

---

**Package**  
**arena**

## arena

### Class Arena

```

java.lang.Object
  |
  +- java.util.Observable
        |
        +- arena.Arena
  
```

#### All Implemented Interfaces:

```
java.lang.Runnable
```

```

public class Arena
  extends java.util.Observable
  implements java.lang.Runnable
  
```

Class that will hold the shared behaviour between the various arena types. Since each arena can implement it's own rules and such, the only thing that will differ from most arenas to one another is when to calculate the arena has ended. This class is observable, since if the arena is being displayed, then it will be required to update its observer every tick. Each arena is threaded, so that we can run multiple arenas at once later on down the line concurrently.

### Constructor Summary

public	<a href="#">Arena</a> (int maxAsteroidCount, int asteroidSpawnChance, int tickDelay) Create a new AbstractArena with the following parameters.
--------	---

### Method Summary

void	<a href="#">addObjectToArena</a> ( <a href="#">AbstractObject</a> object) Adds a new object to the arena.
void	<a href="#">addShipToArena</a> ( <a href="#">AbstractShip</a> shipToAdd) This method will add a given ship to the arena, it will do this by copying it into a new ship, and adding it into the score map.
java.util.ArrayList	<a href="#">getArenaObjects</a> () Get the list of objects that currently exist in the arena.
<a href="#">ArenaWatcher</a>	<a href="#">getArenaWatcher</a> () Get the arena watcher currently tied to this arena.
<a href="#">CollisionManager</a>	<a href="#">getCollisionManager</a> () Get the current collision manager of this arena.
java.util.ArrayList	<a href="#">getInitialShips</a> () Get the initial ships that were used to create this arena, the non-cloned ones.
boolean	<a href="#">isGameRunning</a> () Retrieve whether or not the game is currently running, useful for UI elements.
void	<a href="#">run</a> ()
boolean	<a href="#">updateGameStatus</a> (java.util.ArrayList gameShips) Update the current status of the game given the list of the ships competing.

#### Methods inherited from class java.util.Observable

```
addObserver, countObservers, deleteObserver, deleteObservers, hasChanged,
notifyObservers, notifyObservers
```

#### Methods inherited from class `java.lang.Object`

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

#### Methods inherited from interface `java.lang.Runnable`

```
run
```

## Constructors

### Arena

```
public Arena(int maxAsteroidCount,
              int asteroidSpawnChance,
              int tickDelay)
```

Create a new AbstractArena with the following parameters.

#### Parameters:

`maxAsteroidCount` - Maximum asteroids that can spawn in the game.

`asteroidSpawnChance` - Percentage chance of asteroid spawning in this arena every tick. ( 1- 100 )

`tickDelay` - Amount of delay the thread should have in between every game-tick.

## Methods

### run

```
public void run()
```

### updateGameStatus

```
public boolean updateGameStatus(java.util.ArrayList gameShips)
```

Update the current status of the game given the list of the ships competing.

#### Parameters:

`gameShips` - Current ships in the arena.

#### Returns:

True if the game is still running, false otherwise.

### getArenaObjects

```
public java.util.ArrayList getArenaObjects()
```

Get the list of objects that currently exist in the arena.

#### Returns:

List of all objects.

---

## getCollisionManager

```
public CollisionManager getCollisionManager()
```

Get the current collision manager of this arena.

**Returns:**

Collision manager currently being used by this arena.

---

## getArenaWatcher

```
public ArenaWatcher getArenaWatcher()
```

Get the arena watcher currently tied to this arena.

**Returns:**

Arena watcher created at arena run time.

---

## addShipToArena

```
public void addShipToArena(AbstractShip shipToAdd)
```

This method will add a given ship to the arena, it will do this by copying it into a new ship, and adding it into the score map.

**Parameters:**

shipToAdd - Ship to add to the arena.

---

## addObjectToArena

```
public void addObjectToArena(AbstractObject object)
```

Adds a new object to the arena. If the game is already running, then it will que it to be added. Otherwise it will add it.

**Parameters:**

object - Object to be added.

---

## isGameRunning

```
public boolean isGameRunning()
```

Retrieve whether or not the game is currently running, useful for UI elements.

**Returns:**

Whether the game logic loops is still running.

---

## getInitialShips

```
public java.util.ArrayList getInitialShips()
```

Get the initial ships that were used to create this arena, the non-cloned ones.

**Returns:**

List generated when adding ships to the arena.

---

## arena

### Class ArenaWatcher

```
java.lang.Object
```

```
└--arena.ArenaWatcher
```

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

#### Constructor Summary

public	<a href="#"><code>ArenaWatcher()</code></a> Create a new arena watcher, just creates all of the lists that are needed.
--------	---

#### Method Summary

double	<a href="#"><code>angleToNearestAsteroid(AbstractObject checkObject)</code></a> Calculates the angle from a given object to the nearest asteroid.
double	<a href="#"><code>angleToNearestEnemy(AbstractObject checkObject)</code></a> Calculates the angle from a given object to the nearest enemy.
double	<a href="#"><code>angleToNearestEnemyBullet(AbstractObject checkObject)</code></a> Calculates the angle from a given object to the nearest enemy bullet.
double	<a href="#"><code>angleToObject(AbstractObject check, AbstractObject object)</code></a> Calculate the angle to a given object inside the arena.
double	<a href="#"><code>distanceToNearestAsteroid(AbstractObject check)</code></a> Get the distance to the nearest asteroid.
double	<a href="#"><code>distanceToNearestBullet(AbstractObject check)</code></a> Get the distance to the nearest enemy bullet.
double	<a href="#"><code>distanceToNearestEnemy(AbstractObject check)</code></a> Get the distance to the nearest enemy ship.
<a href="#"><code>Asteroid</code></a>	<a href="#"><code>getNearestAsteroid(AbstractObject check)</code></a> Get the asteroid nearest to the given object.
<a href="#"><code>Bullet</code></a>	<a href="#"><code>getNearestEnemyBullet(AbstractObject check)</code></a> Get the bullet nearest to the given object, which wasn't created by the given object.
<a href="#"><code>AbstractShip</code></a>	<a href="#"><code>getNearestShip(AbstractObject check)</code></a> Gets the ship nearest to the given object.
int	<a href="#"><code>getShipBulletCount(AbstractShip ship)</code></a> Get the number of bullets currently in the arena that belong to a given ship.
double	<a href="#"><code>normalisedAngleNearestAsteroid(AbstractObject checkObject)</code></a> Calculate the angle to the nearest asteroid and return it normalised.

double	<a href="#"><code>normalisedAngleNearestEnemy</code></a> ( <a href="#"><code>AbstractObject</code></a> checkObject) Calculate the angle to the nearest ship and return it normalised.
double	<a href="#"><code>normalisedAngleNearestEnemyBullet</code></a> ( <a href="#"><code>AbstractObject</code></a> checkObject) Calculate the angle to the nearest enemy bullet and return it normalised.
double	<a href="#"><code>normalisedDistanceNearestAsteroid</code></a> ( <a href="#"><code>AbstractObject</code></a> check) Calculate the distance to the nearest asteroid and return it normalised.
double	<a href="#"><code>normalisedDistanceNearestEnemy</code></a> ( <a href="#"><code>AbstractObject</code></a> check) Calculate the distance to the nearest enemy and return it normalised.
double	<a href="#"><code>normalisedDistanceNearestEnemyBullet</code></a> ( <a href="#"><code>AbstractObject</code></a> check) Calculate the distance to the nearest asteroid and return it normalised.
void	<a href="#"><code>setObjects</code></a> (java.util.List gameObjects) Set the objects that the arena watcher should be using for this given game tick.

Methods inherited from class `java.lang.Object`

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

## Constructors

### ArenaWatcher

```
public ArenaWatcher()
```

Create a new arena watcher, just creates all of the lists that are needed.

## Methods

### setObjects

```
public void setObjects(java.util.List gameObjects)
```

Set the objects that the arena watcher should be using for this given game tick.

#### Parameters:

`gameObjects` - Game objects to be used by the methods.

### getNearestShip

```
public AbstractShip getNearestShip(AbstractObject check)
```

Gets the ship nearest to the given object.

#### Parameters:

`check` - Object to check.

#### Returns:

Nearest ship if it exists, null otherwise.

(continued from last page)

---

## getNearestAsteroid

```
public Asteroid getNearestAsteroid(AbstractObject check)
```

Get the asteroid nearest to the given object.

**Parameters:**

check - Object to check.

**Returns:**

Nearest asteroid if it exists, null otherwise.

---

## getNearestEnemyBullet

```
public Bullet getNearestEnemyBullet(AbstractObject check)
```

Get the bullet nearest to the given object, which wasn't created by the given object.

**Parameters:**

check - Object to check.

**Returns:**

Nearest bullet if it exists, null otherwise.

---

## distanceToNearestEnemy

```
public double distanceToNearestEnemy(AbstractObject check)
```

Get the distance to the nearest enemy ship. If no enemy ship can be found, then it returns 0.

**Parameters:**

check - Object to calculate nearest distance to.

**Returns:**

Distance to the nearest enemy ship, or 0 if no ship can be found.

---

## distanceToNearestAsteroid

```
public double distanceToNearestAsteroid(AbstractObject check)
```

Get the distance to the nearest asteroid. If no asteroid can be found, then it returns Double.POSITIVE\_INFINITY.

**Parameters:**

check - Object to calculate nearest distance to.

**Returns:**

Distance to the nearest asteroid, or 0 if no asteroid can be found.

---

## distanceToNearestBullet

```
public double distanceToNearestBullet(AbstractObject check)
```

Get the distance to the nearest enemy bullet. If no bullet can be found, then it returns Double.POSITIVE\_INFINITY.

**Parameters:**

check - Object to calculate nearest distance to.

**Returns:**

Distance to the nearest bullet, or 0 if no bullet can be found.

---

---

## normalisedDistanceNearestEnemy

```
public double normalisedDistanceNearestEnemy(AbstractObject check)
```

Calculate the distance to the nearest enemy and return it normalised.

**Parameters:**

check - Object to calculate distance from.

**Returns:**

Normalised value of the distance to nearest enemy, infinity if no ship nearby.

---

## normalisedDistanceNearestAsteroid

```
public double normalisedDistanceNearestAsteroid(AbstractObject check)
```

Calculate the distance to the nearest asteroid and return it normalised.

**Parameters:**

check - Object to calculate distance from.

**Returns:**

Normalised value of the distance to nearest asteroid, infinity if no asteroid nearby.

---

## normalisedDistanceNearestEnemyBullet

```
public double normalisedDistanceNearestEnemyBullet(AbstractObject check)
```

Calculate the distance to the nearest asteroid and return it normalised.

**Parameters:**

check - Object to calculate distance from.

**Returns:**

Normalised value of the distance to nearest asteroid, infinity if no asteroid nearby.

---

## angleToObject

```
public double angleToObject(AbstractObject check,  
    AbstractObject object)
```

Calculate the angle to a given object inside the arena.

**Parameters:**

check - Object to calculate the angle from.

object - Object to calculate the angle to.

**Returns:**

-ve if objects need to turn RIGHT, and +ve if object needs to turn LEFT.

---

## angleToNearestEnemy

```
public double angleToNearestEnemy(AbstractObject checkObject)
```

Calculates the angle from a given object to the nearest enemy.

**Parameters:**

checkObject - Object to calculate angle from.

---



**Returns:**

Angle to the nearest enemy, 0 if no ship exists.

---

**angleToNearestAsteroid**

```
public double angleToNearestAsteroid(AbstractObject checkObject)
```

Calculates the angle from a given object to the nearest asteroid.

**Parameters:**

`checkObject` - Object to calculate angle from.

**Returns:**

Angle to the nearest enemy, 0 if no asteroid exists.

---

**angleToNearestEnemyBullet**

```
public double angleToNearestEnemyBullet(AbstractObject checkObject)
```

Calculates the angle from a given object to the nearest enemy bullet.

**Parameters:**

`checkObject` - Object to calculate angle from.

**Returns:**

Angle to the nearest enemy bullet, 0 if no bullet exists.

---

**normalisedAngleNearestEnemy**

```
public double normalisedAngleNearestEnemy(AbstractObject checkObject)
```

Calculate the angle to the nearest ship and return it normalised. The normalised angle has to be between -1 and 1.

**Parameters:**

`check` - Object to calculate angle from.

**Returns:**

Normalised value of the angle to nearest ship, infinity if no ship nearby.

---

**normalisedAngleNearestAsteroid**

```
public double normalisedAngleNearestAsteroid(AbstractObject checkObject)
```

Calculate the angle to the nearest asteroid and return it normalised. The normalised angle has to be between -1 and 1.

**Parameters:**

`check` - Object to calculate angle from.

**Returns:**

Normalised value of the angle to nearest asteroid, infinity if no asteroid nearby.

---

**normalisedAngleNearestEnemyBullet**

```
public double normalisedAngleNearestEnemyBullet(AbstractObject checkObject)
```

Calculate the angle to the nearest enemy bullet and return it normalised. The normalised angle has to be between -1 and 1.

---

(continued from last page)

**Parameters:**

check - Object to calculate angle from.

**Returns:**

Normalised value of the angle to nearest enemy bullet, infinity if no bullet nearby.

---

## getShipBulletCount

```
public int getShipBulletCount(AbstractShip ship)
```

Get the number of bullets currently in the arena that belong to a given ship.

**Parameters:**

ship - Ship to check.

**Returns:**

Number of bullets that are owned by that object.

## arena

### Class BatchArena

```
java.lang.Object
└--arena.BatchArena
```

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

Creates a new batch arena, the aim of this class is to be able to manage a thread pool and run all of the arenas at the same time. In order to do this, I will use an `ExecutorService`, which allows me to easily run a bounded thread pool. The batch arena will take in a list of ships that will be competing across all of the games, along with how many games will be running. This class will later hold methods to retrieve the score of each set of batch arenas.

### Constructor Summary

public	<a href="#"><code>BatchArena</code></a> (java.util.ArrayList batchShips, int numberOfGames, int maxAsteroids, int asteroidSpawnChance) Creates a new batch arena.
--------	--

### Method Summary

boolean	<a href="#"><code>isBatchRunning</code></a> () Get whether or not the thread pool is still executing.
void	<a href="#"><code>startBatch</code></a> () Start this batch.

### Methods inherited from class java.lang.Object

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

## Constructors

### BatchArena

```
public BatchArena(java.util.ArrayList batchShips,
                  int numberOfGames,
                  int maxAsteroids,
                  int asteroidSpawnChance)
```

Creates a new batch arena.

#### Parameters:

- `batchShips` - Ship to be used in the arenas.
- `numberOfGames` - Number of games to be played by each ship.
- `maxAsteroids` - Maximum number of asteroids that can occur in each arena spawned.
- `asteroidSpawnChance` - Chance of asteroids spawning in each arena ( 1 - 100% each tick ).

## Methods

(continued from last page)

## **startBatch**

```
public void startBatch()
```

Start this batch.

---

## **isBatchRunning**

```
public boolean isBatchRunning()
```

Get whether or not the thread pool is still executing.

### **Returns:**

Whether the thread pool is terminated.

## arena

### Class EScoring

```

java.lang.Object
  |
  +- java.lang.Enum
        +- arena.EScoring
  
```

#### All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

```

public final class EScoring
extends java.lang.Enum
  
```

An enum to help with the scoring conventions. Easily scalable/changeable.

### Field Summary

public static final	<a href="#">ASTEROID_HIT</a>
public static final	<a href="#">SHIP_HIT</a>
public static final	<a href="#">SURVIVING_BONUS</a>

### Method Summary

int	<a href="#">getScore()</a>
static <a href="#">EScoring</a>	<a href="#">valueOf(java.lang.String name)</a>
static <a href="#">EScoring[]</a>	<a href="#">values()</a>

#### Methods inherited from class java.lang.Enum

compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

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

#### Methods inherited from interface java.lang.Comparable

compareTo

## Fields

### ASTEROID\_HIT

```
public static final arena.EScoring ASTEROID_HIT
```

(continued from last page)

---

## SHIP\_HIT

```
public static final arena.EScoring SHIP_HIT
```

---

## SURVIVING\_BONUS

```
public static final arena.EScoring SURVIVING_BONUS
```

## Methods

### values

```
public static EScoring\[\] values()
```

---

### valueOf

```
public static EScoring valueOf(java.lang.String name)
```

---

### getScore

```
public int getScore()
```

---

**Package**  
**arena.collisions**

## arena.collisions

### Class Collision

```
java.lang.Object
|
+--arena.collisions.Collision
```

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

A helper class to easily set and determine who collided with who. Tye type of collision is used to easily determine what's colliding with what when dealing with collisions.

### Constructor Summary

public	<a href="#">Collision</a> ( <a href="#">AbstractObject</a> object, <a href="#">AbstractObject</a> collidingWith, <a href="#">ECollisionTypes</a> colType) Create a new collision between two objects.
--------	--

### Method Summary

<a href="#">AbstractObject</a>	<a href="#">getCollidingWithObject</a> () Get the object that was collided with.
<a href="#">ECollisionTypes</a>	<a href="#">getCollisionType</a> () Get the type of collision that this collision represents.
<a href="#">AbstractObject</a>	<a href="#">getObject</a> () Get the object that iniated the collision.

### Methods inherited from class java.lang.Object

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

## Constructors

### Collision

```
public Collision(AbstractObject object,
               AbstractObject collidingWith,
               ECollisionTypes colType)
```

Create a new collision between two objects.

#### Parameters:

object  
collidingWith

## Methods

### getObject

```
public AbstractObject getObject()
```



(continued from last page)

Get the object that initiated the collision.

**Returns:**

AbstractObject that collided into the other object.

---

## getCollidingWithObject

```
public AbstractObject getCollidingWithObject()
```

Get the object that was collided with.

**Returns:**

AbstractObject that was collided into.

---

## getCollisionType

```
public ECollisionTypes getCollisionType()
```

Get the type of collision that this collision represents.

**Returns:**

Collision type was given by the enum ECollisionTypes

## arena.collisions

# Class CollisionManager

```
java.lang.Object
```

```
└--arena.collisions.CollisionManager
```

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

The collision watcher is a class which will be able to detect, and deal with collisions. It is created with a reference to the arena that it is currently watching. This is to make collision watching be thread-safe.

## Constructor Summary

public	<a href="#">CollisionManager</a> ( <a href="#">Arena</a> arena) Generate a new collision manager.
--------	--

## Method Summary

void	<a href="#">addCollision</a> ( <a href="#">Collision</a> collision) Add a new collision that has to be dealt with this tick, in order to add this, it makes sure the collision doesn't already exist.
void	<a href="#">checkForCollisions</a> () Go through each object in the game and see if it collides with anything else.
boolean	<a href="#">isMoveValid</a> ( <a href="#">AbstractObject</a> objectToCheck) Checks to see whether or not a position is safe to move to.

## Methods inherited from class java.lang.Object

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

## Constructors

### CollisionManager

```
public CollisionManager(Arena arena)
```

Generate a new collision manager. With the given arena as a reference.

#### Parameters:

arena - Arena this class is managing collisions for.

## Methods

### addCollision

```
public void addCollision(Collision collision)
```

Add a new collision that has to be dealt with this tick, in order to add this, it makes sure the collision doesn't already exist. This shouldn't be an issue, but just incase it's there.

(continued from last page)

**Parameters:**

`collision` - Collision to add to the system.

---

**isMoveValid**

```
public boolean isMoveValid(AbstractObject objectToCheck)
```

Checks to see whether or not a position is safe to move to. This is mainly for ships to use, and the value checking is determined by this. The "magic numbers" 0.035 is the width of a given ship. So first of all it checks that any of the given points of the ship are going to be outside of the bounds. It then checks to see if it collides with any other given object. This method is useful as it allows ships to check if they need to reverse after moving. To avoid colliding or "warping" inside another object.

**Parameters:**

`objectToCheck` - Object to check for safe movement.

**Returns:**

Is the move safe.

---

**checkForCollisions**

```
public void checkForCollisions()
```

Go through each object in the game and see if it collides with anything else.

## arena.collisions

### Class CollisionPolygon

java.lang.Object

└─arena.collisions.CollisionPolygon

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

#### Constructor Summary

public	<a href="#">CollisionPolygon</a> (java.awt.geom.Point2D.Double objectCentre, int verticeCount) Create a new collision polygon.
--------	---

#### Method Summary

void	<a href="#">addVertice</a> (double x, double y) Add a new vertice to this collision polygon.
void	<a href="#">calculateInterval</a> ( <a href="#">Vector</a> axis) Calculate the interval between this polygon to the given axis.
boolean	<a href="#">collide</a> ( <a href="#">CollisionPolygon</a> poly) Determine if this collision polygon is colliding with another.
int	<a href="#">getVerticeCount</a> () Get the number of vertices in this collision polygon.
boolean	<a href="#">intervalsSeparated</a> (float mina, float maxa, float minb, float maxb) Method to determine if the given intervals for two vectors are seperated.
boolean	<a href="#">separatedByAxis</a> ( <a href="#">Vector</a> axis, <a href="#">CollisionPolygon</a> poly) Determine to see if a given axis and another collision polygon are seperated by an axis.
void	<a href="#">translate</a> (int x, int y) Translate all of the vertices of this collision polygon.

#### Methods inherited from class java.lang.Object

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

#### Constructors

##### CollisionPolygon

public **CollisionPolygon**(java.awt.geom.Point2D.Double objectCentre,  
int verticeCount)

Create a new collision polygon.

(continued from last page)

**Parameters:**

objectCentre - Centre of the object.  
verticeCount - Number of vertices.

## Methods

### getVerticeCount

```
public int getVerticeCount()
```

Get the number of vertices in this collision polygon.

**Returns:**

Number of vertices.

### addVertice

```
public void addVertice(double x,  
                        double y)
```

Add a new vertice to this collision polygon.

**Parameters:**

x - X Location of vertice.  
y - Y Location of vertice.

### translate

```
public void translate(int x,  
                      int y)
```

Translate all of the vertices of this collision polygon.

**Parameters:**

x - Amount to translate in X axis.  
y - Amount to translate in Y axis.

### collide

```
public boolean collide(CollisionPolygon poly)
```

Determine if this collision polygon is colliding with another.

**Parameters:**

poly - Polygon to check collision with.

**Returns:**

True if they are colliding.

### calculateInterval

```
public void calculateInterval(Vector axis)
```

Calculate the interval between this polygon to the given axis.

**Parameters:**

axis - Vector representing the axis.

## intervalsSeparated

```
public boolean intervalsSeparated(float mina,
    float maxa,
    float minb,
    float maxb)
```

Method to determine if the given intervals for two vectors are seperated.

**Parameters:**

mina - Min value of the first vector.  
maxa - Max value of the first vector.  
minb - Min value of the second vector.  
maxb - Max value of the second vector.

**Returns:**

True if for any of the two vectors the min is greater than the max.

---

## separatedByAxis

```
public boolean separatedByAxis(Vector axis,
    CollisionPolygon poly)
```

Determine to see if a given axis and another collision polygon are seperated by an axis.

**Parameters:**

axis - The vector represnting the axis.  
poly - CollisonPolygon of the other object.

**Returns:**

Whether the min/max values are seperated.

---

## arena.collisions

### Class ECollisionTypes

```

java.lang.Object
  |
  +- java.lang.Enum
        +- arena.collisions.ECollisionTypes
  
```

#### All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

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

Enum to help with determining what type of collision it is. The reason for this is so changing the results of the behavior of the collisions will be easier.

### Field Summary

public static final	<a href="#">ASTEROID_COL</a>
public static final	<a href="#">BULLET_COL</a>

### Method Summary

static <a href="#">ECollisionTypes</a>	<a href="#">valueOf</a> (java.lang.String name)
static <a href="#">ECollisionTypes[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

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

#### Methods inherited from interface java.lang.Comparable

compareTo

### Fields

#### BULLET\_COL

public static final arena.collisions.ECollisionTypes **BULLET\_COL**

## ASTEROID\_COL

```
public static final arena.collisions.ECollisionTypes ASTEROID_COL
```

## Methods

### values

```
public static ECollisionTypes\[\] values()
```

---

### valueOf

```
public static ECollisionTypes valueOf(java.lang.String name)
```



## arena.collisions

### Class Vector

```
java.lang.Object
```

```
└--arena.collisions.Vector
```

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

Class representing a vector , this class is merely a way of representing parts of the collision polygon.

#### Field Summary

public	<a href="#"><u>x</u></a>
public	<a href="#"><u>y</u></a>

#### Constructor Summary

public	<a href="#"><u>Vector</u></a> (double Ix, double Iy)
--------	--

#### Method Summary

double	<a href="#"><u>dot</u></a> ( <a href="#"><u>Vector</u></a> axis) Get the dot product of this vector and an axis vector.
<a href="#"><u>Vector</u></a>	<a href="#"><u>multiply</u></a> ( <a href="#"><u>Vector</u></a> other) Multiple the vector, returning the new multiplied vector)
<a href="#"><u>Vector</u></a>	<a href="#"><u>perp</u></a> () Get the vector perpindicular to this vector.
void	<a href="#"><u>randomize</u></a> ( <a href="#"><u>Vector</u></a> max, <a href="#"><u>Vector</u></a> min) Randomize this vector between two other vectors.
<a href="#"><u>Vector</u></a>	<a href="#"><u>scale</u></a> (double scale) Scale the vector, returning the new scaled vector.

#### Methods inherited from class java.lang.Object

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

#### Fields

**x**

```
public double x
```

**y**

```
public double y
```

## Constructors

**Vector**

```
public Vector(double Ix,  
              double Iy)
```

## Methods

**scale**

```
public Vector scale(double scale)
```

Scale the vector, returning the new scaled vector.

**Parameters:**

scale - Amount to scale the vector by.

**Returns:**

New vector that is scaled.

---

**multiply**

```
public Vector multiply(Vector other)
```

Multiple the vector, returning the new multiplied vector)

**Parameters:**

other - Other vector to multiply with.

**Returns:**

Vector of the multiplied vectors.

---

**perp**

```
public Vector perp()
```

Get the vector perpendicular to this vector.

**Returns:**

Vector which is perpendicular to this vector.

---

**randomize**

```
public void randomize(Vector max,  
                     Vector min)
```

Randomize this vector between two other vectors.

---

(continued from last page)

**Parameters:**

`max` - Vector representing max.  
`min` - Vector representing min.

---

**dot**

```
public double dot(Vector axis)
```

Get the dot product of this vector and an axis vector.

**Parameters:**

`axis` - Vector representing the axis.

**Returns:**

Dot product of the two vectors.

---

**Package**  
**arena.objects**

## arena.objects

### Class AbstractObject

java.lang.Object

└--arena.objects.AbstractObject

Direct Known Subclasses:

[AbstractShip](#), [Asteroid](#), [Bullet](#)

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

Abstract class representing the core behaviour and methods that belong to any given object in the arena.

#### Constructor Summary

public	<a href="#">AbstractObject</a> ( java.awt.geom.Point2D.Double spawnLocation, double direction) Constructor that most objects will use, specifying the object type when creating
public	<a href="#">AbstractObject</a> () This constructor is only used when the object being created is to have a random spawn location and direction.

#### Method Summary

void	<a href="#">applyDamage</a> (int amount) Apply damage to the object by directly removing value from its health.
double	<a href="#">getDirection</a> () Get the current direction of this object.
abstract <a href="#">CollisionPolygon</a>	<a href="#">getObjectCollisionModel</a> () Gets the objects collision model.
int	<a href="#">getObjectHealth</a> () Method to return the current health of the given object.
java.awt.geom.Point2D. Double	<a href="#">getObjectPosition</a> () Get the objects current position.
abstract <a href="#">EObjects</a>	<a href="#">getObjectType</a> () Gets the current type of this object.
boolean	<a href="#">isObjectAlive</a> () Method to check if a given object is alive or not.
void	<a href="#">modifyDirection</a> (double amount) Modify the direction the object is facing by a given amount.
void	<a href="#">setNewObjectPosition</a> ( java.awt.geom.Point2D.Double newPos) Sets the position of the object.
abstract void	<a href="#">tickObject</a> () This method determines the objects behavior each time the game ticks.

**Methods inherited from class `java.lang.Object`**

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

## Constructors

### AbstractObject

```
public AbstractObject(java.awt.geom.Point2D.Double spawnLocation,
                      double direction)
```

Constructor that most objects will use, specifying the object type when creating

**Parameters:**

`spawnLocation` - - Location the object is spawning at.  
`direction` - - Direction the object is facing on creation.

### AbstractObject

```
public AbstractObject()
```

This constructor is only used when the object being created is to have a random spawn location and direction. This is useful for randomly placing ships at the start of each round.

## Methods

### getObjectType

```
public abstract EObjects getObjectType()
```

Gets the current type of this object. Useful for differentiating between different objects in different parts of the program.

**Returns:**

Type of object defined by `EObjects`.

### tickObject

```
public abstract void tickObject()
```

This method determines the objects behavior each time the game ticks.

### modifyDirection

```
public void modifyDirection(double amount)
```

Modify the direction the object is facing by a given amount. If the direction ends up being a negative value, normalise it to be above 0 again.

**Parameters:**

`amount` - Amount to change the direction by, positive OR negative.

### getObjectHealth

```
public int getObjectHealth()
```

Method to return the current health of the given object.

(continued from last page)

**Returns:**

Current health of the object.

---

**isObjectAlive**

```
public boolean isObjectAlive()
```

Method to check if a given object is alive or not. For an object to be alive, it needs to have at least one health remaining. If it doesn't, then it is dead.

**Returns:**

True If health is &gt;1 , otherwise false.

---

**getObjectPosition**

```
public java.awt.geom.Point2D.Double getObjectPosition()
```

Get the objects current position.

**Returns:**

Current position of object ( x , y )

---

**getDirection**

```
public double getDirection()
```

Get the current direction of this object.

**Returns:**

this.objectDirection % 360.

---

**applyDamage**

```
public void applyDamage(int amount)
```

Apply damage to the object by directly removing value from its health.

**Parameters:**

amount - Amount of damage to apply.

---

**setNewObjectPosition**

```
public void setNewObjectPosition(java.awt.geom.Point2D.Double newPos)
```

Sets the position of the object.

**Parameters:**

newPos - New position.

---

**getObjectCollisionModel**

```
public abstract CollisionPolygon getObjectCollisionModel()
```

Gets the objects collision model.

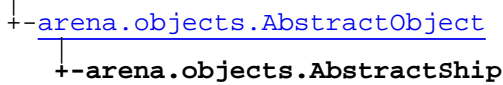
**Returns:**

CollisionPolygon representing the object.

## arena.objects

### Class AbstractShip

java.lang.Object



All Implemented Interfaces:

[IDecisionMaker](#)

Direct Known Subclasses:

[AsteroidHunter](#), [AsteroidTurret](#), [KamakazieShip](#), [NNShip](#), [PlayerHunter](#), [ShipTurret](#), [SmartHunter](#), [SmartTurret](#), [TheRock](#)

public abstract class **AbstractShip**

extends [AbstractObject](#)

implements [IDecisionMaker](#)

### Constructor Summary

public	<a href="#">AbstractShip()</a> Create a new ship.
--------	--

### Method Summary

abstract <a href="#">AbstractShip</a>	<a href="#">cloneShip()</a> Returns a deep copy of this abstract ship, useful for running one ship across many threads while maintaining scoring.
abstract void	<a href="#">determineAction()</a> Determine the action that this ship has to take this tick, setting boolean values such as fire, left, right, forward and backward to true and false figure this out.
<a href="#">AbstractShip</a>	<a href="#">getCloneOfShip()</a> Gets the ship that this ship was copied from.
<a href="#">CollisionPolygon</a>	<a href="#">getObjectCollisionModel()</a>
<a href="#">EObjects</a>	<a href="#">getObjectType()</a>
int	<a href="#">getScore()</a> Get the current score of the ship
abstract java.lang.String	<a href="#">getShipName()</a> Get the name of this ship.
<a href="#">ArenaWatcher</a>	<a href="#">getShipsArenaWatcher()</a> Get the arena watcher that this ship is to use.
void	<a href="#">incrementScore(int amount)</a> Increment the score of this ship in the given arena.



void	<a href="#">resetScore()</a> Reset the score of this ship in the given arena.
void	<a href="#">setCloneOfShip(<a href="#">AbstractShip</a> motherShip)</a> Set the ships cloneOf ship.
boolean	<a href="#">setCurrentGame(<a href="#">Arena</a> arena)</a> Attempts to set the current game this ships is competing in.
void	<a href="#">tickObject()</a> When a ship ticks, it checks all of its boolean flags and determines movement from them, that's all it does.

#### Methods inherited from class [arena.objects.AbstractObject](#)

[applyDamage](#), [getDirection](#), [getObjectCollisionModel](#), [getObjectHealth](#),  
[getObjectPosition](#), [getObjectType](#), [isObjectAlive](#), [modifyDirection](#),  
[setNewObjectPosition](#), [tickObject](#)

#### Methods inherited from class [java.lang.Object](#)

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface [arena.objects.IDecisionMaker](#)

[determineAction](#)

## Constructors

### AbstractShip

```
public AbstractShip()
```

Create a new ship.

## Methods

### cloneShip

```
public abstract AbstractShip cloneShip()
```

Returns a deep copy of this abstract ship, useful for running one ship across many threads while maintaining scoring.

**Returns:**

AbstractShip copy of the current ship.

### getShipName

```
public abstract java.lang.String getShipName()
```

Get the name of this ship.

**Returns:**

Name of the ship.

(continued from last page)

---

## getObjectType

```
public EObject getObjectType()
```

Gets the current type of this object. Useful for differentiating between different objects in different parts of the program.

---

## tickObject

```
public void tickObject()
```

When a ship ticks, it checks all of its boolean flags and determines movement from them, that's all it does.

---

## determineAction

```
public abstract void determineAction()
```

Determine the action that this ship has to take this tick, setting boolean values such as fire, left, right, forward and backward to true and false figure this out.

---

## setCurrentGame

```
public boolean setCurrentGame(Arena arena)
```

Attempts to set the current game this ships is competing in.

**Parameters:**

arena - Arena in which the ship is competing.

**Returns:**

True if set correctly, false if otherwise.

---

## getCloneOfShip

```
public AbstractShip getCloneOfShip()
```

Gets the ship that this ship was copied from. This is used for when scoring has to be calculated when I changed to threading the arenas.

**Returns:**

---

## getObjectCollisionModel

```
public CollisionPolygon getObjectCollisionModel()
```

Gets the objects collision model.

---

## getScore

```
public int getScore()
```

Get the current score of the ship

**Returns:**

Score of ship in arena.

---

(continued from last page)

## incrementScore

```
public void incrementScore(int amount)
```

Increment the score of this ship in the given arena. It is synchronised since incrementing the score will happen across multiple threads. If the ship has a "cloneOfShip". Then it will increment that ships score instead. This is to track scores across multiple games easily.

**Parameters:**

amount - Amount to increase score by.

---

## resetScore

```
public void resetScore()
```

Reset the score of this ship in the given arena. If the ship has a "cloneOfShip". Then it will reset that ships score instead. This is to track scores across multiple games easily.

---

## setCloneOfShip

```
public void setCloneOfShip(AbstractShip motherShip)
```

Set the ships cloneOf ship.

**Parameters:**

motherShip - Ship to set as clone of ship.

---

## getShipsArenaWatcher

```
public ArenaWatcher getShipsArenaWatcher()
```

Get the arena watcher that this ship is to use. Could foresee some errors if this was called in a ships constructor.

**Returns:**

arenawatcher to get information from.

---

## arena.objects

### Class AsteroidFactory

```
java.lang.Object
  |
  +--arena.objects.AsteroidFactory
```

All Implemented Interfaces:

[IObjectFactory](#)

```
public class AsteroidFactory
extends java.lang.Object
implements IObjectFactory
```

This class is used to generate random asteroids that move onto the game arena when they are spawned. In order to do this, they are spawned off-screen with an on-screen target, they move towards and by this. Ships know where asteroids are when they spawn, but until they spawn, ships have no idea where they are.

#### Constructor Summary

public	<a href="#">AsteroidFactory()</a>
--------	-----------------------------------

#### Method Summary

<a href="#">AbstractObject</a>	<a href="#">produceObject()</a>
--------------------------------	---------------------------------

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface [arena.objects.IObjectFactory](#)

[produceObject](#)

#### Constructors

##### AsteroidFactory

```
public AsteroidFactory()
```

#### Methods

##### produceObject

```
public AbstractObject produceObject()
```

## arena.objects

### Class EObjects

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- arena.objects.EObjects
  
```

#### All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

```

public final class EObjects
extends java.lang.Enum
  
```

Enum to represent the basic objects and hold some of the default values for them, such as the default health for any given object. Health isn't modifiable inside the game or simulation, so storing it along with the type of object seemed like a sensible option.

### Field Summary

public static final	<a href="#">OBJ_ASTEROID</a>
public static final	<a href="#">OBJ_BULLET</a>
public static final	<a href="#">OBJ_SHIP</a>

### Method Summary

int	<a href="#">getBaseHealth()</a> Get the base health of the given object.
static <a href="#">EObjects</a>	<a href="#">valueOf(java.lang.String name)</a>
static <a href="#">EObjects[]</a>	<a href="#">values()</a>

#### Methods inherited from class java.lang.Enum

compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

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

#### Methods inherited from interface java.lang.Comparable

compareTo

### Fields

## OBJ\_SHIP

```
public static final arena.objects.EObjects OBJ_SHIP
```

---

## OBJ\_ASTEROID

```
public static final arena.objects.EObjects OBJ_ASTEROID
```

---

## OBJ\_BULLET

```
public static final arena.objects.EObjects OBJ_BULLET
```

---

## Methods

### values

```
public static EObjects\[\] values()
```

---

### valueOf

```
public static EObjects valueOf(java.lang.String name)
```

---

### getBaseHealth

```
public int getBaseHealth()
```

Get the base health of the given object.

**Returns:**

Health as given in the enum.

## arena.objects

# Interface IDecisionMaker

All Known Implementing Classes:

[AbstractShip](#)

```
public interface IDecisionMaker
extends
```

Interface for an object that can make a decision.

## Method Summary

void	<a href="#">determineAction()</a>
------	-----------------------------------

Determine the action to be made by this object.

## Methods

### determineAction

```
public void determineAction()
```

Determine the action to be made by this object.

## arena.objects

# Interface IObjectFactory

All Known Implementing Classes:

[AsteroidFactory](#)

public interface **IObjectFactory**  
extends

## Method Summary

[AbstractObject](#)

[produceObject](#)( )

Produce the object that the factory requires.

## Methods

### produceObject

public [AbstractObject](#) **produceObject**( )

Produce the object that the factory requires.

**Returns:**

Object created.



---

**Package**

**arena.objects.objects**

## arena.objects.objects

### Class Asteroid

```

java.lang.Object
  |
+- arena.objects.AbstractObject
    |
    +- arena.objects.objects.Asteroid
  
```

public class **Asteroid**  
 extends [AbstractObject](#)

#### Field Summary

public final	<a href="#">ASTEROID_SIZE</a> Value: <b>0.08</b>
--------------	---

#### Constructor Summary

public	<a href="#">Asteroid</a> (java.awt.geom.Point2D.Double spawnLocation, double direction) Creates a new asteroid with the following parameters.
--------	--

#### Method Summary

int	<a href="#">getAsteroidComplexity</a> () Gets the complexity of this asteroid, this is mainly used for the drawing code to be able to draw the correct shape of each asteroid later on.
<a href="#">CollisionPolygon</a>	<a href="#">getObjectCollisionModel</a> ()
<a href="#">EObjects</a>	<a href="#">getObjectType</a> ()
double	<a href="#">getTheta</a> () Retrieve the theta value for this asteroid, which is the angle in between every point being drawn from the centre point.
void	<a href="#">tickObject</a> ()

#### Methods inherited from class [arena.objects.AbstractObject](#)

[applyDamage](#), [getDirection](#), [getObjectCollisionModel](#), [getObjectHealth](#),  
[getObjectPosition](#), [getObjectType](#), [isObjectAlive](#), [modifyDirection](#),  
[setNewObjectPosition](#), [tickObject](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Fields

(continued from last page)

## ASTEROID\_SIZE

```
public final double ASTEROID_SIZE
```

Constant value: **0.08**

## Constructors

### Asteroid

```
public Asteroid(java.awt.geom.Point2D.Double spawnLocation,  
                double direction)
```

Creates a new asteroid with the following parameters.

**Parameters:**

spawnLocation - Location the asteroid is spawning.  
direction - Direction the asteroid is facing.

## Methods

### getObjectType

```
public EObject getObjectType()
```

Gets the current type of this object. Useful for differentiating between different objects in different parts of the program.

### tickObject

```
public void tickObject()
```

This method determines the objects behavior each time the game ticks.

### getObjectCollisionModel

```
public CollisionPolygon getObjectCollisionModel()
```

Gets the objects collision model.

### getAsteroidComplexity

```
public int getAsteroidComplexity()
```

Gets the complexity of this asteroid, this is mainly used for the drawing code to be able to draw the correct shape of each asteroid later on.

**Returns:**

The complexity of the asteroid.

### getTheta

```
public double getTheta()
```

Retrieve the theta value for this asteroid, which is the angle in between every point being drawn from the centre point.

**Returns:**

Theta, being 360 / asteroidComplexity

## arena.objects.objects

### Class Bullet

```
java.lang.Object
  |
  +-- arena.objects.AbstractObject
        |
        +-- arena.objects.objects.Bullet
```

public class **Bullet**  
 extends [AbstractObject](#)

#### Constructor Summary

public	<a href="#">Bullet</a> ( <a href="#">AbstractObject</a> bulletCreator, java.awt.geom.Point2D.Double spawnLocation, double bulletDirection) Creates a new bullet with the given parameters.
--------	---

#### Method Summary

<a href="#">AbstractObject</a>	<a href="#">getBulletCreator</a> () Gets the object that created this bullet.
<a href="#">CollisionPolygon</a>	<a href="#">getObjectCollisionModel</a> ()
<a href="#">EObjects</a>	<a href="#">getObjectType</a> ()
void	<a href="#">tickObject</a> ()

#### Methods inherited from class [arena.objects.AbstractObject](#)

[applyDamage](#), [getDirection](#), [getObjectCollisionModel](#), [getObjectHealth](#),  
[getObjectPosition](#), [getObjectType](#), [isObjectAlive](#), [modifyDirection](#),  
[setNewObjectPosition](#), [tickObject](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructors

### Bullet

```
public Bullet(AbstractObject bulletCreator,  
              java.awt.geom.Point2D.Double spawnLocation,  
              double bulletDirection)
```

Creates a new bullet with the given parameters.

#### Parameters:

bulletCreator - Object that created this bullet.

(continued from last page)

`spawnLocation` - Location the bullet is spawning at.

`bulletDirection` - Direction the bullet is facing.

## Methods

### **getObjectType**

```
public EObject getObjectType()
```

Gets the current type of this object. Useful for differentiating between different objects in different parts of the program.

---

### **tickObject**

```
public void tickObject()
```

This method determines the objects behavior each time the game ticks.

---

### **getObjectCollisionModel**

```
public CollisionPolygon getObjectCollisionModel()
```

Gets the objects collision model.

---

### **getBulletCreator**

```
public AbstractObject getBulletCreator()
```

Gets the object that created this bullet.

**Returns:**

Object that created this bullet.

---

**Package**

**arena.objects.ships**

## arena.objects.ships

### Class AsteroidHunter

```

java.lang.Object
  +- arena.objects.AbstractObject
    +- arena.objects.AbstractShip
      +- arena.objects.ships.AsteroidHunter
  
```

All Implemented Interfaces:

[IDecisionMaker](#)

public class **AsteroidHunter**  
 extends [AbstractShip](#)

#### Constructor Summary

public	<a href="#">AsteroidHunter()</a>
--------	----------------------------------

#### Method Summary

<a href="#">AsteroidHunter</a>	<a href="#">cloneShip()</a>
void	<a href="#">determineAction()</a>
java.lang.String	<a href="#">getShipName()</a>

#### Methods inherited from class [arena.objects.AbstractShip](#)

[cloneShip](#), [determineAction](#), [getCloneOfShip](#), [getObjectCollisionModel](#), [getObjectType](#), [getScore](#), [getShipName](#), [getShipsArenaWatcher](#), [incrementScore](#), [resetScore](#), [setCloneOfShip](#), [setCurrentGame](#), [tickObject](#)

#### Methods inherited from class [arena.objects.AbstractObject](#)

[applyDamage](#), [getDirection](#), [getObjectCollisionModel](#), [getObjectHealth](#), [getObjectPosition](#), [getObjectType](#), [isObjectAlive](#), [modifyDirection](#), [setNewObjectPosition](#), [tickObject](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface [arena.objects.IDecisionMaker](#)

[determineAction](#)

#### Constructors

(continued from last page)

## AsteroidHunter

```
public AsteroidHunter()
```

## Methods

### determineAction

```
public void determineAction()
```

Determine the action that this ship has to take this tick, setting boolean values such as fire, left, right, forward and backward to true and false figure this out.

---

### cloneShip

```
public AsteroidHunter cloneShip()
```

Returns a deep copy of this abstract ship, useful for running one ship across many threads while maintaining scoring.

---

### getShipName

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

Get the name of this ship.



## arena.objects.ships

### Class AsteroidTurret

```

java.lang.Object
  +- arena.objects.AbstractObject
    +- arena.objects.AbstractShip
      +- arena.objects.ships.AsteroidTurret
  
```

All Implemented Interfaces:

[IDecisionMaker](#)

```

public class AsteroidTurret
extends AbstractShip
  
```

#### Constructor Summary

public	<a href="#">AsteroidTurret()</a>
--------	----------------------------------

#### Method Summary

<a href="#">AsteroidTurret</a>	<a href="#">cloneShip()</a>
void	<a href="#">determineAction()</a>
java.lang.String	<a href="#">getShipName()</a>

#### Methods inherited from class [arena.objects.AbstractShip](#)

[cloneShip](#), [determineAction](#), [getCloneOfShip](#), [getObjectCollisionModel](#), [getObjectType](#), [getScore](#), [getShipName](#), [getShipsArenaWatcher](#), [incrementScore](#), [resetScore](#), [setCloneOfShip](#), [setCurrentGame](#), [tickObject](#)

#### Methods inherited from class [arena.objects.AbstractObject](#)

[applyDamage](#), [getDirection](#), [getObjectCollisionModel](#), [getObjectHealth](#), [getObjectPosition](#), [getObjectType](#), [isObjectAlive](#), [modifyDirection](#), [setNewObjectPosition](#), [tickObject](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface [arena.objects.IDecisionMaker](#)

[determineAction](#)

#### Constructors

(continued from last page)

## AsteroidTurret

```
public AsteroidTurret()
```

## Methods

### determineAction

```
public void determineAction()
```

Determine the action that this ship has to take this tick, setting boolean values such as fire, left, right, forward and backward to true and false figure this out.

---

### cloneShip

```
public AsteroidTurret cloneShip()
```

Returns a deep copy of this abstract ship, useful for running one ship across many threads while maintaining scoring.

---

### getShipName

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

Get the name of this ship.

## arena.objects.ships

### Class KamakazieShip

```

java.lang.Object
  |
  +- arena.objects.AbstractObject
      |
      +- arena.objects.AbstractShip
          |
          +- arena.objects.ships.KamakazieShip
  
```

All Implemented Interfaces:

[IDecisionMaker](#)

public class **KamakazieShip**  
 extends [AbstractShip](#)

#### Constructor Summary

public	<a href="#">KamakazieShip()</a>
--------	---------------------------------

#### Method Summary

<a href="#">KamakazieShip</a>	<a href="#">cloneShip()</a>
void	<a href="#">determineAction()</a>
java.lang.String	<a href="#">getShipName()</a>

#### Methods inherited from class [arena.objects.AbstractShip](#)

[cloneShip](#), [determineAction](#), [getCloneOfShip](#), [getObjectCollisionModel](#), [getObjectType](#), [getScore](#), [getShipName](#), [getShipsArenaWatcher](#), [incrementScore](#), [resetScore](#), [setCloneOfShip](#), [setCurrentGame](#), [tickObject](#)

#### Methods inherited from class [arena.objects.AbstractObject](#)

[applyDamage](#), [getDirection](#), [getObjectCollisionModel](#), [getObjectHealth](#), [getObjectPosition](#), [getObjectType](#), [isObjectAlive](#), [modifyDirection](#), [setNewObjectPosition](#), [tickObject](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface [arena.objects.IDecisionMaker](#)

[determineAction](#)

#### Constructors

(continued from last page)

## KamakazieShip

```
public KamakazieShip()
```

## Methods

### determineAction

```
public void determineAction()
```

Determine the action that this ship has to take this tick, setting boolean values such as fire, left, right, forward and backward to true and false figure this out.

---

### cloneShip

```
public KamakazieShip cloneShip()
```

Returns a deep copy of this abstract ship, useful for running one ship across many threads while maintaining scoring.

---

### getShipName

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

Get the name of this ship.

## arena.objects.ships

### Class NNShip

```

java.lang.Object
  +- arena.objects.AbstractObject
    +- arena.objects.AbstractShip
      +- arena.objects.ships.NNShip
  
```

All Implemented Interfaces:

[IDecisionMaker](#)

```

public class NNShip
extends AbstractShip
  
```

### Constructor Summary

public	<a href="#">NNShip</a> (int numberOfHiddenLayers, int neuronsPerHiddenLayer, <a href="#">Chromosome</a> chromosome) Create a new Neural Network ship.
--------	--

### Method Summary

<a href="#">NNShip</a>	<a href="#">cloneShip()</a>
void	<a href="#">determineAction()</a>
java.lang.String	<a href="#">getShipName()</a>

#### Methods inherited from class [arena.objects.AbstractShip](#)

[cloneShip](#), [determineAction](#), [getCloneOfShip](#), [getObjectCollisionModel](#), [getObjectType](#), [getScore](#), [getShipName](#), [getShipsArenaWatcher](#), [incrementScore](#), [resetScore](#), [setCloneOfShip](#), [setCurrentGame](#), [tickObject](#)

#### Methods inherited from class [arena.objects.AbstractObject](#)

[applyDamage](#), [getDirection](#), [getObjectCollisionModel](#), [getObjectHealth](#), [getObjectPosition](#), [getObjectType](#), [isObjectAlive](#), [modifyDirection](#), [setNewObjectPosition](#), [tickObject](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface [arena.objects.IDecisionMaker](#)

[determineAction](#)

(continued from last page)

## Constructors

### NNShip

```
public NNShip(int numberOfHiddenLayers,  
               int neuronsPerHiddenLayer,  
               Chromosome chromosome)
```

Create a new Neural Network ship. It creates the neural network based on the final fields above for the time being, if this project ended up having an extensive UI, then this ship would receive the values for the NNetwork as additional parameters.

**Parameters:**

`chromosome` - - The chromosome this NNShip is to base its actions upon.

## Methods

### determineAction

```
public void determineAction()
```

Determine the action that this ship has to take this tick, setting boolean values such as fire, left, right, forward and backward to true and false figure this out.

---

### cloneShip

```
public NNShip cloneShip()
```

Returns a deep copy of this abstract ship, useful for running one ship across many threads while maintaining scoring.

---

### getShipName

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

Get the name of this ship.

## arena.objects.ships

### Class PlayerHunter

```

java.lang.Object
  +- arena.objects.AbstractObject
    +- arena.objects.AbstractShip
      +- arena.objects.ships.PlayerHunter
  
```

All Implemented Interfaces:

[IDecisionMaker](#)

```

public class PlayerHunter
extends AbstractShip
  
```

#### Constructor Summary

public	<a href="#">PlayerHunter()</a>
--------	--------------------------------

#### Method Summary

<a href="#">AbstractShip</a>	<a href="#">cloneShip()</a>
void	<a href="#">determineAction()</a>
java.lang.String	<a href="#">getShipName()</a>

#### Methods inherited from class [arena.objects.AbstractShip](#)

[cloneShip](#), [determineAction](#), [getCloneOfShip](#), [getObjectCollisionModel](#), [getObjectType](#), [getScore](#), [getShipName](#), [getShipsArenaWatcher](#), [incrementScore](#), [resetScore](#), [setCloneOfShip](#), [setCurrentGame](#), [tickObject](#)

#### Methods inherited from class [arena.objects.AbstractObject](#)

[applyDamage](#), [getDirection](#), [getObjectCollisionModel](#), [getObjectHealth](#), [getObjectPosition](#), [getObjectType](#), [isObjectAlive](#), [modifyDirection](#), [setNewObjectPosition](#), [tickObject](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface [arena.objects.IDecisionMaker](#)

[determineAction](#)

#### Constructors

(continued from last page)

## PlayerHunter

```
public PlayerHunter()
```

## Methods

### cloneShip

```
public AbstractShip cloneShip()
```

Returns a deep copy of this abstract ship, useful for running one ship across many threads while maintaining scoring.

---

### getShipName

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

Get the name of this ship.

---

### determineAction

```
public void determineAction()
```

Determine the action that this ship has to take this tick, setting boolean values such as fire, left, right, forward and backward to true and false figure this out.



## arena.objects.ships

### Class ShipTurret

```

java.lang.Object
  +- arena.objects.AbstractObject
    +- arena.objects.AbstractShip
      +- arena.objects.ships.ShipTurret
  
```

All Implemented Interfaces:

[IDecisionMaker](#)

```

public class ShipTurret
extends AbstractShip
  
```

#### Constructor Summary

public	<a href="#">ShipTurret()</a>
--------	------------------------------

#### Method Summary

<a href="#">ShipTurret</a>	<a href="#">cloneShip()</a>
void	<a href="#">determineAction()</a>
java.lang.String	<a href="#">getShipName()</a>

#### Methods inherited from class [arena.objects.AbstractShip](#)

[cloneShip](#), [determineAction](#), [getCloneOfShip](#), [getObjectCollisionModel](#), [getObjectType](#), [getScore](#), [getShipName](#), [getShipsArenaWatcher](#), [incrementScore](#), [resetScore](#), [setCloneOfShip](#), [setCurrentGame](#), [tickObject](#)

#### Methods inherited from class [arena.objects.AbstractObject](#)

[applyDamage](#), [getDirection](#), [getObjectCollisionModel](#), [getObjectHealth](#), [getObjectPosition](#), [getObjectType](#), [isObjectAlive](#), [modifyDirection](#), [setNewObjectPosition](#), [tickObject](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface [arena.objects.IDecisionMaker](#)

[determineAction](#)

#### Constructors

(continued from last page)

## ShipTurret

```
public ShipTurret()
```

## Methods

### determineAction

```
public void determineAction()
```

Determine the action that this ship has to take this tick, setting boolean values such as fire, left, right, forward and backward to true and false figure this out.

---

### cloneShip

```
public ShipTurret cloneShip()
```

Returns a deep copy of this abstract ship, useful for running one ship across many threads while maintaining scoring.

---

### getShipName

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

Get the name of this ship.

## arena.objects.ships

### Class SmartHunter

```

java.lang.Object
  +- arena.objects.AbstractObject
    +- arena.objects.AbstractShip
      +- arena.objects.ships.SmartHunter
  
```

All Implemented Interfaces:

[IDecisionMaker](#)

public class **SmartHunter**  
 extends [AbstractShip](#)

### Constructor Summary

public	<a href="#">SmartHunter()</a>
--------	-------------------------------

### Method Summary

<a href="#">SmartHunter</a>	<a href="#">cloneShip()</a>
void	<a href="#">determineAction()</a>
java.lang.String	<a href="#">getShipName()</a>

#### Methods inherited from class [arena.objects.AbstractShip](#)

[cloneShip](#), [determineAction](#), [getCloneOfShip](#), [getObjectCollisionModel](#), [getObjectType](#), [getScore](#), [getShipName](#), [getShipsArenaWatcher](#), [incrementScore](#), [resetScore](#), [setCloneOfShip](#), [setCurrentGame](#), [tickObject](#)

#### Methods inherited from class [arena.objects.AbstractObject](#)

[applyDamage](#), [getDirection](#), [getObjectCollisionModel](#), [getObjectHealth](#), [getObjectPosition](#), [getObjectType](#), [isObjectAlive](#), [modifyDirection](#), [setNewObjectPosition](#), [tickObject](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface [arena.objects.IDecisionMaker](#)

[determineAction](#)

### Constructors

(continued from last page)

## SmartHunter

```
public SmartHunter()
```

## Methods

### determineAction

```
public void determineAction()
```

Determine the action that this ship has to take this tick, setting boolean values such as fire, left, right, forward and backward to true and false figure this out.

---

### cloneShip

```
public SmartHunter cloneShip()
```

Returns a deep copy of this abstract ship, useful for running one ship across many threads while maintaining scoring.

---

### getShipName

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

Get the name of this ship.

## arena.objects.ships

### Class SmartTurret

```

java.lang.Object
  |
  +- arena.objects.AbstractObject
      |
      +- arena.objects.AbstractShip
          |
          +- arena.objects.ships.SmartTurret
  
```

All Implemented Interfaces:

[IDecisionMaker](#)

```

public class SmartTurret
extends AbstractShip
  
```

#### Constructor Summary

public	<a href="#">SmartTurret()</a>
--------	-------------------------------

#### Method Summary

<a href="#">SmartTurret</a>	<a href="#">cloneShip()</a>
void	<a href="#">determineAction()</a>
java.lang.String	<a href="#">getShipName()</a>

#### Methods inherited from class [arena.objects.AbstractShip](#)

[cloneShip](#), [determineAction](#), [getCloneOfShip](#), [getObjectCollisionModel](#), [getObjectType](#), [getScore](#), [getShipName](#), [getShipsArenaWatcher](#), [incrementScore](#), [resetScore](#), [setCloneOfShip](#), [setCurrentGame](#), [tickObject](#)

#### Methods inherited from class [arena.objects.AbstractObject](#)

[applyDamage](#), [getDirection](#), [getObjectCollisionModel](#), [getObjectHealth](#), [getObjectPosition](#), [getObjectType](#), [isObjectAlive](#), [modifyDirection](#), [setNewObjectPosition](#), [tickObject](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface [arena.objects.IDecisionMaker](#)

[determineAction](#)

#### Constructors

(continued from last page)

## SmartTurret

```
public SmartTurret()
```

## Methods

### determineAction

```
public void determineAction()
```

Determine the action that this ship has to take this tick, setting boolean values such as fire, left, right, forward and backward to true and false figure this out.

---

### cloneShip

```
public SmartTurret cloneShip()
```

Returns a deep copy of this abstract ship, useful for running one ship across many threads while maintaining scoring.

---

### getShipName

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

Get the name of this ship.

## arena.objects.ships

### Class TheRock

```

java.lang.Object
  |
  +-- arena.objects.AbstractObject
        |
        +-- arena.objects.AbstractShip
              |
              +-- arena.objects.ships.TheRock
  
```

All Implemented Interfaces:

[IDecisionMaker](#)

```

public class TheRock
extends AbstractShip
  
```

### Constructor Summary

public	<a href="#">TheRock()</a>
--------	---------------------------

### Method Summary

<a href="#">AbstractShip</a>	<a href="#">cloneShip()</a>
void	<a href="#">determineAction()</a>
java.lang.String	<a href="#">getShipName()</a>

#### Methods inherited from class [arena.objects.AbstractShip](#)

[cloneShip](#), [determineAction](#), [getCloneOfShip](#), [getObjectCollisionModel](#), [getObjectType](#), [getScore](#), [getShipName](#), [getShipsArenaWatcher](#), [incrementScore](#), [resetScore](#), [setCloneOfShip](#), [setCurrentGame](#), [tickObject](#)

#### Methods inherited from class [arena.objects.AbstractObject](#)

[applyDamage](#), [getDirection](#), [getObjectCollisionModel](#), [getObjectHealth](#), [getObjectPosition](#), [getObjectType](#), [isObjectAlive](#), [modifyDirection](#), [setNewObjectPosition](#), [tickObject](#)

#### Methods inherited from class java.lang.Object

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

#### Methods inherited from interface [arena.objects.IDecisionMaker](#)

[determineAction](#)

### Constructors

(continued from last page)

## TheRock

```
public TheRock()
```

## Methods

### cloneShip

```
public AbstractShip cloneShip()
```

Returns a deep copy of this abstract ship, useful for running one ship across many threads while maintaining scoring.

---

### getShipName

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

Get the name of this ship.

---

### determineAction

```
public void determineAction()
```

Determine the action that this ship has to take this tick, setting boolean values such as fire, left, right, forward and backward to true and false figure this out.



---

# Package driver

## driver

### Class BatchDriver

java.lang.Object  
└─driver.BatchDriver

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

#### Constructor Summary

public	<a href="#">BatchDriver()</a>
--------	-------------------------------

#### Method Summary

static void	<a href="#">main</a> (java.lang.String[] args)
-------------	--

Methods inherited from class java.lang.Object

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

#### Constructors

##### BatchDriver

public **BatchDriver**()

#### Methods

##### main

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

## driver

### Class GADriver

```
java.lang.Object
|
+--driver.GADriver
```

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

Class used to easily run a genetic algorithm with a list of pre-set values.

### Constructor Summary

public	<a href="#">GADriver</a> ( <a href="#">AbstractShip</a> firstShip, <a href="#">AbstractShip</a> secondShip, <a href="#">AbstractShip</a> thirdShip)
Creates a new driver that will run with a set of ships and pre-defined values.	

### Method Summary

static void	<a href="#">main</a> (java.lang.String[] args)
-------------	--

### Methods inherited from class java.lang.Object

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

## Constructors

### GADriver

```
public GADriver(AbstractShip firstShip,
               AbstractShip secondShip,
               AbstractShip thirdShip)
```

Creates a new driver that will run with a set of ships and pre-defined values.

#### Parameters:

firstShip - First ship to compete with the NNShip  
 secondShip - Second ship to compete with the NNShip  
 thirdShip - Third ship to compete with the NNShip

## Methods

### main

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

## driver

### Class NNShipDemo

java.lang.Object  
└─driver.NNShipDemo

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

#### Constructor Summary

public	<a href="#">NNShipDemo()</a>
--------	------------------------------

#### Method Summary

static void	<a href="#">main</a> (java.lang.String[] args)
-------------	--

Methods inherited from class java.lang.Object

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

#### Constructors

##### NNShipDemo

public **NNShipDemo**()

#### Methods

##### main

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

## driver

# Class TrainedNNShipDemo

java.lang.Object

└-driver.TrainedNNShipDemo

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

## Constructor Summary

public	<a href="#">TrainedNNShipDemo()</a>
--------	-------------------------------------

## Method Summary

static void	<a href="#">main</a> (java.lang.String[] args)
-------------	--

Methods inherited from class java.lang.Object

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

## Constructors

### TrainedNNShipDemo

```
public TrainedNNShipDemo()
```

## Methods

### main

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

## driver

### Class UIDemo

java.lang.Object  
└─driver.UIDemo

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

#### Constructor Summary

public	<a href="#">UIDemo()</a>
--------	--------------------------

#### Method Summary

static void	<a href="#">main</a> (java.lang.String[] args)
-------------	--

Methods inherited from class java.lang.Object

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

#### Constructors

##### UIDemo

```
public UIDemo()
```

#### Methods

##### main

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

---

# Package genetic

## genetic Class Chromosome

```
java.lang.Object
|
+--genetic.Chromosome
```

All Implemented Interfaces:  
[ISaveableChromosome](#)

```
public class Chromosome
extends java.lang.Object
implements ISaveableChromosome
```

Class that represents the weights that any given neural network can work with.

### Constructor Summary

public	<a href="#">Chromosome</a> (double[] weights) Creates a new chromosome with a given set of pre-defined weights.
public	<a href="#">Chromosome</a> (int numberWeights) Generate a new chromosome with random weights, truely random creation of a new chromosome.

### Method Summary

int	<a href="#">getChromosomeScore</a> () Get the score this chromosome achieved.
double[]	<a href="#">getWeights</a> () Get the weights that this chromosome represents.
void	<a href="#">saveToFile</a> (java.lang.String folderName)
void	<a href="#">setChromosomeScore</a> (int score) Set the score of this given chromosome.

#### Methods inherited from class java.lang.Object

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

#### Methods inherited from interface [io.ISaveableChromosome](#)

[saveToFile](#)

## Constructors

### Chromosome

```
public Chromosome(double[] weights)
```



(continued from last page)

Creates a new chromosome with a given set of pre-defined weights.

**Parameters:**  
weights

---

## Chromosome

```
public Chromosome(int numberWeights)
```

Generate a new chromosome with random weights, truly random creation of a new chromosome.

**Parameters:**  
numberWeights - Number of weights the chromosome is to have.

## Methods

### getWeights

```
public double[] getWeights()
```

Get the weights that this chromosome represents.

**Returns:**  
Weights array.

---

### setChromosomeScore

```
public void setChromosomeScore(int score)
```

Set the score of this given chromosome.

**Parameters:**  
score - Value to set the score.

---

### getChromosomeScore

```
public int getChromosomeScore()
```

Get the score this chromosome achieved. REQUIRES setChromosoScore to be called prior or will be 0.

**Returns:**  
Score of this chromosome.

---

### saveToFile

```
public void saveToFile(java.lang.String folderName)  
    throws java.io.IOException
```

## genetic Class GA

```
java.lang.Object
|
+-genetic.GA
```

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

This class is an adapted version of the GA I used for a snake AI project in 3rd year.

### Field Summary

public static	<a href="#">CROSSOVER_CHANCE</a>
public static	<a href="#">MUTATION_CHANCE</a>

### Constructor Summary

public	<a href="#">GA</a> (java.util.ArrayList otherShips, int populationSize, int generations, int gamesPerGeneration, int maxAsteroids, int asteroidSpawnChance) Creates a new genetic algorithm that will run with the given parameters.
--------	---

### Methods inherited from class java.lang.Object

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

## Fields

### CROSSOVER\_CHANCE

```
public static double CROSSOVER_CHANCE
```

### MUTATION\_CHANCE

```
public static double MUTATION_CHANCE
```

## Constructors

### GA

```
public GA(java.util.ArrayList otherShips,
          int populationSize,
          int generations,
          int gamesPerGeneration,
          int maxAsteroids,
          int asteroidSpawnChance)
```

(continued from last page)

Creates a new genetic algorithm that will run with the given parameters. At the current time, it is only possible to run one NNShip with a set of given ships, although with further expansion, this GA would be able to run with any number of NNShips against any number of enemy ships. MUST have other ships and values for populationSize, generations and gamesPerGeneration must be >1.

**Parameters:**

- otherShips - Ships that the NNShips will compete with.
- populationSize - How many chromosomes will be in each generation.
- generations - How many generations will be ran by the GA.
- gamesPerGeneration - How many games are played per generation.
- maxAsteroids - Maximum number of asteroids in each of the arenas created by the GA.
- asteroidSpawnChance - % chance to spawn an asteroid in each arena per tick. ( 1-100 ).

## genetic Class GAHelper

```
java.lang.Object
  |
  +--genetic.GAHelper
```

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

### Constructor Summary

public	<a href="#">GAHelper()</a>
--------	----------------------------

### Method Summary

static <a href="#">Chromosome</a>	<a href="#">breedChromosomes</a> ( <a href="#">Chromosome</a> A, <a href="#">Chromosome</a> B)
static void	<a href="#">mutateChromosome</a> ( <a href="#">Chromosome</a> chromo, double mutationChance)

#### Methods inherited from class java.lang.Object

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

## Constructors

### GAHelper

```
public GAHelper()
```

## Methods

### breedChromosomes

```
public static Chromosome breedChromosomes(Chromosome A,
Chromosome B)
```

### mutateChromosome

```
public static void mutateChromosome(Chromosome chromo,
double mutationChance)
```

## genetic Interface ISelectionProcess

All Known Implementing Classes:

[RouletteSection](#)

---

public interface **ISelectionProcess**  
extends

---

### Method Summary

java.util.ArrayList	<a href="#">generateNewPopulation</a> (java.util.ArrayList currentPopulation) Calculate the next population given an input population.
---------------------	---

---

### Methods

#### generateNewPopulation

public java.util.ArrayList **generateNewPopulation**(java.util.ArrayList currentPopulation)

Calculate the next population given an input population.

**Parameters:**

inputPopulation - Population to choose next populous from.

**Returns:**

List of chromosomes that make up the next population.

---

**Package**

**genetic.selectionAlgorithms**

# genetic.selectionAlgorithms

## Class RouletteSection

java.lang.Object

└─genetic.selectionAlgorithms.RouletteSection

All Implemented Interfaces:

[ISelectionProcess](#)

public class **RouletteSection**  
 extends java.lang.Object  
 implements [ISelectionProcess](#)

### Constructor Summary

public	<a href="#">RouletteSection()</a>
--------	-----------------------------------

### Method Summary

java.util.ArrayList	<a href="#">generateNewPopulation</a> (java.util.ArrayList inputPopulation)
---------------------	---

Methods inherited from class java.lang.Object

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

Methods inherited from interface [genetic.ISelectionProcess](#)

[generateNewPopulation](#)

### Constructors

#### RouletteSection

public **RouletteSection()**

### Methods

#### generateNewPopulation

public java.util.ArrayList **generateNewPopulation**(java.util.ArrayList inputPopulation)

---

# Package io



## io Class GeneticIO

```
java.lang.Object
  |
  +--io.GeneticIO
```

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

### Constructor Summary

public	<a href="#">GeneticIO()</a>
--------	-----------------------------

### Method Summary

void	<a href="#">addToFitnessFile</a> (int generationNumber, int gamesPlayed, java.util.ArrayList chromosomes)
void	<a href="#">closeFitnessWriter</a> ()
boolean	<a href="#">createChromosomeFolder</a> () Creates the folder that all chromosomes will be saved in.
void	<a href="#">createFitnessFile</a> ()
void	<a href="#">saveChromosomeToFolder</a> ( <a href="#">Chromosome</a> chromoToSave) Saves a given chromosome to the folder that this GeneticIO represents.
void	<a href="#">writeParametersToFile</a> (java.util.ArrayList ships, int popSize, int generationCount, int gamesPerGeneration, int maxAsteroids, int asteroidSpawnChance, double mutationChance, double crossoverChance) Write the paramaters of this GA to a file in the folder, so that you can easily check the variables later on.

#### Methods inherited from class java.lang.Object

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

### Constructors

#### GeneticIO

```
public GeneticIO()
```

### Methods

(continued from last page)

---

## createChromosomeFolder

```
public boolean createChromosomeFolder()
```

Creates the folder that all chromosomes will be saved in.

**Returns:**

True if folder created successfully.

---

## saveChromosomeToFolder

```
public void saveChromosomeToFolder(Chromosome chromoToSave)
```

Saves a given chromosome to the folder that this GeneticIO represents.

**Parameters:**

chromoToSave - Chromosome to save.

---

## writeParametersToFile

```
public void writeParametersToFile(java.util.ArrayList ships,  
    int popSize,  
    int generationCount,  
    int gamesPerGeneration,  
    int maxAsteroids,  
    int asteroidSpawnChance,  
    double mutationChance,  
    double crossoverChance)
```

Write the paramaters of this GA to a file in the folder, so that you can easily check the variables later on.

**Parameters:**

ships - Other ships taking part with the NNShip.  
popSize - Size of the population.  
generationCount - Number of generations being ran.  
gamesPerGeneration - Number of games per generation.  
maxAsteroids - Max number of asteroids per arena created.  
asteroidSpawnChance - Asteroid spawn chance ( 0-100 ).  
mutationChance - Chance that an allele will be mutated.  
crossoverChance - Chance that a chromosome will incur crossover.

---

## createFitnessFile

```
public void createFitnessFile()
```

---

## addToFitnessFile

```
public void addToFitnessFile(int generationNumber,  
    int gamesPlayed,  
    java.util.ArrayList chromosomes)
```

---

## closeFitnessWriter

```
public void closeFitnessWriter()
```

---

(continued from last page)

## io Interface ISaveableChromosome

All Known Implementing Classes:

[Chromosome](#)

---

public interface **ISaveableChromosome**  
extends

Interface that allows the saving of certain objects.

---

### Method Summary

void	<a href="#">saveToFile</a> (java.lang.String folderName) Save elements of a chromosome so that it can be loaded/edited later easily.
------	---

---

### Methods

#### saveToFile

```
public void saveToFile(java.lang.String folderName)  
    throws java.io.IOException
```

Save elements of a chromosome so that it can be loaded/edited later easily.

#### Parameters:

folderName - Name of the folder the chromo has to be saved to.

#### Throws:

IOException - Exception is thrown if there are any issues with file opening/creation/editing.

---

**Package**  
**neuralnetwork**

## neuralnetwork

### Class ENetworkInputs

```

java.lang.Object
  |
  +- java.lang.Enum
        +- neuralnetwork.ENetworkInputs
  
```

#### All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

```

public class ENetworkInputs
extends java.lang.Enum
  
```

An enum that stores a reference to all of the available inputs for a neural network.

Field Summary	
public static final	<a href="#">ASTEROID_ANGLE</a>
public static final	<a href="#">ASTEROID_DISTANCE</a>
public static final	<a href="#">BOTTOM_WALL_DISTANCE</a>
public static final	<a href="#">BULLET_ANGLE</a>
public static final	<a href="#">BULLET_DISTANCE</a>
public static final	<a href="#">ENEMY_ANGLE</a>
public static final	<a href="#">ENEMY_DISTANCE</a>
public static final	<a href="#">LEFT_WALL_DISTANCE</a>
public static final	<a href="#">RIGHT_WALL_DISTANCE</a>
public static final	<a href="#">TOP_WALL_DISTANCE</a>

Method Summary	
abstract double	<a href="#">getInputValue</a> ( <a href="#">AbstractShip</a> ship)
static <a href="#">ENetworkInputs</a>	<a href="#">valueOf</a> (java.lang.String name)
static <a href="#">ENetworkInputs[]</a>	<a href="#">values</a> ()

**Methods inherited from class** java.lang.Enum

```
compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf
```

**Methods inherited from class** `java.lang.Object`

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

**Methods inherited from interface** `java.lang.Comparable`

```
compareTo
```

---

## Fields

### ENEMY\_DISTANCE

```
public static final neuralnetwork.ENetworkInputs ENEMY_DISTANCE
```

---

### ENEMY\_ANGLE

```
public static final neuralnetwork.ENetworkInputs ENEMY_ANGLE
```

---

### ASTEROID\_DISTANCE

```
public static final neuralnetwork.ENetworkInputs ASTEROID_DISTANCE
```

---

### ASTEROID\_ANGLE

```
public static final neuralnetwork.ENetworkInputs ASTEROID_ANGLE
```

---

### BULLET\_DISTANCE

```
public static final neuralnetwork.ENetworkInputs BULLET_DISTANCE
```

---

### BULLET\_ANGLE

```
public static final neuralnetwork.ENetworkInputs BULLET_ANGLE
```

---

### LEFT\_WALL\_DISTANCE

```
public static final neuralnetwork.ENetworkInputs LEFT_WALL_DISTANCE
```

---

(continued from last page)

## RIGHT\_WALL\_DISTANCE

```
public static final neuralnetwork.ENetworkInputs RIGHT_WALL_DISTANCE
```

---

## TOP\_WALL\_DISTANCE

```
public static final neuralnetwork.ENetworkInputs TOP_WALL_DISTANCE
```

---

## BOTTOM\_WALL\_DISTANCE

```
public static final neuralnetwork.ENetworkInputs BOTTOM_WALL_DISTANCE
```

## Methods

### values

```
public static ENetworkInputs\[\] values()
```

---

### valueOf

```
public static ENetworkInputs valueOf(java.lang.String name)
```

---

### getInputValue

```
public abstract double getInputValue(AbstractShip ship)
```



## neuralnetwork

### Class ENetworkOutputs

```

java.lang.Object
|
+- java.lang.Enum
|
+- neuralnetwork.ENetworkOutputs

```

#### All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

```

public final class ENetworkOutputs
extends java.lang.Enum

```

Enum that holds all of the possible output for the evoships neural network. Original version used to have 5 outputs , one for each movement and one for fire. But this was found to be too complicated for the simple NNetwork.

#### Field Summary

public static final	<a href="#">FIRE</a>
public static final	<a href="#">MOVE</a>
public static final	<a href="#">TURN</a>

#### Method Summary

boolean	<a href="#">isActivatedNegativeBound</a> (double value) Check to see if a given value activates the given output in the negative bound, is value < -activationThreshold.
boolean	<a href="#">isActivatedPositiveBound</a> (double value) Check to see if a given value activates the given output in the positive bound, is value > +activationThreshold.
static <a href="#">ENetworkOutputs</a>	<a href="#">valueOf</a> ( java.lang.String name)
static <a href="#">ENetworkOutputs[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

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

#### Methods inherited from interface java.lang.Comparable

compareTo

## Fields

### MOVE

```
public static final neuralnetwork.ENetworkOutputs MOVE
```

---

### TURN

```
public static final neuralnetwork.ENetworkOutputs TURN
```

---

### FIRE

```
public static final neuralnetwork.ENetworkOutputs FIRE
```

## Methods

### values

```
public static ENetworkOutputs\[\] values()
```

---

### valueOf

```
public static ENetworkOutputs valueOf(java.lang.String name)
```

---

### isActivePositiveBound

```
public boolean isActivePositiveBound(double value)
```

Check to see if a given value activates the given output in the positive bound, is value > +activationThreshold.

**Parameters:**

value - Value to check against the activation threshold.

**Returns:**

Whether the value is greater than the POSITIVE activation threshold.

---

### isActiveNegativeBound

```
public boolean isActiveNegativeBound(double value)
```

Check to see if a given value activates the given output in the negative bound, is value < -activationThreshold.

**Parameters:**

value - Value to check against the activation threshold.

(continued from last page)

**Returns:**

Whether the value is less than the negative activation threshold.

## neuralnetwork

### Class ENeuronTypes

```

java.lang.Object
  |
  +- java.lang.Enum
        +- neuralnetwork.ENeuronTypes
  
```

#### All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

```

public final class ENeuronTypes
extends java.lang.Enum
  
```

Enum representing the three different types of neurons that exist. INPUT Neurons. HIDDEN Neurons. OUTPUT Neurons.

### Field Summary

public static final	<a href="#">HIDDEN</a>
public static final	<a href="#">INPUT</a>
public static final	<a href="#">OUTPUT</a>

### Method Summary

static <a href="#">ENeuronTypes</a>	<a href="#">valueOf</a> (java.lang.String name)
static <a href="#">ENeuronTypes[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

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

#### Methods inherited from interface java.lang.Comparable

compareTo

### Fields

#### INPUT

```
public static final neuralnetwork.ENeuronTypes INPUT
```

---

## HIDDEN

```
public static final neuralnetwork.ENeuronTypes HIDDEN
```

---

## OUTPUT

```
public static final neuralnetwork.ENeuronTypes OUTPUT
```

## Methods

### values

```
public static ENeuronTypes\[\] values()
```

---

### valueOf

```
public static ENeuronTypes valueOf(java.lang.String name)
```

## neuralnetwork

### Class Neuron

java.lang.Object

└─neuralnetwork.Neuron

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

### Constructor Summary

public	<a href="#">Neuron</a> ( <a href="#">ENeuronTypes</a> neuronType) Create a new neuron, with a given type.
--------	--

### Method Summary

void	<a href="#">connectToNeuron</a> ( <a href="#">Neuron</a> otherNeuron, double weight) Add a connection to this neuron.
<a href="#">ENeuronTypes</a>	<a href="#">getNeuronType</a> () Get the type of neuron this is as governed by ENeuronTypes.
double	<a href="#">getNeuronValue</a> () Get the value contained at this neuron.
void	<a href="#">setInputValue</a> (double inputValue) Set the value of this neuron if it happens to be an input value.
void	<a href="#">updateNeuron</a> () Update this neuron by calculating it's value from the weight of it's given connections.

### Methods inherited from class java.lang.Object

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

### Constructors

#### Neuron

public **Neuron**([ENeuronTypes](#) neuronType)

Create a new neuron, with a given type.

#### Parameters:

neuronType - Type of neuron to create, INPUT/HIDDEN/OUTPUT.

### Methods

(continued from last page)

---

## connectToNeuron

```
public void connectToNeuron(Neuron otherNeuron,  
    double weight)
```

Add a connection to this neuron. It places it in a hashmap, so that I can easily check what neurons it's connected to and the respective weights of each of those connections.

### Parameters:

otherNeuron - Other neuron to connect to.  
weight - Weight of the connection.

---

## getNeuronType

```
public ENeuronTypes getNeuronType()
```

Get the type of neuron this is as governed by ENeuronTypes.

### Returns:

Type of neuron ( INPUT / HIDDEN / OUTPUT ).

---

## setInputValue

```
public void setInputValue(double inputValue)
```

Set the value of this neuron if it happens to be an input value. The reason for this is because in order to start firing the neurons, the input values need to be set.

### Parameters:

inputValue

---

## getNeuronValue

```
public double getNeuronValue()
```

Get the value contained at this neuron. Requires updateNeuron() to be called before value is set.

### Returns:

Value of the neuron.

---

## updateNeuron

```
public void updateNeuron()
```

Update this neuron by calculating it's value from the weight of it's given connections.

## neuralnetwork Class NNetwork

java.lang.Object

└─neuralnetwork.NNetwork

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

Class that will hold neural network details used by specific ships.

### Constructor Summary

public	<a href="#">NNetwork</a> ( <a href="#">Chromosome</a> chromosome, int inputCount, int hiddenCount, int numberHiddenLayers, int outputCount) Create a new neural network, with the given amount of input / hidden / output layers.
--------	--

### Method Summary

double[]	<a href="#">getNetworkOutputs</a> () Get the outputs of the networks output neurons.
void	<a href="#">setupNeurons</a> () Set up the neurons inside the neural network, in given time this will be given a chromosome representing the random choices for a given chromosome.
void	<a href="#">updateInputNeurons</a> ( <a href="#">AbstractShip</a> ship) Update the input neurons so that they reflect the correct starting values.
void	<a href="#">updateNetwork</a> () This method will force an update of each neuron of the network in turn.

### Methods inherited from class java.lang.Object

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

## Constructors

### NNetwork

```
public NNetwork(Chromosome chromosome,
               int inputCount,
               int hiddenCount,
               int numberHiddenLayers,
               int outputCount)
```

Create a new neural network, with the given amount of input / hidden / output layers.

#### Parameters:

chromosome - Chromosome to be used by this neural network.  
inputCount - Number of input layers.  
hiddenCount - Number of hidden neurons in each layer.  
numberHiddenLayers - Number of hidden layers that exist in the NN.



(continued from last page)

outputCount - Number of output layers.

## Methods

### setupNeurons

```
public void setupNeurons()
```

Set up the neurons inside the neural network, in given time this will be given a chromosome representing the random choices for a given chromosome. At the current time, all neurons are currently being made with a threshold of 0.5, not too sure if I'll remain with thresholds or go with a more basic NN due to time constraints.

---

### updateNetwork

```
public void updateNetwork()
```

This method will force an update of each neuron of the network in turn. INPUT -> HIDDEN -> OUTPUTS.

---

### getNetworkOutputs

```
public double[] getNetworkOutputs()
```

Get the outputs of the networks output neurons.

**Returns:**

Array storing all of the outputs.

---

### updateInputNeurons

```
public void updateInputNeurons(AbstractShip ship)
```

Update the input neurons so that they reflect the correct starting values.

**Parameters:**

ship - Ship to update the neural network for.

---

# Package ui

## ui

# Class ArenaFrame

```

java.lang.Object
  |-- java.awt.Component
        |-- java.awt.Container
              |-- java.awt.Window
                    |-- java.awt.Frame
                          |-- javax.swing.JFrame
                                |-- ui.ArenaFrame

```

### All Implemented Interfaces:

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 ArenaFrame
extends javax.swing.JFrame

```

#### Fields inherited from class javax.swing.JFrame

EXIT\_ON\_CLOSE

#### Fields inherited from class java.awt.Frame

CROSSHAIR\_CURSOR, DEFAULT\_CURSOR, E\_RESIZE\_CURSOR, HAND\_CURSOR, ICONIFIED, MAXIMIZED\_BOTH, MAXIMIZED\_HORIZ, MAXIMIZED\_VERT, MOVE\_CURSOR, N\_RESIZE\_CURSOR, NE\_RESIZE\_CURSOR, NORMAL, NW\_RESIZE\_CURSOR, S\_RESIZE\_CURSOR, SE\_RESIZE\_CURSOR, SW\_RESIZE\_CURSOR, TEXT\_CURSOR, W\_RESIZE\_CURSOR, WAIT\_CURSOR

#### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

#### Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

#### Fields inherited from interface javax.swing.WindowConstants

DISPOSE\_ON\_CLOSE, DO\_NOTHING\_ON\_CLOSE, EXIT\_ON\_CLOSE, HIDE\_ON\_CLOSE

## Constructor Summary

public [ArenaFrame](#)([Arena](#) game, java.awt.Dimension frameSize)

#### Methods inherited from class javax.swing.JFrame

getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getGraphics, getJMenuBar, getLayeredPane, getRootPane, getTransferHandler, isDefaultLookAndFeelDecorated, remove, repaint, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setIconImage, setJMenuBar, setLayeredPane, setLayout, setTransferHandler, update

#### Methods inherited from class java.awt.Frame

addNotify, getAccessibleContext, getCursorType, getExtendedState, getFrames, getIconImage, getMaximizedBounds, getMenuBar, getState, getTitle, isResizable, isUndecorated, remove, removeNotify, setBackground, setCursor, setExtendedState, setIconImage, setMaximizedBounds, setMenuBar, setOpacity, setResizable, setShape, setState, setTitle, setUndecorated

#### Methods inherited from class java.awt.Window

addNotify, addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getAccessibleContext, getBackground, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getIconImages, getInputContext, getListeners, getLocale, getModalExclusionType, getMostRecentFocusOwner, getOpacity, getOwnedWindows, getOwner, getOwnerlessWindows, getShape, getToolkit, getType, getWarningString, getWindowFocusListeners, getWindowListeners, getWindows, getWindowStateListeners, hide, isActive, isAlwaysOnTop, isAlwaysOnTopSupported, isAutoRequestFocus, isFocusableWindow, isFocusCycleRoot, isFocused, isLocationByPlatform, isOpaque, isShowing, isValidRoot, pack, paint, postEvent, removeNotify, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, reshape, setAlwaysOnTop, setAutoRequestFocus, setBackground, setBounds, setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot, setIconImage, setIconImages, setLocation, setLocation, setLocationByPlatform, setLocationRelativeTo, setMinimumSize, setModalExclusionType, setOpacity, setShape, setSize, setSize, setType, setVisible, show, toBack, toFront

#### Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, isValidRoot, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, remove, remove, removeAll, removeContainerListener, removeNotify, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout, transferFocusDownCycle, update, validate

#### Methods inherited from class java.awt.Component

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

imageUpdate
-------------

```
getFont, postEvent, remove
```

```
getAccessibleContext
```

---

---

```
getFont, postEvent, remove
```

**Methods inherited from interface** javax.accessibility.Accessible

```
getAccessibleContext
```

**Methods inherited from interface** javax.swing.RootPaneContainer

```
getContentPane, getGlassPane, getLayeredPane, getRootPane, setContentPane,  
setGlassPane, setLayeredPane
```

**Methods inherited from interface** javax.swing.TransferHandler.HasGetTransferHandler

```
getTransferHandler
```

---

## Constructors

### ArenaFrame

```
public ArenaFrame(Arena game,  
                  java.awt.Dimension frameSize)
```

## ui

# Class ArenaPanel

```

java.lang.Object
  |-- java.awt.Component
        |-- java.awt.Container
              |-- javax.swing.JComponent
                    |-- javax.swing.JPanel
                          |-- ui.ArenaPanel

```

### All Implemented Interfaces:

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

```

public class ArenaPanel
extends javax.swing.JPanel
implements javax.accessibility.Accessible, java.io.Serializable,
javax.swing.TransferHandler.HasGetTransferHandler, java.awt.image.ImageObserver,
java.awt.MenuContainer, java.io.Serializable, java.util.Observer

```

The arena panel is used whenever the arena has to be drawn for single game-purposes. This class is an observer as it will receive notifications of when to update from the arena, which is an observable.

#### Fields inherited from class javax.swing.JComponent

TOOL\_TIP\_TEXT\_KEY, UNDEFINED\_CONDITION, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT, WHEN\_FOCUSED, WHEN\_IN\_FOCUSED\_WINDOW

#### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

#### Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

## Constructor Summary

public	<a href="#">ArenaPanel</a> ( <a href="#">Arena</a> currentArena, java.awt.Dimension panelSize) Creates a new arena
--------	---

## Method Summary

void	<a href="#">update</a> (java.util.Observable arena, java.lang.Object arg1)
------	--

#### Methods inherited from class javax.swing.JPanel

getAccessibleContext, getUI, getUIClassID, setUI, updateUI

#### Methods inherited from class javax.swing.JComponent

```

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect,
contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange,
firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap,
getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBaseline,
getBaselineResizeBehavior, getBorder, getBounds, getClientProperty,
getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions,
getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu,
getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners,
getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent,
getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize,
getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor,
getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget,
getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus,
isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque,
isOptimizedDrawingEnabled, isPaintingForPrint, isPaintingTile, isRequestFocusEnabled,
isValidateRoot, paint, paintImmediately, paintImmediately, print, printAll,
putClientProperty, registerKeyboardAction, registerKeyboardAction,
removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint,
requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow,
resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap,
setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder,
setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered,
setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu,
setInputMap, setInputVerifier, setMaximumSize, setMinimumSize,
setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled,
setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible,
unregisterKeyboardAction, update, updateUI

```

#### Methods inherited from class java.awt.Container

```

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt,
getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt,
getComponentCount, getComponents, getComponentZOrder, getContainerListeners,
getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners,
getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets,
invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot,
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, isValidateRoot, layout,
list, list, locate, minimumSize, paint, paintComponents, preferredSize, print,
printComponents, remove, remove, removeAll, removeContainerListener, removeNotify,
setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys,
setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout,
transferFocusDownCycle, update, validate

```

#### Methods inherited from class java.awt.Component



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

imageUpdate

```
getFont, postEvent, remove
```

```
getTransferHandler
```

---

getAccessibleContext
----------------------

<b>Methods inherited from interface</b> <code>java.util.Observer</code>
---

update
--------

---

## Constructors

### ArenaPanel

```
public ArenaPanel(Arena currentArena,  
                  java.awt.Dimension panelSize)
```

Creates a new arena

**Parameters:**

currentArena - The Arena the game has to be based upon.

panelSize - The dimensions of the panel to be created.

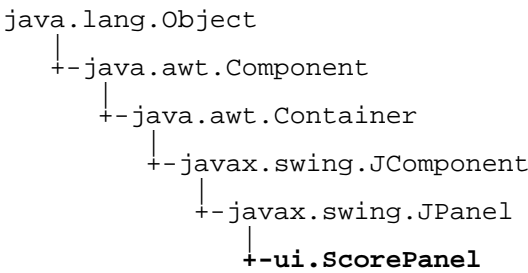
## Methods

### update

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

ui

# Class ScorePanel



All Implemented Interfaces:  
java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

public class **ScorePanel**  
extends javax.swing.JPanel

Fields inherited from class javax.swing.JComponent
TOOL_TIP_TEXT_KEY, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW
Fields inherited from class java.awt.Component
BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT
Fields inherited from interface java.awt.image.ImageObserver
ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary	
public	<a href="#">ScorePanel</a> (java.awt.Dimension screenSize, java.util.ArrayList ships)

Methods inherited from class javax.swing.JPanel
getAccessibleContext, getUI, getUIClassID, setUI, updateUI
Methods inherited from class javax.swing.JComponent

```

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect,
contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange,
firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap,
getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBaseline,
getBaselineResizeBehavior, getBorder, getBounds, getClientProperty,
getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions,
getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu,
getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners,
getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent,
getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize,
getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor,
getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget,
getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus,
isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque,
isOptimizedDrawingEnabled, isPaintingForPrint, isPaintingTile, isRequestFocusEnabled,
isValidateRoot, paint, paintImmediately, paintImmediately, print, printAll,
putClientProperty, registerKeyboardAction, registerKeyboardAction,
removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint,
requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow,
resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap,
setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder,
setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered,
setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu,
setInputMap, setInputVerifier, setMaximumSize, setMinimumSize,
setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled,
setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible,
unregisterKeyboardAction, update, updateUI

```

#### Methods inherited from class java.awt.Container

```

add, add, add, add, add, addContainerListener, addNotify, addPropertyChangeListener,
addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet,
countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt,
getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt,
getComponentCount, getComponents, getComponentZOrder, getContainerListeners,
getFocusTraversalKeys, getFocusTraversalPolicy, getInsets, getLayout, getListeners,
getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets,
invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot,
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, isValidateRoot, layout,
list, list, locate, minimumSize, paint, paintComponents, preferredSize, print,
printComponents, remove, remove, removeAll, removeContainerListener, removeNotify,
setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys,
setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout,
transferFocusDownCycle, update, validate

```

#### Methods inherited from class java.awt.Component

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

imageUpdate

```
getFont, postEvent, remove
```

```
getTransferHandler
```

---

getAccessibleContext

---

## Constructors

### ScorePanel

```
public ScorePanel(java.awt.Dimension screenSize,  
                  java.util.ArrayList ships)
```