

Pier436 Common Infrastructure Team Design Documentation v 1.0

Appendix to Architecture Design v 1.0

Package pier436.application

Package Contents

- **Interfaces**
 - **Engine**
 - **EngineMessageListener** Supports 3.3.2, 3.3.3 This specified message listener will update game engine state when it receives a message from other users.
- **Classes**
 - **MapController**
 - **PierEngine**
 - **PierEngine.ClientConnectionListener**
 - **PierEngine.GameLoop** Loop for main thread
 - **PierEngine.SystemMessageListener**
 - **PierGraphicalController**

Interface Engine

Declaration

```
public interface Engine
```

All known subinterfaces

PierEngine

All classes known to implement interface

PierEngine

Field summary

- **CLIENT_FAST_PORT**
- **CLIENT_RELIABLE_PORT**
- **SERVER_FAST_PORT**
- **SERVER_RELIABLE_PORT**
- **WAIT_PORT**

Method summary

- **add(Entity)** Add the entity to the Engine's registry.
- **add(Entity, boolean)** Add the entity to the registry and share it across the network.
- **addCamera(Camera)** addCamera void

- **addEngineMessageListener(EngineMessageListener)** Adds an EngineMessageListener.
- **connect(String)** Connect to the specified host.
- **connect(String, int)**
- **disconnect()** Disconnects either client or server if they are connected or hosting.
- **host()** Begin hosting your session across the network.
- **initResources()** Initialize resources, including loading images that will be used and any entities that will exist at the beginning of the engine's life.
- **remove(Entity)** Removes the specified entity from the world.
- **removeEngineMessageListener(EngineMessageListener)** Removes the specified EngineMessageListener
- **render()** Routine that should call the render commands on all created viewports.
- **send(SystemMessage)** Broadcasts system message across network.
- **share(Entity)** Share the given Entity across the network.
- **start()** Start the engine.
- **stop()** Stop/Pause the engine.
- **update(int)** Update all entities managed by the engine.

Fields

- int **SERVER_RELIABLE_PORT**
- int **SERVER_FAST_PORT**
- int **CLIENT_RELIABLE_PORT**
- int **CLIENT_FAST_PORT**
- int **WAIT_PORT**

Methods

- **add**
void **add(Entity entity)**
 - **Description** Add the entity to the Engine's registry. It will be updated in the game loop now.
 - **Parameters**
 - entity -- - Entity to be added.
- **add**
void **add(Entity entity, boolean share)**
 - **Description** Add the entity to the registry and share it across the network. This is equivalent to:
add(entity);
share(entity);
 - **Parameters**
 - entity -- - entity to be shared.
 - share -- - whether or not to share the entity.
- **addCamera**
void **addCamera(Camera cam)**
 - **Description** addCamera void
 - **Parameters**
 - cam --
- **addEngineMessageListener**
void **addEngineMessageListener(EngineMessageListener ml)**
 - **Description** Adds an EngineMessageListener.
 - **Parameters**
 - ml --
- **connect**
void **connect(java.lang.String address)**
 - **Description** Connect to the specified host. All Engine data will be lost in this operation.
 - **Parameters**
 - address -- - Address of the hosting computer.
- **connect**

```

void connect( java.lang.String address, int port )
● disconnect
void disconnect( )
    ○ Description Disconnects either client or server if they are connected or hosting.
● host
void host( )
    ○ Description Begin hosting your session across the network.
● initResources
void initResources( )
    ○ Description Initialize resources, including loading images that will be used and any entities that will exist at the beginning of the engine's life.
● remove
void remove( Entity entity )
    ○ Description Removes the specified entity from the world.
    ○ Parameters
        ■ entity -- - entity to be removed.
● removeEngineMessageListener
void removeEngineMessageListener( EngineMessageListener ml )
    ○ Description Removes the specified EngineMessageListener
    ○ Parameters
        ■ ml --
● render
void render( )
    ○ Description Routine that should call the render commands on all created viewports.
● send
void send( SystemMessage msg )
    ○ Description Broadcasts system message across network.
    ○ Parameters
        ■ msg --
● share
void share( Entity entity )
    ○ Description Share the given Entity across the network.
    ○ Parameters
        ■ entity -- - Entity to be shared.
● start
void start( )
    ○ Description Start the engine.
● stop
void stop( )
    ○ Description Stop/Pause the engine. To restart, use start().
● update
void update( int timeElapsed )
    ○ Description Update all entities managed by the engine.
    ○ Parameters
        ■ timeElapsed -- - amount of time since last update.

```

Interface EngineMessageListener

Supports 3.3.2, 3.3.3 This specified message listener will update game engine state when it receives a message from other users.

Declaration

```
public interface EngineMessageListener
```

Method summary

- `messageReceived(EngineMessage)`

Methods

- `messageReceived`
`void messageReceived(EngineMessage msg)`

Class MapController

Declaration

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

Constructor summary

- `MapController()`

Method summary

- `Load(String)` Will support 2.2.3, 2.2.3.1

Constructors

- `MapController`
`public MapController()`

Methods

- `Load`
`public void Load(java.lang.String filepathname)`
 - `Description` Will support 2.2.3, 2.2.3.1
 - `Parameters`
 - `filepathname` -- complete path to config file.

Class PierEngine

Declaration

```
public class PierEngine
extends java.lang.Object
implements Engine
```

Constructor summary

- **PierEngine()**

Method summary

- **add(Entity)** Add an entity to the manager.
- **add(Entity, boolean)** Add a shared entity to the manager TODO:Delete, shared entities should implement SharedEntity interface.
- **addCamera(Camera)** Add a camera for viewing.
- **addEngineMessageListener(EngineMessageListener)**
- **connect(String)** Connect to the reliable port
- **connect(String, int)** Connect JGN to specified port.
- **disconnect()**
- **getNextAvailableClientSocket()** Get a Socket for the client component.
- **host()**
- **initResources()**
- **remove(Entity)** Remove an entity from the manager.
- **removeEngineMessageListener(EngineMessageListener)**
- **render()**
- **send(SystemMessage)**
- **share(Entity)** TODO:delete, entities that are shared should just implement shared interface?
- **start()** Start game loop thread.
- **stop()** Pause game loop thread
- **update(int)**

Constructors

- **PierEngine**

```
public PierEngine( )
```

Methods

- **add**

```
public void add( Entity entity )
```

 - **Description** Add an entity to the manager.
 - **Parameters**
 - entity --
- **add**

```
public void add( Entity entity, boolean share )
```

 - **Description** Add a shared entity to the manager TODO:Delete, shared entities should implement SharedEntity interface.
- **addCamera**

```
public void addCamera( Camera cam )
```

 - **Description** Add a camera for viewing.
 - **Parameters**
 - cam --
- **addEngineMessageListener**

```
void addEngineMessageListener( EngineMessageListener ml )
```

 - **Description copied from EngineAdds** an EngineMessageListener.
 - **Parameters**
 - ml --
- **connect**

```
public void connect( java.lang.String address )
```

 - **Description** Connect to the reliable port
 - **Parameters**
 - address --
- **connect**

```

public void connect( java.lang.String address, int port )
    ◦ Description Connect JGN to specified port.
● disconnect
void disconnect( )
    ◦ Description copied from EngineDisconnects either client or server if they are connected or
    hosting.
● getNextAvailableClientSocket
public static JGNClient getNextAvailableClientSocket( )
    ◦ Description Get a Socket for the client component.
    ◦ Returns --
● host
void host( )
    ◦ Description copied from EngineBegin hosting your session across the network.
● initResources
void initResources( )
    ◦ Description copied from EngineInitialize resources, including loading images that will be
    used and any entities that will exist at the beginning of the engine's life.
● remove
public void remove( Entity entity )
    ◦ Description Remove an entity from the manager.
    ◦ Parameters
        ■ entity --
● removeEngineMessageListener
void removeEngineMessageListener( EngineMessageListener ml )
    ◦ Description copied from EngineRemoves the specified EngineMessageListener
    ◦ Parameters
        ■ ml --
● render
void render( )
    ◦ Description copied from EngineRoutine that should call the render commands on all created
    viewports.
● send
void send( SystemMessage msg )
    ◦ Description copied from EngineBroadcasts system message across network.
    ◦ Parameters
        ■ msg --
● share
public void share( Entity entity )
    ◦ Description TODO:delete, entities that are shared should just implement shared interface?
● start
public void start( )
    ◦ Description Start game loop thread.
● stop
public void stop( )
    ◦ Description Pause game loop thread
● update
void update( int timeElapsed )
    ◦ Description copied from EngineUpdate all entities managed by the engine.
    ◦ Parameters
        ■ timeElapsed -- - amount of time since last update.

```

Class PierEngine.ClientConnectionListener

Declaration

```
class PierEngine.ClientConnectionListener
extends java.lang.Object
```

Constructor summary

- `PierEngine.ClientConnectionListener()`

Method summary

- `connected(JGNConnection)`
- `disconnected(JGNConnection)`

Constructors

- `PierEngine.ClientConnectionListener`
`PierEngine.ClientConnectionListener()`

Methods

- `connected`
`public void connected(JGNConnection conn)`
- `disconnected`
`public void disconnected(JGNConnection conn)`

Class PierEngine.GameLoop

Loop for main thread

Declaration

```
class PierEngine.GameLoop
extends java.lang.Thread
```

Constructor summary

- `PierEngine.GameLoop()`

Method summary

- `isPaused()`
- `pause()`
- `restart()`
- `run()`
- `start()`

Constructors

- `PierEngine.GameLoop`
`PierEngine.GameLoop()`

Methods

- **isPaused**
public boolean **isPaused**()
- **pause**
public synchronized void **pause**()
- **restart**
public synchronized void **restart**()
- **run**
void **run**()
- **start**
public synchronized void **start**()

Members inherited from class java.lang.Thread

- static void ()
- public static int **activeCount**()
- public final void **checkAccess**()
- public native int **countStackFrames**()
- public static native Thread **currentThread**()
- public void **destroy**()
- public static void **dumpStack**()
- public static int **enumerate**(Thread[] **arg0**)
- public static Map **getAllStackTraces**()
- public ClassLoader **getContextClassLoader**()
- public static Thread.UncaughtExceptionHandler **getDefaultUncaughtExceptionHandler**()
- public long **getId**()
- public final String **getName**()
- public final int **getPriority**()
- public StackTraceElement **getStackTrace**()
- public Thread.State **getState**()
- public final ThreadGroup **getThreadGroup**()
- public Thread.UncaughtExceptionHandler **getUncaughtExceptionHandler**()
- public static native boolean **holdsLock**(Object **arg0**)
- **inheritableThreadLocals**
- public void **interrupt**()
- public static boolean **interrupted**()
- public final native boolean **isAlive**()
- public final boolean **isDaemon**()
- public boolean **isInterrupted**()
- public final void **join**() throws InterruptedException
- public final synchronized void **join**(long **arg0**) throws InterruptedException
- public final synchronized void **join**(long **arg0**, int **arg1**) throws InterruptedException
- public static final **MAX_PRIORITY**
- public static final **MIN_PRIORITY**
- public static final **NORM_PRIORITY**
- public final void **resume**()
- public void **run**()
- public void **setContextClassLoader**(ClassLoader **arg0**)
- public final void **setDaemon**(boolean **arg0**)
- public static void **setDefaultUncaughtExceptionHandler**(Thread.UncaughtExceptionHandler **arg0**)
- public final void **setName**(String **arg0**)
- public final void **setPriority**(int **arg0**)
- public void **setUncaughtExceptionHandler**(Thread.UncaughtExceptionHandler **arg0**)
- public static native void **sleep**(long **arg0**) throws InterruptedException

- `public static void sleep(long arg0, int arg1) throws InterruptedException`
- `public synchronized void start()`
- `public final void stop()`
- `public final synchronized void stop(Throwable arg0)`
- `public final void suspend()`
- **threadLocals**
- `public String toString()`
- `public static native void yield()`

Class `PierEngine.SystemMessageListener`

Declaration

```
class PierEngine.SystemMessageListener
extends DynamicMessageAdapter
```

Constructor summary

- `PierEngine.SystemMessageListener()`

Method summary

- `messageReceived(Message)`

Constructors

- `PierEngine.SystemMessageListener`
`public PierEngine.SystemMessageListener()`

Methods

- `messageReceived`
`public void messageReceived(Message msg)`

Class `PierGraphicalController`

Declaration

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

Constructor summary

- `PierGraphicalController()`

Method summary

- `applySynchronizationMessage(SynchronizeMessage, Entity)`
- `createSynchronizationMessage(Entity)`
- `proximity(Entity, short)`
- `validateMessage(SynchronizeMessage, Entity)`

Constructors

- `PierGraphicalController`
`public PierGraphicalController()`

Methods

- `applySynchronizationMessage`
`public void applySynchronizationMessage(SynchronizeMessage arg0, Entity arg1)`
- `createSynchronizationMessage`
`public SynchronizeMessage createSynchronizationMessage(Entity arg0)`
- `proximity`
`public float proximity(Entity arg0, short arg1)`
- `validateMessage`
`public boolean validateMessage(SynchronizeMessage arg0, Entity arg1)`

Package pier436.client

Package Contents

- Interfaces
 - `GameClient`

Interface GameClient

Declaration

```
public interface GameClient
```

Method summary

- `connect(URL)`
- `disconnect()`
- `getNetworkAddress()`
- `getServerAddress()`
- `initialize()`
- `isConnected()`
- `send(NetworkMessage)`

Methods

- `connect`
`void connect(java.net.URL u)`

- **disconnect**
void **disconnect**()
- **getNetworkAddress**
java.net.URL **getNetworkAddress**()
- **getServerAddress**
java.net.URL **getServerAddress**()
- **initialize**
void **initialize**()
- **isConnected**
boolean **isConnected**()
- **send**
void **send**(NetworkMessage n)

Package pier436.controllogic

Package Contents

- **Interfaces**
 - ControlLogic
- **Classes**
 - ConstrainedMotionLogic
 - MovingFunction ABC for a patterned movement function.
 - PatternedMotionLogic Satisfies 4.2.3
 - RandomControlLogic Supports 4.2.1

Interface ControlLogic

Declaration

```
public interface ControlLogic
```

All known subinterfaces

ConstrainedMotionLogic, PatternedMotionLogic, RandomControlLogic

All classes known to implement interface

ConstrainedMotionLogic, PatternedMotionLogic, RandomControlLogic

Method summary

- **nextMove()** Use this to decide where the entity should move to next nextMove Vector2d

Methods

- **nextMove**
javax.vecmath.Vector2d **nextMove**()
 - **Description** Use this to decide where the entity should move to next nextMove Vector2d

- Returns --

Class ConstrainedMotionLogic

Declaration

```
public class ConstrainedMotionLogic
extends java.lang.Object
implements ControlLogic
```

Constructor summary

- **ConstrainedMotionLogic()**

Method summary

- **nextMove()**
- **nextMove(Point2d)**

Constructors

- **ConstrainedMotionLogic**
`public ConstrainedMotionLogic()`

Methods

- **nextMove**
`javax.vecmath.Vector2d nextMove()`
 - **Description copied from ControlLogic** Use this to decide where the entity should move to next
`nextMove Vector2d`
 - **Returns --**
- **nextMove**
`public javax.vecmath.Vector2d nextMove(javax.vecmath.Point2d currentPosition)`

Class MovingFunction

ABC for a patterned movement function.

Declaration

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

Constructor summary

- **MovingFunction()**

Method summary

- `moveFrom(Point2d)`

Constructors

- **MovingFunction**
`public MovingFunction()`

Methods

- **moveFrom**
`public abstract javax.vecmath.Vector2d moveFrom(javax.vecmath.Point2d position)`
 - **Parameters**
 - `position` -- original position
 - **Returns** -- movement from original position.

Class PatternedMotionLogic

Satisfies 4.2.3

Declaration

```
public class PatternedMotionLogic
extends java.lang.Object
implements ControlLogic
```

Field summary

- `function`
- `lastPosition`

Constructor summary

- **PatternedMotionLogic(MovingFunction)** TODO: Add support for patterned motion function.

Method summary

- `nextMove()` TODO:delete?
- `nextMove(Point2d)` Return a new position based on specified pattern

Fields

- `javax.vecmath.Point2d lastPosition`
- `MovingFunction function`

Constructors

- **PatternedMotionLogic**

```
public PatternedMotionLogic( MovingFunction function )  
  ○ Description TODO: Add support for patterned motion function.
```

Methods

- **nextMove**

```
public javax.vecmath.Vector2d nextMove( )  
  ○ Description TODO:delete?  
  ○ See also  
    ■ ControlLogic.nextMove
```

- **nextMove**

```
public javax.vecmath.Vector2d nextMove( javax.vecmath.Point2d currentPosition )  
  ○ Description Return a new position based on specified pattern  
  ○ Returns -- new position.
```

Class RandomControlLogic

Supports 4.2.1

Declaration

```
public class RandomControlLogic  
  extends java.lang.Object  
  implements ControlLogic
```

Constructor summary

- **RandomControlLogic()**

Method summary

- **nextMove()** Randomly output a new direction for the entity nextMove Vector2d

Constructors

- **RandomControlLogic**

```
public RandomControlLogic( )
```

Methods

- **nextMove**

```
public javax.vecmath.Vector2d nextMove( )  
  ○ Description Randomly output a new direction for the entity nextMove Vector2d  
  ○ Returns --
```

Package pier436.entity

Package Contents

- Interfaces
 - ControlledEntity
 - HumanControlledEntity
 - ImpassibleEntity
 - MovableEntity
 - PassableEntity
 - SharedEntity Satisfies 3.4.2
- Classes
 - Entity

Interface ControlledEntity

Declaration

```
public interface ControlledEntity
extends MovableEntity
```

All known subinterfaces

HumanControlledEntity

Method summary

- **getNextMove()** Use the control logic to get the next move for the entity.
- **setControlLogic(ControlLogic)** Set the control logic for the entity, which controls its movement.

Methods

- **getNextMove**
void **getNextMove**()
 - **Description** Use the control logic to get the next move for the entity. getNextMove void
- **setControlLogic**
void **setControlLogic**(ControlLogic **cl**)
 - **Description** Set the control logic for the entity, which controls its movement. setControlLogic void
 - **Parameters**
 - **cl** --

Members inherited from class pier436.entity.MovableEntity

- public void **moveTo**(double **dx**, double **dy**)
- public void **moveX**(double **dx**)
- public void **moveY**(double **dy**)

Interface HumanControlledEntity

Declaration

```
public interface HumanControlledEntity
extends ControlledEntity
```

Members inherited from class `pier436.entity.ControlledEntity`

- `public void getNextMove()`
- `public void setControlLogic(ControlLogic cl)`

Members inherited from class `pier436.entity.MovableEntity`

- `public void moveTo(double dx, double dy)`
- `public void moveX(double dx)`
- `public void moveY(double dy)`

Interface `ImpassibleEntity`

Declaration

```
public interface ImpassibleEntity
```

Interface `MovableEntity`

Declaration

```
public interface MovableEntity
```

All known subinterfaces

ControlledEntity, HumanControlledEntity

Method summary

- **`moveTo(double, double)`** Move the entity as far as delta x and delta y.
- **`moveX(double)`** Move the entity's x coordinate as far as delta x.
- **`moveY(double)`** Move the entity's y coordinate as far as delta y.

Methods

- **`moveTo`**
void `moveTo(double dx, double dy)`
 - **Description** Move the entity as far as delta x and delta y. moveTo void
 - **Parameters**
 - dx --
 - dy --
- **`moveX`**
void `moveX(double dx)`

- **Description** Move the entity's x coordinate as far as delta x. `moveX` void
- **Parameters**
 - `dx` --
- **moveY**
`void moveY(double dy)`
 - **Description** Move the entity's y coordinate as far as delta y. `moveY` void
 - **Parameters**
 - `dy` --

Interface `PassableEntity`

See also

- `ImpassibleEntity`

Declaration

```
public interface PassableEntity
```

Interface `SharedEntity`

Satisfies 3.4.2

Declaration

```
public interface SharedEntity
```

Class `Entity`

Declaration

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

Constructor summary

- `Entity()`

Method summary

- `getX()` Get the entity's x coordinate.
- `getY()` Get the entity's y coordinate.
- `setEventActionLogic(EventActionLogic)` Set the event action logic which handles what happens during a collision.

Constructors

- `Entity`

```
public Entity( )
```

Methods

- **getX**
public double **getX**()
 - **Description** Get the entity's x coordinate. getX double
 - **Returns** --
- **getY**
public double **getY**()
 - **Description** Get the entity's y coordinate. getY double
 - **Returns** --
- **setEventActionLogic**
public void **setEventActionLogic**(EventActionLogic **ea**)
 - **Description** Set the event action logic which handles what happens during a collision. setEventActionLogic void
 - **Parameters**
 - **ea** --

Package pier436.entitymanager

Package Contents

- **Interfaces**
 - EntityManager
- **Classes**
 - PierEntityManager

Interface EntityManager

Declaration

```
public interface EntityManager
```

All known subinterfaces

PierEntityManager

All classes known to implement interface

PierEntityManager

Method summary

- **add(Entity)**
- **add(Entity, boolean)**
- **getEntities(Class)** Get all entities of a given type

- `getEntities(Point2D, double)`
- `remove(Entity)`
- `share(Entity)`

Methods

- `add`
`void add(Entity entity)`
- `add`
`void add(Entity entity, boolean share)`
- `getEntities`
`java.util.ArrayList getEntities(java.lang.Class entityType)`
 - **Description** Get all entities of a given type
 - **Parameters**
 - `entityType` --
 - **Returns** --
- `getEntities`
`java.util.ArrayList getEntities(java.awt.geom.Point2D location, double distance)`
- `remove`
`void remove(Entity entity)`
- `share`
`void share(Entity entity)`

Class PierEntityManager

Declaration

```
public class PierEntityManager
extends java.lang.Object
implements EntityManager
```

Constructor summary

- `PierEntityManager()`

Method summary

- `add(Entity)`
- `add(Entity, boolean)`
- `applySynchronizationMessage(SynchronizeMessage, Entity)`
- `createSynchronizationMessage(Entity)`
- `getEntities(Class)`
- `getEntities(Point2D, double)`
- `proximity(Entity, short)`
- `remove(Entity)`
- `share(Entity)`
- `validateMessage(SynchronizeMessage, Entity)`

Constructors

- `PierEntityManager`
`public PierEntityManager()`

Methods

- **add**
void **add**(Entity **entity**)
- **add**
void **add**(Entity **entity**, boolean **share**)
- **applySynchronizationMessage**
public void **applySynchronizationMessage**(SynchronizeMessage **arg0**, Entity **arg1**)
- **createSynchronizationMessage**
public SynchronizeMessage **createSynchronizationMessage**(Entity **arg0**)
- **getEntities**
java.util.ArrayList **getEntities**(java.lang.Class **entityType**)
 - Description copied from EntityManagerGet all entities of a given type
 - Parameters
 - entityType --
 - Returns --
- **getEntities**
java.util.ArrayList **getEntities**(java.awt.geom.Point2D **location**, double **distance**)
- **proximity**
public float **proximity**(Entity **arg0**, short **arg1**)
- **remove**
void **remove**(Entity **entity**)
- **share**
void **share**(Entity **entity**)
- **validateMessage**
public boolean **validateMessage**(SynchronizeMessage **arg0**, Entity **arg1**)

Package pier436.server

Package Contents

- Interfaces
 - GameServer

Interface GameServer

Declaration

```
public interface GameServer
```

Method summary

- **broadcast(NetworkMessage)**
- **getClientAddresses()**
- **getNetworkAddress()**
- **isRunning()**
- **startServer()**
- **stopServer()**

Methods

- **broadcast**
void **broadcast**(NetworkMessage n)
- **getClientAddresses**
java.util.ArrayList **getClientAddresses**()
- **getNetworkAddress**
java.net.URL **getNetworkAddress**()
- **isRunning**
boolean **isRunning**()
- **startServer**
void **startServer**()
- **stopServer**
void **stopServer**()

Package pier436.util

Package Contents

- **Interfaces**
 - Camera
 - EventActionLogic
- **Classes**
 - EngineMessage
 - NetworkMessage
 - SystemMessage

Interface Camera

Declaration

```
public interface Camera
```

Method summary

- **getCanvas()** Get a Canvas object that is embeddable into a UI.
- **height()** Get current camera height 1.2.4.2
- **height(double)** Set height of camera, relative to map surface
- **pan(double, double)** Move camera by specified amount 1.2.4.1
- **Position()** Current camera position
- **rotate(double)** Rotate camera view by specified degrees 1.2.4.3
- **zoom(double)** Zoom in or out (< 1, > 1) 1.2.4.2

Methods

- **getCanvas**
java.awt.Canvas **getCanvas**()
 - **Description** Get a Canvas object that is embeddable into a UI. This will be updated by the game loop. 1.1.1

- Returns --
- **height**
double **height**()
 - **Description** Get current camera height 1.2.4.2
 - **Returns** -- camera height
- **height**
void **height**(double **height**)
 - **Description** Set height of camera, relative to map surface
 - **Parameters**
 - **height** -- desired height of camera, greater than zero.
- **pan**
void **pan**(double **dx**, double **dy**)
 - **Description** Move camera by specified amount 1.2.4.1
 - **Parameters**
 - **dx** -- distance east west
 - **dy** -- distance north south
- **Position**
java.awt.Point **Position**()
 - **Description** Current camera position
 - **Returns** -- camera position (x,y)
- **rotate**
void **rotate**(double **angle**)
 - **Description** Rotate camera view by specified degrees 1.2.4.3
 - **Parameters**
 - **angle** -- angle to rotate in degrees
- **zoom**
void **zoom**(double **ratio**)
 - **Description** Zoom in or out (< 1, > 1) 1.2.4.2
 - **Parameters**
 - **ratio** -- zoom ratio.

Interface EventActionLogic

Declaration

```
public interface EventActionLogic
```

Method summary

- **collided(Entity)** Called on collided void

Methods

- **collided**
void **collided**(Entity **e**)
 - **Description** Called on collided void
 - **Parameters**
 - **e** --

Class EngineMessage

Declaration

```
public class EngineMessage  
extends pier436.util.SystemMessage
```

Constructor summary

- `EngineMessage()`

Constructors

- `EngineMessage`
`public EngineMessage()`

Class NetworkMessage

Declaration

```
public abstract class NetworkMessage  
extends java.lang.Object  
implements java.io.Serializable, java.lang.Comparable
```

Field summary

- `priority`
- `timestamp`

Constructor summary

- `NetworkMessage(int)`

Method summary

- `compareTo(Object)`
- `getLatency()`
- `getPriority()`

Fields

- protected long `timestamp`
- protected int `priority`

Constructors

- `NetworkMessage`
`public NetworkMessage(int priority)`

Methods

- **compareTo**
public int **compareTo**(java.lang.Object **obj**)
- **getLatency**
public long **getLatency**()
- **getPriority**
public int **getPriority**()

Class SystemMessage

Declaration

```
public class SystemMessage  
extends Message
```

All known subclasses

EngineMessage

Constructor summary

- **SystemMessage()**

Constructors

- **SystemMessage**
public **SystemMessage**()