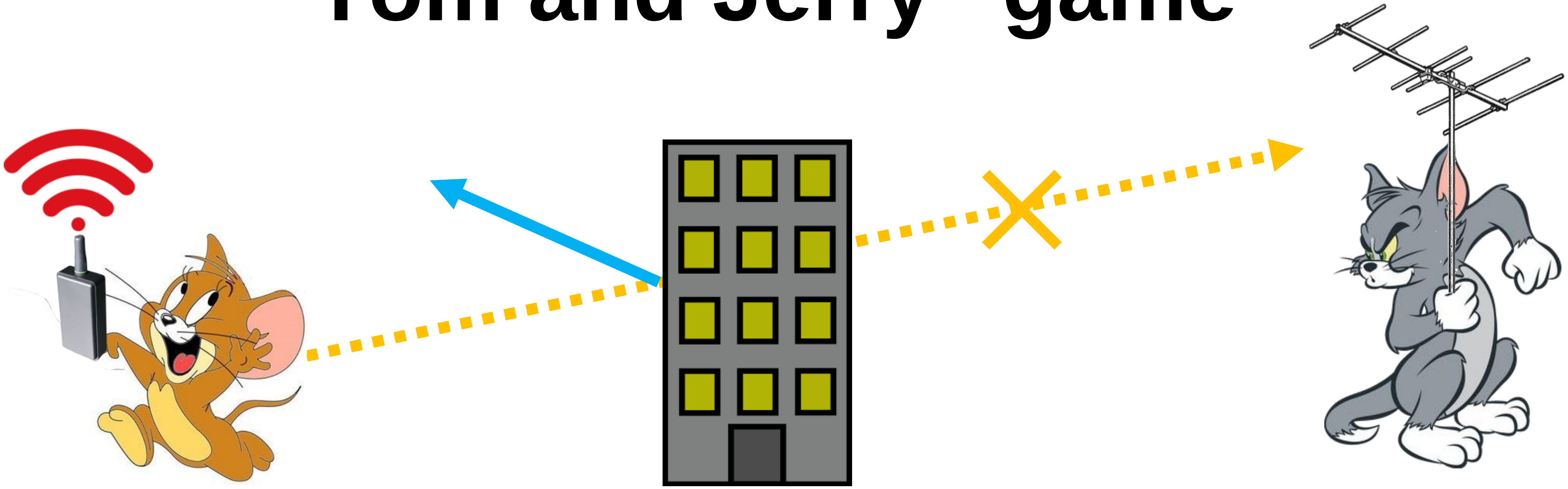
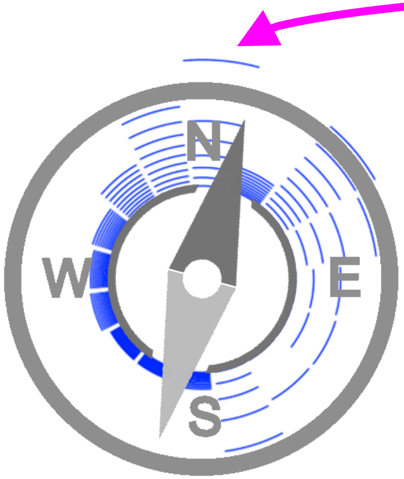
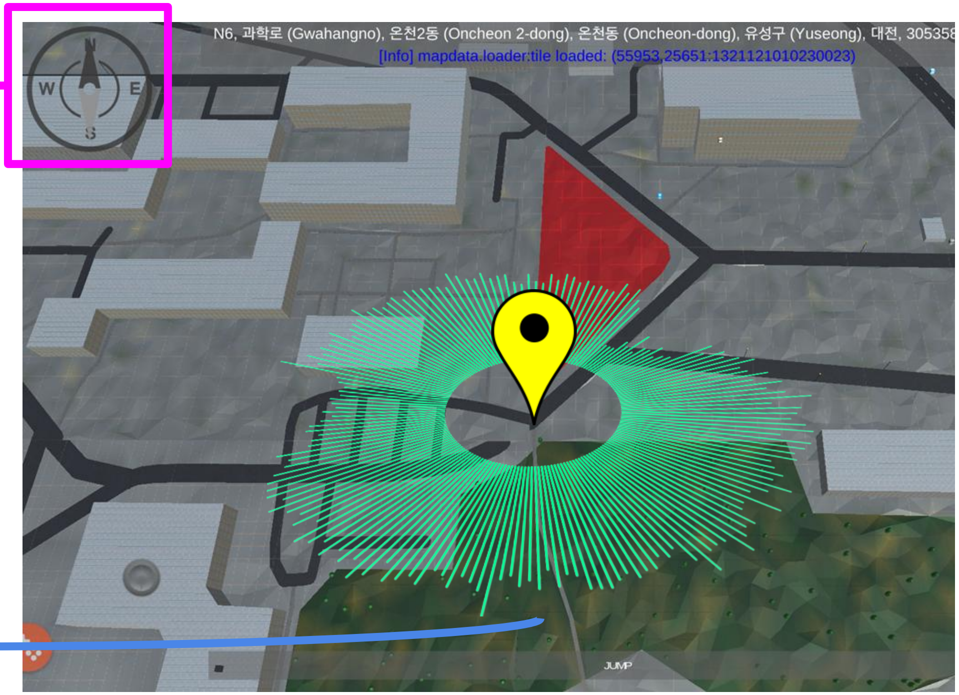
“Tom and Jerry” game





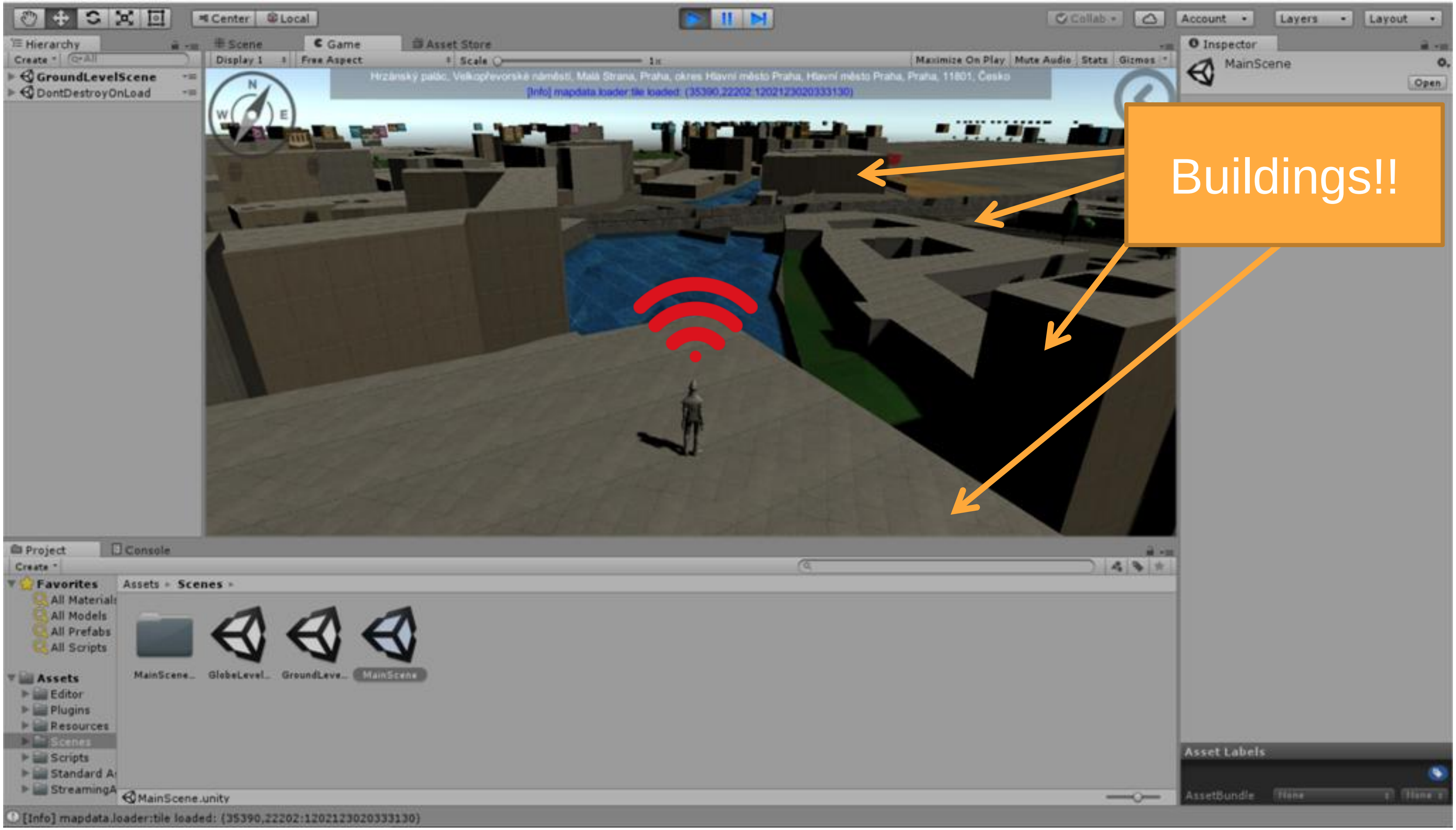

N6, 과학로 (Gwahangno), 온천2동 (Oncheon 2-dong), 온천동 (Oncheon-dong), 유성구 (Yuseong), 대전, 305358

[Info] mapdata.loader:tile loaded: (55953,25651:1321121010230023)

Signal Graphs

Problem Statement

How can we simulate the signal detection?

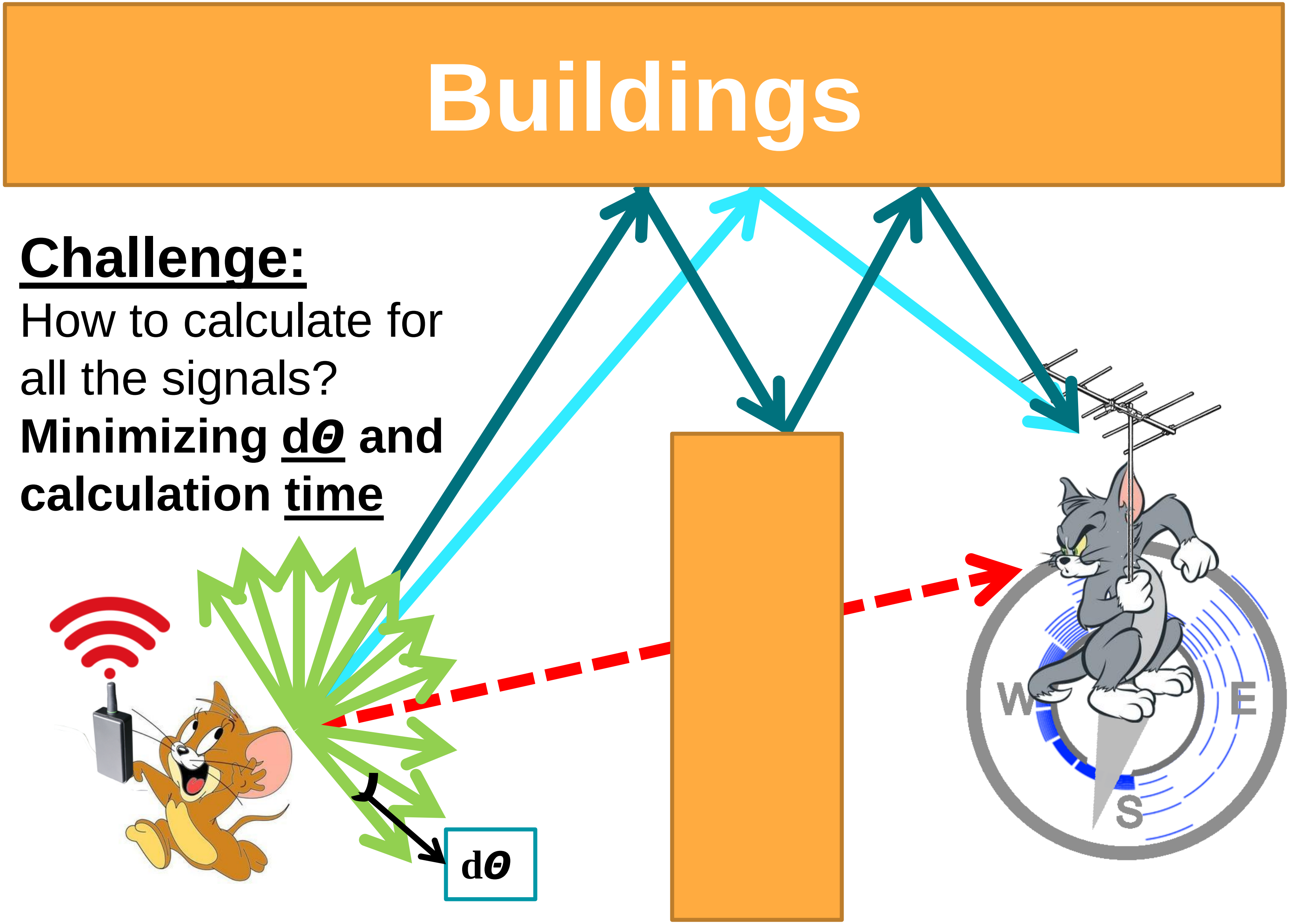


Hrzánský palác, Velkopřevorské náměstí, Malá Strana, Praha, okres Hlavní město Praha, Hlavní město Praha, Praha, 11801, Česko

[Info] mapdata.loader:tile loaded: (35390,22202:1202123020333130)

[Info] mapdata.loader:tile loaded: (35390,22202:1202123020333130)

Buildings!!



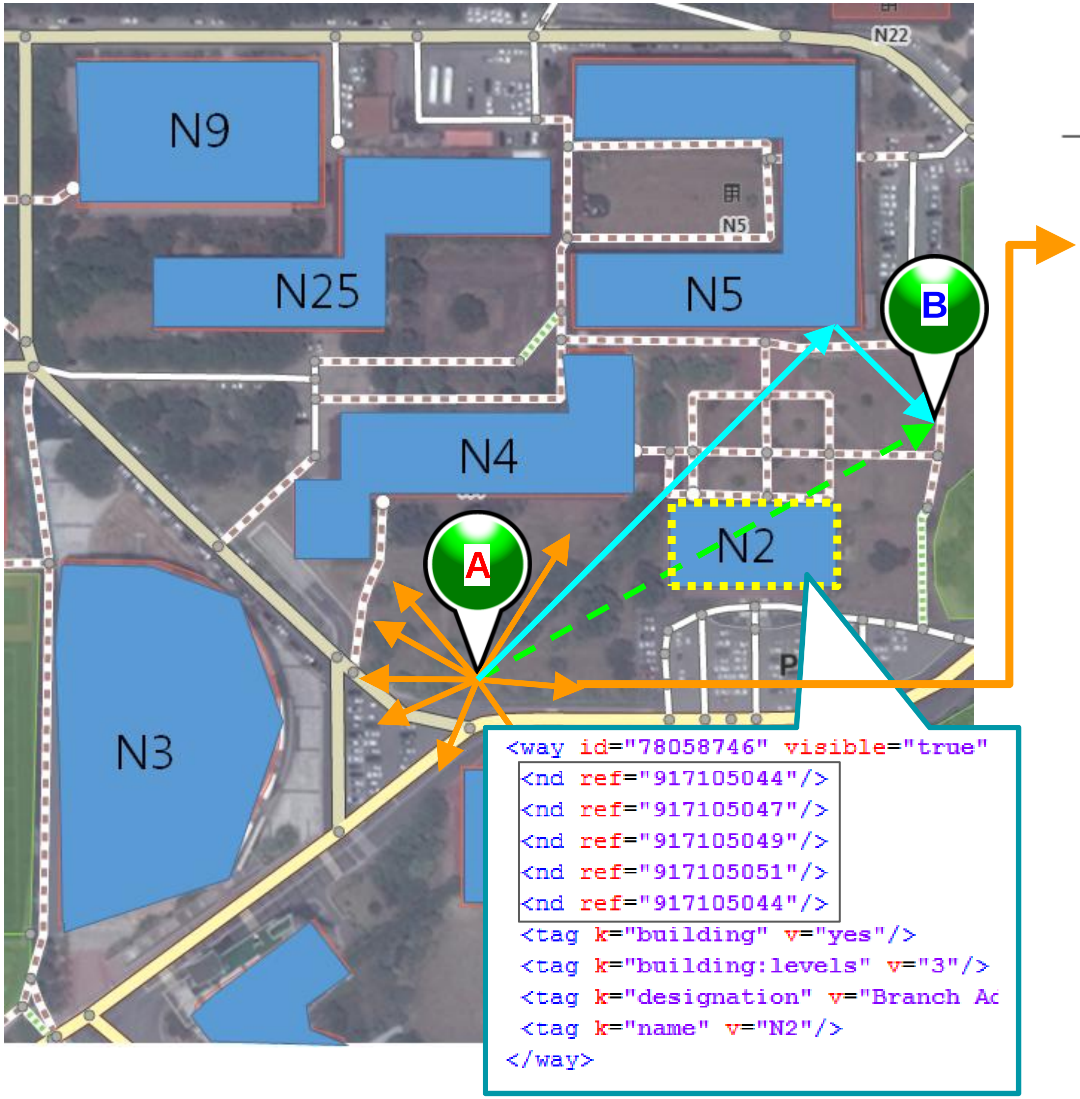
Challenge:

How to calculate for all the signals?

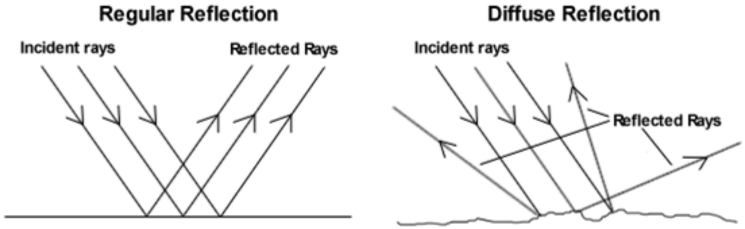
Minimizing dθ and calculation time

- **Distance (Sig ∝ 1/d)**
- **Blocked, Reflection**

Buildings =“Polygons”



```
<way id="78058746" visible="true"
  <nd ref="917105044"/>
  <nd ref="917105047"/>
  <nd ref="917105049"/>
  <nd ref="917105051"/>
  <nd ref="917105044"/>
  <tag k="building" v="yes"/>
  <tag k="building:levels" v="3"/>
  <tag k="designation" v="Branch Ac
  <tag k="name" v="N2"/>
</way>
```

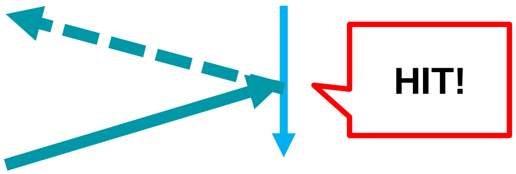


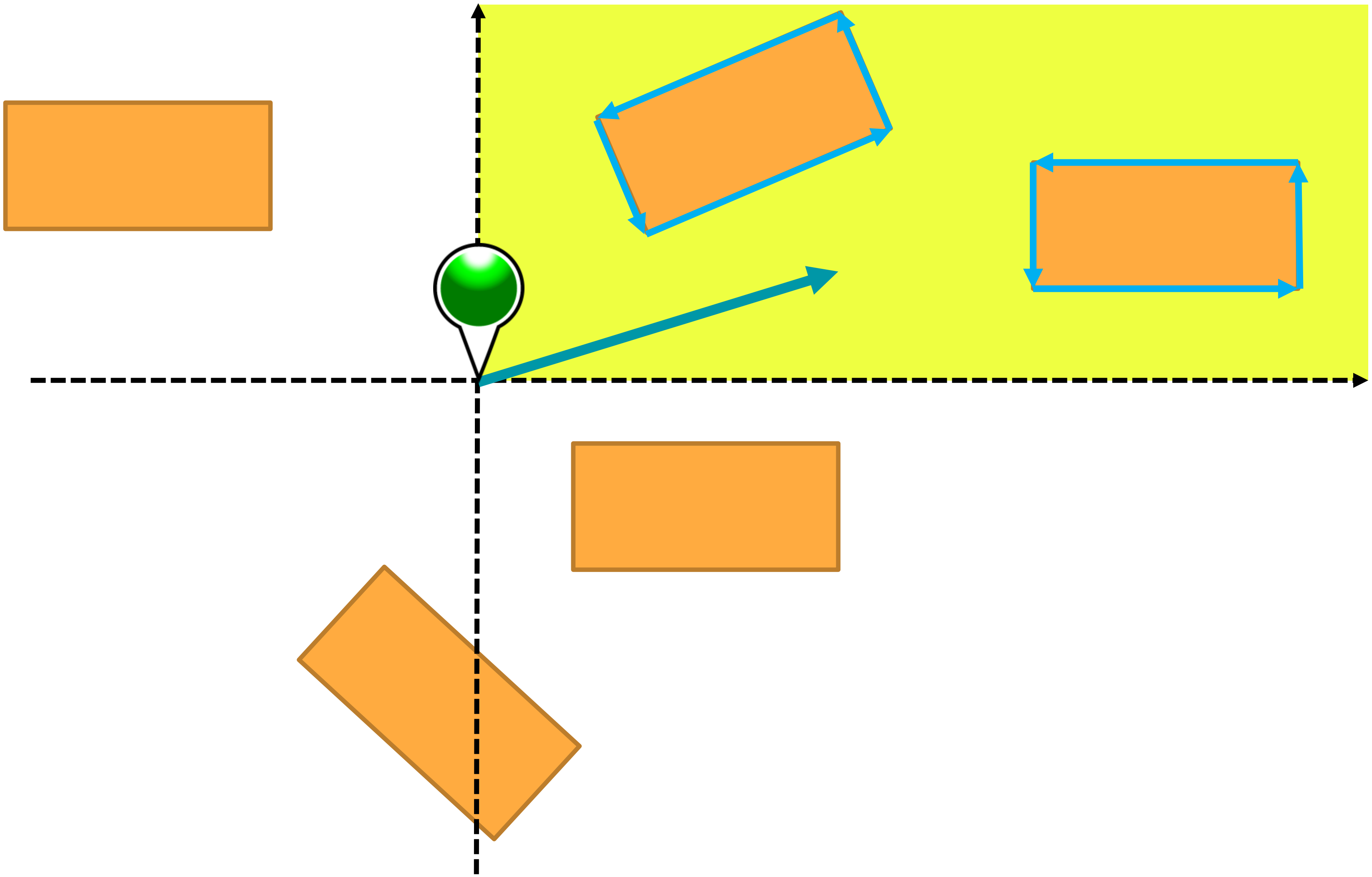
1) **blocked X, but reflected O**
2) **Calculate <u>all the direction</u>**
   - **DISCRETE MATH**

Algorithm Draft:

```
for all signal Si from A:
    if reachable(Si, B):
        s = 1/distance(Si, B)
        Θ = angle(Si, B)
        set(B, s, Θ)
(*s=strength)
```

Reflected Building Search Algorithm





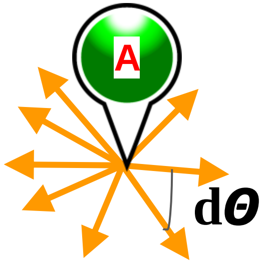
Enhanced Algorithm (Parallel Computing)

```
Q ← signals from A by $d\theta$
while not empty(Q):
    Si ← Dequeue(Q)
    if strength(Si) < Threshold:
        continue;
    if reachable(Si, B):
        s = strength(Si, B)
        $\theta$ = angle(Si, B)
        set(B, s, $\theta$)
        continue;
    if reflected(Si, Buildings):
        Enqueue(Q, reflected Signals)
         (include diffused reflection)
```

Too weak

Detected

Reflected

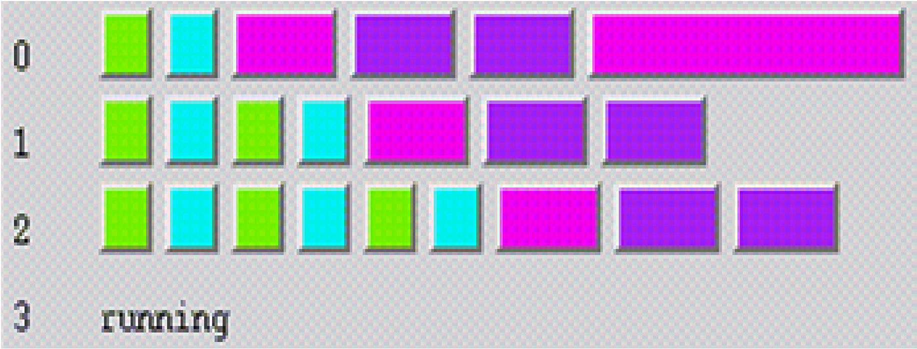


Realtime → <u>Time Limit (시간제약성)</u>

Advantages of Queue: Parallelism

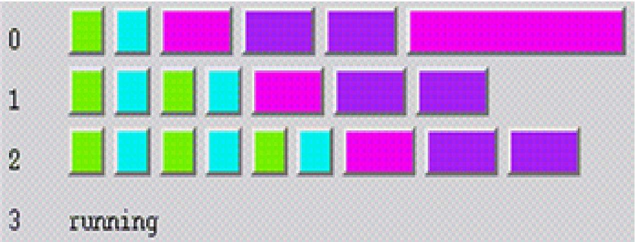
- **<u>Map/Reduce</u>**
- Runs **<u>multiple cores</u>** at the same time.
- **<u>Minimize dθ</u>** and **<u>Delay</u>** for **realism.**

<u>Master(3), Slave (0, 1, 2)</u>



Master / Slave (Map / Reduce)

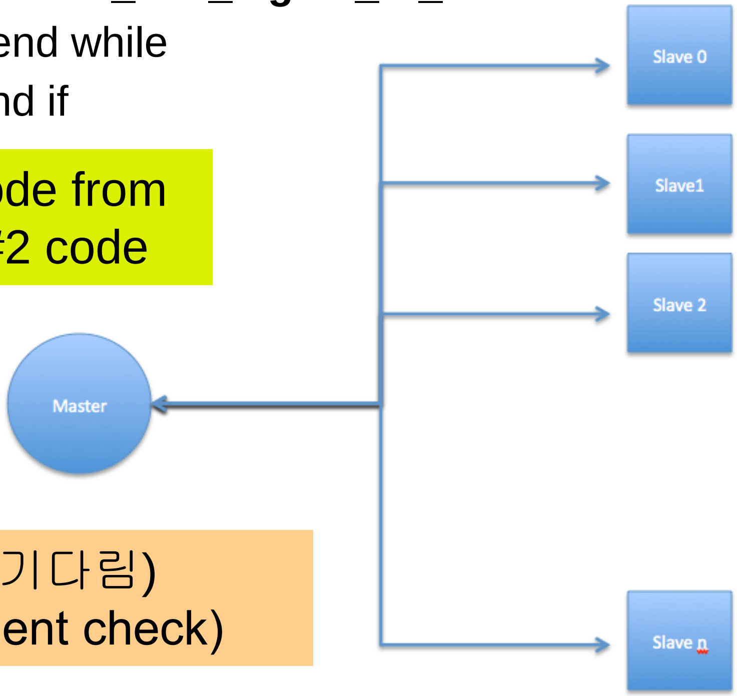
Not a definite answer, we should improve this



```
If master then
  for jobs begin
    Generate_Signals
    while search_for_idle_slave begin
      if an_idle_slave found then
        send_signal_data_to_an_idle_slave
      else if slave_wants_to_report_result then
        get_report_result_from_the_slave
      end if
    end while
  end for
  while result_report begin
    get_result_from_a_slave
  end while
  send_terminaling_signal_to_slaves
```

```
else (slave)
  send_idle_signal_to_master
  while job_request_from_master begin
    get_data_from_master
    calculate_signal_using_data
    send_result_to_master
    send_idle_signal_to_master
  end while
end if
```

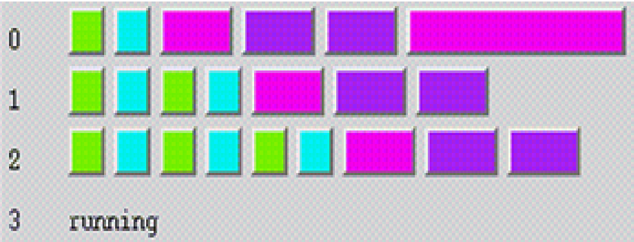
Reference: Modified code from KISTI KSC 2014 Prob#2 code



MPI Blocking Communication (신호가 올 때 까지 기다림)
MPI_Send(), MPI_Recv() … (Wait until the opponent check)

Master / Slave (Map / Reduce)

Not a definite answer, we should improve this



Who is IDLE!?

Aye, SIR!

I give you an order.

Wait for a minute, SIR! (MIA)

While Job Exists

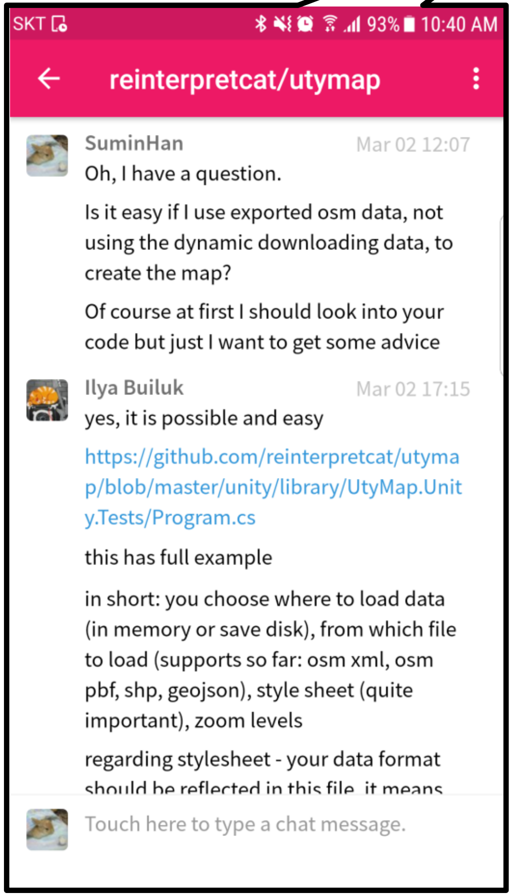
Master (rank 0)

Slave (rank n != 0)

unity

Ilya Builuk
reinterpretcat

GUI is lower priority, but still important for **realism**



SuminHan Mar 02 12:07
Oh, I have a question.

Is it easy if I use exported osm data, not using the dynamic downloading data, to create the map?

Of course at first I should look into your code but just I want to get some advice

Ilya Builuk Mar 02 17:15
yes, it is possible and easy

https://github.com/reinterpretcat/utymap/blob/master/unity/library/UtyMap.Unity.Tests/Program.cs

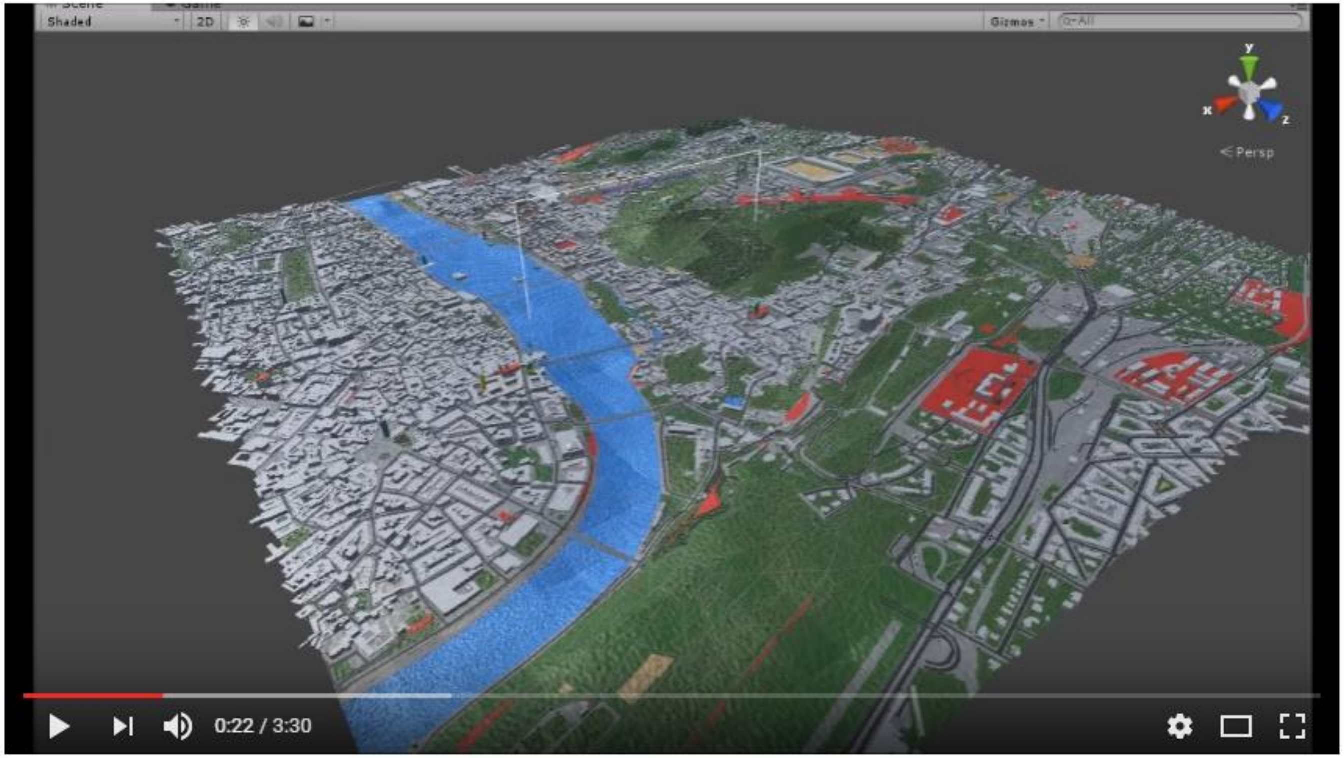
this has full example

in short: you choose where to load data (in memory or save disk), from which file to load (supports so far: osm xml, osm pbf, shp, geojson), style sheet (quite important), zoom levels

regarding stylesheet - your data format should be reflected in this file, it means

Touch here to type a chat message.

☞ Gitter chat with the author



↑ Graphical Simulation of Prague (https://youtu.be/mZzOWsoM5EY)

Client (Android)

GPS, Magnetic Field(direction)

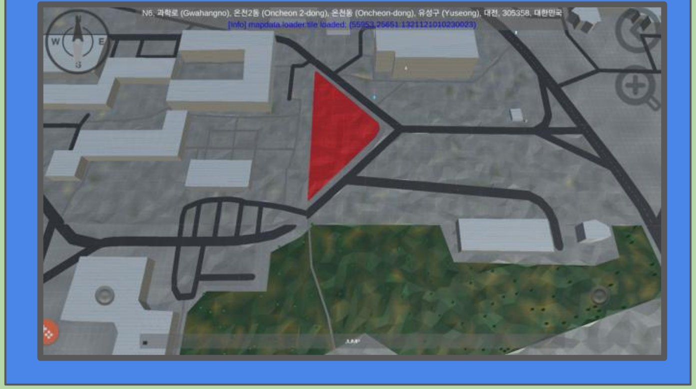
OSM.XML Map data

New Signal
Data from
Server

Messaging

Unity

GUI

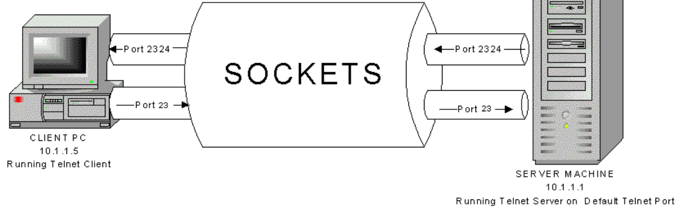

N6, 과학로 (Gwahangno), 온천2동 (Oncheon 2-dong), 온천동 (Oncheon-dong), 유성구 (Yuseong), 대전, 305358, 대한민국
[info] mapdata.loader.tile loaded: (55953,25651:1321121010230023)

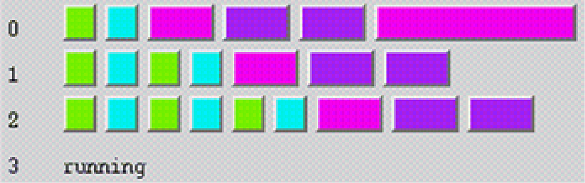
Server (Multi-core)

Login Info
Current GPS position

Signal Calculation
(Parallelism)
Synchronization

TCP: bidirectional





MPI

OpenCL

Plan

1) **Implement signal simulation**

2) Run the server computer with multiple cores (Parallel computing)

   ❖ **Minimize dθ and Delay(Time) for realism.**

3) Connect with Android by TCP communication

4) Draw the signal graph in Unity

5) Improve GUI

6) Invent more rules

Biggest Challenge