VIM-Android Version

Go to Github

What is VIM

VIM(Voice Indoor Maps) Application for Android has been developed for visually impaired persons to support indoor navigation.

Actually two versions of VIM were developped for

1. indoor space with braille blocks
2. indoor space without braille blocks

VIM has been developed for visually impaired, but it will also be useful for non-impaired users in finding indoor routes.

Demonstration

location	departure	destination	video
Ease Daegu Station	Information Center	RestRoom Near By Number 2 Entrance	YouTube
Ease Daegu Station	RestRoom Near By Number 2 Entrance	Number 6 Entrance	YouTube

Form for expressing indoor space information

VIM follows the OGC international standard IndoorGML format to express indoor spatial information. if you want to add or modify indoor space information, you must follow the schema format of the IndoorGML 1.0 core version.

> What is IndoorGML?

> indoorgmlcore.xsd

IndoorGML For VIM

In order to run VIM smoothly, a total of 4 `<core:SpaceLayer/>`s which has attribute `gml:id` as below table must exist in IndoorGML.

gml:id	require	description
base	optional	This layer expresses only general indoor spatial information like a room, door, stair, elevator...
network	essential	This layer expresses route which user can use. It only consists of state and transition.

gml:id	require	description
landmark	optional	This layer has poi which you want to recognize for user. It only consists of state.
safety	optional	This layer has dangerous or relative safety poi when user face it. It only consists of state.

In order to express VIM object where exist in network layer you need to add value at IndoorGML element.

layer	VIM object	indoorGML	element	value	description
base	stair	`<core:CellSpace/>`	`<name/>`	contain 'stair'	It represent the space of interLayerConnected state which has description value as `type="stair"` and exist in network layer
base	elevator	`<core:CellSpace/>`	`<name/>`	contain 'elevator'	It represent the space of interLayerConnected state which has description value as `type="elevator"` and exist in network layer
base	escalator	`<core:CellSpace/>`	`<name/>`	contain 'escalator'	It represent the space of interLayerConnected state which has description value as `type="escalator"` and exist in network layer
network	stair	`<core:State/>`	`<description/>`	type="stair"	It must be connected to other floor state as transition and
network	elevator	`<core:State/>`	`<description/>`	type="elevator"	It must be connected to other floor state as transition

layer	VIM object	indoorGML	element	value	description
network	escalator	`<core:State/>`	`<description/>`	type="escalator"	It must be connected to other floor state as transition
network	dotted block	`<core:State/>`	`<description/>`	dot="true"	It will work when turn on the visually impaired version
network	linear block	`<core:State/>`	`<description/>`	null	It will work when turn on the visually impaired version

More information

In order to make the space that can move floor, you need to make cellSpace in base layer and state which has been interconnected as contains based on base layer in network layer.

Location.json

Using the location.json you can modify or add destination information. basically, destinations are `<core:State/>` which exist in network layer. the format of location.json is as bellow.

```
{
  "service_no" : {
    "209" : { // the number which is positioning system service number
      "name": "pnu_abc.gml", // indoorGML file name
      "floors": {
        "2": { // floor number
          "states": {
            "S71": { // <core:State/> gml:id in network layer
              "name": {
                "en": "2F north man's restroom",
                "ko": "2층 북쪽 남자 화장실"
              }
            },
            ...
            ...
          }
        },
        ...
        ...
      }
    },
    "braille_blocks": false // if the place has been include braille blocks for
  visually impaired you can put true value
  }
}
```

Tree

```
└app
  └src
     ├main
     ├assets
     │  └indoorgml
     ├java
     │  └lab
     │     └stem
     │        └vim
     │           ├core
     │           ├guideFactory
     │           ├indoorGML
     │           ├log
     │           └observer
     ├raw
     └res
```