Module Interface Specification: GrateBox.js

Members

CAM_SPEED

This variable keeps track of the default speed of the camera in the x-direction.

Source: GrateBox.js, line 72

CAM_X_TRANSLATION

This variable keeps track of the default shift factor in the movement of the camera in the x-direction.

Source: GrateBox.js, line 66

camerax

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

Source: GrateBox.js, line 101

cameray

This variable keeps track of the vertical velocity of the camera.

Source: GrateBox.js, line 107

car

This variable holds the car model

Source: GrateBox.js, line 95

carsArray

This variable array contains the cars in the population cars.

Source: GrateBox.js, line 153

currentGeneration

This is the current generation that the simulation is in.

Source: GrateBox.js, line 173

currentMember

This variable integer indicates the current member of the group of cars.

Source: GrateBox.js, line 163

Contents

DEFAULT_CAM_X

This variable keeps track of the default shift factor for the camera in the x-direction.

Source: GrateBox.js, line 60

diffx

This variable keeps track of the change in the horizontal displacement of the camera.

Source: GrateBox.js, line 112

diffy

This variable keeps track of the change in the vertical displacement of the camera.

Source: GrateBox.js, line 117

DRAW_SCALE

This variable keeps track of the scaling factor of the display of the simulation.

Source: GrateBox.js, line 16

FILL_ALPHA

This variable keeps track of the alpha value of the display of the simulation.

Source: GrateBox.js, line 21

frameRate

This is the frame rate for the simulation.

Source: GrateBox.js, line 168

GRAVITY

This variable keeps track of the acceleration of gravity in the simulation.

Source: GrateBox.js, line 6

INTERVAL_RATE

This variable keeps track of how often the simulation updates.

Source: GrateBox.js, line 49

LINE_THICKNESS

This variable keeps track of the line thickness.

Source: GrateBox.js, line 26

MIN_NUMBER_OF_CARS

This variable keeps track of the minimum number of cars allowed in the simulation.

Source: GrateBox.js, line 78

MOVEMENT_THRESHOLD

This variable keeps track of the minimum amount that a car has to move per iteration in order for it to be considered moving.

Source: GrateBox.js, line 44

mutationRate

This variable indicates the rate at which mutations occur.

Source: GrateBox.js, line 148

NUMBER_OF_GENES

This variable keeps track of the number of genes that each car has.

Source: GrateBox.js, line 83

parentPool

This variable indicates the size of the pool from which parents creat offspring.

Source: GrateBox.js, line 143

paused

This variable represents whether the user has paused the simulation.

Source: GrateBox.js, line 179

points

This variable keeps track of points on a car

Source: GrateBox.js, line 90

populationSize

This variable indicates the size of the initial population of cars.

Source: GrateBox.js, line 138

POSITION_ITERATION

This variable keeps track of the timestep used to update the position in the simulation.

Source: GrateBox.js, line 38

proc1

This variable keeps track of the game loop thread.

Source: GrateBox.js, line 122

proc2

This variable keeps track of updateCar thread.

Source: GrateBox.js, line 127

TIMEOUT_RATE

This variable keeps track of the maximum lifespan of the car.

Source: GrateBox.js, line 54

topCars

This variable array contains the highest performing cars for the purpose of creating the next generation.

Source: GrateBox.js, line 158

VELOCITY_ITERATION

This variable keeps track of the timestep used to update velocity in the simulation.

Source: GrateBox.js, line 32

WORLD_SCALE

This variable keeps track of the scaling factor for the objects in the simulation.

Source: GrateBox.js, line 11

Methods

cameraPos()

This method sets the camera position to the position of the car.

Source: GrateBox.js, line 337

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: GrateBox.js, line 377

Returns:

An array of the crossed-over cars

Type
Array.<Cars>

drawworld(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: GrateBox.js, line 323

getRandomArbitrary(min, max) → {Float}

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

Source: GrateBox.js, line 597

Returns:

A floating point number between min and max.

Type
Float

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: GrateBox.js, line 584

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: GrateBox.js, line 190

Returns:

The created Box2D world

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: GrateBox.js, line 418

Returns:

An array of the mutated cars

Type
Array.<Cars>

nextCar()

This method selects the next car to be simulated.

Source: GrateBox.js, line 263

partition(items, left, right)

This method partitions the array into two sets based on a pivot. The following code was modified from: https://www.nczonline.net/blog/2012/11/27/computer-science-in-javascript-quicksort/

Parameters:

Name	Type	Description
items	Array.<Cars>	An array of cars
left	Integer	The left index of the pivot
right	Integer	The right index of the pivot

Source: GrateBox.js, line 555

Returns:

The left index of the partitioned array

quicksort(cars, left, right)

This method preforms quicksort on an array of cars according to fitness value.
The following code was modified from:
https://www.nczonline.net/blog/2012/11/27/computer-science-in-javascript-quicksort/

Parameters:

Name	Type	Description
cars	Array.<Cars>	The array of cars to sort
left	Integer	The left index
right	Integer	The right index

Source: GrateBox.js, line 513

Returns:

The sorted cars array

resetCamera(world, context)

This method resets the camera for the next simulation.

Parameters:

Name	Type	Description
world	b2World	The world on which the camera is reset.
context	Canvas	The canvas to draw the world on

Source: GrateBox.js, line 308

resetWorld(world)

This method resets the world for the next simulation.

Parameters:

Name	Type	Description
world	b2World	The world to be reset.

Source: GrateBox.js, line 296

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: GrateBox.js, line 356

Returns:

An array of the top n cars.

Type
Array.<Cars>

swap(items, firstIndex, secondIndex)

This method swaps 2 items in an array. The following code was obtained from: https://www.nczonline.net/blog/2012/11/27/computer-science-in-javascript-quicksort/

Parameters:

Name	Type	Description
items	Array.<Cars>	An array of cars
firstIndex	Integer	The index of the first car to swap
secondIndex	Integer	The index of the second car to swap

Source: GrateBox.js, line 539

update()

This method updates the screen.

Source: GrateBox.js, line 238
