Collision Detection

EECS 367
Intro. to Autonomous Robotics

ROB 320
Robot Operating Systems

Winter 2022

autorob.org

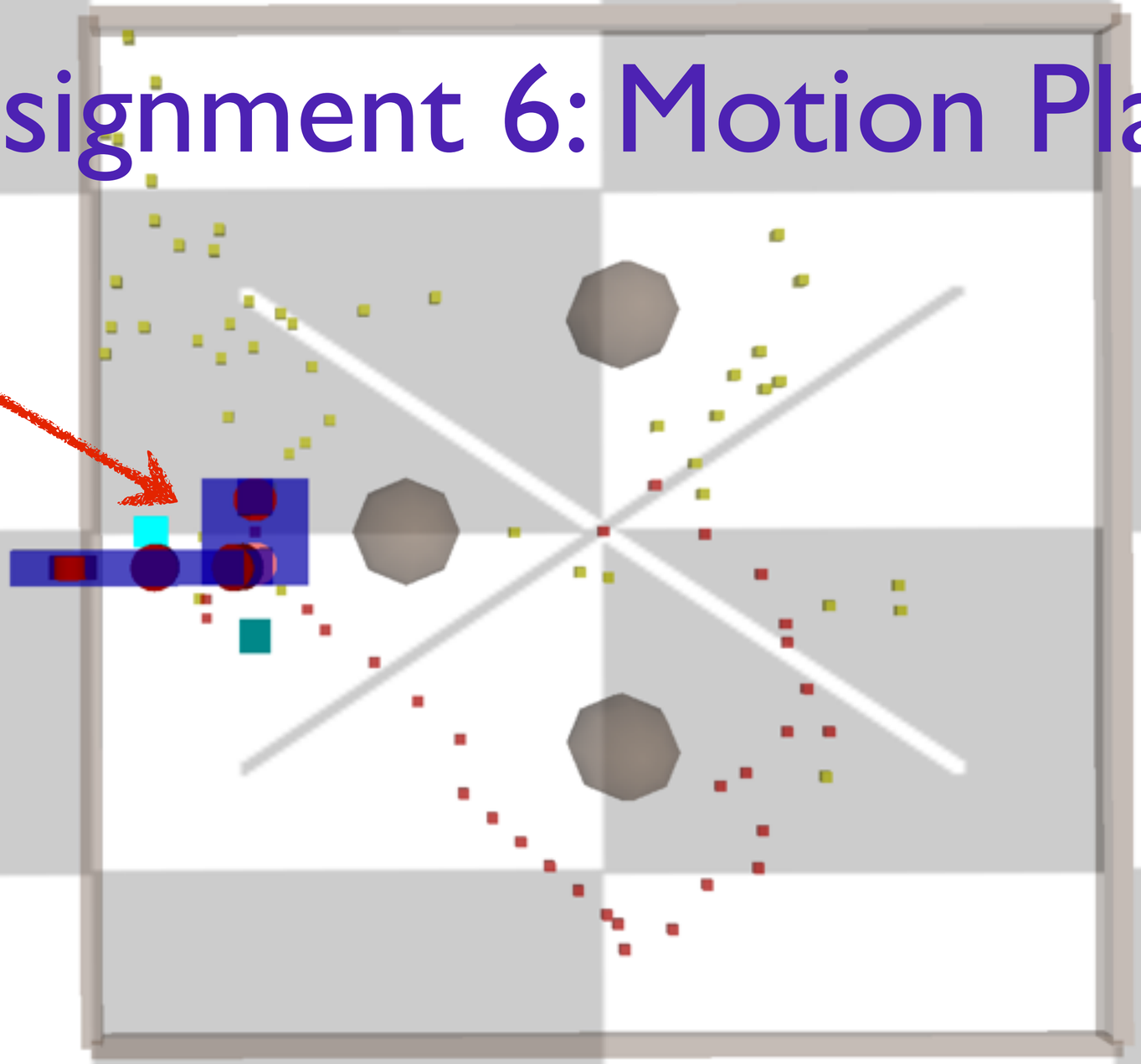
U.S. Department of Transportation
https://www.youtube.com/watch?v=B8Ct5hjs0Jw

Assignment 6: Motion Planning

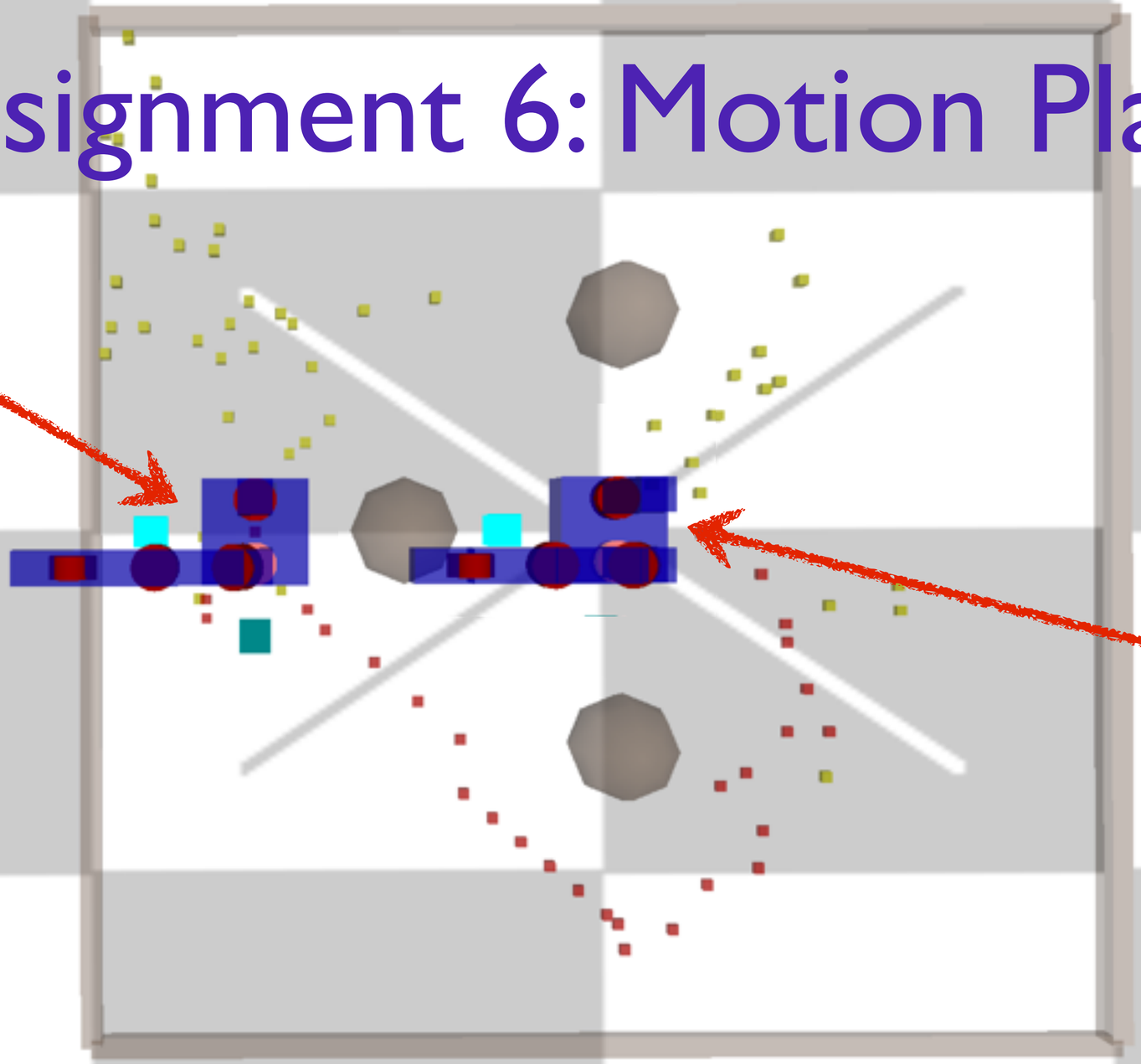
- Generate a collision free motion plan to the world origin and zero joint angle configuration

Assignment 6: Motion Planning

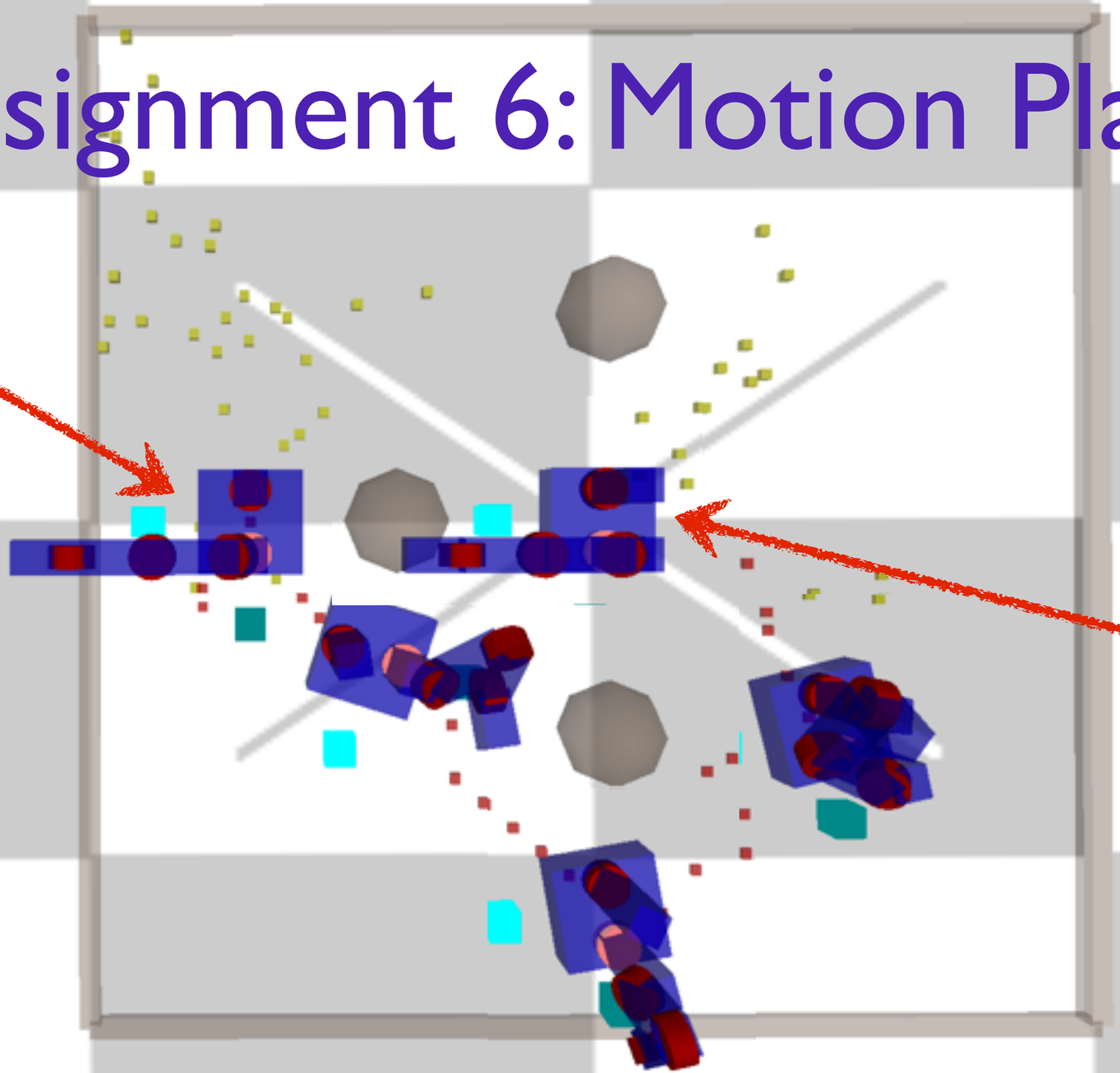
Start: random non-colliding configuration



Assignment 6: Motion Planning



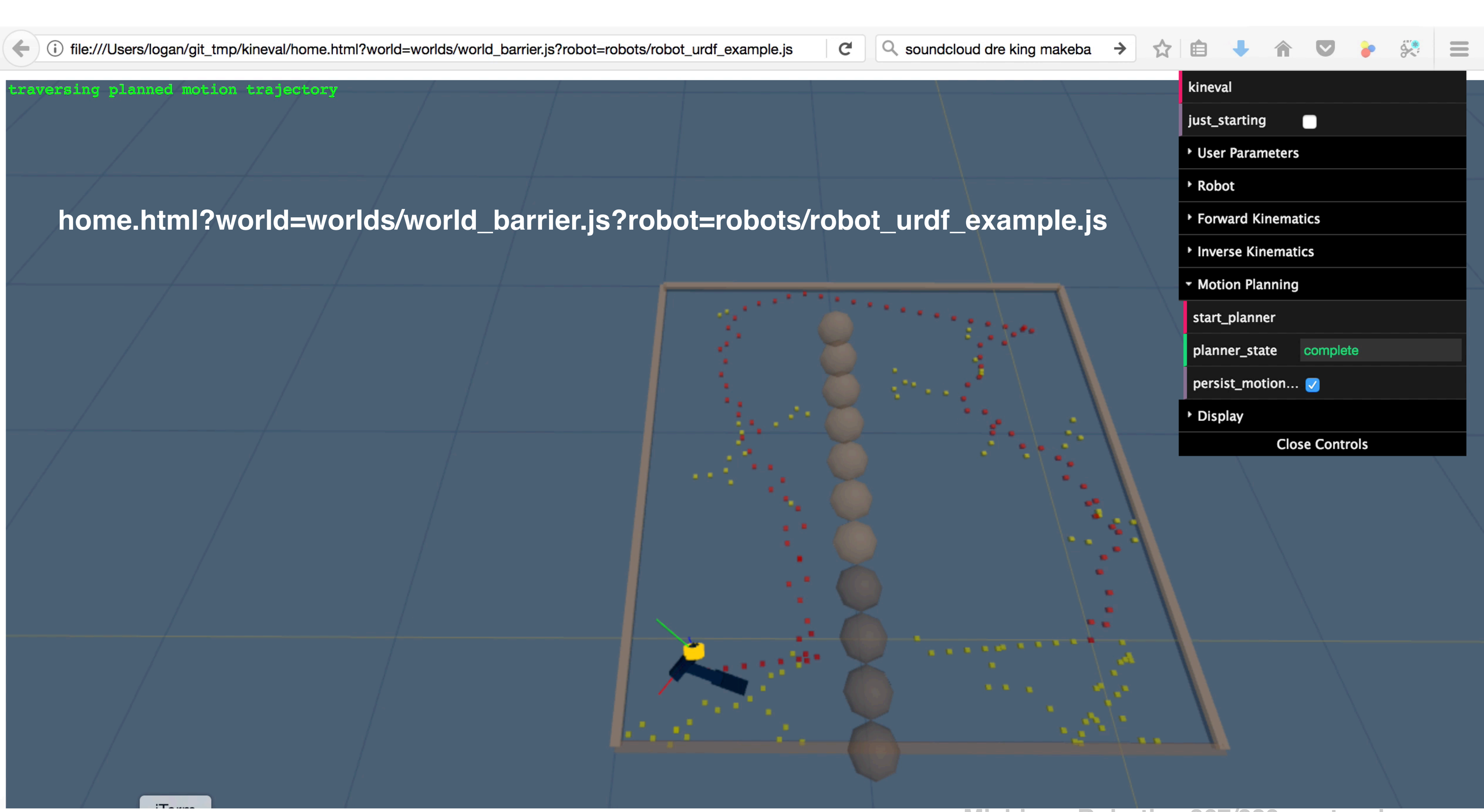
Assignment 6: Motion Planning



Start: random non-colliding configuration

Goal: zero configuration at world origin

Generate collision-free motion plan


file:///Users/logan/git_tmp/kineval/home.html?world=worlds/world_barrier.js?robot=robots/robot_urdf_example.js
soundcloud dre king makeba
traversing planned motion trajectory
home.html?world=worlds/world_barrier.js?robot=robots/robot_urdf_example.js
kineval
just_starting
User Parameters
Robot
Forward Kinematics
Inverse Kinematics
Motion Planning
start_planner
planner_state
complete
persist_motion...
Display
Close Controls


file:///Users/logan/git_tmp/kineval/home.html?world=worlds/world_barrier.js?robot=robots/robot_crawler.js
soundcloud dre king makeba
Welcome to KinEval. I want to see some text. Can you place a message here?
home.html?world=worlds/world_barrier.js?robot=robots/robot_crawler.js
kineval
just_starting
User Parameters
Robot
Forward Kinematics
Inverse Kinematics
Motion Planning
Display
Close Controls

GitHub, Inc. (US) https://github.com/ohseejay/kineval-stencil-fall16 rrt connect

ohseejay / kineval-stencil-fall16

Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs

Stencil code for KinEval (Kinematic Evaluato

2 commits

Branch: master New pull request

odestcj initial commit

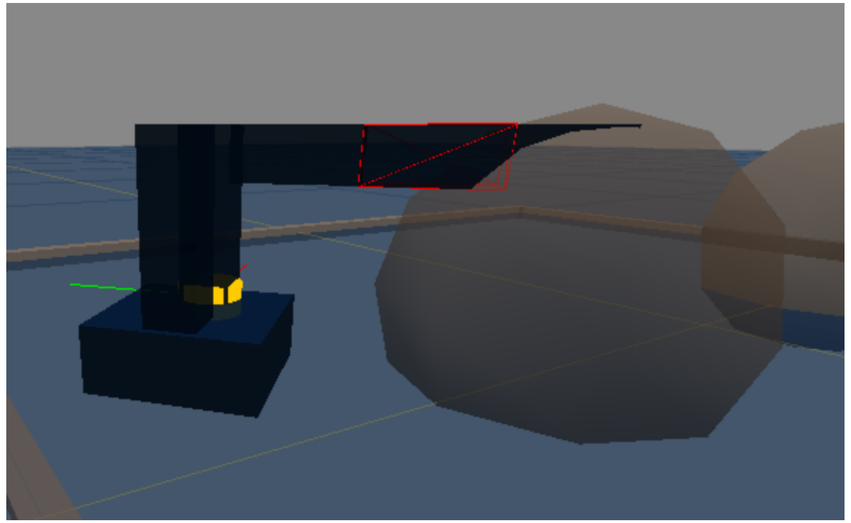
js		
kineval		
project_pathplan		
project_pendularm		
robots		
tutorial_heapsort		
tutorial_js	initial commit	26 days ago
worlds	initial commit	26 days ago
README.md	initial commit	26 days ago
home.html	initial commit	26 days ago

home.html

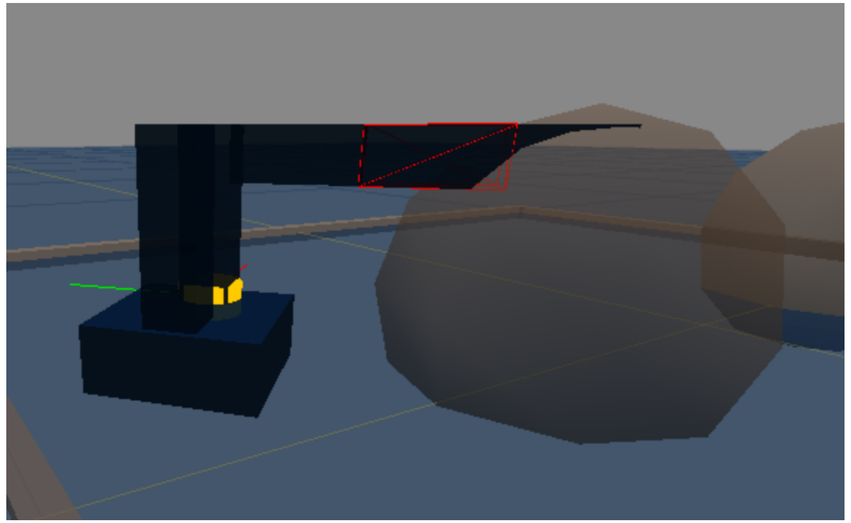
```
<script src="worlds/world_basic.js"></script>
...

function my_animate() {
    ...
    // detect robot collisions
    kineval.robotIsCollision();
    ...
    // if requested, perform configuration space
motion planning to home pose
    kineval.planMotionRRTConnect();
}
```

home.html

```
<script src="worlds/world_basic.js"></script>
...

function my_animate() {
    ...
    // detect robot collisions
    kineval.robotIsCollision();
    ...
    // if requested, perform configuration space
motion planning to home pose
    kineval.planMotionRRTConnect();
}
}
```



world file can be alternatively loaded by a script tag (avoid doing this)



home.html

```
<script src="worlds/world_basic.js"></script>
...

function my_animate() {
    ...
    // detect robot collisions
    kineval.robotIsCollision();
    ...
    // if requested, perform configuration space
motion planning to home pose
    kineval.planMotionRRTConnect();

}
```

detect if current configuration is in collision (colliding link turns red)

iterate motion planner

https://github.com/ohseejay/kineval-stencil/tree/master/kineval

Branch: **master** ▾ **kineval-stencil** / **kineval** / | New file | Upload files | Find file | History

odestcj initial commit — Latest commit 2a1bd6e on Jan 11

..		
kineval.js		
kineval_collision.js		
kineval_controls.js		
kineval_forward_kinematics.js		
kineval_inverse_kinematics.js	initial commit	2 months ago
kineval_matrix.js	initial commit	2 months ago
kineval_quaternion.js	initial commit	2 months ago
kineval_robot_init.js	initial commit	2 months ago
kineval_rosbridge.js	initial commit	2 months ago
kineval_rrt_connect.js	initial commit	2 months ago
kineval_servo_control.js	initial commit	2 months ago
kineval_startingpoint.js	initial commit	2 months ago
kineval_threejs.js	initial commit	2 months ago
kineval_userinput.js	initial commit	2 months ago

`kineval.robotIsCollision();`

Update collision detection with your forward kinematics

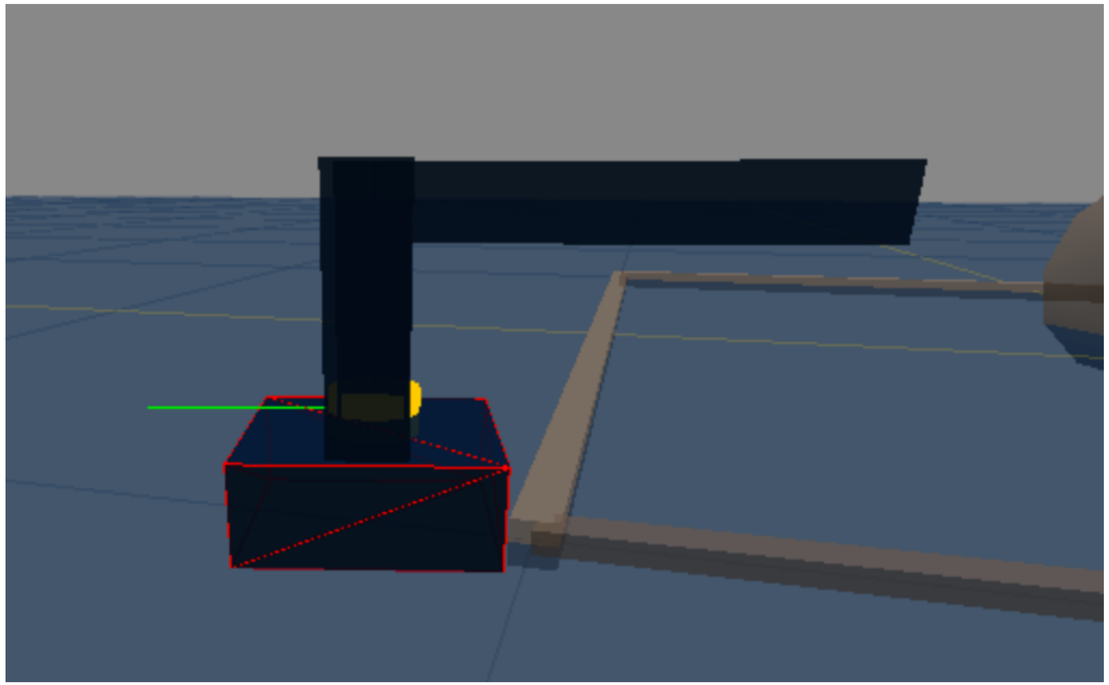
`kineval.planMotionRRTConnect();`

Implement RRT-Connect planner

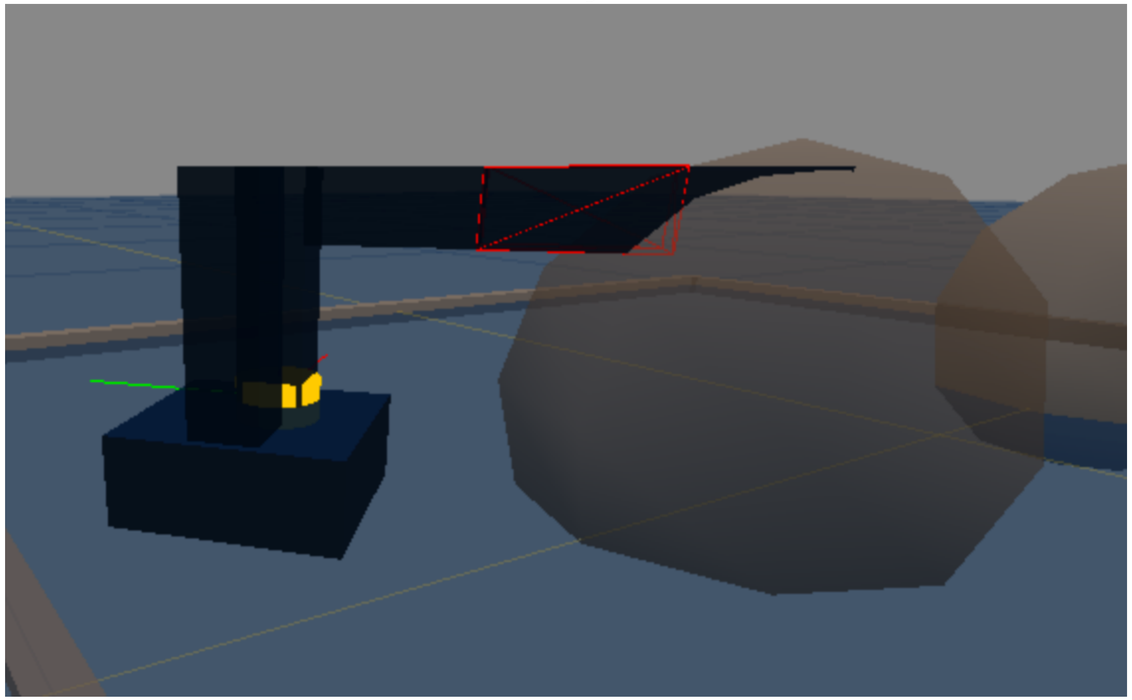
```
kineval.robotIsCollision();
```

Assignment 6 collision detection

Boundary Collision
(provided by default)



Link Collision
(requires your FK)



input: q (robot configuration)
output: false (for no collision) or name of link in collision

kineval_collision.js

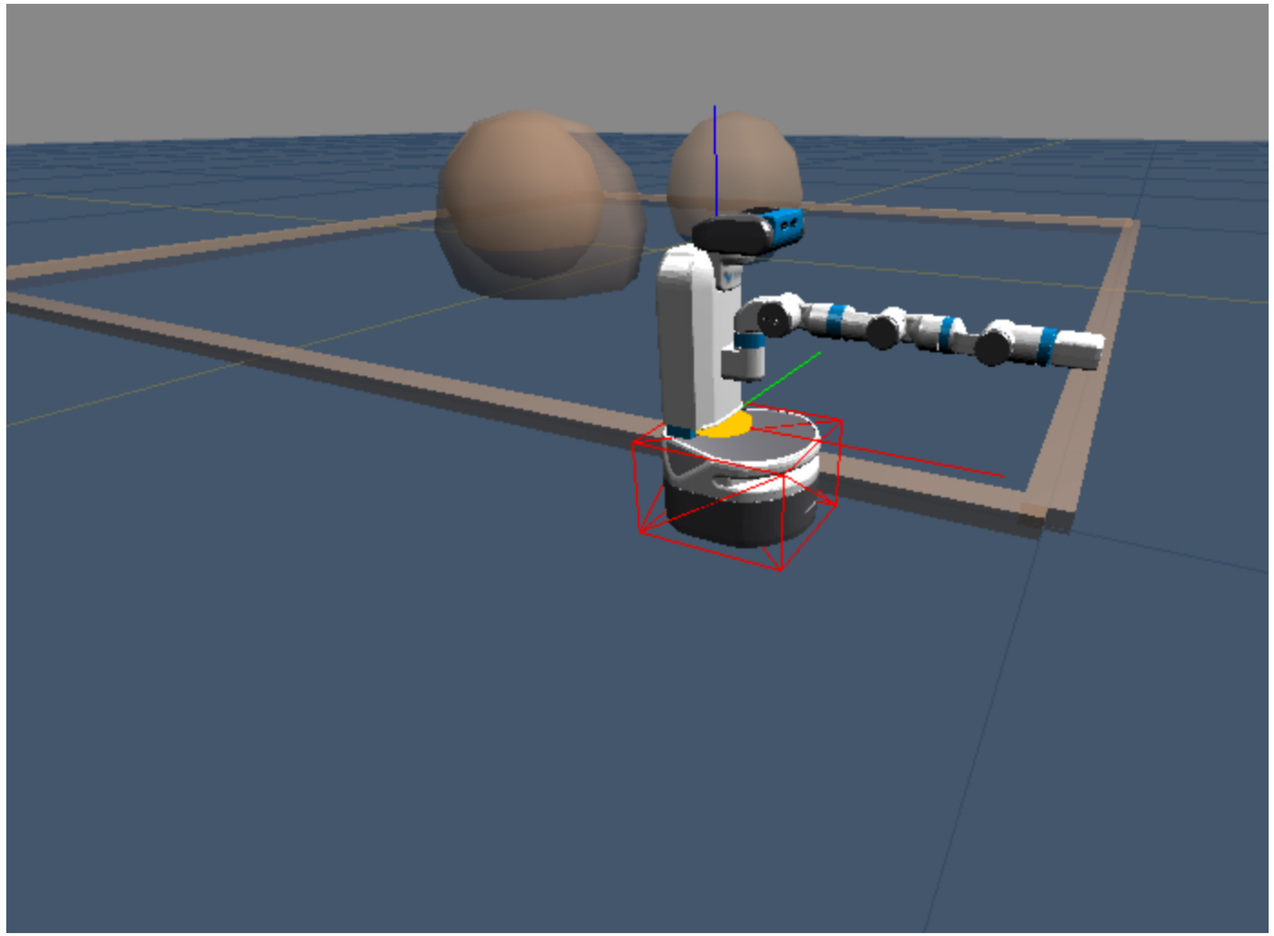
```
kineval.poseIsCollision = function robot_collision_test(q) {
    // perform collision test of robot geometry against planning world

    // test base origin (not extents) against world boundary extents
    if ((q[0]<robot_boundary[0][0])||(q[0]>robot_boundary[1][0])||
        (q[2]<robot_boundary[0][2])||(q[2]>robot_boundary[1][2]))
        return robot.base;

    // traverse robot kinematics to test each body for collision
    // STENCIL: implement forward kinematics for collision detection
    return robot_collision_forward_kinematics(q);

}
```

world boundary detection is provided

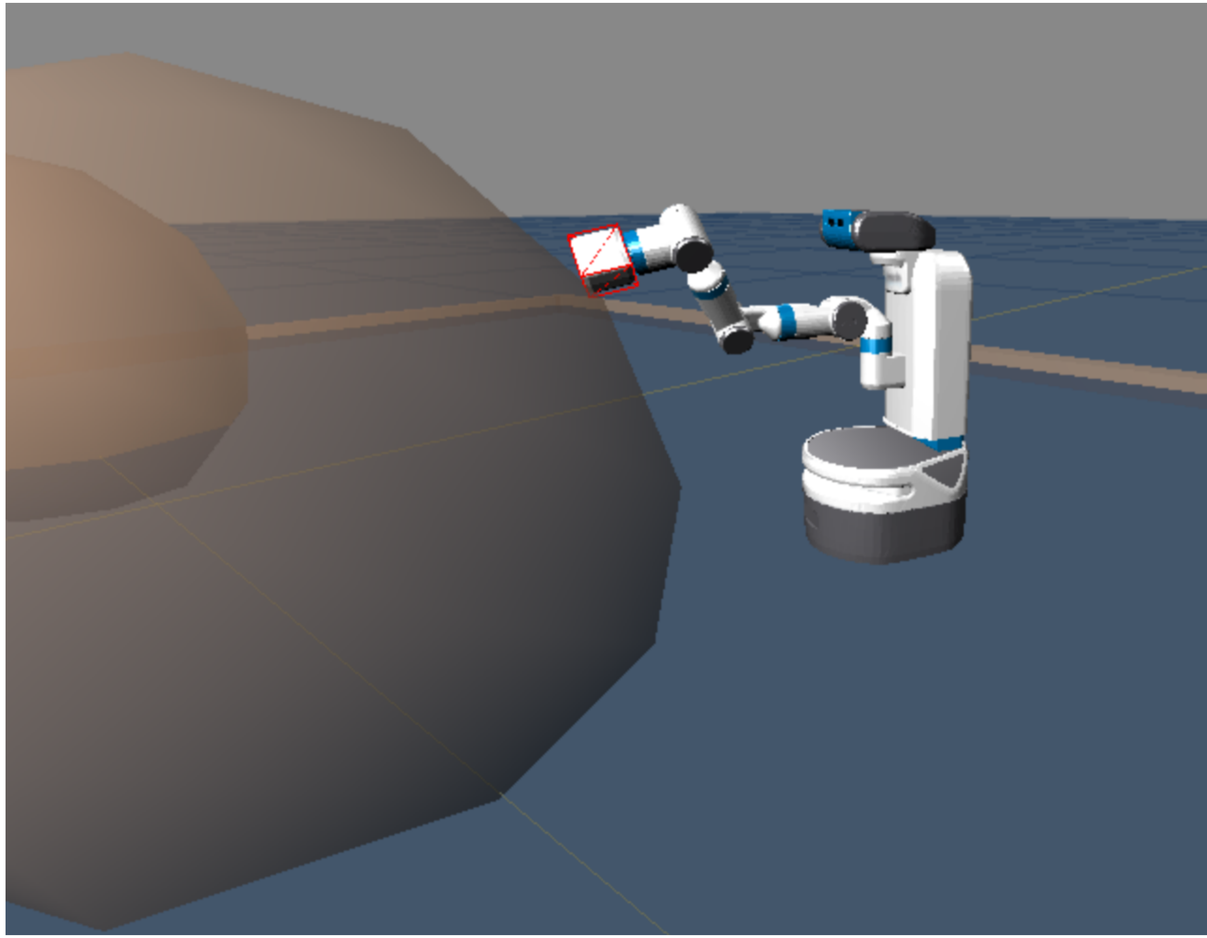
Uncomment this call;

Implement this function with FK transforms to test for collisions;

Use provided link collision function to test bounding box of each link



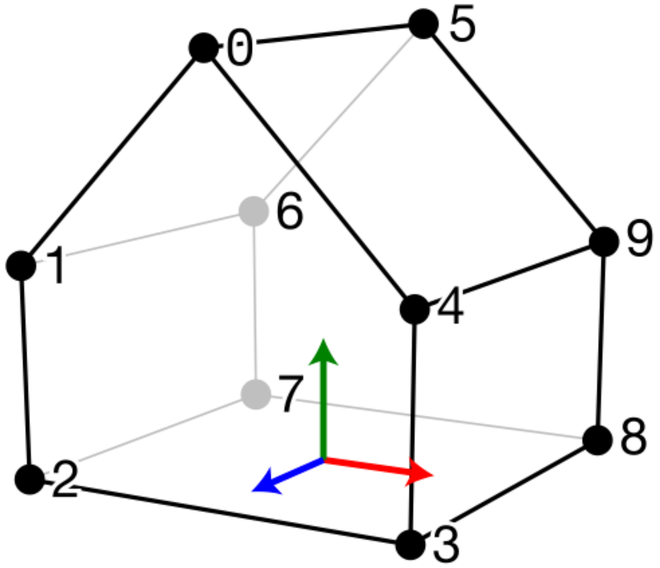
```
// test base origin (not extents) against world boundary extents
if ((q[0]<robot_boundary[0][0])||(q[0]>robot_boundary[1][0])||
    (q[2]<robot_boundary[0][2])||(q[2]>robot_boundary[1][2]))
    return robot.base;
```



```
// traverse robot kinematics to test each body for collision
// STENCIL: implement forward kinematics for collision detection
return robot_collision_forward_kinematics(q);
```

Remember:

Link Geometry



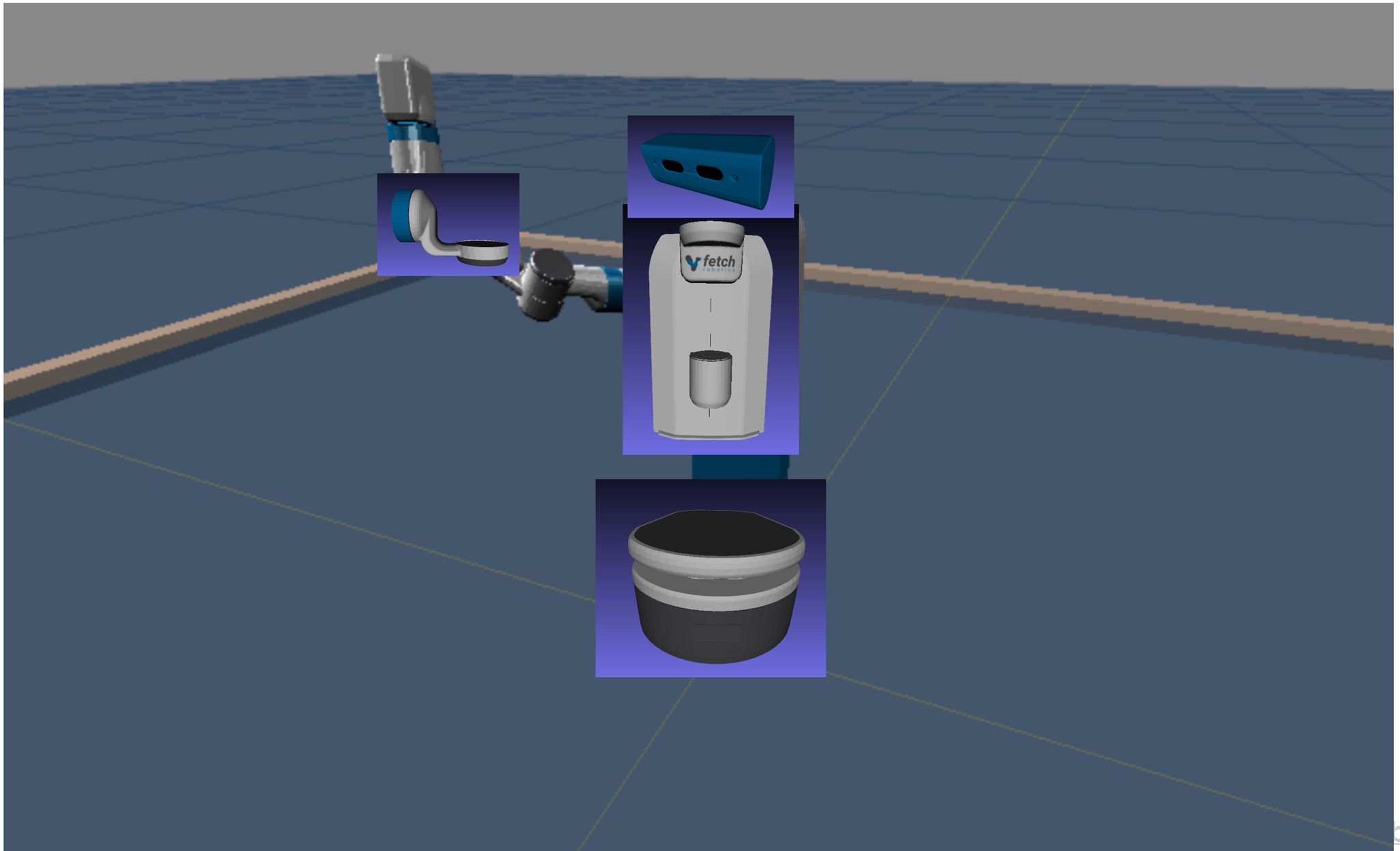
i	x	y	z
0	0.0	1.0	0.5
1	–0.5	0.5	0.5
2	–0.5	0.0	0.5
3	0.5	0.0	0.5
4	0.5	0.5	0.5
5	0.0	1.0	–0.5
6	–0.5	0.5	–0.5
7	–0.5	0.0	–0.5
8	0.5	0.0	–0.5
9	0.5	0.5	–0.5

Each link has a geometry specified as 3D vertices in the frame of the link connected into faces of its surface

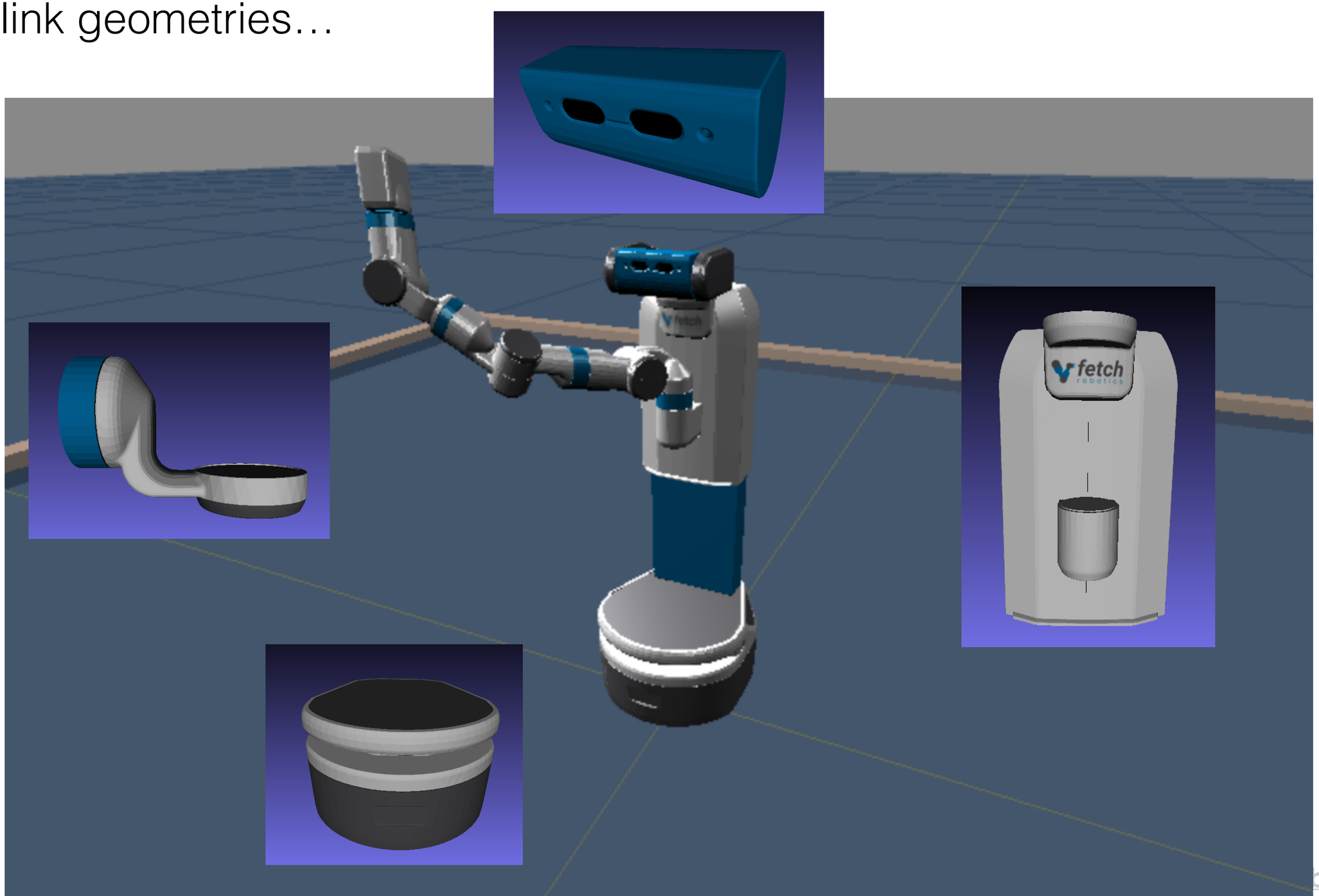
http://csc.lsu.edu/~kooima/courses/csc4356/

Individual link geometries…

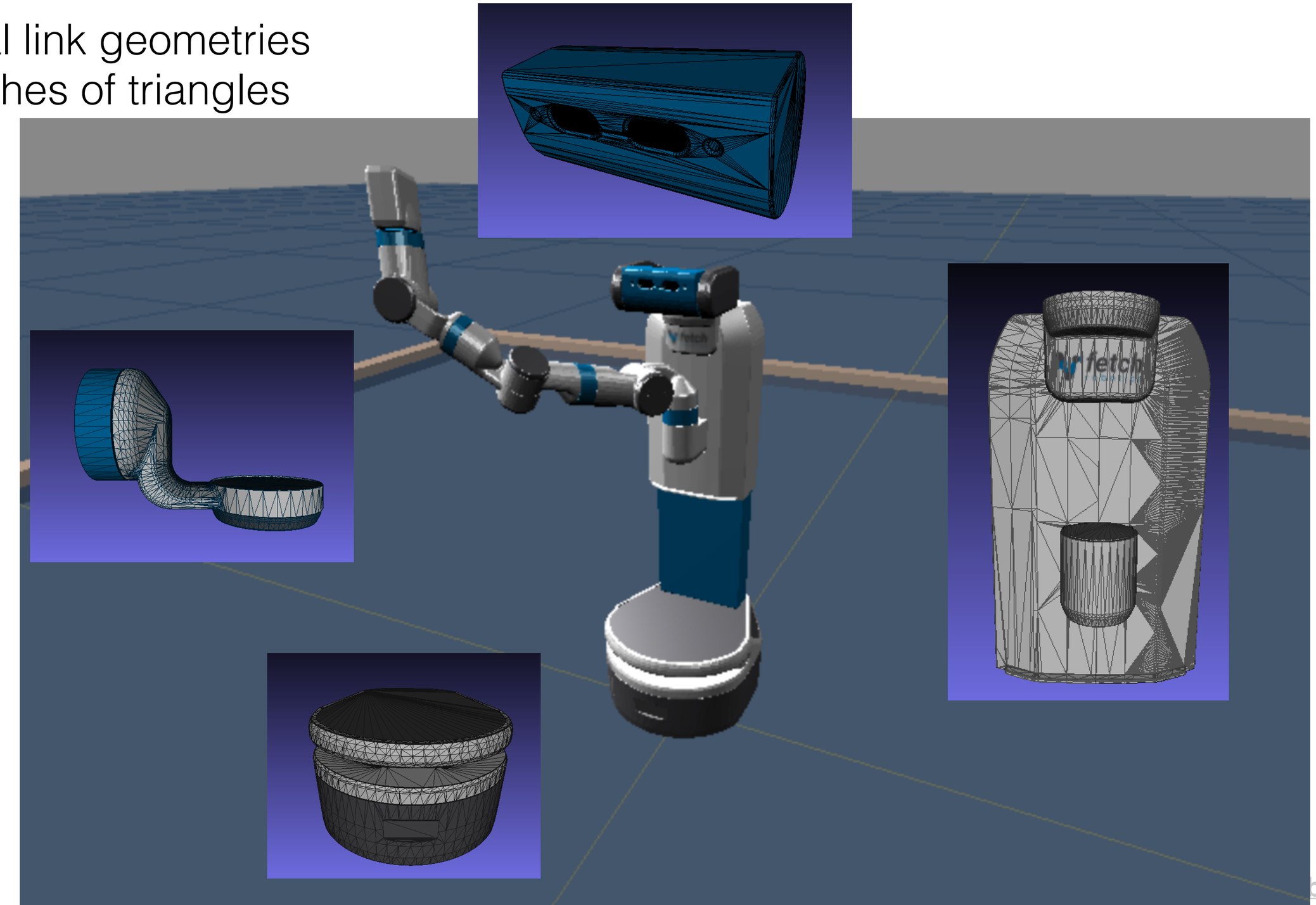
Individual link geometries…



Individual link geometries…



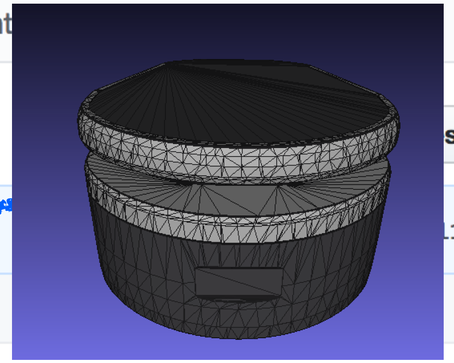
Individual link geometries are meshes of triangles



fetchrobotics / **fetch_ros**

Watch 18 Star 35 Fork 57

Code Issues 3 Pull requests 1 Projects 0 Wiki Insight



Branch: **indigo-devel** **fetch_ros** / fetch_description / **meshes** / s Find file History

mikeferguson update gripper model 119857 on Oct 10, 2015

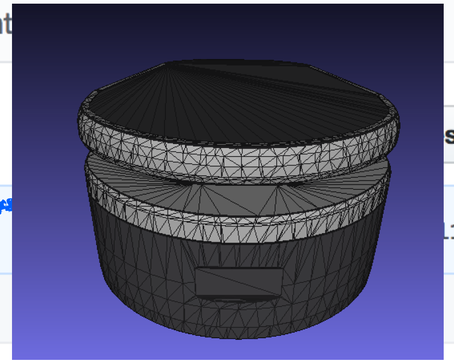
..

base_link.dae	add fetch description package	3 years ago
base_link_collision.STL	remove laser opening from collision mesh	3 years ago
base_link_uv.png	add fetch description package	3 years ago
bellows_link.STL	add fetch description package	3 years ago
bellows_link_collision.STL	add fetch description package	3 years ago
elbow_flex_link.dae	add fetch description package	3 years ago
elbow_flex_link_collision.STL	add fetch description package	3 years ago
elbow_flex_uv.png	add fetch description package	3 years ago
estop_link.STL	add fetch description package	3 years ago
forearm_roll_link.dae	add fetch description package	3 years ago
forearm_roll_link_collision.STL	add fetch description package	3 years ago

fetchrobotics / fetch_ros

Watch 18 Star 35 Fork 57

Code Issues 3 Pull requests 1 Projects 0 Wiki Insight



Branch: indigo-devel fetch_ros / fetch_description / meshes / s Find file History

mikeferguson update gripper model 119857 on Oct 10, 2015

..

base_link.dae	add fetch description package	3 years ago
base_link_collision.STL	remove laser opening from collision mesh	3 years ago
base_li		
bellows		
bellows		
elbow_		
elbow_		
elbow_		
estop_l		
forearm		
forearm_roll_link_collision.STL	add fetch description package	3 years ago

COLLADA

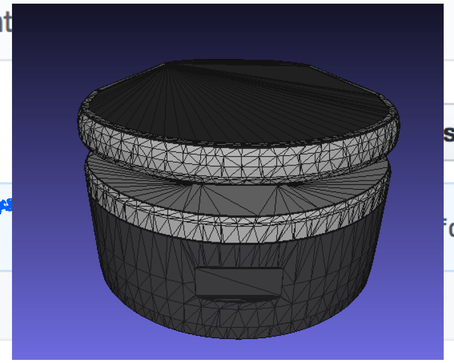
COLLADA (***COLLA**borative **D**esign **A**ctivity*) is an interchange file format for interactive 3D applications. It is managed by the nonprofit technology consortium, the Khronos Group, and has been adopted by ISO as a publicly available specification, ISO/PAS 17506.[1]

COLLADA defines an open standard XML schema for exchanging digital assets among various graphics software applications that might otherwise store their assets in incompatible file formats. COLLADA documents that describe digital assets are XML files, usually identified with a **.dae** (digital asset exchange) filename extension.

ohseejay / kineval-stencil-fall16

Watch 2 Star 1 Fork 2

Code Issues 0 Pull requests 0 Projects 0 Wiki Insight



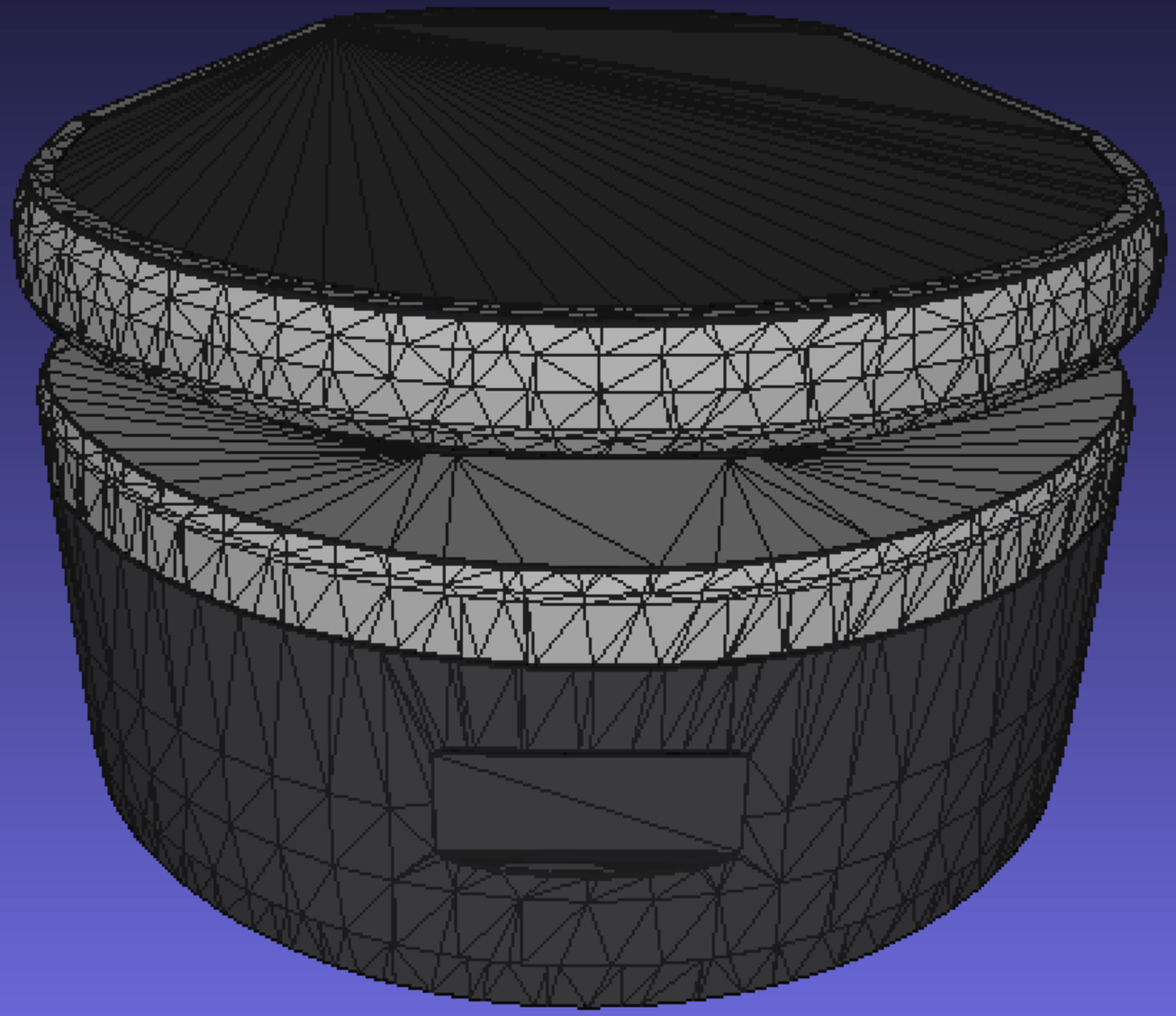
Branch: master kineval-stencil-fall16 / robots / fetch / s Find file History

odestcj initial commit fd521e on Oct 19, 2016

..

base_link.dae	initial commit	a year ago
base_link_adjusted.dae	initial commit	a year ago
base_link_collision.STL	initial commit	a year ago
base_link_uv.png	initial commit	a year ago
bellows_link.STL	initial commit	a year ago
bellows_link_collision.STL	initial commit	a year ago
elbow_flex_link.dae	initial commit	a year ago
elbow_flex_link_adjusted.dae	initial commit	a year ago
elbow_flex_link_collision.STL	initial commit	a year ago
elbow_flex_uv.png	initial commit	a year ago
estop_link.STL	initial commit	a year ago

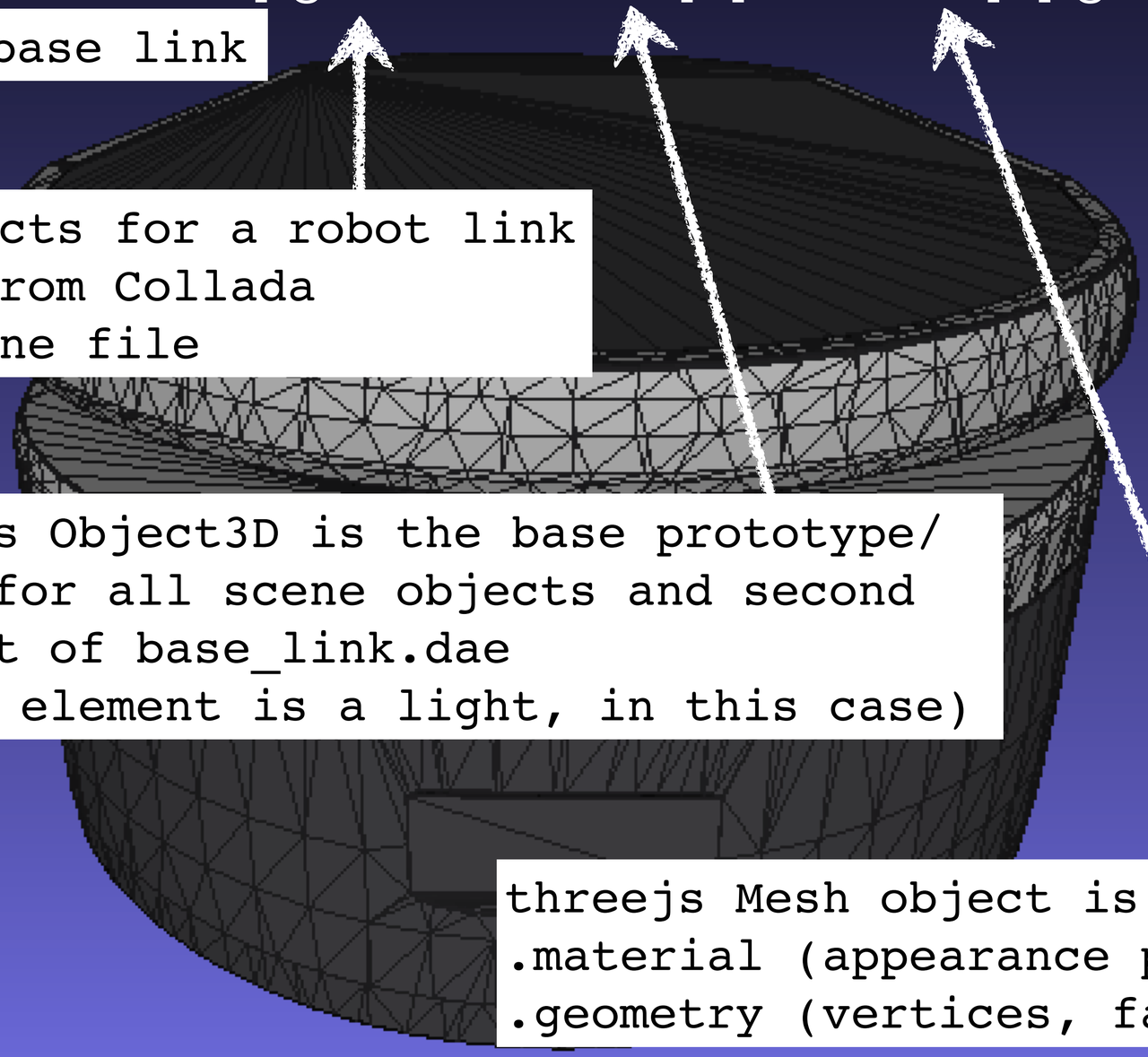
Vertices: robot.links[robot.base].geom.children[1].children[0].geometry.vertices
Faces: robot.links[robot.base].geom.children[1].children[0].geometry.faces



Vertices: robot.links[robot.base].geom.children[1].children[0].geometry.vertices

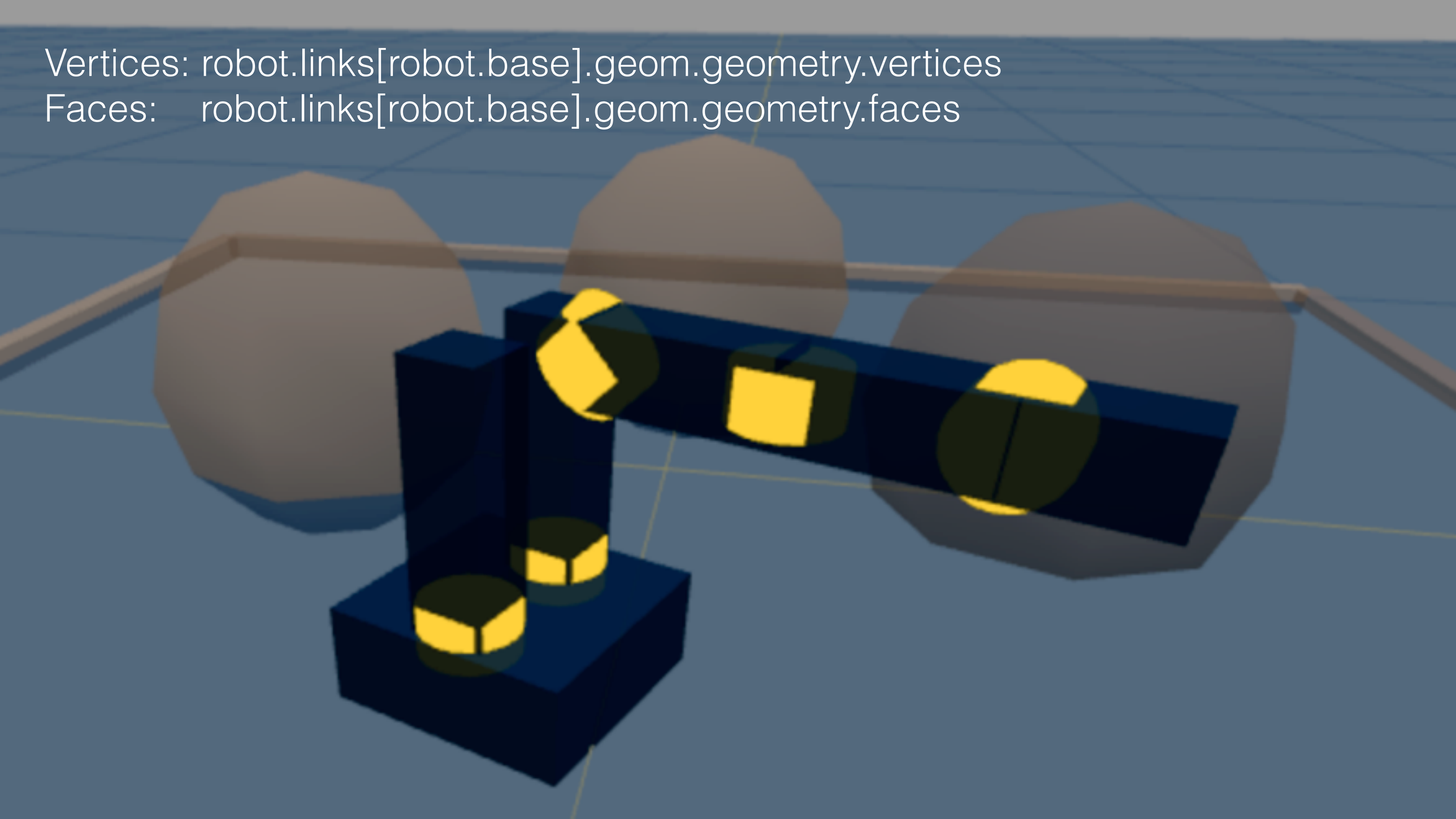
KinEval robot base link

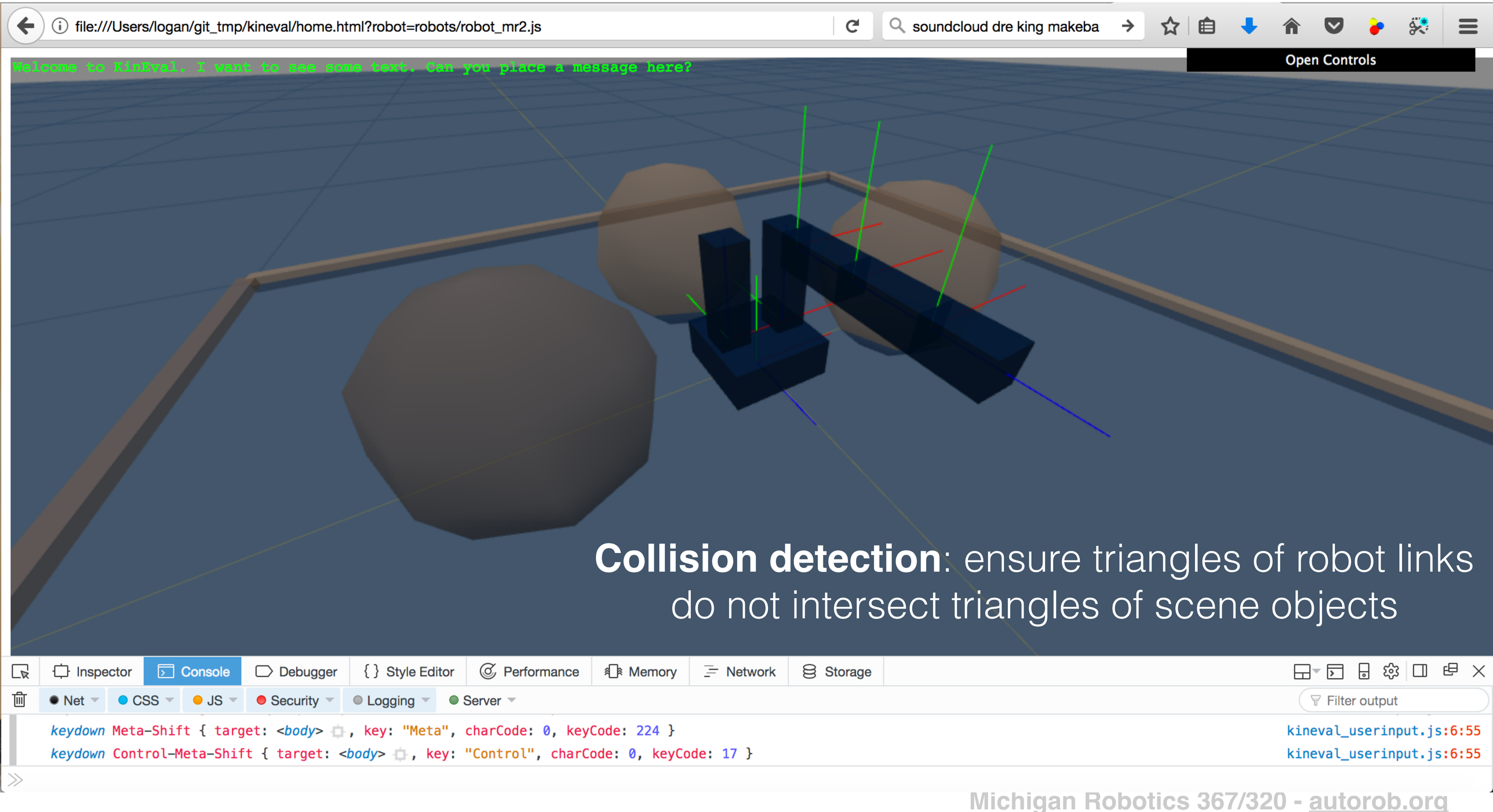
.geom: threejs objects for a robot link (or joint) loaded from Collada (base_link.dae) scene file

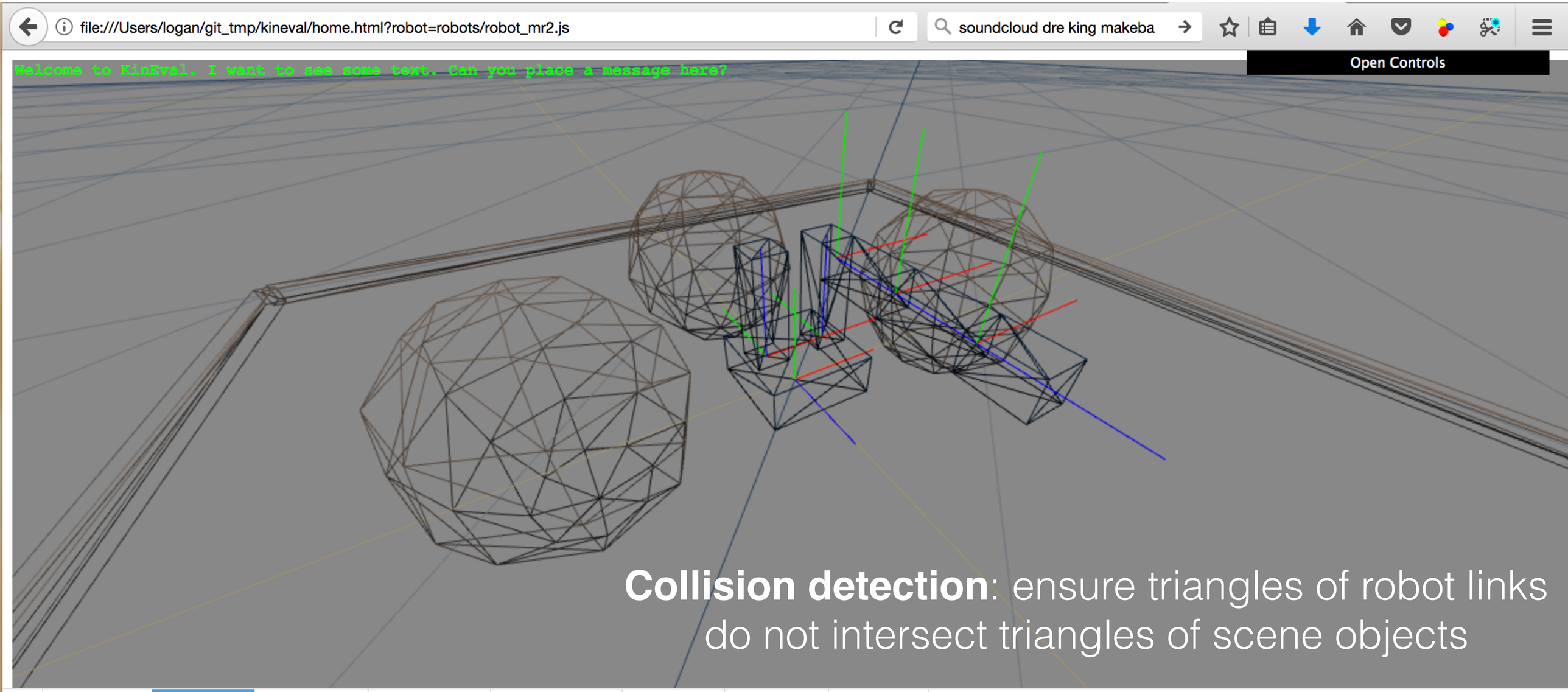


threejs Object3D is the base prototype/ class for all scene objects and second element of base_link.dae (first element is a light, in this case)

threejs Mesh object is consists of:
.material (appearance properties)
.geometry (vertices, faces, normals)


Vertices: robot.links[robot.base].geom.geometry.vertices
Faces: robot.links[robot.base].geom.geometry.faces

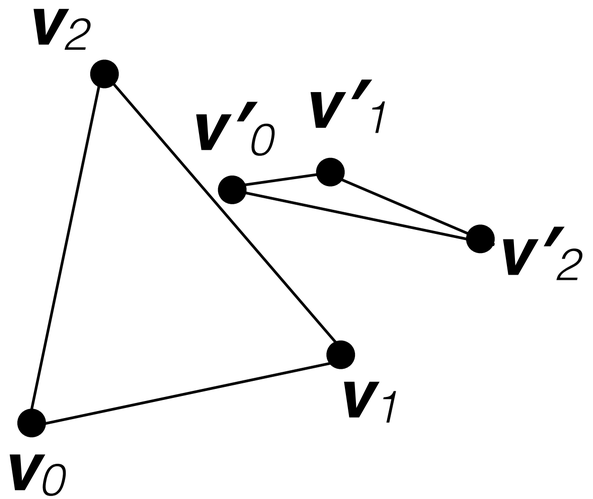

file:///Users/logan/git_tmp/kineval/home.html?robot=robots/robot_mr2.js
soundcloud dre king makeba
Welcome to KinEval. I want to see some text. Can you place a message here?
Open Controls
Collision detection: ensure triangles of robot links do not intersect triangles of scene objects
Inspector
Console
Debugger
Style Editor
Performance
Memory
Network
Storage
Net
CSS
JS
Security
Logging
Server
Filter output
keydown Meta-Shift { target: <body> , key: "Meta", charCode: 0, keyCode: 224 }
kineval_userinput.js:6:55
keydown Control-Meta-Shift { target: <body> , key: "Control", charCode: 0, keyCode: 17 }
kineval_userinput.js:6:55


file:///Users/logan/git_tmp/kineval/home.html?robot=robots/robot_mr2.js
soundcloud dre king makeba
Open Controls
Welcome to KinEval. I want to see some text. Can you place a message here?
Collision detection: ensure triangles of robot links do not intersect triangles of scene objects
Inspector
Console
Debugger
Style Editor
Performance
Memory
Network
Storage
Net
CSS
JS
Security
Logging
Server
Filter output
keydown Meta-Shift { target: <body>, key: "Shift", charCode: 0, keyCode: 16 }
kineval_userinput.js:6:55
keydown Control-Meta-Shift { target: <body>, key: "Control", charCode: 0, keyCode: 17 }
kineval_userinput.js:6:55
for (i=0;i<scene.children.length;i++) { a = scene.children[i]; if (typeof a.material !== 'undefined') a.material.wireframe=true;};

How do we test whether two triangles intersect?

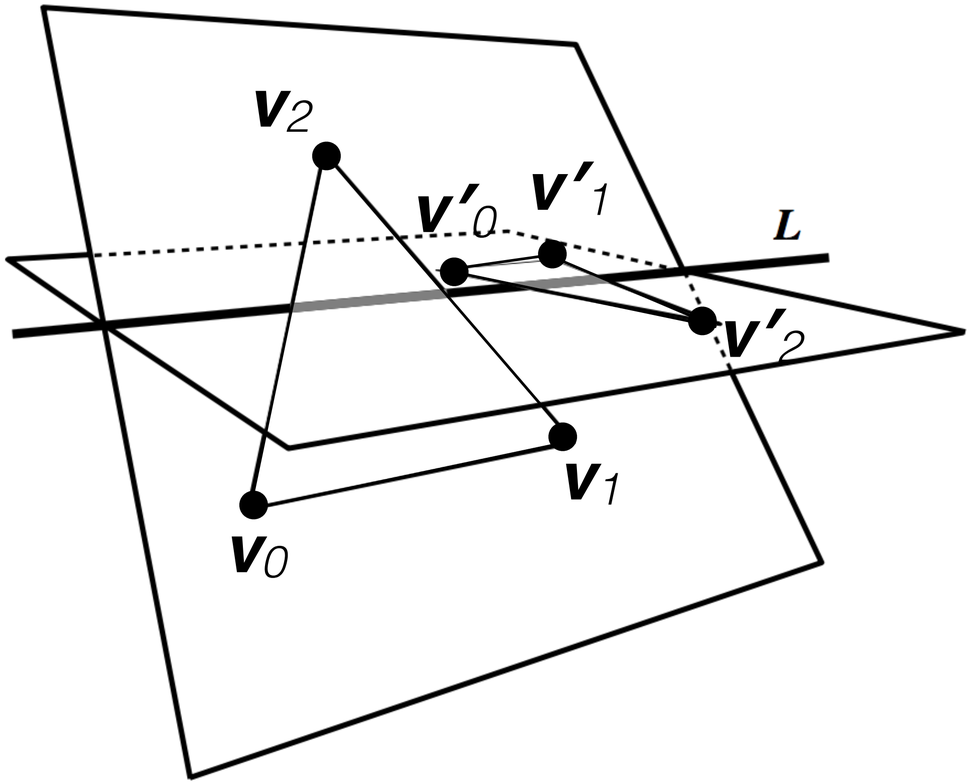
3D Triangle-Triangle Test

- Given two triangles each with three vertices
  - $T = \{\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2\}$
  - $T' = \{\boldsymbol{v}'_0, \boldsymbol{v}'_1, \boldsymbol{v}'_2\}$
- Return true if T and T' intersect



3D Triangle-Triangle Test

1. Compute plane equation of triangle 2.
2. Reject as trivial if all points of triangle 1 are on same side.
3. Compute plane equation of triangle 1.
4. Reject as trivial if all points of triangle 2 are on same side.
5. Compute intersection line and project onto largest axis.
6. Compute the intervals for each triangle.
7. Intersect the intervals.

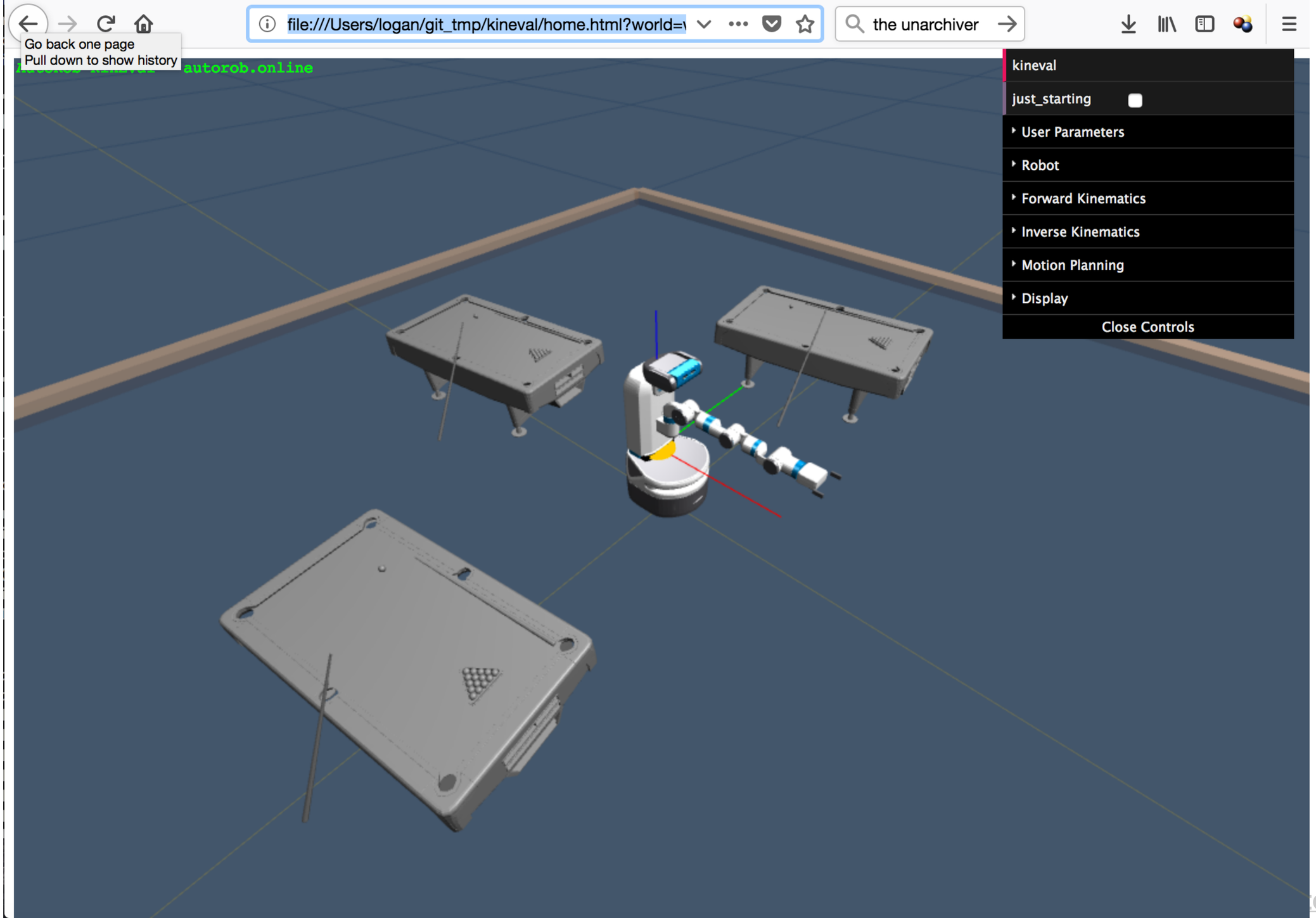


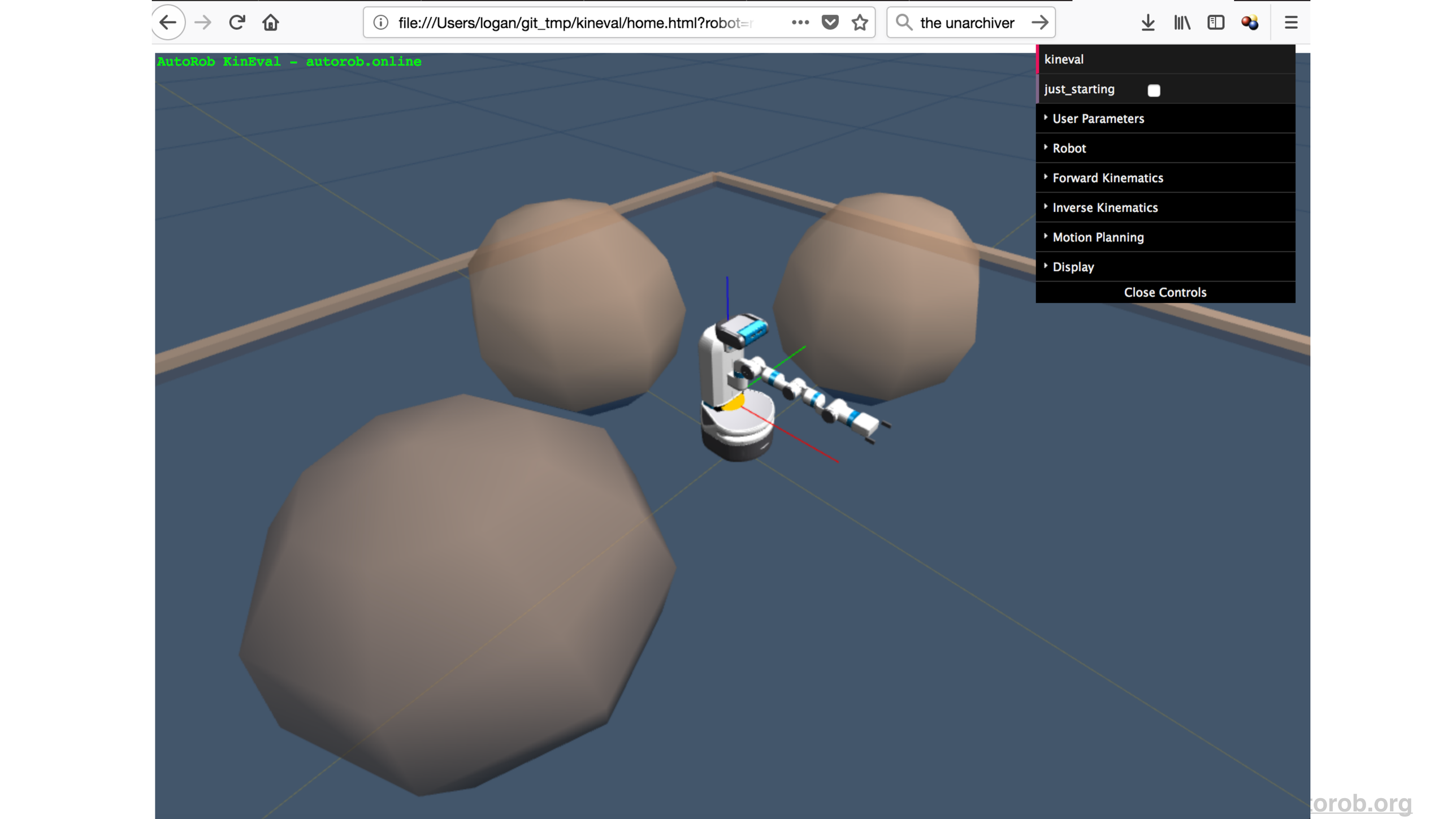
Möller 1997

How many triangle tests must be performed?

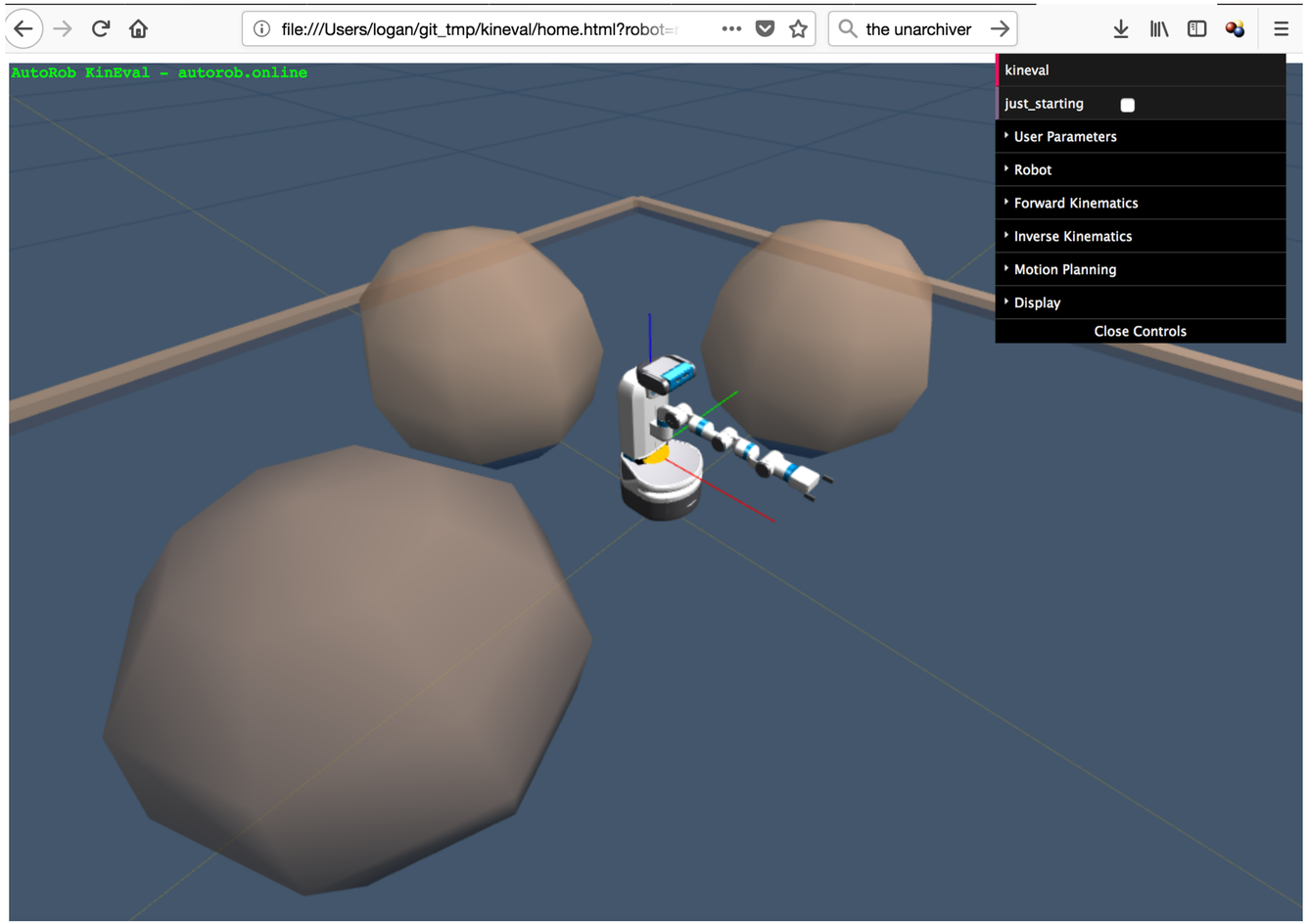
How many triangle tests must be performed?

Can we reduce the number of tests to evaluate?


Go back one page
Pull down to show history
file:///Users/logan/git_tmp/kineval/home.html?world=v
the unarchiver
autorob.online
kineval
just_starting
User Parameters
Robot
Forward Kinematics
Inverse Kinematics
Motion Planning
Display
Close Controls

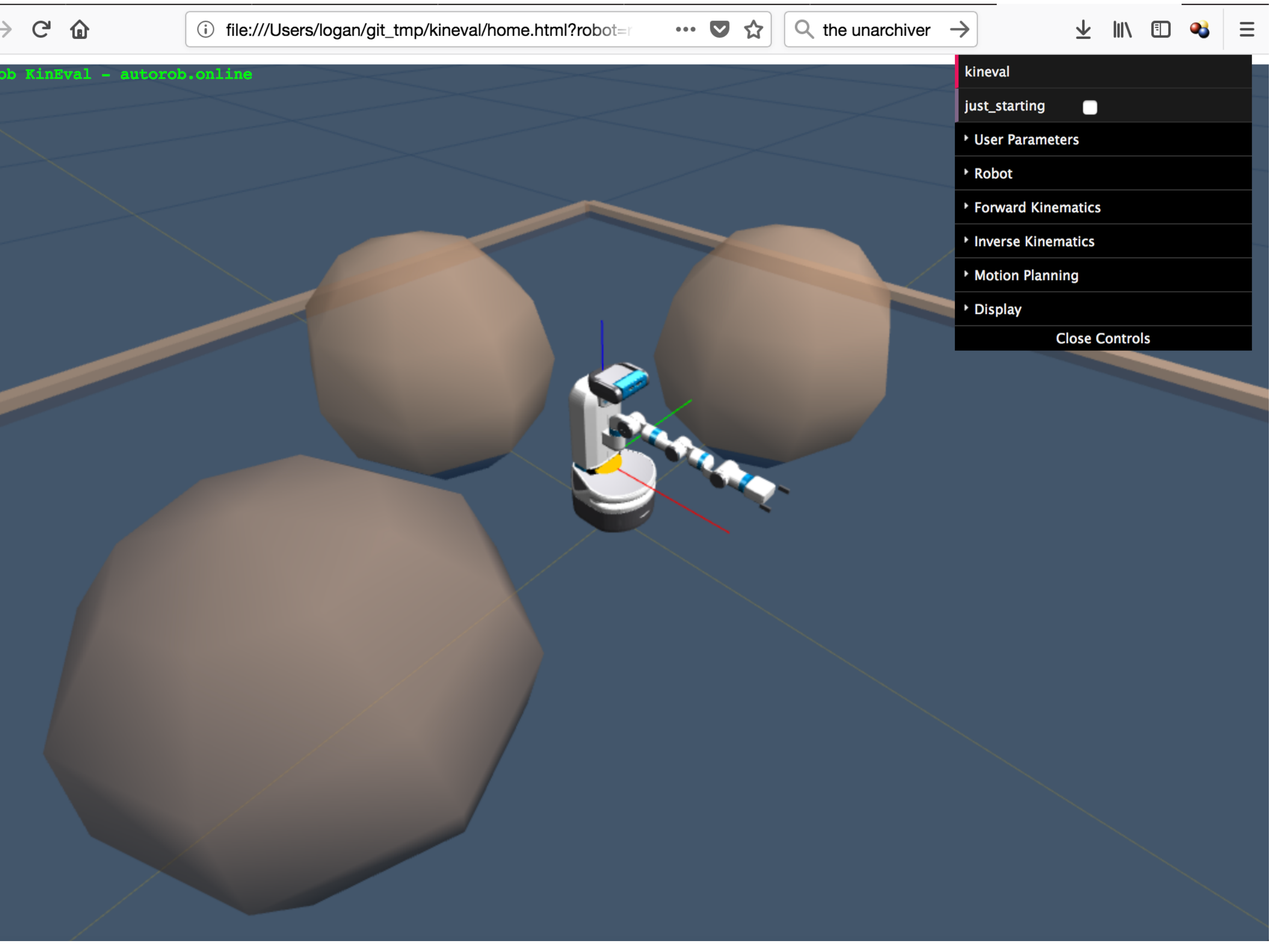

file:///Users/logan/git_tmp/kineval/home.html?robot=r
the unarchiver
AutoRob KinEval - autorob.online
kineval
just_starting
User Parameters
Robot
Forward Kinematics
Inverse Kinematics
Motion Planning
Display
Close Controls

KinEval approximates obstacles with bounding spheres

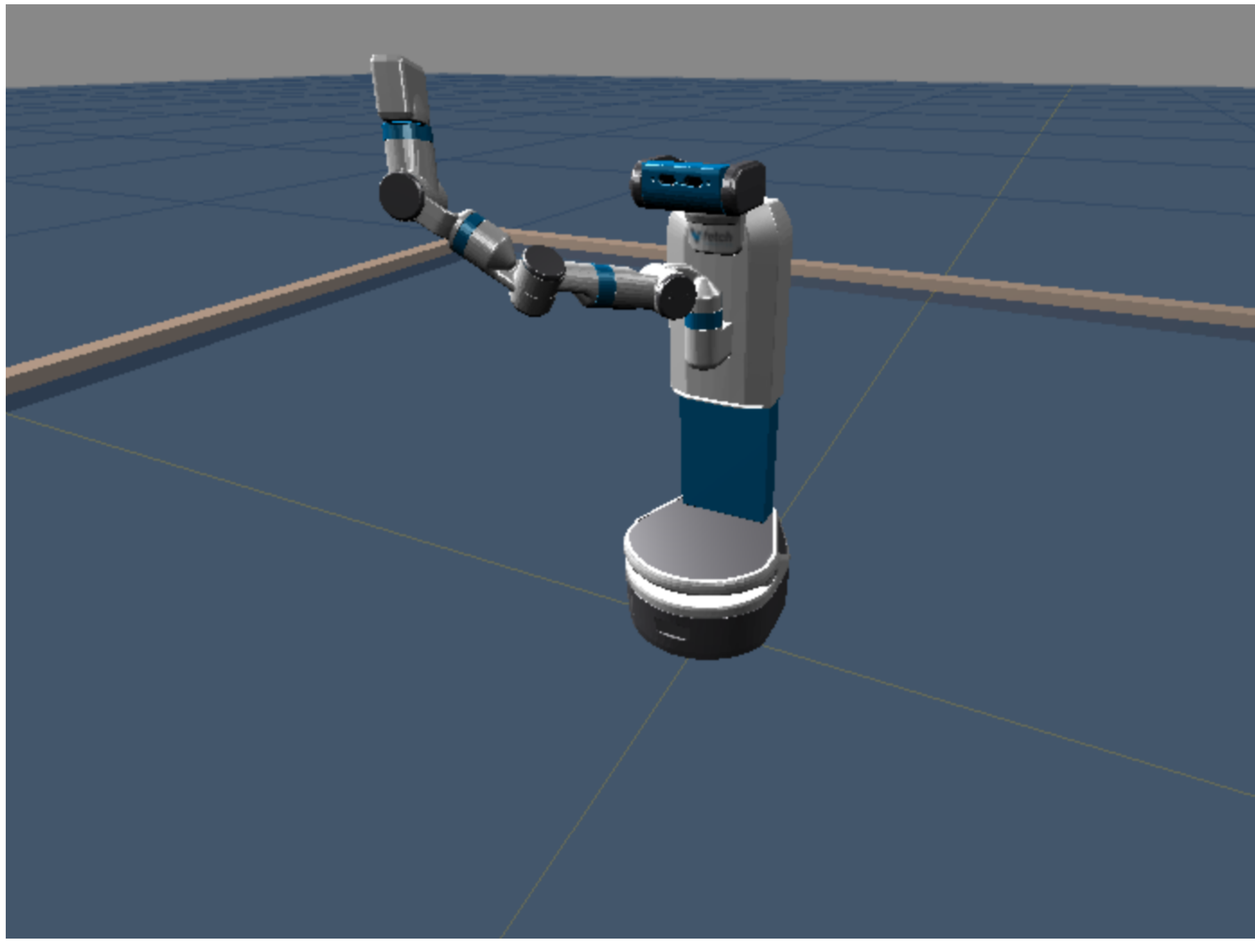


KinEval approximates

obstacles with bounding spheres

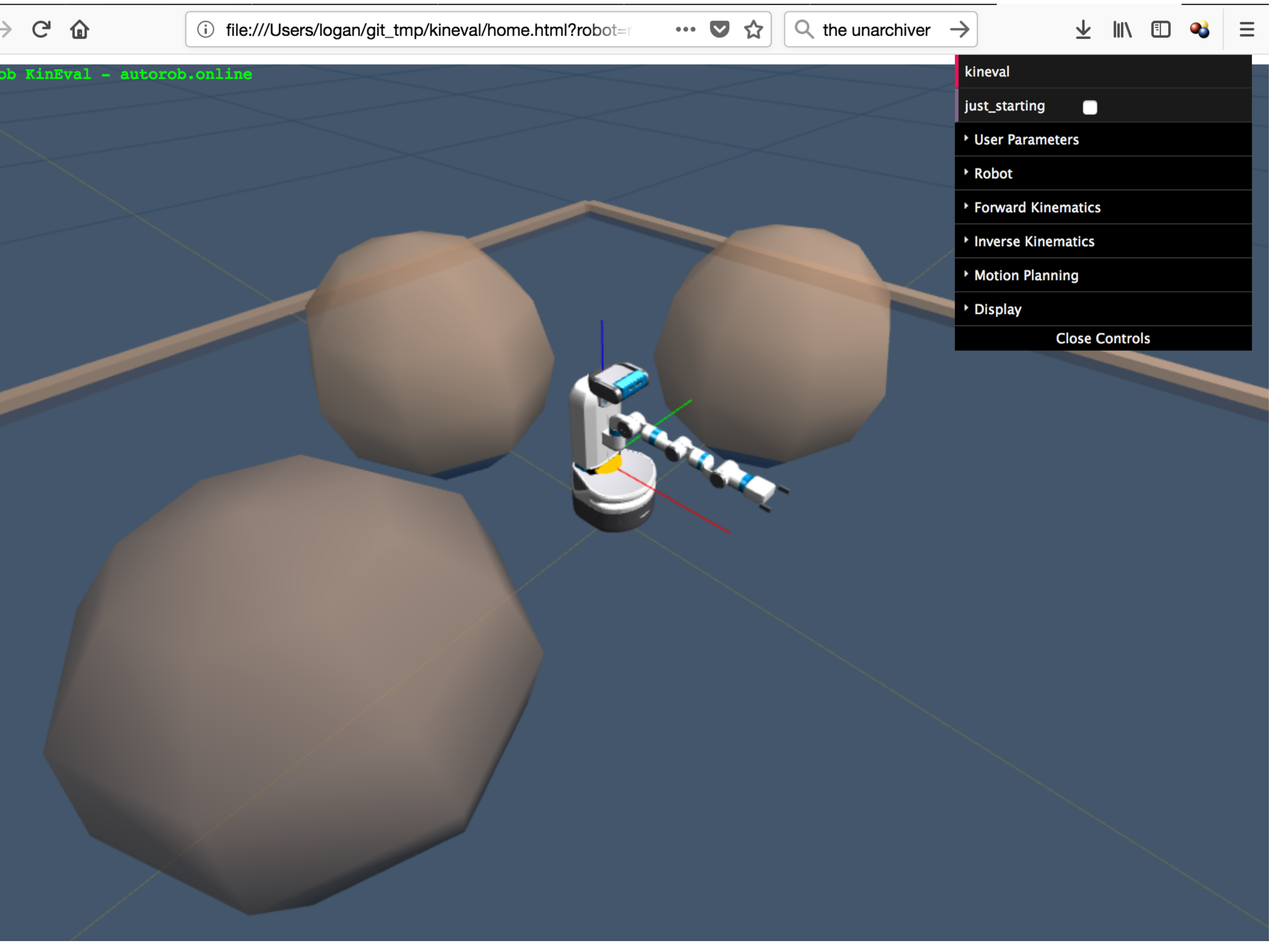


and the robot?

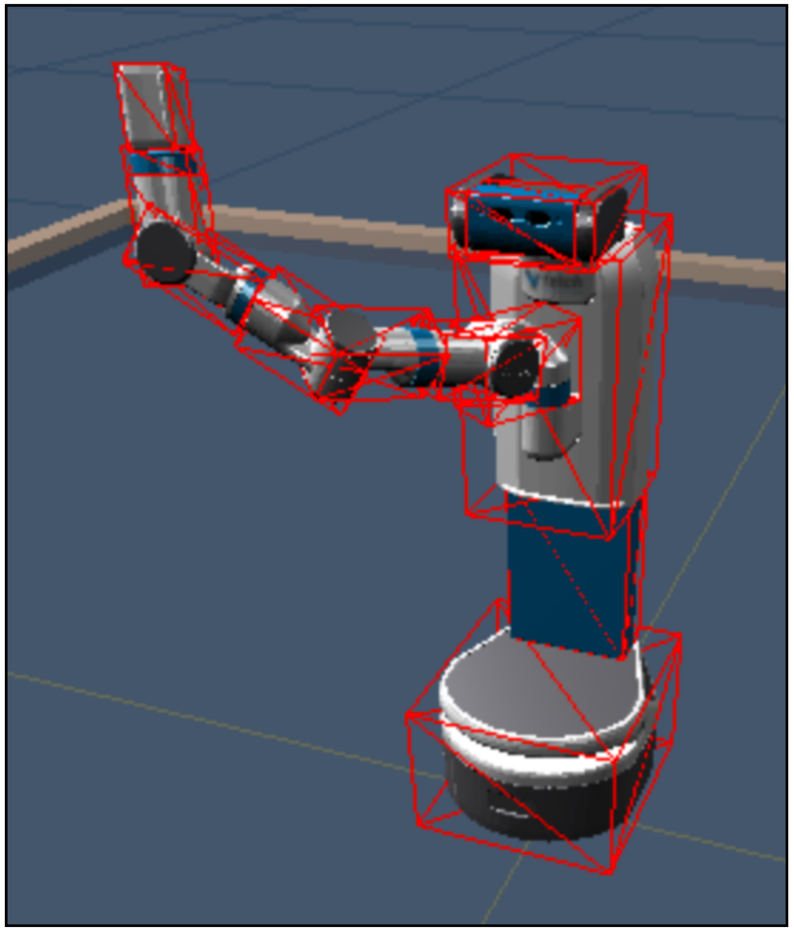


KinEval approximates

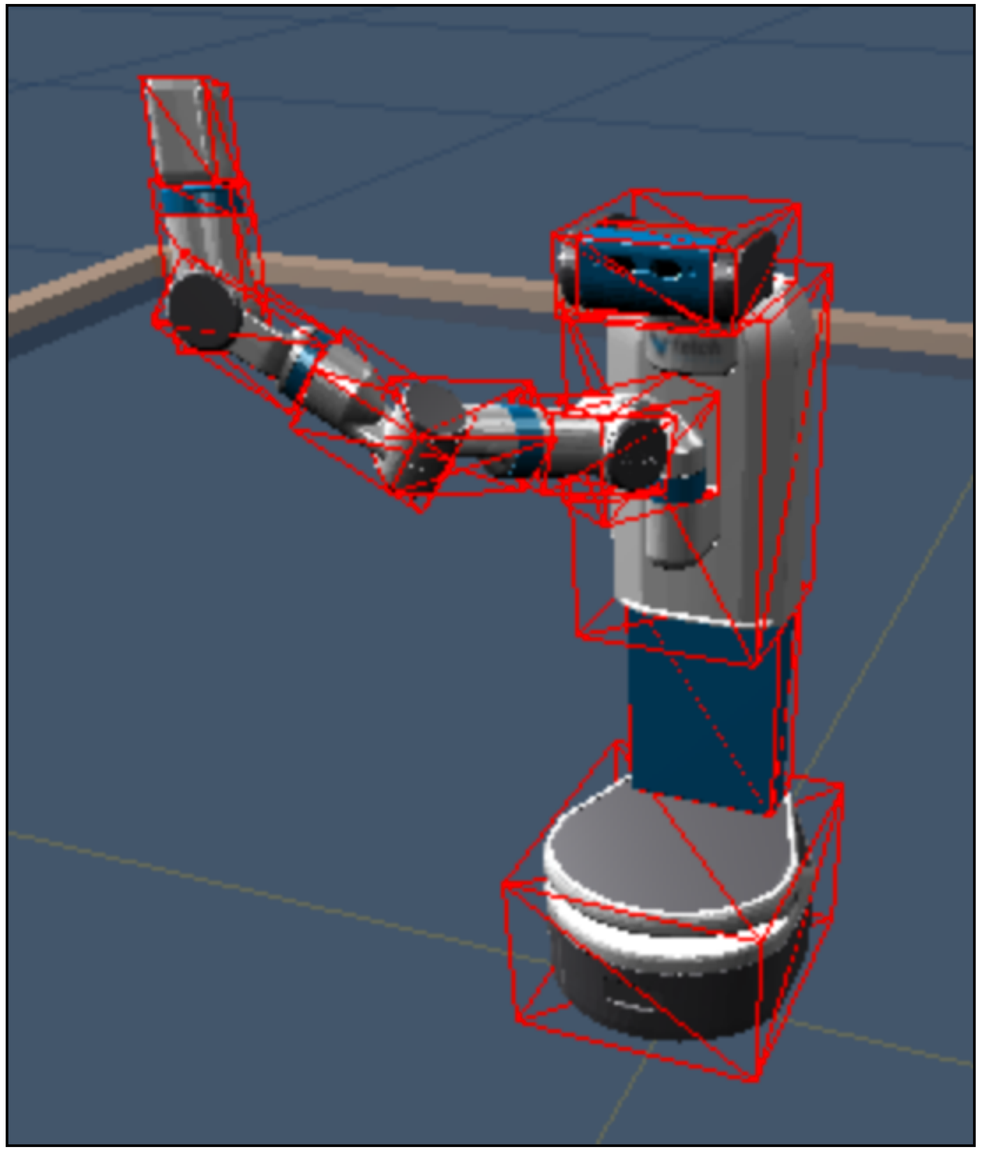
obstacles with bounding spheres

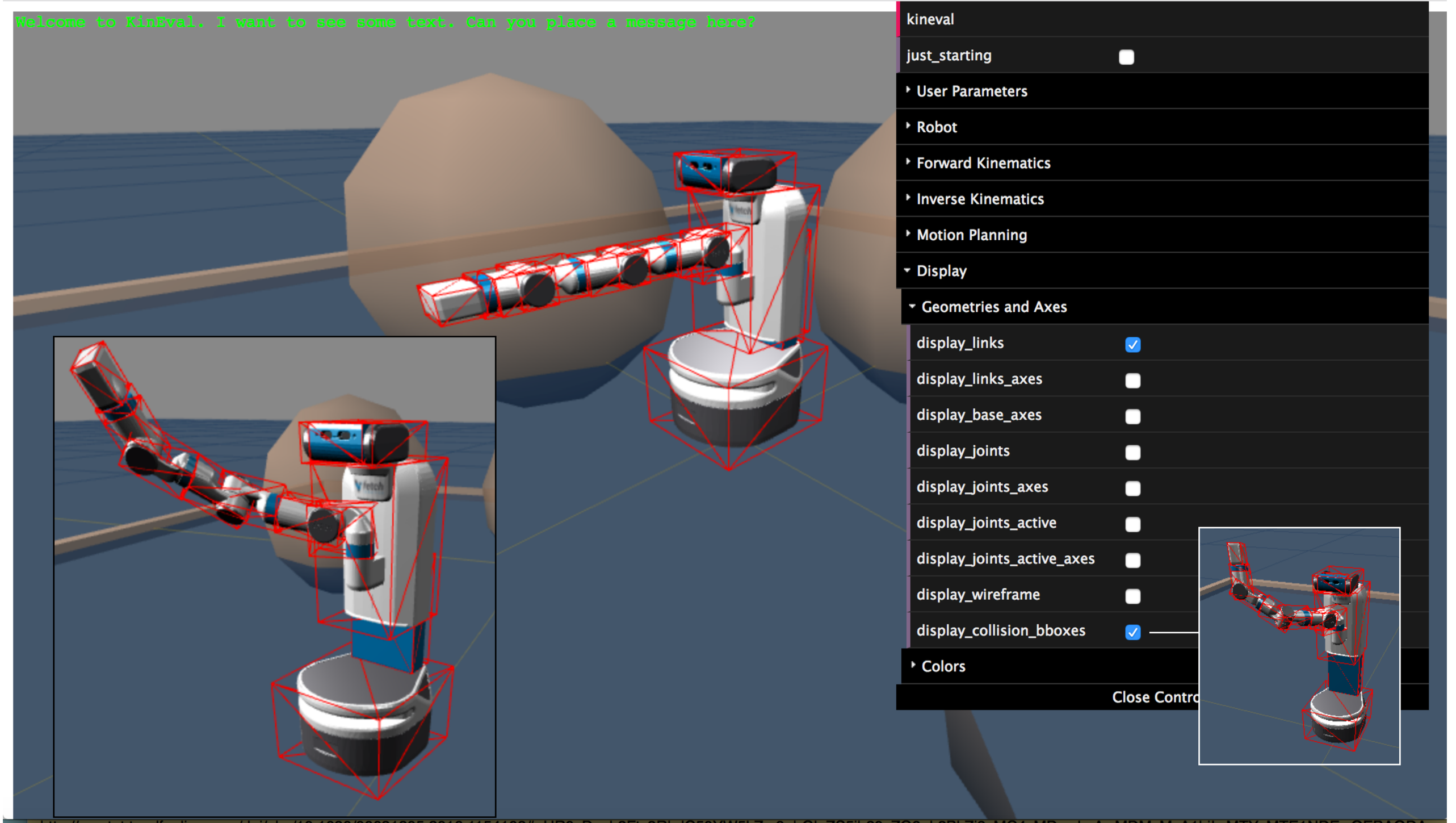


robot as bounding boxes

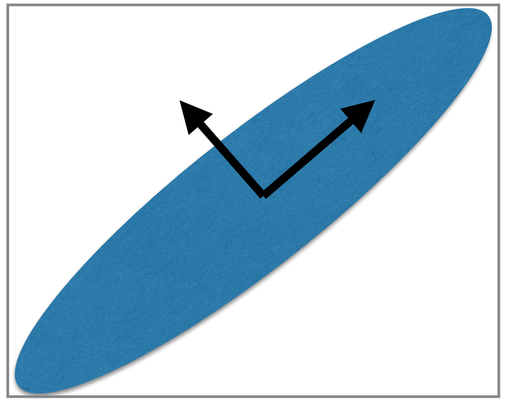


KinEval approximates
link geometries
with bounding boxes

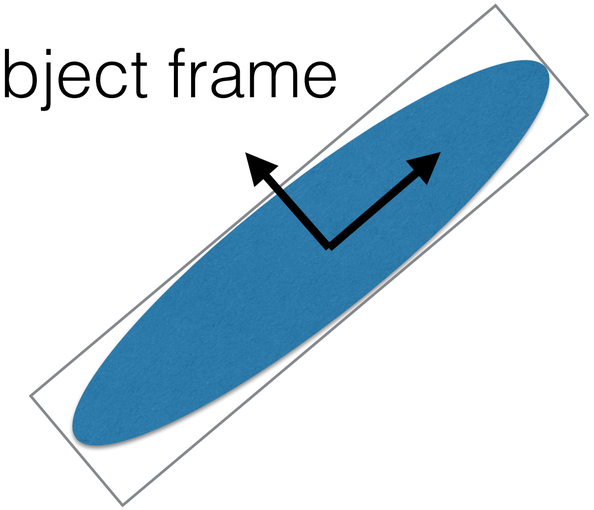



Welcome to KinEval. I want to see some text. Can you place a message here?
kineval
just_starting
User Parameters
Robot
Forward Kinematics
Inverse Kinematics
Motion Planning
Display
Geometries and Axes
display_links
display_links_axes
display_base_axes
display_joints
display_joints_axes
display_joints_active
display_joints_active_axes
display_wireframe
display_collision_bboxes
Colors
Close Contro

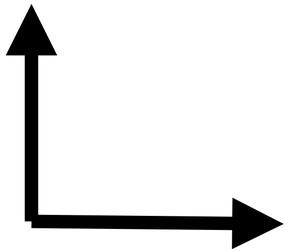
Bounding Boxes



Axis-aligned Bounding Box
(AABB)



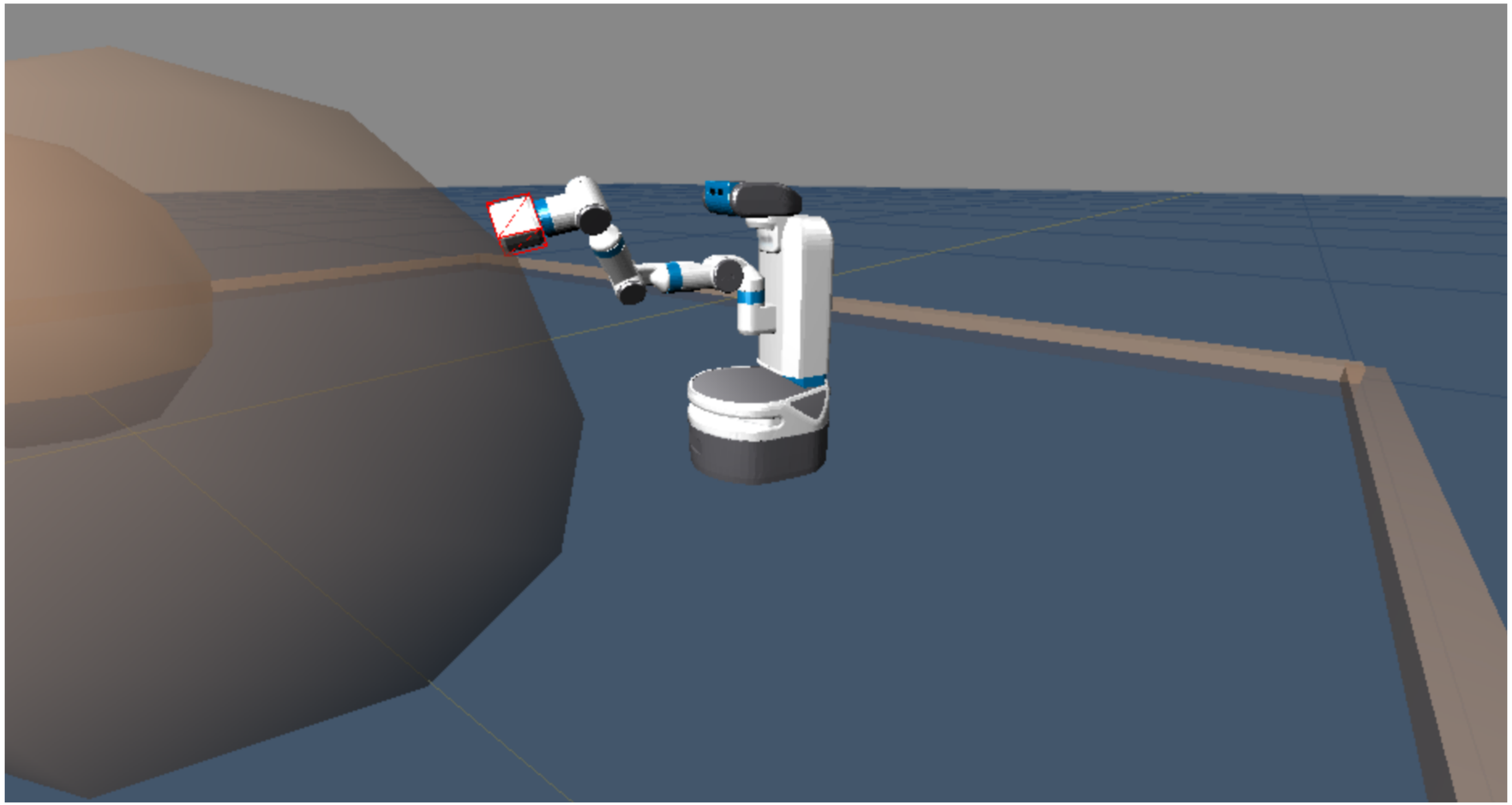
Oriented Bounding Box
(OBB)



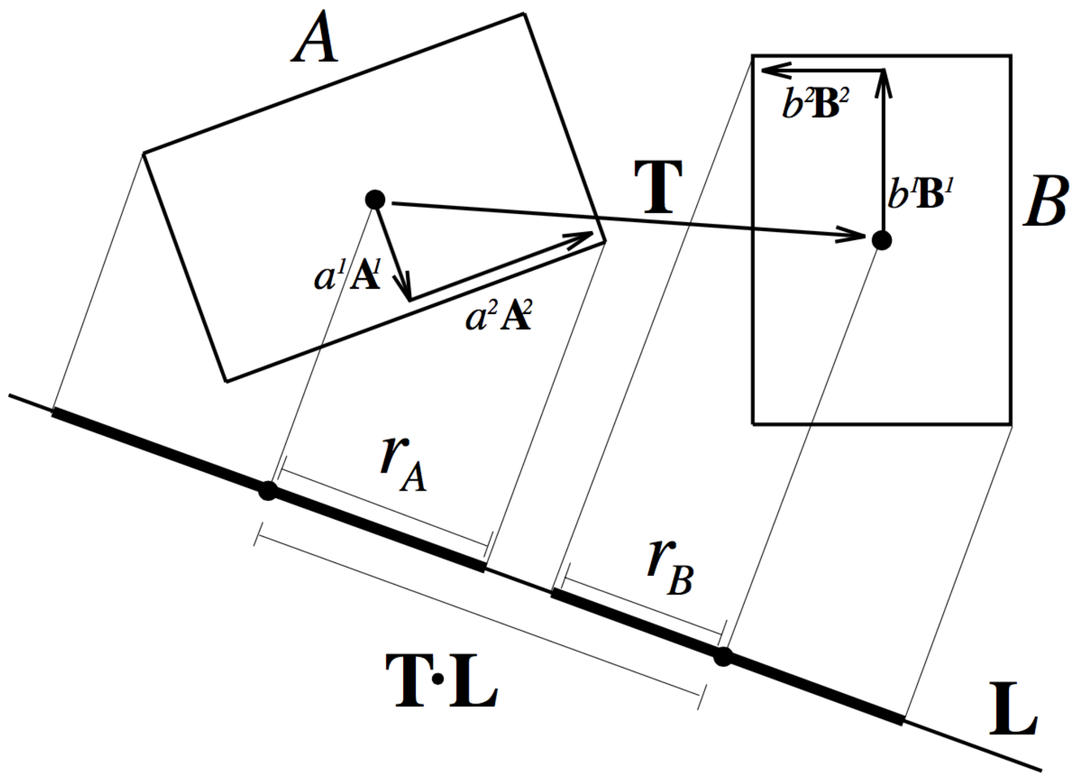
world frame

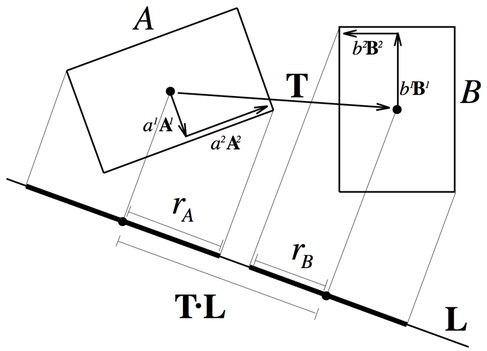
Gottschalk et al. 1996

Only a “separating axis” needs to be found



Separating Axis Theorem





Hyperplane separation theorem

From Wikipedia, the free encyclopedia

(Redirected from Separating axis theorem)

In geometry, the **hyperplane separation theorem** is a theorem about disjoint convex sets in *n*-dimensional Euclidean space. There are several rather similar versions. In one version of the theorem, if both these sets are closed and at least one of them is compact, then there is a hyperplane in between them and even two parallel hyperplanes in between them separated by a gap. In another version, if both disjoint convex sets are open, then there is a hyperplane in between them, but not necessarily any gap. An axis which is orthogonal to a separating hyperplane is a **separating axis**, because the orthogonal projections of the convex bodies onto the axis are disjoint.

The hyperplane separation theorem is due to Hermann Minkowski. The Hahn–Banach separation theorem generalizes the result to topological vector spaces.

A related result is the supporting hyperplane theorem.

In geometry, a **maximum-margin hyperplane** is a hyperplane which separates two 'clouds' of points and is at equal distance from the two. The margin between the hyperplane and the clouds is maximal. See the article on Support Vector Machines for more details.

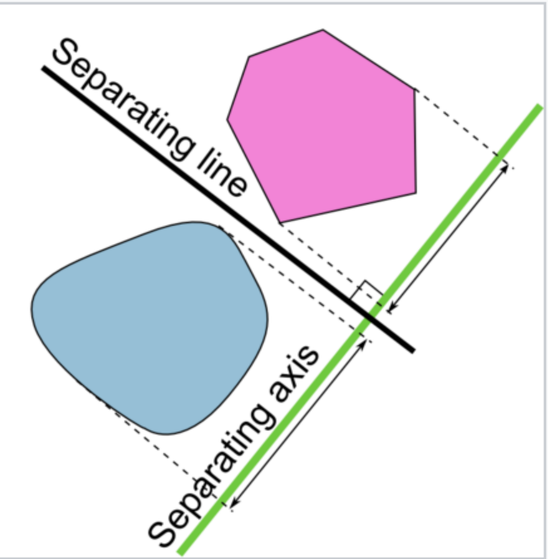
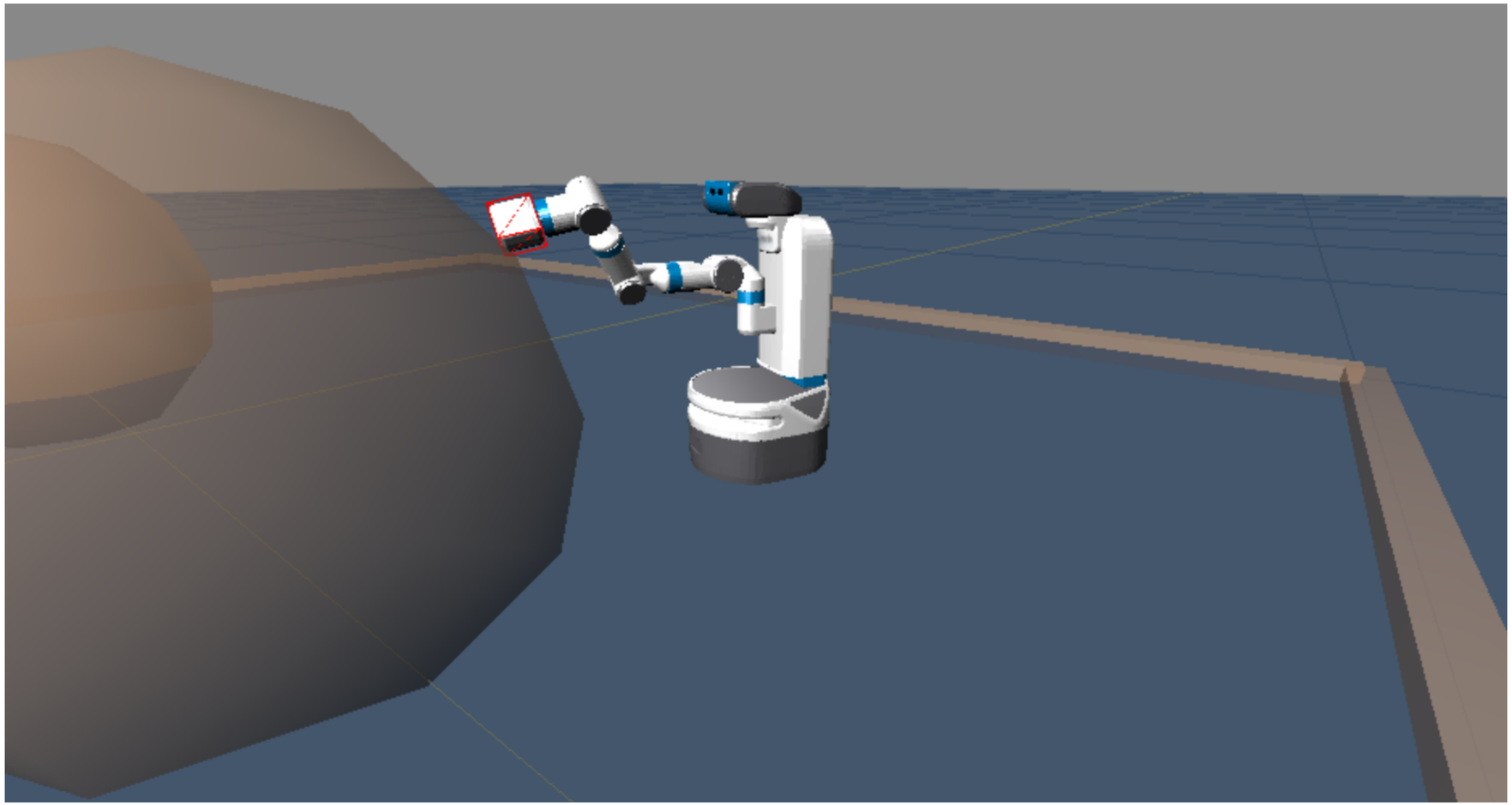


Illustration of the hyperplane separation theorem.

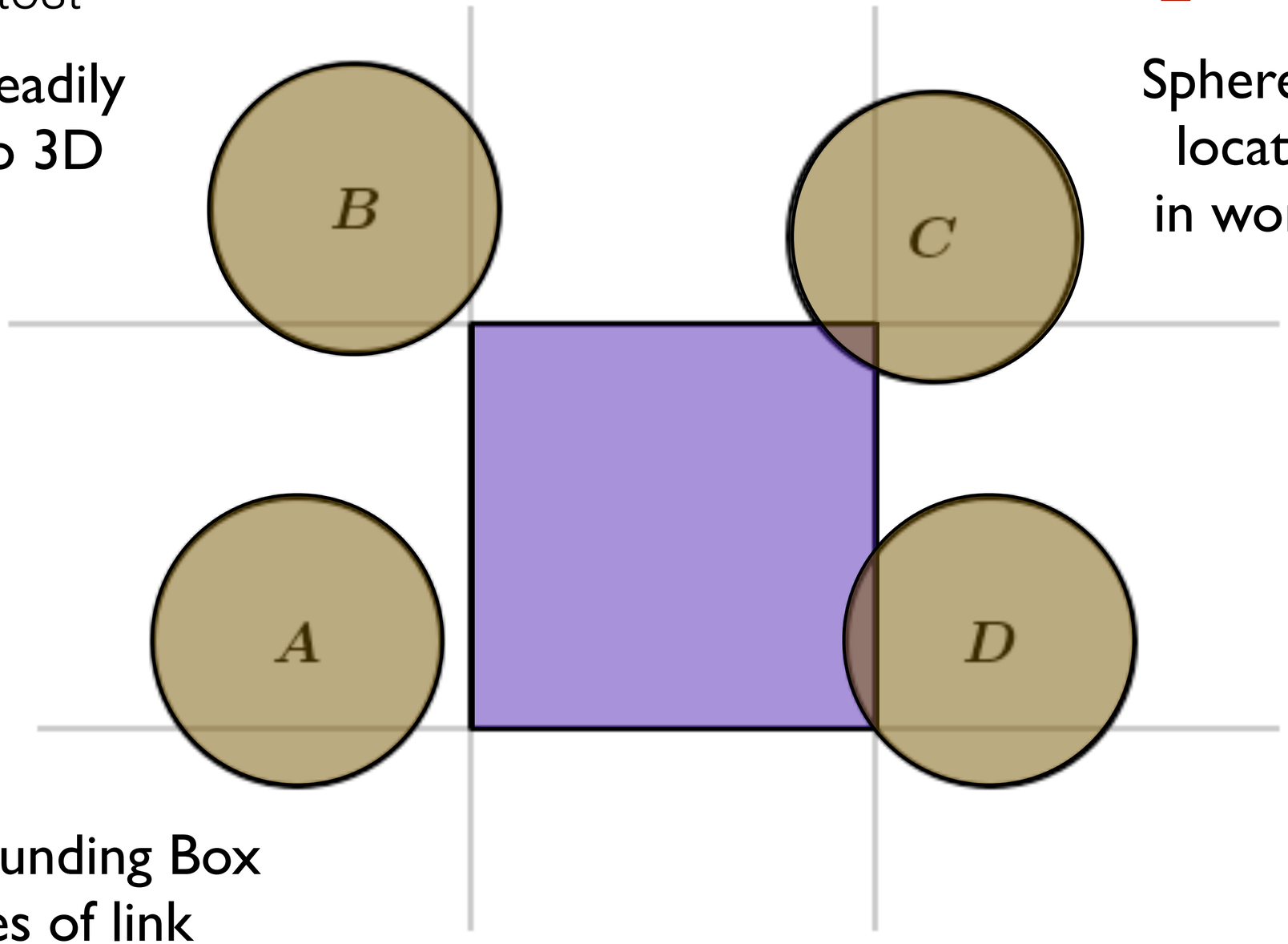
Consider AABB link tested against spherical obstacles in link frame



Sphere-bbox test

`robot_obstacles[i]`

2D example readily generalizes to 3D



Sphere obstacles with location and radius in world coordinates

Axis Aligned Bounding Box in coordinates of link

```
robot.links[x].bbox = [[x_min,y_min,z_min], [x_max,y_max,z_max]]
```

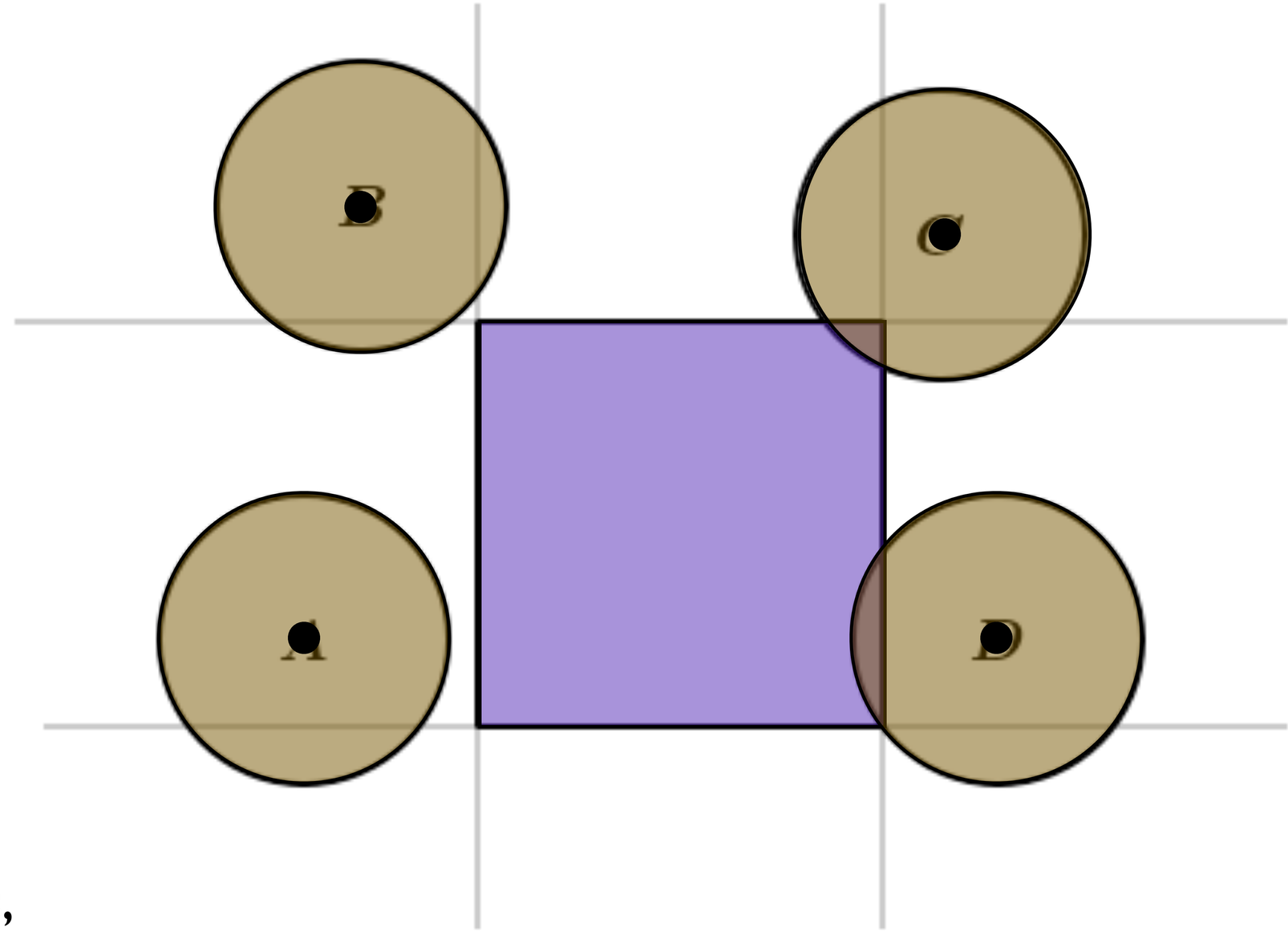
Sphere-bbox test

If sphere separable from AABB in any dimension, return no collision

`loc_y-radius>y_max?`

`loc_y+radius<y_min?`

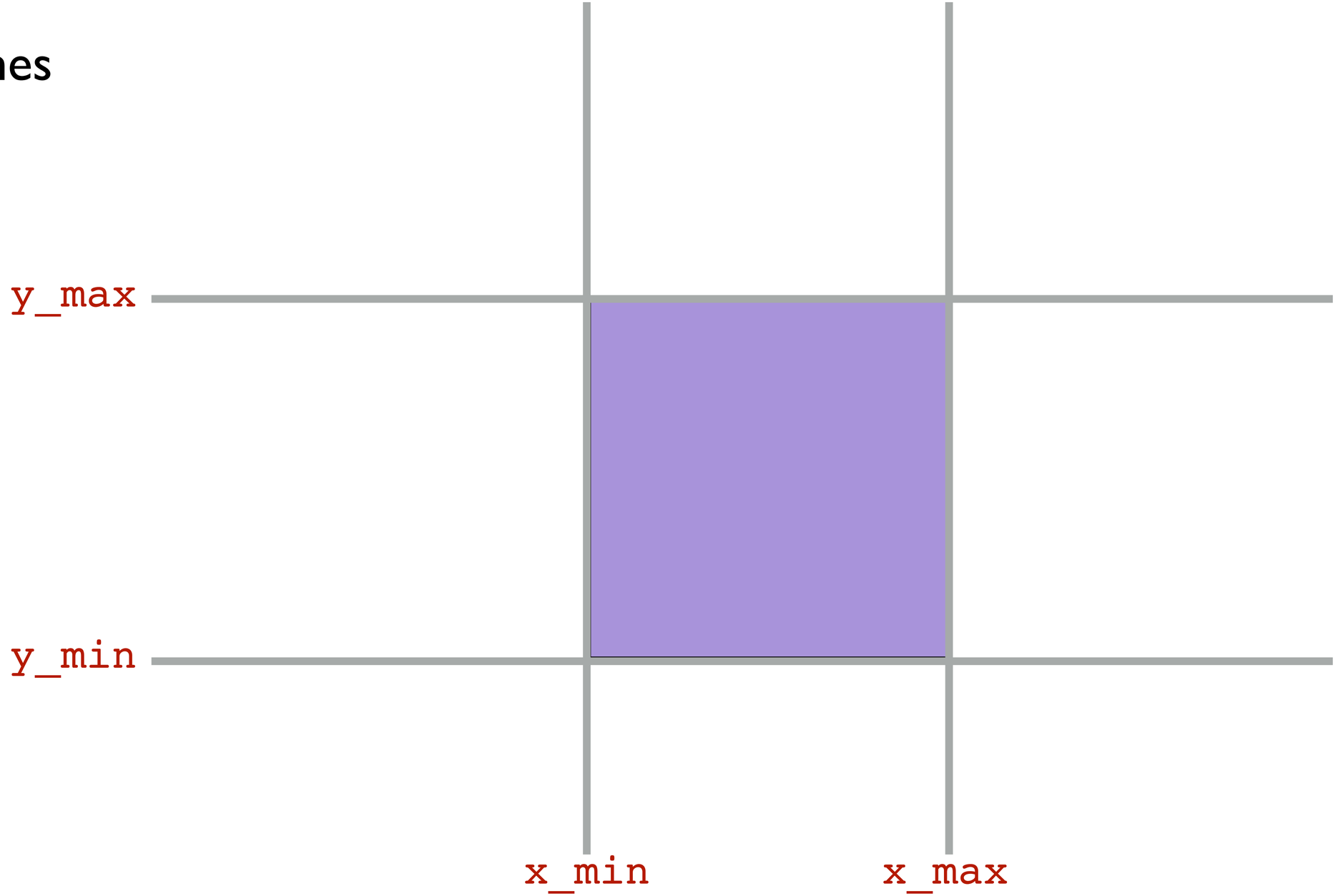
If sphere collides on all tests, return collision



`loc_x+radius<x_min?`

`loc_x-radius>x_max?`

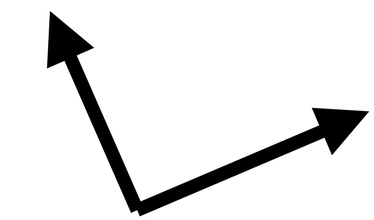
Separating planes



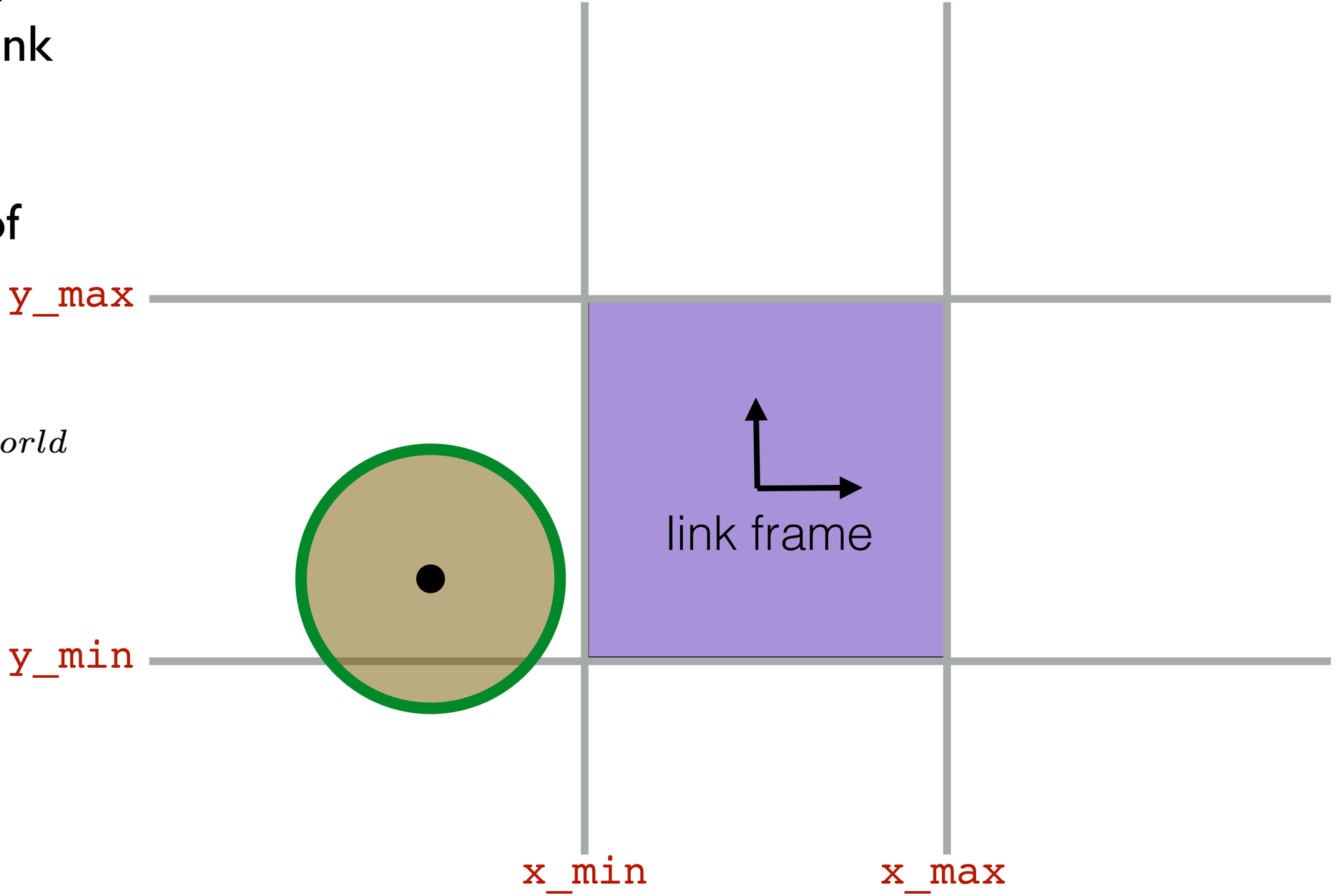
Transform centers of sphere obstacles into link coordinates

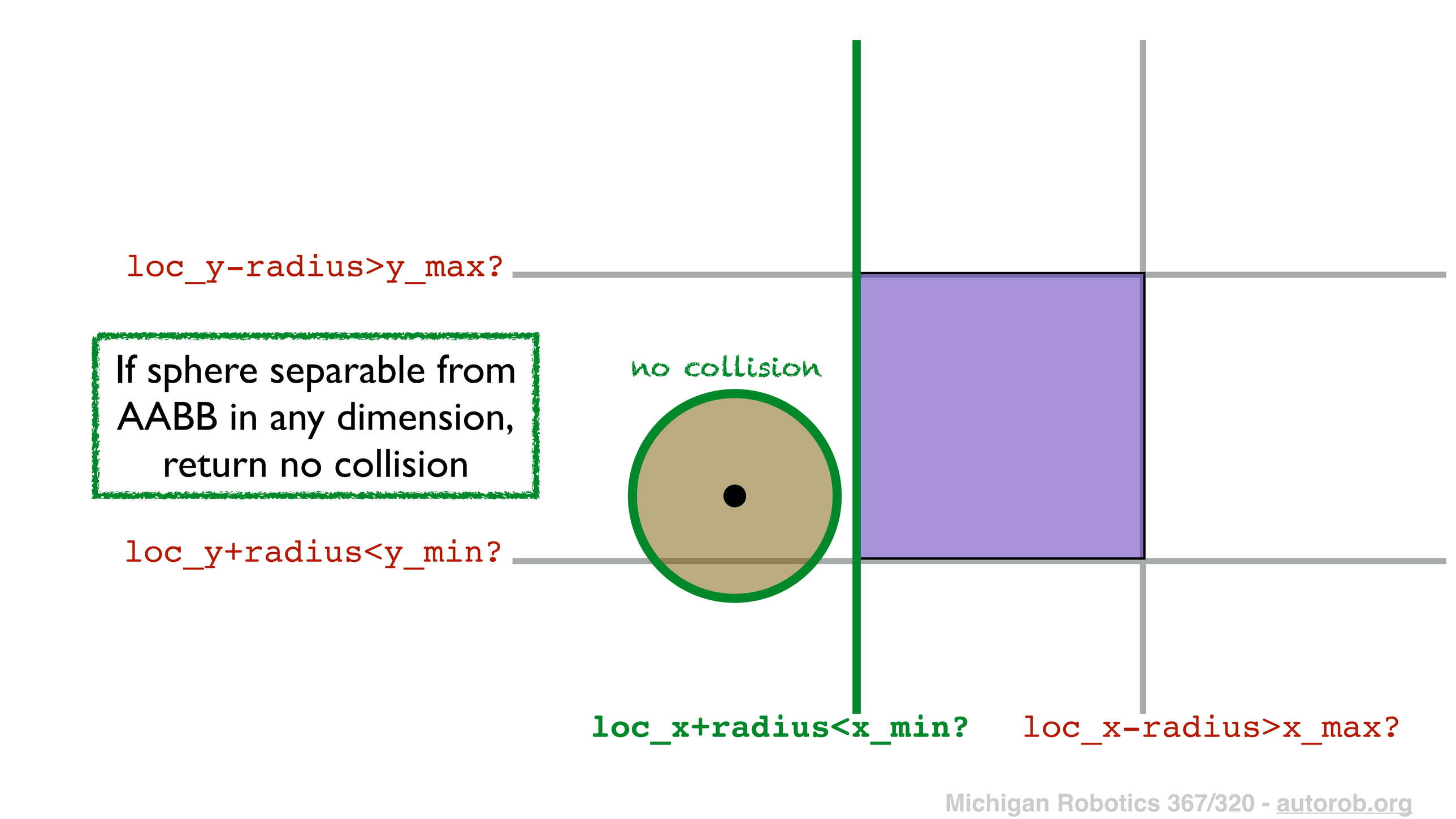
(Remember inverse of homogeneous transform?)

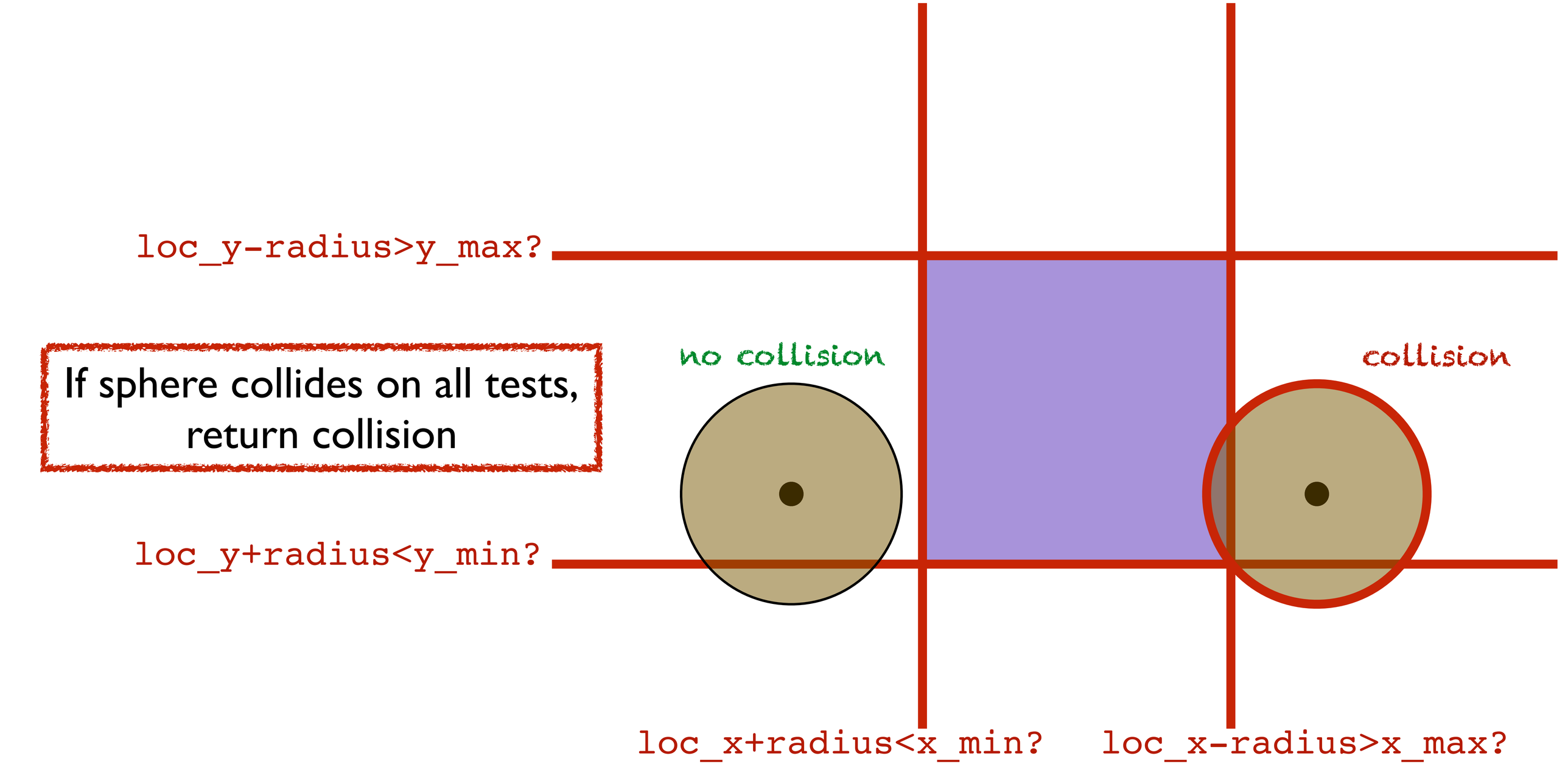
$$p^{link} = (T_{link}^{world})^{-1} p^{world}$$

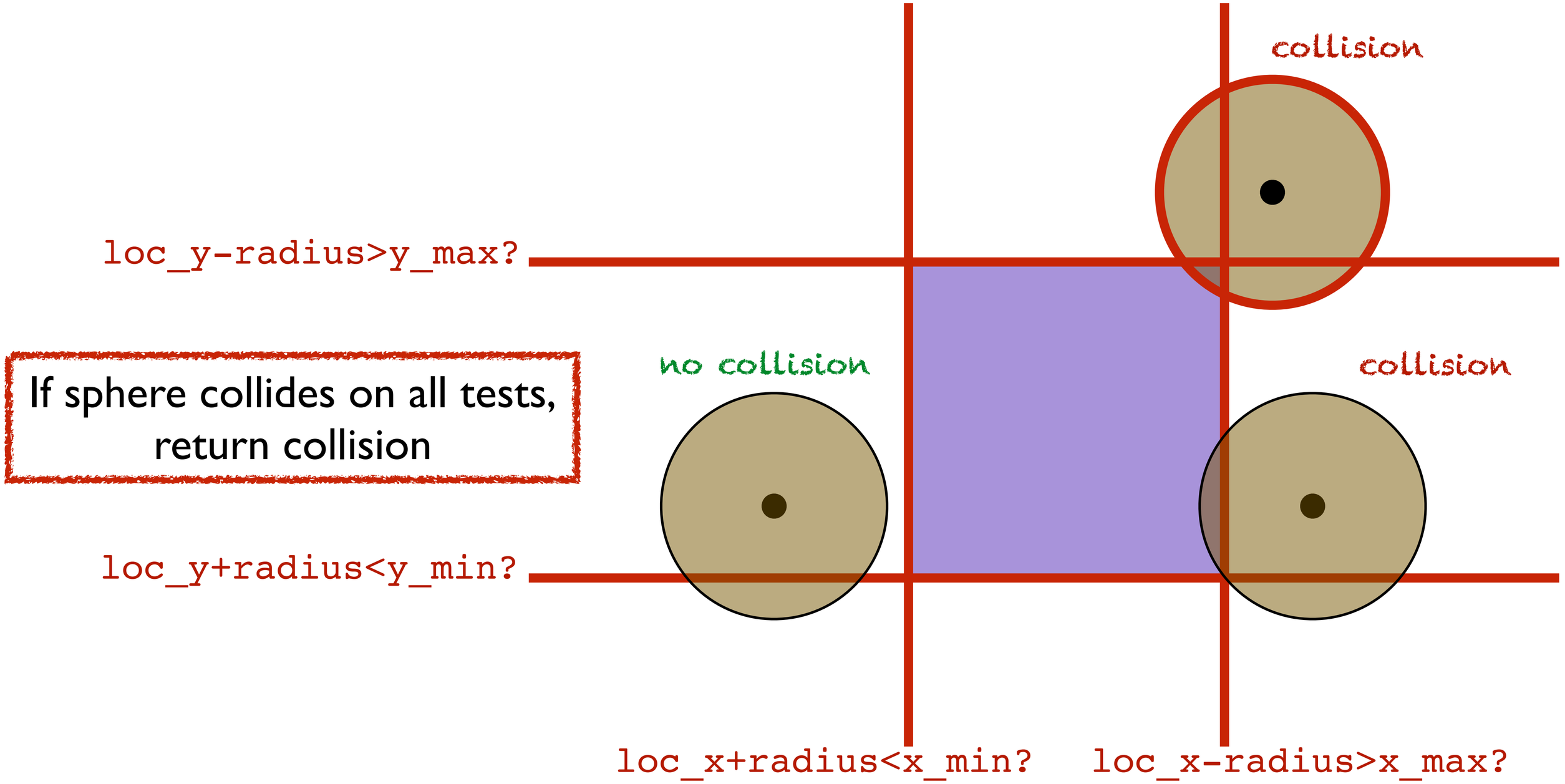


world frame

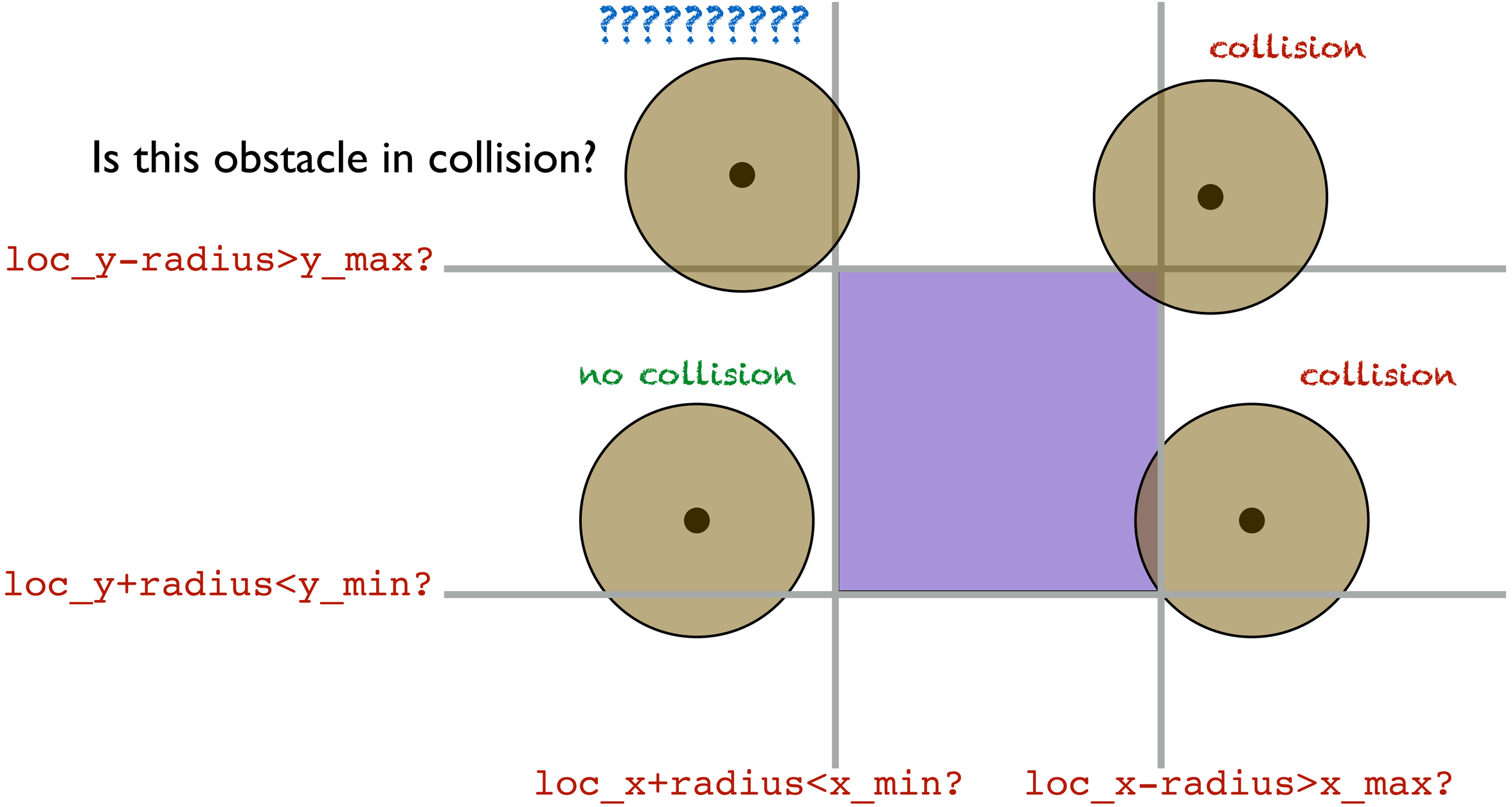



loc_y-radius>y_max?
If sphere separable from AABB in any dimension, return no collision
no collision
loc_y+radius<y_min?
loc_x+radius<x_min?
loc_x-radius>x_max?

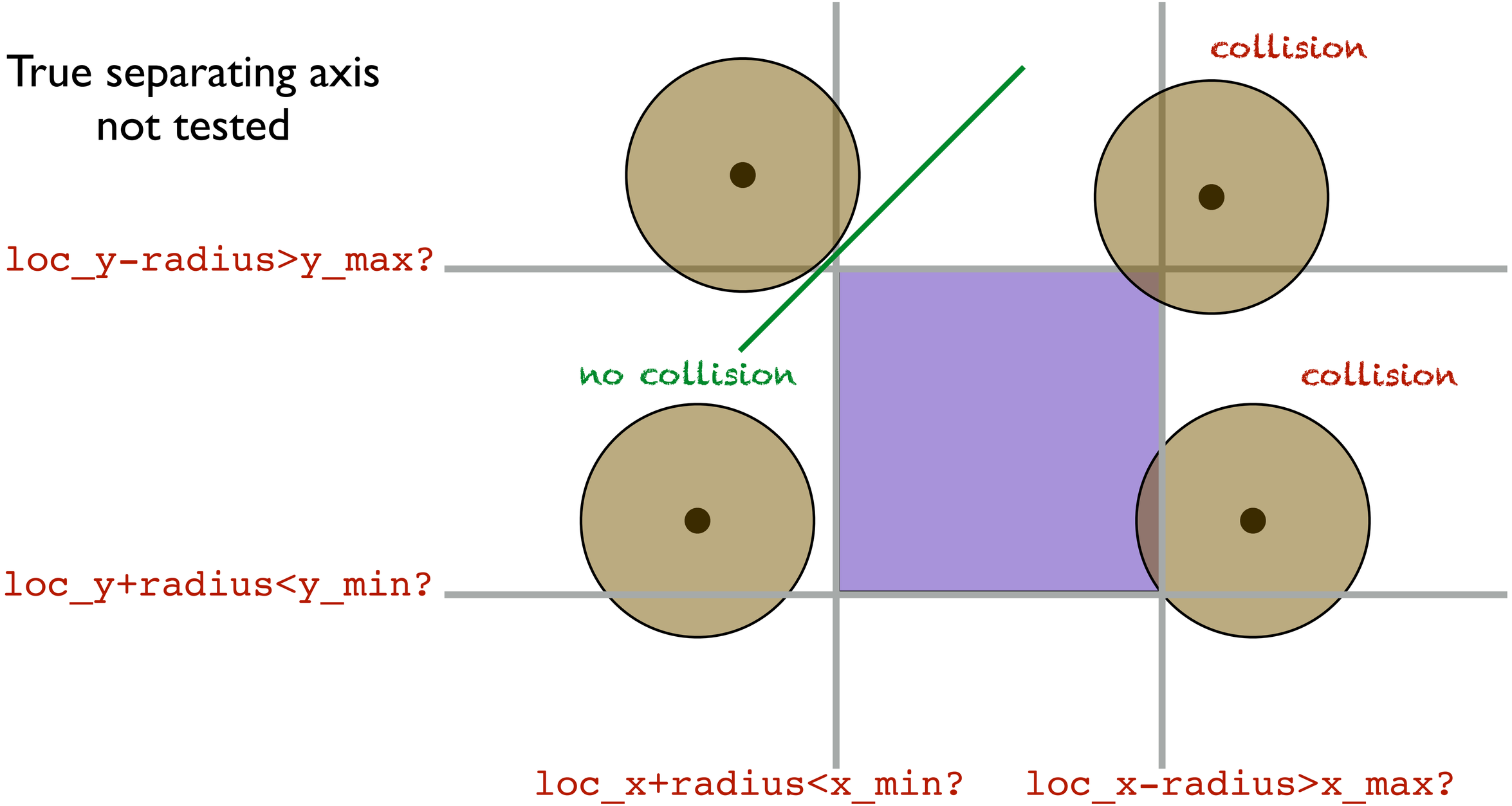

loc_y-radius>y_max?
If sphere collides on all tests, return collision
no collision
collision
loc_y+radius<y_min?
loc_x+radius<x_min?
loc_x-radius>x_max?

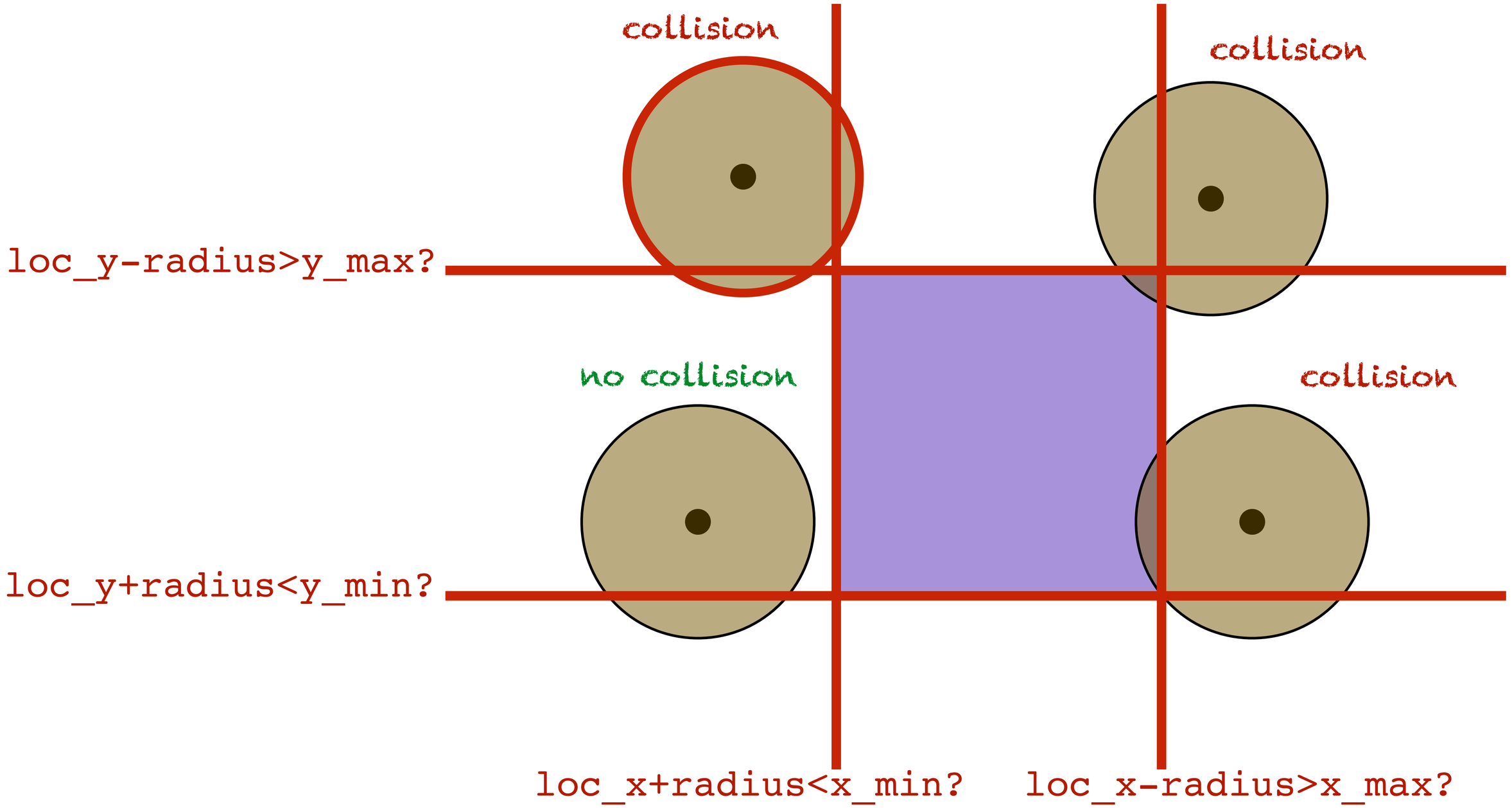

collision
loc_y-radius>y_max?
If sphere collides on all tests, return collision
no collision
collision
loc_y+radius<y_min?
loc_x+radius<x_min?
loc_x-radius>x_max?

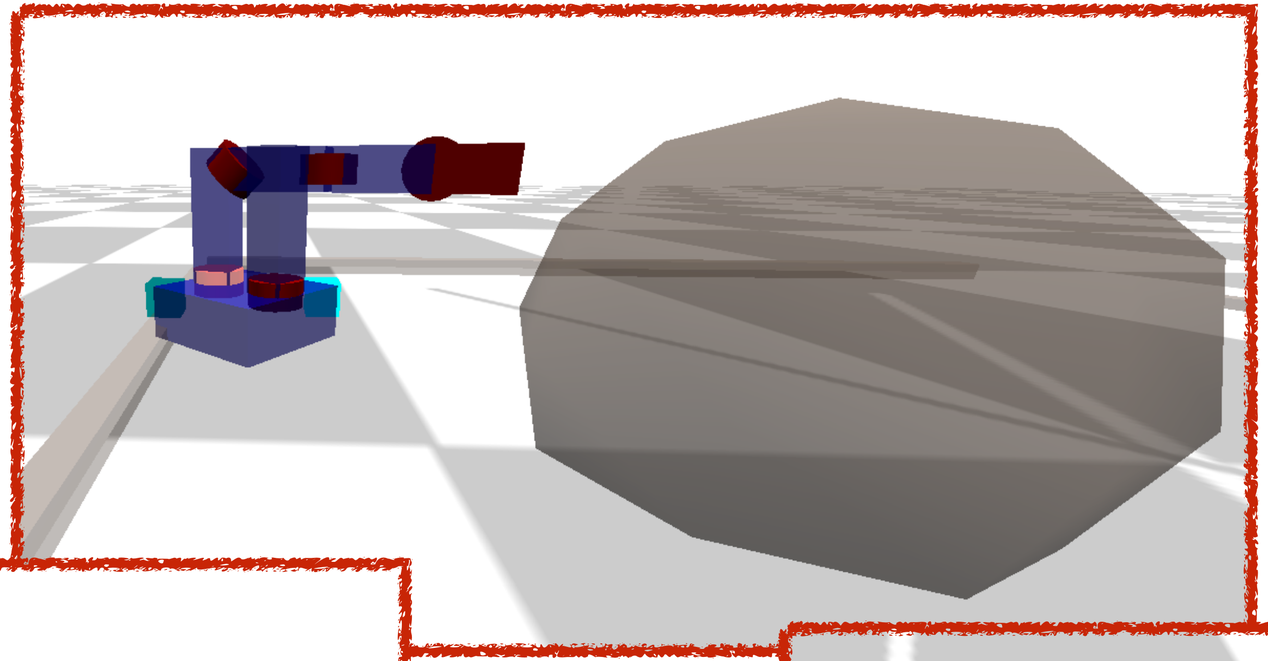
Is this obstacle in collision?

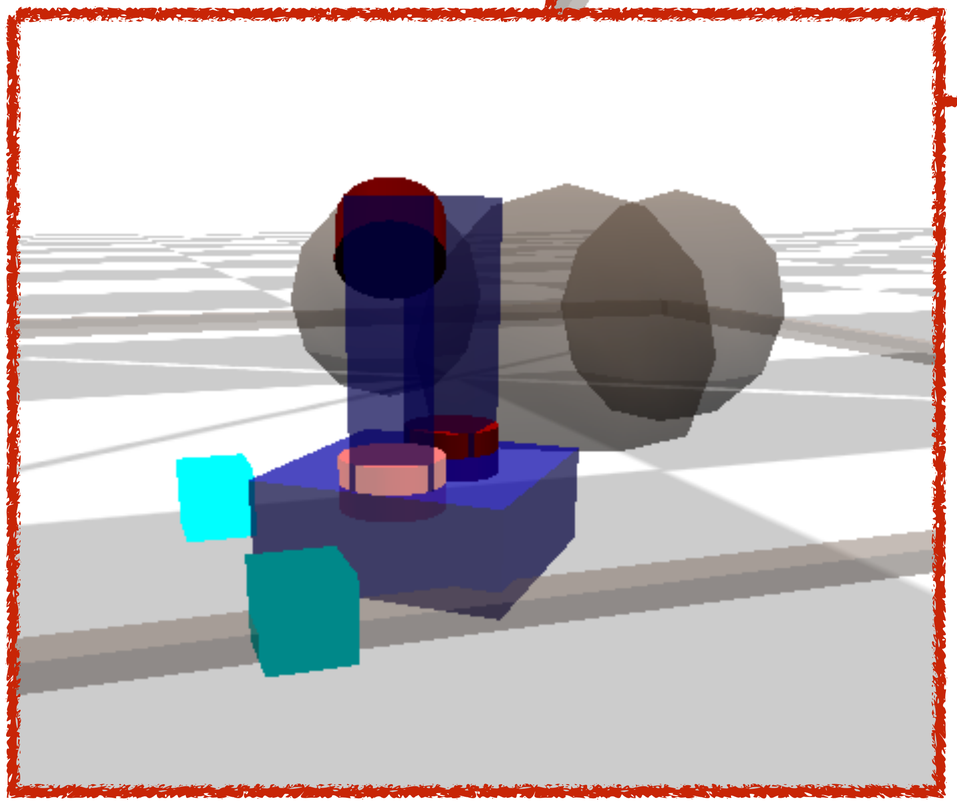


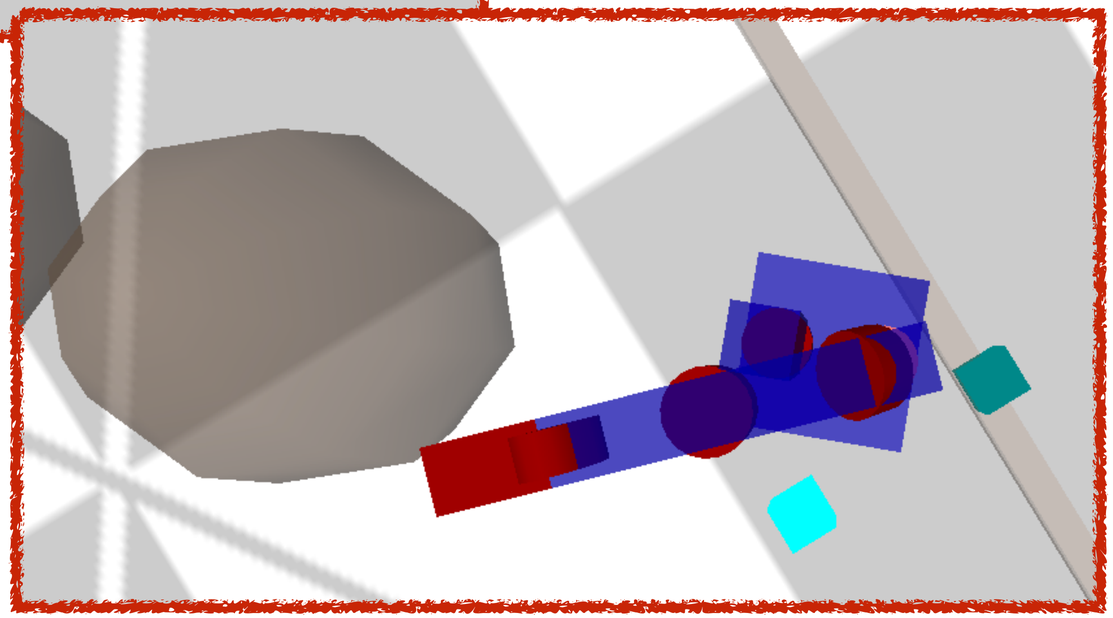
True separating axis not tested




collision
collision
loc_y-radius>y_max?
no collision
collision
loc_y+radius<y_min?
loc_x+radius<x_min?
loc_x-radius>x_max?



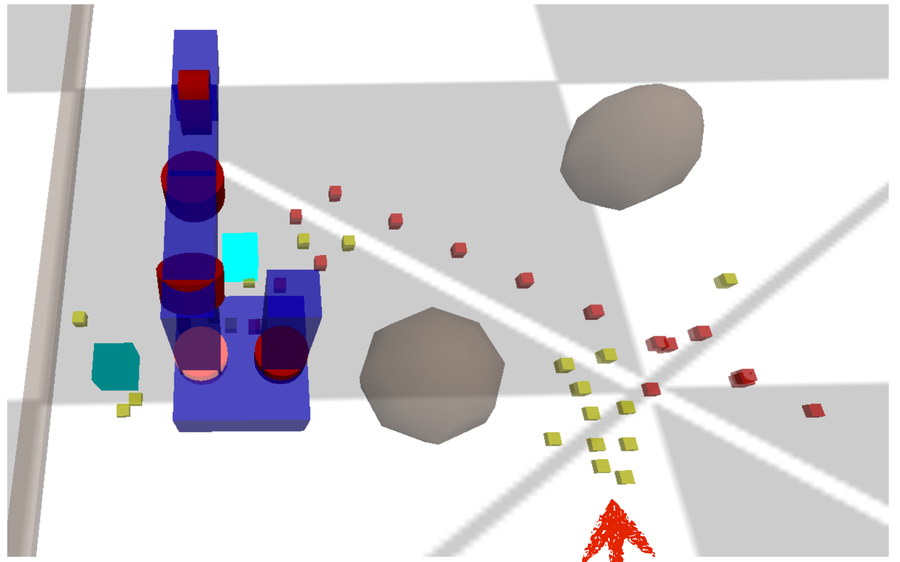




Last notes about planning visualization

kineval_rrt_connect.js

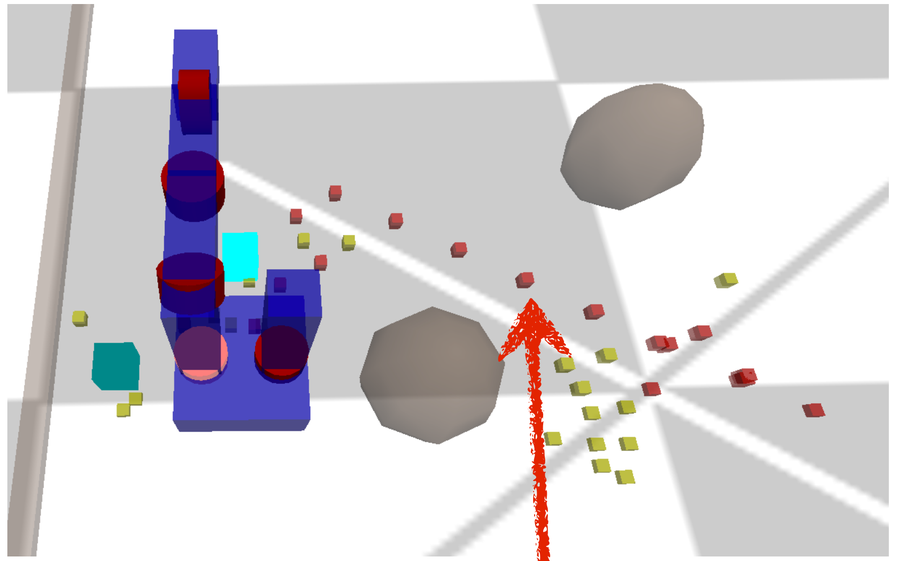
```
function tree_init(q) {

    // create tree object
    var tree = {};

    // initialize with vertex for given configuration
    tree.vertices = [];
    tree.vertices[0] = {};
    tree.vertices[0].vertex = q;
    tree.vertices[0].edges = [];

    // create rendering geometry for base location of vertex configuration
    add_config_origin_indicator_geom(tree.vertices[0]);

    // maintain index of newest vertex added to tree
    tree.newest = 0;

    return tree;
}
```



creates "geom" property of tree vertex with cube at base location for explored tree configuration

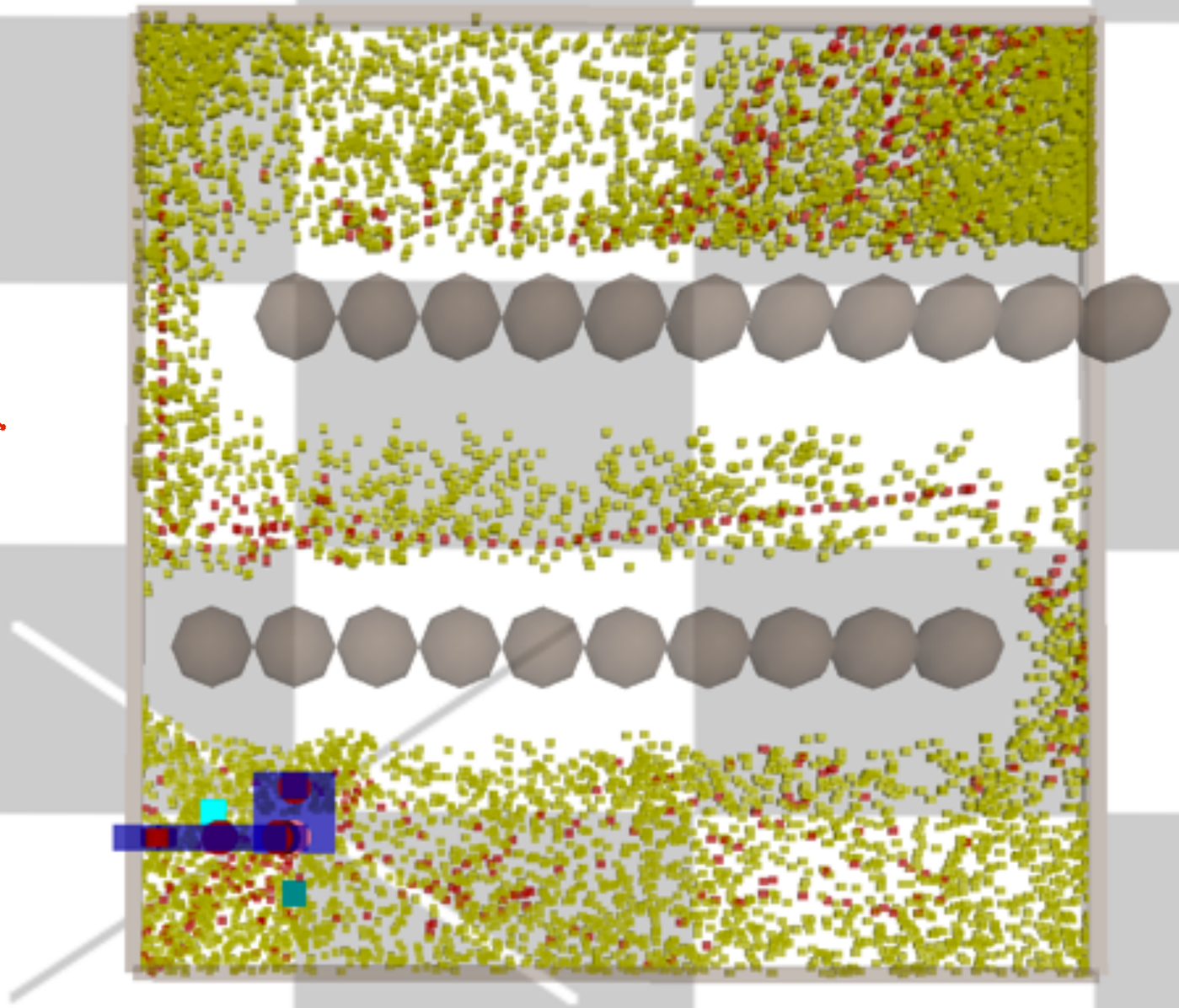
kineval_rrt_connect.js

```
    for (i=0;i<robot_path.length;i++) {
         robot_path[i].geom.material.color = {r:1,g:0,b:0};
     }
```



found motion path highlighted in red with this code

make sure to test against all provided worlds!




planner execution complete
kineval
just_starting
User Parameters
Robot
Forward Kinematics
Inverse Kinematics
Motion Planning
Display
Close Controls
make sure to test against all provided worlds!