UNIVERSITÄT OSNABRÜCK

Lecture: Map-based - PACMAN - programming internship

Play, Assess, Cooperate - Modeling Agile Networks PACMAN

Authors: Henrik Gerdes

Abstract

Getting into agile network modeling, requires knowledge about the network technology stack and human movement processes. The PacMan-Game simplifies this by offering a visually appealing and playful way into agile networks and movement simulations.
Players can navigate Pacman on real streets to capture as many packages (fig. 1 red circle) as possible and deliver them to their destination (fig. 1 blue circle). Other players and ghosts should be avoided as their contact will lead to package loss.

Game Architecture

Pacman uses a classic client-server architecture. The communication layer is provided by Pod-SixNet and uses TCP/IP to synchronize player position, map selection and difficulty.

Osnabrueck and Tokyo gameplay

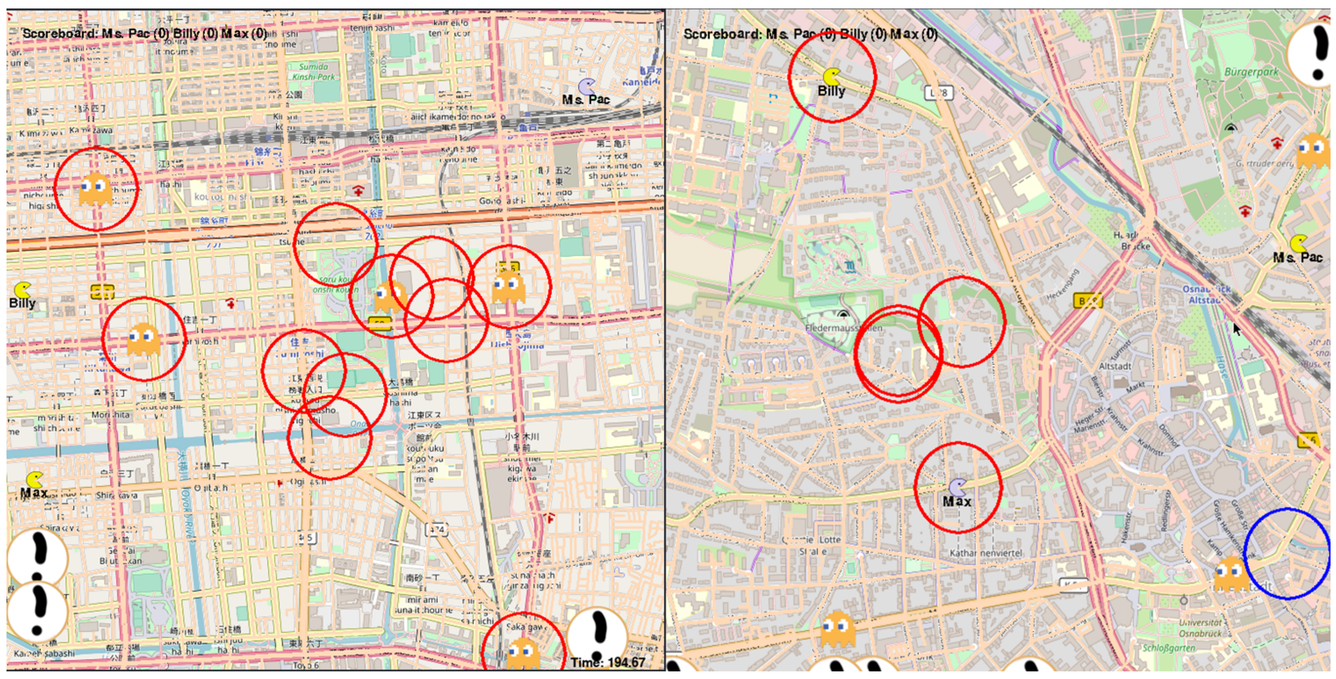

Figure 1: Gameplay Screenshots: Tokyo (left); Osnabrueck (right)

Data source and Ghost movement

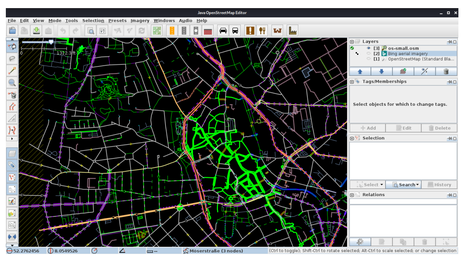

Figure 2: Visualisation of OSM-data in JOSM

OpenStreetMaps (OSM) provides the data for the cities, streets, walkways and point of interests. A custom written extension of the osmium library allows to filter the OSM data for car- and highways.

These streets than get aggregated in a graph-structure provided by networkx. This ensures one big connected street grid. To improve performance this grid gets cached after the first run.
This allows great flexibility in playable maps. Every region with OSM data can be used. Due to very different city rod densities playability may differ. New OSM-data can be downloaded and managed with **JOSM** (fig. 2).

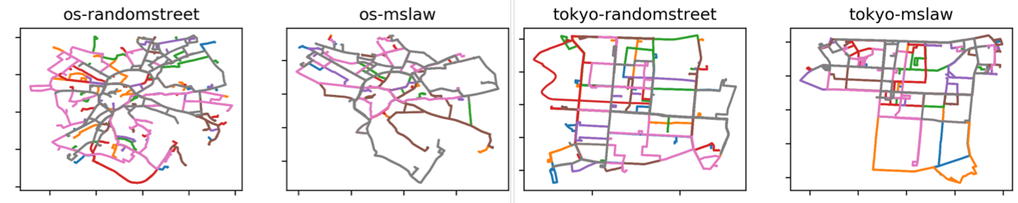

Figure 3: Ghost paths in Tokyo and Osnabrück

In addition to the other players, ghosts pose a challenge for package delivery. The ghosts move based on pre-generated traces. Trace generation is done by scientific mobility models to simulate human movement. **BonnMotion** provides various movement models by using the same OSM data that is used for the game. Users can choose between the MSLAW and RandomStreet model. This allows visualization of differences between the models, different gameplay on the same map and difficulty differences. Figure 3 shows the traces for every ghost on various maps.

Impact of movement parameters on difficulty

	MSLAW/Randomstreet Speed parameter				
Speed	0-8 m/s	2-10 m/s	4-12 m/s	6-14 m/s	8-16 m/s
AVG-Time	7.28 / 7.26	10.74 / 9.75	107 / 21.5	180 / 52.0	262 / 97.0

	MSLAW/Randomstreet Ghost number				
GhostNum	8	10	12	14	16
AVG-Time	10.74 / 9.57	10.82 / 9.28	11.5 / 9.80	10.3 / 9.91	10.9 / 8.91

Figure 4: Impact on average package get time

Furthermore, specific mobility model parameters found to have impact on game difficulty. The results of a routing-service controlled game-ai show (fig. 4) it gets increasingly more difficult to get packages (AVG-Time) and deliver them. It also shows that RandomStreet is consistently more difficult than MSLAW.

Signal propagation

The signal propagation is illustrated by the time it takes to take a package. Signal limits are the circles, the denser the player is in the middle, the faster it is taken. Additionally Pacman has an optional mode to display realistic signal propagation shown in fig. 5. The signal propagation is calculated by **RaLaNS**. The calculation takes into account obstacles such as buildings.

RaLaNS

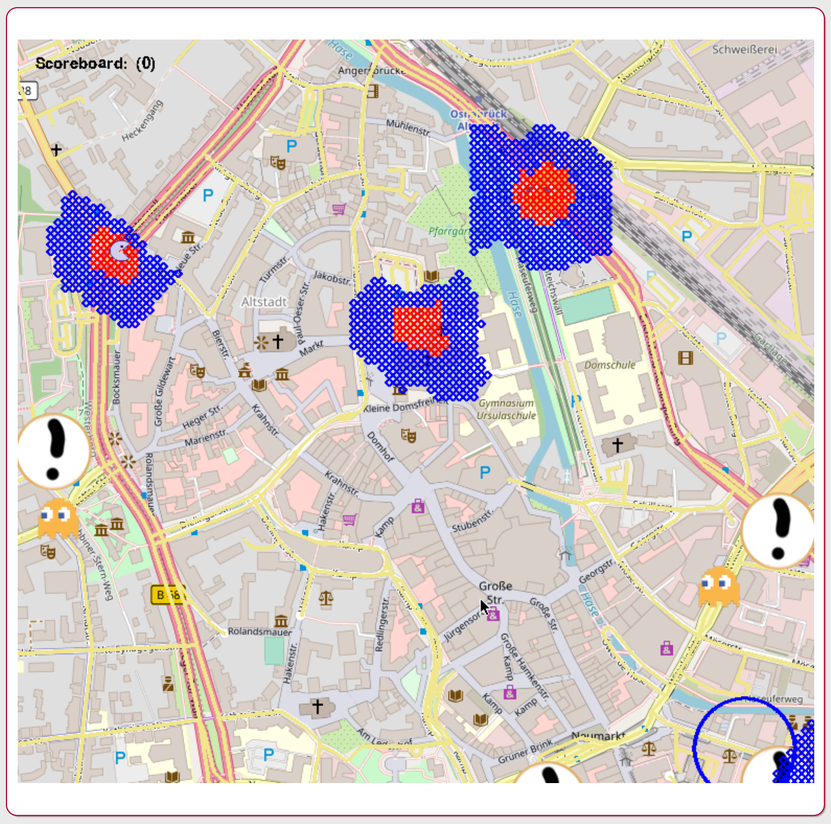

Figure 5: Realistic signal distribution with RaLaNS-data

Further information:
M. Schwamborn, N. Aschenbruck, “Introducing Geographic Restrictions to the SLAW Human Mobility Model,”
BonnMotion, Source: www.sys.cs.uos.de/bonnmotion/
RaLaNS, Source: www.sys.cs.uos.de/ralans/
Did i miss something?

Distributed Systeme
https://sys.cs.uos.de/