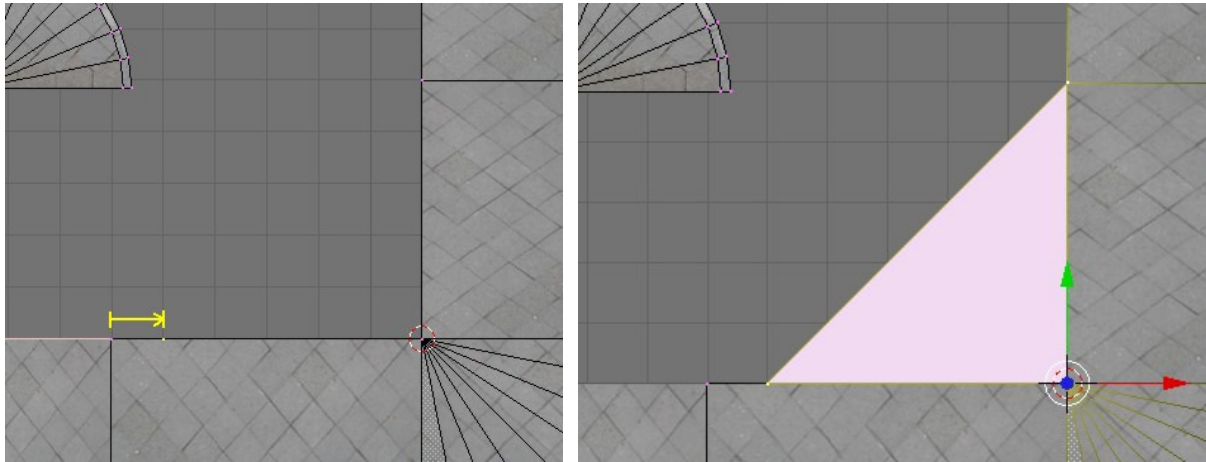


Ein Blick auf die fertige Gehwegbiegung:

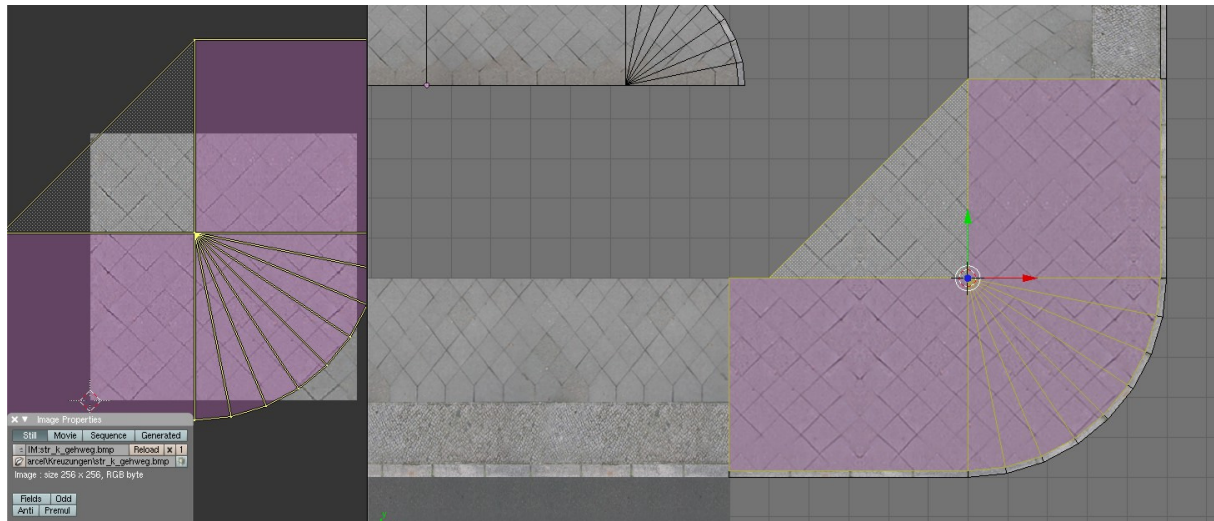


Zwar in diesem Fall nicht unbedingt nötig, aber der Einheitlichkeit halber soll noch ein 5m x 5m-großes Bürgersteig-Dreieck eingefügt werden, welches die Gehwegecke mit 45° abschrägt und somit Begegnungen von Fußgängern vereinfacht. Derartige Konstruktionen findet man eigentlich immer dort, wo der Platz dafür ausreicht.

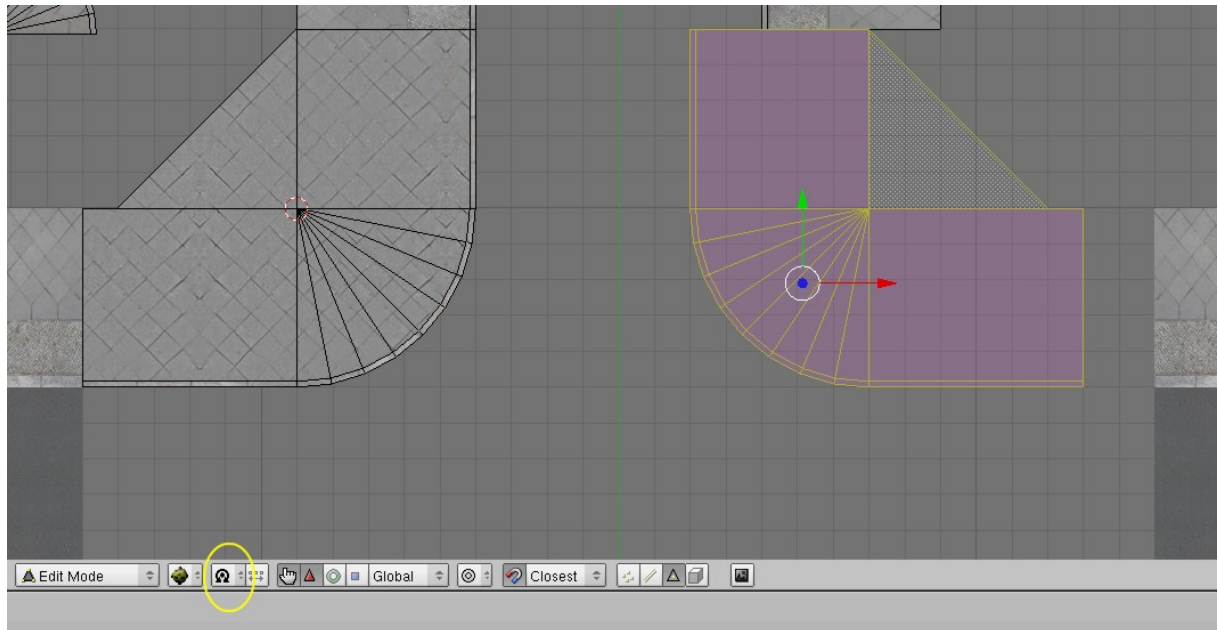
Zuerst werden wieder mal alle Dubletten entfernt (alle Punkte markieren, W, „Remove Doubles“). Dann schauen wir uns die Situation von oben an: am Gehweg der nördlichen Straße ist bereits ein geeigneter Punkt, 5m von der Ecke entfernt. Jedoch nicht an der westlichen: dieser wird durch Dublizieren eines bestehenden Punktes erzeugt (mit „X“ und der Eingabe von „1“ und Enter) und schließlich das Dreieck mit Selektion der Punkte und „F“ erzeugt:



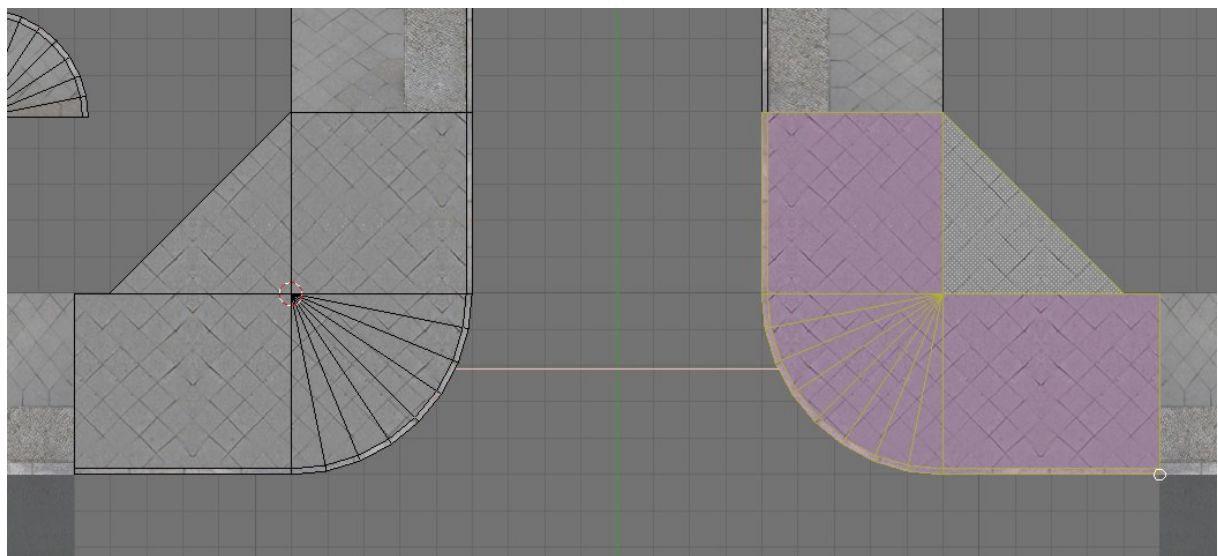
Nun muss das Dreieck noch texturiert werden. Hierfür eignet sich am Besten die Textur „str\_k\_gehweg.bmp“, die kachelbar auf beliebig großen Flächen ist. Bei der Gelegenheit habe ich mich der besseren Optik wegen um entschieden, auch die direkt an der Straße befindlichen Gehwegflächen mit der gekachelten Textur zu versehen:



Vorm Dublizieren und Spiegeln der Gehwegecke für die rechte Seite empfiehlt es sich, den „Rotating/Scaling Pivot“ wieder auf „Median Point“ zurückzustellen:

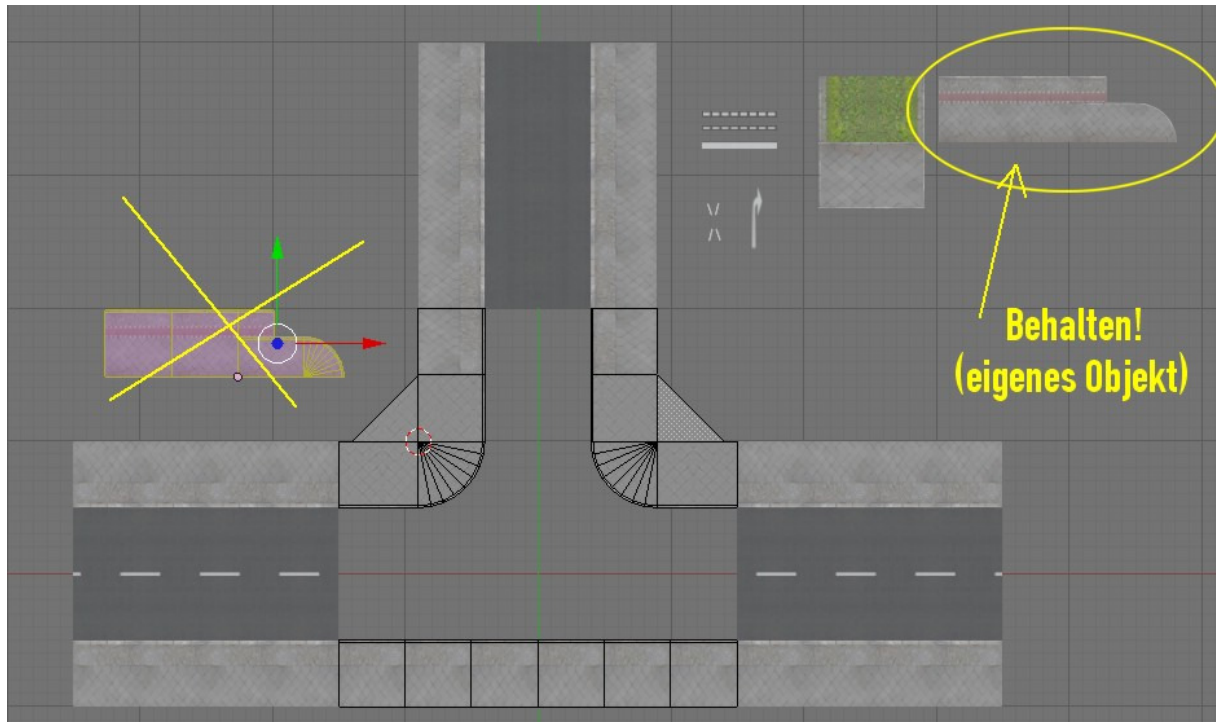


Nach dem Spiegeln (Taste M, dann X und dann Enter) müssen die Normalen umgedreht werden (Taste W, „Flip Normals“). Das fertige Segment kann nun mit der Snapfunktion exakt platziert werden:



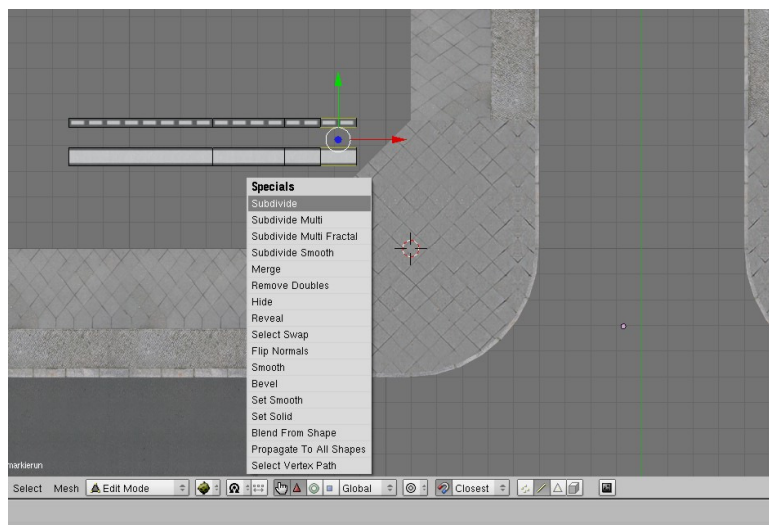
Die Bürgersteige sind nun fertig! Die übriggebliebene Vorlage (im selben Objekt, im oberen Bild noch erkennbar) kann gelöscht werden – da dies nur eine Kopie war, sollte nach wie vor das Original der Vorlagen in der Datei enthalten sein. Dieses behalte ich immer da, um bei nötigen Änderungen schnell auf die Vorlagen zugreifen zu können. Es sollte aber für einen unproblematischen Export auf jeden Fall in einem getrennten Objekt liegen:





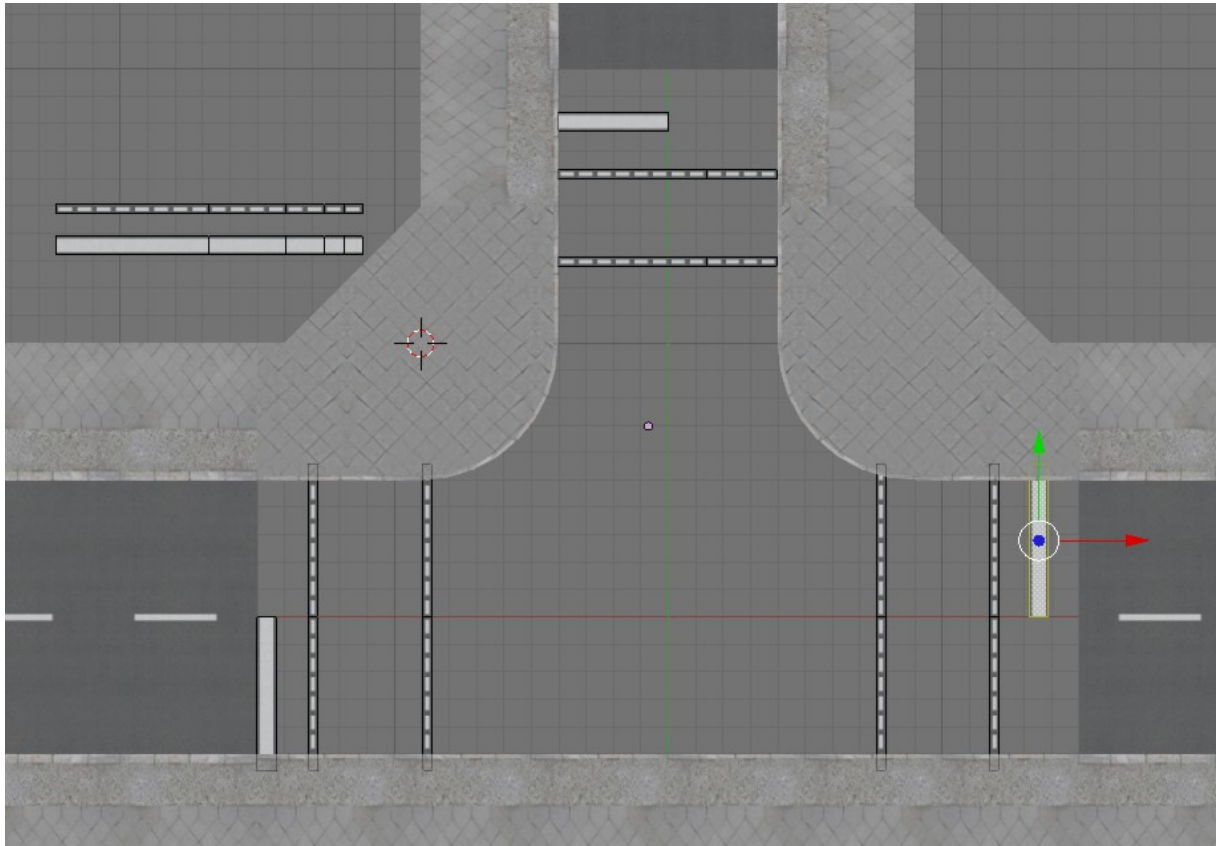
Nun geht's weiter mit der Straßenfläche. Hierbei gibt es wiederum entsprechende Vorlagen für die Bodenmarkierungen: eine Haltelinie, eine Radwegbegrenzung und eine Fußwegbegrenzung sowie Richtungspfeile und ein Kreuz aus vier Strichen.

Für diese Kreuzung werden lediglich Haltelinien und Fußwegbegrenzungen benötigt. Wie vorhin bereits mit der Bürgersteig-Vorlage wird verfahren: kopieren des Vorlagen-Objektes in Kreuzungsnähe und Wechsel in den Edit-Mode um darauf aufbauend die Fahrbahnoberfläche zu konstruieren. Bei der Gelegenheit können gleich (auf Edit-Mode-Ebene) die nicht benötigten Markierungen gelöscht werden. Außerdem wird das lange Segment des Fußweges kopiert und die Kopie in kleinere Teile aufgeteilt (jeweils markieren der oberen und unteren Kante im Edge-Mode und W / „Subdivide“), um für kürzere Abschnitte passende Segmente vorrätig zu haben:

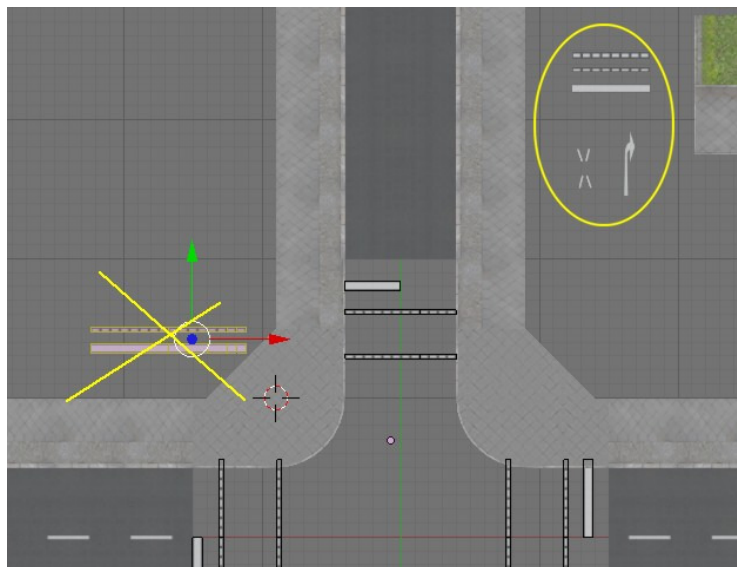


Nun können diese Segmente mit Kopieren (Shift+D), Drehen (R) und Verschieben (D) nach Lust und Laune platziert werden. Eine große Hilfe stellt natürlich auch hier wieder die Snapfunktion dar – aber auch hier sollte darauf geachtet werden, dass insbesondere bei

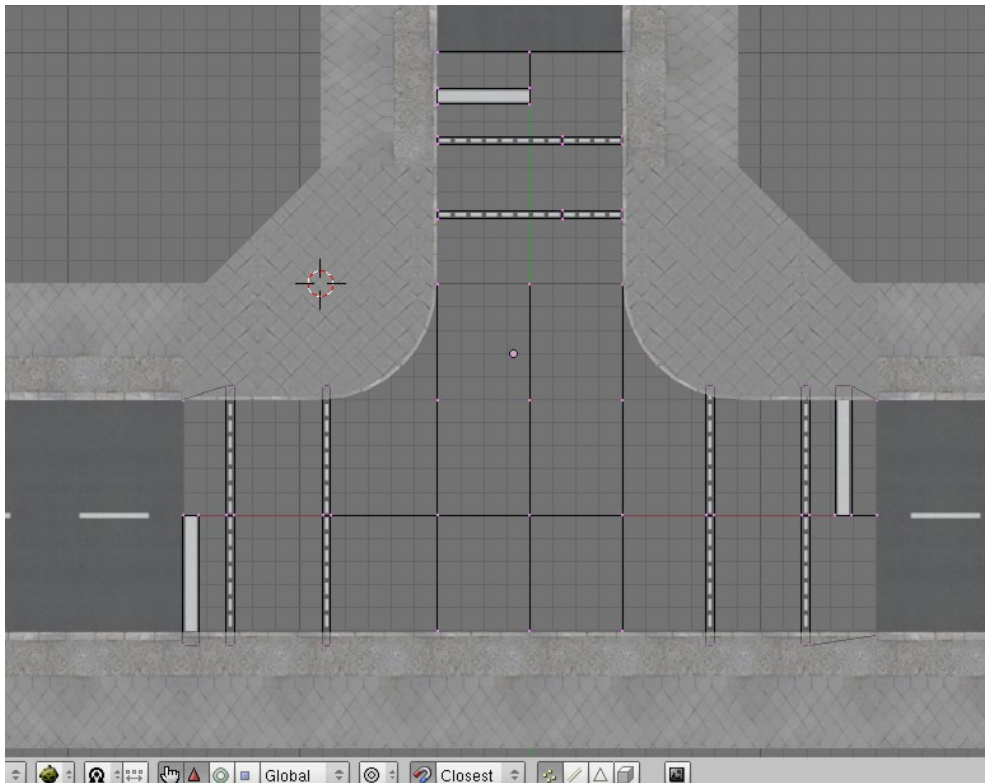
„Snaps“ mit dem Bürgersteig mit Shift+Z sichergestellt wird, dass alle Polygone in der ursprünglichen Höhe bleiben (0,1m bei Fahrbahnmarkierungen):



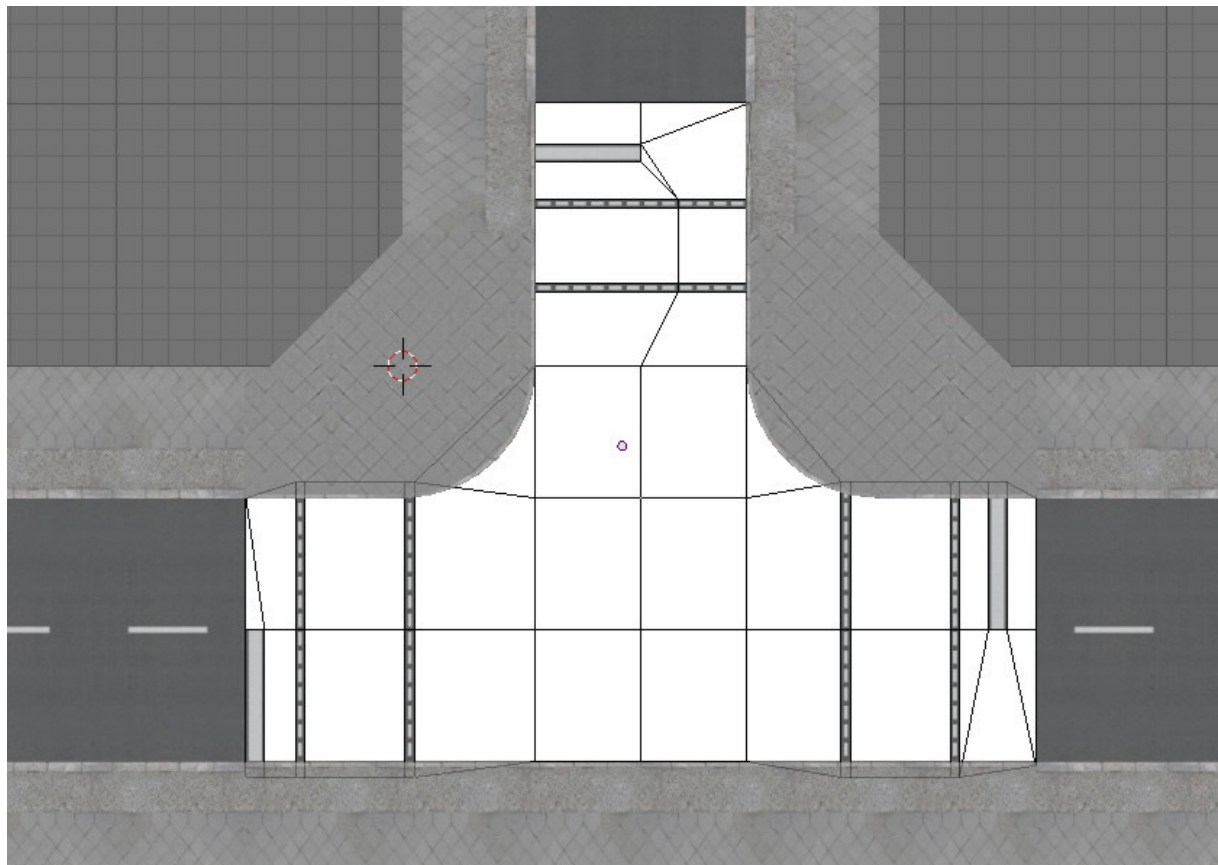
Wieder werden die Kopien der Vorlagen gelöscht, wohingegen die Originale oben rechts liegen bleiben:



Nun werden alle noch fehlenden Eckpunkte der Asphaltfläche hinzugefügt – es sollten keine allzu großen Polygone entstehen, weshalb auch einige Punkte in der Mitte hinzugefügt wurden. Wird beim Erstellen der Punkte jeweils die Extrude-Funktion (E) verwendet, lassen sich später die Flächen dazwischen leichter erstellen:



Das Erstellen der zwischenliegenden Flächen geschieht am schnellsten im Edge-Mode: es sind jeweils zwei gegenüberliegende Kanten zu selektieren und dann ist das Rechteck oder Dreieck mit „F“ zu erstellen, bis alle Flächen gefüllt sind:



Schließlich müssen diese Flächen noch texturiert werden. Hierbei wähle ich meistens wieder den Modus „Occlude Background Geometry“ (nicht vergessen, ihn wieder auszuschalten!) und die Selektions-Funktion „B“. Tipp: wird im Modus „B“ nicht die linke Maustaste sondern das Mausrad benutzt, um den Rahmen zu ziehen, werden die entsprechenden Flächen Deselektiert!

Wenn alle Flächen markiert sind, wird ihnen die „str\_asphdrk.bmp“ zugewiesen und mit einem „U“ => „Project from View“ ein neues Mapping verpasst. Dieses sollte noch so angepasst werden, dass die Auflösung in etwa gleichgroß der der Straßen ist (eine Texturkante sollte in etwa 5m Länge entsprechen):



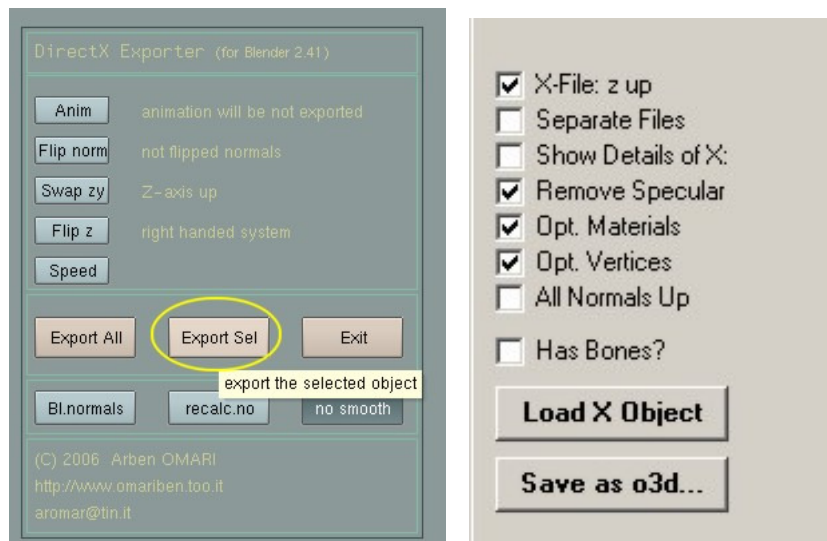
Damit ist das Kreuzungsmesh fertig!



Der nächste Schritt ist die Einbindung in Omsi.



Zunächst wird aus Blender eine x-Datei erstellt. Hierzu werden die beiden Objekte, der Bürgersteig und die Fahrbahn, selektiert und der Export mit den Optionen wie auf dem Screenshot sichtbar und mit dem Button „Export Sel“ durchgeführt. Die fertige x-Datei wird dann in eine o3d-Datei konvertiert.



Dann wird die \*.sco-Datei einer möglichst ähnlichen Kreuzung zunächst kopiert (entscheidend ist das Ampelsystem, was auf diese Weise zunächst übernommen wird) – ich verwende hierfür die Kreuzung „Kreuz\_See\_Elsflether.sco“.

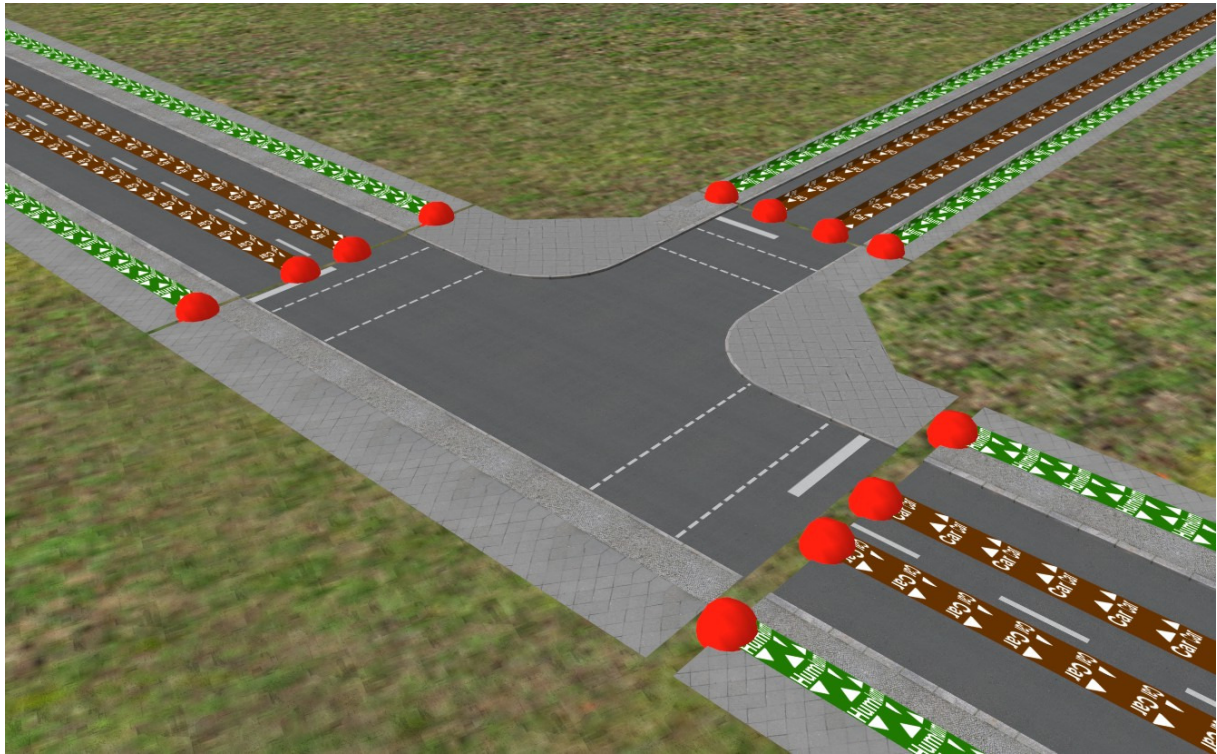
Die Kopie, die entsprechend dem Blender-Namen „Einm\_See\_Pawesiner.sco“ genannt wird, wird wie folgt bearbeitet:

- unter „[friendlyname]“ wird der neue Name eingetragen: „Junction Seeburger Str./Päwesiner Weg“
- alles zum (exklusive!) ersten „[splinehelper]“ bleibt unverändert.
- alle „[splinehelper]“- und alle „[path]“- / „[use\_traffic\_light]“-Einträge werden entfernt (sie werden für diese Kreuzung mit dem Kreuzungs-Editor neu erstellt)
- ganz unten steht „#### Model Data ####“ und darunter die o3d-Datei. Diese wird natürlich auf „einm\_see\_paewesiner.o3d“ geändert.

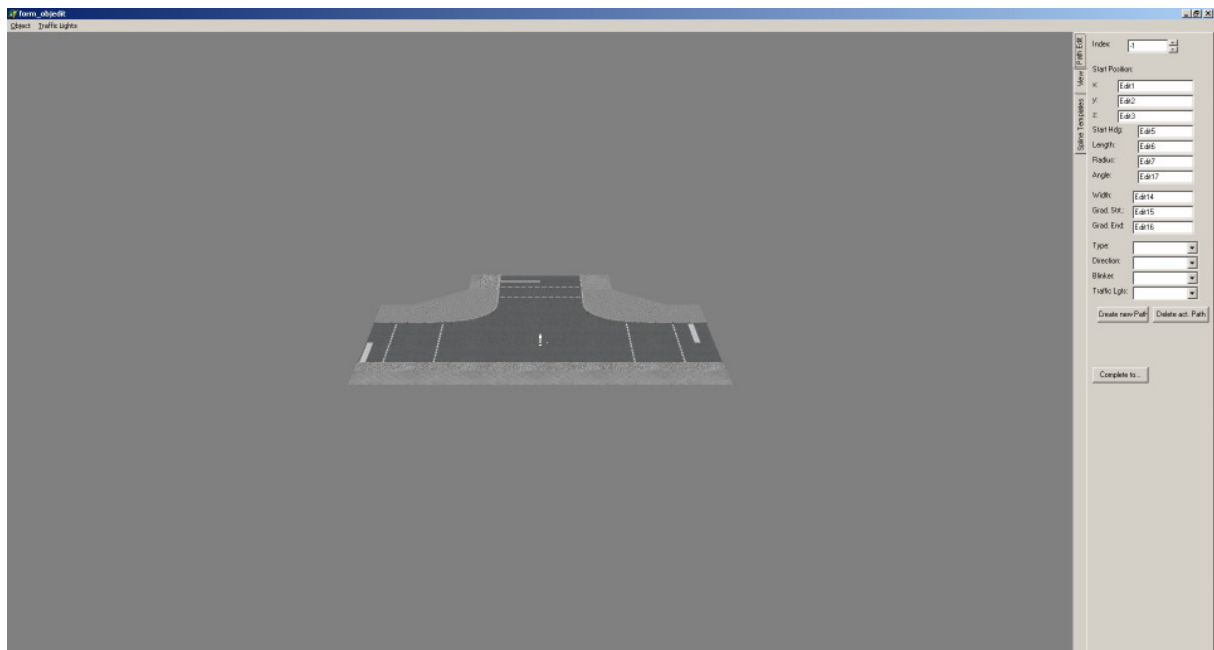
Die o3d-Datei kommt in den entsprechenden model-Ordner, die Texturen sollten bereits alle vorhanden sein – in diesem Fall.

Die Kreuzung sollte nun als Objekt im Omsi-Editor verfügbar sein. Allerdings verfügt sie noch nicht über die sog. Pfade, worauf die AI-Fahrzeuge fahren und welche dem Map-Editor die Informationen über das Andocken an bestehende Straßen liefern.

Im Omsi-Map-Editor:



Nun sollen die Pfade hinzugefügt werden. Dazu wird der Kreuzungs-Editor „OmsiObjEditP.exe“ gestartet, im Menü „Object“ => „Load“ aufgerufen und die neue Kreuzungsdatei „einm\_see\_paewesiner.sco“ aufgerufen.



Die Bewegung der Kamera erfolgt hier ähnlich wie in Omsi: Rechtsklick = Neuer Zentrumspunkt, Mausrad/Mitteltaste-Drag'N'Drop = schwenken. Für den Zoom gibt's einen Regler auf der Registerkarte „View“.

Als erstes werden wieder die angrenzenden Straßen wie folgt platziert:

- Wechsel auf die Registerkarte „Spline Templates“



- einmal auf „Create“ klicken
- auf „Load“ klicken und die entsprechende Spline-Datei auswählen (hier: „str\_2spur\_11m\_SeeburgerStr1.sli“)
- die Koordinaten eingeben (können direkt aus Blender entnommen werden!) und Enter drücken

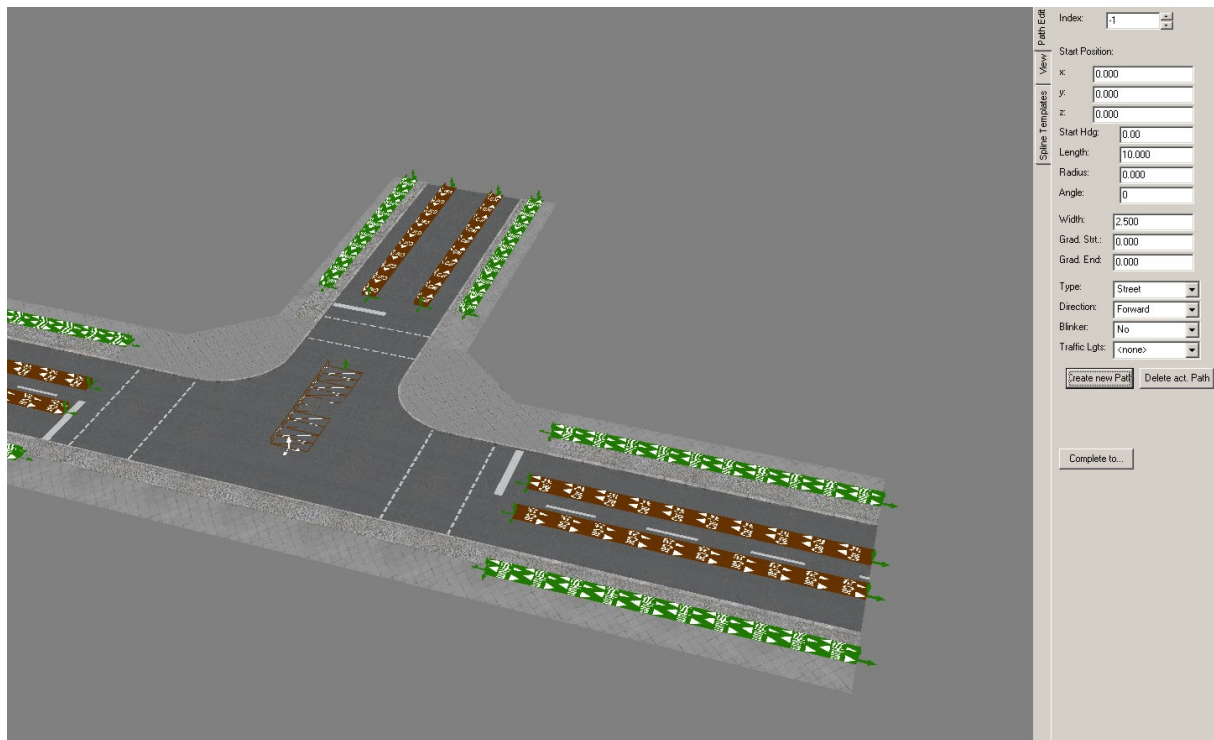
Dann sollte das Straßensegment korrekt liegen. Mit „Create“ wird dann das andere Straßenstück gegenüber erstellt – hier muss man nicht noch mal auf „Load“ klicken – denn der Straßentyp stimmt ja bereits. Beim dritten Straßensegment, dem Päwesiner Weg, ist es aber wieder nötig, da hier ein anderer Splinetyp verwendet wird („str\_2spur\_8m\_altonaer1.sli“):



Es gilt sowohl hier als auch beim späteren Bearbeiten der Pfade: Das jeweils markierte Straßensegment blinkt. Die Straßensegmente können dann mit dem Nummernfeld ganz oben durchgeschaltet und bearbeitet oder gelöscht werden.

Nachdem nun alle angrenzenden Straßen als Vorlagen eingefügt wurden, können die Pfade verlegt werden.

Zuerst ist auf die Registerkarte „Path Edit“ zu wechseln, dann mit „Create“ ein neues Pfadsegment zu erstellen:



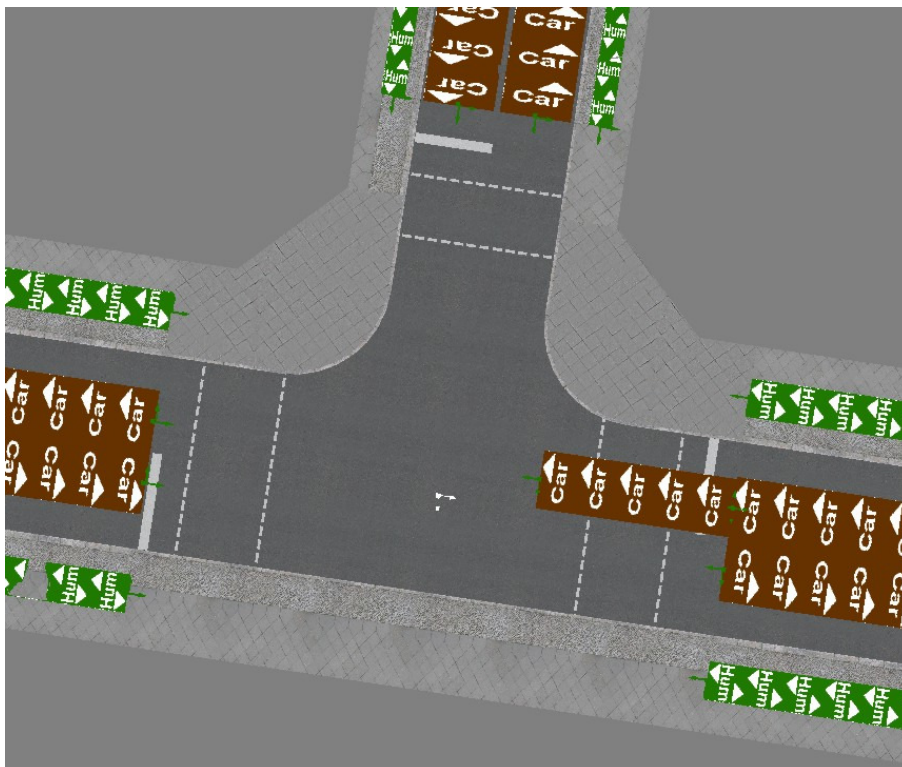
An den Enden der Pfade befinden sich kleine grüne Symbole mit drei Achs-Pfeilen. Diese können verwendet werden beim Platzieren der Pfade: drücken Sie Strg und klicken Sie dabei auf das Ende eines Pfades – dann wird der Pfad dort angedockt.



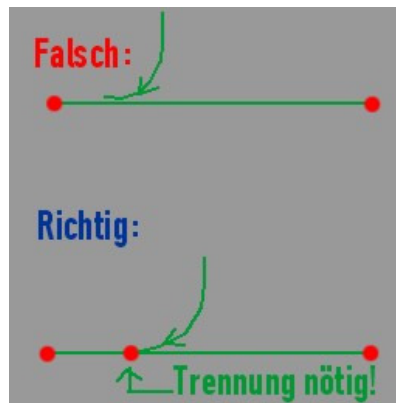
Kommen wir nun zur Breite der Pfade: der Übersicht halber werden alle Pfade mit 1m Breite gezeichnet, nur das aktuell selektierte wird in Originalbreite gezeichnet. Sollen alle Pfade in Originalbreite gezeichnet werden, kann die Option „True width of paths“ aktiviert werden, diese befindet sich auf der „View“ Registerkarte:



Klar erkennbar ist, dass die neue Spline nicht ganz so breit ist wie die Vorherige. Dies soll nun angepasst werden – durch ausprobieren kann man schnell feststellen, dass die Vorherige eine Breite hat von ungefähr 3m. Deshalb soll in das Feld „Width“ nun „3“ eingetragen werden. Bestätigen mit Enter ist nötig, damit die Breite übernommen wird (so wie bei allen anderen Feldern mit Texteingabe auch)



Zusammenführungen und Aufspaltungen von Pfaden müssen stets an einem gemeinsamen Punkt erfolgen. D.h. „verboten“ wäre es, wenn man in der Geraden von rechts nach links nur *einen* Pfad erstellen würden und von oben her kommend eine Kurve so münden ließe, dass diese irgendwo in der Mitte des Pfades enden würde. Stattdessen ist es nötig, die Gerade an der entsprechenden Stelle in zwei Pfade aufzuteilen.



Aus diesem Grund ist es immer am Besten, wenn man mit den Rechtsabbiegern beginnt.

Hierbei kommt eine weitere Funktion zum Einsatz, die Auto-Vervollständigung. Sie versucht eine Lösung zu finden, die eine Verbindung zwischen einem bestehenden Pfad und einem gewünschten Endpunkt aus einem Bogen und einer Geraden (davor oder dahinter) darstellt.

Hierzu wird auf „Complete to...“ geklickt (die Beschriftung wird fett/kursiv) und dann auf den gewünschten Zielpunkt. Die Beschriftung wird wieder zurückgesetzt und wenn es möglich war, dann ist nun die Lösung zusehen:



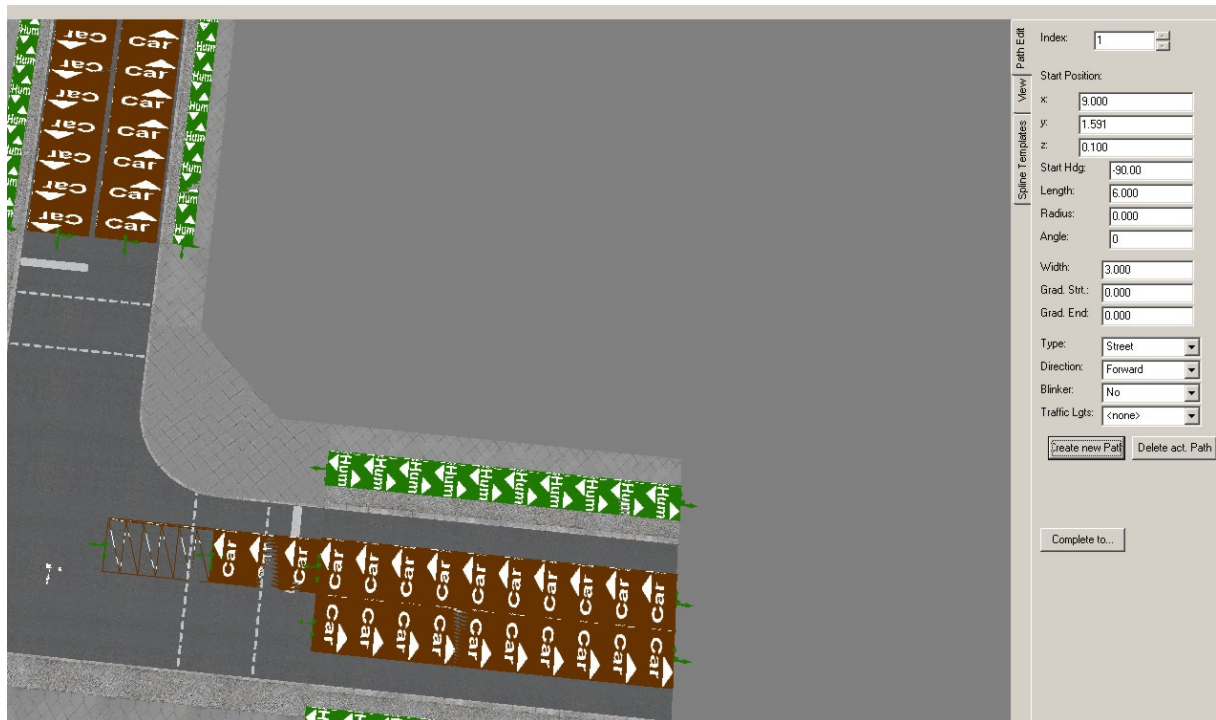
Leider ist diese Lösung ungeeignet... deshalb wird *einmal* auf „Delete act. Path“ geklickt (der Bogen ist ja das bereits vorher erstellte Pfadstück, das lediglich einmal von der Vervollständigungsfunktion verformt wurde).

Der übrig gebliebene Pfad wird nun wieder „geradegebogen“ (Eingabe von Radius = 0) und auf 6m Länge zurückgestellt.

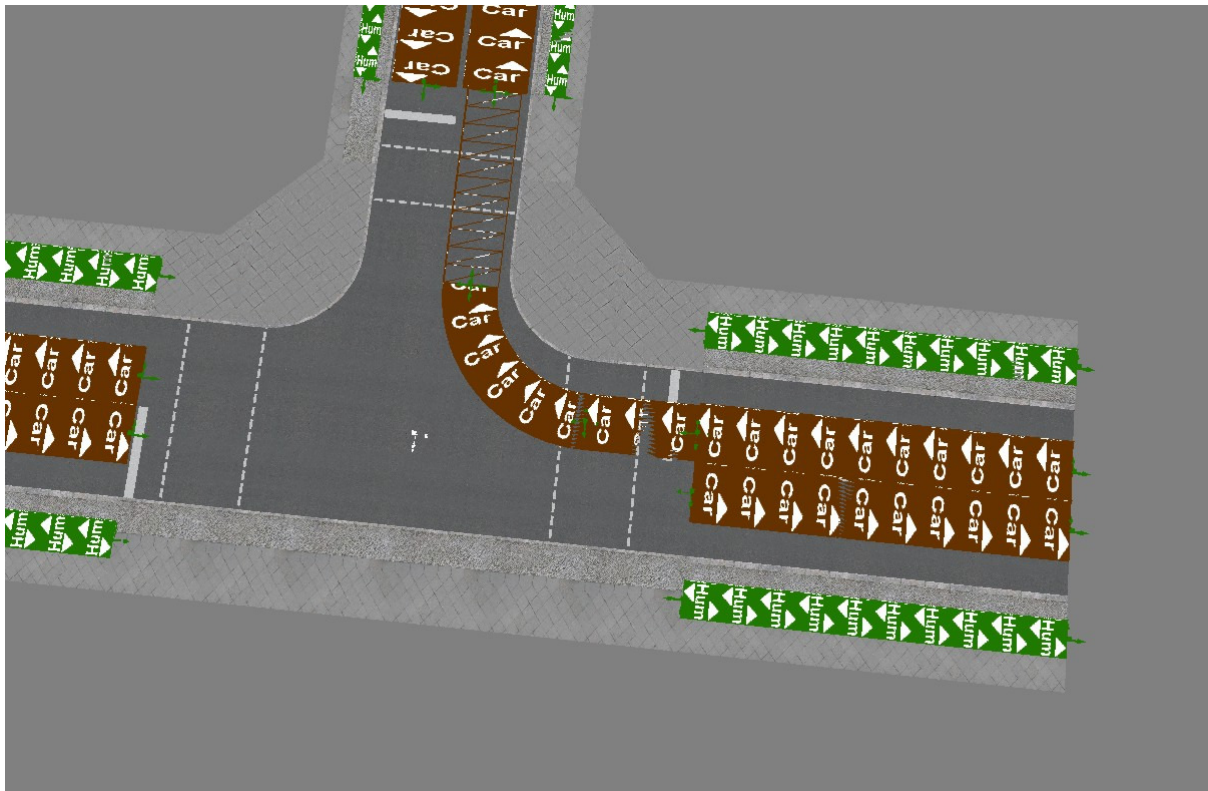
Damit dieses Stück nun *nicht* einbezogen wird, muss ein weiteres Pfadstück angehängt werden, von welchem aus dann die Vervollständigung durchgeführt wird:

Zuerst also wird mit „Create“ ein weiteres Splinstück erstellt. Dies wird automatisch ans aktuell selektierte angehängt und übernimmt dessen Eigenschaften.





Nun kann, wie oben erklärt, die Funktion „Complete to...“ angewendet werden. Da in diesem Fall nur das zweite, zuletzt erstellte Pfadstück verwendet wird, sieht die Lösung nun gut aus:

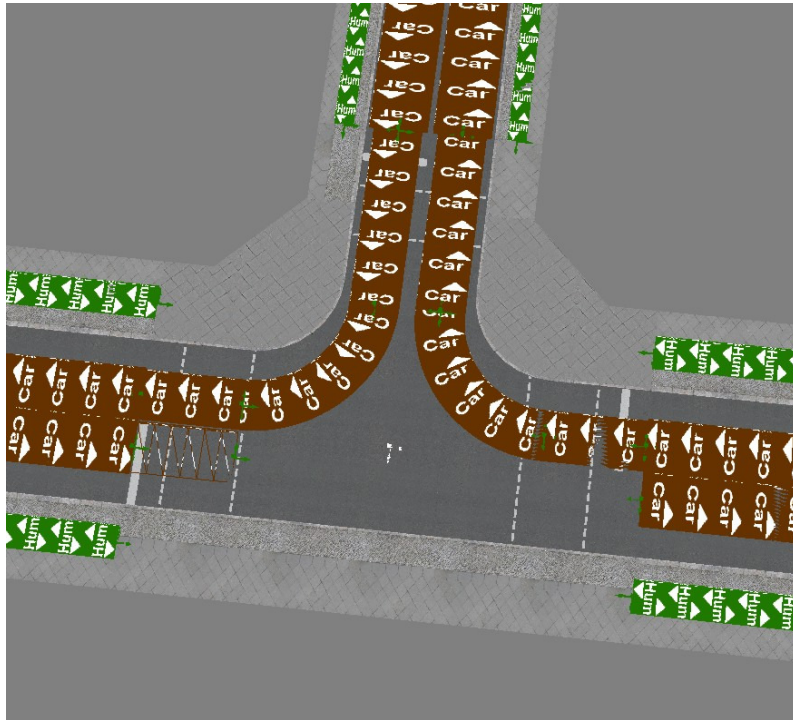


Merke: beim Auto-Vervollständigen mit „Complete to...“ wird stets das aktuell selektierte Pfadstück mit verarbeitet!

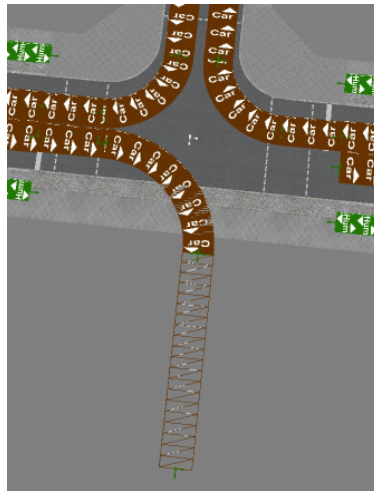
Das nun selektierte Pfadstück oben rechts kann nun mit „Create“ kopiert werden (dann liegt es – beinahe unsichtbar – innerhalb der oben ansetzenden Straße) und an die Gegensepur mittels Strg + Mausklick platziert werden. Weil genauso wie eben verfahren werden soll und







Nun funktioniert „Complete to...“ u.U. nicht korrekt: denn jetzt liegen oben zwei entgegengesetzte „Ansatzpunkte“ übereinander. Deshalb kommt es zu folgendem Bild:



Dieses Problem kann aber umgangen werden, in dem (nach einmaligem Löschen der Geraden!) ein weiteres Mal „Complete to...“ verwendet wird – aber diesmal beim Klicken auf den Zielpunkt die Shift-Taste gehalten wird! Dann wird nämlich „umgekehrt“ andockt und das Ergebnis stimmt:

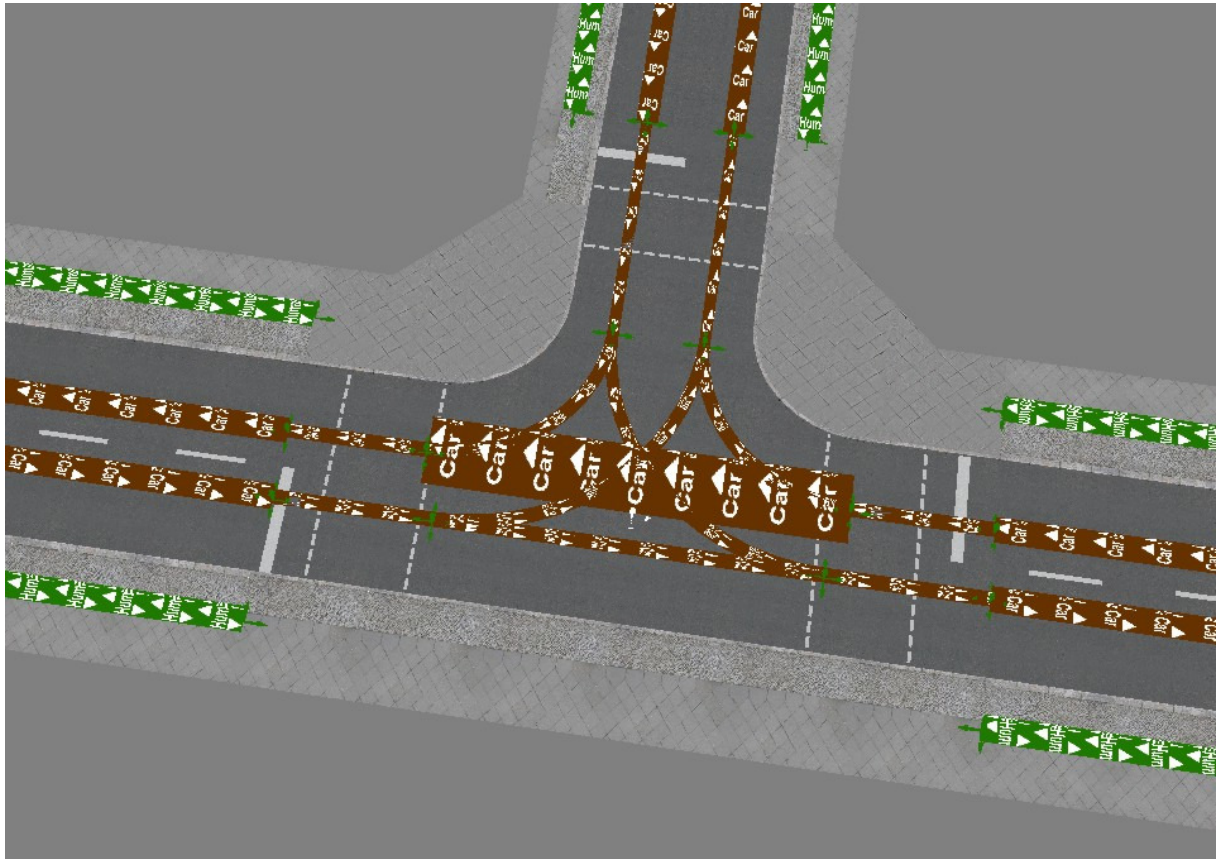


Um den anderen Linksabbieger zu erstellen, muss das neue Pfadstück nun oben links zwischen Gerade und Rechtsabbieger platziert werden – auch hierbei hilft die Shift-Taste beim Platzieren um die richtige Richtung auszuwählen. Nach dem Vervollständigen sollte das Ganze so aussehen:



Zu beachten ist, dass die Pfeile auf den Pfaden stets korrekt ausgerichtet sein müssen!

Als letztes werden nun die Geraden zwischen den Abbiegespuren platziert, auch hier hilft meistens die Auto-Vervollständigung – allerdings kommt es manchmal auch zu Fehlern! Dann müssen die erstellten Segmente wieder gelöscht werden und muss ggf. die Länge durch Ausrechnen oder Ausprobieren ermittelt werden. Wenn alles fertig ist (Pfad-Breite nun wieder nur 1m):

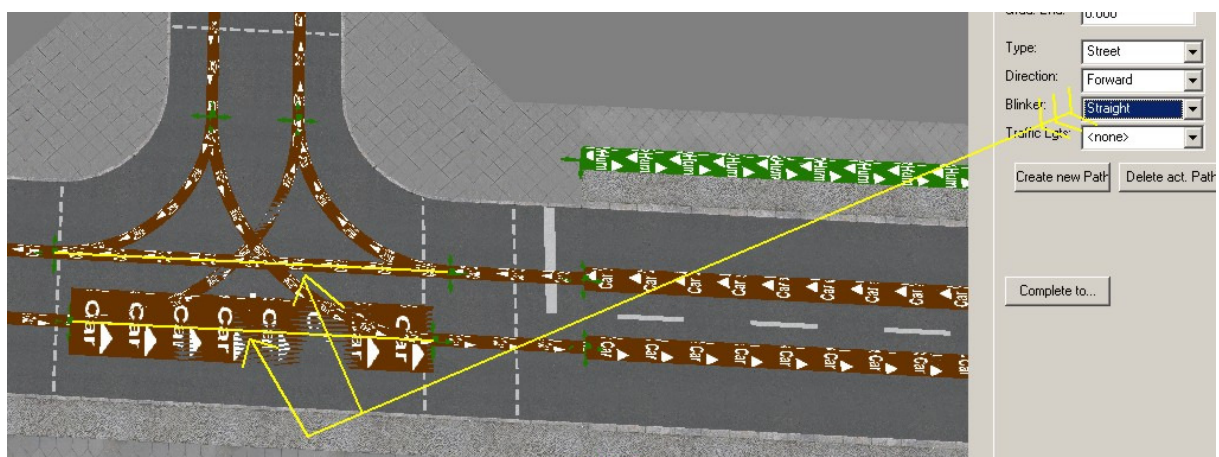


Die Fußwege können später ähnlich erstellt werden (siehe hierzu bitte bestehende Kreuzungen), jetzt sollen noch zum Abschluss die Blink-Eigenschaften und die Zuweisung der Ampelschaltung vorgenommen werden.

An dieser Stelle besonders wichtig: durch einfaches klicken mit der linken Maustaste können Pfade gezieht ausgewählt werden zum Bearbeiten oder Löschen!

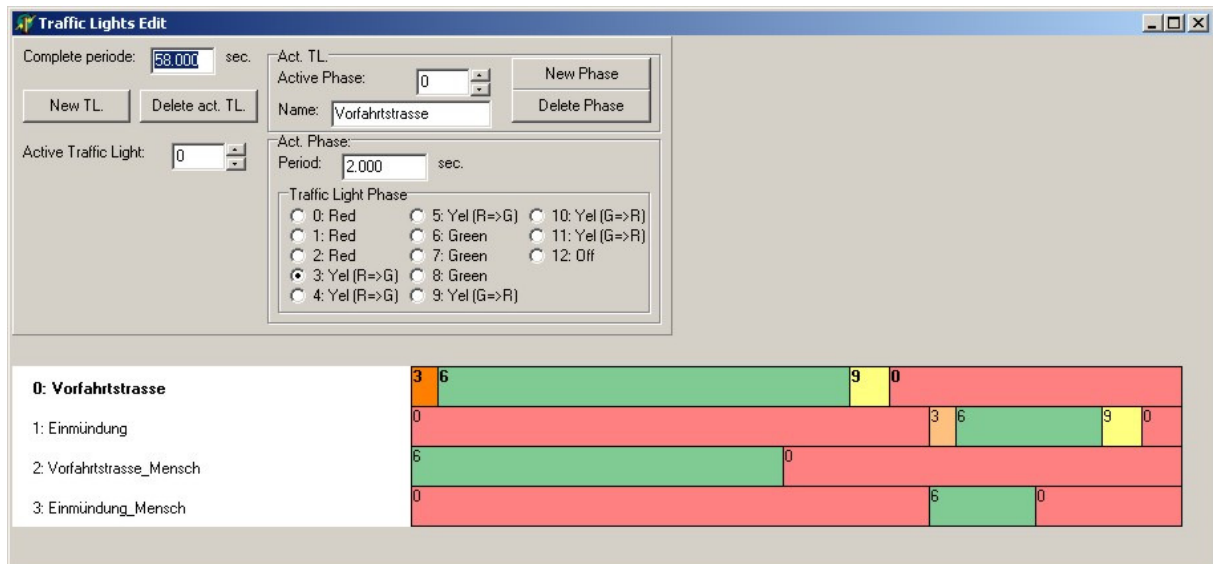
Die Eigenschaft „Blinker“ wird nun bei allen Rechtsabbiegern auf „Right“ und bei allen Linksabbiegern auf „Left“ eingestellt.

Bei den geraden Verbindungen sollte die Eigenschaft „Straight“ eingestellt werden – damit wird sichergestellt, dass die Autos, die dort langfahren, auch wirklich nicht (bspw. schon für eine nächste Kreuzung) blinken.





Als letztes nun schauen wir uns ganz kurz den Ampel-Editor an. Hierzu wird auf „Traffic Lights“ und „Edit“ geklickt und die entsprechende Warnmeldung bestätigt. Diese Warnmeldung weist darauf hin, dass es in diesem Dialogfeld keinen „Abbrechen“-Knopf gibt und alle Änderungen sofort Gültigkeit erlangen (natürlich kann man immer noch abbrechen, indem man die Datei verwirft – aber dann verliert man auch alle Änderungen seit dem letzten Speichern!)



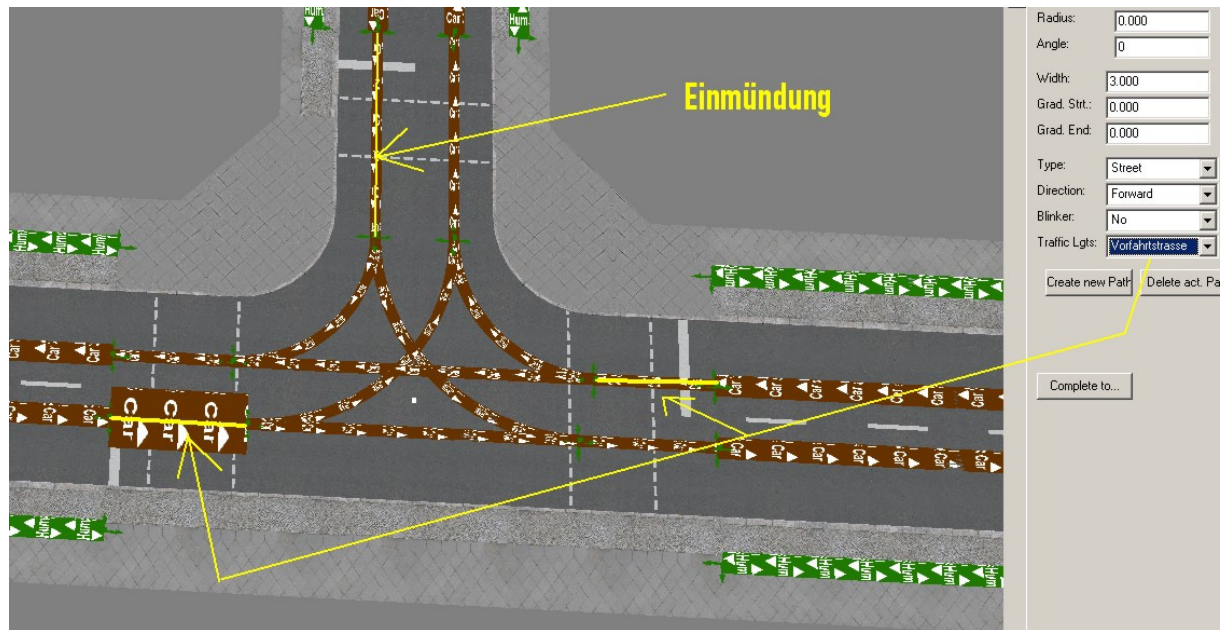
Auf den Ampel-Editor soll später noch detaillierter eingegangen werden.

Zunächst soviel: Jede Kreuzung verfügt über mehrere „TLs“ (Traffic Lights), bspw. eine Ampel für die Vorfahrtstraße, eine für die Einmündung usw.

Jede „TL“ besteht aus mehreren „Phasen“, die nacheinander „abgespielt“ werden.

Im Schaubild unten sieht man dann, in welcher Reihenfolge und wie lange welche Phasen gezeigt werden – begonnen wird, wenn die Hauptrichtung auf „Rot-Gelb“ wechselt (im Schaubild durch Orange gekennzeichnet). Gleichzeitig schaltet die Fußgängerampel in Hauptrichtung auf grün. Dann diese wieder auf rot, kurz darauf die Kfz-Ampel auf gelb und auf rot. Dann dasselbe mit der Einmündung, bis alles wieder von vorne anfängt.

Wir schließen nun den Ampeleditor (die Phasen sind so gut! ;-) ) und weisen jeweils dem ERSTEN Segment, welches bei Rot NICHT mehr befahren werden darf, die entsprechende Ampelphase zu:



Die dahinterliegenden Segmente werden NICHT mit einer Ampel verknüpft – denn auch wenn rot ist, sollte ein Auto, welches sich auf einem solchen Pfadstück befindet, weiterfahren um die Kreuzung zu räumen! ;-)

Tchjaaa... nun eigentlich nur noch speichern – und dann ist die Kreuzung einsatzbereit!

**Postscriptum:** jedes Mal, wenn im Kreuzungseditor gespeichert wird, legt dieser für alle Fälle eine Backup-Datei an (Originaldateiname + Datum + Uhrzeit + „.bu“) – wenn alles funktioniert, können diese Dateien gelöscht werden!