Package me.miles.matthew.spaceflight.physics

Class CelestialBody

java.lang.Object
    me.miles.matthew.spaceflight.physics.PhysicsObject
        me.miles.matthew.spaceflight.physics.CelestialBody

```
public class CelestialBody
extends PhysicsObject
```

Field Summary

Fields

Modifier and Type	Field	Description
`static int`	**`ASTEROID`**	
`static int`	**`BLACK_HOLE`**	
`static int`	**`MOON`**	
`static int`	**`PLANET`**	
`static int`	**`STAR`**	

Fields inherited from class me.miles.matthew.spaceflight.physics.PhysicsObject

`GRAVITATIONAL_CONSTANT`

Constructor Summary

Constructors

Constructor	Description
`CelestialBody(double mass, double xPos, double yPos, boolean isLandable, int colour, double radius, java.lang.String name, int type)`	Creates a new celestial body
`CelestialBody(double mass, Vector2d position, boolean isLandable, int colour, double radius, java.lang.String name, int type)`	Creates a new celestial body

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description

Modifier and Type	Method	Description
`void`	`draw(java.awt.Graphics2D g2, double lX, double tY, int windowWidth, int windowHeight, double zoom)`	Draws the object on screen with set screen centre position cX, cY and zoom
`int`	`getColour()`	Get the primary colour of the body
`java.lang.String`	`getName()`	Get the name of the body
`float`	`getPointsPerSecond()`	Get the number of points that are added to the trail per second
`java.awt.image.BufferedImage`	`getTexture()`	Get the texture of the body
`Vector2d[]`	`getTrail()`	Gets a list of points that make up the trail of the body
`int`	`getTrailSize()`	Get the trail length
`int`	`getType()`	Get the type of the body
`boolean`	`isClickedOn(double lX, double tY, int xClick, int yClick, double zoom)`	Gets if the object is being clicked on for a mouse at a certain screen coordinate

Modifier and Type	Method	Description
`boolean`	`isFocussed()`	Get if the body is focussed on
`boolean`	`isLandable()`	Find if the body could be landed on by a spacecraft
`void`	`physicsTick(long timePassedMillis, long simulationSpeed)`	Applies movement over a certain time period, based on the real time passed and the simulation speed
`void`	`setCanLandOn(boolean canLandOn)`	Set if the body can be landed on by a spacecraft
`void`	`setColour(int colour)`	Set the primary colour of the body
`void`	`setFocus(boolean isFocussed)`	Set if the body is focussed on
`void`	`setLandable(boolean canLandOn)`	Set whether the body can be landed on by a spacecraft
`void`	`setName(java.lang.String name)`	Set the name of the body

Modifier and Type	Method	Description
`void`	`setPointsPerSecond (float pointsPerSecond)`	Set the number of points that are added to the trail per second
`void`	`setTexture (java.awt.image.BufferedImage texture)`	Set the texture of the body
`void`	`setTrail(Queue<Vector2d> trail)`	Set the entire trail of the body
`void`	`setTrailSize(int trailSize)`	Set the trail length
`void`	`setType(int type)`	Set the type of the body

Methods inherited from class me.miles.matthew.spaceflight.physics.PhysicsObject

`doGAcceleration, getAngleTo, getAttractionTo, getMass, getPos, getRadius, getSurfaceAcceleration, getXPos, getXVel, getYPos, getYVel, setMass, setPos, setPos, setRadius, setXPos, setXVel, setYPos, setYVel`

Methods inherited from class java.lang.Object

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

Field Details

BLACK_HOLE

```
public static final int BLACK_HOLE
```

See Also:

Constant Field Values

STAR

```
public static final int STAR
```

See Also:

Constant Field Values

PLANET

```
public static final int PLANET
```

See Also:

Constant Field Values

MOON

```
public static final int MOON
```

See Also:

Constant Field Values

ASTEROID

```
public static final int ASTEROID
```

See Also:

Constant Field Values

Constructor Details

CelestialBody

```
public CelestialBody(double mass,
                     double xPos,
                     double yPos,
                     boolean isLandable,
                     int colour,
                     double radius,
                     java.lang.String name,
                     int type)
```

Creates a new celestial body

Parameters:

`mass` - the mass of the celestial body

`xPos` - the x position of the celestial body (Space coordinates) `yPos` - the y position of the celestial body (Space coordinates)

`isLandable` - whether or not the celestial body could be landed on

`colour` - the colour of the celestial body

`radius` - the radius of the celestial body

`name` - the name of the celestial body

`type` - the type of celestial body

CelestialBody

```
public CelestialBody(double mass,
                     Vector2d position,
                     boolean isLandable,
                     int colour,
                     double radius,
                     java.lang.String name,
                     int type)
```

Creates a new celestial body

Parameters:

`mass` - the mass of the celestial body

`position` - the position of the celestial body (Space coordinates)

`isLandable` - whether or not the celestial body could be landed on

`colour` - the colour of the celestial body

`radius` - the radius of the celestial body

`name` - the name of the celestial body

`type` - the type of celestial body

Method Details

draw

```
public void draw(java.awt.Graphics2D g2,
                 double lX,
                 double tY,
                 int windowWidth,
                 int windowHeight,
                 double zoom)
```

Description copied from class: `PhysicsObject`

Draws the object on screen with set screen centre position cX, cY and zoom

Specified by:

`draw` in class `PhysicsObject`

physicsTick

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

Description copied from class: `PhysicsObject`

Applies movement over a certain time period, based on the real time passed and the simulation speed

Overrides:

`physicsTick` in class `PhysicsObject`

Parameters:

`timePassedMillis` - the time passed since the last update in milliseconds

`simulationSpeed` - the number of seconds passed in the simulation per real world second

isLandable

```
public boolean isLandable()
```

Find if the body could be landed on by a spacecraft

Returns:

true if the body can be landed on, false otherwise

setLandable

```
public void setLandable(boolean canLandOn)
```

Set whether the body can be landed on by a spacecraft

Parameters:

`canLandOn` - true if the body can be landed on, false otherwise

getColour

```
public int getColour()
```

Get the primary colour of the body

Returns:

the primary colour of the body

setColour

```
public void setColour(int colour)
```

Set the primary colour of the body

Parameters:

`colour` - the primary colour of the body

getName

```
public java.lang.String getName()
```

Get the name of the body

Returns:

the name of the body

setName

```
public void setName(java.lang.String name)
```

Set the name of the body

Parameters:

`name` - the name of the body

setCanLandOn

```
public void setCanLandOn(boolean canLandOn)
```

Set if the body can be landed on by a spacecraft

Parameters:

`canLandOn` - true if the body can be landed on, false otherwise

getTexture

```
public java.awt.image.BufferedImage getTexture()
```

Get the texture of the body

Returns:

the texture of the body

setTexture

```
public void setTexture(java.awt.image.BufferedImage texture)
```

Set the texture of the body

Parameters:

`texture` - the texture of the body

isFocussed

```
public boolean isFocussed()
```

Get if the body is focussed on

Returns:

true if the body is focussed on, false otherwise

setFocus

```
public void setFocus(boolean isFocussed)
```

Set if the body is focussed on

Parameters:

`focussed` - true if the body is focussed on, false otherwise

getTrail

```
public Vector2d[] getTrail()
```

Gets a list of points that make up the trail of the body

Returns:

a list of points that make up the trail of the body

setTrail

```
public void setTrail(Queue<Vector2d> trail)
```

Set the entire trail of the body

Parameters:

`trail` - a queue containing the entire trail of the body

getTrailSize

```
public int getTrailSize()
```

Get the trail length

Returns:

the trail length

setTrailSize

```
public void setTrailSize(int trailSize)
```

Set the trail length

Parameters:

`trailSize` - the trail length

getPointsPerSecond

```
public float getPointsPerSecond()
```

Get the number of points that are added to the trail per second

Returns:

the number of points that are added to the trail per second

setPointsPerSecond

```
public void setPointsPerSecond(float pointsPerSecond)
```

Set the number of points that are added to the trail per second

Parameters:

`pointsPerSecond` - the number of points that are added to the trail per second

getType

```
public int getType()
```

Get the type of the body

Returns:

the type of the body

setType

```
public void setType(int type)
```

Set the type of the body

Parameters:

`type` - the type of the body

isClickedOn

```
public boolean isClickedOn(double lX,
                           double tY,
                           int xClick,
                           int yClick,
                           double zoom)
```

Description copied from class: `PhysicsObject`

Gets if the object is being clicked on for a mouse at a certain screen coordinate

Specified by:

`isClickedOn` in class `PhysicsObject`

Parameters:

`lX` - The left most x coordinate of the screen

`tY` - The top most y coordinate of the screen

`xClick` - The x coordinate of the mouse

`yClick` - The y coordinate of the mouse

`zoom` - The zoom of the screen

Returns:

If the object is being clicked on