AutoRob

Introduction to Autonomous Robotics
Michigan EECS 367

Robot Kinematics and Dynamics
Michigan ME 567 EECS 567 ROB 510

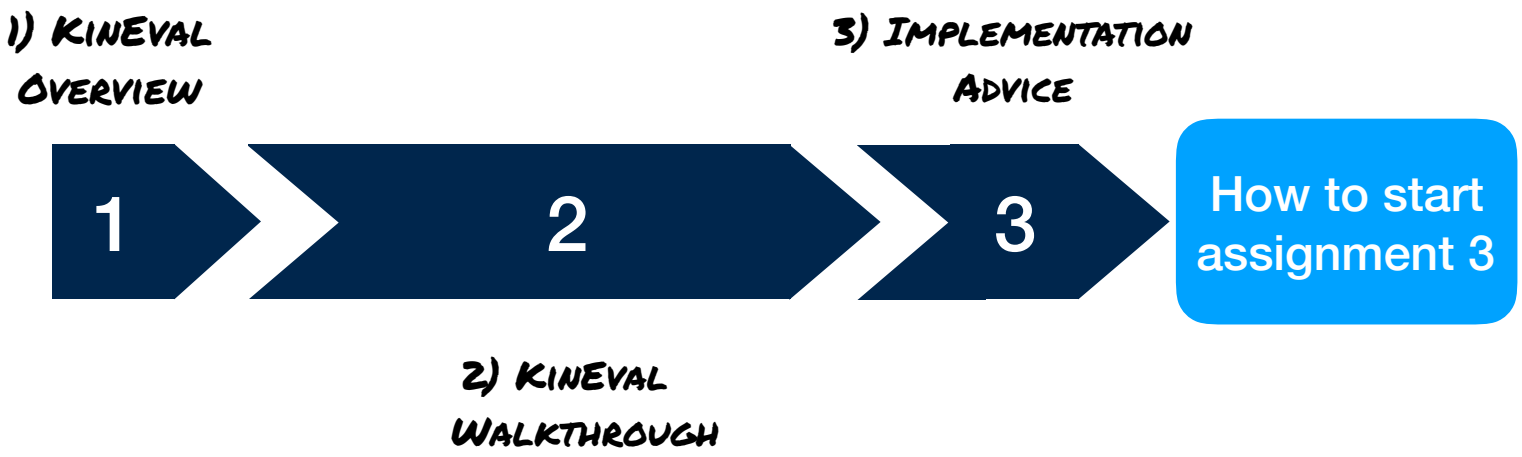
Fall 2019

EECS 367 Lab:
KinEval and urdf.js code overview

Administrative

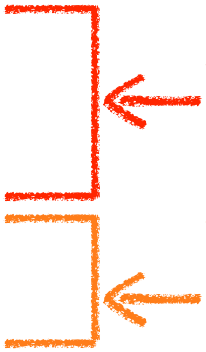
- Quiz #2: Monday, October 7
- Assignment #3: Forward Kinematics
  - Due 11:59pm, October 16

Lab Takeaways



Forward Kinematics Overview

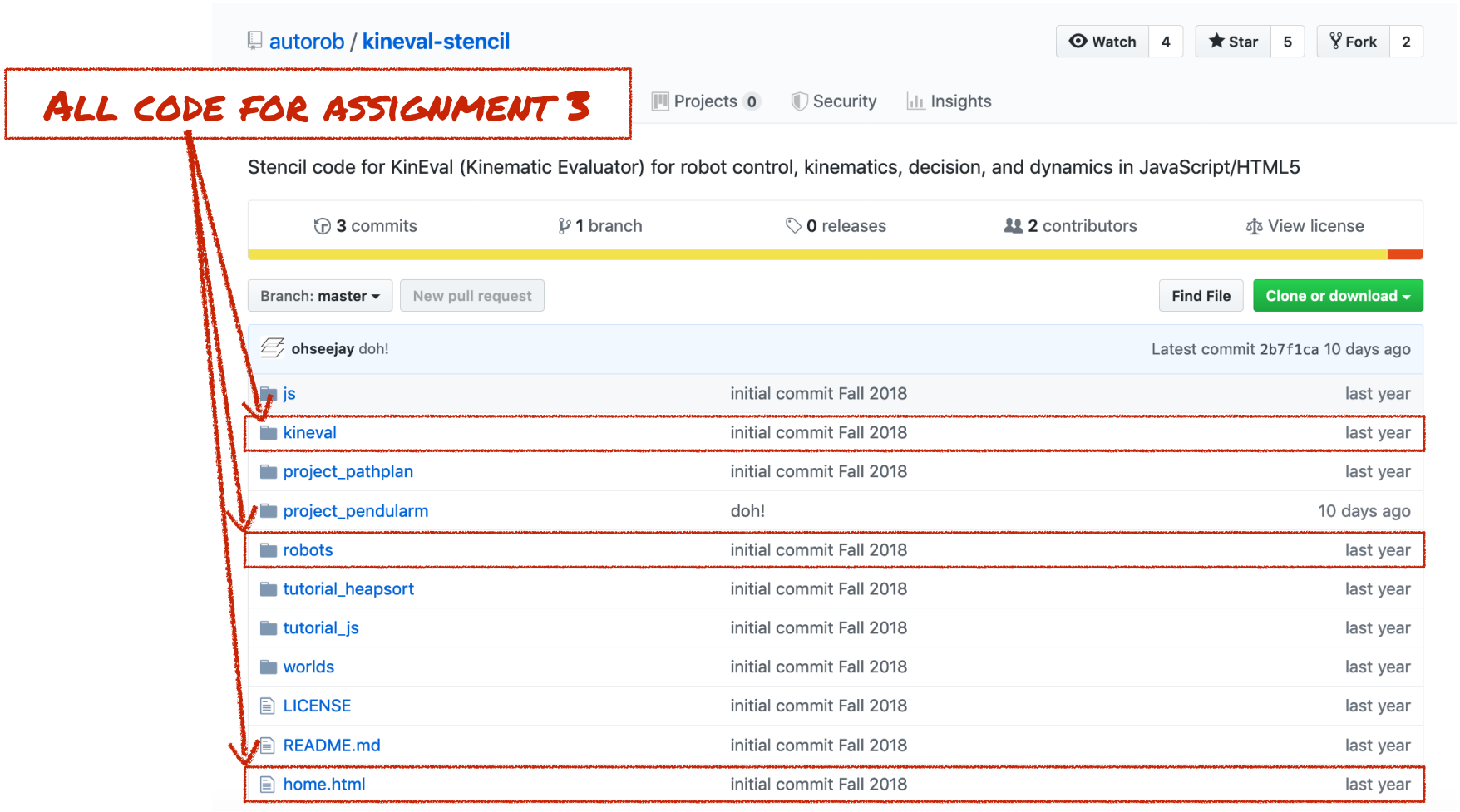
Assignment 3: Forward Kinematics		
2	All	Core matrix routines
8	All	FK transforms
2	All	Joint selection/rendering
2	Grad	Base offset transform
4	Grad	New robot definition



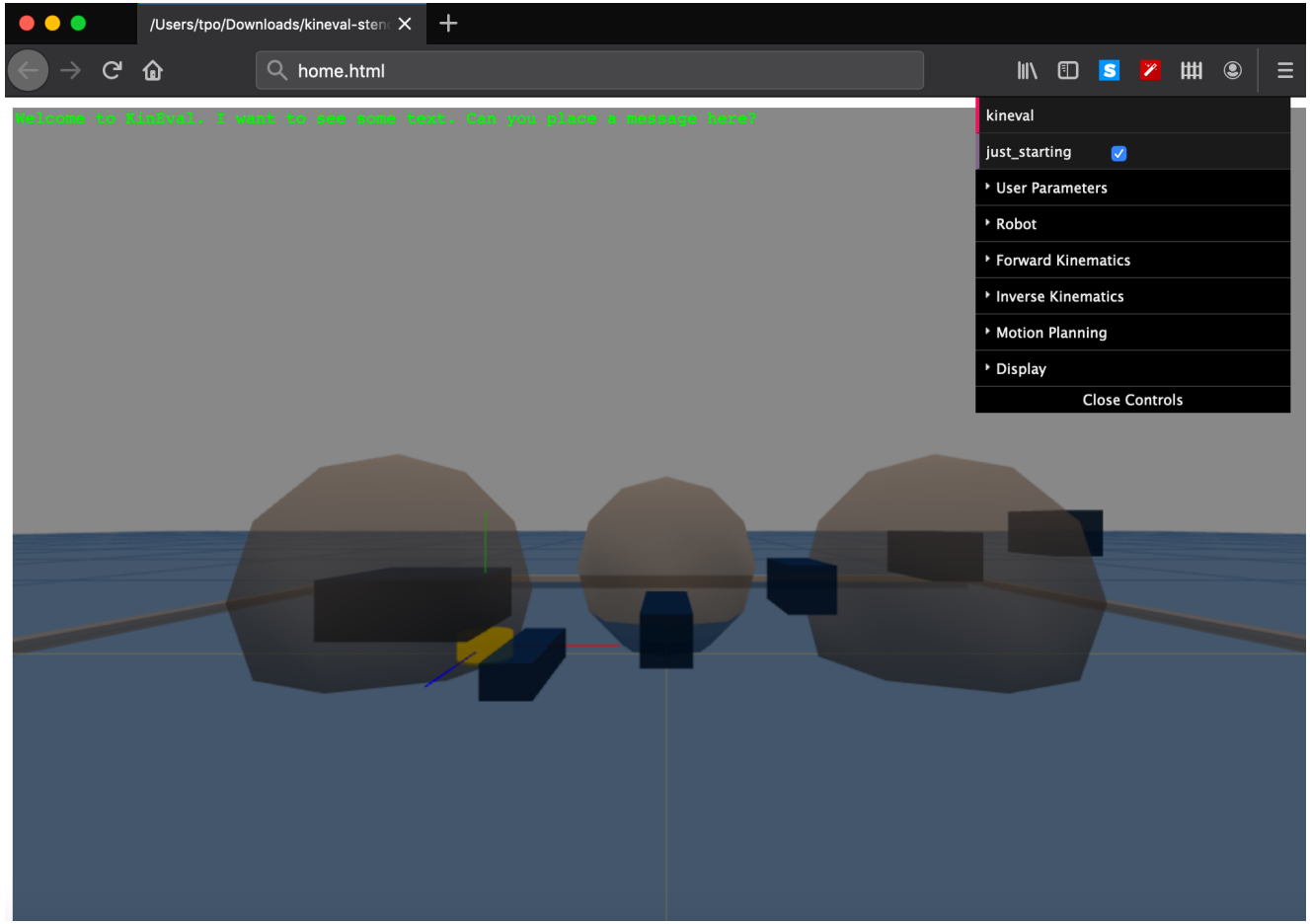
Features assigned to all sections

Features assigned to graduate sections

KinEval Overview



home.html


/Users/tpo/Downloads/kineval-sten
home.html
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
Close Controls

home.html

home.html

```
61
62  <!-- kineval includes -->
63  <script src="kineval/kineval.js"></script>
64  <script src="kineval/kineval_startingpoint.js"></script>
65  <script src="kineval/kineval_robot_init.js"></script>
66  <script src="kineval/kineval_threejs.js"></script>
67  <script src="kineval/kineval_userinput.js"></script>
68
69  <!-- kineval FK/drawing -->
70  <script src="kineval/kineval_forward_kinematics.js"></script>
71  <script src="kineval/kineval_matrix.js"></script>
72  <script src="kineval/kineval_quaternion.js"></script>
73
74  <!-- kineval FK/joint control -->
75  <script src="kineval/kineval_controls.js"></script>
76  <script src="kineval/kineval_servo_control.js"></script>
77
78  <!-- kineval IK -->
79  <script src="kineval/kineval_inverse_kinematics.js"></script>
80
81  <!-- kineval motion planning -->
82  <script src="kineval/kineval_rrt_connect.js"></script>
83  <script src="kineval/kineval_collision.js"></script>
84
85  <!-- kineval experimental rosbridge/ROS for connectivity to a real robot -->
86  <script type="text/javascript" src="js/eventemitter2.min.js"></script>
87  <script type="text/javascript" src="js/roslib.min.js"></script>
88  <script src="kineval/kineval_rosbridge.js"></script>
89
90  <!-- KE: not supported point cloud as JSON version of PCD format -->
91  <!--
92  <script src="experimental/point_clouds/three_objects.pcd.js"></script>
93  <script src="experimental/point_clouds/eight_objects.pcd.js"></script>
94  -->
```

KinEval source files must be included

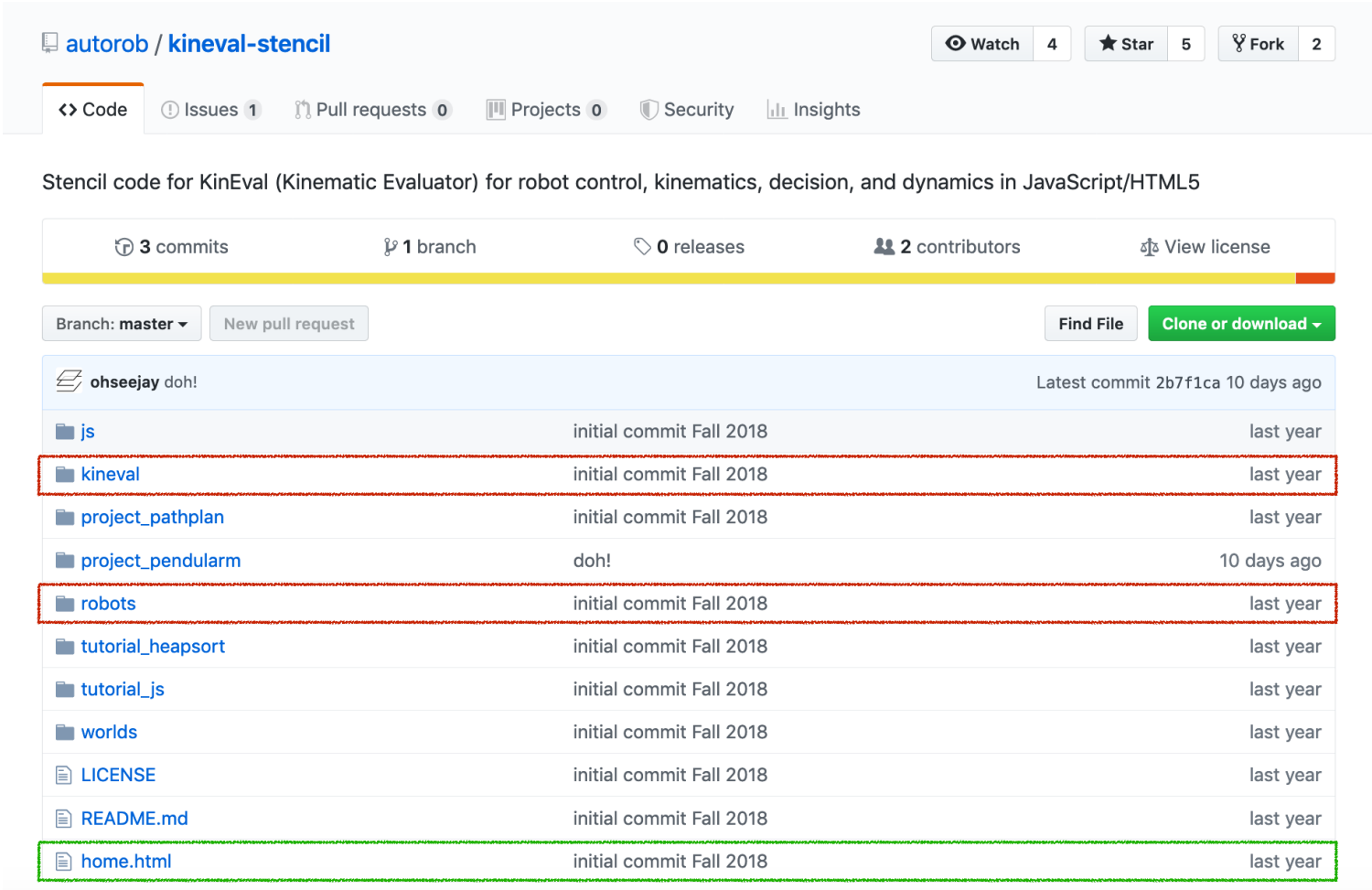
home.html

home.html

```
149  // STUDENT: my_animate is where your robot's controls and movement are updated over
150  function my_animate() {
151
152      // set to starting point mode is true as default (initialized in kineval.js)
153      //   set to false once starting forward kinematics project
154      //kineval.params.just_starting = false;
155
156      if (kineval.params.just_starting == true) {
157          startingPlaceholderAnimate();
158          kineval.robotDraw();
159          return;
160      }
161
162      // ROBOT DYNAMICS
163
164      // update robot configuration from applied robot controls
165      //   (assuming pure kinematics for now)
166      kineval.applyControls(robot);
167
168      // HANDLE USER CONTROLS
169
170      // handle user input
171      kineval.handleUserInput();
172
173      // perform forward kinematics placing robot links in space wr
174      kineval.robotForwardKinematics();
175
176      // determine if robot is currently in collision with world
177      kineval.robotIsCollision();
178
179      // render robot and world in 3D scene
180      kineval.robotDraw();
181
182      // if requested, perform setpoint control with joint servo controllers
183      kineval.setpointClockMovement(); // simple clock tick movement
184      kineval.setpointDanceSequence(); // dance routine
185      kineval.robotArmControllerSetpoint(); // PID controller
186      kineval.params.update_pd = false; // clear PID request for next iteration
187
188      // if requested, perform inverse kinematics control to reach to point
189      kineval.robotInverseKinematics(kineval.params.ik_target, robot.endeffector.frame
             position);
190
191      // if requested, perform configuration space motion planning to home pose
192      kineval.planMotionRRTConnect();
193  }
```

my_animate() WILL BE CALLED AT EVERY ANIMATION FRAME

KinEval Overview



autorob / kineval-stencil

Watch 4 | Star 5 | Fork 2

Code | Issues 1 | Pull requests 0 | Projects 0 | Security | Insights

Stencil code for KinEval (Kinematic Evaluator) for robot control, kinematics, decision, and dynamics in JavaScript/HTML5

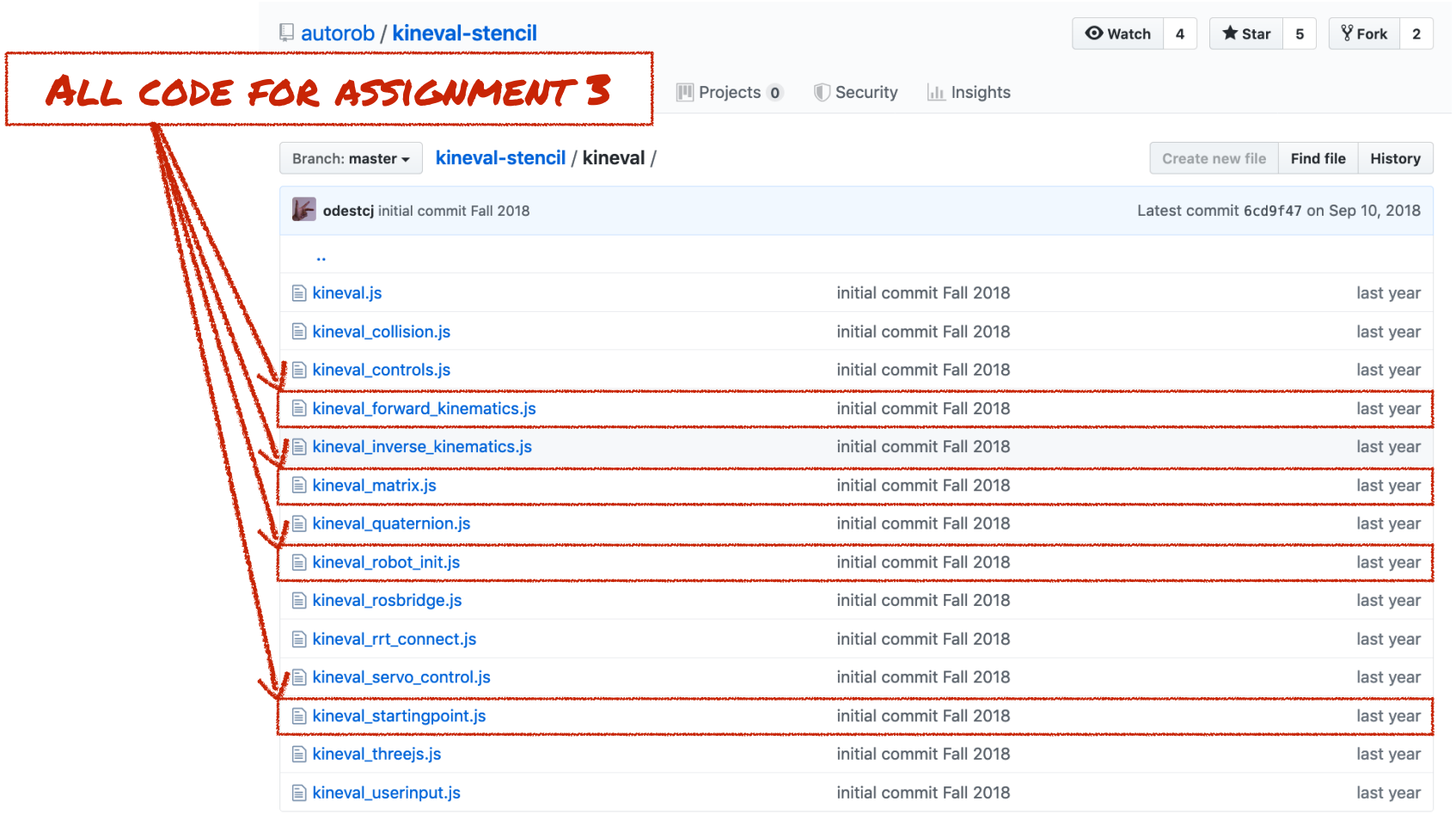
3 commits | 1 branch | 0 releases | 2 contributors | View license

Branch: master | New pull request | Find File | Clone or download

ohseejay doh! — Latest commit 2b7f1ca 10 days ago

js	initial commit Fall 2018	last year
kineval	initial commit Fall 2018	last year
project_pathplan	initial commit Fall 2018	last year
project_pendularm	doh!	10 days ago
robots	initial commit Fall 2018	last year
tutorial_heapsort	initial commit Fall 2018	last year
tutorial_js	initial commit Fall 2018	last year
worlds	initial commit Fall 2018	last year
LICENSE	initial commit Fall 2018	last year
README.md	initial commit Fall 2018	last year
home.html	initial commit Fall 2018	last year

KinEval Overview



kineval_startingpoint.js

As the name suggests, this file is meant to build your comfort with the source code

kineval_startingpoint.js

```
365  kineval.startingPlaceholderUserInput = function startingPlaceholderUserInput() {
366
367      /* keyboard is a threejs helper object for reading keyboard state.
368          keyboard.pressed() will return true if a particular key is being
369          pressed, without the need for a callback event handler
370      */
371      if (keyboard.pressed("shift+x")) {
372          textbar.innerHTML = "come on down";  // make the objects move down
373      // STENCIL: update the vertical offset variable
374      }
375      else if (keyboard.pressed("x")) {
376          textbar.innerHTML = "moving on up";  // make the objects move up
377      // STENCIL: update the vertical offset variable
378      }
379      else if (keyboard.pressed("shift+z")) {
380          // increase the jittering of the objects
381          textbar.innerHTML = "its time for the percolator";
382      // STENCIL: update the radius of the jittering
383      }
384      else if (keyboard.pressed("z")) {
385          // decrease the jittering of the objects
386          textbar.innerHTML = "relax your mind, let your conscience be free";
387      // STENCIL: update the radius of the jittering
388      }
389      else if (keyboard.pressed("shift+1")) {
390          // increase spacing along the x-axis between the objects
391          textbar.innerHTML = "sail away";
392      // STENCIL: update the global spacing variable
393      }
394      else if (keyboard.pressed("1")) {
395          // decrease spacing along the x-axis between the objects
396          textbar.innerHTML = "come together";
397      // STENCIL: update the global spacing variable
398      }
399      else if (keyboard.pressed("shift+2")) {
400          // increase the amplitude of the animating sine wave
401          textbar.innerHTML = "ain't no mountain high enough";
402      // STENCIL: update the wave amplitude variable
403      }
404      else if (keyboard.pressed("2")) {
405          // decrease the amplitude of the animating sine wave
406          textbar.innerHTML = "got 'til it's gone";
407      // STENCIL: update the wave amplitude variable
408      }
```

Light implementation exercises for controlling webpage marked with 'STENCIL'

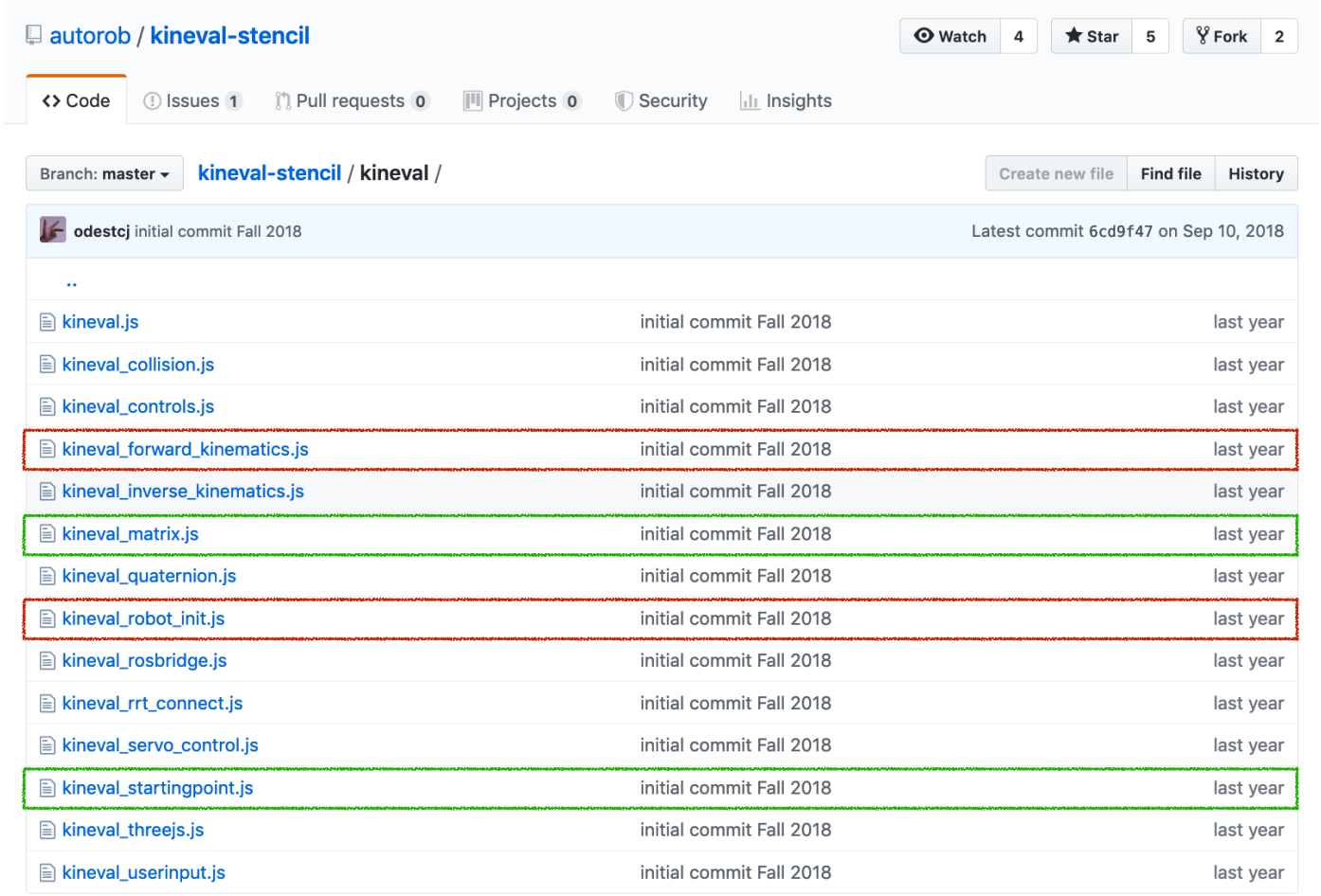
kineval_matrix.js

kineval_matrix.js

```
1  ////////////////////////////////////////////////
2  /////     MATRIX ALGEBRA AND GEOMETRIC TRANSFORMS
3  ////////////////////////////////////////////////
4
5  function matrix_copy(m1) {
6      // returns 2D array that is a copy of m1
7
8      var mat = [];
9      var i,j;
10
11     for (i=0;i<m1.length;i++) { // for each row of m1
12         mat[i] = [];
13         for (j=0;j<m1[0].length;j++) { // for each column of m1
14             mat[i][j] = m1[i][j];
15         }
16     }
17     return mat;
18 }
19
21     // STENCIL: reference matrix code has the following functions:
22     //   matrix_multiply
23     //   matrix_transpose
24     //   matrix_pseudoinverse
25     //   matrix_invert_affine
26     //   vector_normalize
27     //   vector_cross
28     //   generate_identity
29     //   generate_translation_matrix
30     //   generate_rotation_matrix_X
31     //   generate_rotation_matrix_Y
32     //   generate_rotation_matrix_Z
```

Matrix operations you will eventually need to implement

KinEval Overview



kineval_forward_kinematics.js

Your recursive traversal of links and joints to build up matrix stack

kineval_forward_kinematics.js

```
/*-- |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/|

    KinEval | Kinematic Evaluator | forward kinematics

    Implementation of robot kinematics, control, decision making, and dynamics
        in HTML5/JavaScript and threejs

    @author ohseejay / https://github.com/ohseejay / https://bitbucket.org/ohseejay

    Chad Jenkins
    Laboratory for Perception RObotics and Grounded REasoning Systems
    University of Michigan

    License: Creative Commons 3.0 BY-SA

 |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| |\/| --*/

kineval.robotForwardKinematics = function robotForwardKinematics () {

    if (typeof kineval.buildFKTransforms === 'undefined') {
        textbar.innerHTML = "forward kinematics not implemented";
        return;
    }

    // STENCIL: implement kineval.buildFKTransforms();

}

    // STENCIL: reference code alternates recursive traversal over
    //   links and joints starting from base, using following functions:
    //     traverseFKBase
    //     traverseFKLink
    //     traverseFKJoint
    //
    // user interface needs the heading (z-axis) and lateral (x-axis) directions
    //   of robot base in world coordinates stored as 4x1 matrices in
    //   global variables "robot_heading" and "robot_lateral"
    //
    // if geometries are imported and using ROS coordinates (e.g., fetch),
    //   coordinate conversion is needed for kineval/threejs coordinates:
    //
```

Your traversal should result with updated `.xform`'s of each link and joint

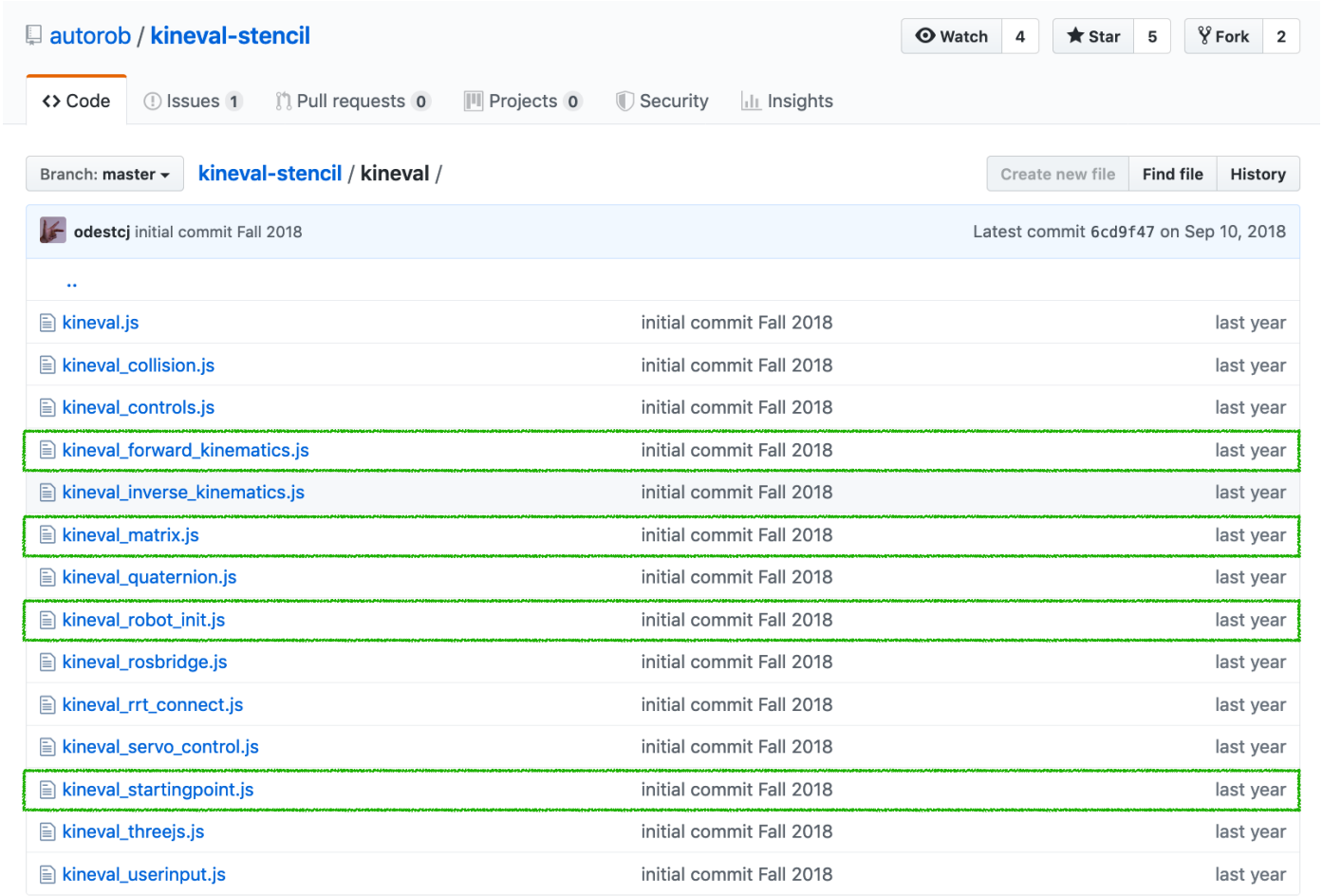
kineval_robot_init.js

kineval_robot_init.js

```
25
26  kineval.initRobotLinks = function initRobotLinks() {
27
28      for (x in robot.links) {
29          robot.links[x].name = x;
30      }
31
32      // initialize controls for robot base link
33      robot.control = {xyz: [0,0,0], rpy:[0,0,0]};
34  }
35
36  kineval.initRobotJoints = function initRobotJoints() {
37      // build kinematic hierarchy by looping over each joint in the robot
38      //   (object fields can be index through array-style indices, object[field] = property)
39      //   and insert threejs scene graph (each joint and link are directly connect to scene root)
40      // NOTE: kinematic hierarchy is maintained independently by this code, not threejs
41
42      var x,tempmat;
43
44      for (x in robot.joints) {
45
46          // give the joint its name as an id
47          robot.joints[x].name = x;
48
49          // initialize joint angle value and control input value
50          robot.joints[x].angle = 0;
51          robot.joints[x].control = 0;
52          robot.joints[x].servo = {};
53      // STENCIL: set appropriate servo gains for arm setpoint control
54          robot.joints[x].servo.p_gain = 0;
55          robot.joints[x].servo.p_desired = 0;
56          robot.joints[x].servo.d_gain = 0;
57
58      // STENCIL: complete kinematic hierarchy of robot for convenience.
59      //   robot description only specifies parent and child links for joints.
60      //   additionally specify parent and child joints for each link
61
62      }
63
64  }
65
```

USE URDF OBJECT TO INITIALIZE ROBOT'S INTERNAL STRUCTURE BETWEEN LINKS/ JOINTS

KinEval Overview



autorob / kineval-stencil

Watch 4 | Star 5 | Fork 2

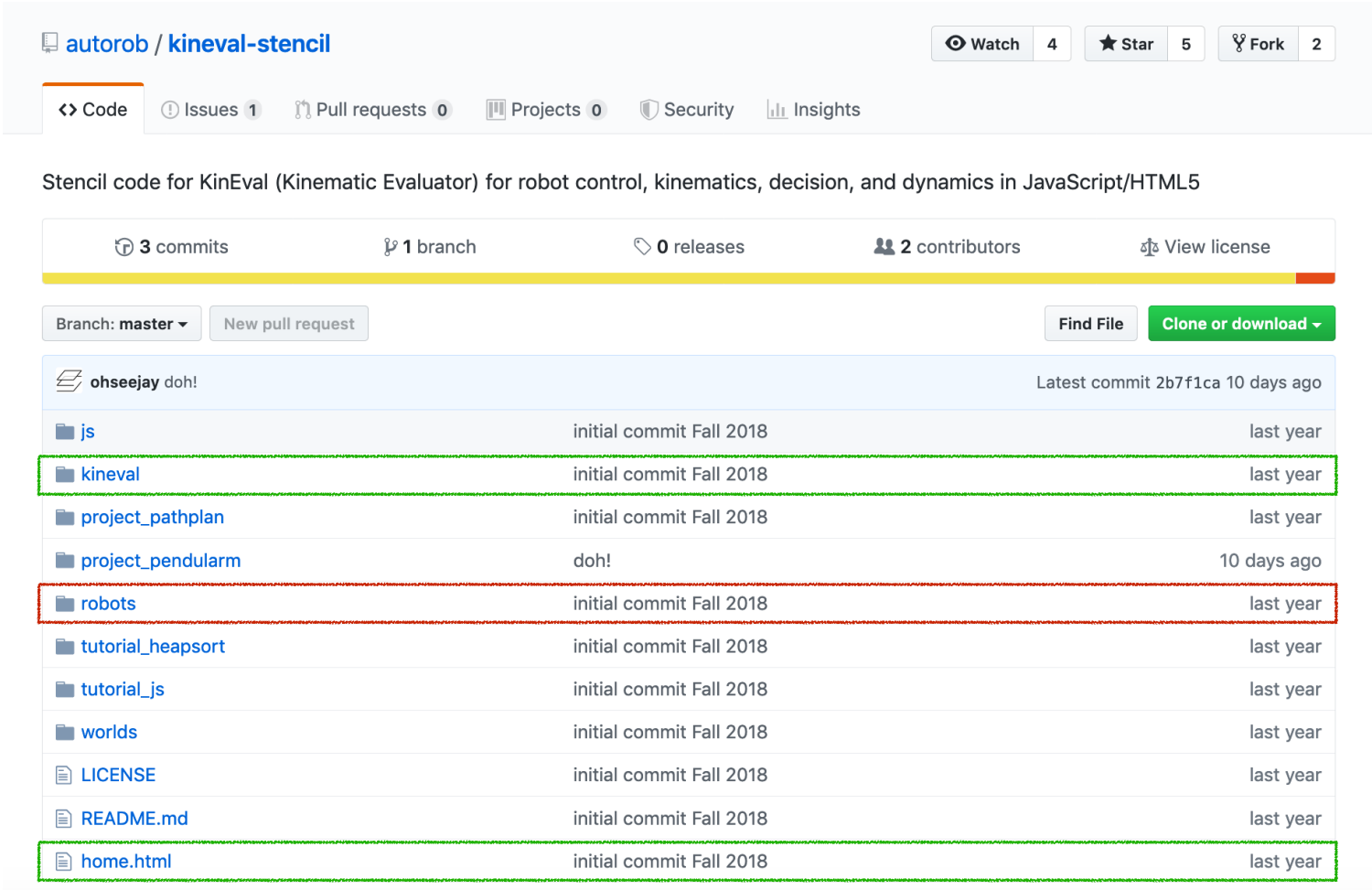
Code | Issues 1 | Pull requests 0 | Projects 0 | Security | Insights

Branch: master ▾ kineval-stencil / kineval / Create new file | Find file | History

odestcj initial commit Fall 2018 — Latest commit 6cd9f47 on Sep 10, 2018

..		
kineval.js	initial commit Fall 2018	last year
kineval_collision.js	initial commit Fall 2018	last year
kineval_controls.js	initial commit Fall 2018	last year
kineval_forward_kinematics.js	initial commit Fall 2018	last year
kineval_inverse_kinematics.js	initial commit Fall 2018	last year
kineval_matrix.js	initial commit Fall 2018	last year
kineval_quaternion.js	initial commit Fall 2018	last year
kineval_robot_init.js	initial commit Fall 2018	last year
kineval_rosbridge.js	initial commit Fall 2018	last year
kineval_rrt_connect.js	initial commit Fall 2018	last year
kineval_servo_control.js	initial commit Fall 2018	last year
kineval_startingpoint.js	initial commit Fall 2018	last year
kineval_threejs.js	initial commit Fall 2018	last year
kineval_userinput.js	initial commit Fall 2018	last year

KinEval Overview



autorob / kineval-stencil

Watch 4 | Star 5 | Fork 2

Code | Issues 1 | Pull requests 0 | Projects 0 | Security | Insights

Stencil code for KinEval (Kinematic Evaluator) for robot control, kinematics, decision, and dynamics in JavaScript/HTML5

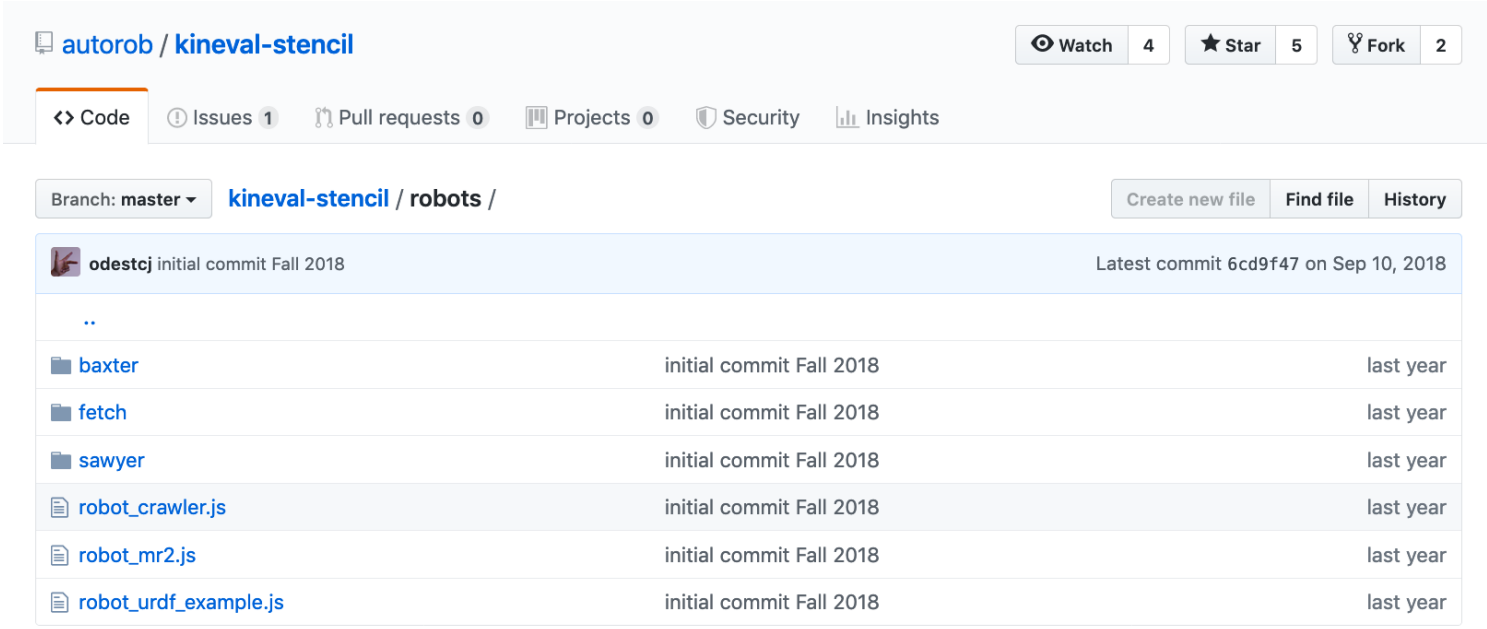
3 commits | 1 branch | 0 releases | 2 contributors | View license

Branch: master | New pull request | Find File | Clone or download

ohseejay doh! — Latest commit 2b7f1ca 10 days ago

js	initial commit Fall 2018	last year
kineval	initial commit Fall 2018	last year
project_pathplan	initial commit Fall 2018	last year
project_pendularm	doh!	10 days ago
robots	initial commit Fall 2018	last year
tutorial_heapsort	initial commit Fall 2018	last year
tutorial_js	initial commit Fall 2018	last year
worlds	initial commit Fall 2018	last year
LICENSE	initial commit Fall 2018	last year
README.md	initial commit Fall 2018	last year
home.html	initial commit Fall 2018	last year

KinEval Overview



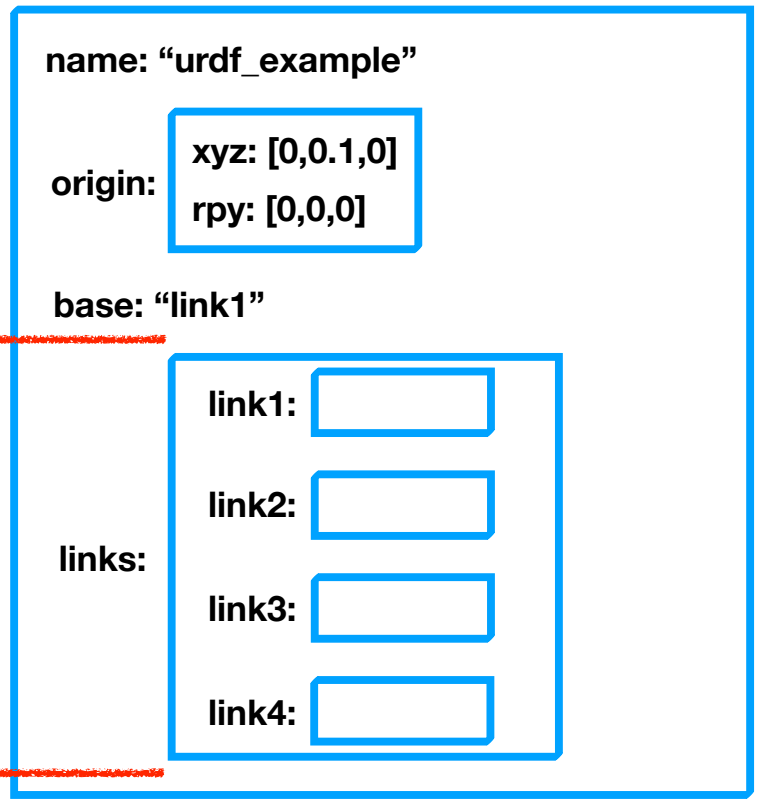
Each robot has its kinematic structure defined in URDF format within these JavaScript files

robot_urdf_example.js

robot_urdf_example.js

```
1   //   CREATE ROBOT STRUCTURE
2
3   ////////////////////////////////////////////////
4   /////     DEFINE ROBOT AND LINKS
5   ////////////////////////////////////////////////
6
7   // create robot data object
8   robot = new Object(); // or just {} will create new object
9
10  // give the robot a name
11  robot.name = "urdf_example";
12
13  // initialize start pose of robot in the world
14  robot.origin = {xyz: [0,0.1,0], rpy:[0,0,0]};
15
16  // specify base link of the robot; robot.origin is transform of world to the robot base
17  robot.base = "link1";
18
19  // specify and create data objects for the links of the robot
20  robot.links = {"link1": {}, "link2": {}, "link3": {}, "link4": {} };
21
```

STENCIL IN KINEVAL_ROBOT_INIT.JS FOR INITIALIZING THIS INFORMATION

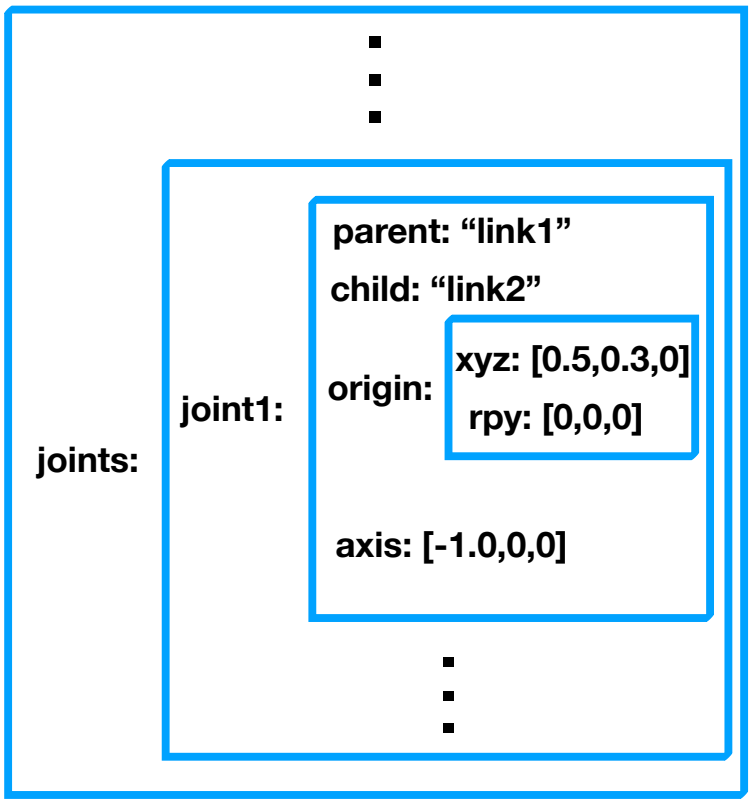


robot_urdf_example.js

robot_urdf_example.js

```
22  ////////////////////////////////////////////////
23  /////     DEFINE JOINTS AND KINEMATIC HIERARCHY
24  ////////////////////////////////////////////////
25
26  /*      joint definition template
27          // specify parent/inboard link and child/outboard link
28          robot.joints.joint1 = {parent:"link1", child:"link2"};
29          // joint origin's offset transform from parent link origin
30          robot.joints.joint1.origin = {xyz: [5,3,0], rpy:[0,0,0]};
31          // joint rotation axis
32          robot.joints.joint1.axis = [0.0,0.0,1.0];
33  */
34
35
36  // roll-pitch-yaw defined by ROS as corresponding to x-y-z
37  //http://wiki.ros.org/urdf/Tutorials/Create%20your%20own%20urdf%20file
38
39  // specify and create data objects for the joints of the robot
40  robot.joints = {};
41
42  robot.joints.joint1 = {parent:"link1", child:"link2"};
43  robot.joints.joint1.origin = {xyz: [0.5,0.3,0.0], rpy:[0,0,0]};
44  robot.joints.joint1.axis = [-1.0,0.0,0];  // simpler axis
45
46  robot.joints.joint2 = {parent:"link1", child:"link3"};
47  //robot.joints.joint2.origin = {xyz: [-0.2,0.5,0], rpy:[0,0,1.57]};
48  robot.joints.joint2.origin = {xyz: [-0.2,0.5,0], rpy:[0,0,Math.PI/2]};
49  //robot.joints.joint2.axis = [-0.707,0.707,0];
50  robot.joints.joint2.axis = [-Math.cos(Math.PI/4),Math.cos(Math.PI/4),0];
51
52  robot.joints.joint3 = {parent:"link3", child:"link4"};
53  //robot.joints.joint3.origin = {xyz: [0.5,0,0], rpy:[0,0,-1.57]};
54  robot.joints.joint3.origin = {xyz: [0.5,0,0], rpy:[0,0,-Math.PI/2]};
55  //robot.joints.joint3.axis = [0.707,-0.707,0];
56  robot.joints.joint3.axis = [Math.cos(Math.PI/4),-Math.cos(Math.PI/4),0];
57
58
59  // specify name of endeffector frame
60  robot.endeffector = {};
61  robot.endeffector.frame = "joint3";
62  robot.endeffector.position = [[0.5],[0],[0],[1]]
63
```

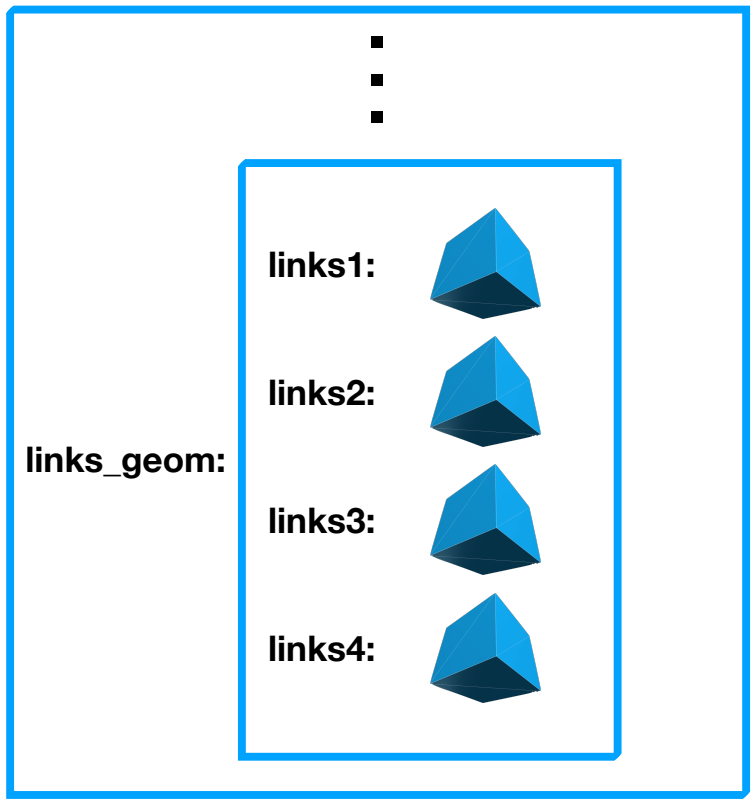
robot =



robot_urdf_example.js

robot_urdf_example.js

```
64  ////////////////////////////////////////////////
65  /////     DEFINE LINK threejs GEOMETRIES
66  ////////////////////////////////////////////////
67
68  /*  threejs geometry definition template, will be used by THREE.Mesh() to create threejs object
69      // create threejs geometry and insert into links_geom data object
70      links_geom["link1"] = new THREE.CubeGeometry( 5+2, 2, 2 );
71
72      // example of translating geometry (in object space)
73      links_geom["link1"].applyMatrix( new THREE.Matrix4().makeTranslation(5/2, 0, 0) );
74
75      // example of rotating geometry 45 degrees about y-axis (in object space)
76      var temp3axis = new THREE.Vector3(0,1,0);
77      links_geom["link1"].rotateOnAxis(temp3axis,Math.PI/4);
78  */
79
80  // define threejs geometries and associate with robot links
81  links_geom = {};
82
83  links_geom["link1"] = new THREE.CubeGeometry( 0.7+0.2, 0.5+0.2, 0.2 );
84  links_geom["link1"].applyMatrix( new THREE.Matrix4().makeTranslation((0.5-0.2)/2, 0.5/2, 0) );
85
86  links_geom["link2"] = new THREE.CubeGeometry( 0.5+0.2, 0.2, 0.2 );
87  links_geom["link2"].applyMatrix( new THREE.Matrix4().makeTranslation(0.5/2, 0, 0) );
88
89  links_geom["link3"] = new THREE.CubeGeometry( 0.5+0.2, 0.2, 0.2 );
90  links_geom["link3"].applyMatrix( new THREE.Matrix4().makeTranslation(0.5/2, 0, 0) );
91
92  links_geom["link4"] = new THREE.CubeGeometry( 0.5+0.2, 0.2, 0.2 );
93  links_geom["link4"].applyMatrix( new THREE.Matrix4().makeTranslation(0.5/2, 0, 0) );
94
```



URDF Usefulness

- Get the base link object: `robot.links[robot.base]`
- Get `link`'s parent joint's transform: `robot.joints[link.parent].xform`
- Get `joint`'s child link: `robot.links[joint.child]`
- Get `joint`'s parent link's joint children: `robot.links[joint.parent].children`

Implementation Advice

- **GLOBAL VARIABLE SCOPE**
  - In scope across **all** included JavaScript files
  - Change a global variable in one file, and that change will be reflected for **all** other files

Implementation Advice

- **GLOBAL VARIABLE SCOPE**

Image Source: https://giphy.com/gifs/MCZ39lz83o5lC

With great power comes many citations?

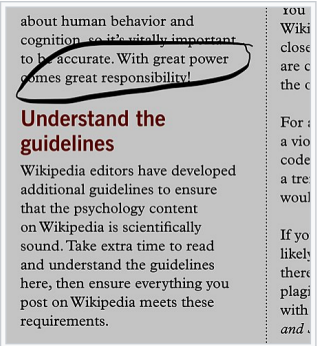
With great power comes great responsibility

From Wikipedia, the free encyclopedia

"Peter Parker principle" redirects here. It is not to be confused with Peter principle.

"With great power there must also come -- great responsibility!", also known as the **Peter Parker principle**, is a popular quote / phrase, motto and proverb that was made popular within Spider-Man comic books that debuted in *Amazing Fantasy #15* written by Stan Lee. The quote is often used in politics, monarchy and law enforcement, is commonly used among journalists and book authors and is used in popular culture such as in various media and memes.[1][2][3][4]

Contents [hide]


about human behavior and cognition, so it's vitally important to be accurate. With great power comes great responsibility!

Understand the guidelines

Wikipedia editors have developed additional guidelines to ensure that the psychology content on Wikipedia is scientifically sound. Take extra time to read and understand the guidelines here, then ensure everything you post on Wikipedia meets these requirements.

Screenshot of Wikimedia using the phrase for its guidelines for how to edit Wikipedia.

Implementation Advice

- **GLOBAL VARIABLE SCOPE**
  - In scope across **all** included JavaScript files
  - Change a global variable in one file, and that change will be reflected for **all** other files
- Direction of transform
  - `.xform`'s represent: component_frame->world_frame
  - Rotate then translate!

Motivation of Assignment

A Computational Perspective

- Robots exist as a collection of parts within an environment
  - Each component part has information associated with it
    - Part geometry, configuration state, control signal, etc.

      ↳ By definition, this information is independent from all other parts and the environment

  - Collectively, the robot has information relating each part to all other parts

    ↳ Similarly independent of individual component states and the environment

  - Can acquire knowledge of the environment through sensing

    ↳ I argue that the robot's internal information is a source of prior knowledge about the environment

    ↳ The robot is given information that it exists with some known (computable) configuration within the environment

Motivation of Assignment

A Computational Perspective

- Robots exist as a collection of parts within an environment
  - Each component part has information associated with it
  - Collectively, the robot has information relating each part to all other parts
  - Can acquire knowledge of the environment through sensing

To accomplish some desired task, our robot should make use of all available knowledge; its actions should be as fully informed as possible

↳ We need some way of relating each source of information

↳ Transform all information into unified frame of reference; Forward Kinematics

Final Thoughts

- If you think of an idea for an advanced extension, **which you are excited about,** let the course staff know!
  - We will consider the idea for possible extension points (following the working implementation)
- Have a great weekend!