Package me.miles.matthew.spaceflight.physics

Class SpaceEnvironment

java.lang.Object
    me.miles.matthew.spaceflight.physics.SpaceEnvironment

```
public class SpaceEnvironment
extends java.lang.Object
```

Constructor Summary

Constructors

Constructor	Description
`SpaceEnvironment()`	Creates a new space environment

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
`void`	`addBody(CelestialBody body)`	Add a body to the environment
`void`	`addBody(PhysicsObject object)`	Add a body to the environment
`void`	`autoDoubleOrbit(CelestialBody satellite, CelestialBody centre, CelestialBody master)`	Automatically gives the necessary x and y velocity to orbit a body which is in turn orbiting another body
`void`	`autoOrbit(CelestialBody satellite, CelestialBody centre)`	Automatically gives the necessary x and y velocity to orbit a body
`int`	`getBackgroundColour()`	get the background colour

Modifier and Type	Method	Description
`java.util.ArrayList<PhysicsObject>`	`getBodies()`	Adds a new body to the environment
`PhysicsObject[]`	`getBodiesArray()`	Get an array of all bodies in the environment
`PhysicsObject`	`getBodyAt(int index)`	Gets a body based on the order it was added to the environment
`int`	`getBodyCount()`	Get the number of bodies in the environment
`void`	`physicsTick(long millisPassed, long simulationSpeed)`	Updates the environment and applies acceleration after a certain time passed
`void`	`removeBody(int pos)`	Remove a body from the environment
`void`	`setBackgroundColour(int backgroundColour)`	Set the background colour
`void`	`setBodies(java.util.ArrayList<PhysicsObject> bodies)`	Sets the entire list of bodies in the environment

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Details

SpaceEnvironment

```
public SpaceEnvironment()
```

Creates a new space environment

Method Details

getBodies

```
public java.util.ArrayList<PhysicsObject> getBodies()
```

Adds a new body to the environment

Parameters:

body - The body to add

getBodiesArray

```
public PhysicsObject[] getBodiesArray()
```

Get an array of all bodies in the environment

Returns:

An array of all bodies in the environment

setBodies

```
public void setBodies(java.util.ArrayList<PhysicsObject> bodies)
```

Sets the entire list of bodies in the environment

Parameters:

bodies - The list of bodies to set

getBackgroundColour

```
public int getBackgroundColour()
```

get the background colour

Returns:

The background colour

setBackgroundColour

```
public void setBackgroundColour(int backgroundColour)
```

Set the background colour

Parameters:

backgroundColour - The colour to set

getBodyAt

```
public PhysicsObject getBodyAt(int index)
```

Gets a body based on the order it was added to the environment

Parameters:

`index` - The index of the body to get

getBodyCount

```
public int getBodyCount()
```

Get the number of bodies in the environment

Returns:

The number of bodies in the environment

addBody

```
public void addBody(PhysicsObject object)
```

Add a body to the environment

Parameters:

`object` - The body to add

addBody

```
public void addBody(CelestialBody body)
```

Add a body to the environment

Parameters:

`object` - The body to add

removeBody

```
public void removeBody(int pos)
```

Remove a body from the environment

Parameters:

`object` - The index of the body to remove

autoOrbit

```
public void autoOrbit(CelestialBody satellite,
                      CelestialBody centre)
```

Automatically gives the necessary x and y velocity to orbit a body

Parameters:

`satellite` - The satellite being put in orbit

`centre` - The body being orbited

autoDoubleOrbit

```
public void autoDoubleOrbit(CelestialBody satellite,
                            CelestialBody centre,
                            CelestialBody master)
```

Automatically gives the necessary x and y velocity to orbit a body which is in turn orbiting another body

Parameters:

`satellite` - The satellite being put in orbit

`centre` - The body being orbited

`master` - The body that is being orbited by the body the satellite is orbitting

physicsTick

```
public void physicsTick(long millisPassed,
                        long simulationSpeed)
```

Updates the environment and applies acceleration after a certain time passed

Parameters:

`millisPassed` - The time passed since the last update

`simulationSpeed` - The speed of the simulation