Plants vs Zombies Simulation

Generated by Doxygen 1.11.0

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:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

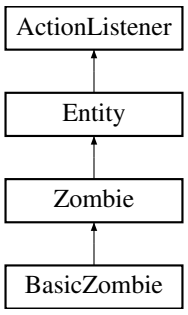
Here is a list of all files with brief descriptions:

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

x	x coordinate
y	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

e	tick of `Panel#theTimer`

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

g	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.

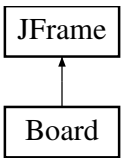
Parameters

damage	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

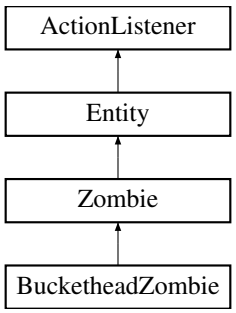
5.2.2.1 Board()

```
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

x	x coordinate
y	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

e	tick of `Panel#theTimer`

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

g	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.

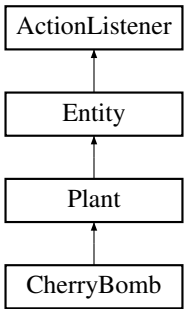
Parameters

damage	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

x	x coordinate
y	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

e	tick of `Panel#theTimer`

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

g	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

damage	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

plants	list of plants alive
zombies	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

explosion	an explosion to check
zombies	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

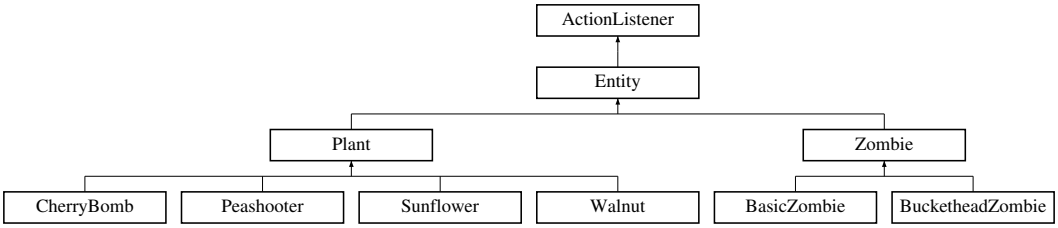
projectile	a projectile to check
zombies	list of zombies alive

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

x	x coordinate
y	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

e	tick of `Panel#theTimer`

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

g	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

damage	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

x	x coordinate
y	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

g	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]
```

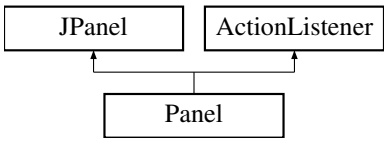
Main function, used to launch the program.

Parameters

args	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 `Zombie`s are eliminated. It also checks attacking `Zombie`s using `CollisionManager#checkAttacks(List, L`
It is also updating the counters.

Parameters

e	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	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 `Plant`s and `Zombie`s 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

g	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_C` `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.

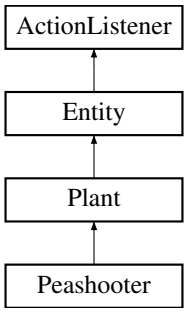
5.9.4.22 Zombies

```
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 `Projectile`s 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

x	x coordinate
y	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

e	tick of `Panel#theTimer`

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

g	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

damage	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 `Projectile`s.

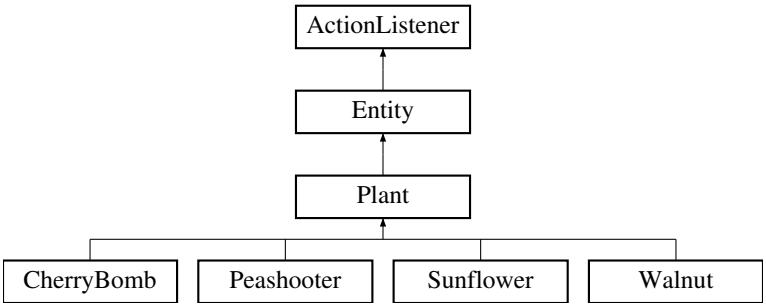
5.10.4.2 Zombies

```
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)
```

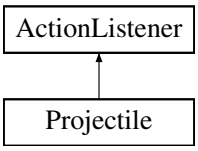
Constructor of a plant

Parameters

x	x coordinate
y	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

x	x coordinate
y	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

e	tick of `Panel#theTimer`

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

g	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

points	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

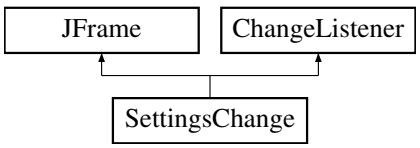
points	amount of Sun Points to spend.

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

e	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.

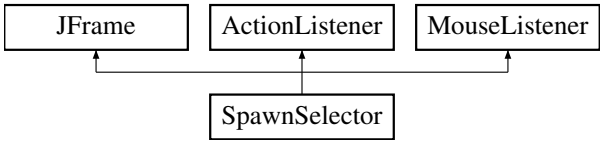
5.14.4.13 ZOMBIE_SPAWN_INTERVAL

```
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

e	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

listOfLists	a list of lists
targetArray	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

e	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

g	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.

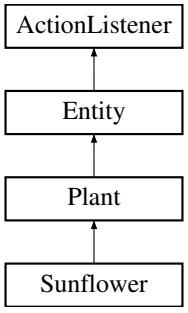
5.15.4.2 theTimer

```
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

x	x coordinate
y	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

e	tick of `Panel#theTimer`

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

g	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

damage	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.

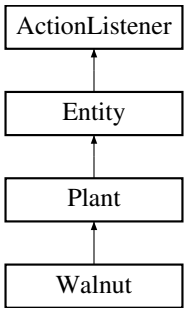
5.16.4.3 resourceManager

```
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

x	x coordinate
y	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

e	tick of `Panel#theTimer`

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

g	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.

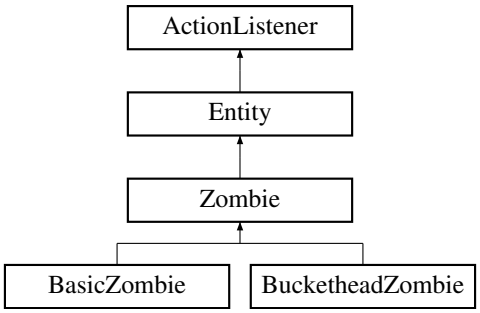
Parameters

damage	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

x	x coordinate
y	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