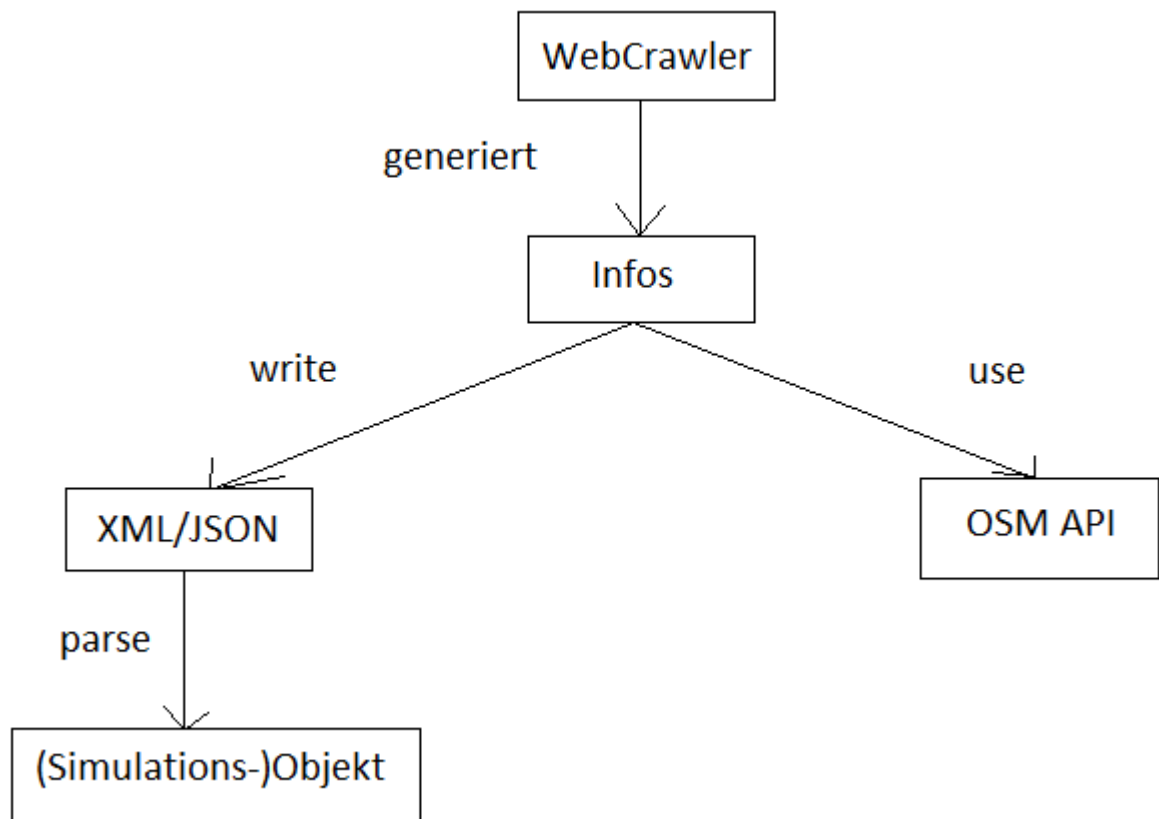


Ergebnisse des Treffens vom 27.11.2017:

- Das nächste Treffen findet am 18.12.17 um 13 Uhr statt (vorher geht es bei Patrick nicht)
- MovementDirector (MD) und Distanzmatrix (DM) funktionieren schon.
- Es hat sich eine Änderung des Projekts ergeben: Der Hauptautor (Nils), möchte nun an dem Projekt weiterarbeiten und verhindern, dass wir an dem schon vorhandenem funktionierendem Code etwas ändern. Deshalb möchte Nils den MD aus MiniWorld rausziehen und eine API anbieten, auf der man DMs setzen kann. Somit würden zwei voneinander unabhängige Programme entstehen: die Simulation von MiniWorld an sich und die Simulation der Bewegungsabläufe über die Distanzmatrix. Unsere Aufgabe wäre es dann die Simulation der Bewegungsabläufe der Distanzmatrix zu entwerfen. Das bedeutet insbesondere, dass wir uns zunächst nicht mehr mit dem schon vorhandenen Code auseinandersetzen müssen, sondern ein eigenes erstmal unabhängiges Projekt entwerfen können.
- Dadurch, dass Nils eine API für MiniWorld erstellt, ist es möglich, ihm die von uns erzeugten Distanzmatrizen zu übergeben.
- Das Ziel wäre somit ein Programm zu schreiben, welches am besten sekundlich eine Distanzmatrix erzeugt. Hierfür sollten wir am besten OSM benutzen und dafür ggf. die Daten in OSM erweitern/ändern
- Zu der Benutzung von OSM:
 - OSM erhält seine Daten dadurch, dass User dort neue Informationen eintragen (z.B. einen neuen Knoten (tag) für ein in OSM noch nicht vorhandenes Restaurant eintragen) oder schon vorhandene Informationen ändern (z.B. in der Straße XYZ ist nun eine 30er-Zone statt eine 50er-Zone)
 - Diese Änderungen können auf zwei verschiedene Weisen ablaufen:
 1. Änderungen verlaufen online, das heißt man meldet sich bei OSM an, ändert/aktualisiert mit der API von OSM die Daten in OSM und liest diese mit einem Parser für OSM (müsste schon existieren) aus und verwendet die so erhaltenen Infos/Daten dann für die Distanzmatrix (oder Listen von Restaurants, Hotels, etc.)
 2. Änderungen verlaufen lokal auf dem Rechner, d.h. indem man mit einem Parser für OSM (müsste schon existieren) die OSM-Files in Objekte einer entsprechenden Programmiersprache umwandelt, diese dann ändert, und anschließend mit einem erneuten Parser oder WebCrawler diese Objekte in OSM einließt (also hochlädt, sodass sich OSM online ändert)
 - Für die erste Variante benötigen wir also den Parser, der die Daten aus OSM ausliest, und für die zweite Variante sind benötigen wir zusätzlich noch einen zweiten Parser, der die geänderten Daten wieder in OSM hochlädt
 - Der Gedanke, dass wir OSM erweitern, ist optional und wie folgt gemeint:
 - Falls uns die Daten in OSM nicht ausreichen, können wir versuchen OSM durch Daten/Infos von Plattformen wie zum Beispiel YELP zu aktualisieren und zu erweitern. Dies würde nicht nur uns helfen, sondern der kompletten OSM-Community.
 - Dies wäre zum Beispiel möglich, indem wir einen WebCrawler für YELP entwickeln, um die Daten aus YELP für OSM nutzen zu können
 - Dies kann wiederum auf zwei Arten erfolgen:
 1. online, indem die API von OSM benutzt wird, um die Daten des WebCrawlers in OSM zu importieren
 2. offline, indem wir die Daten des WebCrawlers in XML- oder JSON-Daten schreiben und anhand eines Parsers Simulationsobjekte für unsere Bewegungen und Distanzmatrizen erzeugen
- Als ersten Schritt dahin hat Patrick folgende Aufgaben genannt, welche am besten in 2er-Teams gelöst werden können:
 1. Das erste Team schaut sich OSM genauer an:
 - gibt es schon einen OSM-Parser, welcher uns die Daten aus OSM in Objekte der jeweiligen Programmiersprache ausgeben kann?
 - Allgemein sich mit OSM vertraut machen
 2. Das zweite Team schaut sich den WebCrawler an:
 - überlegen, ob man die OSM-API benutzen möchte (online) und dann nur einen Parser benötigt, um die Daten aus OSM zu ziehen

- oder ob man selbst Objekte erzeugt (XML/JSON) und diese dann mit einem zweiten Parser in OSM importiert
- Anschaulich kann man es sich wie folgt vorstellen:



- Generell ist der Gedanke so, dass der WebCrawler nicht bei jedem Aufruf der Simulation von MiniWorld die Daten von OSM abrufen soll, sondern die Daten zum Beispiel nur alle 2 Wochen aktualisieren/abrufen soll
- Sonstige Gedanken für die Zukunft:
 - Die „Menschen“ in MiniWorld sollen der Einfachheit halber zunächst erstmal nur einen Wohnort (zum Beispiel Marburg) und einen Arbeitsort (zum Beispiel Wehrda) besitzen
 - Dabei sollen sie den Wohn- und Arbeitsort zufällig wählen dürfen (nur einmal?)
 - Anschließend sollen die „Menschen“ zwischen den beiden Orten hin und her laufen
 - Hierfür könnte man gucken, ob ein Routing für OSM-Karten existiert