

## Plants vs Zombies Simulation

Generated by Doxygen 1.11.0



<b>1 Plants vs Zombies simulation</b>	<b>1</b>
1.1 Author	1
1.2 Description	1
1.3 Class Diagram	1
1.4 Documentation	1
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List	5
<b>4 File Index</b>	<b>7</b>
4.1 File List	7
<b>5 Class Documentation</b>	<b>9</b>
5.1 BasicZombie Class Reference	9
5.1.1 Detailed Description	10
5.1.2 Constructor & Destructor Documentation	10
5.1.2.1 BasicZombie()	10
5.1.3 Member Function Documentation	10
5.1.3.1 actionPerformed()	10
5.1.3.2 getAttackDamage()	11
5.1.3.3 getBounds()	11
5.1.3.4 getHealth()	11
5.1.3.5 getSpeed()	11
5.1.3.6 getType()	12
5.1.3.7 paintComponent()	12
5.1.3.8 takeDamage()	12
5.2 Board Class Reference	12
5.2.1 Detailed Description	13
5.2.2 Constructor & Destructor Documentation	13
5.2.2.1 Board()	13
5.3 BucketheadZombie Class Reference	13
5.3.1 Detailed Description	14
5.3.2 Constructor & Destructor Documentation	14
5.3.2.1 BucketheadZombie()	14
5.3.3 Member Function Documentation	14
5.3.3.1 actionPerformed()	14
5.3.3.2 getAttackDamage()	15
5.3.3.3 getBounds()	15
5.3.3.4 getHealth()	15
5.3.3.5 getSpeed()	15
5.3.3.6 getType()	16

5.3.3.7 paintComponent()	16
5.3.3.8 takeDamage()	16
5.4 CherryBomb Class Reference	16
5.4.1 Detailed Description	17
5.4.2 Constructor & Destructor Documentation	17
5.4.2.1 CherryBomb()	17
5.4.3 Member Function Documentation	18
5.4.3.1 actionPerformed()	18
5.4.3.2 explode()	18
5.4.3.3 getBounds()	18
5.4.3.4 getCost()	18
5.4.3.5 getHealth()	19
5.4.3.6 getType()	19
5.4.3.7 paintComponent()	19
5.4.3.8 takeDamage()	19
5.4.4 Member Data Documentation	20
5.4.4.1 Zombies	20
5.5 CollisionManager Class Reference	20
5.5.1 Detailed Description	20
5.5.2 Member Function Documentation	20
5.5.2.1 checkAttacks()	20
5.5.2.2 checkExplosionDeaths()	20
5.5.2.3 checkProjectileHit()	20
5.6 Entity Class Reference	21
5.6.1 Detailed Description	21
5.6.2 Constructor & Destructor Documentation	21
5.6.2.1 Entity()	21
5.6.3 Member Function Documentation	22
5.6.3.1 actionPerformed()	22
5.6.3.2 getBounds()	22
5.6.3.3 getHealth()	22
5.6.3.4 getType()	23
5.6.3.5 paintComponent()	23
5.6.3.6 takeDamage()	23
5.6.4 Member Data Documentation	23
5.6.4.1 health	23
5.6.4.2 imageBounds	23
5.6.4.3 theTimer	24
5.6.4.4 x	24
5.6.4.5 y	24
5.7 Explosion Class Reference	24
5.7.1 Detailed Description	24

5.7.2 Constructor & Destructor Documentation	24
5.7.2.1 Explosion()	24
5.7.3 Member Function Documentation	25
5.7.3.1 getBounds()	25
5.7.3.2 getDamage()	25
5.7.3.3 paintComponent()	25
5.7.4 Member Data Documentation	25
5.7.4.1 x	25
5.7.4.2 y	26
5.8 Main Class Reference	26
5.8.1 Detailed Description	26
5.8.2 Member Function Documentation	26
5.8.2.1 main()	26
5.9 Panel Class Reference	26
5.9.1 Detailed Description	27
5.9.2 Constructor & Destructor Documentation	27
5.9.2.1 Panel()	27
5.9.3 Member Function Documentation	27
5.9.3.1 actionPerformed()	27
5.9.3.2 counterUpdater()	28
5.9.3.3 gameStart()	28
5.9.3.4 paintComponent()	28
5.9.3.5 spawnRandomPlant()	28
5.9.3.6 spawnRandomZombie()	29
5.9.4 Member Data Documentation	29
5.9.4.1 BASIC_ZOMBIE_SPAWN_CHANCE	29
5.9.4.2 BUCKETHEAD_ZOMBIE_SPAWN_CHANCE	29
5.9.4.3 CHERRY_BOMB_SPAWN_CHANCE	29
5.9.4.4 COLUMNS	29
5.9.4.5 DELAY	29
5.9.4.6 PEASHOOTER_SPAWN_CHANCE	29
5.9.4.7 PLANT_COLUMNS	30
5.9.4.8 PLANT_SPAWN_INTERVAL	30
5.9.4.9 Plants	30
5.9.4.10 resourceManager	30
5.9.4.11 ROWS	30
5.9.4.12 SpawnSquares	30
5.9.4.13 SQUARE_SIZE	30
5.9.4.14 START_SUN_POINTS	30
5.9.4.15 START_ZOMBIE_AMOUNT	31
5.9.4.16 strings	31
5.9.4.17 SUNFLOWER_SPAWN_CHANCE	31

5.9.4.18 theTimer	31
5.9.4.19 WALNUT_SPAWN_CHANCE	31
5.9.4.20 ZOMBIE_COLUMNS	31
5.9.4.21 ZOMBIE_SPAWN_INTERVAL	31
5.9.4.22 Zombies	32
5.10 Peashooter Class Reference	32
5.10.1 Detailed Description	33
5.10.2 Constructor & Destructor Documentation	33
5.10.2.1 Peashooter()	33
5.10.3 Member Function Documentation	33
5.10.3.1 actionPerformed()	33
5.10.3.2 getBounds()	34
5.10.3.3 getCost()	34
5.10.3.4 getHealth()	34
5.10.3.5 getType()	34
5.10.3.6 paintComponent()	34
5.10.3.7 shoot()	35
5.10.3.8 takeDamage()	35
5.10.4 Member Data Documentation	35
5.10.4.1 Projectiles	35
5.10.4.2 Zombies	35
5.11 Plant Class Reference	36
5.11.1 Detailed Description	36
5.11.2 Constructor & Destructor Documentation	36
5.11.2.1 Plant()	36
5.12 Projectile Class Reference	37
5.12.1 Detailed Description	37
5.12.2 Constructor & Destructor Documentation	37
5.12.2.1 Projectile()	37
5.12.3 Member Function Documentation	38
5.12.3.1 actionPerformed()	38
5.12.3.2 getBounds()	38
5.12.3.3 getDamage()	38
5.12.3.4 getWidth()	38
5.12.3.5 paintComponent()	38
5.12.4 Member Data Documentation	39
5.12.4.1 theTimer	39
5.12.4.2 x	39
5.12.4.3 y	39
5.13 ResourceManager Class Reference	39
5.13.1 Detailed Description	39
5.13.2 Constructor & Destructor Documentation	39

5.13.2.1 ResourceManager()	39
5.13.3 Member Function Documentation	39
5.13.3.1 addSunPoints()	39
5.13.3.2 getSunPoints()	40
5.13.3.3 spendSunPoints()	40
5.14 SettingsChange Class Reference	40
5.14.1 Detailed Description	41
5.14.2 Constructor & Destructor Documentation	41
5.14.2.1 SettingsChange()	41
5.14.3 Member Function Documentation	41
5.14.3.1 stateChanged()	41
5.14.4 Member Data Documentation	41
5.14.4.1 BASIC_ZOMBIE_SPAWN_CHANCE	41
5.14.4.2 BUCKETHEAD_ZOMBIE_SPAWN_CHANCE	42
5.14.4.3 CHERRY_BOMB_SPAWN_CHANCE	42
5.14.4.4 DELAY	42
5.14.4.5 PEASHOOTER_SPAWN_CHANCE	42
5.14.4.6 PLANT_SPAWN_INTERVAL	42
5.14.4.7 START_SUN_POINTS	42
5.14.4.8 START_ZOMBIE_AMOUNT	42
5.14.4.9 strings	43
5.14.4.10 SUNFLOWER_SPAWN_CHANCE	43
5.14.4.11 values	43
5.14.4.12 WALNUT_SPAWN_CHANCE	43
5.14.4.13 ZOMBIE_SPAWN_INTERVAL	43
5.15 SpawnSelector Class Reference	43
5.15.1 Detailed Description	44
5.15.2 Constructor & Destructor Documentation	44
5.15.2.1 SpawnSelector()	44
5.15.3 Member Function Documentation	44
5.15.3.1 actionPerformed()	44
5.15.3.2 findArrayIndex()	45
5.15.3.3 mouseClicked()	45
5.15.3.4 mouseEntered()	45
5.15.3.5 mouseExited()	45
5.15.3.6 mousePressed()	45
5.15.3.7 mouseReleased()	46
5.15.3.8 paint()	46
5.15.4 Member Data Documentation	46
5.15.4.1 SpawnSquares	46
5.15.4.2 theTimer	46
5.16 Sunflower Class Reference	46

5.16.1 Detailed Description	48
5.16.2 Constructor & Destructor Documentation	48
5.16.2.1 Sunflower()	48
5.16.3 Member Function Documentation	48
5.16.3.1 actionPerformed()	48
5.16.3.2 getBounds()	48
5.16.3.3 getCost()	49
5.16.3.4 getHealth()	49
5.16.3.5 getType()	49
5.16.3.6 paintComponent()	49
5.16.3.7 takeDamage()	49
5.16.4 Member Data Documentation	50
5.16.4.1 alive	50
5.16.4.2 produceCycle	50
5.16.4.3 resourceManager	50
5.17 Walnut Class Reference	50
5.17.1 Detailed Description	51
5.17.2 Constructor & Destructor Documentation	51
5.17.2.1 Walnut()	51
5.17.3 Member Function Documentation	51
5.17.3.1 actionPerformed()	51
5.17.3.2 getBounds()	52
5.17.3.3 getCost()	52
5.17.3.4 getHealth()	52
5.17.3.5 getType()	52
5.17.3.6 paintComponent()	52
5.17.3.7 takeDamage()	53
5.18 Zombie Class Reference	53
5.18.1 Detailed Description	54
5.18.2 Constructor & Destructor Documentation	54
5.18.2.1 Zombie()	54
5.18.3 Member Function Documentation	54
5.18.3.1 getAttackDamage()	54
5.18.3.2 getSpeed()	55
5.18.3.3 getType()	55
<b>6 File Documentation</b>	<b>57</b>
6.1 README.md File Reference	57
6.2 BasicZombie.java File Reference	57
6.3 Board.java File Reference	57
6.4 BucketheadZombie.java File Reference	57
6.5 CherryBomb.java File Reference	57



---

6.6 CollisionManager.java File Reference . . . . .	58
6.7 Entity.java File Reference . . . . .	58
6.8 Explosion.java File Reference . . . . .	58
6.9 Main.java File Reference . . . . .	58
6.10 Panel.java File Reference . . . . .	58
6.11 Peashooter.java File Reference . . . . .	58
6.12 Plant.java File Reference . . . . .	58
6.13 Projectile.java File Reference . . . . .	59
6.14 ResourceManager.java File Reference . . . . .	59
6.15 SettingsChange.java File Reference . . . . .	59
6.16 SpawnSelector.java File Reference . . . . .	59
6.17 Sunflower.java File Reference . . . . .	59
6.18 Walnut.java File Reference . . . . .	59
6.19 Zombie.java File Reference . . . . .	59
<b>Index</b>	<b>61</b>



# Chapter 1

## Plants vs Zombies simulation

A Plants vs Zombies-themed Agent-based model simulation, visualised using Java Swing library.

### 1.1 Author

- [Antoni Nasternak](#)

### 1.2 Description

In this simulation, plants spawning on the left side of the board need to defend against zombies, coming from the right side of the board. The board is a 5x9 squares, first 5x5 squares are designated for plant spawning and last column is the place of zombies spawning. Plants spawn randomly in predetermined spawn squares, that the user needs to specify before the simulation. There are different types of plants, that each play a different role in the defense:

- Sunflowers generate Sun Points, which are needed to create other plants.
- Peashooters spawn peas (Projectiles), that go towards zombies to deal damage
- Cherry Bombs explode on death or after a certain amount of time, killing all zombies in a 3x3 radius
- Walnuts defend other plants by having a huge amount of health There are also currently two types of zombies:
- Basic Zombies aren't that strong, but they are fast
- Buckethead Zombies are the opposite, they deal more damage, whilst being slower. The simulation ends when either all zombies are killed, or one of the zombies reaches the left border of the board. A user can change different settings, like percentage of certain plants/zombies spawning or the interval between plants/zombies spawning using the GUI based on the Java Swing library.

### 1.3 Class Diagram

### 1.4 Documentation

[A javadoc Documentation](#)



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

CollisionManager . . . . .	20
Explosion . . . . .	24
JFrame	
Board . . . . .	12
SettingsChange . . . . .	40
SpawnSelector . . . . .	43
JPanel	
Panel . . . . .	26
Main . . . . .	26
ResourceManager . . . . .	39
ActionListener	
Entity . . . . .	21
Plant . . . . .	36
CherryBomb . . . . .	16
Peashooter . . . . .	32
Sunflower . . . . .	46
Walnut . . . . .	50
Zombie . . . . .	53
BasicZombie . . . . .	9
BucketheadZombie . . . . .	13
Panel . . . . .	26
Projectile . . . . .	37
SpawnSelector . . . . .	43
ChangeListener	
SettingsChange . . . . .	40
MouseListener	
SpawnSelector . . . . .	43



## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">BasicZombie</a>	9
<a href="#">Board</a>	12
<a href="#">BucketheadZombie</a>	13
<a href="#">CherryBomb</a>	16
<a href="#">CollisionManager</a>	20
<a href="#">Entity</a>	21
<a href="#">Explosion</a>	24
<a href="#">Main</a>	26
<a href="#">Panel</a>	26
<a href="#">Peashooter</a>	32
<a href="#">Plant</a>	36
<a href="#">Projectile</a>	37
<a href="#">ResourceManager</a>	39
<a href="#">SettingsChange</a>	40
<a href="#">SpawnSelector</a>	43
<a href="#">Sunflower</a>	46
<a href="#">Walnut</a>	50
<a href="#">Zombie</a>	53





# Chapter 4

## File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">BasicZombie.java</a>	57
<a href="#">Board.java</a>	57
<a href="#">BucketheadZombie.java</a>	57
<a href="#">CherryBomb.java</a>	57
<a href="#">CollisionManager.java</a>	58
<a href="#">Entity.java</a>	58
<a href="#">Explosion.java</a>	58
<a href="#">Main.java</a>	58
<a href="#">Panel.java</a>	58
<a href="#">Peashooter.java</a>	58
<a href="#">Plant.java</a>	58
<a href="#">Projectile.java</a>	59
<a href="#">ResourceManager.java</a>	59
<a href="#">SettingsChange.java</a>	59
<a href="#">SpawnSelector.java</a>	59
<a href="#">Sunflower.java</a>	59
<a href="#">Walnut.java</a>	59
<a href="#">Zombie.java</a>	59

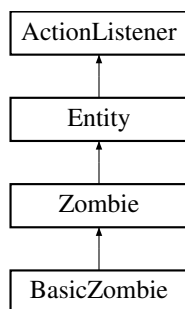


## Chapter 5

# Class Documentation

### 5.1 BasicZombie Class Reference

Inheritance diagram for BasicZombie:



#### Public Member Functions

- [BasicZombie](#) (int *x*, int *y*)
- int [getHealth](#) ()
- int [getSpeed](#) ()
- int [getType](#) ()
- void [paintComponent](#) (Graphics *g*)
- void [takeDamage](#) (int *damage*)
- int [getAttackDamage](#) ()
- Rectangle [getBounds](#) ()
- void [actionPerformed](#) (ActionEvent *e*)

#### Public Member Functions inherited from [Zombie](#)

- [Zombie](#) (int *x*, int *y*)

#### Public Member Functions inherited from [Entity](#)

- [Entity](#) (int *x*, int *y*)

## Additional Inherited Members

### Protected Attributes inherited from [Entity](#)

- int [x](#)
- int [y](#)
- int [health](#)
- Rectangle [imageBounds](#)

### Static Protected Attributes inherited from [Entity](#)

- static Timer [theTimer](#) = Panel.theTimer

## 5.1.1 Detailed Description

A weaker, but faster type of zombie.

## 5.1.2 Constructor & Destructor Documentation

### 5.1.2.1 BasicZombie()

```
BasicZombie.BasicZombie (
    int x,
    int y)
```

Constructor of a zombie

#### Parameters

<i>x</i>	x coordinate
<i>y</i>	y coordinate

## 5.1.3 Member Function Documentation

### 5.1.3.1 actionPerformed()

```
void BasicZombie.actionPerformed (
    ActionEvent e)
```

Every [Panel#theTimer](#) tick, this function is called. Used for updating position, health, image on screen etc.

#### Parameters

<i>e</i>	tick of <a href="#">Panel#theTimer</a>
----------	--

Reimplemented from [Entity](#).

### 5.1.3.2 getAttackDamage()

```
int BasicZombie.getAttackDamage ()
```

Returns the attack damage of a zombie entity.

#### Returns

attack damage of a zombie entity

Reimplemented from [Zombie](#).

### 5.1.3.3 getBounds()

```
Rectangle BasicZombie.getBounds ()
```

Returns the entity's hitbox

#### Returns

entity's hitbox

Reimplemented from [Entity](#).

### 5.1.3.4 getHealth()

```
int BasicZombie.getHealth ()
```

Returns the amount of health of an entity.

#### Returns

health amount

Reimplemented from [Entity](#).

### 5.1.3.5 getSpeed()

```
int BasicZombie.getSpeed ()
```

Returns the speed of a zombie entity.

#### Returns

speed of a zombie entity

Reimplemented from [Zombie](#).

### 5.1.3.6 `getType()`

```
int BasicZombie.getType ()
```

Returns the type value of a zombie entity.

#### Returns

type value of a zombie entity

Reimplemented from [Zombie](#).

### 5.1.3.7 `paintComponent()`

```
void BasicZombie.paintComponent (  
    Graphics g)
```

Paints the entity on screen.

#### Parameters

<i>g</i>	required to paint on screen
----------	-----------------------------

Reimplemented from [Entity](#).

### 5.1.3.8 `takeDamage()`

```
void BasicZombie.takeDamage (  
    int damage)
```

Remove a specified amount of health from an entity.

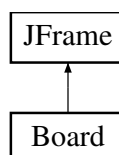
#### Parameters

<i>damage</i>	amount of damage dealt
---------------	------------------------

Reimplemented from [Entity](#).

## 5.2 Board Class Reference

Inheritance diagram for Board:



## Public Member Functions

- [Board](#) ()

### 5.2.1 Detailed Description

The window, where the simulation is created. Simple JFrame, that adds the [Panel](#) to the window.

### 5.2.2 Constructor & Destructor Documentation

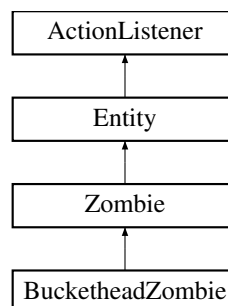
#### 5.2.2.1 Board()

```
Board.Board ()
```

Creates the window and adds the [Panel](#).

## 5.3 BucketheadZombie Class Reference

Inheritance diagram for BucketheadZombie:



## Public Member Functions

- [BucketheadZombie](#) (int x, int y)
- int [getHealth](#) ()
- int [getSpeed](#) ()
- int [getType](#) ()
- void [paintComponent](#) (Graphics g)
- void [takeDamage](#) (int damage)
- int [getAttackDamage](#) ()
- Rectangle [getBounds](#) ()
- void [actionPerformed](#) (ActionEvent e)

## Public Member Functions inherited from [Zombie](#)

- [Zombie](#) (int x, int y)

## Public Member Functions inherited from [Entity](#)

- [Entity](#) (int [x](#), int [y](#))

## Additional Inherited Members

## Protected Attributes inherited from [Entity](#)

- int [x](#)
- int [y](#)
- int [health](#)
- Rectangle [imageBounds](#)

## Static Protected Attributes inherited from [Entity](#)

- static Timer [theTimer](#) = Panel.theTimer

### 5.3.1 Detailed Description

A stronger, but slower type of zombie.

### 5.3.2 Constructor & Destructor Documentation

#### 5.3.2.1 BucketheadZombie()

```
BucketheadZombie.BucketheadZombie (
    int x,
    int y)
```

Constructor

#### Parameters

<i>x</i>	x coordinate
<i>y</i>	y coordinate

### 5.3.3 Member Function Documentation

#### 5.3.3.1 actionPerformed()

```
void BucketheadZombie.actionPerformed (
   (ActionEvent e)
```

Every [Panel#theTimer](#) tick, this function is called. Used for updating position, health, image on screen etc.



#### Parameters

<i>e</i>	tick of <a href="#">Panel#theTimer</a>
----------	--

Reimplemented from [Entity](#).

#### 5.3.3.2 getAttackDamage()

```
int BucketheadZombie.getAttackDamage ()
```

Returns the attack damage of a zombie entity.

#### Returns

attack damage of a zombie entity

Reimplemented from [Zombie](#).

#### 5.3.3.3 getBounds()

```
Rectangle BucketheadZombie.getBounds ()
```

Returns the entity's hitbox

#### Returns

entity's hitbox

Reimplemented from [Entity](#).

#### 5.3.3.4 getHealth()

```
int BucketheadZombie.getHealth ()
```

Returns the amount of health of an entity.

#### Returns

health amount

Reimplemented from [Entity](#).

#### 5.3.3.5 getSpeed()

```
int BucketheadZombie.getSpeed ()
```

Returns the speed of a zombie entity.

#### Returns

speed of a zombie entity

Reimplemented from [Zombie](#).

#### 5.3.3.6 getType()

```
int BucketheadZombie.getType ()
```

Returns the type value of a zombie entity.

##### Returns

type value of a zombie entity

Reimplemented from [Zombie](#).

#### 5.3.3.7 paintComponent()

```
void BucketheadZombie.paintComponent (  
    Graphics g)
```

Paints the entity on screen.

##### Parameters

<i>g</i>	required to paint on screen
----------	-----------------------------

Reimplemented from [Entity](#).

#### 5.3.3.8 takeDamage()

```
void BucketheadZombie.takeDamage (  
    int damage)
```

Remove a specified amount of health from an entity.

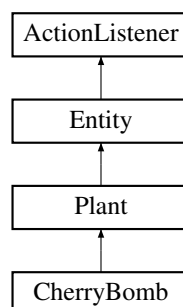
##### Parameters

<i>damage</i>	amount of damage dealt
---------------	------------------------

Reimplemented from [Entity](#).

## 5.4 CherryBomb Class Reference

Inheritance diagram for CherryBomb:



### Public Member Functions

- [CherryBomb](#) (int *x*, int *y*)
- int [getType](#) ()
- void [paintComponent](#) (Graphics *g*)
- void [takeDamage](#) (int *damage*)
- int [getHealth](#) ()
- void [explode](#) ()
- void [actionPerformed](#) (ActionEvent *e*)
- Rectangle [getBounds](#) ()

### Public Member Functions inherited from [Plant](#)

- [Plant](#) (int *x*, int *y*)

### Public Member Functions inherited from [Entity](#)

- [Entity](#) (int *x*, int *y*)

### Static Public Member Functions

- static int [getCost](#) ()

### Public Attributes

- List< [Zombie](#) > [Zombies](#)

### Additional Inherited Members

### Protected Attributes inherited from [Entity](#)

- int *x*
- int *y*
- int *health*
- Rectangle *imageBounds*

### Static Protected Attributes inherited from [Entity](#)

- static Timer *theTimer* = Panel.theTimer

## 5.4.1 Detailed Description

A plant, that explodes in a 3x3 square radius after death or certain amount of time.

## 5.4.2 Constructor & Destructor Documentation

### 5.4.2.1 CherryBomb()

```
CherryBomb.CherryBomb (  
    int x,  
    int y)
```

Constructor of a plant

## Parameters

<i>x</i>	x coordinate
<i>y</i>	y coordinate

### 5.4.3 Member Function Documentation

#### 5.4.3.1 actionPerformed()

```
void CherryBomb.actionPerformed (  
    ActionEvent e)
```

Every [Panel#theTimer](#) tick, this function is called. Used for updating position, health, image on screen etc.

## Parameters

<i>e</i>	tick of <a href="#">Panel#theTimer</a>
----------	--

Reimplemented from [Entity](#).

#### 5.4.3.2 explode()

```
void CherryBomb.explode ()
```

Explode by creating an [Explosion](#).

#### 5.4.3.3 getBounds()

```
Rectangle CherryBomb.getBounds ()
```

Returns the entity's hitbox

## Returns

entity's hitbox

Reimplemented from [Entity](#).

#### 5.4.3.4 getCost()

```
static int CherryBomb.getCost () [static]
```

Returns cost of the plant in Sun Points.

## Returns

cost of the plant in Sun Points

#### 5.4.3.5 getHealth()

```
int CherryBomb.getHealth ()
```

Returns the amount of health of an entity.

##### Returns

health amount

Reimplemented from [Entity](#).

#### 5.4.3.6 getType()

```
int CherryBomb.getType ()
```

Returns the type value defined by each entity.

##### Returns

type of entity

Reimplemented from [Entity](#).

#### 5.4.3.7 paintComponent()

```
void CherryBomb.paintComponent (  
    Graphics g)
```

Paints the entity on screen.

##### Parameters

<i>g</i>	required to paint on screen
----------	-----------------------------

Reimplemented from [Entity](#).

#### 5.4.3.8 takeDamage()

```
void CherryBomb.takeDamage (  
    int damage)
```

Remove a specified amount of health from an entity.

##### Parameters

<i>damage</i>	amount of damage dealt
---------------	------------------------

Reimplemented from [Entity](#).

## 5.4.4 Member Data Documentation

### 5.4.4.1 Zombies

List<Zombie> CherryBomb.Zombies

Copy of Panel#Zombies used in CollisionManager#checkExplosionDeaths.

## 5.5 CollisionManager Class Reference

### Static Public Member Functions

- static void [checkAttacks](#) (List< [Plant](#) > plants, List< [Zombie](#) > zombies)
- static boolean [checkProjectileHit](#) ([Projectile](#) projectile, List< [Zombie](#) > zombies)
- static void [checkExplosionDeaths](#) ([Explosion](#) explosion, List< [Zombie](#) > zombies)

### 5.5.1 Detailed Description

It has all functions connected with checking, whether different entities hit each other.

### 5.5.2 Member Function Documentation

#### 5.5.2.1 checkAttacks()

```
static void CollisionManager.checkAttacks (
    List< Plant > plants,
    List< Zombie > zombies) [static]
```

Checks, whether a zombie attacked any plant

#### Parameters

<i>plants</i>	list of plants alive
<i>zombies</i>	list of zombies alive

#### 5.5.2.2 checkExplosionDeaths()

```
static void CollisionManager.checkExplosionDeaths (
    Explosion explosion,
    List< Zombie > zombies) [static]
```

Checks, whether an explosion hit a zombie.

#### Parameters

<i>explosion</i>	an explosion to check
<i>zombies</i>	list of zombies alive

#### 5.5.2.3 checkProjectileHit()

```
static boolean CollisionManager.checkProjectileHit (
    Projectile projectile,
    List< Zombie > zombies) [static]
```

Checks, whether a projectile hit a zombie.

## Parameters

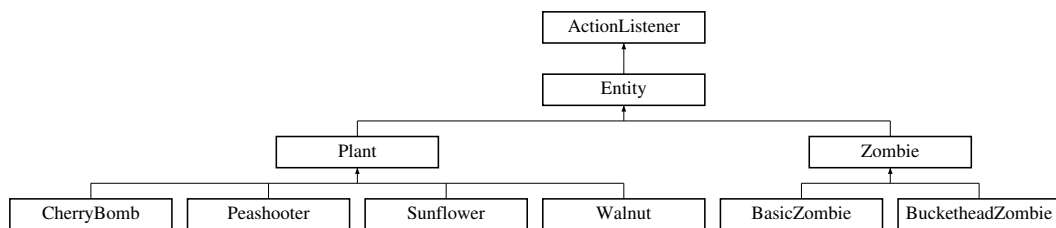
<i>projectile</i>	a projectile to check
<i>zombies</i>	list of zombies alive

## Returns

if the projectile hit anything

## 5.6 Entity Class Reference

Inheritance diagram for Entity:



## Public Member Functions

- Entity (int *x*, int *y*)
- abstract int *getHealth* ()
- abstract void *paintComponent* (Graphics *g*)
- abstract void *takeDamage* (int *damage*)
- abstract Rectangle *getBounds* ()
- abstract int *getType* ()
- abstract void *actionPerformed* (ActionEvent *e*)

## Protected Attributes

- int *x*
- int *y*
- int *health*
- Rectangle *imageBounds*

## Static Protected Attributes

- static Timer *theTimer* = Panel.theTimer

### 5.6.1 Detailed Description

Base of almost all painted objects.

### 5.6.2 Constructor & Destructor Documentation

#### 5.6.2.1 Entity()

```
Entity.Entity (
    int x,
    int y)
```

Constructor

## Parameters

<i>x</i>	x coordinate
<i>y</i>	y coordinate

## 5.6.3 Member Function Documentation

### 5.6.3.1 actionPerformed()

```
abstract void Entity.actionPerformed (  
    ActionEvent e) [abstract]
```

Every [Panel#theTimer](#) tick, this function is called. Used for updating position, health, image on screen etc.

## Parameters

<i>e</i>	tick of <a href="#">Panel#theTimer</a>
----------	--

Reimplemented in [BasicZombie](#), [BucketheadZombie](#), [CherryBomb](#), [Peashooter](#), [Sunflower](#), and [Walnut](#).

### 5.6.3.2 getBounds()

```
abstract Rectangle Entity.getBounds () [abstract]
```

Returns the entity's hitbox

## Returns

entity's hitbox

Reimplemented in [BasicZombie](#), [BucketheadZombie](#), [CherryBomb](#), [Peashooter](#), [Sunflower](#), and [Walnut](#).

### 5.6.3.3 getHealth()

```
abstract int Entity.getHealth () [abstract]
```

Returns the amount of health of an entity.

## Returns

health amount

Reimplemented in [BasicZombie](#), [BucketheadZombie](#), [CherryBomb](#), [Peashooter](#), [Sunflower](#), and [Walnut](#).



#### 5.6.3.4 getType()

```
abstract int Entity.getType () [abstract]
```

Returns the type value defined by each entity.

##### Returns

type of entity

Reimplemented in [BasicZombie](#), [BucketheadZombie](#), [CherryBomb](#), [Peashooter](#), [Sunflower](#), [Walnut](#), and [Zombie](#).

#### 5.6.3.5 paintComponent()

```
abstract void Entity.paintComponent (  
    Graphics g) [abstract]
```

Paints the entity on screen.

##### Parameters

<i>g</i>	required to paint on screen
----------	-----------------------------

Reimplemented in [BasicZombie](#), [BucketheadZombie](#), [CherryBomb](#), [Peashooter](#), [Sunflower](#), and [Walnut](#).

#### 5.6.3.6 takeDamage()

```
abstract void Entity.takeDamage (  
    int damage) [abstract]
```

Remove a specified amount of health from an entity.

##### Parameters

<i>damage</i>	amount of damage dealt
---------------	------------------------

Reimplemented in [BasicZombie](#), [BucketheadZombie](#), [CherryBomb](#), [Peashooter](#), [Sunflower](#), and [Walnut](#).

### 5.6.4 Member Data Documentation

#### 5.6.4.1 health

```
int Entity.health [protected]
```

#### 5.6.4.2 imageBounds

```
Rectangle Entity.imageBounds [protected]
```

a standardized size of entity's hitbox

#### 5.6.4.3 theTimer

```
Timer Entity.theTimer = Panel.theTimer [static], [protected]
```

Copy of the base timer, located in [Panel#theTimer](#).

#### 5.6.4.4 x

```
int Entity.x [protected]
```

identifying values of an entity

#### 5.6.4.5 y

```
int Entity.y [protected]
```

## 5.7 Explosion Class Reference

### Public Member Functions

- [Explosion](#) (int x, int y)
- void [paintComponent](#) (Graphics g)
- int [getDamage](#) ()
- Rectangle [getBounds](#) ()

### Public Attributes

- int [x](#)
- int [y](#)

### 5.7.1 Detailed Description

An object created by the [CherryBomb](#), explodes on creation, dealing damage in 3x3 radius.

### 5.7.2 Constructor & Destructor Documentation

#### 5.7.2.1 Explosion()

```
Explosion.Explosion (  
    int x,  
    int y)
```

Constructor

## Parameters

<i>x</i>	x coordinate
<i>y</i>	y coordinate

## 5.7.3 Member Function Documentation

### 5.7.3.1 getBounds()

```
Rectangle Explosion.getBounds ()
```

Returns the explosion's hitbox

## Returns

explosion's hitbox

### 5.7.3.2 getDamage()

```
int Explosion.getDamage ()
```

Returns the damage of the explosion.

## Returns

damage of the explosion

### 5.7.3.3 paintComponent()

```
void Explosion.paintComponent (  
    Graphics g)
```

Paints the entity on screen.

## Parameters

<i>g</i>	required to paint on screen
----------	-----------------------------

## 5.7.4 Member Data Documentation

### 5.7.4.1 x

```
int Explosion.x
```

Coordinates

### 5.7.4.2 y

```
int Explosion.y
```

## 5.8 Main Class Reference

### Static Public Member Functions

- static void [main](#) (String[] args)

### 5.8.1 Detailed Description

[Main](#) function. Starts the simulation by creating the [Board](#).

### 5.8.2 Member Function Documentation

#### 5.8.2.1 main()

```
static void Main.main (  
    String[] args) [static]
```

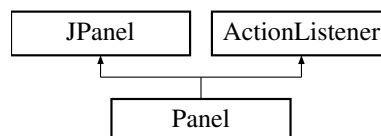
[Main](#) function, used to launch the program.

#### Parameters

<i>args</i>	arguments
-------------	-----------

## 5.9 Panel Class Reference

Inheritance diagram for Panel:



### Public Member Functions

- [Panel](#) ()
- void [gameStart](#) ()
- int [counterUpdater](#) (int i)
- void [spawnRandomPlant](#) ()
- void [spawnRandomZombie](#) ()
- void [actionPerformed](#) (ActionEvent e)
- void [paintComponent](#) (Graphics g)

### Static Public Attributes

- static List< [Plant](#) > [Plants](#) = new ArrayList<>()
- static List< [Zombie](#) > [Zombies](#) = new ArrayList<>()
- static List< List< int[] > > [SpawnSquares](#) = new ArrayList<>()
- static [ResourceManager](#) [resourceManager](#)
- static Timer [theTimer](#)
- static int [DELAY](#) = 50
- static int [START\\_ZOMBIE\\_AMOUNT](#) = 10
- static int [PLANT\\_SPAWN\\_INTERVAL](#) = 2
- static int [ZOMBIE\\_SPAWN\\_INTERVAL](#) = 16
- static int [START\\_SUN\\_POINTS](#) = 900
- static int [BASIC\\_ZOMBIE\\_SPAWN\\_CHANCE](#) = 70
- static int [BUCKETHEAD\\_ZOMBIE\\_SPAWN\\_CHANCE](#) = 30
- static int [SUNFLOWER\\_SPAWN\\_CHANCE](#) = 45
- static int [PEASHOOTER\\_SPAWN\\_CHANCE](#) = 25
- static int [CHERRY\\_BOMB\\_SPAWN\\_CHANCE](#) = 10
- static int [WALNUT\\_SPAWN\\_CHANCE](#) = 20
- static final int [SQUARE\\_SIZE](#) = 100
- static final int [ROWS](#) = 5
- static final int [PLANT\\_COLUMNS](#) = 5
- static final int [ZOMBIE\\_COLUMNS](#) = 4
- static final int [COLUMNS](#) = [PLANT\\_COLUMNS](#) + [ZOMBIE\\_COLUMNS](#)
- static String[] [strings](#)

### 5.9.1 Detailed Description

This class controls everything, that is happening within the simulation. As a JPanel, it draws all the buttons, the agents using [Panel#paintComponent \(Graphics\)](#) and all the counters [Panel#counterUpdater \(int\)](#). It also controls the main [Panel#resourceManager](#) and a Swing timer: [Panel#theTimer](#) by re/starting the simulation using [Panel#gameStart \(\)](#) and updates every tick using [Panel#actionPerformed \(ActionEvent\)](#). It is also responsible for spawning zombies with [Panel#spawnRandomZombie \(\)](#) and plants with [Panel#spawnRandomPlant \(\)](#).

### 5.9.2 Constructor & Destructor Documentation

#### 5.9.2.1 Panel()

```
Panel.Panel ()
```

Creates the main panel of simulation. Adds buttons, introduction and counters.

### 5.9.3 Member Function Documentation

#### 5.9.3.1 actionPerformed()

```
void Panel.actionPerformed (
   (ActionEvent e)
```

Checks, whether [Panel#toSpawnSelector](#), [Panel#toSettingsChange](#), [Panel#startSimulation](#) and [Panel#pauseSimulation](#) buttons are pressed. When starting the simulation, it checks if [Panel#SpawnSquares](#) is empty. It is also responsible for controlling the simulation logic: it is calling [Panel#spawnRandomPlant \(\)](#) and [Panel#spawnRandomZombie \(\)](#) every [Panel#PLANT\\_SPAWN\\_INTERVAL](#) and [Panel#ZOMBIE\\_SPAWN\\_INTERVAL](#) respectively, it is checking, whether any [Zombie](#) reached the left border or if all [Zombies](#) are eliminated. It also checks attacking [Zombies](#) using [CollisionManager#checkAttacks \(List, L](#). It is also updating the counters.

## Parameters

<i>e</i>	the event to be processed
----------	---------------------------

**5.9.3.2 counterUpdater()**

```
int Panel.counterUpdater (
    int i)
```

Calculates value of every counter in `Panel#strings` array, which is later updated in `actionPerformed(ActionEvent)`.

## Parameters

<i>i</i>	counter to update
----------	-------------------

## Returns

value of updated counter

**5.9.3.3 gameStart()**

```
void Panel.gameStart ()
```

Re/starting the simulation by clearing everything from the previous one, like clearing `Panel#Plants` and `Panel#Zombies` lists, hiding the introduction, spending all Sun Points using `ResourceManager#spendSunPoints(int)`, resetting `Panel#plantSpawnCycle` and `Panel#zombieSpawnCycle`, spawning the right amount of `Plants` and `Zombies` and starting `Panel#theTimer`.

**5.9.3.4 paintComponent()**

```
void Panel.paintComponent (
    Graphics g)
```

Paints and removes dead plants/zombies by checking the `Panel#Plants` and `Panel#Zombies` lists.

## Parameters

<i>g</i>	necessary to paint on the JPanel.
----------	-----------------------------------

**5.9.3.5 spawnRandomPlant()**

```
void Panel.spawnRandomPlant ()
```

Spawns a random `Plant` based on the percentages in `Panel#SUNFLOWER_SPAWN_CHANCE`, `Panel#PEASHOOTER_SPAWN_CHANCE`, `Panel#CHERRY_BOMB_SPAWN_CHANCE` and `Panel#WALNUT_SPAWN_CHANCE`. It checks the ability to spawn a `Plant` by checking its cost using the `ResourceManager.spendSunPoints(int)`. The location is determined by checking the `Panel#SpawnSquares` list.

### 5.9.3.6 spawnRandomZombie()

```
void Panel.spawnRandomZombie ()
```

Spawns a random [Zombie](#) based on the percentages in [Panel#BASIC\\_ZOMBIE\\_SPAWN\\_CHANCE](#) and [Panel#BUCKETHEAD\\_ZOMBIE\\_SPAWN\\_CHANCE](#).

## 5.9.4 Member Data Documentation

### 5.9.4.1 BASIC\_ZOMBIE\_SPAWN\_CHANCE

```
int Panel.BASIC_ZOMBIE_SPAWN_CHANCE = 70 [static]
```

Percentage of chance to spawn a [BasicZombie](#).

### 5.9.4.2 BUCKETHEAD\_ZOMBIE\_SPAWN\_CHANCE

```
int Panel.BUCKETHEAD_ZOMBIE_SPAWN_CHANCE = 30 [static]
```

Percentage of chance to spawn a [BucketheadZombie](#).

### 5.9.4.3 CHERRY\_BOMB\_SPAWN\_CHANCE

```
int Panel.CHERRY_BOMB_SPAWN_CHANCE = 10 [static]
```

Percentage of chance to spawn a [CherryBomb](#).

### 5.9.4.4 COLUMNS

```
final int Panel.COLUMNS = PLANT_COLUMNS + ZOMBIE_COLUMNS [static]
```

Amount of horizontal squares.

### 5.9.4.5 DELAY

```
int Panel.DELAY = 50 [static]
```

Tick value

### 5.9.4.6 PEASHOOTER\_SPAWN\_CHANCE

```
int Panel.PEASHOOTER_SPAWN_CHANCE = 25 [static]
```

Percentage of chance to spawn a [Peashooter](#).

#### 5.9.4.7 PLANT\_COLUMNS

```
final int Panel.PLANT_COLUMNS = 5 [static]
```

Amount of horizontal squares for [Plant](#) spawning.

#### 5.9.4.8 PLANT\_SPAWN\_INTERVAL

```
int Panel.PLANT_SPAWN_INTERVAL = 2 [static]
```

Amount of ticks till [Plant](#) spawn.

#### 5.9.4.9 Plants

```
List<Plant> Panel.Plants = new ArrayList<>() [static]
```

List of all alive [Plants](#).

#### 5.9.4.10 resourceManager

```
ResourceManager Panel.resourceManager [static]
```

Main instance of [ResourceManager](#).

#### 5.9.4.11 ROWS

```
final int Panel.ROWS = 5 [static]
```

Amount of vertical squares.

#### 5.9.4.12 SpawnSquares

```
List<List<int[]> > Panel.SpawnSquares = new ArrayList<>() [static]
```

This is where all possible squares to spawn [Plants](#) are stored.

#### 5.9.4.13 SQUARE\_SIZE

```
final int Panel.SQUARE_SIZE = 100 [static]
```

Size of one square.

#### 5.9.4.14 START\_SUN\_POINTS

```
int Panel.START_SUN_POINTS = 900 [static]
```

Starting amount of Sun Points in [ResourceManager](#).



#### 5.9.4.15 START\_ZOMBIE\_AMOUNT

```
int Panel.START_ZOMBIE_AMOUNT = 10 [static]
```

Amount of Zombies in beginning of simulation.

#### 5.9.4.16 strings

```
String [] Panel.strings [static]
```

**Initial value:**

```
= {"Amount of Basic Zombies", "Amount of Buckethead Zombies", "Amount of Sunflowers", "Amount of  
Peashooters", "Amount of Cherry Bombs",  
"Amount of Walnuts", "Amount of Peas", "Amount of Sun Points", "Time Elapsed"}
```

Keeps the String value of every counter.

#### 5.9.4.17 SUNFLOWER\_SPAWN\_CHANCE

```
int Panel.SUNFLOWER_SPAWN_CHANCE = 45 [static]
```

Percentage of chance to spawn a [Sunflower](#).

#### 5.9.4.18 theTimer

```
Timer Panel.theTimer [static]
```

[Main](#) instance of Timer.

#### 5.9.4.19 WALNUT\_SPAWN\_CHANCE

```
int Panel.WALNUT_SPAWN_CHANCE = 20 [static]
```

Percentage of chance to spawn a [Walnut](#).

#### 5.9.4.20 ZOMBIE\_COLUMNS

```
final int Panel.ZOMBIE_COLUMNS = 4 [static]
```

Amount of horizontal squares without [Plant](#) spawning.

#### 5.9.4.21 ZOMBIE\_SPAWN\_INTERVAL

```
int Panel.ZOMBIE_SPAWN_INTERVAL = 16 [static]
```

Amount of ticks till [Zombie](#) spawn.

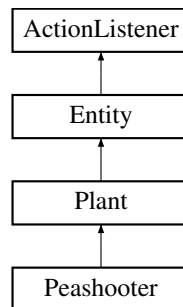
#### 5.9.4.22 Zombies

```
List<Zombie> Panel.Zombies = new ArrayList<>() [static]
```

List of all alive [Zombies](#).

## 5.10 Peashooter Class Reference

Inheritance diagram for Peashooter:



### Public Member Functions

- [Peashooter](#) (int *x*, int *y*)
- int [getType](#) ()
- void [paintComponent](#) (Graphics *g*)
- void [takeDamage](#) (int *damage*)
- void [actionPerformed](#) (ActionEvent *e*)
- void [shoot](#) ()
- int [getHealth](#) ()
- Rectangle [getBounds](#) ()

### Public Member Functions inherited from [Plant](#)

- [Plant](#) (int *x*, int *y*)

### Public Member Functions inherited from [Entity](#)

- [Entity](#) (int *x*, int *y*)

### Static Public Member Functions

- static int [getCost](#) ()

### Public Attributes

- List< [Projectile](#) > [Projectiles](#)
- List< [Zombie](#) > [Zombies](#)

## Additional Inherited Members

### Protected Attributes inherited from [Entity](#)

- int [x](#)
- int [y](#)
- int [health](#)
- Rectangle [imageBounds](#)

### Static Protected Attributes inherited from [Entity](#)

- static Timer [theTimer](#) = Panel.theTimer

## 5.10.1 Detailed Description

A plant, that shoots [Projectiles](#) every certain amount of time, that deal damage on impact.

## 5.10.2 Constructor & Destructor Documentation

### 5.10.2.1 Peashooter()

```
Peashooter.Peashooter (
    int x,
    int y)
```

Constructor of a plant

#### Parameters

<i>x</i>	x coordinate
<i>y</i>	y coordinate

## 5.10.3 Member Function Documentation

### 5.10.3.1 actionPerformed()

```
void Peashooter.actionPerformed (
   (ActionEvent e)
```

Every [Panel#theTimer](#) tick, this function is called. Used for updating position, health, image on screen etc.

#### Parameters

<i>e</i>	tick of <a href="#">Panel#theTimer</a>
----------	--

Reimplemented from [Entity](#).

### 5.10.3.2 getBounds()

```
Rectangle Peashooter.getBounds ()
```

Returns the entity's hitbox

#### Returns

entity's hitbox

Reimplemented from [Entity](#).

### 5.10.3.3 getCost()

```
static int Peashooter.getCost () [static]
```

Returns cost of the plant in Sun Points.

#### Returns

cost of the plant in Sun Points

### 5.10.3.4 getHealth()

```
int Peashooter.getHealth ()
```

Returns the amount of health of an entity.

#### Returns

health amount

Reimplemented from [Entity](#).

### 5.10.3.5 getType()

```
int Peashooter.getType ()
```

Returns the type value defined by each entity.

#### Returns

type of entity

Reimplemented from [Entity](#).

### 5.10.3.6 paintComponent()

```
void Peashooter.paintComponent (  
    Graphics g)
```

Paints the entity on screen.

## Parameters

<i>g</i>	required to paint on screen
----------	-----------------------------

Reimplemented from [Entity](#).

### 5.10.3.7 shoot()

```
void Peashooter.shoot ()
```

Shoot a [Projectile](#) by creating a new instance of it in the [Peashooter#Projectiles](#) list.

### 5.10.3.8 takeDamage()

```
void Peashooter.takeDamage (  
    int damage)
```

Remove a specified amount of health from an entity.

## Parameters

<i>damage</i>	amount of damage dealt
---------------	------------------------

Reimplemented from [Entity](#).

## 5.10.4 Member Data Documentation

### 5.10.4.1 Projectiles

```
List<Projectile> Peashooter.Projectiles
```

List of all of this [Peashooter](#)'s [Projectiles](#).

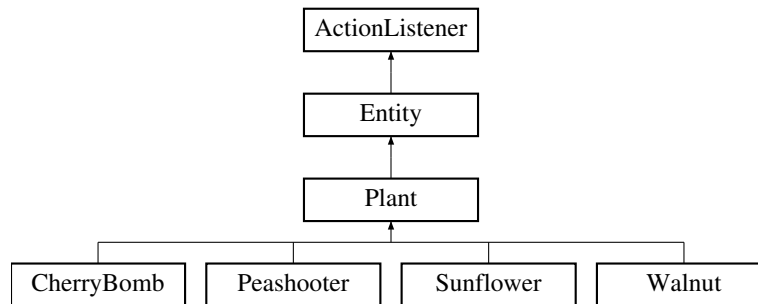
### 5.10.4.2 Zombies

```
List<Zombie> Peashooter.Zombies
```

Copy of [Panel#Zombies](#) used in [CollisionManager#checkExplosionDeaths](#).

## 5.11 Plant Class Reference

Inheritance diagram for Plant:



### Public Member Functions

- [Plant](#) (int *x*, int *y*)

### Public Member Functions inherited from [Entity](#)

- [Entity](#) (int *x*, int *y*)
- abstract int [getHealth](#) ()
- abstract void [paintComponent](#) (Graphics *g*)
- abstract void [takeDamage](#) (int *damage*)
- abstract Rectangle [getBounds](#) ()
- abstract int [getType](#) ()
- abstract void [actionPerformed](#) (ActionEvent *e*)

### Additional Inherited Members

### Protected Attributes inherited from [Entity](#)

- int *x*
- int *y*
- int [health](#)
- Rectangle [imageBounds](#)

### Static Protected Attributes inherited from [Entity](#)

- static Timer [theTimer](#) = Panel.theTimer

#### 5.11.1 Detailed Description

A base of all plant-type entities.

#### 5.11.2 Constructor & Destructor Documentation

##### 5.11.2.1 [Plant\(\)](#)

```

Plant.Plant (
    int x,
    int y)

```

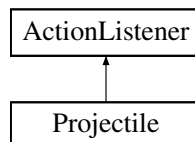
Constructor of a plant

## Parameters

<i>x</i>	x coordinate
<i>y</i>	y coordinate

## 5.12 Projectile Class Reference

Inheritance diagram for Projectile:



### Public Member Functions

- `Projectile` (int *x*, int *y*)
- void `paintComponent` (Graphics *g*)
- int `getDamage` ()
- int `getWidth` ()
- Rectangle `getBounds` ()
- void `actionPerformed` (ActionEvent *e*)

### Public Attributes

- int *x*
- int *y*

### Static Public Attributes

- static Timer `theTimer` = Panel.theTimer

### 5.12.1 Detailed Description

An object created by the `Peashooter`, moves towards zombies and deals damage when it hits one.

### 5.12.2 Constructor & Destructor Documentation

#### 5.12.2.1 Projectile()

```
Projectile.Projectile (  
    int x,  
    int y)
```

Constructor

**Parameters**

<i>x</i>	x coordinate
<i>y</i>	y coordinate

### 5.12.3 Member Function Documentation

#### 5.12.3.1 actionPerformed()

```
void Projectile.actionPerformed (  
    ActionEvent e)
```

Every [Panel#theTimer](#) tick, this function is called. Used for updating position of the projectile.

**Parameters**

<i>e</i>	tick of <a href="#">Panel#theTimer</a>
----------	--

#### 5.12.3.2 getBounds()

```
Rectangle Projectile.getBounds ()
```

Returns a projectile's hitbox

**Returns**

projectile's hitbox

#### 5.12.3.3 getDamage()

```
int Projectile.getDamage ()
```

Returns the damage of a projectile.

**Returns**

damage of a projectile

#### 5.12.3.4 getWidth()

```
int Projectile.getWidth ()
```

Returns a projectile's width

**Returns**

projectile's width

#### 5.12.3.5 paintComponent()

```
void Projectile.paintComponent (  
    Graphics g)
```

Paints the entity on screen.



## Parameters

<code>g</code>	required to paint on screen
----------------	-----------------------------

## 5.12.4 Member Data Documentation

### 5.12.4.1 theTimer

`Timer Projectile.theTimer = Panel.theTimer [static]`

Copy of the base timer, located in [Panel#theTimer](#).

### 5.12.4.2 x

`int Projectile.x`

coordinates

### 5.12.4.3 y

`int Projectile.y`

## 5.13 ResourceManager Class Reference

### Public Member Functions

- [ResourceManager](#) ()
- void [addSunPoints](#) (int points)
- int [getSunPoints](#) ()
- boolean [spendSunPoints](#) (int points)

### 5.13.1 Detailed Description

Stores the Sun Points amount and removes/adds them accordingly.

### 5.13.2 Constructor & Destructor Documentation

#### 5.13.2.1 ResourceManager()

`ResourceManager.ResourceManager ()`

Reset Sun Points.

### 5.13.3 Member Function Documentation

#### 5.13.3.1 addSunPoints()

`void ResourceManager.addSunPoints (`  
     `int points)`

Adds a specified amount of points.

**Parameters**

<i>points</i>	amount of points to add.
---------------	--------------------------

**5.13.3.2 getSunPoints()**

```
int ResourceManager.getSunPoints ()
```

Returns the amount of Sun Points available.

**Returns**

amount of Sun Points

**5.13.3.3 spendSunPoints()**

```
boolean ResourceManager.spendSunPoints (  
    int points)
```

Checks if after spending specified points amount, the remaining value isn't negative.

**Parameters**

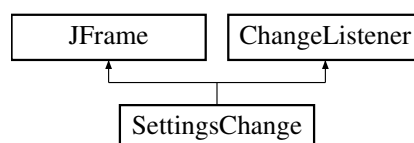
<i>points</i>	amount of Sun Points to spend.
---------------	--------------------------------

**Returns**

if possible to spend Sun Points

## 5.14 SettingsChange Class Reference

Inheritance diagram for SettingsChange:

**Public Member Functions**

- [SettingsChange](#) ()
- void [stateChanged](#) (ChangeEvent e)

**Static Public Attributes**

- static int [DELAY](#) = Panel.DELAY
- static int [START\\_ZOMBIE\\_AMOUNT](#) = Panel.START\_ZOMBIE\_AMOUNT
- static int [PLANT\\_SPAWN\\_INTERVAL](#) = Panel.PLANT\_SPAWN\_INTERVAL
- static int [ZOMBIE\\_SPAWN\\_INTERVAL](#) = Panel.ZOMBIE\_SPAWN\_INTERVAL
- static int [START\\_SUN\\_POINTS](#) = Panel.START\_SUN\_POINTS
- static int [BASIC\\_ZOMBIE\\_SPAWN\\_CHANCE](#) = Panel.BASIC\_ZOMBIE\_SPAWN\_CHANCE
- static int [BUCKETHEAD\\_ZOMBIE\\_SPAWN\\_CHANCE](#) = Panel.BUCKETHEAD\_ZOMBIE\_SPAWN\_CHANCE
- static int [SUNFLOWER\\_SPAWN\\_CHANCE](#) = Panel.SUNFLOWER\_SPAWN\_CHANCE
- static int [PEASHOOTER\\_SPAWN\\_CHANCE](#) = Panel.PEASHOOTER\_SPAWN\_CHANCE
- static int [CHERRY\\_BOMB\\_SPAWN\\_CHANCE](#) = Panel.CHERRY\_BOMB\_SPAWN\_CHANCE
- static int [WALNUT\\_SPAWN\\_CHANCE](#) = Panel.WALNUT\_SPAWN\_CHANCE
- static int[] [values](#)
- static String[] [strings](#)

**5.14.1 Detailed Description**

A window with sliders to change settings specified in [SettingsChange.strings](#).

**5.14.2 Constructor & Destructor Documentation****5.14.2.1 SettingsChange()**

```
SettingsChange.SettingsChange ()
```

Creates the window and sliders.

**5.14.3 Member Function Documentation****5.14.3.1 stateChanged()**

```
void SettingsChange.stateChanged (
    ChangeEvent e)
```

If a slider is moved, this function is called to change the value of the field based on the slider changed.

**Parameters**

<i>e</i>	slider was moved
----------	------------------

**5.14.4 Member Data Documentation****5.14.4.1 BASIC\_ZOMBIE\_SPAWN\_CHANCE**

```
int SettingsChange.BASIC_ZOMBIE_SPAWN_CHANCE = Panel.BASIC_ZOMBIE_SPAWN_CHANCE [static]
```

Percentage of chance to spawn a [BasicZombie](#).

#### 5.14.4.2 BUCKETHEAD\_ZOMBIE\_SPAWN\_CHANCE

```
int SettingsChange.BUCKETHEAD_ZOMBIE_SPAWN_CHANCE = Panel.BUCKETHEAD_ZOMBIE_SPAWN_CHANCE [static]
```

Percentage of chance to spawn a [BucketheadZombie](#).

#### 5.14.4.3 CHERRY\_BOMB\_SPAWN\_CHANCE

```
int SettingsChange.CHERRY_BOMB_SPAWN_CHANCE = Panel.CHERRY_BOMB_SPAWN_CHANCE [static]
```

Percentage of chance to spawn a [CherryBomb](#).

#### 5.14.4.4 DELAY

```
int SettingsChange.DELAY = Panel.DELAY [static]
```

Tick value

#### 5.14.4.5 PEASHOOTER\_SPAWN\_CHANCE

```
int SettingsChange.PEASHOOTER_SPAWN_CHANCE = Panel.PEASHOOTER_SPAWN_CHANCE [static]
```

Percentage of chance to spawn a [Peashooter](#).

#### 5.14.4.6 PLANT\_SPAWN\_INTERVAL

```
int SettingsChange.PLANT_SPAWN_INTERVAL = Panel.PLANT_SPAWN_INTERVAL [static]
```

Amount of ticks till [Plant](#) spawn.

#### 5.14.4.7 START\_SUN\_POINTS

```
int SettingsChange.START_SUN_POINTS = Panel.START_SUN_POINTS [static]
```

Starting amount of Sun Points in [ResourceManager](#).

#### 5.14.4.8 START\_ZOMBIE\_AMOUNT

```
int SettingsChange.START_ZOMBIE_AMOUNT = Panel.START_ZOMBIE_AMOUNT [static]
```

Amount of Zombies in beginning of simulation.

#### 5.14.4.9 strings

```
String [] SettingsChange.strings [static]
```

##### Initial value:

```
= {"DELAY", "START_ZOMBIE_AMOUNT", "PLANT_SPAWN_INTERVAL", "ZOMBIE_SPAWN_INTERVAL", "START_SUN_POINTS",  
  "BASIC_ZOMBIE_SPAWN_CHANCE",  
  "BUCKETHEAD_ZOMBIE_SPAWN_CHANCE", "SUNFLOWER_SPAWN_CHANCE", "PEASHOOTER_SPAWN_CHANCE",  
  "CHERRY_BOMB_SPAWN_CHANCE", "WALNUT_SPAWN_CHANCE"}
```

A list of all the settings' String names.

#### 5.14.4.10 SUNFLOWER\_SPAWN\_CHANCE

```
int SettingsChange.SUNFLOWER_SPAWN_CHANCE = Panel.SUNFLOWER_SPAWN_CHANCE [static]
```

Percentage of chance to spawn a [Sunflower](#).

#### 5.14.4.11 values

```
int [] SettingsChange.values [static]
```

##### Initial value:

```
= {DELAY, START_ZOMBIE_AMOUNT, PLANT_SPAWN_INTERVAL, ZOMBIE_SPAWN_INTERVAL, START_SUN_POINTS,  
  BASIC_ZOMBIE_SPAWN_CHANCE, BUCKETHEAD_ZOMBIE_SPAWN_CHANCE,  
  SUNFLOWER_SPAWN_CHANCE, PEASHOOTER_SPAWN_CHANCE, CHERRY_BOMB_SPAWN_CHANCE, WALNUT_SPAWN_CHANCE}
```

A list of all the settings' values.

#### 5.14.4.12 WALNUT\_SPAWN\_CHANCE

```
int SettingsChange.WALNUT_SPAWN_CHANCE = Panel.WALNUT_SPAWN_CHANCE [static]
```

Percentage of chance to spawn a [Walnut](#).

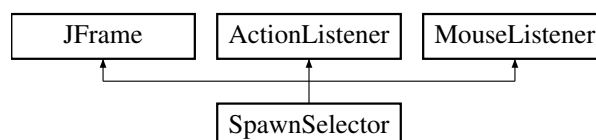
#### 5.14.4.13 ZOMBIE\_SPAWN\_INTERVAL

```
int SettingsChange.ZOMBIE_SPAWN_INTERVAL = Panel.ZOMBIE_SPAWN_INTERVAL [static]
```

Amount of ticks till [Zombie](#) spawn.

## 5.15 SpawnSelector Class Reference

Inheritance diagram for SpawnSelector:



## Public Member Functions

- [SpawnSelector](#) ()
- void [paint](#) (Graphics g)
- void [actionPerformed](#) (ActionEvent e)
- void [mouseClicked](#) (MouseEvent e)
- void [mousePressed](#) (MouseEvent e)
- void [mouseReleased](#) (MouseEvent e)
- void [mouseEntered](#) (MouseEvent e)
- void [mouseExited](#) (MouseEvent e)

## Static Public Member Functions

- static int[] [findArrayIndex](#) (List< List< int[]> > listOfLists, int[] targetArray)

## Public Attributes

- List< List< int[]> > [SpawnSquares](#) = Panel.SpawnSquares

## Static Public Attributes

- static Timer [theTimer](#) = Panel.theTimer

### 5.15.1 Detailed Description

A window, where a user selects the squares, where certain plants can spawn. The coordinate of square clicked is found using `SpawnSelector#mouseClicked(java.awt.event.MouseEvent)` and the squares are stored in `SpawnSelector#SpawnSquares`. To choose a different plant, a `javax.swing.JRadioButton` can be clicked, which is managed by `SpawnSelector#actionPerformed(java.awt.e`

### 5.15.2 Constructor & Destructor Documentation

#### 5.15.2.1 SpawnSelector()

```
SpawnSelector.SpawnSelector ()
```

Creates the window and buttons and puts them in a `javax.swing.ButtonGroup`.

### 5.15.3 Member Function Documentation

#### 5.15.3.1 actionPerformed()

```
void SpawnSelector.actionPerformed (
   (ActionEvent e)
```

Checks, which button is clicked and chooses the selected button accordingly.

## Parameters

<i>e</i>	the event to be processed
----------	---------------------------

**5.15.3.2 findArrayIndex()**

```
static int[] SpawnSelector.findArrayIndex (  
    List< List< int[] > > listOfLists,  
    int[] targetArray) [static]
```

Helper function to find an array within a list of lists. Used to find clicked square and putting them in the correct place in `SpawnSelector#mouseClicked(java.awt.event.MouseEvent)`.

## Parameters

<i>listOfLists</i>	a list of lists
<i>targetArray</i>	array to find

## Returns

index of `targetArray` or `int[]{-1, -1}` if not found

**5.15.3.3 mouseClicked()**

```
void SpawnSelector.mouseClicked (  
    MouseEvent e)
```

Function to find the clicked square and put them in the [SpawnSelector#SpawnSquares](#) list.

## Parameters

<i>e</i>	the event to be processed
----------	---------------------------

**5.15.3.4 mouseEntered()**

```
void SpawnSelector.mouseEntered (  
    MouseEvent e)
```

**5.15.3.5 mouseExited()**

```
void SpawnSelector.mouseExited (  
    MouseEvent e)
```

**5.15.3.6 mousePressed()**

```
void SpawnSelector.mousePressed (  
    MouseEvent e)
```

### 5.15.3.7 mouseReleased()

```
void SpawnSelector.mouseReleased (
    MouseEvent e)
```

### 5.15.3.8 paint()

```
void SpawnSelector.paint (
    Graphics g)
```

Paints all the squares to choose from painted black, and currently chosen ones painted blue

#### Parameters

<i>g</i>	the specified Graphics window
----------	-------------------------------

## 5.15.4 Member Data Documentation

### 5.15.4.1 SpawnSquares

```
List<List<int[]>> SpawnSelector.SpawnSquares = Panel.SpawnSquares
```

List of all squares, where spawning [Plants](#) is possible.

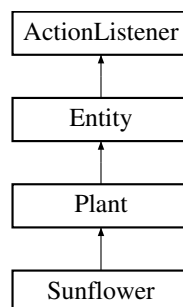
### 5.15.4.2 theTimer

```
Timer SpawnSelector.theTimer = Panel.theTimer [static]
```

A copy of the [Panel#theTimer](#).

## 5.16 Sunflower Class Reference

Inheritance diagram for Sunflower:





### Public Member Functions

- [Sunflower](#) (int *x*, int *y*)
- int [getType](#) ()
- void [paintComponent](#) (Graphics *g*)
- void [takeDamage](#) (int *damage*)
- int [getHealth](#) ()
- void [actionPerformed](#) (ActionEvent *e*)
- Rectangle [getBounds](#) ()

### Public Member Functions inherited from [Plant](#)

- [Plant](#) (int *x*, int *y*)

### Public Member Functions inherited from [Entity](#)

- [Entity](#) (int *x*, int *y*)

### Static Public Member Functions

- static int [getCost](#) ()

### Public Attributes

- boolean [alive](#)

### Static Public Attributes

- static [ResourceManager](#) [resourceManager](#)

### Protected Attributes

- int [produceCycle](#)

### Protected Attributes inherited from [Entity](#)

- int [x](#)
- int [y](#)
- int [health](#)
- Rectangle [imageBounds](#)

### Additional Inherited Members

### Static Protected Attributes inherited from [Entity](#)

- static Timer [theTimer](#) = Panel.theTimer

### 5.16.1 Detailed Description

A plant, that generates Sun Points.

### 5.16.2 Constructor & Destructor Documentation

#### 5.16.2.1 Sunflower()

```
Sunflower.Sunflower (  
    int x,  
    int y)
```

Constructor of a plant

##### Parameters

<i>x</i>	x coordinate
<i>y</i>	y coordinate

### 5.16.3 Member Function Documentation

#### 5.16.3.1 actionPerformed()

```
void Sunflower.actionPerformed (  
   (ActionEvent) e)
```

Every [Panel#theTimer](#) tick, this function is called. Used for updating position, health, image on screen etc.

##### Parameters

<i>e</i>	tick of <a href="#">Panel#theTimer</a>
----------	--

Reimplemented from [Entity](#).

#### 5.16.3.2 getBounds()

```
Rectangle Sunflower.getBounds ()
```

Returns the entity's hitbox

##### Returns

entity's hitbox

Reimplemented from [Entity](#).

### 5.16.3.3 `getCost()`

```
static int Sunflower.getCost () [static]
```

Returns cost of the plant in Sun Points.

#### Returns

cost of the plant in Sun Points

### 5.16.3.4 `getHealth()`

```
int Sunflower.getHealth ()
```

Returns the amount of health of an entity.

#### Returns

health amount

Reimplemented from [Entity](#).

### 5.16.3.5 `getType()`

```
int Sunflower.getType ()
```

Returns the type value defined by each entity.

#### Returns

type of entity

Reimplemented from [Entity](#).

### 5.16.3.6 `paintComponent()`

```
void Sunflower.paintComponent (  
    Graphics g)
```

Paints the entity on screen.

#### Parameters

<i>g</i>	required to paint on screen
----------	-----------------------------

Reimplemented from [Entity](#).

### 5.16.3.7 `takeDamage()`

```
void Sunflower.takeDamage (  
    int damage)
```

Remove a specified amount of health from an entity.

#### Parameters

<i>damage</i>	amount of damage dealt
---------------	------------------------

Reimplemented from [Entity](#).

### 5.16.4 Member Data Documentation

#### 5.16.4.1 alive

```
boolean Sunflower.alive
```

Additional check to make sure [Sunflower](#) is dead or alive.

#### 5.16.4.2 produceCycle

```
int Sunflower.produceCycle [protected]
```

Amount of ticks to add a certain amount of Sun Points.

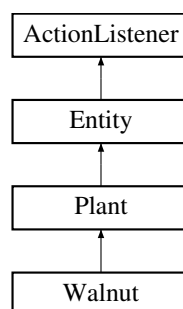
#### 5.16.4.3 resourceManager

```
ResourceManager Sunflower.resourceManager [static]
```

Copy of [Panel#resourceManager](#), used to add Sun Points.

## 5.17 Walnut Class Reference

Inheritance diagram for Walnut:



#### Public Member Functions

- [Walnut](#) (int *x*, int *y*)
- int [getType](#) ()
- void [paintComponent](#) (Graphics *g*)
- void [takeDamage](#) (int *damage*)
- int [getHealth](#) ()
- void [actionPerformed](#) (ActionEvent *e*)
- Rectangle [getBounds](#) ()

**Public Member Functions inherited from [Plant](#)**

- [Plant](#) (int *x*, int *y*)

**Public Member Functions inherited from [Entity](#)**

- [Entity](#) (int *x*, int *y*)

**Static Public Member Functions**

- static int [getCost](#) ()

**Additional Inherited Members****Protected Attributes inherited from [Entity](#)**

- int *x*
- int *y*
- int *health*
- Rectangle *imageBounds*

**Static Protected Attributes inherited from [Entity](#)**

- static Timer *theTimer* = Panel.theTimer

**5.17.1 Detailed Description**

A plant, that has a very high health.

**5.17.2 Constructor & Destructor Documentation****5.17.2.1 Walnut()**

```
Walnut.Walnut (
    int x,
    int y)
```

Constructor of a plant

**Parameters**

<i>x</i>	x coordinate
<i>y</i>	y coordinate

**5.17.3 Member Function Documentation****5.17.3.1 actionPerformed()**

```
void Walnut.actionPerformed (
    ActionEvent e)
```

Every [Panel#theTimer](#) tick, this function is called. Used for updating position, health, image on screen etc.

**Parameters**

<i>e</i>	tick of <a href="#">Panel#theTimer</a>
----------	--

Reimplemented from [Entity](#).

**5.17.3.2 getBounds()**

```
Rectangle Walnut.getBounds ()
```

Returns the entity's hitbox

**Returns**

entity's hitbox

Reimplemented from [Entity](#).

**5.17.3.3 getCost()**

```
static int Walnut.getCost () [static]
```

Returns cost of the plant in Sun Points.

**Returns**

cost of the plant in Sun Points

**5.17.3.4 getHealth()**

```
int Walnut.getHealth ()
```

Returns the amount of health of an entity.

**Returns**

health amount

Reimplemented from [Entity](#).

**5.17.3.5 getType()**

```
int Walnut.getType ()
```

Returns the type value defined by each entity.

**Returns**

type of entity

Reimplemented from [Entity](#).

**5.17.3.6 paintComponent()**

```
void Walnut.paintComponent (  
    Graphics g)
```

Paints the entity on screen.

## Parameters

<i>g</i>	required to paint on screen
----------	-----------------------------

Reimplemented from [Entity](#).

## 5.17.3.7 takeDamage()

```
void Walnut.takeDamage (
    int damage)
```

Remove a specified amount of health from an entity.

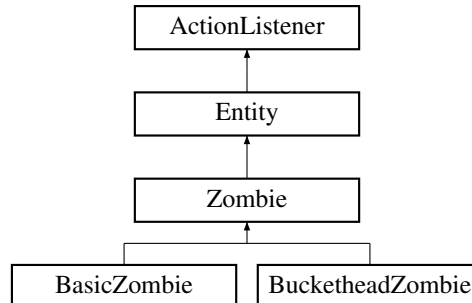
## Parameters

<i>damage</i>	amount of damage dealt
---------------	------------------------

Reimplemented from [Entity](#).

## 5.18 Zombie Class Reference

Inheritance diagram for Zombie:



## Public Member Functions

- [Zombie](#) (int *x*, int *y*)
- abstract int [getAttackDamage](#) ()
- abstract int [getSpeed](#) ()
- abstract int [getType](#) ()

Public Member Functions inherited from [Entity](#)

- [Entity](#) (int *x*, int *y*)
- abstract int [getHealth](#) ()
- abstract void [paintComponent](#) (Graphics *g*)
- abstract void [takeDamage](#) (int *damage*)
- abstract Rectangle [getBounds](#) ()
- abstract void [actionPerformed](#) (ActionEvent *e*)

## Additional Inherited Members

### Protected Attributes inherited from [Entity](#)

- int [x](#)
- int [y](#)
- int [health](#)
- Rectangle [imageBounds](#)

### Static Protected Attributes inherited from [Entity](#)

- static Timer [theTimer](#) = Panel.theTimer

## 5.18.1 Detailed Description

A base of all zombie-type entities.

## 5.18.2 Constructor & Destructor Documentation

### 5.18.2.1 [Zombie\(\)](#)

```
Zombie.Zombie (
    int x,
    int y)
```

Constructor

Parameters

<i>x</i>	x coordinate
<i>y</i>	y coordinate

## 5.18.3 Member Function Documentation

### 5.18.3.1 [getAttackDamage\(\)](#)

```
abstract int Zombie.getAttackDamage () [abstract]
```

Returns the attack damage of a zombie entity.

Returns

attack damage of a zombie entity

Reimplemented in [BasicZombie](#), and [BucketheadZombie](#).



### 5.18.3.2 `getSpeed()`

```
abstract int Zombie.getSpeed () [abstract]
```

Returns the speed of a zombie entity.

#### Returns

speed of a zombie entity

Reimplemented in [BasicZombie](#), and [BucketheadZombie](#).

### 5.18.3.3 `getType()`

```
abstract int Zombie.getType () [abstract]
```

Returns the type value of a zombie entity.

#### Returns

type value of a zombie entity

Reimplemented from [Entity](#).

Reimplemented in [BasicZombie](#), and [BucketheadZombie](#).



# Chapter 6

## File Documentation

### 6.1 README.md File Reference

### 6.2 BasicZombie.java File Reference

#### Classes

- class [BasicZombie](#)

### 6.3 Board.java File Reference

#### Classes

- class [Board](#)

### 6.4 BucketheadZombie.java File Reference

#### Classes

- class [BucketheadZombie](#)

### 6.5 CherryBomb.java File Reference

#### Classes

- class [CherryBomb](#)

## 6.6 CollisionManager.java File Reference

### Classes

- class [CollisionManager](#)

## 6.7 Entity.java File Reference

### Classes

- class [Entity](#)

## 6.8 Explosion.java File Reference

### Classes

- class [Explosion](#)

## 6.9 Main.java File Reference

### Classes

- class [Main](#)

## 6.10 Panel.java File Reference

### Classes

- class [Panel](#)

## 6.11 Peashooter.java File Reference

### Classes

- class [Peashooter](#)

## 6.12 Plant.java File Reference

### Classes

- class [Plant](#)

## 6.13 Projectile.java File Reference

### Classes

- class [Projectile](#)

## 6.14 ResourceManager.java File Reference

### Classes

- class [ResourceManager](#)

## 6.15 SettingsChange.java File Reference

### Classes

- class [SettingsChange](#)

## 6.16 SpawnSelector.java File Reference

### Classes

- class [SpawnSelector](#)

## 6.17 Sunflower.java File Reference

### Classes

- class [Sunflower](#)

## 6.18 Walnut.java File Reference

### Classes

- class [Walnut](#)

## 6.19 Zombie.java File Reference

### Classes

- class [Zombie](#)



# Index

- actionPerformed
  - BasicZombie, [10](#)
  - BucketheadZombie, [14](#)
  - CherryBomb, [18](#)
  - Entity, [22](#)
  - Panel, [27](#)
  - Peashooter, [33](#)
  - Projectile, [38](#)
  - SpawnSelector, [44](#)
  - Sunflower, [48](#)
  - Walnut, [51](#)
- addSunPoints
  - ResourceManager, [39](#)
- alive
  - Sunflower, [50](#)
- BASIC\_ZOMBIE\_SPAWN\_CHANCE
  - Panel, [29](#)
  - SettingsChange, [41](#)
- BasicZombie, [9](#)
  - actionPerformed, [10](#)
  - BasicZombie, [10](#)
  - getAttackDamage, [10](#)
  - getBounds, [11](#)
  - getHealth, [11](#)
  - getSpeed, [11](#)
  - getType, [11](#)
  - paintComponent, [12](#)
  - takeDamage, [12](#)
- BasicZombie.java, [57](#)
- Board, [12](#)
  - Board, [13](#)
- Board.java, [57](#)
- BUCKETHEAD\_ZOMBIE\_SPAWN\_CHANCE
  - Panel, [29](#)
  - SettingsChange, [41](#)
- BucketheadZombie, [13](#)
  - actionPerformed, [14](#)
  - BucketheadZombie, [14](#)
  - getAttackDamage, [15](#)
  - getBounds, [15](#)
  - getHealth, [15](#)
  - getSpeed, [15](#)
  - getType, [15](#)
  - paintComponent, [16](#)
  - takeDamage, [16](#)
- BucketheadZombie.java, [57](#)
- checkAttacks
  - CollisionManager, [20](#)
- checkExplosionDeaths
  - CollisionManager, [20](#)
- checkProjectileHit
  - CollisionManager, [20](#)
- CHERRY\_BOMB\_SPAWN\_CHANCE
  - Panel, [29](#)
  - SettingsChange, [42](#)
- CherryBomb, [16](#)
  - actionPerformed, [18](#)
  - CherryBomb, [17](#)
  - explode, [18](#)
  - getBounds, [18](#)
  - getCost, [18](#)
  - getHealth, [18](#)
  - getType, [19](#)
  - paintComponent, [19](#)
  - takeDamage, [19](#)
  - Zombies, [20](#)
- CherryBomb.java, [57](#)
- CollisionManager, [20](#)
  - checkAttacks, [20](#)
  - checkExplosionDeaths, [20](#)
  - checkProjectileHit, [20](#)
- CollisionManager.java, [58](#)
- COLUMNS
  - Panel, [29](#)
- counterUpdater
  - Panel, [28](#)
- DELAY
  - Panel, [29](#)
  - SettingsChange, [42](#)
- Entity, [21](#)
  - actionPerformed, [22](#)
  - Entity, [21](#)
  - getBounds, [22](#)
  - getHealth, [22](#)
  - getType, [22](#)
  - health, [23](#)
  - imageBounds, [23](#)
  - paintComponent, [23](#)
  - takeDamage, [23](#)
  - theTimer, [23](#)
  - x, [24](#)
  - y, [24](#)
- Entity.java, [58](#)
- explode
  - CherryBomb, [18](#)
- Explosion, [24](#)

- Explosion, [24](#)
- getBounds, [25](#)
- getDamage, [25](#)
- paintComponent, [25](#)
- x, [25](#)
- y, [25](#)
- Explosion.java, [58](#)
- findArrayIndex
  - SpawnSelector, [45](#)
- gameStart
  - Panel, [28](#)
- getAttackDamage
  - BasicZombie, [10](#)
  - BucketheadZombie, [15](#)
  - Zombie, [54](#)
- getBounds
  - BasicZombie, [11](#)
  - BucketheadZombie, [15](#)
  - CherryBomb, [18](#)
  - Entity, [22](#)
  - Explosion, [25](#)
  - Peashooter, [33](#)
  - Projectile, [38](#)
  - Sunflower, [48](#)
  - Walnut, [52](#)
- getCost
  - CherryBomb, [18](#)
  - Peashooter, [34](#)
  - Sunflower, [48](#)
  - Walnut, [52](#)
- getDamage
  - Explosion, [25](#)
  - Projectile, [38](#)
- getHealth
  - BasicZombie, [11](#)
  - BucketheadZombie, [15](#)
  - CherryBomb, [18](#)
  - Entity, [22](#)
  - Peashooter, [34](#)
  - Sunflower, [49](#)
  - Walnut, [52](#)
- getSpeed
  - BasicZombie, [11](#)
  - BucketheadZombie, [15](#)
  - Zombie, [54](#)
- getSunPoints
  - ResourceManager, [40](#)
- getType
  - BasicZombie, [11](#)
  - BucketheadZombie, [15](#)
  - CherryBomb, [19](#)
  - Entity, [22](#)
  - Peashooter, [34](#)
  - Sunflower, [49](#)
  - Walnut, [52](#)
  - Zombie, [55](#)
- getWidth
  - Projectile, [38](#)
- health
  - Entity, [23](#)
- imageBounds
  - Entity, [23](#)
- Main, [26](#)
  - main, [26](#)
- main
  - Main, [26](#)
- Main.java, [58](#)
- mouseClicked
  - SpawnSelector, [45](#)
- mouseEntered
  - SpawnSelector, [45](#)
- mouseExited
  - SpawnSelector, [45](#)
- mousePressed
  - SpawnSelector, [45](#)
- mouseReleased
  - SpawnSelector, [45](#)
- paint
  - SpawnSelector, [46](#)
- paintComponent
  - BasicZombie, [12](#)
  - BucketheadZombie, [16](#)
  - CherryBomb, [19](#)
  - Entity, [23](#)
  - Explosion, [25](#)
  - Panel, [28](#)
  - Peashooter, [34](#)
  - Projectile, [38](#)
  - Sunflower, [49](#)
  - Walnut, [52](#)
- Panel, [26](#)
  - actionPerformed, [27](#)
  - BASIC\_ZOMBIE\_SPAWN\_CHANCE, [29](#)
  - BUCKETHEAD\_ZOMBIE\_SPAWN\_CHANCE, [29](#)
  - CHERRY\_BOMB\_SPAWN\_CHANCE, [29](#)
  - COLUMNS, [29](#)
  - counterUpdater, [28](#)
  - DELAY, [29](#)
  - gameStart, [28](#)
  - paintComponent, [28](#)
  - Panel, [27](#)
  - PEASHOOTER\_SPAWN\_CHANCE, [29](#)
  - PLANT\_COLUMNS, [29](#)
  - PLANT\_SPAWN\_INTERVAL, [30](#)
  - Plants, [30](#)
  - resourceManager, [30](#)
  - ROWS, [30](#)
  - spawnRandomPlant, [28](#)
  - spawnRandomZombie, [28](#)
  - SpawnSquares, [30](#)
  - SQUARE\_SIZE, [30](#)
  - START\_SUN\_POINTS, [30](#)



- START\_ZOMBIE\_AMOUNT, [30](#)
- strings, [31](#)
- SUNFLOWER\_SPAWN\_CHANCE, [31](#)
- theTimer, [31](#)
- WALNUT\_SPAWN\_CHANCE, [31](#)
- ZOMBIE\_COLUMNS, [31](#)
- ZOMBIE\_SPAWN\_INTERVAL, [31](#)
- Zombies, [31](#)
- Panel.java, [58](#)
- Peashooter, [32](#)
  - actionPerformed, [33](#)
  - getBounds, [33](#)
  - getCost, [34](#)
  - getHealth, [34](#)
  - getType, [34](#)
  - paintComponent, [34](#)
  - Peashooter, [33](#)
  - Projectiles, [35](#)
  - shoot, [35](#)
  - takeDamage, [35](#)
  - Zombies, [35](#)
- Peashooter.java, [58](#)
- PEASHOOTER\_SPAWN\_CHANCE
  - Panel, [29](#)
  - SettingsChange, [42](#)
- Plant, [36](#)
  - Plant, [36](#)
- Plant.java, [58](#)
- PLANT\_COLUMNS
  - Panel, [29](#)
- PLANT\_SPAWN\_INTERVAL
  - Panel, [30](#)
  - SettingsChange, [42](#)
- Plants
  - Panel, [30](#)
- Plants vs Zombies simulation, [1](#)
- produceCycle
  - Sunflower, [50](#)
- Projectile, [37](#)
  - actionPerformed, [38](#)
  - getBounds, [38](#)
  - getDamage, [38](#)
  - getWidth, [38](#)
  - paintComponent, [38](#)
  - Projectile, [37](#)
  - theTimer, [39](#)
  - x, [39](#)
  - y, [39](#)
- Projectile.java, [59](#)
- Projectiles
  - Peashooter, [35](#)
- README.md, [57](#)
- ResourceManager, [39](#)
  - addSunPoints, [39](#)
  - getSunPoints, [40](#)
  - ResourceManager, [39](#)
  - spendSunPoints, [40](#)
- resourceManager
  - Panel, [30](#)
  - Sunflower, [50](#)
- ResourceManager.java, [59](#)
- ROWS
  - Panel, [30](#)
- SettingsChange, [40](#)
  - BASIC\_ZOMBIE\_SPAWN\_CHANCE, [41](#)
  - BUCKETHEAD\_ZOMBIE\_SPAWN\_CHANCE, [41](#)
  - CHERRY\_BOMB\_SPAWN\_CHANCE, [42](#)
  - DELAY, [42](#)
  - PEASHOOTER\_SPAWN\_CHANCE, [42](#)
  - PLANT\_SPAWN\_INTERVAL, [42](#)
  - SettingsChange, [41](#)
  - START\_SUN\_POINTS, [42](#)
  - START\_ZOMBIE\_AMOUNT, [42](#)
  - stateChanged, [41](#)
  - strings, [42](#)
  - SUNFLOWER\_SPAWN\_CHANCE, [43](#)
  - values, [43](#)
  - WALNUT\_SPAWN\_CHANCE, [43](#)
  - ZOMBIE\_SPAWN\_INTERVAL, [43](#)
- SettingsChange.java, [59](#)
- shoot
  - Peashooter, [35](#)
- spawnRandomPlant
  - Panel, [28](#)
- spawnRandomZombie
  - Panel, [28](#)
- SpawnSelector, [43](#)
  - actionPerformed, [44](#)
  - findArrayIndex, [45](#)
  - mouseClicked, [45](#)
  - mouseEntered, [45](#)
  - mouseExited, [45](#)
  - mousePressed, [45](#)
  - mouseReleased, [45](#)
  - paint, [46](#)
  - SpawnSelector, [44](#)
  - SpawnSquares, [46](#)
  - theTimer, [46](#)
- SpawnSelector.java, [59](#)
- SpawnSquares
  - Panel, [30](#)
  - SpawnSelector, [46](#)
- spendSunPoints
  - ResourceManager, [40](#)
- SQUARE\_SIZE
  - Panel, [30](#)
- START\_SUN\_POINTS
  - Panel, [30](#)
  - SettingsChange, [42](#)
- START\_ZOMBIE\_AMOUNT
  - Panel, [30](#)
  - SettingsChange, [42](#)
- stateChanged
  - SettingsChange, [41](#)
- strings
  - Panel, [31](#)

- SettingsChange, [42](#)
- Sunflower, [46](#)
  - actionPerformed, [48](#)
  - alive, [50](#)
  - getBounds, [48](#)
  - getCost, [48](#)
  - getHealth, [49](#)
  - getType, [49](#)
  - paintComponent, [49](#)
  - produceCycle, [50](#)
  - resourceManager, [50](#)
  - Sunflower, [48](#)
  - takeDamage, [49](#)
- Sunflower.java, [59](#)
- SUNFLOWER\_SPAWN\_CHANCE
  - Panel, [31](#)
  - SettingsChange, [43](#)
- takeDamage
  - BasicZombie, [12](#)
  - BucketheadZombie, [16](#)
  - CherryBomb, [19](#)
  - Entity, [23](#)
  - Peashooter, [35](#)
  - Sunflower, [49](#)
  - Walnut, [53](#)
- theTimer
  - Entity, [23](#)
  - Panel, [31](#)
  - Projectile, [39](#)
  - SpawnSelector, [46](#)
- values
  - SettingsChange, [43](#)
- Walnut, [50](#)
  - actionPerformed, [51](#)
  - getBounds, [52](#)
  - getCost, [52](#)
  - getHealth, [52](#)
  - getType, [52](#)
  - paintComponent, [52](#)
  - takeDamage, [53](#)
  - Walnut, [51](#)
- Walnut.java, [59](#)
- WALNUT\_SPAWN\_CHANCE
  - Panel, [31](#)
  - SettingsChange, [43](#)
- x
  - Entity, [24](#)
  - Explosion, [25](#)
  - Projectile, [39](#)
- y
  - Entity, [24](#)
  - Explosion, [25](#)
  - Projectile, [39](#)
- Zombie, [53](#)
  - getAttackDamage, [54](#)
  - getSpeed, [54](#)
  - getType, [55](#)
  - Zombie, [54](#)
- Zombie.java, [59](#)
- ZOMBIE\_COLUMNS
  - Panel, [31](#)
- ZOMBIE\_SPAWN\_INTERVAL
  - Panel, [31](#)
  - SettingsChange, [43](#)
- Zombies
  - CherryBomb, [20](#)
  - Panel, [31](#)
  - Peashooter, [35](#)