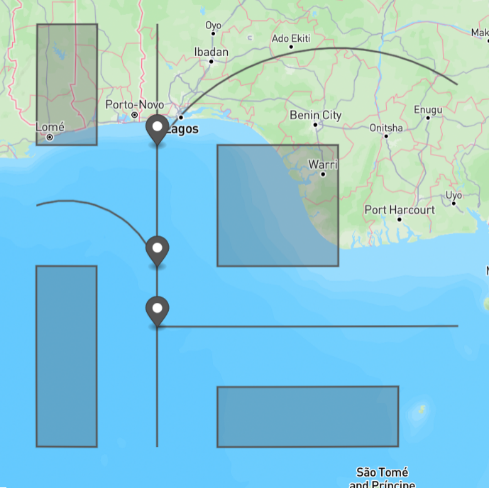
OPIS TESTNIH PRIMEROV

# PRVI PRIMER

Prvi primer prikazuje mesto s štirimi stavbami in štirimi cestami. Stavbe so razporejene v dve skupini, levo in desno, medtem ko ceste vključujejo vertikalne in horizontalne črte ter dva ukrivljena ovinka. Ta primer preverja osnovne elemente, kot so box, line in bend.

city MyCity {

building A {

box ( 1.0 , 1.0 , 2.0 , 4.0 )

} ;

building B {

box ( 1.0 , 6.0 , 2.0 , 8.0 )

} ;

building C {

box ( 4.0 , 4.0 , 6.0 , 6.0 )

} ;

building D {

box ( 4.0 , 1.0 , 7.0 , 2.0 )

} ;

road a {

line ( 3.0 , 1.0 , 3.0 , 8.0 )

} ;

road b {

bend ( 3.0 , 4.0 , 1.0 , 5.0 , 90.0 )

} ;

road c {

line ( 3.0 , 3.0 , 8.0 , 3.0 )

} ;

road d {

bend ( 8.0 , 7.0 , 3.0 , 6.0 , 90.0 )

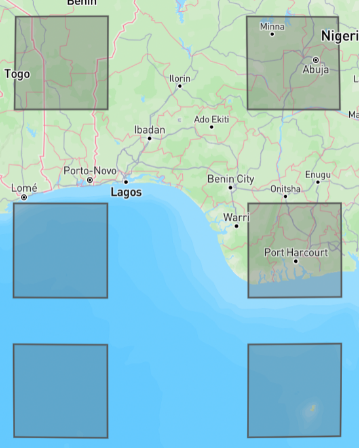
} ;

}

# DRUGI PRIMER

Drugi primer vsebuje šest stavb, razporejenih simetrično v dveh stolpcih, vendar brez cest ali drugih objektov. Namenjen je preverjanju pravilne razporeditve več stavb brez dodatnih motenj.

city MyCity {

 building A {

box ( 1.0 , 1.0 , 3.0 , 3.0 )

} ;

building B {

box ( 1.0 , 4.0 , 3.0 , 6.0 )

} ;

building C {

box ( 1.0 , 8.0 , 3.0 , 10.0 )

} ;

building D {

box ( 6.0 , 1.0 , 8.0 , 3.0 )

} ;

building E {

box ( 6.0 , 4.0 , 8.0 , 6.0 )

} ;

building F {

box ( 6.0 , 8.0 , 8.0 , 10.0 )

} ;

}

# TRETJI PRIMER

Tretji primer je razširjena različica drugega, z dodatkom ene vertikalne ceste, ki povezuje vse stavbe. S tem se preverja, kako se line cesta vključi v že obstoječo razporeditev stavb.

city MyCity {

building A {

box ( 1.0 , 1.0 , 3.0 , 3.0 )

} ;

building B {

box ( 1.0 , 4.0 , 3.0 , 6.0 )

} ;

building C {

box ( 1.0 , 8.0 , 3.0 , 10.0 )

} ;

building D {

box ( 6.0 , 1.0 , 8.0 , 3.0 )

} ;

building E {

box ( 6.0 , 4.0 , 8.0 , 6.0 )

} ;

building F {

box ( 6.0 , 8.0 , 8.0 , 10.0 )

} ;

road a {

line ( 4.5 , 1.0 , 4.5 , 10.0 )

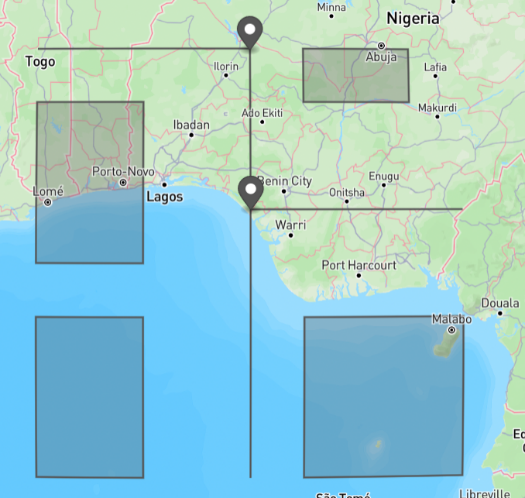
} ;

}

# ČETRTI PRIMER

Četrti primer prikazuje mesto s uporabo spremenljivke x, štirimi stavbami in tremi cestami. Stavbe so razporejene levo in desno, ceste pa povezujejo zgornje, spodnje in srednje dele mesta. Namenjen je testiranju spremenljivk in kombinacij vodoravnih ter navpičnih cest.

let x = 5.0 ;



city MyCity {

building A {

box ( 1.0 , x , 3.0 , 8.0 )

} ;

building B {

box ( 1.0 , 1.0 , 3.0 , 4.0 )

} ;

building C {

box ( 6.0 , 1.0 , 9.0 , 4.0 )

} ;

building D {

box ( 6.0 , 8.0 , 8.0 , 9.0 )

} ;

road a {

line ( 5.0 , 9.0 , 5.0 , 1.0 )

} ;

road b {

line ( 5.0 , 9.0 , 1.0 , 9.0 )

} ;

road c {

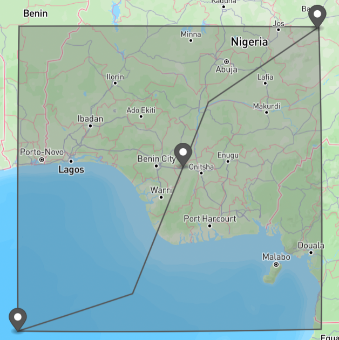
line ( 5.0 , 6.0 , 9.0 , 6.0 )

} ;

}

# PETI PRIMER

Peti primer prikazuje mesto s parkom in eno cesto, ki ima več točk preloma. Uporabljeni sta dve spremenljivki – ena za številsko vrednost in druga za par koordinat. Namen primera je preverjanje delovanja ukaza polyline ter pravilne uporabe spremenljivk v različnih strukturah.

let x = 5.0 ;

let y = new ( 10.0 , 10.0 ) ;

city MyCity {

park A {

box ( 2.0 , 2.0 , y , )

} ;

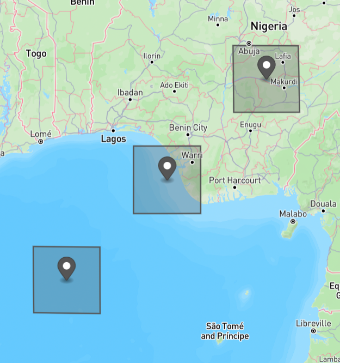
road a {

polyline ( 2.0 , 2.0 , 5.0 , 3.0 , 7.0 , 8.0 , y , )

} ; }

# ŠESTI PRIMER

Šesti primer predstavlja mesto z enim parkom, dvema stavbama in tremi označevalci. Vrednosti koordinat so delno podane s pomočjo spremenljivk in izrazov, kar omogoča preverjanje pravilnega računanja izrazov znotraj ukazov. Namen primera je testirati uporabo aritmetike v definiciji objektov ter vključevanje markerjev za označevanje točk v mestu.



let x = 3.0 ;

let y = new ( 10.0 , 10.0 ) ;

city MyCity {

park A {

box ( 1.0 , 1.0 , x , x )

} ;

building B {

box ( ( 2.0 + 2.0 ) , ( 1.0 \* 4.0 ) , 6.0 , 6.0 )

} ;

building C {

box ( 7.0 , 7.0 , 9.0 , 9.0 )

} ;

marker a {

mark ( 2.0 , 2.0 )

} ;

marker b {

mark ( 5.0 , 5.0 )

} ;

marker c {

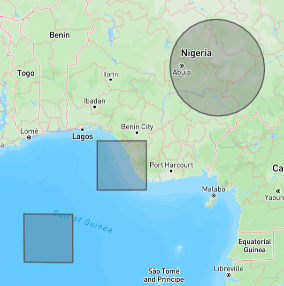
mark ( 8.0 , 8.0 )

} ;

}

# SEDMI PRIMER

Sedmi primer je podoben šestemu, vendar dodaja krožno stavbo (circ), s čimer se testira nova geometrijska oblika. Cilj je preveriti podporo za krožne objekte poleg pravokotnih.

let x = 3.0 ;

let y = new ( 10.0 , 10.0 ) ;

city MyCity {

park A {

box ( 1.0 , 1.0 , x , x )

} ;

building B {

box ( ( 2.0 + 2.0 ) , ( 1.0 \* 4.0 ) , 6.0 , 6.0 )

} ;

building C {

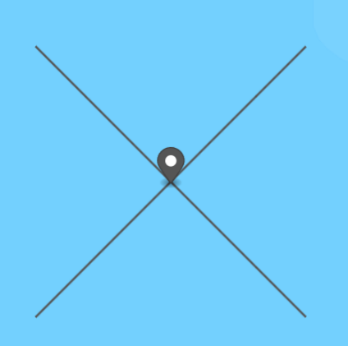
circ ( 9.0 , 9.0 , 2.0 )

} ;

}

# OSMI PRIMER

Osmi primer vsebuje izključno eno cesto road A, ki vsebuje dve liniji. S tem se testira možnost več line elementov znotraj ene ceste ter preverja pravilnost (validacija) takšne strukture.



let x = 3.0 ;

let y = new ( 10.0 , 10.0 ) ;

city MyCity {

road A {

line ( 2.0 , 2.0 , 3.0 , 3.0 )

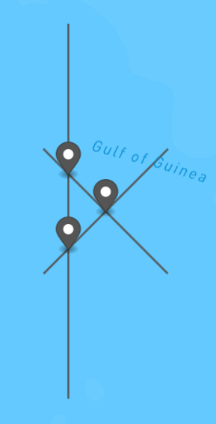
line ( 2.0 , 3.0 , 3.0 , 2.0 )

} ;

}

# DEVET PRIMER

Deveti primer uporablja spremenljivke in izraze za definiranje več poti. Vključuje seštevanje in druge izraze v koordinatah, s čimer se testira podpora za kompleksnejše izraze znotraj line ter validacija njihove pravilnosti.



let x = 2.0 ;

let y = new ( 3.0 , 3.0 ) ;

city MyCity {

road A {

line ( x , x , y , )

line ( x , ( x + 1 ) , 3.0 , x )

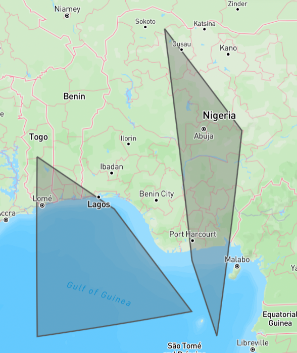
line ( 2.2 , 1.0 , 2.2 , 4.0 )

} ;

}

# DESETI PRIMER

Deseti primer vsebuje zgradbo z dvema box definicijama, vsaka z več kot štirimi koordinatami. S tem se preverja, ali parser zmore obdelati kompleksne, razširjene definicije boxov in več koordinat v istem objektu.



let x = 2.0 ;

let y = new ( 3.0 , 3.0 ) ;

city MyCity {

building A {

box ( 1.0 , 1.0 , 7.0 , 2.0 , 4.0 , 6.0 , 1.0 , 8.0 )

box ( 9.0 , 9.0 , 8.0 , 1.0 , 7.0 , 4.0 , 6.0 , 13.0 )

} ;

}