Table Of Content

PhysicsActions 2
PhysicsActions.accelerateBus 2
PhysicsActions.deccelerateBus 3
PhysicsActions.moveBus 3
PhysicsEngine 4
PhysicsFormulas 5
PhysicsFormulas.TooManyNullArgumentsException 7
EditObjectUI 8
MainWindow 9
SoftwareUI 10
ObjectManager 11
PhysicsObject 12
RelationshipManager 15
RelationshipTypes 15
RoundObject 16
SquareObject 17
Index 20

Class PhysicsActions

```
java.lang.Object
    |
    +--edu.vu.vuse.cs278.g3.engine.PhysicsActions
```

< Constructors >

public class **PhysicsActions**
extends java.lang.Object

Constructors

PhysicsActions

```
public  PhysicsActions()
```

Class PhysicsActions.accelerateBus

```
java.lang.Object
    |
    +--edu.vu.vuse.cs278.g3.engine.PhysicsActions.accelerateBus
```

All Implemented Interfaces:
java.lang.Runnable

< Constructors > < Methods >

public class **PhysicsActions.accelerateBus**
extends java.lang.Object
implements java.lang.Runnable

Constructors

PhysicsActions.accelerateBus

```
public  PhysicsActions.accelerateBus()
```

Methods

run

```
public void run()
```

Class PhysicsActions.deccelerateBus

```
java.lang.Object
    |
    +--edu.vu.vuse.cs278.g3.engine.PhysicsActions.deccelerateBus
```

All Implemented Interfaces:
java.lang.Runnable

< Constructors > < Methods >

---

public class **PhysicsActions.deccelerateBus**
extends java.lang.Object
implements java.lang.Runnable

Constructors

PhysicsActions.deccelerateBus

```
public  PhysicsActions.deccelerateBus()
```

Methods

run

```
public void run()
```

Class PhysicsActions.moveBus

```
java.lang.Object
    |
    +--edu.vu.vuse.cs278.g3.engine.PhysicsActions.moveBus
```

All Implemented Interfaces:
java.lang.Runnable

< Constructors > < Methods >

public class **PhysicsActions.moveBus**
extends java.lang.Object
implements java.lang.Runnable

The bus itself should never move on the screen. We will simulate movement by adjusting the position of the background.

Author:

Matthew Shea

Constructors

PhysicsActions.moveBus

```
public  PhysicsActions.moveBus()
```

Methods

run

```
public void run()
```

Class PhysicsEngine

```
java.lang.Object
    |
    +--edu.vu.vuse.cs278.g3.engine.PhysicsEngine
```

< Methods >

public class **PhysicsEngine**
extends java.lang.Object

This class is a singleton wrapper around a QueueExecutor that runs the physics calculations and updates the display. Most of the work will be done on this QueueExecutor.

Author:

syddraf

Methods

addtoQueue

`public void` **`addtoQueue`**`(java.lang.Runnable runnable)`

Adds an item to the queue for the thread to execute.

Parameters:

runnable - PhysicsAction for the engine to perform

---

disable

`public void` **`disable`**`()`

Calling this method will allow the engine to finish it's current operation, and then suspend and flush the queue.

---

enable

`public void` **`enable`**`()`

This method will allow the engine to perform operations in its run queue.

---

getInstance

`public static` `PhysicsEngine` **`getInstance`**`()`

This method returns the singleton PhysicsEngine object.

Returns:

The singleton PhysicsEngine

Class PhysicsFormulas

```
java.lang.Object
    |
    +--edu.vu.vuse.cs278.g3.engine.PhysicsFormulas
```

< Constructors > < Methods >

---

public class **PhysicsFormulas**
extends java.lang.Object

Constructors

PhysicsFormulas

```
public  PhysicsFormulas()
```

Methods

frictionalForce

```
public static java.lang.Double frictionalForce(java.lang.Double f,
                                               java.lang.Double m,
                                               java.lang.Double u)
                                throws
PhysicsFormulas.TooManyNullArgumentsException
```

Implements the formula F = um; Pass one parameter as null and the calculated value of that parameter will be returned.

Parameters:

f - The frictional force applied to the object
m - The mass of the object
u - The coefficient of friction

Returns:

The returned value is the calculated value of the parameter that was passed as null.

Throws:

edu.vu.vuse.cs278.g3.engine.PhysicsFormulas.TooManyNullArgumentsException - Thrown if more than one argument is null.

---

momentum

```
public static java.lang.Double momentum(java.lang.Double P,
                                        java.lang.Double m,
                                        java.lang.Double v)
                                throws
PhysicsFormulas.TooManyNullArgumentsException
```

This function implements the formula for an objects momentum.

Parameters:

P - The momentum of the object.
m - The mass of the object.
v - The velocity of the object.

Returns:

The value returned is the argument that was set to null in the arguments.

Throws:

edu.vu.vuse.cs278.g3.engine.PhysicsFormulas.TooManyNullArgumentsException - Thrown if more than one argument is null.

Class PhysicsFormulas.TooManyNullArgumentsExcep

```
java.lang.Object
    |
    +--java.lang.Throwable
        |
        +--java.lang.Exception
            |
            +--edu.vu.vuse.cs278.g3.engine.PhysicsFormulas.TooManyNullArgumentsExce
```

All Implemented Interfaces:
java.io.Serializable

< Constructors > < Methods >

---

public static class **PhysicsFormulas.TooManyNullArgumentsException**
extends java.lang.Exception

This exception indicates that too many null arguments were passed to one of the above PhysicsFormulas.

Author:
Matthew Shea

Constructors

PhysicsFormulas.TooManyNullArgumentsException

```
public  PhysicsFormulas.TooManyNullArgumentsException(int number)
```

Methods

getError

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

Class EditObjectUI

```
java.lang.Object
    |
    +--java.awt.Component
        |
        +--java.awt.Container
            |
            +--java.awt.Window
                |
                +--java.awt.Frame
                    |
                    +--javax.swing.JFrame
                        |
                        +--edu.vu.vuse.cs278.g3.gui.EditObjectUI
```

All Implemented Interfaces:

java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.TransferHandler.HasGetTransferHandler, javax.swing.WindowConstants

---

< Constructors > < Methods >

---

public class **EditObjectUI**
extends javax.swing.JFrame

Author:

Amber Maria

Constructors

EditObjectUI

`public` **`EditObjectUI`**`()`

Creates new form EditObjectUI

Methods

main

`public static void` **`main`**`(java.lang.String[] args)`

Parameters:

args - the command line arguments

Class MainWindow

```
java.lang.Object
    |
    +--java.awt.Component
        |
        +--java.awt.Container
            |
            +--java.awt.Window
                |
                +--java.awt.Frame
                    |
                    +--javax.swing.JFrame
                        |
                        +--edu.vu.vuse.cs278.g3.gui.MainWindow
```

All Implemented Interfaces:

java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.TransferHandler.HasGetTransferHandler, javax.swing.WindowConstants

---

< Constructors > < Methods >

---

public class **MainWindow**
extends javax.swing.JFrame

Author:

Amber Maria

Constructors

MainWindow

`public` **`MainWindow`**`()`

Creates new form MainWindow

Methods

main

`public static void` **`main`**`(java.lang.String[] args)`

Parameters:

args - the command line arguments

Class SoftwareUI

```
java.lang.Object
    |
    +--java.awt.Component
        |
        +--java.awt.Container
            |
            +--java.awt.Window
                |
                +--java.awt.Frame
                    |
                    +--javax.swing.JFrame
                        |
                        +--edu.vu.vuse.cs278.g3.gui.SoftwareUI
```

All Implemented Interfaces:

java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.TransferHandler.HasGetTransferHandler, javax.swing.WindowConstants

---

< Constructors > < Methods >

---

public class **SoftwareUI**
extends javax.swing.JFrame

Author:

Brandon

Constructors

SoftwareUI

```
public  SoftwareUI()
```

Creates new form SoftwareUI

Methods

main

```
public static void main(java.lang.String[] args)
```

Parameters:

args - the command line arguments

Class ObjectManager

```
java.lang.Object
    |
    +--edu.vu.vuse.cs278.g3.model.ObjectManager
```

< Methods >

---

public class **ObjectManager**
extends java.lang.Object

Methods

addObject

```
public boolean addObject(java.lang.String id,
                         PhysicsObject object)
```

---

createCircle

```
public RoundObject createCircle(double _xCoord,
                                double _yCoord,
                                double _radius)
```

---

createSquare

```
public SquareObject createSquare(double _xCoord,
                                 double _yCoord,
                                 double _width,
                                 double _height)
```

---

getInstance

```
public static ObjectManager getInstance()
```

---

getObject

`public PhysicsObject getObject(java.lang.String id)`

---

removeObject

`public void removeObject(java.lang.String id)`

Class PhysicsObject

```
java.lang.Object
    |
    +--edu.vu.vuse.cs278.g3.model.PhysicsObject
```

Direct Known Subclasses:
RoundObject, SquareObject

< Fields > < Constructors > < Methods >

---

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

Fields

acceleration

`protected double acceleration`
In Pixels/Frame^2

---

speed

`protected double speed`
In Pixels/Frame

---

xCoord

`protected double xCoord`

---

yCoord

```
protected double yCoord
```

Constructors

PhysicsObject

```
protected  PhysicsObject(double xcoord,
                         double ycoord,
                         double speed,
                         double acceleration)
```

Methods

commit

```
public abstract void commit()
```

Commits the changes to the NetLogo backend to update the graphical display. You do not need to call this function unless the POSITION has changed.

---

getAcceleration

```
public double getAcceleration()
```

---

getSpeed

```
public double getSpeed()
```

---

getXCoord

```
public double getXCoord()
```

---

getYCoord

```
public double getYCoord()
```

---

setAcceleration

```
public void setAcceleration(double acc)
```

---

setShape

```
public abstract void setShape(java.lang.String shape)
```

Immediately changes the object's shape. Must exist in netlogo shapes library.

Parameters:

shape -

---

setSpeed

```
public void setSpeed(double iSpeed)
```

---

setXCoord

```
public void setXCoord(double xcoord)
```

---

setYCoord

```
public void setYCoord(double ycoord)
```

---

updatePosition

```
public void updatePosition(double xcoord,
                           double ycoord)
```

Class RelationshipManager

```
java.lang.Object
    |
    +--edu.vu.vuse.cs278.g3.model.RelationshipManager
```

< Methods >

public class **RelationshipManager**
extends java.lang.Object

Methods

getInstance

```
public static RelationshipManager getInstance()
```

Class RelationshipTypes

```
java.lang.Object
    |
    +--java.lang.Enum
        |
        +--edu.vu.vuse.cs278.g3.model.RelationshipTypes
```

All Implemented Interfaces:
java.io.Serializable, java.lang.Comparable

< Fields > < Methods >

public final class **RelationshipTypes**
extends java.lang.Enum

Fields

ABOVE_RESTRAINED

```
public static final RelationshipTypes ABOVE_RESTRAINED
```

ABOVE_UNRESTRAINED

public static final RelationshipTypes **ABOVE_UNRESTRAINED**

BEHIND_ATTACHED

public static final RelationshipTypes **BEHIND_ATTACHED**

INSIDE_RESTRAINED

public static final RelationshipTypes **INSIDE_RESTRAINED**

INSIDE_UNRESTRAINED

public static final RelationshipTypes **INSIDE_UNRESTRAINED**

Methods

valueOf

public static RelationshipTypes **valueOf**(java.lang.String name)

---

values

public static edu.vu.vuse.cs278.g3.model.RelationshipTypes[] **values**()

Class RoundObject

```
java.lang.Object
    |
    +--PhysicsObject
        |
        +--edu.vu.vuse.cs278.g3.model.RoundObject
```

< Methods >

---

public class **RoundObject**
extends PhysicsObject

Methods

commit

```
public void commit()
```

Overrides:

commit in class PhysicsObject

---

getRadius

```
public double getRadius()
```

---

setShape

```
public void setShape(java.lang.String shape)
```

Overrides:

setShape in class PhysicsObject

---

updatePosition

```
public void updatePosition(double xcoord,
                           double ycoord)
```

Overrides:

updatePosition in class PhysicsObject

Class SquareObject

```
java.lang.Object
    |
    +--PhysicsObject
        |
        +--edu.vu.vuse.cs278.g3.model.SquareObject
```

< Methods >

---

public class **SquareObject**
extends PhysicsObject

Methods

commit

`public void commit()`

Overrides:

commit in class PhysicsObject

---

getHeight

`public double getHeight()`

---

getWidth

`public double getWidth()`

---

getXCoord

`public double getXCoord()`

Overrides:

getXCoord in class PhysicsObject

---

getYCoord

`public double getYCoord()`

Overrides:

getYCoord in class PhysicsObject

---

setShape

`public void setShape(java.lang.String shape)`

Overrides:

setShape in class PhysicsObject

updatePosition

```
public void updatePosition(double xcoord,
                           double ycoord)
```

Overrides:

updatePosition in class PhysicsObject

INDEX

A

acceleration ... *12*
addObject ... *11*
addtoQueue ... *5*
ABOVE_RESTRAINED ... *15*
ABOVE_UNRESTRAINED ... *15*

B

BEHIND_ATTACHED ... *16*

C

commit ... *13*
commit ... *17*
commit ... *18*
createCircle ... *11*
createSquare ... *11*

D

disable ... *5*

E

enable ... *5*
EditObjectUI ... *8*
EditObjectUI ... *8*

F

frictionalForce ... *6*

G

getAcceleration ... *13*
getError ... *7*
getHeight ... *18*
getInstance ... *5*
getInstance ... *11*
getInstance ... *15*
getObject ... *12*
getRadius ... *17*
getSpeed ... *13*
getWidth ... *18*
getXCoord ... *13*
getXCoord ... *18*
getYCoord ... *14*
getYCoord ... *18*

I

INSIDE_RESTRAINED ... *16*
INSIDE_UNRESTRAINED ... *16*

M

main ... *8*
main ... *9*
main ... *10*
momentum ... *6*
MainWindow ... *9*
MainWindow ... *9*

O

ObjectManager ... *11*

P

PhysicsActions ... *2*
PhysicsActions ... *2*
PhysicsActions.accelerateBus ... *2*
PhysicsActions.accelerateBus ... *2*
PhysicsActions.deccelerateBus ... *3*
PhysicsActions.deccelerateBus ... *3*
PhysicsActions.moveBus ... *3*
PhysicsActions.moveBus ... *4*
PhysicsEngine ... *4*
PhysicsFormulas ... *5*
PhysicsFormulas ... *6*
PhysicsFormulas.TooManyNullArgumentsException ... *7*
PhysicsFormulas.TooManyNullArgumentsException ... *7*
PhysicsObject ... *12*
PhysicsObject ... *13*

R

removeObject ... *12*
run ... *3*
run ... *3*
run ... *4*
RelationshipManager ... *15*
RelationshipTypes ... *15*
RoundObject ... *16*

S

setAcceleration ... *14*
setShape ... *14*
setShape ... *17*
setShape ... *18*
setSpeed ... *14*
setXCoord ... *14*
setYCoord ... *14*
speed ... *12*
SoftwareUI ... *10*
SoftwareUI ... *10*
SquareObject ... *17*

U

updatePosition ... *14*
updatePosition ... *17*
updatePosition ... *19*

V

valueOf ... *16*
values ... *16*

X

xCoord ... *12*

Y

yCoord ... *13*