

[PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)[SUMMARY: NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

code

## Class AlienBomb

java.lang.Object  
code.AlienBomb

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

This class is used to control a bomb that each alien can try to drop onto the shooter (player)

### Field Summary

#### Fields

Modifier and Type	Field and Description
-------------------	-----------------------

int	<b>bombSpeed</b>
(package private) java.awt.Rectangle	<b>boundingBox</b>
(package private) boolean	<b>collision</b>
	Boolean variable specifying if there was a collision between the player (shooter) and the bomb
int	<b>height</b>
boolean	<b>isShooting</b>
	If value is true, a bomb is currently in motion.
boolean	<b>moveLeft</b>
boolean	<b>moveRight</b>
int	<b>moveSpeed</b>
int	<b>width</b>
int	<b>x</b>
int	<b>y</b>

### Constructor Summary

#### Constructors

**Constructor and Description**

**AlienBomb**(**Alien** alien)

A bomb is positioned directly behind each alien.

**Method Summary**

**All Methods**    **Instance Methods**    **Concrete Methods**

Modifier and Type	Method and Description
private void	<b>collide</b> ( <b>spaceDefender</b> game) collide method is to determine if there is a collision between the alien bomb and the shooter.
void	<b>render</b> (java.awt.Graphics graphics) Strictly for displaying graphics associated with AlienBomb instance
void	<b>tick</b> ( <b>spaceDefender</b> game, <b>Alien</b> alien) tick actuates the movement of the bomb.

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

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

**Field Detail**

**x**

public int x

**y**

public int y

**bombSpeed**

public int bombSpeed

**width**

public final int width

**See Also:**[Constant Field Values](#)**height**

```
public final int height
```

**See Also:**[Constant Field Values](#)**isShooting**

```
public boolean isShooting
```

If value is true, a bomb is currently in motion. Value is false otherwise.

**moveLeft**

```
public boolean moveLeft
```

**moveRight**

```
public boolean moveRight
```

**moveSpeed**

```
public int moveSpeed
```

**boundingBox**

```
java.awt.Rectangle boundingBox
```

**collision**

```
boolean collision
```

Boolean variable specifying if there was a collision between the player (shooter) and the bomb

**Constructor Detail****AlienBomb**

```
public AlienBomb(Alien alien)
```

A bomb is positioned directly behind each alien.

**Parameters:**

alien - Instance of Alien that newly constructed alien bomb will belong to

## Method Detail

### tick

```
public void tick(spaceDefender game,  
                Alien alien)
```

tick actuates the movement of the bomb. As long as the bomb isn't scheduled to drop, it maintains position behind the alien. If bomb drops below the shooter, its position is reset to behind the alien

**Parameters:**

game - Driver instance of the game

alien - Instance of Alien that alien bomb belongs to

### collide

```
private void collide(spaceDefender game)
```

collide method is to determine if there is a collision between the alien bomb and the shooter.

**Parameters:**

game - Driver instance of the game

### render

```
public void render(java.awt.Graphics graphics)
```

Strictly for displaying graphics associated with AlienBomb instance

**Parameters:**

graphics - Driver instance of the game

