Package me.miles.matthew.spaceflight.physics

Class PhysicsObject

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

Direct Known Subclasses:

`CelestialBody`

```
public abstract class PhysicsObject
extends java.lang.Object
```

Field Summary

Fields

Modifier and Type	Field	Description
`static double`	`GRAVITATIONAL_CONSTANT`	

Constructor Summary

Constructors

Constructor	Description
`PhysicsObject(double mass, double xPos, double yPos, double radius)`	Creates a new physics object
`PhysicsObject(double mass, Vector2d position, double radius)`	Creates a new physics object

Method Summary

All Methods | Instance Methods | Abstract Methods | Concrete Methods

Modifier and Type	Method	Description
`void`	`doGAcceleration(PhysicsObject o, long timePassedMillis, long simulationSpeed)`	Applies acceleration towards another body within the environment Uses the equation F = G*(m1*m2)/(r^2)
`abstract 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

Modifier and Type	Method	Description
`double`	`getAngleTo(double targetX, double targetY)`	Returns the angle to another object in degrees
`double`	`getAttractionTo(PhysicsObject o)`	Get the force due to gravity of attraction to an object in Newtons
`double`	`getMass()`	Gets the mass of the object
`Vector2d`	`getPos()`	Gets the position of the object
`double`	`getRadius()`	Gets the radius of the object
`double`	`getSurfaceAcceleration()`	Get the acceleration due to gravity of attraction to an object in m/s^2
`double`	`getXPos()`	Gets the x position of the object
`double`	`getXVel()`	Gets the x velocity of the object
`double`	`getYPos()`	Gets the y position of the object
`double`	`getYVel()`	Gets the y velocity of the object
`abstract 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
`void`	`physicsTick(long timePassedMillis, long simulationSpeed)`	Applies movement over a certain time period, based on the real time passed and the simulation speed
`void`	`setMass(double mass)`	Gets the mass of the object
`void`	`setPos(double xPos, double yPos)`	Sets the position of the object
`void`	`setPos(Vector2d pos)`	Sets the position of the object
`void`	`setRadius(double radius)`	Sets the radius of the object
`void`	`setXPos(double xPos)`	Sets the x position of the object
`void`	`setXVel(double xVel)`	Sets the x velocity of the object
`void`	`setYPos(double yPos)`	Sets the y position of the object
`void`	`setYVel(double yVel)`	Sets the y velocity of the object

Methods inherited from class java.lang.Object

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

Field Details

GRAVITATIONAL_CONSTANT

```
public static final double GRAVITATIONAL_CONSTANT
```

See Also:

Constant Field Values

Constructor Details

PhysicsObject

```
public PhysicsObject(double mass,
                     double xPos,
                     double yPos,
                     double radius)
```

Creates a new physics object

Parameters:

`mass` - The mass of the object

`xPos` - The x position of the object (Space coords)

`yPos` - The y position of the object (Space coords)

`radius` - The radius of the object

PhysicsObject

```
public PhysicsObject(double mass,
                     Vector2d position,
                     double radius)
```

Creates a new physics object

Parameters:

`mass` - The mass of the object

`position` - The position of the object (Space coords)

`radius` - The radius of the object

Method Details

isClickedOn

```
public abstract 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

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

getMass

```
public double getMass()
```

Gets the mass of the object

Returns:

The mass of the object

setMass

```
public void setMass(double mass)
```

Gets the mass of the object

Parameters:

`mass` - The mass of the object

getXVel

```
public double getXVel()
```

Gets the x velocity of the object

Returns:

The x velocity of the object

setXVel

```
public void setXVel(double xVel)
```

Sets the x velocity of the object

Parameters:

`xVel` - The x velocity of the object

getYVel

```
public double getYVel()
```

Gets the y velocity of the object

Returns:

The y velocity of the object

setYVel

```
public void setYVel(double yVel)
```

Sets the y velocity of the object

Parameters:

`yVel` - The y velocity of the object

getRadius

```
public double getRadius()
```

Gets the radius of the object

Returns:

setRadius

```
public void setRadius(double radius)
```

Sets the radius of the object

Parameters:

`radius` - The radius of the object

getXPos

```
public double getXPos()
```

Gets the x position of the object

Returns:

The x position of the object

setXPos

```
public void setXPos(double xPos)
```

Sets the x position of the object

Parameters:

`xPos` - The x position of the object

getYPos

```
public double getYPos()
```

Gets the y position of the object

Returns:

The y position of the object

setYPos

```
public void setYPos(double yPos)
```

Sets the y position of the object

Parameters:

`yPos` - The y position of the object

getPos

```
public Vector2d getPos()
```

Gets the position of the object

Returns:

The position of the object

setPos

```
public void setPos(double xPos,
                   double yPos)
```

Sets the position of the object

Parameters:

`xPos` - The x position of the object

`yPos` - The y position of the object

setPos

```
public void setPos(Vector2d pos)
```

Sets the position of the object

Parameters:

`pos` - The position of the object

draw

```
public abstract 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

Parameters:

`g2` -

`cX` -

`cY` -

`zoom` -

getAttractionTo

```
public double getAttractionTo(PhysicsObject o)
```

Get the force due to gravity of attraction to an object in Newtons

Parameters:

`o` - the other object being attracted to

Returns:

the force in newtons which is experienced by each object

getSurfaceAcceleration

```
public double getSurfaceAcceleration()
```

Get the acceleration due to gravity of attraction to an object in m/s^2

Returns:

the acceleration in m/s^2

getAngleTo

```
public double getAngleTo(double targetX,
                         double targetY)
```

Returns the angle to another object in degrees

Parameters:

`targetX` -

`targetY` -

Returns:

angle in degrees

doGAcceleration

```
public void doGAcceleration(PhysicsObject o,
                            long timePassedMillis,
                            long simulationSpeed)
```

Applies acceleration towards another body within the environment Uses the equation F = G*(m1*m2)/(r^2)

Parameters:

`o` - the other object being attracted to

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

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

physicsTick

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

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

Parameters:

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

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