Module Interface Specification

Contents

Members

b2Vec2

b2Vec2 This imports the Box2D Vector and any associated methods.

Source: AsimpleCar.js, line 6

cameraX

cameraX This variable keeps track of the horizontal velocity of the camera.

Source: AsimpleCar.js, line 33

points

points This variable keeps track of points on a car

Source: AsimpleCar.js, line 25

Methods

createBox(world, x, y, width, height, angle) → {Body}

This method creates a box with a specified width and height rotated at a specified angle on the screen.

Parameters:

Name	Type	Description
world	b2World	The Box2D world that the box is created in
x	Integer	The x-coordinate of the upper left corner
y	Integer	The y-coordinate of the upper left corner
width	Integer	The width of the box
height	Integer	The height of the box
angle	Float	The rotation of the box, counterclockwise from the horizontal, in radians

Source: AsimpleCar.js, line 241

Returns:

The box

Type
Body

createGeneration(n) → {Array.<Car>}

This method creates a new generation of cars.

Parameters:

Name	Type	Description
n	Integer	The number of cars in the generation

Source: AsimpleCar.js, line 320

Returns:

An array of cars, denoting the new generation.

Type
Array.<Car>

createRoad(world, startX, startY, maxNumberOfTiles)

This method creates a road on the screen

Parameters:

Name	Type	Description
world	b2World	The Box2D world that the box is created in
startX	Integer	The x-coordinate of the upper left corner of the first tile
startY	Integer	The y-coordinate of the upper left corner of the first tile
maxNumberOfTiles	Integer	The number of tiles the road is made out of

Source: AsimpleCar.js, line 269

crossOverOffsprings(cars, topCars) → {Array.<Cars>}

This method crosses over the chromosomes of the offspring cars.

Parameters:

Name	Type	Description
cars	Array.<Cars>	The array of cars to crossover
topCars	Integer	The number of cars in the surviving parent generation

Source: AsimpleCar.js, line 341

Returns:

An array of the crossed-over cars

Type
Array.<Cars>

draw_world(world, context)

This method draws the world on the screen, before it is updated.

Parameters:

Name	Type	Description
world	b2World	The world to draw on
context	Canvas	The canvas to draw the world on

Source: AsimpleCar.js, line 302

drawCar(world, worldScale, vertex1X, vertex1Y, vertex2X, vertex2Y, vertex3X, vertex3Y, vertex4X, vertex4Y, vertex5X, vertex5Y, vertex6X, vertex6Y, vertex7X, vertex7Y, vertex8X, vertex8Y, frontwheelPos, rearWheelPos) → {b2BodyDef}

This method creates a car to the screen.

Parameters:

Name	Type	Description
world	b2World	The Box2D world where the car will be placed in
worldScale	Integer	The scaling factor for the Box2D world
vertex1X	Integer	The x-coordinate of the first vertex
vertex1Y	Integer	The y-coordinate of the first vertex
vertex2X	Integer	The x-coordinate of the second vertex
vertex2Y	Integer	The y-coordinate of the second vertex
vertex3X	Integer	The x-coordinate of the third vertex
vertex3Y	Integer	The y-coordinate of the third vertex
vertex4X	Integer	The x-coordinate of the fourth vertex
vertex4Y	Integer	The y-coordinate of the fourth vertex
vertex5X	Integer	The x-coordinate of the fifth vertex
vertex5Y	Integer	The y-coordinate of the fifth vertex
vertex6X	Integer	The x-coordinate of the sixth vertex
vertex6Y	Integer	The y-coordinate of the sixth vertex
vertex7X	Integer	The x-coordinate of the seventh vertex
vertex7Y	Integer	The y-coordinate of the seventh vertex
vertex8X	Integer	The x-coordinate of the eighth vertex
vertex8Y	Integer	The y-coordinate of the eighth vertex
frontwheelPos	Integer	The vertex that the front wheel is attached to
rearWheelPos	Integer	The vertex that the back wheel is attached to

Source: AsimpleCar.js, line 176

Returns:

The completed car

Type
b2BodyDef

getChromosome() → {Array.<Object>}

This method returns the chromosome of the car.

Source: AsimpleCar.js, line 364

Returns:

A multi-typed array representing the chromosome of the car.

Type
Array.<Object>

getRandomArbitrary(min, max, A)

This method generates a random floating point number between min and max, exclusive.

Parameters:

Name	Type	Description
min	Integer	The lower bound
max	Integer	The upper bound
A	Float	floating point number between min and max.

Source: AsimpleCar.js, line 389

getRandomArbitraryInteger(min, max) → {Integer}

This method generates a random integer between min and max, exclusive.

Parameters:

Name	Type	Description
min	Integer	The lower bound
max	Integer	The upper bound

Source: AsimpleCar.js, line 378

Returns:

A random number between min and max

Type
Integer

init()

This method initializes the Box2D environment, and any objects within the Box2D world.

Source: AsimpleCar.js, line 40

Returns:

The created Box2D world

makeCarJoints(world, bodyA, bodyB, wheelPosX, wheelPosY) → {b2RevoluteJointDef}

This method creates joints used to connect the wheels to the car chassis.

Parameters:

Name	Type	Description
world	b2World	The Box2D world where the joint will be placed in
bodyA	b2BodyDef	The first object to connect the joint to
bodyB	b2BodyDef	The second object to connect the joint to

Name	Type	Description
wheelPosX	Integer	The x-coordinate of the wheel
wheelPosY	Integer	The y-coordinate of the wheel

Source: AsimpleCar.js, line 123

Returns:

The joint connecting bodyA to bodyB

Type
b2RevoluteJointDef

makePolygon(num, vertex1X, vertex1Y, vertex2X, vertex2Y) → {b2FixtureDef}

This method creates a polygon for the car given 4 points on the Cartesian plane. It assumes that one of the points of the polygon will be at the origin.

Parameters:

Name	Type	Description
num	Integer	The index of the polygon of the car
vertex1X	Integer	The x-coordinate of the first vertex
vertex1Y	Integer	The y-coordinate of the first vertex
vertex2X	Integer	The x-coordinate of the second vertex
vertex2Y	Integer	The y-coordinate of the second vertex

Source: AsimpleCar.js, line 74

Returns:

The polygon created

Type
b2FixtureDef

makeWheelFixture(world, car, wheelbodyDef, wheelFixture) → {Body}

This method connects the wheel to the car chassis.

Parameters:

Name	Type	Description
world	b2World	The Box2D world where the wheel will be placed in
car	b2BodyDef	The car to connect the wheels to
wheelbodyDef	b2BodyDef	The body (physics) definition of the wheel
wheelFixture	b2FixtureDef	The shape definition of the wheel

Source: AsimpleCar.js, line 144

Returns:

The wheel

Type

Body

makeWheelShape(world, worldScale, radius) → {b2FixtureDef}

This method creates the shape of a wheel for the car given its radius.

Parameters:

Name	Type	Description
world	b2World	The Box2D world where the wheel will be placed in
worldScale	Integer	The scaling factor
radius	Float	The radius of the wheel

Source: AsimpleCar.js, line 102

Returns:

The shape of the wheel created

Type
b2FixtureDef

mutateOffsprings(cars, numberOfParents, mutationFactor) → {Array.<Cars>}

This method mutates the genes in the offspring's chromosomes.

Parameters:

Name	Type	Description
cars	Array.<Cars>	The array of cars to crossover
numberOfParents	Integer	The number of parents in the cars array
mutationFactor	Float	The likelihood of mutation

Source: AsimpleCar.js, line 352

Returns:

An array of the mutated cars

Type
Array.<Cars>

selectNextGeneration(cars, n) → {Array.<Cars>}

This method selects for the next generation of cars.

Parameters:

Name	Type	Description
cars	Array.<Cars>	The array of cars to choose from.
n	Integer	The number of cars to select for.

Source: AsimpleCar.js, line 330

Returns:

An array of the top n cars.

Type

Array.<Cars>

update()

This method updates the screen.

Source: AsimpleCar.js, line 284
