

Non-Sequential Project Javadoc Documentation



Group 08
Andrew Denner
James Yoder
Stephen Atterbury
Jason Lackore

Package dataView

dataView

Class CircularClip

java.lang.Object

↳ dataView.CircularClip

All Implemented Interfaces:

java.awt.Shape

```
public class CircularClip
  extends java.lang.Object
  implements java.awt.Shape
```

Constructor Summary

public	CircularClip (double center_x, double center_y, double radius)
--------	--

Method Summary

boolean	contains (double px, double py)
boolean	contains (double x, double y, double w, double h)
boolean	contains (java.awt.geom.Point2D arg0)
boolean	contains (java.awt.geom.Rectangle2D arg0)
java.awt.Rectangle	getBounds ()
java.awt.geom.Rectangle2D	getBounds2D ()
java.awt.geom.PathIterator	getPathIterator (java.awt.geom.AffineTransform arg0)
java.awt.geom.PathIterator	getPathIterator (java.awt.geom.AffineTransform arg0, double flatness)
boolean	intersects (double arg0, double arg1, double arg2, double arg3)
boolean	intersects (java.awt.geom.Rectangle2D arg0)

Constructors

(continued on next page)

(continued from last page)

CircularClip

```
public CircularClip(double center_x,  
                    double center_y,  
                    double radius)
```

Methods

contains

```
public boolean contains(java.awt.geom.Point2D arg0)
```

contains

```
public boolean contains(java.awt.geom.Rectangle2D arg0)
```

contains

```
public boolean contains(double px,  
                        double py)
```

contains

```
public boolean contains(double x,  
                        double y,  
                        double w,  
                        double h)
```

getBounds

```
public java.awt.Rectangle getBounds()
```

getBounds2D

```
public java.awt.geom.Rectangle2D getBounds2D()
```

getPathIterator

```
public java.awt.geom.PathIterator getPathIterator(java.awt.geom.AffineTransform arg0)
```

(continued on next page)

(continued from last page)

getPathIterator

```
public java.awt.geom.PathIterator getPathIterator(java.awt.geom.AffineTransform arg0,  
double flatness)
```

intersects

```
public boolean intersects(java.awt.geom.Rectangle2D arg0)
```

intersects

```
public boolean intersects(double arg0,  
double arg1,  
double arg2,  
double arg3)
```

dataView

Class Display2D

```

java.lang.Object
  |
  +- java.awt.Component
      |
      +- java.awt.Container
          |
          +- javax.swing.JComponent
              |
              +- dataView.Display2D
  
```

All Implemented Interfaces:

java.util.Observer, java.awt.event.ComponentListener, java.lang.Runnable, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable

public class **Display2D**

extends javax.swing.JComponent

implements java.io.Serializable, javax.swing.TransferHandler.HasGetTransferHandler, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.lang.Runnable, java.awt.event.ComponentListener, java.util.Observer

Constructor Summary

public	Display2D (ObservableSettingsData settings)
--------	--

Method Summary

void	animate (float timeDelta)
void	componentHidden (java.awt.event.ComponentEvent arg0)
void	componentMoved (java.awt.event.ComponentEvent arg0)
void	componentResized (java.awt.event.ComponentEvent arg0)
void	componentShown (java.awt.event.ComponentEvent arg0)
static java.awt.Image	create2DImage (java.lang.String fileName, java.awt.Component listener)
boolean	getDraggable ()
void	initialize (GameCard[] boardCards, int sizeX, int sizeY, int sizeZ, Move theMove)
GameCard[]	initialize (java.util.LinkedList cardTypeID, Move theMove)
void	paint (java.awt.Graphics g)

void	run()
void	setDraggable (boolean draggable)
void	update (java.util.Observable arg0, java.lang.Object arg1)

Constructors

Display2D

```
public Display2D(ObservableSettingsData settings)
```

Methods

getDraggable

```
public boolean getDraggable()
```

setDraggable

```
public void setDraggable(boolean draggable)
```

initialize

```
public void initialize(GameCard\[\] boardCards,  
    int sizeX,  
    int sizeY,  
    int sizeZ,  
    Move theMove)
```

initialize

```
public GameCard\[\] initialize(java.util.LinkedList cardTypeID,  
    Move theMove)
```

animate

```
public void animate(float timeDelta)
```

paint

```
public void paint(java.awt.Graphics g)
```

run

```
public void run()
```

create2DImage

```
public static java.awt.Image create2DImage(java.lang.String fileName,  
    java.awt.Component listener)
```

componentHidden

```
public void componentHidden(java.awt.event.ComponentEvent arg0)
```

componentMoved

```
public void componentMoved(java.awt.event.ComponentEvent arg0)
```

componentResized

```
public void componentResized(java.awt.event.ComponentEvent arg0)
```

componentShown

```
public void componentShown(java.awt.event.ComponentEvent arg0)
```

update

```
public void update(java.util.Observable arg0,  
    java.lang.Object arg1)
```

dataView

Class Display3D

```

java.lang.Object
  |-- java.awt.Component
        |-- java.awt.Canvas
              |-- javax.media.j3d.Canvas3D
                    |-- dataView.Display3D
  
```

All Implemented Interfaces:

java.util.Observer, java.lang.Runnable, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, javax.accessibility.Accessible

```

public class Display3D
  extends javax.media.j3d.Canvas3D
  implements javax.accessibility.Accessible, java.awt.image.ImageObserver, java.awt.MenuContainer,
  java.io.Serializable, java.lang.Runnable, java.util.Observer
  
```

Constructor Summary

public	Display3D(ObservableSettingsData settings) Default constructor.
--------	--

Method Summary

void	animate(float deltaTime) Animate the scene.
static javax.media.j3d.Texture	create3DImage(java.lang.String fileName) Creates a texture suitable for placement on 3D objects
void	initialize(GameCard[] boardCards, int size_x, int size_y, int size_z, Move theMove) Creates the geometry, moves the cards into place on the cube, initializes the display for gameplay.
void	run() Used for an animation thread that will animate the cards in the game
void	update(java.util.Observable arg0, java.lang.Object arg1)

Constructors

Display3D

```
public Display3D(ObservableSettingsData settings)
```

Default constructor. Initializes the Display3D device

Methods

animate

```
public void animate(float deltaTime)
```

Animate the scene. Change the colors of the cards when someone is hovering over the card, add tokens to the card, etc.

Parameters:

deltaTime - The change in time since this method was last called

run

```
public void run()
```

Used for an animation thread that will animate the cards in the game

initialize

```
public void initialize(GameCard\[\] boardCards,  
    int size_x,  
    int size_y,  
    int size_z,  
    Move theMove)
```

Creates the geometry, moves the cards into place on the cube, initializes the display for gameplay. Called once before each game.

Parameters:

data - The observable for Game Data

update

```
public void update(java.util.Observable arg0,  
    java.lang.Object arg1)
```

create3DImage

```
public static javax.media.j3d.Texture create3DImage(java.lang.String fileName)
```

Creates a texture suitable for placement on 3D objects

Parameters:

fileName - - The filename of the texture to load

dataView

Class Interpolaters

```
java.lang.Object
```

```
└-dataView.Interpolaters
```

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

Constructor Summary

public	Interpolaters()
--------	---------------------------------

Method Summary

static float	linear (float valc, float from, float to, float min, float max, float time_delta, float transition_time) A linear interpolator
--------------	---

Constructors

Interpolaters

```
public Interpolaters()
```

Methods

linear

```
public static float linear(float valc,
    float from,
    float to,
    float min,
    float max,
    float time_delta,
    float transition_time)
```

A linear interpolator

Parameters:

valc - The current value
 from - Where we're interpolating from
 to - Where we're interpolating to
 min - The minimum value that should be returned
 max - The maximum value that should be returned
 time_delta - How much time has passed? (In milliseconds)
 transition_time - How long should the animatoin take? (In seconds)

Returns:

(continued from last page)

The new, interpolated value of valc that is closer to the value of "to" such that after transition_time valc will be equal to "to"

dataView

Class MouseCheck3D

java.lang.Object

↳ **dataView.MouseCheck3D**

All Implemented Interfaces:

com.sun.j3d.utils.behaviors.mouse.MouseBehaviorCallback

```
public class MouseCheck3D
extends java.lang.Object
implements com.sun.j3d.utils.behaviors.mouse.MouseBehaviorCallback
```

Limits the zooming of the mouse so they can't see inside the cube or outside the background sphere

Constructor Summary

public	MouseCheck3D (javax.media.j3d.TransformGroup group)
--------	--

Method Summary

void	transformChanged (int type, javax.media.j3d.Transform3D matrix) Called when the zoom is chaned on a MouseZoom event
------	--

Constructors

MouseCheck3D

```
public MouseCheck3D( javax.media.j3d.TransformGroup group)
```

Parameters:

group - The group (The cube) that the MouseBefavior is acting on

Methods

transformChanged

```
public void transformChanged(int type,
    javax.media.j3d.Transform3D matrix)
```

Called when the zoom is chaned on a MouseZoom event

dataView

Class TextureManager

java.lang.Object

└-dataView.TextureManager

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

StoreContains all the textures for every card

Constructor Summary

public	TextureManager()
--------	----------------------------------

Method Summary

static javax.media.j3d.Texture re	getTexture (javax.swing.ImageIcon ref)
static Pair	getTexture (java.lang.String name, boolean create3DImage, java.awt.Component listener)
static Pair	getTexture (java.lang.String name, boolean create3DImage, java.awt.Component listener, boolean isCard) If name has already been loaded, then returns the 2D & 3D images (if requested) Otherwise, attempts to load the textures.
static Pair	getTextureByID (int ID, boolean create3DImage, java.awt.Component listener)
static int	getTextureID (java.lang.String path) Gets the ID associated with a texture path
static java.lang.String	getTexturePath (int ID) Gets the path to the texture

Constructors

TextureManager

public **TextureManager**()

Methods

(continued from last page)

getTexture

```
public static Pair getTexture(java.lang.String name,  
    boolean create3DImage,  
    java.awt.Component listener)
```

getTexture

```
public static javax.media.j3d.Texture getTexture(javax.swing.ImageIcon ref)
```

getTexture

```
public static Pair getTexture(java.lang.String name,  
    boolean create3DImage,  
    java.awt.Component listener,  
    boolean isCard)
```

If name has already been loaded, then returns the 2D & 3D images (if requested) Otherwise, attempts to load the textures.

Parameters:

name - The filename of the texture
create3DImage - Should the 3D texture be created?

Returns:

A pair of 3D/2D images if they can be successfully loaded.

getTextureByID

```
public static Pair getTextureByID(int ID,  
    boolean create3DImage,  
    java.awt.Component listener)
```

Parameters:

ID
create3DImage
listener

Returns:

getTexturePath

```
public static java.lang.String getTexturePath(int ID)
```

Gets the path to the texture

Parameters:

ID - The ID of the texture

Returns:

null if ID doesn't exist, or the path to the texture associated with ID

(continued from last page)

getTextureID

```
public static int getTextureID(java.lang.String path)
```

Gets the ID associated with a texture path

Parameters:

path - The path to look for

Returns:

-1 if the image at path hasn't been loaded, or the ID of the texture otherwise

Package
gameLogic

gameLogic Class Board

```
java.lang.Object
  |
  +-gameLogic.Board
```

All Implemented Interfaces:
java.io.Serializable

```
public class Board
extends java.lang.Object
implements java.io.Serializable
```

Board class for creating a box with row column and diagonal links to each other Is the power house for the game logic

Constructor Summary

public	Board (int xcor, int ycor, int zcor, Player ad, ObservableGameData ogdin) Constructor of the board
--------	---

Method Summary

boolean	check_move (Move request) Checks the specified move on the board to see if it is valid
GameCard[]	generateGameCardArray () Generates the game card array for OGD
int	perform_move (Move request) Performs the move in question.

Constructors

Board

```
public Board(int xcor,
             int ycor,
             int zcor,
             Player ad,
             ObservableGameData ogdin)
```

Constructor of the board

Parameters:

xcor - width from front
 ycor - depth from front
 zcor - height from front
 ad - admin
 ogdin - OGD data

Methods

(continued from last page)

generateGameCardArray

```
public GameCard\[\] generateGameCardArray()
```

Generates the game card array for OGD

Returns:

the game card array

check_move

```
public boolean check_move(Move request)
```

Checks the specified move on the board to see if it is valid

Parameters:

request - the move to be checked

Returns:

true if the move is valid for the specific players board

perform_move

```
public int perform_move(Move request)
```

Performs the move in question. Cannot be performed unless the move has been approved by the admin

Parameters:

request - move in question

Returns:

returns the number of wins from the move

gameLogic

Class BoardRandomizer

java.lang.Object

└-gameLogic.BoardRandomizer

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

Constructor Summary

public	BoardRandomizer()
--------	-----------------------------------

Method Summary

static java.util.LinkedList	buildUniverseSet (ObservableSettingsData settings)
static void	randomizeBoard (ObservableSettingsData settings, GameCard[] cards)

Constructors

BoardRandomizer

public **BoardRandomizer**()

Methods

buildUniverseSet

public static java.util.LinkedList **buildUniverseSet**([ObservableSettingsData](#) settings)

randomizeBoard

public static void **randomizeBoard**([ObservableSettingsData](#) settings,
[GameCard\[\]](#) cards)

gameLogic Class Node

java.lang.Object

↳-gameLogic.Node

All Implemented Interfaces:

java.io.Serializable

```
public class Node
extends java.lang.Object
implements java.io.Serializable
```

Node class which is used to reduce the amount of calculations needed to check if sequence Quadrant1 begins on the top left row, the middle is the current row

Constructor Summary

public	Node() Default constructor
public	Node(Node Quad1, Node Quad2, Node Quad3, Node Quad4, Node Quad5, Node Quad6, Node Quad7, Node Quad8) Constructor for Node that you already have all of the values for

Method Summary

boolean	check_mark() Checks if the Node is marked
int	check_sequence(int sequence) Recursive check at most sequence deep to see if there is a Sequence, checks diagonals and columns and rows
GameCard	getGameCard()
boolean	immutable_mark() Returns the team that the Node is owned by
int	immutableSet(int sequence)
void	mark_immutable() Marks the node as immutable
boolean	mark(int player, int team) Marks the Node, setting is_marked to true
boolean	set_card(GameCard in) Sets the card for the current Node
void	setCheck(boolean set) Sets the check to true or false

void	setQ1(Node set) Sets Q1
void	setQ2(Node set) Sets Q1
void	setQ3(Node set) Sets Q1
void	setQ4(Node set) Sets Q1
void	setQ5(Node set) Sets Q1
void	setQ6(Node set) Sets Q1
void	setQ7(Node set) Sets Q1
void	setQ8(Node set) Sets Q1

Constructors

Node

```
public Node()
```

Default constructor

Node

```
public Node(Node Quad1,
            Node Quad2,
            Node Quad3,
            Node Quad4,
            Node Quad5,
            Node Quad6,
            Node Quad7,
            Node Quad8)
```

Constructor for Node that you already have all of the values for

Parameters:

- Quad1 - top left
- Quad2 - top middle
- Quad3 - top right
- Quad4 - left
- Quad5 - right
- Quad6 - bottom left
- Quad7 - bottom middle
- Quad8 - bottom right

Methods

(continued from last page)

setQ1

```
public void setQ1(Node set)
```

Sets Q1

Parameters:

set

setQ2

```
public void setQ2(Node set)
```

Sets Q1

Parameters:

set

getGameCard

```
public GameCard getGameCard()
```

setQ3

```
public void setQ3(Node set)
```

Sets Q1

Parameters:

set

setQ4

```
public void setQ4(Node set)
```

Sets Q1

Parameters:

set

setQ5

```
public void setQ5(Node set)
```

Sets Q1

Parameters:

set

setQ6

```
public void setQ6(Node set)
```

Sets Q1

Parameters:

(continued from last page)

set

setQ7

```
public void setQ7(Node set)
```

Sets Q1

Parameters:

set

setQ8

```
public void setQ8(Node set)
```

Sets Q1

Parameters:

set

setCheck

```
public void setCheck(boolean set)
```

Sets the check to true or false

Parameters:

set

check_mark

```
public boolean check_mark()
```

Checks if the Node is marked

Returns:

true if marked, false if not marked

mark

```
public boolean mark(int player,  
int team)
```

Marks the Node, setting is_marked to true

Returns:

true if the Node is already marked

mark_immutable

```
public void mark_immutable()
```

Marks the node as immutable

immutable_mark

```
public boolean immutable_mark()
```

(continued from last page)

Returns the team that the Node is owned by

Returns:

the team ID, -1 if not owned

set_card

```
public boolean set_card(GameCard in)
```

Sets the card for the current Node

Parameters:

in - Node type

Returns:

true if the card for the Node is set

check_sequence

```
public int check_sequence(int sequence)
```

Recursive check at most sequence deep to see if there is a Sequence, checks diagonals and columns and rows

Returns:

1,2,3,4 (because there can be 4 wins off of one Node), 0 if not

immutableSet

```
public int immutableSet(int sequence)
```

Package gui

gui

Class AvatarChooserLabel

```

java.lang.Object
  |-- java.awt.Component
      |-- java.awt.Container
          |-- javax.swing.JComponent
              |-- javax.swing.JLabel
                  |-- gui.AvatarChooserLabel
  
```

All Implemented Interfaces:

```

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible,
javax.swing.SwingConstants
  
```

```

public class AvatarChooserLabel
extends javax.swing.JLabel
  
```

Constructor Summary

public	AvatarChooserLabel (javax.swing.JFileChooser chooser, ObservableSettingsData dataP)
--------	--

Method Summary

void	loadImage ()
void	paint (java.awt.Graphics g)
void	preview (java.io.File f)

Constructors

AvatarChooserLabel

```

public AvatarChooserLabel(javax.swing.JFileChooser chooser,
ObservableSettingsData dataP)
  
```

Methods

preview

```

public void preview(java.io.File f)
  
```

loadImage

```
public void loadImage()
```

paint

```
public void paint(java.awt.Graphics g)
```

gui

Class AvatarFilter

```
java.lang.Object
  |
  +- javax.swing.filechooser.FileFilter
    |
    +- gui.AvatarFilter
```

```
public class AvatarFilter
extends javax.swing.filechooser.FileFilter
```

Constructor Summary

public	AvatarFilter()
--------	--------------------------------

Method Summary

boolean	accept (java.io.File f)
java.lang.String	getDescription ()

Constructors

AvatarFilter

```
public AvatarFilter()
```

Methods

accept

```
public boolean accept(java.io.File f)
```

getDescription

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

gui

Class AvatarFilterUtil

java.lang.Object

└-gui.AvatarFilterUtil

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

Constructor Summary

public	AvatarFilterUtil()
--------	------------------------------------

Method Summary

static java.lang.String	getExtension (java.io.File f)
static boolean	isImage (java.lang.String ext)

Constructors

AvatarFilterUtil

```
public AvatarFilterUtil()
```

Methods

getExtension

```
public static java.lang.String getExtension(java.io.File f)
```

isImage

```
public static boolean isImage(java.lang.String ext)
```

gui

Class InGame

```

java.lang.Object
  |-- java.awt.Component
        |-- java.awt.Container
              |-- java.awt.Window
                    |-- java.awt.Frame
                          |-- javax.swing.JFrame
                                |-- gui.InGame
  
```

All Implemented Interfaces:

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

public class InGame

extends javax.swing.JFrame

implements javax.swing.WindowConstants, javax.accessibility.Accessible,

javax.swing.RootPaneContainer, javax.swing.TransferHandler.HasGetTransferHandler,

java.awt.MenuContainer, javax.accessibility.Accessible, java.awt.image.ImageObserver,

java.awt.MenuContainer, java.io.Serializable, java.util.Observer

This code was edited or generated using CloudGarden's Jigloo SWT/Swing GUI Builder, which is free for non-commercial use. If Jigloo is being used commercially (ie, by a corporation, company or business for any purpose whatever) then you should purchase a license for each developer using Jigloo. Please visit www.cloudgarden.com for details. Use of Jigloo implies acceptance of these licensing terms. A COMMERCIAL LICENSE HAS NOT BEEN PURCHASED FOR THIS MACHINE, SO JIGLOO OR THIS CODE CANNOT BE USED LEGALLY FOR ANY CORPORATE OR COMMERCIAL PURPOSE.

Constructor Summary

public	InGame (ObservableGameData dataP, ObservableChat chatP, Player thisPlayerP, ObservableTeamData teamsP, ObservableChatAction chatActP, ObservableSettingsData settings) Auto-generated main method to display this JFrame
--------	--

Method Summary

void	update (java.util.Observable obsP, java.lang.Object o)
------	--

Constructors

(continued from last page)

InGame

```
public InGame(ObservableGameData dataP,  
              ObservableChat chatP,  
              Player thisPlayerP,  
              ObservableTeamData teamsP,  
              ObservableChatAction chatActP,  
              ObservableSettingsData settings)
```

Auto-generated main method to display this JFrame

Methods

update

```
public void update(java.util.Observable obsP,  
                  java.lang.Object o)
```


gui

Class Lobby

```

java.lang.Object
  |-- java.awt.Component
      |-- java.awt.Container
          |-- java.awt.Window
              |-- java.awt.Frame
                  |-- javax.swing.JFrame
                      |-- gui.Lobby
  
```

All Implemented Interfaces:

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

public class **Lobby**

extends javax.swing.JFrame

implements javax.swing.WindowConstants, javax.accessibility.Accessible,

javax.swing.RootPaneContainer, javax.swing.TransferHandler.HasGetTransferHandler,

java.awt.MenuContainer, javax.accessibility.Accessible, java.awt.image.ImageObserver,

java.awt.MenuContainer, java.io.Serializable, java.util.Observer

This code was edited or generated using CloudGarden's Jigloo SWT/Swing GUI Builder, which is free for non-commercial use. If Jigloo is being used commercially (ie, by a corporation, company or business for any purpose whatever) then you should purchase a license for each developer using Jigloo. Please visit www.cloudgarden.com for details. Use of Jigloo implies acceptance of these licensing terms. A COMMERCIAL LICENSE HAS NOT BEEN PURCHASED FOR THIS MACHINE, SO JIGLOO OR THIS CODE CANNOT BE USED LEGALLY FOR ANY CORPORATE OR COMMERCIAL PURPOSE.

Constructor Summary

public	Lobby (ObservableChat chatP, Player thisPlayerP, ObservableSettingsData settingsP, ObservableRoomData roomsP)
--------	--

Method Summary

static void	main (java.lang.String[] args) Auto-generated main method to display this JFrame
void	update (java.util.Observable obs, java.lang.Object o)

Constructors

(continued from last page)

Lobby

```
public Lobby(ObservableChat chatP,  
            Player thisPlayerP,  
            ObservableSettingsData settingsP,  
            ObservableRoomData roomsP)
```

Methods

main

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

Auto-generated main method to display this JFrame

update

```
public void update(java.util.Observable obs,  
                  java.lang.Object o)
```

gui

Class PreGame

```

java.lang.Object
  |-- java.awt.Component
      |-- java.awt.Container
          |-- java.awt.Window
              |-- java.awt.Frame
                  |-- javax.swing.JFrame
                      |-- gui.PreGame
  
```

All Implemented Interfaces:

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

public class **PreGame**

extends javax.swing.JFrame

implements javax.swing.WindowConstants, javax.accessibility.Accessible,

javax.swing.RootPaneContainer, javax.swing.TransferHandler.HasGetTransferHandler,

java.awt.MenuContainer, javax.accessibility.Accessible, java.awt.image.ImageObserver,

java.awt.MenuContainer, java.io.Serializable, java.util.Observer

This code was edited or generated using CloudGarden's Jigloo SWT/ Swing GUI Builder, which is free for non-commercial use. If Jigloo is being used commercially (ie, by a corporation, company or business for any purpose whatever) then you should purchase a license for each developer using Jigloo. Please visit www.cloudgarden.com for details. Use of Jigloo implies acceptance of these licensing terms. A COMMERCIAL LICENSE HAS NOT BEEN PURCHASED FOR THIS MACHINE, SO JIGLOO OR THIS CODE CANNOT BE USED LEGALLY FOR ANY CORPORATE OR COMMERCIAL PURPOSE.

Constructor Summary

public	PreGame(RoomClass roomP, ObservableChat chatP, ObservableChatAction actionP, ObservableTeamData teamP, ObservableSettingsData settings, Player thisPlayerP, ObservablePreGameData dataP)
--------	--

Method Summary

void	createTree (javax.swing.tree.DefaultMutableTreeNode root)
Player	getPlayer (java.lang.Object o)
boolean	isPlayer (java.lang.Object o)
boolean	isTeam (java.lang.Object o)
static void	main (java.lang.String[] args) Auto-generated main method to display this JFrame
void	update (java.util.Observable obsP, java.lang.Object o)

Constructors

PreGame

```
public PreGame(RoomClass roomP,  
               ObservableChat chatP,  
               ObservableChatAction actionP,  
               ObservableTeamData teamP,  
               ObservableSettingsData settings,  
               Player thisPlayerP,  
               ObservablePreGameData dataP)
```

Methods

main

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

Auto-generated main method to display this JFrame

update

```
public void update(java.util.Observable obsP,  
                  java.lang.Object o)
```

createTree

```
public void createTree(javax.swing.tree.DefaultMutableTreeNode root)
```

isPlayer

```
public boolean isPlayer(java.lang.Object o)
```

isTeam

```
public boolean isTeam(java.lang.Object o)
```

getPlayer

```
public Player getPlayer(java.lang.Object o)
```

gui

Class Settings

```

java.lang.Object
  |-- java.awt.Component
      |-- java.awt.Container
          |-- java.awt.Window
              |-- java.awt.Frame
                  |-- javax.swing.JFrame
                      |-- gui.Settings

```

All Implemented Interfaces:

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

public class **Settings**

extends `javax.swing.JFrame`

implements `javax.swing.WindowConstants`, `javax.accessibility.Accessible`,

`javax.swing.RootPaneContainer`, `javax.swing.TransferHandler.HasGetTransferHandler`,

`java.awt.MenuContainer`, `javax.accessibility.Accessible`, `java.awt.image.ImageObserver`,

`java.awt.MenuContainer`, `java.io.Serializable`, `java.util.Observer`

This code was edited or generated using CloudGarden's Jigloo SWT/Swing GUI Builder, which is free for non-commercial use. If Jigloo is being used commercially (ie, by a corporation, company or business for any purpose whatever) then you should purchase a license for each developer using Jigloo. Please visit www.cloudgarden.com for details. Use of Jigloo implies acceptance of these licensing terms. A COMMERCIAL LICENSE HAS NOT BEEN PURCHASED FOR THIS MACHINE, SO JIGLOO OR THIS CODE CANNOT BE USED LEGALLY FOR ANY CORPORATE OR COMMERCIAL PURPOSE.

Constructor Summary

public	Settings (ObservableSettingsData dataP)
--------	--

Method Summary

static void	main (java.lang.String[] args) Auto-generated main method to display this JFrame
void	update (java.util.Observable obs, java.lang.Object o)

Constructors

Settings

public **Settings**([ObservableSettingsData](#) dataP)

(continued from last page)

Methods

main

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

Auto-generated main method to display this JFrame

update

```
public void update(java.util.Observable obs,  
    java.lang.Object o)
```

gui

Class TreeIconRenderer

```

java.lang.Object
  |-- java.awt.Component
    |-- java.awt.Container
      |-- javax.swing.JComponent
        |-- javax.swing.JLabel
          |-- javax.swing.tree.DefaultTreeCellRenderer
            |-- gui.TreeIconRenderer
  
```

All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible, javax.swing.SwingConstants, javax.swing.tree.TreeCellRenderer

```

public class TreeIconRenderer
  extends javax.swing.tree.DefaultTreeCellRenderer
  
```

Constructor Summary

public	TreeIconRenderer()
--------	------------------------------------

Method Summary

java.awt.Component	getTreeCellRendererComponent (javax.swing.JTree tree, java.lang.Object value, boolean sel, boolean expanded, boolean leaf, int row, boolean hasFocus)
--------------------	---

Constructors

TreeIconRenderer

```
public TreeIconRenderer()
```

Methods

getTreeCellRendererComponent

```

public java.awt.Component getTreeCellRendererComponent(javax.swing.JTree tree,
    java.lang.Object value,
    boolean sel,
    boolean expanded,
    boolean leaf,
    int row,
    boolean hasFocus)
  
```

(continued from last page)

gui

Class TreeTransferable

```

java.lang.Object
  |-- java.awt.Component
    |-- java.awt.Container
      |-- javax.swing.JComponent
        |-- javax.swing.JTree
          |-- gui.TreeTransferable

```

All Implemented Interfaces:

java.awt.dnd.DragGestureListener, java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible, javax.swing.Scrollable

public class **TreeTransferable**

extends javax.swing.JTree

implements javax.swing.Scrollable, javax.accessibility.Accessible, java.io.Serializable, javax.swing.TransferHandler.HasGetTransferHandler, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.awt.dnd.DragGestureListener

Constructor Summary

public	TreeTransferable()
public	TreeTransferable (javax.swing.tree.DefaultTreeModel model, ObservableTeamData teamsP, RoomClass roomP)

Method Summary

void	dragGestureRecognized (java.awt.dnd.DragGestureEvent dge)
void	repaint ()

Constructors

TreeTransferable

public **TreeTransferable**()

TreeTransferable

public **TreeTransferable**(javax.swing.tree.DefaultTreeModel model, [ObservableTeamData](#) teamsP, [RoomClass](#) roomP)

Methods

dragGestureRecognized

```
public void dragGestureRecognized(java.awt.dnd.DragGestureEvent dge)
```

repaint

```
public void repaint()
```

Package
lobbyServer

lobbyServer

Class LobbyRoomThread

```
java.lang.Object
  |
  +- java.lang.Thread
      |
      +- lobbyServer.LobbyRoomThread
```

All Implemented Interfaces:

java.lang.Runnable, java.lang.Runnable

```
public class LobbyRoomThread
  extends java.lang.Thread
  implements java.lang.Runnable, java.lang.Runnable
```

this is a thread that fires the room list down the pipe every 5 seconds

Method Summary

void	run() thread that sends room list every 5 seconds
------	--

Methods

run

```
public void run()
```

thread that sends room list every 5 seconds

lobbyServer

Class LobbyServer

```

java.lang.Object
  |
  +- java.lang.Thread
      |
      +- lobbyServer.LobbyServer
  
```

All Implemented Interfaces:

java.util.Observer, java.lang.Runnable

```

public class LobbyServer
extends java.lang.Thread
implements java.lang.Runnable, java.util.Observer
  
```

This is the game lobby server it is responsible for running the lobby server... this is a console based standalone server that listens for connections on port 6900 and then will act as the coordinator to pass clients to game servers

Method Summary

java.net.InetAddress	getIp() gets IP that isn't localhost
static void	main (java.lang.String[] args) main process that is run by starting this object
void	update (java.util.Observable arg0, java.lang.Object arg1)

Methods

update

```

public void update(java.util.Observable arg0,
                  java.lang.Object arg1)
  
```

getIp

```

public java.net.InetAddress getIp()
throws java.net.SocketException,
       java.net.UnknownHostException
  
```

gets IP that isn't localhost

Returns:

InetAddress

Throws:

SocketException

UnknownHostException

main

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

main process that is run by starting this object

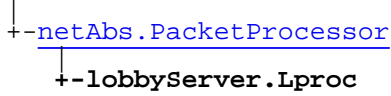
Parameters:

args

lobbyServer

Class Lproc

java.lang.Object



All Implemented Interfaces:

java.util.Observer

```
public class Lproc
extends PacketProcessor
```

This is the Lproc lobby server packet processor it is responsible for handling all downbound packets.

Constructor Summary

public	Lproc (ObservableChat c, ObservableRoomData r) creates new Lproc with pointers to observable chat and roomdata
--------	--

Method Summary

void	update (java.util.Observable obs, java.lang.Object o) called when new items are added to the downstack
------	---

Constructors

Lproc

```
public Lproc(ObservableChat c,
             ObservableRoomData r)
```

creates new Lproc with pointers to observable chat and roomdata

Parameters:

c
r

Methods

update

```
public void update(java.util.Observable obs,
                  java.lang.Object o)
```

called when new items are added to the downstack

Package
netAbs

netAbs Class Client

```

java.lang.Object
  |
  +- java.lang.Thread
      |
      +- netAbs.Client
  
```

All Implemented Interfaces:
java.lang.Runnable

```

public class Client
extends java.lang.Thread
  
```

Abstract Client class connects to Inet, port 6900

Constructor Summary

public	Client (PacketStore u, PacketStore d, java.net.InetAddress i) creates client from provided objects
--------	--

Method Summary

boolean	connect () connects to Inetaddress provided in constructor
PacketStore	getDownstack () provides access to the downstack
PacketStore	getUpstack () provides access to the upstack
boolean	shutdown () destroys all open connections, stops all threads, and cleans up (or as far as java lets us

Constructors

Client

```

public Client(PacketStore u,
             PacketStore d,
             java.net.InetAddress i)
  
```

creates client from provided objects

Parameters:

```

u
d
i
  
```

Methods

(continued from last page)

connect

```
public boolean connect()
```

connects to InetAddress provided in constructor

Returns:

true upon completion

shutdown

```
public boolean shutdown()
```

destroys all open connections, stops all threads, and cleans up (or as far as java lets us

Returns:

true upon finish

getUpstack

```
public PacketStore getUpstack()
```

provides access to the upstack

Returns:

packetstore

getDownstack

```
public PacketStore getDownstack()
```

provides access to the downstack

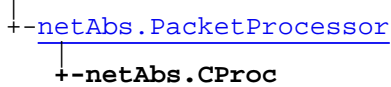
Returns:

packetstore

netAbs

Class CProc

java.lang.Object



All Implemented Interfaces:

java.util.Observer

```
public class CProc
extends PacketProcessor
```

Processes inbound packets on the client side this will take a packet determine what items need to be worked on and then do it

Constructor Summary

public	CProc (ObservableRoomData roomsP, ObservableChat chatP, ObservableChatAction actionsP, Move moveP, ObservablePreGameData preGameP, RoomClass roomP, ObservableTeamData teams, Player me) initializes cproc with provided data
--------	---

Method Summary

void	setGameData (ObservableGameData dataP)
void	update (java.util.Observable obs, java.lang.Object o) called everytime there is a new packet recieved from the downstream basically a huge if block

Constructors

CProc

```
public CProc(ObservableRoomData roomsP,
             ObservableChat chatP,
             ObservableChatAction actionsP,
             Move moveP,
             ObservablePreGameData preGameP,
             RoomClass roomP,
             ObservableTeamData teams,
             Player me)
```

initializes cproc with provided data

Parameters:

roomsP
 chatP
 actionsP
 moveP
 preGameP
 roomP
 teams

(continued from last page)

me

Methods

update

```
public void update(java.util.Observable obs,  
                  java.lang.Object o)
```

called everytime there is a new packet recieved from the downstream basically a huge if block

setGameData

```
public void setGameData(ObservableGameData dataP)
```

netAbs Class Down

```

java.lang.Object
  |
  +- java.lang.Thread
      |
      +- netAbs.Down
  
```

All Implemented Interfaces:

java.lang.Runnable, java.lang.Runnable

public class **Down**
 extends java.lang.Thread
 implements java.lang.Runnable, java.lang.Runnable

down bound network connection. this listens for packets from the network and then translates them out and sneds them to the downstack

Constructor Summary

public	Down (java.net.Socket s, ThreadTracker t, PacketStore u) initializes the downbound connection and all of the data values for them
--------	--

Method Summary

void	run () this thread listens to the inputstream and waits for new data to come down it then breaks apart the packet and sends it's payload to the downstack.
------	---

Constructors

Down

```

public Down(java.net.Socket s,
            ThreadTracker t,
            PacketStore u)
  
```

initializes the downbound connection and all of the data values for them

Parameters:

s
t
u

Methods

run

```

public void run()
  
```

this thread listens to the inputstream and waits for new data to come down it then breaks apart the packet and sends it's payload to the downstack.

netAbs

Class PacketProcessor

java.lang.Object

↳ **netAbs.PacketProcessor**

All Implemented Interfaces:

java.util.Observer

Direct Known Subclasses:

[CProc](#), [SProc](#), [Lproc](#)

```
public class PacketProcessor
extends java.lang.Object
implements java.util.Observer
```

this is the super of all of the proc classes

Constructor Summary

public	PacketProcessor()
--------	-----------------------------------

Method Summary

java.lang.Object	GetPayload() get object in the current packet
java.lang.String	getString() get what kind of action this packet is
void	listento() (java.util.Observable o) register this class to observable object listed
void	stoplistening() unregister me with object o as an observable
void	update() (java.util.Observable arg0, java.lang.Object arg1) update to be set by children of the object

Constructors

PacketProcessor

```
public PacketProcessor()
```

Methods

(continued from last page)

listento

```
public void listento(java.util.Observable o)
```

register this class to observable object listed

Parameters:

- o

stoplistening

```
public void stoplistening()
```

unregister me with object o as an observable

getString

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

get what kind of action this packet is

Returns:

string

GetPayload

```
public java.lang.Object GetPayload()
```

get object in the current packet

Returns:

object

update

```
public void update(java.util.Observable arg0,  
                  java.lang.Object arg1)
```

update to be set by children of the object

netAbs

Class PacketStore

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- netAbs.PacketStore
  
```

```

public class PacketStore
extends java.util.Observable
  
```

An observable stack of packets

Constructor Summary

public	PacketStore()
--------	-------------------------------

Method Summary

java.lang.Object	get() returns top item from stack.
void	put (java.lang.Object o) this pushes packet into stack then notify everyone...

Constructors

PacketStore

```
public PacketStore()
```

Methods

put

```
public void put(java.lang.Object o)
```

this pushes packet into stack then notify everyone... after every one who is watching is done then we pop back off the stack

Parameters:

o

get

```
public java.lang.Object get()
```

returns top item from stack.

Returns:

(continued from last page)

netAbs Class Server

```

java.lang.Object
  |
  +- java.lang.Thread
      |
      +- netAbs.Server
  
```

All Implemented Interfaces:
java.lang.Runnable

```

public class Server
extends java.lang.Thread
  
```

This is the abstract Server Class To use Call Server (Upstore, Downstore) and then StartListening

Constructor Summary

public	Server (PacketStore U, PacketStore D) Constructor...
--------	--

Method Summary

PacketStore	getDownstack () usefull for down.addObserver()
PacketStore	getUpstack () usefull for up.addObserver()
static void	main (java.lang.String[] args) for testing purposes
java.lang.Object	recieve () same as picking up an item from the down stack...
void	send (java.lang.Object o) same as adding a object to the upstack
boolean	shutdown () destroys all open connections, stops all threads, and cleans up (or as far as java lets us
void	StartListening () Startlistening to port 6900
void	StopListening () stop listening for new connections

Constructors

(continued from last page)

Server

```
public Server(PacketStore U,  
              PacketStore D)
```

Constructor...

Parameters:

- U - Up PacketStore
- D - Down Packet Store

Methods

send

```
public void send(java.lang.Object o)
```

same as adding a object to the upstack

Parameters:

- o

recieve

```
public java.lang.Object recieve()
```

same as picking up an item from the down stack... you wont be notified when things come in so this is not so useful

Returns:

object

StartListening

```
public void StartListening()
```

Startlistening to port 6900

StopListening

```
public void StopListening()
```

stop listening for new connections

shutdown

```
public boolean shutdown()
```

destroys all open connections, stops all threads, and cleans up (or as far as java lets us

Returns:

true upon finish

getUpstack

```
public PacketStore getUpstack()
```

usefull for up.addObserver()

(continued from last page)

Returns:

upstack

getDownstack

```
public PacketStore getDownstack()
```

usefull for down.addObserver()

Returns:

downstack

main

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

for testing purposes

Parameters:

args

netAbs

Class ServerListener

```

java.lang.Object
  |
  +- java.lang.Thread
      |
      +- netAbs.ServerListener
  
```

All Implemented Interfaces:

java.lang.Runnable, java.lang.Runnable

```

public class ServerListener
extends java.lang.Thread
implements java.lang.Runnable, java.lang.Runnable
  
```

Listens for new connections as it's own thread

Constructor Summary

public	ServerListener (ThreadTracker l, ThreadTracker t, PacketStore u, PacketStore d) initializes server listener to data provided
--------	--

Method Summary

void	run () thread that listens for new connections and spawns off a new server up/down thread proess
void	shutdown () shuts down the listening thread

Constructors

ServerListener

```

public ServerListener(ThreadTracker l,
                     ThreadTracker t,
                     PacketStore u,
                     PacketStore d)
  
```

initializes server listener to data provided

Parameters:

l
t
u
d

Methods

(continued from last page)

shutdown

```
public void shutdown()
```

shuts down the listening thread

run

```
public void run()
```

thread that listens for new connections and spawns off a new server up/down thread process

netAbs

Class ServerThread

```

java.lang.Object
  |
  +- java.lang.Thread
      |
      +- netAbs.ServerThread
  
```

All Implemented Interfaces:
 java.lang.Runnable, java.lang.Runnable

```

public class ServerThread
  extends java.lang.Thread
  implements java.lang.Runnable, java.lang.Runnable
  
```

This class handles the creation of the up/down infrastructure

Constructor Summary

public	ServerThread (java.net.Socket s, ThreadTracker t, PacketStore u, PacketStore d) handles the initialization of the class with provided data
--------	--

Method Summary

void	start () thread that spawns the up down reader writer structure
------	--

Constructors

ServerThread

```

public ServerThread(java.net.Socket s,
                   ThreadTracker t,
                   PacketStore u,
                   PacketStore d)
  
```

handles the initialization of the class with provided data

Parameters:

s
t
u
d

Methods

start

```

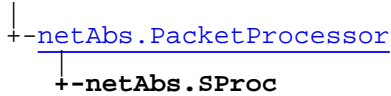
public void start()
  
```

thread that spawns the up down reader writer structure

netAbs

Class SProc

java.lang.Object



All Implemented Interfaces:
java.util.Observer

```
public class SProc
extends PacketProcessor
```

Server Processor processes all outbound objects into packets

Constructor Summary

public	SProc()
public	SProc (RoomClass roomP, ObservableChat chatP, ObservableTeamData teamsP, Move moveP)

Method Summary

void	update (java.util.Observable obs, java.lang.Object o) massive ifblock that is basically a packet factory called each time an observable is updated
------	---

Constructors

SProc

```
public SProc()
```

SProc

```
public SProc(RoomClass roomP,
            ObservableChat chatP,
            ObservableTeamData teamsP,
            Move moveP)
```

Methods

update

```
public void update(java.util.Observable obs,
                  java.lang.Object o)
```


(continued from last page)

massive ifblock that is basically a packet factory called each time an observable is updated

netAbs

Class ThreadTracker

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- netAbs.ThreadTracker
  
```

```

public class ThreadTracker
extends java.util.Observable
  
```

a crude rude little hashmap that tracks the current running threads and allows anyone who has this object to kill all the threads that are associated with it uses the deprecated thread.kill but it works so ... Warning: This is in no way shape or form safe. The author takes no responsibility if this breaks things, or kills baby puppies or etc...

Method Summary

void	add (int id, java.lang.Thread t) adds new thread to hashmap
java.util.HashMap	getSt () get hashmap st
void	killall () kills all threads from the stack
void	remove (int id) removes thread #id from the thread
void	setSt (java.util.HashMap st) set hahmap st

Methods

add

```

public void add(int id,
                java.lang.Thread t)
  
```

adds new thread to hashmap

Parameters:

```

id
t
  
```

remove

```

public void remove(int id)
  
```

removes thread #id from the thread

Parameters:

```

id
  
```

killall

```
public void killall()
```

kills all threads from the stack

setSt

```
public void setSt(java.util.HashMap st)
```

set hahmap st

Parameters:

st

getSt

```
public java.util.HashMap getSt()
```

get hashmap st

Returns:

netAbs Class Up

```
java.lang.Object
  |
  +- java.lang.Thread
      |
      +- netAbs.Up
```

All Implemented Interfaces:

java.util.Observer, java.lang.Runnable, java.lang.Runnable

public class **Up**
 extends java.lang.Thread
 implements java.lang.Runnable, java.lang.Runnable, java.util.Observer

Upthread handles all upbound writing to the network from packetstack

Constructor Summary

public	Up (java.net.Socket s, ThreadTracker t, PacketStore u) creates upbound connection and initializes all the values for it
--------	--

Method Summary

void	run () This maintains connection and then sleeps a lot.
void	update (java.util.Observable o, java.lang.Object arg) called by upstack everytime there is something placed on it this writes the packet from upstack to the network

Constructors

Up

```
public Up(java.net.Socket s,  

  ThreadTracker t,  

  PacketStore u)
```

creates upbound connection and initializes all the values for it

Parameters:

s
t
u

Methods

run

```
public void run()
```

(continued from last page)

This maintains connection and then sleeps a lot. May not be needed soon

update

```
public void update(java.util.Observable o,  
                  java.lang.Object arg)
```

called by upstack everytime there is something placed on it this writes the packet from upstack to the network

Package network

network

Class LobbyClient

java.lang.Object

↳ **network.LobbyClient**

All Implemented Interfaces:

java.util.Observer

```
public class LobbyClient
extends java.lang.Object
implements java.util.Observer
```

Lobby client is the client responsible for running the connection to the lobby server this will spon off a client connection and all the processes that belong to it.

Constructor Summary

public	LobbyClient (ObservableChat c, ObservableChatAction a, ObservableSettingsData settings, ObservableRoomData rooms) initializes the lobby client.
--------	---

Method Summary

void	killConnection () shuts down the connection
void	update (java.util.Observable arg0, java.lang.Object arg1) handles inbound packets placed on the down stack

Constructors

LobbyClient

```
public LobbyClient(ObservableChat c,
ObservableChatAction a,
ObservableSettingsData settings,
ObservableRoomData rooms)
```

initilizes the lobby client.

Parameters:

c
 a
 settings
 rooms

Methods

(continued from last page)

update

```
public void update(java.util.Observable arg0,  
                  java.lang.Object arg1)
```

handles inbound packets placed on the down stack

killConnection

```
public void killConnection()
```

shuts down the connection

network

Class objectHandler

java.lang.Object

↳-network.objectHandler

All Implemented Interfaces:

java.util.Observer

public class **objectHandler**
 extends java.lang.Object
 implements java.util.Observer

Handles all sending and receiving of network items

Constructor Summary

public	objectHandler (SnetworkStack nsIn, SnetworkStack nsOut) sets in and out network stacks...
--------	---

Method Summary

void	send (java.lang.Object O, java.lang.String t) sends a object
void	update (java.util.Observable arg0, java.lang.Object arg1) right now just a stub, not used

Constructors

objectHandler

public **objectHandler**([SnetworkStack](#) nsIn,
[SnetworkStack](#) nsOut)

sets in and out network stacks... right now not used

Parameters:

nsIn

nsOut

Methods

send

public void **send**(java.lang.Object O,
 java.lang.String t)

sends a object

(continued from last page)

update

```
public void update(java.util.Observable arg0,  
                  java.lang.Object arg1)
```

right now just a stub, not used

network

Class PacketFactory

java.lang.Object

└-network.PacketFactory

Deprecated. *we no longer use this class*

public class **PacketFactory**

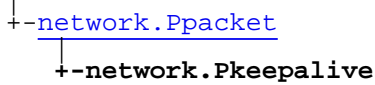
extends java.lang.Object

Creates new packet from item passed and then will inject it into the upbound stack

network

Class Pkeepalive

java.lang.Object



All Implemented Interfaces:

java.io.Serializable, java.io.Serializable

```

public class Pkeepalive
  extends Ppacket
  implements java.io.Serializable, java.io.Serializable
  
```

Constructor Summary

public	Pkeepalive()
--------	------------------------------

Method Summary

boolean	isReply() if is a "Pong" or reply then we should not respond back with another packet.
void	setReply(boolean b) sets the reply to b.

Constructors

Pkeepalive

```
public Pkeepalive()
```

Methods

isReply

```
public boolean isReply()
```

if is a "Pong" or reply then we should not respond back with another packet.

setReply

```
public void setReply(boolean b)
```

sets the reply to b. False for Pong True for ping

network

Class Ppacket

java.lang.Object
 |
 +-network.Ppacket

All Implemented Interfaces:
 java.io.Serializable

Direct Known Subclasses:
[Pkeepalive](#), [Pstandard](#)

```
public class Ppacket
extends java.lang.Object
implements java.io.Serializable
```

This is the generalization of all packets

Constructor Summary

public	Ppacket()
--------	---------------------------

Method Summary

int	gettype() this will return what kind of packet this is 0 Keep alive packet "ping/pong" 1 regular packet
void	settype(int t) this sets packet type right now there are the following options: 0 Keep alive packet "ping/pong" 1 regular packet

Constructors

Ppacket

```
public Ppacket()
```

Methods

gettype

```
public int gettype()
```

this will return what kind of packet this is 0 Keep alive packet "ping/pong" 1 regular packet

settype

```
public void settype(int t)
```

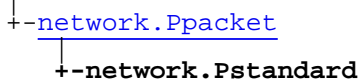
(continued from last page)

this sets packet type right now there are the following options: 0 Keep alive packet "ping/pong" 1 regular packet

network

Class Pstandard

java.lang.Object



All Implemented Interfaces:

java.io.Serializable, java.io.Serializable

```

public class Pstandard
  extends Ppacket
  implements java.io.Serializable, java.io.Serializable
  
```

Constructor Summary

public	Pstandard()
--------	-----------------------------

Method Summary

java.lang.Object	getObject() returns payload object.
java.lang.String	getObjectType() returns object type as a String
void	setObject(java.lang.Object ob, java.lang.String type) sets serializable payload and type.
java.lang.String	toString()

Constructors

Pstandard

```
public Pstandard()
```

Methods

setObject

```
public void setObject(java.lang.Object ob,
  java.lang.String type)
```

sets serializable payload and type. type is the class name.

(continued from last page)

getObject

```
public java.lang.Object getObject()
```

returns payload object. you must cast this into an object type to use

getObjectType

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

returns object type as a String

toString

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


network

Class RoomClient

```
java.lang.Object
  |
  +-network.RoomClient
```

All Implemented Interfaces:

```
java.util.Observer
```

```
public class RoomClient
  extends java.lang.Object
  implements java.util.Observer
```

room client is the network portion for the client user that is currently playing the game in a game room

Constructor Summary

public	RoomClient (ObservableChat chatP, Player thisPlayerP, ObservableSettingsData settingsP, Move moveP, ObservableChatAction actionsP, ObservableRoomData roomsP, ObservablePreGameData preGameP, ObservableTeamData teams) constructor for the roomclient.
public	RoomClient () Test constructor...

Method Summary

boolean	connect (RoomClass r) connect to room r
void	setGameData (ObservableGameData dataP) when we have the gamedata set we can set it using this
void	update (java.util.Observable o, java.lang.Object arg) update listens to the observable data and adds the object to the outbound network stack

Constructors

RoomClient

```
public RoomClient(ObservableChat chatP,
                 Player thisPlayerP,
                 ObservableSettingsData settingsP,
                 Move moveP,
                 ObservableChatAction actionsP,
                 ObservableRoomData roomsP,
                 ObservablePreGameData preGameP,
                 ObservableTeamData teams)
```

constructor for the roomclient. Pass in all the observables that we need to watch

Parameters:

chatP

(continued from last page)

```
thisPlayerP
settingsP
moveP
actionsP
roomsP
preGameP
teams
```

RoomClient

```
public RoomClient()
```

Test constructor... this is not used in regular game play creates new dummy objects

Methods

connect

```
public boolean connect(RoomClass r)
```

connect to room r

Parameters:

r

Returns:

update

```
public void update(java.util.Observable o,
    java.lang.Object arg)
```

update listens to the observable data and adds the object to the outbound network stack

setGameData

```
public void setGameData(ObservableGameData dataP)
```

when we have the gamedata set we can set it using this

Parameters:

dataP

network

Class RoomServer

```

java.lang.Object
  |
  +- java.lang.Thread
      |
      +- network.RoomServer
  
```

All Implemented Interfaces:

java.util.Observer, java.lang.Runnable

```

public class RoomServer
extends java.lang.Thread
implements java.lang.Runnable, java.util.Observer
  
```

this is the room server. It is responsible for serving the game out to the clients it runs on the room admin's computer

Constructor Summary

public	RoomServer (ObservablePreGameData dataP, ObservableTeamData teamsP, ObservableChat chatP, ObservableChatAction actP, RoomClass room, Move moveP, ObservableSettingsData settingsP) Creates and sets up the roomserver object
--------	--

Method Summary

boolean	checkgamestarted () Checks to see if gamestarted his occured yet
java.net.InetAddress	getIp () gets current non localhost ip address
PacketStore	getUp () getter for upstack
void	kill () This will destroy the server and disconnect everyone
void	run () Deprecated. <i>not really needed anymore after rewrite!</i>
void	setGameData (ObservableGameData gameData_Admin) setter for gamedata
void	setUp (PacketStore up) setter for up packetstack
void	StartGame (ObservablePreGameData p, ObservableGameData g) this is called when we move from the pregame phase to the ingame phase it will send pre and game data to all of the clients and stop the server from listening for new connections
void	update (java.util.Observable o, java.lang.Object arg) update is called when any of the observers change, observer is the packet factory that creates a packet and then sends it on the upbound stack

Constructors

RoomServer

```
public RoomServer(ObservablePreGameData dataP,  
                  ObservableTeamData teamsP,  
                  ObservableChat chatP,  
                  ObservableChatAction actP,  
                  RoomClass room,  
                  Move moveP,  
                  ObservableSettingsData settingsP)
```

Creates and sets up the roomserver object

Parameters:

dataP
teamsP
chatP
actP
room
moveP
settingsP

Methods

getIp

```
public java.net.InetAddress getIp()  
    throws java.net.SocketException,  
           java.net.UnknownHostException
```

gets current non localhost ip address

Returns:

InetAddress

Throws:

SocketException
UnknownHostException

StartGame

```
public void StartGame(ObservablePreGameData p,  
                      ObservableGameData g)
```

this is called when we move from the pregame phase to the ingame phase it will send pre and game data to all of the clients and stop the server from listening for new connections

Parameters:

p - observable pregame data
g - observable game data

checkgamestarted

```
public boolean checkgamestarted()
```

Checks to see if gamestarted his occurred yet

(continued from last page)

Returns:

update

```
public void update(java.util.Observable o,  
                  java.lang.Object arg)
```

update is called when any of the observers change, observer is the packet factory that creates a packet and then sends it on the upbound stack

kill

```
public void kill()
```

This will destroy the server and disconnect everyone

run

```
public void run()
```

Deprecated. *not really needed anymore after rewrite!*

setUp

```
public void setUp(PacketStore up)
```

setter for up packetstack

getUp

```
public PacketStore getUp()
```

getter for upstack

Returns:

setGameData

```
public void setGameData(ObservableGameData gameData_Admin)
```

setter for gamedata

Parameters:

gameData_Admin

network

Class ServerRunTest

java.lang.Object

└─network.ServerRunTest

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

test for Server

Constructor Summary

public	ServerRunTest()
--------	---------------------------------

Method Summary

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

Constructors

ServerRunTest

```
public ServerRunTest()
```

Methods

main

```
public static void main(java.lang.String[] args)
throws java.net.UnknownHostException,
       java.net.SocketException
```

Parameters:

args

Throws:

UnknownHostException

SocketException

network

Class SnetworkStack

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- network.SnetworkStack
  
```

```

public class SnetworkStack
extends java.util.Observable
  
```

This is the que for all inbound/outbound network actions

Constructor Summary

public	SnetworkStack()
--------	---------------------------------

Method Summary

void	hup() notify all that the connection is no longer live
boolean	isempty()
boolean	islive() returns true if connection is still alive
boolean	isPing() first packet is a ping packet
boolean	isStd() is a standard packet
Ppacket	peek()
Ppacket	pop() getsfirst packet in que
boolean	push(Ppacket P, boolean pop) adds new packet to the stack

Constructors

SnetworkStack

```

public SnetworkStack()
  
```

Methods

(continued from last page)

islive

```
public boolean islive()
```

returns true if connection is still alive

hup

```
public void hup()
```

notify all that the connection is no longer live

push

```
public boolean push(Ppacket P,  
                    boolean pop)
```

adds new packet to the stack

pop

```
public Ppacket pop()
```

getsfirst packet in que

peek

```
public Ppacket peek()
```

isPing

```
public boolean isPing()
```

first packet is a ping packet

isStd

```
public boolean isStd()
```

is a standard packet

isempty

```
public boolean isempty()
```


network

Class TnetworkMaint

```

java.lang.Object
  |
  +- java.lang.Thread
      |
      +- network.TnetworkMaint
  
```

All Implemented Interfaces:

java.util.Observer, java.lang.Runnable

```

public class TnetworkMaint
  extends java.lang.Thread
  implements java.lang.Runnable, java.util.Observer
  
```

Handles the sending and recieveing of all network keepalive packets

Constructor Summary

public	TnetworkMaint (SnetworkStack nsIn, SnetworkStack nsOut)
--------	--

Method Summary

void	run () spins off a thread that sends a ping packet every once and a while
void	update (java.util.Observable arg0, java.lang.Object arg1)

Constructors

TnetworkMaint

```

public TnetworkMaint(SnetworkStack nsIn,
                    SnetworkStack nsOut)
  
```

Methods

update

```

public void update(java.util.Observable arg0,
                  java.lang.Object arg1)
  
```

run

```

public void run()
  
```

spins off a thread that sends a ping packet every once and a while

network

Class XNetworkTest

```

java.lang.Object
  |-- junit.framework.Assert
      |-- junit.framework.TestCase
          |-- network.XNetworkTest
  
```

All Implemented Interfaces:
junit.framework.Test

```

public class XNetworkTest
extends junit.framework.TestCase
  
```

This tests the network package.

Constructor Summary

public	XNetworkTest()
--------	--------------------------------

Method Summary

void	testNetStack() tests Network packet stack
void	testPing() Tests the ping class to be sure all of the inputs are proper
void	testPong() Tests the ping class to be sure that reply packets are also proper
void	testStandard() Tests standard packets for proper responses

Constructors

XNetworkTest

```
public XNetworkTest()
```

Methods

testPing

```
public void testPing()
```

Tests the ping class to be sure all of the inputs are proper

testPong

```
public void testPong()
```

Tests the ping class to be sure that reply packets are also proper

testStandard

```
public void testStandard()
```

Tests standard packets for proper responses

testNetStack

```
public void testNetStack()
```

tests Network packet stack

Package
observables

observables

Class ObservableChat

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- observables.ObservableChat
  
```

All Implemented Interfaces:
 java.io.Serializable

```

public class ObservableChat
extends java.util.Observable
implements java.io.Serializable
  
```

Constructor Summary

public	ObservableChat (ObservableChatAction chatActP)
--------	---

Method Summary

void	addChatMessage (ChatMessage message, java.lang.String description)
java.util.Iterator	getMessages ()
java.util.Iterator	getMessagesForPlayer (Player player)
java.util.Iterator	getMessagesForPlayerWithIgnore (Player player)
java.util.Iterator	getMessagesWithIgnored ()

Constructors

ObservableChat

```
public ObservableChat(ObservableChatAction chatActP)
```

Methods

addChatMessage

```
public void addChatMessage(ChatMessage message,
    java.lang.String description)
```

getMessages

```
public java.util.Iterator getMessages()
```

getMessagesWithIgnored

```
public java.util.Iterator getMessagesWithIgnored()
```

getMessagesForPlayer

```
public java.util.Iterator getMessagesForPlayer(Player player)
```

getMessagesForPlayerWithIgnore

```
public java.util.Iterator getMessagesForPlayerWithIgnore(Player player)
```

observables

Class ObservableChatAction

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- observables.ObservableChatAction
  
```

All Implemented Interfaces:

```
java.io.Serializable
```

```

public class ObservableChatAction
  extends java.util.Observable
  implements java.io.Serializable
  
```

Constructor Summary

public	ObservableChatAction()
--------	--

Method Summary

void	addGlobalIgnoredPlayer (Player player)
void	addIgnoredPlayer (Player player)
java.util.LinkedList	getGlobabllyIgnoredPlayers ()
boolean	isNextMessageWhisper ()
boolean	isPlayerIgnored (Player player)
void	removeGlobalIgnoredPlayer (Player player)
void	removeIgnoredPlayer (Player player)
void	setGloballyIgnoredPlayers (java.util.LinkedList list)
void	setNextMessageWhisper (boolean state)

Constructors

ObservableChatAction

```
public ObservableChatAction()
```

Methods

addIgnoredPlayer

```
public void addIgnoredPlayer(Player player)
```

addGlobalIgnoredPlayer

```
public void addGlobalIgnoredPlayer(Player player)
```

removeIgnoredPlayer

```
public void removeIgnoredPlayer(Player player)
```

removeGlobalIgnoredPlayer

```
public void removeGlobalIgnoredPlayer(Player player)
```

isPlayerIgnored

```
public boolean isPlayerIgnored(Player player)
```

isNextMessageWhisper

```
public boolean isNextMessageWhisper()
```

setNextMessageWhisper

```
public void setNextMessageWhisper(boolean state)
```

getGloballyIgnoredPlayers

```
public java.util.LinkedList getGloballyIgnoredPlayers()
```

setGloballyIgnoredPlayers

```
public void setGloballyIgnoredPlayers(java.util.LinkedList list)
```


observables

Class ObservableGameData

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- observables.ObservableGameData
  
```

All Implemented Interfaces:

java.util.Observer, java.io.Serializable

```

public class ObservableGameData
extends java.util.Observable
implements java.io.Serializable, java.util.Observer
  
```

Observable for keeping track of the in game activities and winning situations, stores the information necessary for the players to properly maintain a game

Constructor Summary

public	ObservableGameData (RoomClass roomin, Player curPlayer, ObservableTeamData otidin, ObservableSettingsData settings, Move move) Default constructor
--------	--

Method Summary

Player	currentTurnPlayer ()
Player	findPlayer (int ID) Returns the player with the specified ID, null if not found
Team	findTeam (int ID) Returns the team with the specified Player ID, null if not found
Board	getBoard ()
GameCard[]	getBoardCards () Get an array of cards on the board
Player	getCurrentTurn ()
Player	getMe ()
Move	getMove ()
int	getSequenceSize ()
int	getSizeX () Get the size of the cube along the x-axis
int	getSizeY () Get the size of the cube along the y-axis

int	getSizeZ() Get the size of the cube along the z-axis
Team	getWinner() This function returns winning team else returns null
boolean	Isgameover() This function returns a boolean true if a winner is set and a false if game is on
void	setBoard(Board b)
void	setCurrentTurn(Player currentTurn)
void	setData(ObservableGameData obj)
void	setWinner(Team t) This function sets the winner
void	update (java.util.Observable arg0, java.lang.Object arg1)

Constructors

ObservableGameData

```
public ObservableGameData(RoomClass roomin,
    Player curPlayer,
    ObservableTeamData otin,
    ObservableSettingsData settings,
    Move move)
```

Default constructor

Methods

findPlayer

```
public Player findPlayer(int ID)
```

Returns the player with the specified ID, null if not found

Parameters:

ID

Returns:

findTeam

```
public Team findTeam(int ID)
```

Returns the team with the specified Player ID, null if not found

Parameters:

ID

(continued from last page)

Returns:

Isgameover

public boolean **Isgameover**()

This function returns a boolean true if a winner is set and a false if game is on

setWinner

public void **setWinner**([Team](#) t)

This function sets the winner

getSequenceSize

public int **getSequenceSize**()

currentTurnPlayer

public [Player](#) **currentTurnPlayer**()

getWinner

public [Team](#) **getWinner**()

This function returns winning team else returns null

getBoardCards

public [GameCard\[\]](#) **getBoardCards**()

Get an array of cards on the board

Returns:

getSizeX

public int **getSizeX**()

Get the size of the cube along the x-axis

Returns:

The size of the cube along the x-axis

getSizeY

public int **getSizeY**()

Get the size of the cube along the y-axis

Returns:

(continued from last page)

The size of the cube along the y-axis

getSizeZ

```
public int getSizeZ()
```

Get the size of the cube along the z-axis

Returns:The size of the cube along the z-axis

getMove

```
public Move getMove()
```

getBoard

```
public Board getBoard()
```

getMe

```
public Player getMe()
```

update

```
public void update(java.util.Observable arg0,  
                  java.lang.Object arg1)
```

setBoard

```
public void setBoard(Board b)
```

setData

```
public void setData(ObservableGameData obj)
```

getCurrentTurn

```
public Player getCurrentTurn()
```

setCurrentTurn

```
public void setCurrentTurn(Player currentTurn)
```

(continued from last page)

observables

Class ObservablePreGameData

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- observables.ObservablePreGameData
  
```

All Implemented Interfaces:

```

java.io.Serializable
  
```

```

public class ObservablePreGameData
extends java.util.Observable
implements java.io.Serializable
  
```

Constructor Summary

public	ObservablePreGameData()
public	ObservablePreGameData(int x, int y, int z, int gameTime, int roundTime, boolean talk, int numPlayers, int numComputers, boolean showHand, java.lang.String Password)

Method Summary

int	GetBoardSizeX()
int	GetBoardSizeY()
int	GetBoardSizeZ()
int	GetGameTimeLimit()
boolean	getIsAdmin()
int	GetMaxNumberOfPlayers()
int	GetNumberOfComputers()
java.lang.String	getPassword()
boolean	getReady()
int	GetRoundTimeLimit()
boolean	GetShowHand()

boolean	GetTableTalk()
void	SetBoardSizeX(int x)
void	SetBoardSizeY(int y)
void	SetBoardSizeZ(int z)
void	SetGameTimeLimit(int time)
void	SetIsAdmin(boolean admin)
void	SetMaxNumberOfComputers(int max)
void	SetMaxNumberOfPlayers(int max)
void	SetPassword(java.lang.String pass)
void	setReady(boolean state)
void	SetRoundTimeLimit(int time)
void	SetShowHand(boolean showHand)
void	SetTableTalk(boolean talk)
java.lang.String	toString()

Constructors

ObservablePreGameData

```
public ObservablePreGameData()
```

ObservablePreGameData

```
public ObservablePreGameData(int x,
                             int y,
                             int z,
                             int gameTime,
                             int roundTime,
                             boolean talk,
                             int numPlayers,
                             int numComputers,
                             boolean showHand,
                             java.lang.String Password)
```

(continued from last page)

Methods

SetBoardSizeX

```
public void SetBoardSizeX(int x)
```

SetBoardSizeY

```
public void SetBoardSizeY(int y)
```

SetBoardSizeZ

```
public void SetBoardSizeZ(int z)
```

SetGameTimeLimit

```
public void SetGameTimeLimit(int time)
```

SetRoundTimeLimit

```
public void SetRoundTimeLimit(int time)
```

SetTableTalk

```
public void SetTableTalk(boolean talk)
```

SetMaxNumberOfPlayers

```
public void SetMaxNumberOfPlayers(int max)
```

SetMaxNumberOfComputers

```
public void SetMaxNumberOfComputers(int max)
```

SetShowHand

```
public void SetShowHand(boolean showHand)
```

(continued from last page)

SetPassword

```
public void SetPassword(java.lang.String pass)
```

SetIsAdmin

```
public void SetIsAdmin(boolean admin)
```

GetBoardSizeX

```
public int GetBoardSizeX()
```

GetBoardSizeY

```
public int GetBoardSizeY()
```

GetBoardSizeZ

```
public int GetBoardSizeZ()
```

GetRoundTimeLimit

```
public int GetRoundTimeLimit()
```

GetGameTimeLimit

```
public int GetGameTimeLimit()
```

GetMaxNumberOfPlayers

```
public int GetMaxNumberOfPlayers()
```

GetNumberOfComputers

```
public int GetNumberOfComputers()
```

GetShowHand

```
public boolean GetShowHand()
```

(continued from last page)

getTableTalk

```
public boolean getTableTalk()
```

getPassword

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

getIsAdmin

```
public boolean getIsAdmin()
```

setReady

```
public void setReady(boolean state)
```

getReady

```
public boolean getReady()
```

toString

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

observables

Class ObservableRoomData

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- observables.ObservableRoomData
  
```

All Implemented Interfaces:

```
java.util.Observer
```

```

public class ObservableRoomData
extends java.util.Observable
implements java.util.Observer
  
```

Constructor Summary

public	ObservableRoomData()
--------	--------------------------------------

Method Summary

boolean	addRoom(RoomClass room)
boolean	clearRoom(RoomClass room)
void	enteringRoom(RoomClass r)
java.util.LinkedList	GetAllRooms()
java.util.LinkedList	getRooms()
void	setRooms (java.util.LinkedList list)
void	update (java.util.Observable arg0, java.lang.Object arg1)

Constructors

ObservableRoomData

```
public ObservableRoomData()
```

Methods

(continued from last page)

GetAllRooms

```
public java.util.LinkedList GetAllRooms()
```

addRoom

```
public boolean addRoom(RoomClass room)
```

clearRoom

```
public boolean clearRoom(RoomClass room)
```

getRooms

```
public java.util.LinkedList getRooms()
```

enteringRoom

```
public void enteringRoom(RoomClass r)
```

update

```
public void update(java.util.Observable arg0,  
                  java.lang.Object arg1)
```

setRooms

```
public void setRooms(java.util.LinkedList list)
```

observables

Class ObservableSettingsData

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- observables.ObservableSettingsData
  
```

All Implemented Interfaces:

```
java.io.Serializable
```

public class **ObservableSettingsData**

extends `java.util.Observable`

implements `java.io.Serializable`

Constructor Summary

public	ObservableSettingsData()
public	ObservableSettingsData (<code>javax.swing.ImageIcon</code> ava, <code>java.lang.String</code> board, <code>java.lang.String</code> nick)

Method Summary

<code>java.lang.String</code>	getAvatar_path()
<code>javax.swing.ImageIcon</code>	getAvatar()
<code>java.lang.String</code>	getBoardType()
<code>java.net.InetAddress</code>	getGameServer()
<code>java.net.InetAddress</code>	getLobbyServerAddress()
<code>java.lang.String</code>	getNickname()
<code>java.lang.String</code>	getTheme_folder()
void	setAvatar_path (<code>java.lang.String</code> avatar_path)
void	SetAvatar (<code>javax.swing.ImageIcon</code> ava)
void	SetBoardType (<code>java.lang.String</code> board)
void	SetNickname (<code>java.lang.String</code> nick)
void	setTheme_folder (<code>java.lang.String</code> theme_folder)

Constructors

ObservableSettingsData

```
public ObservableSettingsData()
```

ObservableSettingsData

```
public ObservableSettingsData(javax.swing.ImageIcon ava,  
                               java.lang.String board,  
                               java.lang.String nick)
```

Methods

getGameServer

```
public java.net.InetAddress getGameServer()
```

SetAvatar

```
public void SetAvatar(javax.swing.ImageIcon ava)
```

SetBoardType

```
public void SetBoardType(java.lang.String board)
```

SetNickname

```
public void SetNickname(java.lang.String nick)
```

getAvatar

```
public javax.swing.ImageIcon getAvatar()
```

getBoardType

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

(continued from last page)

getNickname

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

getAvatar_path

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

setAvatar_path

```
public void setAvatar_path(java.lang.String avatar_path)
```

getTheme_folder

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

setTheme_folder

```
public void setTheme_folder(java.lang.String theme_folder)
```

getLobbyServerAddress

```
public java.net.InetAddress getLobbyServerAddress()
```

observables

Class ObservableTeamData

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- observables.ObservableTeamData
  
```

All Implemented Interfaces:

java.io.Serializable, java.util.Observer

```

public class ObservableTeamData
  extends java.util.Observable
  implements java.util.Observer, java.io.Serializable
  
```

Constructor Summary

public	ObservableTeamData()
--------	--------------------------------------

Method Summary

void	addTeam()
void	addTeam(int teamIDP)
void	addTeam(int teamIDP, java.awt.Color teamColorP)
Player	findPlayer(Player other)
Team	getTeam(int id)
java.util.LinkedList	getTeams()
void	removePlayer(Player p)
void	removeTeam(Team t)
void	setTeams(ObservableTeamData t)
int	totalPlayers()
void	update(java.util.Observable arg0, java.lang.Object arg1)

Constructors

(continued from last page)

ObservableTeamData

```
public ObservableTeamData()
```

Methods

getTeam

```
public Team getTeam(int id)
```

removePlayer

```
public void removePlayer(Player p)
```

getTeams

```
public java.util.LinkedList getTeams()
```

addTeam

```
public void addTeam()
```

addTeam

```
public void addTeam(int teamIDP)
```

addTeam

```
public void addTeam(int teamIDP,  
                    java.awt.Color teamColorP)
```

totalPlayers

```
public int totalPlayers()
```

update

```
public void update(java.util.Observable arg0,  
                  java.lang.Object arg1)
```

removeTeam

```
public void removeTeam(Team t)
```

setTeams

```
public void setTeams(ObservableTeamData t)
```

findPlayer

```
public Player findPlayer(Player other)
```

Package sequence

sequence

Class CardSelector

java.lang.Object

└--sequence.CardSelector

All Implemented Interfaces:

java.util.Observer

```
public class CardSelector
  extends java.lang.Object
  implements java.util.Observer
```

Constructor Summary

public	CardSelector (ObservableGameData gameInfo, GameCard[] hand)
--------	--

Method Summary

void	update (java.util.Observable arg0, java.lang.Object arg1)
------	---

Constructors

CardSelector

```
public CardSelector(ObservableGameData gameInfo,
                    GameCard\[\] hand)
```

Methods

update

```
public void update(java.util.Observable arg0,
                  java.lang.Object arg1)
```

sequence

Class ChatMessage

java.lang.Object

└─sequence.ChatMessage

All Implemented Interfaces:

java.io.Serializable

```
public class ChatMessage
extends java.lang.Object
implements java.io.Serializable
```

Constructor Summary

public	ChatMessage (java.lang.String message)
public	ChatMessage (Player fromPlayer, Player toPlayer, java.lang.String message)

Method Summary

Player	getFromPlayer ()
java.lang.String	getMessage ()
Player	getToPlayer ()

Constructors

ChatMessage

```
public ChatMessage(java.lang.String message)
```

ChatMessage

```
public ChatMessage(Player fromPlayer,
                   Player toPlayer,
                   java.lang.String message)
```

Methods

(continued from last page)

getFromPlayer

```
public Player getFromPlayer()
```

getToPlayer

```
public Player getToPlayer()
```

getMessage

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

sequence

Class CustomTextArea

```

java.lang.Object
  |-- java.awt.Component
    |-- java.awt.Container
      |-- javax.swing.JComponent
        |-- javax.swing.text.JTextComponent
          |-- javax.swing.JTextArea
            |-- sequence.CustomTextArea

```

All Implemented Interfaces:

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

public class CustomTextArea

extends javax.swing.JTextArea

implements javax.swing.Scrollable, javax.accessibility.Accessible, java.io.Serializable, javax.swing.TransferHandler.HasGetTransferHandler, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.io.Serializable

Constructor Summary

public	CustomTextArea()
public	CustomTextArea(int lines)
public	CustomTextArea(int lines, int columns)

Method Summary

void	addCMessageCheckForIgnored(ObservableChat chat, Player thisPlayer)
void	addMessage(java.lang.String s)

Constructors

CustomTextArea

```
public CustomTextArea()
```

(continued from last page)

CustomTextArea

```
public CustomTextArea(int lines)
```

CustomTextArea

```
public CustomTextArea(int lines,  
                      int columns)
```

Methods

addMessage

```
public void addMessage(java.lang.String s)
```

addCMessageCheckForIgnored

```
public void addCMessageCheckForIgnored(ObservableChat chat,  
    Player thisPlayer)
```


sequence

Class Deck

java.lang.Object
 ↳ **sequence.Deck**

All Implemented Interfaces:
 java.io.Serializable

public class **Deck**
 extends java.lang.Object
 implements java.io.Serializable

Constructor Summary

public	Deck()
public	Deck(GameCard[] arr) Creates deck
public	Deck(int size)

Method Summary

boolean	acceptCard (java.lang.Integer in)
java.lang.Integer	DealRandom () Deals one card from the deck
boolean	findCard (java.lang.Integer look)
java.util.LinkedList	getCards ()
boolean	isEmpty ()
java.lang.Integer	LastAdded () The last card added to the deck
boolean	UseCard (java.lang.Integer use)

Constructors

Deck

public **Deck**()

Deck

```
public Deck(GameCard\[\] arr)
```

Creates deck

Deck

```
public Deck(int size)
```

Methods

UseCard

```
public boolean UseCard(java.lang.Integer use)
```

acceptCard

```
public boolean acceptCard(java.lang.Integer in)
```

DealRandom

```
public java.lang.Integer DealRandom()
```

Deals one card from the deck

Returns:

GameCard from the deck

LastAdded

```
public java.lang.Integer LastAdded()
```

The last card added to the deck

Returns:

GameCard from the deck

findCard

```
public boolean findCard(java.lang.Integer look)
```

isEmpty

```
public boolean isEmpty()
```

(continued from last page)

getCards

```
public java.util.LinkedList getCards()
```

sequence

Class GameCard

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- sequence.GameCard
  
```

All Implemented Interfaces:
java.io.Serializable

```

public class GameCard
  extends java.util.Observable
  implements java.io.Serializable
  
```

Nested Class Summary

class	GameCard.Faces GameCard.Faces
class	GameCard.NotifyEvents GameCard.NotifyEvents

Constructor Summary

public	GameCard (int cardID, GameCard.Faces face, int x, int y, Player owner) Creates the card given the values
public	GameCard ()

Method Summary

boolean	animate (float timeDiff)
java.awt.Image	get2DTexture ()
javax.media.j3d.Texture	get3DTexture ()
float	getAnimationTransitionTime ()
int	getCardID ()
int	getCardTypeID () Gets the Card Type
java.awt.Color	getColor2D ()
javax.vecmath.Color3f	getColor3D ()

GameCard.Faces	getFace() The face the card is on
java.awt.Color	getFromColor()
com.sun.j3d.utils.geometry.Box	getGeometry()
boolean	getMouseHovering() Is the mouse hovering over the card?
Player	getOwner() Get the owner of this card
int	getPosX()
int	getPosY()
boolean	getSelected()
Pair	getTexture()
java.awt.Color	getToColor()
javax.media.j3d.BranchGroup	getToken()
int	getX() Get the x-coordinate of the card (Distance from the top edge of the face)
int	getY() Get the y-coordinate of the card (Distance from the left edge of the face)
void	selectCard()
void	selectCard(boolean isSelected)
void	setCardID(int cardID)
void	setCardTypeID(int set) Sets the CardType ID
void	setFace(GameCard.Faces setF) Sets the face of the Card
void	setGeometry(com.sun.j3d.utils.geometry.Box newGeometry)
void	setMouseHovering(boolean isHovering) Set if the mouse is hovering over the card
void	setOwner(Player newOwner) Sets the owner of this card
void	setPosX(int newX)

void	setPosY (int newY)
void	setTexture (java.awt.Image tex2D)
void	setTexture (Pair texture)
void	setTexture (javax.media.j3d.Texture tex3D)
void	setToken (javax.media.j3d.BranchGroup newToken)
void	setX (int x)
void	setY (int y)
java.lang.String	toString () Just create a console-friendlier toString method

Constructors

GameCard

```
public GameCard(int cardID,
                 GameCard.Faces face,
                 int x,
                 int y,
                 Player owner)
```

Creates the card given the values

Parameters:

cardID -- The ID of the card
suite -- The suite of the card
rank -- position_x = newX; The rank of the card
face -- The face the card should be drawn on
x -- The x-coordinate of the card from the left edge of the face
y -- The y-coordinate of the card from the top edge of the face
owner -- The owner of the card (null if no owner exists)
texture -- The filename to be used as the texture of this card

GameCard

```
public GameCard()
```

Methods

setFace

```
public void setFace(GameCard.Faces setF)
```

Sets the face of the Card

Parameters:

(continued from last page)

setF

getCardID

```
public int getCardID()
```

getFace

```
public GameCard.Faces getFace()
```

The face the card is on

Returns:

The face the card is on

getX

```
public int getX()
```

Get the x-coordinate of the card (Distance from the top edge of the face)

Returns:

The x-coordinate of the card (Distance from the top edge of the face)

setCardTypeID

```
public void setCardTypeID(int set)
```

Sets the CardType ID

Parameters:

set

getCardTypeID

```
public int getCardTypeID()
```

Gets the Card Type

getY

```
public int getY()
```

Get the y-coordinate of the card (Distance from the left edge of the face)

Returns:

The y-coordinate of the card (Distance from the left edge of the face)

getPosX

```
public int getPosX()
```

(continued from last page)

getPosY

```
public int getPosY()
```

setPosX

```
public void setPosX(int newX)
```

setPosY

```
public void setPosY(int newY)
```

get3DTexture

```
public javax.media.j3d.Texture get3DTexture()
```

get2DTexture

```
public java.awt.Image get2DTexture()
```

getTexture

```
public Pair getTexture()
```

setTexture

```
public void setTexture(javax.media.j3d.Texture tex3D)
```

setTexture

```
public void setTexture(java.awt.Image tex2D)
```

setTexture

```
public void setTexture(Pair texture)
```

getOwner

```
public Player getOwner()
```

(continued from last page)

Get the owner of this card

Returns:

The owner of the card, or null if none exists

setOwner

```
public void setOwner(Player newOwner)
```

Sets the owner of this card

Parameters:

`newOwner` - The new owner of the card, or null to set no owner

getToken

```
public javax.media.j3d.BranchGroup getToken()
```

setToken

```
public void setToken(javax.media.j3d.BranchGroup newToken)
```

getGeometry

```
public com.sun.j3d.utils.geometry.Box getGeometry()
```

setGeometry

```
public void setGeometry(com.sun.j3d.utils.geometry.Box newGeometry)
```

selectCard

```
public void selectCard()
```

selectCard

```
public void selectCard(boolean isSelected)
```

getSelected

```
public boolean getSelected()
```

(continued from last page)

setMouseHovering

```
public void setMouseHovering(boolean isHovering)
```

Set if the mouse is hovering over the card

Parameters:

isHovering - Is the mouse hovering over the card?

getMouseHovering

```
public boolean getMouseHovering()
```

Is the mouse hovering over the card?

Returns:

If the mouse is hovering over the card

toString

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

Just create a console-friendlier toString method

getColor2D

```
public java.awt.Color getColor2D()
```

getColor3D

```
public javax.vecmath.Color3f getColor3D()
```

animate

```
public boolean animate(float timeDiff)
```

getToColor

```
public java.awt.Color getToColor()
```

getFromColor

```
public java.awt.Color getFromColor()
```

getAnimationTransitionTime

```
public float getAnimationTransitionTime()
```

(continued from last page)

setX

```
public void setX(int x)
```

setY

```
public void setY(int y)
```

setCardID

```
public void setCardID(int cardID)
```

sequence

Class GameCard.Faces

java.lang.Object

└- java.lang.Enum

└- **sequence.GameCard.Faces**

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

public static final class **GameCard.Faces**
 extends java.lang.Enum

All possible faces that a card can exist on

Field Summary

public static final	BACK
public static final	BOTTOM
public static final	FRONT
public static final	LEFT
public static final	RIGHT
public static final	TOP

Method Summary

static GameCard.Faces	valueOf (java.lang.String name)
static GameCard.Faces[]	values ()

Fields

TOP

public static final sequence.GameCard.Faces **TOP**

BOTTOM

public static final sequence.GameCard.Faces **BOTTOM**

(continued from last page)

LEFT

```
public static final sequence.GameCard.Faces LEFT
```

RIGHT

```
public static final sequence.GameCard.Faces RIGHT
```

FRONT

```
public static final sequence.GameCard.Faces FRONT
```

BACK

```
public static final sequence.GameCard.Faces BACK
```

Methods

values

```
public static GameCard.Faces\[\] values()
```

valueOf

```
public static GameCard.Faces valueOf(java.lang.String name)
```

sequence

Class GameCard.NotifyEvents

```

java.lang.Object
  |
  +- java.lang.Enum
      |
      +- sequence.GameCard.NotifyEvents
  
```

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

```

public static final class GameCard.NotifyEvents
extends java.lang.Enum
  
```

The types of notify events this class throws

Field Summary	
public static final	CARD_CHANGETYPE
public static final	CARD_SELECT
public static final	OWNER_PLACE

Method Summary	
static GameCard.NotifyEvents	valueOf (java.lang.String name)
static GameCard.NotifyEvents []	values ()

Fields

CARD_SELECT

```
public static final sequence.GameCard.NotifyEvents CARD_SELECT
```

OWNER_PLACE

```
public static final sequence.GameCard.NotifyEvents OWNER_PLACE
```

CARD_CHANGETYPE

```
public static final sequence.GameCard.NotifyEvents CARD_CHANGETYPE
```

(continued from last page)

Methods

values

```
public static GameCard.NotifyEvents\[\] values()
```

valueOf

```
public static GameCard.NotifyEvents valueOf(java.lang.String name)
```

sequence

Class Move

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- sequence.Move

```

All Implemented Interfaces:
java.io.Serializable

```

public class Move
  extends java.util.Observable
  implements java.io.Serializable

```

Constructor Summary

public	Move()
public	Move(boolean approved)

Method Summary

int	getDealtCard()
java.lang.Integer	getFrom()
int	getPlayer()
int	getTo()
boolean	isApproved()
void	setApproved(Player curPlayer, boolean val)
void	setCardFrom(GameCard from)
void	setCardFrom(java.lang.Integer from)
void	setDealtCard(GameCard in)
void	setDealtCard(int in)
void	setFields(Move reference)
void	setPlayer(int play)

void	setPlayer (Player play)
void	setTo (GameCard to)
void	setTo (int to)

Constructors

Move

```
public Move()
```

Move

```
public Move(boolean approved)
```

Methods

setDealtCard

```
public void setDealtCard(GameCard in)
```

setDealtCard

```
public void setDealtCard(int in)
```

setApproved

```
public void setApproved(Player curPlayer,  
                        boolean val)
```

setPlayer

```
public void setPlayer(Player play)
```

setPlayer

```
public void setPlayer(int play)
```

(continued from last page)

setTo

```
public void setTo(GameCard to)
```

setTo

```
public void setTo(int to)
```

setCardFrom

```
public void setCardFrom(java.lang.Integer from)
```

setCardFrom

```
public void setCardFrom(GameCard from)
```

isApproved

```
public boolean isApproved()
```

getPlayer

```
public int getPlayer()
```

getTo

```
public int getTo()
```

getFrom

```
public java.lang.Integer getFrom()
```

getDealtCard

```
public int getDealtCard()
```

setFields

```
public void setFields(Move reference)
```

(continued from last page)

sequence

Class Nonsequential

java.lang.Object

└-sequence.Nonsequential

All Implemented Interfaces:

java.util.Observer, java.lang.Runnable

```
public class Nonsequential
extends java.lang.Object
implements java.lang.Runnable, java.util.Observer
```

Constructor Summary

public	Nonsequential()
--------	---------------------------------

Method Summary

void	addCloseListener (javax.swing.JFrame screen)
void	killGUIs ()
void	killGUIs (javax.swing.JFrame ignore)
static void	main (java.lang.String[] args)
void	run ()
void	update (java.util.Observable arg0, java.lang.Object arg1)

Constructors

Nonsequential

```
public Nonsequential()
```

Methods

run

```
public void run()
```

addCloseListener

```
public void addCloseListener(javax.swing.JFrame screen)
```

killGUIs

```
public void killGUIs(javax.swing.JFrame ignore)
```

killGUIs

```
public void killGUIs()
```

update

```
public void update(java.util.Observable arg0,  
                  java.lang.Object arg1)
```

main

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

Parameters:

args

sequence

Class Pair

```
java.lang.Object
  |
  +-sequence.Pair
```

All Implemented Interfaces:
java.io.Serializable

```
public class Pair
  extends java.lang.Object
  implements java.io.Serializable
```

Class that holds a pair of objects

Parameters:

T - The first value in the pair, v - The second value in the pair

Constructor Summary

public	Pair (java.lang.Object t, java.lang.Object v)
--------	---

Method Summary

boolean	equals (java.lang.Object arg0)
java.lang.Object	getVal1 ()
java.lang.Object	getVal2 ()
void	setVal1 (java.lang.Object val1)
void	setVal2 (java.lang.Object val2)
java.lang.String	toString ()

Constructors

Pair

```
public Pair(java.lang.Object t,
            java.lang.Object v)
```

Methods

(continued from last page)

getVal1

```
public java.lang.Object getVal1()
```

setVal1

```
public void setVal1(java.lang.Object val1)
```

getVal2

```
public java.lang.Object getVal2()
```

setVal2

```
public void setVal2(java.lang.Object val2)
```

equals

```
public boolean equals(java.lang.Object arg0)
```

toString

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

sequence

Class Player

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- sequence.Player
  
```

All Implemented Interfaces:

java.util.Observer, java.io.Serializable

```

public class Player
  extends java.util.Observable
  implements java.io.Serializable, java.util.Observer
  
```

Constructor Summary

public	Player()
public	Player(ObservableSettingsData dataP)

Method Summary

boolean	addCard (java.lang.Integer input)
boolean	equals (java.lang.Object o)
javax.swing.ImageIcon	getAvatar ()
java.util.LinkedList	getHand ()
java.lang.String	getNickname ()
int	getPlayerID ()
Team	getTeam ()
boolean	hasCardType (java.lang.Integer look)
void	initializeHand (int size)
boolean	isAdmin ()
boolean	isComputer ()
void	kicked ()

void	removeCard (java.lang.Integer rem)
void	sendAvatar ()
void	sendOverNetwork ()
void	setAvatar (javax.swing.ImageIcon imageIcon)
void	setIsAdmin (boolean state)
void	setIsComputer (boolean state)
void	setKicked (boolean state)
void	setNickname (java.lang.String nickname)
void	setPlayerID (int playerID)
void	setSettings (ObservableSettingsData data)
void	setTeam (Team team)
java.lang.String	toString ()
void	update (java.util.Observable arg0, java.lang.Object arg1)
boolean	useCardType (java.lang.Integer needed) "Uses" a card from the players hand

Constructors

Player

```
public Player()
```

Player

```
public Player(ObservableSettingsData dataP)
```

Methods

useCardType

```
public boolean useCardType(java.lang.Integer needed)
```

"Uses" a card from the players hand

(continued from last page)

Parameters:

needed - the card to be used

Returns:

true if the card was found and removed from the list

setSettings

```
public void setSettings(ObservableSettingsData data)
```

hasCardType

```
public boolean hasCardType(java.lang.Integer look)
```

initializeHand

```
public void initializeHand(int size)
```

addCard

```
public boolean addCard(java.lang.Integer input)
```

removeCard

```
public void removeCard(java.lang.Integer rem)
```

getHand

```
public java.util.LinkedList getHand()
```

getPlayerID

```
public int getPlayerID()
```

setPlayerID

```
public void setPlayerID(int playerID)
```

getTeam

```
public Team getTeam()
```

(continued from last page)

setTeam

```
public void setTeam(Team team)
```

getNickname

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

setNickname

```
public void setNickname(java.lang.String nickname)
```

setIsAdmin

```
public void setIsAdmin(boolean state)
```

isAdmin

```
public boolean isAdmin()
```

toString

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

getAvatar

```
public javax.swing.ImageIcon getAvatar()
```

setAvatar

```
public void setAvatar(javax.swing.ImageIcon imageIcon)
```

kicked

```
public void kicked()
```

(continued from last page)

setKicked

```
public void setKicked(boolean state)
```

setIsComputer

```
public void setIsComputer(boolean state)
```

isComputer

```
public boolean isComputer()
```

update

```
public void update(java.util.Observable arg0,  
                  java.lang.Object arg1)
```

equals

```
public boolean equals(java.lang.Object o)
```

sendOverNetwork

```
public void sendOverNetwork()
```

sendAvatar

```
public void sendAvatar()
```

sequence

Class RoomClass

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- sequence.RoomClass

```

All Implemented Interfaces:

java.util.Observer, java.io.Serializable

```

public class RoomClass
extends java.util.Observable
implements java.io.Serializable, java.util.Observer

```

This is the data that is kept for each of the rooms. It includes the ip address for the room server, room name, admin's data, and pregame data

Constructor Summary

public	RoomClass (java.net.InetAddress IP, java.lang.String RoomNAME, Player admin, ObservablePreGameData pregame, java.util.LinkedList player)
public	RoomClass ()

Method Summary

void	addPlayer (Player p)
void	ClearPlayers ()
boolean	equals (java.lang.Object other)
Player	findPlayer (Player search)
Player	GetAdminPlayer ()
java.lang.String	getDetails ()
java.net.InetAddress	GetIp ()
java.util.LinkedList	GetPlayers ()
ObservablePreGameData	getPregameData ()
java.lang.String	GetRoomName ()
void	removePlayer (Player p)

void	SetAdminPlayer (Player p)
void	SetIp (java.net.InetAddress i)
void	setPregameData (ObservablePreGameData p)
void	SetRoomName (java.lang.String r)
java.lang.String	toString ()
void	update (java.util.Observable o, java.lang.Object arg)

Constructors

RoomClass

```
public RoomClass(java.net.InetAddress IP,
                 java.lang.String RoomNAME,
                 Player admin,
                 ObservablePreGameData pregame,
                 java.util.LinkedList player)
```

RoomClass

```
public RoomClass()
```

Methods

SetIp

```
public void SetIp(java.net.InetAddress i)
```

GetIp

```
public java.net.InetAddress GetIp()
    throws java.net.UnknownHostException
```

SetRoomName

```
public void SetRoomName(java.lang.String r)
```

(continued from last page)

GetRoomName

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

SetAdminPlayer

```
public void SetAdminPlayer(Player p)
```

GetAdminPlayer

```
public Player GetAdminPlayer()
```

setPregameData

```
public void setPregameData(ObservablePreGameData p)
```

getPregameData

```
public ObservablePreGameData getPregameData()
```

addPlayer

```
public void addPlayer(Player p)
```

removePlayer

```
public void removePlayer(Player p)
```

GetPlayers

```
public java.util.LinkedList GetPlayers()
```

findPlayer

```
public Player findPlayer(Player search)
```

ClearPlayers

```
public void ClearPlayers()
```

(continued from last page)

toString

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

update

```
public void update(java.util.Observable o,  
                  java.lang.Object arg)
```

getDetails

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

equals

```
public boolean equals(java.lang.Object other)
```

sequence

Class Team

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- sequence.Team

```

All Implemented Interfaces:

java.io.Serializable, java.util.Observer

```

public class Team
  extends java.util.Observable
  implements java.util.Observer, java.io.Serializable

```

Constructor Summary

public	Team (int teamIDP, java.awt.Color teamColorP)
--------	---

Method Summary

void	addPlayer (Player p)
boolean	equals (java.lang.Object o)
int	getNumOfSequences ()
java.util.Iterator	getPlayers ()
java.awt.Color	getTeamColor ()
int	getTeamID ()
int	playersOnTeam ()
void	removePlayer (Player p)
void	setNumOfSequences (int numOfSequences)
void	setTeamColor (java.awt.Color c)
java.lang.String	toString ()
void	update (java.util.Observable o, java.lang.Object arg)

(continued from last page)

Constructors

Team

```
public Team(int teamIDP,  
            java.awt.Color teamColorP)
```

Methods

addPlayer

```
public void addPlayer(Player p)
```

getPlayers

```
public java.util.Iterator getPlayers()
```

removePlayer

```
public void removePlayer(Player p)
```

playersOnTeam

```
public int playersOnTeam()
```

getTeamColor

```
public java.awt.Color getTeamColor()
```

update

```
public void update(java.util.Observable o,  
                  java.lang.Object arg)
```

getTeamID

```
public int getTeamID()
```

toString

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

(continued from last page)

setTeamColor

```
public void setTeamColor(java.awt.Color c)
```

getNumOfSequences

```
public int getNumOfSequences()
```

setNumOfSequences

```
public void setNumOfSequences(int numOfSequences)
```

equals

```
public boolean equals(java.lang.Object o)
```
