p2.js P5 library for Planck Physics
https://github.com/bobcgausa/cook-js
https://github.com/shakiba/planck.js/tree/master/docs/api

The p2.js library for P5 provides a pixel-based interface for the Planck Box2D physics implementation. Thus, all the P5 functions and libraries can be used to code physics projects.

P2 Functions	Notes	Planck Equivalent
b2newWorld(scale, gravityVector)	All pixels arguments are divided by scale. b2scalefactor = scale	new planck.World
b2V(3, 4)	defines a new Vec2	Vec2
b2scaleTo(v)	scales any v to a new v	
b2scalexyTo(x, y)	scales to a new v	
b2scalexTo(x)	scales a scalar value	
b2scaleFrom(v)	scales v to pixels	
b2scalexFrom	scales scalar to pixels	
b2Draw(debug?)	draws all, debug=true draws joints, destroys bodies if out of viewBox or lifetime expired. Bodies with visible=false are not drawn. Inactive bodies are destroyed.	Planck provides a testbed version of the library with graphics support.
b2Update(timeScale?, forceStep?, positionStep?)		step(timeScale?, forceStep?, positionStep?)
b2Destroy	destroys all bodies	
new b2Body(type, dynamic, xy, wh, props?)	'box' 'polygon' 'circle' 'edge' 'chain' dynamic = true dynamic = false xy is position v in pixels wh is width/height v in	createDynamicBody(xy) createBody(xy) createFixture(shape,

	pixels or an array of v props { } property list	props)
.addTo(type, xy, wh, props?)	adds additional fixtures after the one created by b2Body	createFixture(shape, props)
b2getBodyFromFixture(fixture)	returns b2Body	getBody()
b2setWorld(x?, y?, w?, h?)	p2 supports a viewBox and supports world scrolling in any direction. The viewBox defaults to 50% larger than width and height in every direction in order to allow bodies to leave the scene and then reappear. The x position is the left edge, but y is the center.	
b2getWorld()	returns AABB	
b2get/setOrigin(v)		
b2shiftOrigin(v)	user must control scrolling using this function	
b2getFixtureAt(x, y)	returns fixture or null	
b2getAABB(body)	returns AABB	
b2queryAABB.query(aabb)	b2query.fixtures accesses array of intersecting fixtures	
b2rayCast.rayCast(point1, point2, mode?, ignoreArray?)	b2rayCast.ANY ALL NEAREST ignoreArray fixtures are ignored b2rayCast.points .normals .fixtures .fractions	
b2getBodyFromBox2d(body)	b2Body from Box2d body	
Body Properties/Functions accessed using getters/setters all arguments in pixels are scaled and all return values		
aabb(g)	Planck.AABB in pixels of body and all fixtures	

active(gs)	p2 destroys inactive	isActive(), setActive()
advance(s)		advance()
angle(gs)		get/setAngle()
angularDamping(gs)		get/setAngularDamping()
angularImpulse(s)		setAngularImpulse()
angularVelocity(gs)		get/setAngularVelocity()
awake(gs)		isAwake(), setAwake()
bullet(gs)		isBullet(), setBullet()
centerOfMass(g)		getWorldCenter()
collision(gs)	sets collision listener function(fixture1,fixture2) for begin-contact	on("begin-contact", listener)
density(gs)	returns density for first fixture, sets all fixtures	setDensity()
display(gs)	get/set user-draw display function(b2Body)	
dynamic(gs)		isDynamic(), setDynamic()
filterCategoryBits(gs)	returns for 1st fixture, sets all	get/setFilterCategoryBits()
filterMaskBits(gs)	returns for 1st fixture, sets all	get/setFilterMaskBits()
filterGroupIndex(gs)	returns for 1st fixture, sets all	get/setFilterGroupIndex()
fixedRotation(gs)		isFixedRotation(), setFixedRotation()
fixtureList(g)	used to start iteration	getFixtureList()
fixture(g)	gets first fixture	
force(s)		applyForceToCenter()
friction(gs)	returns for 1st fixture, sets all	get/setFriction()

gravityScale(gs)		get/setGravityScale()
image(gs)	returns for 1st fixture, sets all, drawn for fixture	
imageResize(gs)	returns for 1st fixture, sets all, v to resize image	
impulse(s)		applyLinearImpulse() to WorldCenter
inertia(g)		getInertia()
jointList(g)	used to start iteration	getJointList()
kinematic(gs)		isKinematic(), setKinematic()
life(gs)	decremented on every update, <=0 will destroy	
linearDamping(gs)		get/setLinearDamping()
linearVelocity(gs)		get/setLinearVelocity()
localCenter(g)		getLocalCenter()
mass(g)		getMass()
massData(gs)	`{ I: 0, center: b2V(0, 0), mass: 0 }`	get/setMassData()
next(g)	used with bodyList	getNext()
position(gs)		get/setPosition()
restitution(gs)	returns for 1st fixture, sets all	get/setRestitution()
sensor(gs)	returns for 1st fixture, sets all	get/setSensor()
sleepingAllowed(gs)		isSleepingAllowed(), setSleepingAllowed()
static(gs)		isStatic(), setStatic()
torque(s)		applyTorque()
transform(gs)	`{position: v, angle: radians}`	get/setTransform

userData(gs)		get/setUserData()
visible(gs)	controls drawing only	
world(g)		getWorld()
worldCenter(g)		getWorldCenter()
worldLocked(g)		isWorldLocked()
body functions that don't fit getter/setter format		
.applyForce(force, position)		applyForce()
.applyImpulse(force, position)		applyLinearImpulse()
.createJoint(type, bodyB, props, anchor?)		createJoint()
.destroy()	sets active=false	destroyBody()
.destroyFixture(fixture)		destroyFixture()
.draw()	helper function that completes drawing after the user has set attributes	
.drawFixture(fixture)	helper function so that user can set attributes for each fixture	
.localPoint(point)		getLocalPoint()
.localVector(vector)		getLocalVector()
.resetMassData()		resetMassData()
.shouldCollide(body)		shouldCollide()
.synchronizeFixtures()		synchronizeFixtures()
.synchronizeTransform()		synchronizeTransform()
.toString()	returns position, velocity, angle	
.velocityFromLocalPoint(point)		getLinearVelocityFromLocalPoint()

.velocityFromWorldPoint(point)		getLinearVelocityFromWorldPoint()
.worldPoint(point)		getWorldPoint()
.worldVector(vector)		getWorldVector()