



POLITECNICO
MILANO 1863

FAST COMMUNICATION LEARNING THROUGH ASYMMERICAL MULTIPLAYER VIDEOGAMES

Relatore:

Marco Gribaudo

21/12/2021

Jacopo Grandi

Serious Games

Serious game definition:

“Any piece of software that merges a non-entertaining purpose (serious) with a videogame structure (game)”

Introduction

Design and implemetation of a videogame that trains the communication skills of the players.

Contents

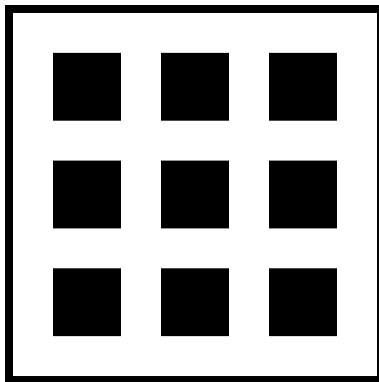
- ▶ Game Design
- ▶ Level Design
- ▶ Traffic Simulation
- ▶ Player movement: Bicycle model
- ▶ UDP Infrastructure
- ▶ State Synchronization
- ▶ Audio: Radio and Sounds
- ▶ Tests: TDD

Contents

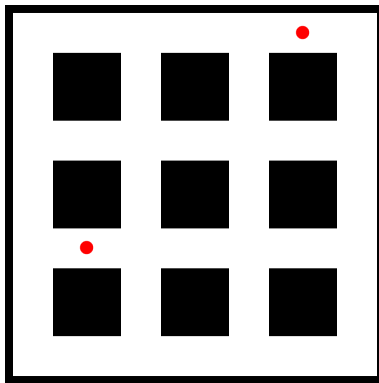
- ▶ Game Design
- ▶ Level Design
- ▶ **Traffic Simulation**
- ▶ Player movement: Bycycle model
- ▶ UDP Infrastructure
- ▶ State Synchronization
- ▶ Audio: Radio and Sounds
- ▶ Tests: TDD

Game Design

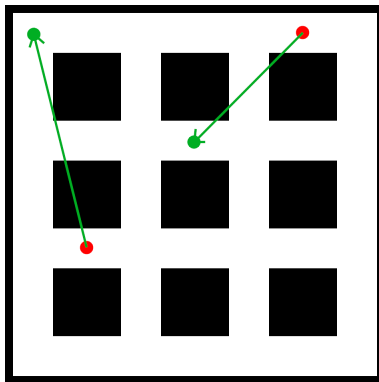
The game is set in a city with a manhattan road layout



There are items scattered on the roads



Items have to be carried to their destination



The city is filled with car traffic and it's subdivided by drawbriges.

The players are either Couriers or Radio Operators:

- ▶ Couriers bike through the city and carry the items
- ▶ Radio Operators have a map of the city

Players can communicate by an half duplex radio.

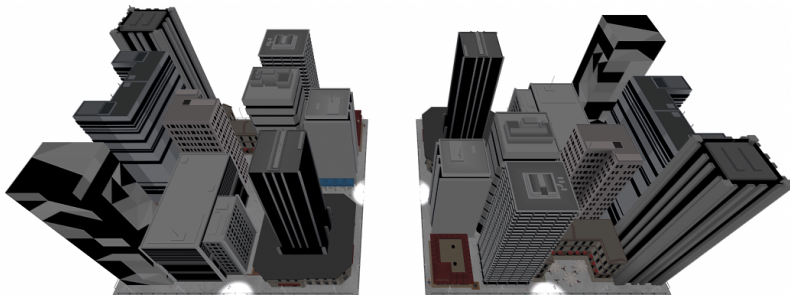
Couriers and Radio Operators have different information:

- ▶ The item location and destination is known by the Radio Operators
- ▶ The Radio Operators don't know the location of the Couriers
- ▶ The traffic levels are known to the Radio Operators
- ▶ The drawbridges state (open/close) is not known by the Radio Operators

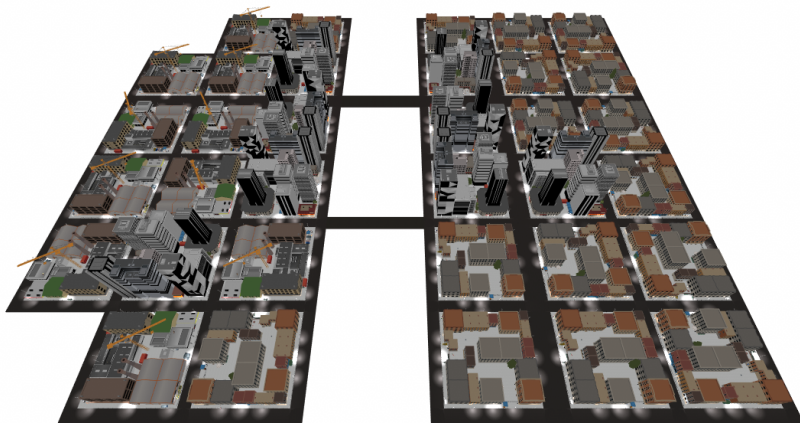
The players win if they can carry all items to their destination within a time limit.

Level Design

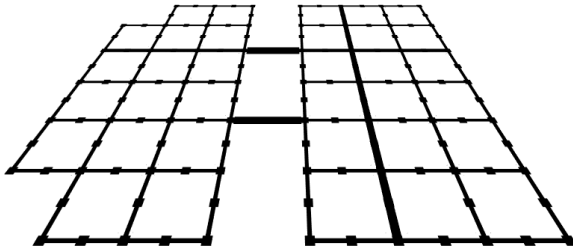
From a library of assorted 3d Models to assembled city blocks



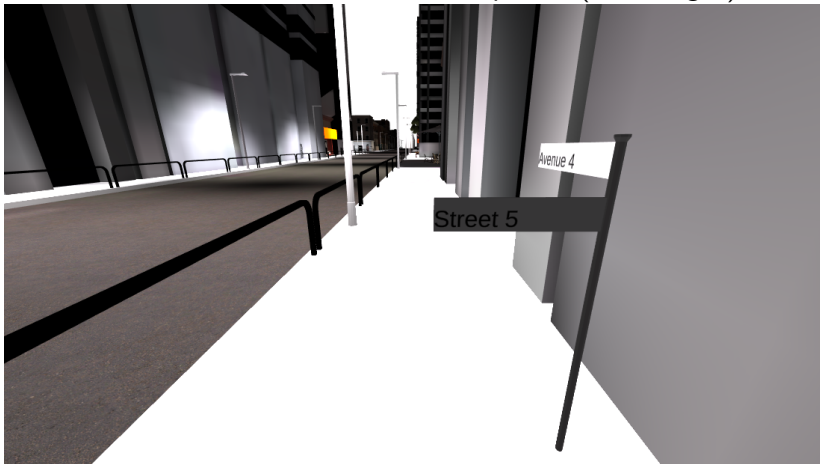
Placed to form a city



Road layout is inferred and a Road Graph is produced



Roads are named and details are placed (street signs)



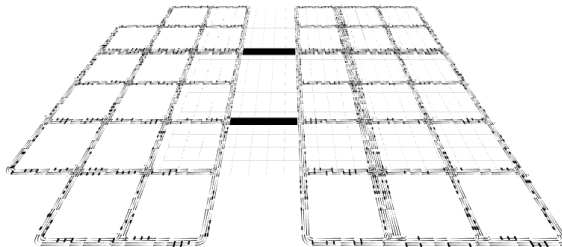
Traffic Simulation

Traffic Simulation

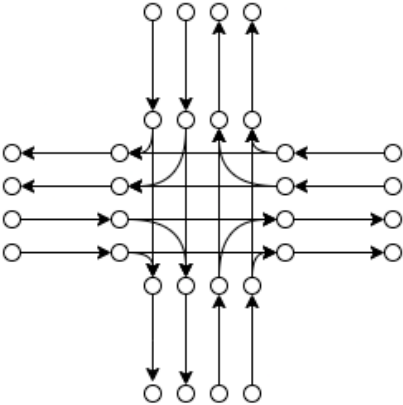
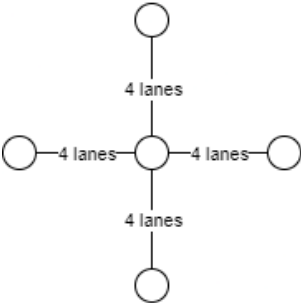
Deterministic agent based simulation.

Each car is simulated as a train following the Rail Graph

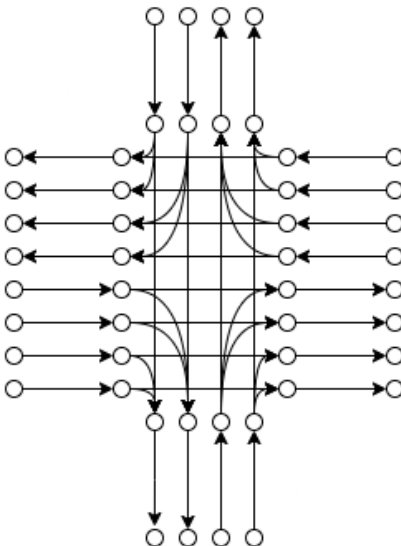
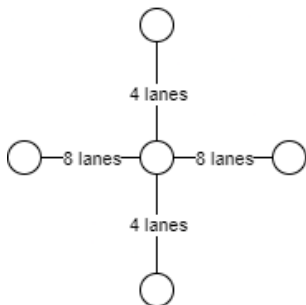
Generation of the Rail Graph from the Road Graph



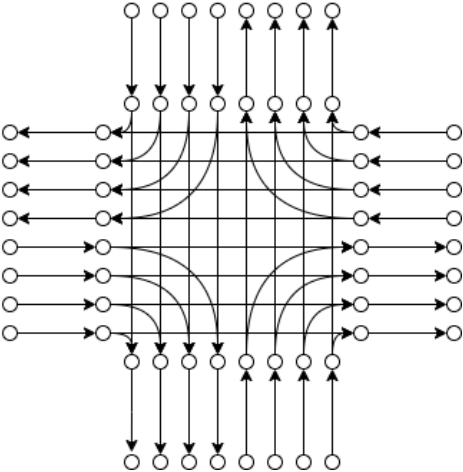
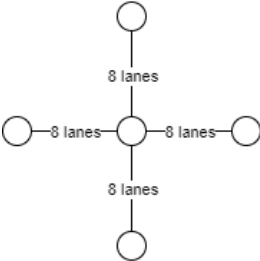
4 way intersection



4 to 8 way intersection



8 way intersection



Traffic lights placement



Car:

- ▶ edge (Rail Graph)
- ▶ relative position
- ▶ velocity
- ▶ seed (acceleration, turn direction)
- ▶ absolute position calculated from edge and relative position
- ▶ direction (angle) calculated from edge and relative position

Integration:

- ▶ The simulation is integrated every 100ms
- ▶ It moves the cars three times and store the positions
- ▶ If a car is intersecting one of the other positions its not moved; otherwise, it is

Minimizing allocations and access:

- ▶ Mutable hashtables to get close to constant lookup time
- ▶ Indexes to lookup on graphs in constant time
- ▶ Precalculation of graph node stars and future car positions

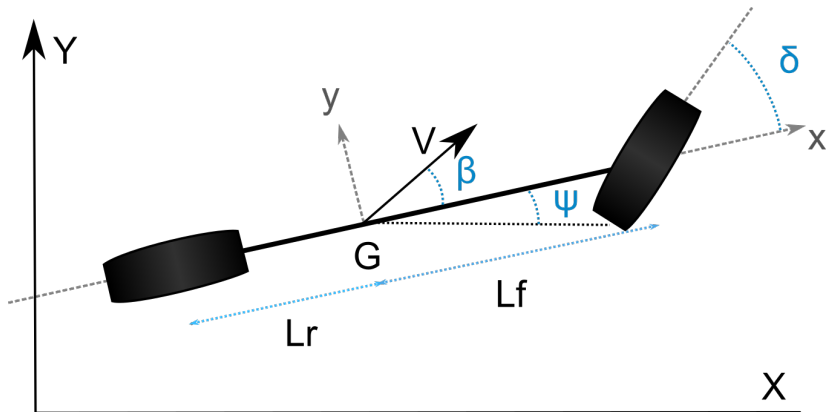
The collision detection is the simulation bottleneck. The bruteforcing approach takes n^2 checks, with n being the number of cars.

Reducing the cars neighbors:

- ▶ Grid indexing
- ▶ Stopped car linking

The collision detection and graph navigation is parallelized.

Kinematic Bicycle Model



UDP Infrastructure

UDP with reliability:

- ▶ optional retransmission
- ▶ message fragmentation
- ▶ integrity check

Synchronization

- ▶ State synchronization
- ▶ Stability
- ▶ Interpolation (Traffic simulation)
- ▶ Extrapolation (Player characters)

Audio

- ▶ Half duplex radio over UDP
- ▶ No loopback
- ▶ Resampling
- ▶ White noise
- ▶ Hi-pass filter
- ▶ Traffic sound effect
- ▶ Bicycle sound effects

Testing

- ▶ Unit testing: TDD (Test Driven Development)
- ▶ Integration testing: Multiplayer testing