Class ArrowUI

java.lang.Object
    greenfoot.Actor
        ArrowUI

```
public class ArrowUI
extends greenfoot.Actor
```

Displays the arrows for switching between options

Version:

2020-11-10

Author:

Lucy Zhao, Young Chen

Constructor Summary

Constructors

Constructor	Description
`ArrowUI(boolean isRight)`	Creates a new arrow ui pointing either left or right
`ArrowUI(boolean isRight, int xSize, int ySize)`	Alternative constructor for the ArrowUI

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	`act()`	Actor act method
void	`click()`	Changes arrow to clicked sprite
void	`unClick()`	Changes arrow to unclicked sprite

Methods inherited from class greenfoot.Actor

addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, getX, getY,

intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

ArrowUI

```
public ArrowUI(boolean isRight)
```

Creates a new arrow ui pointing either left or right

Parameters:

`isRight` - Whether or not this is point right. If not, it will point left.

ArrowUI

```
public ArrowUI(boolean isRight,
               int xSize,
               int ySize)
```

Alternative constructor for the ArrowUI

Parameters:

`isRight` - Whether or not this is point right. If not, it will point left.

`xSize` - the custom width

`ySize` - the custom height

Method Detail

act

```
public void act()
```

Actor act method

Overrides:

`act` in class `greenfoot.Actor`

click

```
public void click()
```

Changes arrow to clicked sprite

unClick

```
public void unClick()
```

Changes arrow to unclicked sprite

Class Background

java.lang.Object
    greenfoot.Actor
        Background

```
public class Background
extends greenfoot.Actor
```

The background sprite of the game that moves with the camera

Version:

2020-11-10

Author:

Young Chen

Constructor Summary

Constructors

Constructor	Description
`Background(int x, int y)`	Constructor for Background class.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`int`	`getX()`	Returns the x position
`int`	`getY()`	Returns the y position

Methods inherited from class greenfoot.Actor

`act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards`

Methods inherited from class java.lang.Object

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

Constructor Detail

Background

```
public Background(int x,
                  int y)
```

Constructor for Background class.

Parameters:

x - the x position

y - the y position

Method Detail

getX

```
public int getX()
```

Returns the x position

Overrides:

getX in class greenfoot.Actor

Returns:

int the x position

getY

```
public int getY()
```

Returns the y position

Overrides:

getY in class greenfoot.Actor

Returns:

int the y position

Class Builder

java.lang.Object
    greenfoot.Actor
        Human
            Builder

```
public class Builder
extends Human
```

Builders build new structures when enough resources are present. Each building created is determine by its current demand.

Version:

2020-11-10

Author:

Lucy Zhao

Field Summary

Fields inherited from class Human

`atLocation, BUILDER, BUILDER_SPRITE, BUILDER_WORK_TIME, buildingType, buildSound, chopSound, dead, DEFAULT_HP, DEFAULT_SPEED, enroute, FARMER, FARMER_SPRITE, FARMER_WORK_TIME, FOOD_BIAS, FULL_HUNGER, HOUSE_BIAS, hp, hpBar, hunger, hurtSound, isStarving, isWorking, LUMBERJACK, LUMBERJACK_SPRITE, LUMBERJACK_WORK_TIME, MINER, MINER_SPRITE, MINER_WORK_TIME, mineSound, nearestIndex, offset, SAFETY_TIME, speed, sprite, STARVE_TIME, starveDeathTime, targetBuilding, targetX, targetY, TOTAL_HUMAN_TYPES, type, workBar, xLoc, xVel, yLoc, yVel, ZOMBIE_CHANCE`

Constructor Summary

Constructors

Constructor	Description
`Builder(int xLoc, int yLoc)`	The constructor for the Builder class.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	_update()	Controls the behavior of the builder.
protected void	work()	The work method where the builder creates buildings for the human population.

Methods inherited from class Human

addHealthBar, checkIsAtLocation, checkRoute, damage, drainFood, getHealthBar, getNearestBuilding, getType, getWorkBar, getX, getY, moveTo, randomZombieChance, setRandomRotation, setVolumes, turnTo

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

Builder

```
public Builder(int xLoc,
               int yLoc)
```

The constructor for the Builder class.

Parameters:

xLoc - the x location

yLoc - the y location

Method Detail

_update

```
public void _update()
```

Controls the behavior of the builder.

Specified by:

`_update` in class `Human`

work

```
protected void work()
```

The work method where the builder creates buildings for the human population.

Specified by:

`work` in class `Human`

Class Building

java.lang.Object
    greenfoot.Actor
        Building

```
public abstract class Building
extends greenfoot.Actor
```

Building abstract class that contains the sprite and helper methods

Version:

2020-10-10

Author:

Young Chen, Leo Foo

Field Summary

Fields

Modifier and Type	Field	Description
static greenfoot.GreenfootImage	EMPTY_SPRITE	
static greenfoot.GreenfootImage	FARM_SPRITE	
static greenfoot.GreenfootImage	HOUSE_SPRITE	
static greenfoot.GreenfootImage	MINE_SPRITE	
static greenfoot.GreenfootImage	SENTRY_SPRITE	
protected greenfoot.GreenfootImage	sprite	
static greenfoot.GreenfootImage	STORAGE_SPRITE	

Constructor Summary

Constructors

Constructor	Description
Building()	

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	**_update**()	Essentially the act method for all buildings.
void	**destroy**()	Destroys the building
static Event	**getNearestEvent**(int eventID, int x, int y)	Get the nearest event.
greenfoot.GreenfootImage	**getSprite**()	Returns the image of the building

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, getX, getY, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Field Detail

sprite

protected greenfoot.GreenfootImage sprite

EMPTY_SPRITE

public static final greenfoot.GreenfootImage EMPTY_SPRITE

FARM_SPRITE

```
public static final greenfoot.GreenfootImage FARM_SPRITE
```

HOUSE_SPRITE

```
public static final greenfoot.GreenfootImage HOUSE_SPRITE
```

MINE_SPRITE

```
public static final greenfoot.GreenfootImage MINE_SPRITE
```

SENTRY_SPRITE

```
public static final greenfoot.GreenfootImage SENTRY_SPRITE
```

STORAGE_SPRITE

```
public static final greenfoot.GreenfootImage STORAGE_SPRITE
```

Constructor Detail

Building

```
public Building()
```

Method Detail

_update

```
public void _update()
```

Essentially the act method for all buildings.

destroy

```
public void destroy()
```

Destroys the building

getSprite

```
public greenfoot.GreenfootImage getSprite()
```

Returns the image of the building

Returns:
GreenfootImage the building's image

getNearestEvent

```
public static Event getNearestEvent(int eventID, int x, int y)
```

Get the nearest event.

Parameters:
`eventID` - the type of event

`x` - the x reference location

`y` - the y reference location

Returns:
Event the nearest event

Class BuildingSlot

java.lang.Object
    greenfoot.Actor
        BuildingSlot

```
public class BuildingSlot
extends greenfoot.Actor
```

Locations where buildings can be built.

Version:

2020-10-11

Author:

Lucy Zhao, Young Chen

Field Summary

Fields

Modifier and Type	Field	Description
static int	ARMOURY	
static int	BARRACKS	
static int	DEFAULT_HP	
static greenfoot.GreenfootSound	destroySound	
static int	EMPTY	
static int	FARM	
static float	FARM_PRODUCTION	
static int	HOUSE	
static int	HOUSE_CAPACITY	
static int	MINE	
static float	MINE_PRODUCTION	
static int	SENTRY	
static int	STORAGE	
static int	STORAGE_CAPACITY	

Constructor Summary

Constructors

Constructor	Description
`BuildingSlot(int x, int y, int index)`	Constructor of BuildingSlot, takes coordinates and a index.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	`_update()`	Object update method
void	`damage(int damage)`	Hurt the current building by set amount of damage
void	`destroy()`	Destroys the building
Building	`getBuilding()`	Returns the building at that slot
boolean	`getTargetStatus()`	Returns whether or not a human is targeting this building.
int	`getType()`	Returns the type of building
int	`getX()`	Returns the x location
int	`getY()`	Returns the y location
boolean	`isDestroyed()`	Gets whether or not the buildingslot is destroyed
void	`setBuilding (int buildingID)`	Sets the the building sprite and type.
void	`setTargetStatus (boolean status)`	Sets whether or not a human is targeting this building.

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Field Detail

EMPTY

public static final int EMPTY

See Also:

Constant Field Values

ARMOURY

public static final int ARMOURY

See Also:

Constant Field Values

BARRACKS

public static final int BARRACKS

See Also:

Constant Field Values

FARM

public static final int FARM

See Also:

Constant Field Values

MINE

```
public static final int MINE
```

See Also:
Constant Field Values

SENTRY

```
public static final int SENTRY
```

See Also:
Constant Field Values

STORAGE

```
public static final int STORAGE
```

See Also:
Constant Field Values

HOUSE

```
public static final int HOUSE
```

See Also:
Constant Field Values

FARM_PRODUCTION

```
public static final float FARM_PRODUCTION
```

See Also:
Constant Field Values

MINE_PRODUCTION

```
public static final float MINE_PRODUCTION
```

See Also:
Constant Field Values

STORAGE_CAPACITY

```
public static final int STORAGE_CAPACITY
```

See Also:
Constant Field Values

HOUSE_CAPACITY

```
public static final int HOUSE_CAPACITY
```

See Also:
Constant Field Values

DEFAULT_HP

```
public static final int DEFAULT_HP
```

See Also:
Constant Field Values

destroySound

```
public static final greenfoot.GreenfootSound destroySound
```

Constructor Detail

BuildingSlot

```
public BuildingSlot(int x,
                    int y,
                    int index)
```

Constructor of BuildingSlot, takes coordinates and a index.

Parameters:

`x` - the x location

`y` - the y location

`index` - the index of the BuildingSlot

Method Detail

setTargetStatus

```
public void setTargetStatus(boolean status)
```

Sets whether or not a human is targeting this building.

Parameters:

`status` - true if a human is targeting it, else false

getTargetStatus

```
public boolean getTargetStatus()
```

Returns whether or not a human is targeting this building.

Returns:

boolean true if so, otherwise false

getType

```
public int getType()
```

Returns the type of building

Returns:

int the id of the building

getX

```
public int getX()
```

Returns the x location

Overrides:
`getX` in class `greenfoot.Actor`

Returns:
int the x location of the building

getY

```
public int getY()
```

Returns the y location

Overrides:
`getY` in class `greenfoot.Actor`

Returns:
int the y location of the building

getBuilding

```
public Building getBuilding()
```

Returns the building at that slot

Returns:
Building the building

setBuilding

```
public void setBuilding(int buildingID)
```

Sets the the building sprite and type.

Parameters:
`buildingID` - the type of building

damage

```
public void damage(int damage)
```

Hurt the current building by set amount of damage

Parameters:
`damage` - how much damage is dealt

destroy

```
public void destroy()
```

Destroys the building

isDestroyed

```
public boolean isDestroyed()
```

Gets whether or not the buildingslot is destroyed

Returns:
Whether its destroyed or not

_update

```
public void _update()
```

Object update method

Class Empty

java.lang.Object
    greenfoot.Actor
        Building
            Empty

```
public class Empty
extends Building
```

Empty building class for when no building has been built in the buildingslot

Version:

2020-10-10

Author:

Leo Foo

Field Summary

Fields inherited from class Building

EMPTY_SPRITE, FARM_SPRITE, HOUSE_SPRITE, MINE_SPRITE, SENTRY_SPRITE, sprite, STORAGE_SPRITE

Constructor Summary

Constructors

Constructor	Description
Empty()	Constructor for the Empty class.

Method Summary

Methods inherited from class Building

_update, destroy, getNearestEvent, getSprite

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, getX, getY, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

Empty

```
public Empty()
```

Constructor for the Empty class.

Class EndScreen

java.lang.Object
    greenfoot.Actor
        EndScreen

```
public class EndScreen
extends greenfoot.Actor
```

Screen that fades in at the end of the game

Version:

2020-11-10

Author:

Young Chen

Constructor Summary

Constructors

Constructor	Description
`EndScreen()`	Constructor of EndScreen

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	`act()`	Act method of EndScreen
boolean	`isFinished()`	Returns if the end screen is finished

Methods inherited from class greenfoot.Actor

addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, getX, getY, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

EndScreen

public EndScreen()

Constructor of EndScreen

Method Detail

act

public void act()

Act method of EndScreen

Overrides:

act in class greenfoot.Actor

isFinished

public boolean isFinished()

Returns if the end screen is finished

Returns:

boolean true is finished, otherwise false

Class Enemy

java.lang.Object
    greenfoot.Actor
        Event
            Enemy

```
public abstract class Enemy
extends Event
```

Class of events that is able to actively attack humans

Version:

2020-10-10

Author:

Young Chen

Field Summary

Fields inherited from class Event

damage, DEFAULT_DAMAGE, DEFAULT_HP, hp, METEOR, range, rot, TORNADO, type, xLoc, yLoc, ZOMBIE

Constructor Summary

Constructors

Constructor	Description
Enemy()	

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
protected Human	getNearestHuman(int x, int y)	Gets the nearest human from specified location

Methods inherited from class Event

_update, damage, die, getBuildingsWithinRange, getHumansWithinRange, getTreesWithinRange, getType, getX, getY, killNearbyThings

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

Enemy

```
public Enemy()
```

Method Detail

getNearestHuman

```
protected Human getNearestHuman(int x, int y)
```

Gets the nearest human from specified location

Parameters:

x - Location in x-axis to search from

y - Location in y-axis to search from

Returns:

The closest human that was found

Class Event

java.lang.Object
    greenfoot.Actor
        Event

```
public abstract class Event
extends greenfoot.Actor
```

Events that attack the humans

Version:

2020-11-04

Author:

Young Chen

Field Summary

Fields

Modifier and Type	Field	Description
protected int	damage	
protected static int	DEFAULT_DAMAGE	
protected static int	DEFAULT_HP	
protected int	hp	
static int	METEOR	
protected int	range	
protected int	rot	
static int	TORNADO	
protected int	type	
protected float	xLoc	
protected float	yLoc	
static int	ZOMBIE	

Constructor Summary

Constructors

Constructor	Description

`Event()`

Method Summary

All Methods Static Methods Instance Methods Abstract Methods Concrete Methods

Modifier and Type	Method	Description
abstract void	`_update()`	Update method that all events have
void	`damage(int damage)`	Hurts the event by a specified damage value
void	`die()`	Removes the event from the world
static java.util.ArrayList<BuildingSlot>	`getBuildingsWithinRange (int x, int y, int range)`	Get the buildings within specified range
protected static java.util.ArrayList<Human>	`getHumansWithinRange (int x, int y, int range)`	Get the humans within specified range
protected static java.util.ArrayList<Tree>	`getTreesWithinRange (int x, int y, int range)`	Get the trees within specified range
int	`getType()`	Get the type of the event
int	`getX()`	Get the x location
int	`getY()`	Get the y location
protected static int	`killNearbyThings (int xLoc, int yLoc, int range, int damage)`	Damages the nearby humans, trees, and buildings

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, intersects,

isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Field Detail

METEOR

public static final int METEOR

See Also:

Constant Field Values

TORNADO

public static final int TORNADO

See Also:

Constant Field Values

ZOMBIE

public static final int ZOMBIE

See Also:

Constant Field Values

DEFAULT_DAMAGE

protected static final int DEFAULT_DAMAGE

See Also:

Constant Field Values

DEFAULT_HP

```
protected static final int DEFAULT_HP
```

See Also:
Constant Field Values

damage

```
protected int damage
```

range

```
protected int range
```

rot

```
protected int rot
```

type

```
protected int type
```

hp

```
protected int hp
```

xLoc

```
protected float xLoc
```

yLoc

```
protected float yLoc
```

Constructor Detail

Event

```
public Event()
```

Method Detail

getX

```
public int getX()
```

Get the x location

Overrides:

`getX` in class `greenfoot.Actor`

Returns:

X location

getY

```
public int getY()
```

Get the y location

Overrides:

`getY` in class `greenfoot.Actor`

Returns:

Y location

getType

```
public int getType()
```

Get the type of the event

Returns:
EventID of event

die

```
public void die()
```

Removes the event from the world

damage

```
public void damage(int damage)
```

Hurts the event by a specified damage value

Parameters:
`damage` - the value of damage inflicted

getBuildingsWithinRange

```
public static java.util.ArrayList<BuildingSlot> getBuildingsWithinRange(int x,
int y, int range)
```

Get the buildings within specified range

Parameters:
`x` - Location in x axis to search from

`y` - Location in y axis to search from

`range` - Range to search for buildings

Returns:
Arraylist of buildings found within the range

getHumansWithinRange

```
protected static java.util.ArrayList<Human> getHumansWithinRange(int x, int y,
int range)
```

Get the humans within specified range

Parameters:

`x` - Location in x axis to search from

`y` - Location in y axis to search from

`range` - Range to search for humans

Returns:

Arraylist of humans found within the range

getTreesWithinRange

```
protected static java.util.ArrayList<Tree> getTreesWithinRange(int x, int y,
int range)
```

Get the trees within specified range

Parameters:

`x` - Location in x axis to search from

`y` - Location in y axis to search from

`range` - Range to search for trees

Returns:

Arraylist of trees found within the range

killNearbyThings

```
protected static int killNearbyThings(int xLoc, int yLoc, int range,
int damage)
```

Damages the nearby humans, trees, and buildings

Parameters:

`xLoc` - Location in x axis to search for objects that are to be damaged from

`yLoc` - Location in y axis to search for objects that are to be damaged from

`range` - The range of the object search

`damage` - How much to damage the objects found

Returns:

number of things that were damaged

_update

```
public abstract void _update()
```

Update method that all events have

Class Fade

java.lang.Object
    greenfoot.Actor
        Fade

```
public class Fade
extends greenfoot.Actor
```

Write a description of class Fade here.

Version:

2020-11-12

Author:

Young Chen

Constructor Summary

Constructors

Constructor	Description
`Fade(boolean in)`	Creates a sprite that fades in or out

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	`act()`	Act - do whatever the Fade wants to do.
boolean	`isFinished()`	Whether or not the fade has finished
void	`start()`	Starts the fade

Methods inherited from class greenfoot.Actor

addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, getX, getY, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

Fade

`public Fade(boolean in)`

Creates a sprite that fades in or out

Parameters:

`in` - Whether or not to fade in. If not, it will fade out.

Method Detail

act

`public void act()`

Act - do whatever the Fade wants to do. This method is called whenever the 'Act' or 'Run' button gets pressed in the environment.

Overrides:

`act` in class `greenfoot.Actor`

start

`public void start()`

Starts the fade

isFinished

```
public boolean isFinished()
```

Whether or not the fade has finished

Returns:

Boolean value of whether or not the fade has finished

Class Farm

java.lang.Object
    greenfoot.Actor
        Building
            Farm

```
public class Farm
extends Building
```

Sprite for the farm buildingslot

Version:

2020-10-10

Author:

Leo Foo

Field Summary

Fields inherited from class Building

EMPTY_SPRITE, FARM_SPRITE, HOUSE_SPRITE, MINE_SPRITE, SENTRY_SPRITE, sprite, STORAGE_SPRITE

Constructor Summary

Constructors

Constructor	Description
Farm()	Constructor for the Farm class.

Method Summary

Methods inherited from class Building

_update, destroy, getNearestEvent, getSprite

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, getX, getY, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

Farm

public Farm()

Constructor for the Farm class.

Class Farmer

java.lang.Object
    greenfoot.Actor
        Human
            Farmer

```
public class Farmer
extends Human
```

Farmers work at farm structures and collect food resources to prevent the population from starving.

Version:

2020-11-10

Author:

Lucy Zhao

Field Summary

Fields inherited from class Human

`atLocation, BUILDER, BUILDER_SPRITE, BUILDER_WORK_TIME, buildingType, buildSound, chopSound, dead, DEFAULT_HP, DEFAULT_SPEED, enroute, FARMER, FARMER_SPRITE, FARMER_WORK_TIME, FOOD_BIAS, FULL_HUNGER, HOUSE_BIAS, hp, hpBar, hunger, hurtSound, isStarving, isWorking, LUMBERJACK, LUMBERJACK_SPRITE, LUMBERJACK_WORK_TIME, MINER, MINER_SPRITE, MINER_WORK_TIME, mineSound, nearestIndex, offset, SAFETY_TIME, speed, sprite, STARVE_TIME, starveDeathTime, targetBuilding, targetX, targetY, TOTAL_HUMAN_TYPES, type, workBar, xLoc, xVel, yLoc, yVel, ZOMBIE_CHANCE`

Constructor Summary

Constructors

Constructor	Description
`Farmer(int xLoc, int yLoc)`	The constructor for the Farmer class.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	_update()	Controls the behavior of the farmer.
protected void	work()	The work method where the farmer gains resources for the human population.

Methods inherited from class Human

addHealthBar, checkIsAtLocation, checkRoute, damage, drainFood, getHealthBar, getNearestBuilding, getType, getWorkBar, getX, getY, moveTo, randomZombieChance, setRandomRotation, setVolumes, turnTo

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

Farmer

```
public Farmer(int xLoc,
              int yLoc)
```

The constructor for the Farmer class.

Parameters:

xLoc - the x location

yLoc - the y location

Method Detail

_update

```
public void _update()
```

Controls the behavior of the farmer.

Specified by:

`_update` in class `Human`

work

```
protected void work()
```

The work method where the farmer gains resources for the human population.

Specified by:

`work` in class `Human`

Class House

java.lang.Object
    greenfoot.Actor
        Building
            House

```
public class House
extends Building
```

House sprite for the house buildingslot

Version:

2020-10-10

Author:

Young Chen

Field Summary

Fields inherited from class Building

EMPTY_SPRITE, FARM_SPRITE, HOUSE_SPRITE, MINE_SPRITE, SENTRY_SPRITE, sprite, STORAGE_SPRITE

Constructor Summary

Constructors

Constructor	Description
House(BuildingSlot slot)	Constructor for the House class.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	_update()	The update method of House.

Methods inherited from class Building

destroy, getNearestEvent, getSprite

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, getX, getY, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

House

```
public House(BuildingSlot slot)
```

Constructor for the House class.

Parameters:

`slot` - the BuildingSlot the House is located at

Method Detail

_update

```
public void _update()
```

The update method of House.

Overrides:

`_update` in class `Building`

Class Human

java.lang.Object
    greenfoot.Actor
        Human

```
public abstract class Human
extends greenfoot.Actor
```

Superclass for the humans, who are beings who try and survive in the world by building structures and collecting resources.

Version:

2020-11-08

Author:

Lucy Zhao, Young Chen

Field Summary

Fields

Modifier and Type	Field	Description
protected boolean	atLocation	
static int	BUILDER	
static greenfoot.GreenfootImage	BUILDER_SPRITE	
static int	BUILDER_WORK_TIME	
protected int	buildingType	
static greenfoot.GreenfootSound	buildSound	
static greenfoot.GreenfootSound	chopSound	
protected boolean	dead	
protected static int	DEFAULT_HP	
protected static float	DEFAULT_SPEED	
protected boolean	enroute	
static int	FARMER	
static greenfoot.GreenfootImage	FARMER_SPRITE	
static int	FARMER_WORK_TIME	
protected static float	FOOD_BIAS	
protected static float	FULL_HUNGER	
protected static float	HOUSE_BIAS	

protected int	**hp**
protected StatBar	**hpBar**
protected float	**hunger**
static greenfoot.GreenfootSound	**hurtSound**
protected boolean	**isStarving**
protected boolean	**isWorking**
static int	**LUMBERJACK**
static greenfoot.GreenfootImage	**LUMBERJACK_SPRITE**
static int	**LUMBERJACK_WORK_TIME**
static int	**MINER**
static greenfoot.GreenfootImage	**MINER_SPRITE**
static int	**MINER_WORK_TIME**
static greenfoot.GreenfootSound	**mineSound**
protected int	**nearestIndex**
protected int	**offset**
protected static int	**SAFETY_TIME**
protected int	**speed**
protected greenfoot.GreenfootImage	**sprite**
protected static float	**STARVE_TIME**
protected float	**starveDeathTime**
protected BuildingSlot	**targetBuilding**
protected int	**targetX**
protected int	**targetY**
static int	**TOTAL_HUMAN_TYPES**
protected int	**type**
protected StatBar	**workBar**
protected int	**xLoc**
protected int	**xVel**
protected int	**yLoc**
protected int	**yVel**
protected static int	**ZOMBIE_CHANCE**

Constructor Summary

Constructors

Constructor	Description
Human()	

Method Summary

All Methods Static Methods Instance Methods Abstract Methods Concrete Methods

Modifier and Type	Method	Description
abstract void	_update()	Essentially the act method for all human instances.
protected void	addHealthBar()	Add the human's health bar to the world
protected void	checkIsAtLocation (int xDest, int yDest)	Checks if the human has reached targeted location.
protected void	checkRoute (int buildngID, int xLoc, int yLoc)	Checks the human's route, if there is none, select a building or random location for the human to move to.
void	damage(int val)	Causes the human instance to lose a specified number of health points.
protected void	drainFood()	Drains food (eaten) and controls whether or not humans die of starvation.
StatBar	getHealthBar()	Returns the human's health bar
protected BuildingSlot	getNearestBuilding (int buildngID, int x, int y)	Returns the nearest building when given a specific starting location and type.
int	getType()	Returns the type of human
StatBar	getWorkBar()	Returns the human's work bar
int	getX()	Returns the x location
int	getY()	Returns the y location
protected	moveTo(int xDest,	Moves the human to the chosen location

void	int yDest)	
protected void	**randomZombieChance**()	Humans can randomly turn into zombies
protected void	**setRandomRotation**()	Set a random rotation for the human.
static void	**setVolumes**()	Set the volume of human sounds.
protected void	**turnTo**(int x, int y)	Turns the human to the direction it is moving towards
protected abstract void	**work**()	Each human has their own work method as they gain different resources and take different amounts of time to complete their tasks.

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Field Detail

BUILDER

public static final int BUILDER

See Also:

Constant Field Values

FARMER

public static final int FARMER

See Also:
Constant Field Values

LUMBERJACK

```
public static final int LUMBERJACK
```

See Also:
Constant Field Values

MINER

```
public static final int MINER
```

See Also:
Constant Field Values

TOTAL_HUMAN_TYPES

```
public static final int TOTAL_HUMAN_TYPES
```

See Also:
Constant Field Values

DEFAULT_HP

```
protected static final int DEFAULT_HP
```

See Also:
Constant Field Values

SAFETY_TIME

```
protected static final int SAFETY_TIME
```

See Also:
Constant Field Values

ZOMBIE_CHANCE

```
protected static final int ZOMBIE_CHANCE
```

See Also:
Constant Field Values

BUILDER_WORK_TIME

```
public static final int BUILDER_WORK_TIME
```

See Also:
Constant Field Values

FARMER_WORK_TIME

```
public static final int FARMER_WORK_TIME
```

See Also:
Constant Field Values

LUMBERJACK_WORK_TIME

```
public static final int LUMBERJACK_WORK_TIME
```

See Also:
Constant Field Values

MINER_WORK_TIME

```
public static final int MINER_WORK_TIME
```

See Also:
Constant Field Values

BUILDER_SPRITE

public static final greenfoot.GreenfootImage BUILDER_SPRITE

FARMER_SPRITE

public static final greenfoot.GreenfootImage FARMER_SPRITE

LUMBERJACK_SPRITE

public static final greenfoot.GreenfootImage LUMBERJACK_SPRITE

MINER_SPRITE

public static final greenfoot.GreenfootImage MINER_SPRITE

buildSound

public static final greenfoot.GreenfootSound buildSound

mineSound

public static final greenfoot.GreenfootSound mineSound

chopSound

public static final greenfoot.GreenfootSound chopSound

hurtSound

public static final greenfoot.GreenfootSound hurtSound

FOOD_BIAS

```
protected static final float FOOD_BIAS
```

See Also:
Constant Field Values

HOUSE_BIAS

```
protected static final float HOUSE_BIAS
```

See Also:
Constant Field Values

DEFAULT_SPEED

```
protected static final float DEFAULT_SPEED
```

See Also:
Constant Field Values

FULL_HUNGER

```
protected static final float FULL_HUNGER
```

See Also:
Constant Field Values

STARVE_TIME

```
protected static final float STARVE_TIME
```

See Also:
Constant Field Values

xLoc

protected int xLoc

yLoc

protected int yLoc

xVel

protected int xVel

yVel

protected int yVel

nearestIndex

protected int nearestIndex

speed

protected int speed

sprite

protected greenfoot.GreenfootImage sprite

offset

protected int offset

targetBuilding

```
protected BuildingSlot targetBuilding
```

atLocation

```
protected boolean atLocation
```

enroute

```
protected boolean enroute
```

targetX

```
protected int targetX
```

targetY

```
protected int targetY
```

buildingType

```
protected int buildingType
```

isWorking

```
protected boolean isWorking
```

workBar

```
protected StatBar workBar
```

hunger

```
protected float hunger
```

isStarving

```
protected boolean isStarving
```

dead

```
protected boolean dead
```

starveDeathTime

```
protected float starveDeathTime
```

hp

```
protected int hp
```

type

```
protected int type
```

hpBar

```
protected StatBar hpBar
```

Constructor Detail

Human

```
public Human()
```

Method Detail

setVolumes

```
public static void setVolumes()
```

Set the volume of human sounds.

_update

```
public abstract void _update()
```

Essentially the act method for all human instances. Allows for better control of which actors act first.

work

```
protected abstract void work()
```

Each human has their own work method as they gain different resources and take different amounts of time to complete their tasks.

randomZombieChance

```
protected void randomZombieChance()
```

Humans can randomly turn into zombies

moveTo

```
protected void moveTo(int xDest, int yDest)
```

Moves the human to the chosen location

Parameters:

`xDest` - the x destination

`yDest` - the y destination

turnTo

```
protected void turnTo(int x, int y)
```

Turns the human to the direction it is moving towards

Parameters:

`x` - the x destination

`y` - the y destination

getNearestBuilding

```
protected BuildingSlot getNearestBuilding(int buildngID, int x, int y)
```

Returns the nearest building when given a specific starting location and type.

Parameters:

`buildngID` - the type of building to be found

`x` - the x starting location

`y` - the y starting location

Returns:

BuildingSlot the closest building

drainFood

```
protected void drainFood()
```

Drains food (eaten) and controls whether or not humans die of starvation.

checkRoute

```
protected void checkRoute(int buildngID, int xLoc, int yLoc)
```

Checks the human's route, if there is none, select a building or random location for the human to move to.

Parameters:

`buildngID` - the type of building

`xLoc` - the x location to check

`yLoc` - the y location to check

damage

```
public void damage(int val)
```

Causes the human instance to lose a specified number of health points.

Parameters:

`damage` - the value of hp lost

getWorkBar

```
public StatBar getWorkBar()
```

Returns the human's work bar

Returns:

StatBar the work bar

getHealthBar

```
public StatBar getHealthBar()
```

Returns the human's health bar

Returns:

StatBar the health bar

setRandomRotation

`protected void setRandomRotation()`

Set a random rotation for the human.

addHealthBar

`protected void addHealthBar()`

Add the human's health bar to the world

checkIsAtLocation

`protected void checkIsAtLocation(int xDest, int yDest)`

Checks if the human has reached targeted location.

Parameters:

`xDest` - the x destination

`yDest` - the y destination

getX

`public int getX()`

Returns the x location

Overrides:

`getX` in class `greenfoot.Actor`

Returns:

int the x location of the building

getY

`public int getY()`

Returns the y location

Overrides:

`getY` in class `greenfoot.Actor`

Returns:

int the y location of the human

getType

```
public int getType()
```

Returns the type of human

Returns:

int the type of the human

Class Info

java.lang.Object
    greenfoot.World
        Info

```
public class Info
extends greenfoot.World
```

World containing the controls and general information about the simulation

Version:

2020-11-13

Author:

Young Chen

Constructor Summary

Constructors

Constructor	Description
`Info()`	Constructor for objects of class Info.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	`act()`	Act method for Info Class

Methods inherited from class greenfoot.World

addObject, getBackground, getCellSize, getColorAt, getHeight, getObjects, getObjectsAt, getWidth, numberOfObjects, removeObject, removeObjects, repaint, setActOrder, setBackground, setBackground, setPaintOrder, showText, started, stopped

Methods inherited from class java.lang.Object

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

Constructor Detail

Info

```
public Info()
```

Constructor for objects of class Info.

Method Detail

act

```
public void act()
```

Act method for Info Class

Overrides:

`act` in class `greenfoot.World`

Class Lumberjack

java.lang.Object
    greenfoot.Actor
        Human
            Lumberjack

```
public class Lumberjack
extends Human
```

Lumberjacks scatter across the world and chop down trees to collect wood for creating buildings.

Version:

2020-11-10

Author:

Lucy Zhao

Field Summary

Fields inherited from class Human

`atLocation, BUILDER, BUILDER_SPRITE, BUILDER_WORK_TIME, buildingType, buildSound, chopSound, dead, DEFAULT_HP, DEFAULT_SPEED, FARMER, FARMER_SPRITE, FARMER_WORK_TIME, FOOD_BIAS, FULL_HUNGER, HOUSE_BIAS, hp, hpBar, hunger, hurtSound, isStarving, isWorking, LUMBERJACK, LUMBERJACK_SPRITE, LUMBERJACK_WORK_TIME, MINER, MINER_SPRITE, MINER_WORK_TIME, mineSound, nearestIndex, offset, SAFETY_TIME, speed, sprite, STARVE_TIME, starveDeathTime, targetBuilding, targetX, targetY, TOTAL_HUMAN_TYPES, type, workBar, xLoc, xVel, yLoc, yVel, ZOMBIE_CHANCE`

Constructor Summary

Constructors

Constructor	Description
`Lumberjack(int xLoc, int yLoc)`	The constructor for the Lumberjack class.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	**_update**()	Controls the behavior of the lumberjack.
void	**die**()	Removes the human instance from the list and the world.
protected void	**work**()	The work method where the lumberjack gains resources for the human population.

Methods inherited from class Human

addHealthBar, checkIsAtLocation, checkRoute, damage, drainFood, getHealthBar, getNearestBuilding, getType, getWorkBar, getX, getY, moveTo, randomZombieChance, setRandomRotation, setVolumes, turnTo

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

Lumberjack

```
public Lumberjack(int xLoc,
                  int yLoc)
```

The constructor for the Lumberjack class.

Parameters:

xLoc - the x location

yLoc - the y location

Method Detail

_update

```
public void _update()
```

Controls the behavior of the lumberjack.

Specified by:
`_update` in class `Human`

die

```
public void die()
```

Removes the human instance from the list and the world.

work

```
protected void work()
```

The work method where the lumberjack gains resources for the human population.

Specified by:
`work` in class `Human`

Class LZTextBox

java.lang.Object
    greenfoot.Actor
        LZTextBox

```
public class LZTextBox
extends greenfoot.Actor
```

A simple, modular class that implements a customizable text box.

Can be used for dialogue, descriptions or as a button. Text boxes can have unique border width, box height/width and unique colors.

Colors can be any valid Greenfoot Color instance. Height, width and border width can be any positive number, so inputting a negative number will just cause the instance variable to have the default value instead.

Implementation Notes -

- Use the newline character (\n) to represent multiple lines
- Images can be used as the text box instead. Their size is based on their width and height, but can be changed with the update() method
- The class will automatically centers text on the y axis, but if the total height of the text exceeds the height of the text box, the text will be cut off
- To make text boxes appear before clicking run, make sure to use a constructor that takes x and y coordinates. Then use the update() method to add the text and finally use the updateText() method to display it on the world

Potential Future Additions- (I plan to add on)

- Padding customization on x axis and y axis
- Ability to add an array of strings, each element representing a newline
- More font customization (but may require more manual alignment, since fonts have different sizes, assuming monospaced fonts are used)
- Ability to use GreenfootImage shapes (ovals, etc)

Version:

1.0.0

Author:

Lucy Zhao

Constructor Summary

Constructors

Constructor	Description
`LZTextBox()`	The default constructor for default text box.

LZTextBox(int x, int y)	Displays the default text box before the scenario starts.
LZTextBox(int fontSize, int borderWidth, int width, int height)	Add more customization to border width, plus width and height of the text box.
LZTextBox(int x, int y, greenfoot.Color textColor, int fontSize, java.lang.String alignment, int borderWidth, int width, int height, greenfoot.Color borderColor, greenfoot.Color boxColor)	Ultimate constructor that allows customization of text color, font size, coordinates, as well as customization for the text box.
LZTextBox(int x, int y, greenfoot.Color textColor, int fontSize, java.lang.String alignment, greenfoot.GreenfootImage image)	Allows for a custom image background and allows specific coordinates, which means it will appear before run is clicked.
LZTextBox(greenfoot.Color textColor, int fontSize, java.lang.String alignment)	Add more customization to text, including color, font size.
LZTextBox(greenfoot.Color textColor, int fontSize, java.lang.String alignment, int borderWidth, int width, int height)	Control the size of the text box, as well as the its border.
LZTextBox(greenfoot.Color textColor, int fontSize, java.lang.String alignment, int borderWidth, int width, int height, greenfoot.Color borderColor, greenfoot.Color boxColor)	Ultimate constructor that allows customization of text color, font size, as well as customization for the text box.
LZTextBox(greenfoot.Color textColor, int fontSize, java.lang.String alignment, greenfoot.GreenfootImage image)	Allows for a custom image background.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	**act**()	Act method for text box instances.
boolean	**getDisplay**()	Returns whether or not the text box is displaying.
void	**update**(int x, int y)	

		Updates the location of the text box.
void	**update**(int boxWidth, int boxHeight, int borderWidth)	Updates size of the text box.
void	**update**(greenfoot.Color boxColor, greenfoot.Color borderColor, greenfoot.Color textColor)	Updates the colors of the text box.
void	**update**(java.lang.String text)	Adds another line(s) of text to be displayed on the text box.
void	**updateText**()	Changes the displayed text to the next one.

Methods inherited from class greenfoot.Actor

addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, getX, getY, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

LZTextBox

```
public LZTextBox()
```

The default constructor for default text box.

LZTextBox

```
public LZTextBox(int x,
                 int y)
```

Displays the default text box before the scenario starts.

Parameters:

`x` - the x coordinate of the text box

`y` - the y coordinate of the text box

LZTextBox

```
public LZTextBox(greenfoot.Color textColor,
                 int fontSize,
                 java.lang.String alignment)
```

Add more customization to text, including color, font size.

Parameters:

`textColor` - color of the message displayed

`fontSize` - font size of the message

`aligment` - the alignment of the text (left, right, center)

LZTextBox

```
public LZTextBox(int fontSize,
                 int borderWidth,
                 int width,
                 int height)
```

Add more customization to border width, plus width and height of the text box.

Parameters:

`fontSize` - size of the font

`borderWidth` - width of box border

`width` - width of text box

`height` - height of text box

LZTextBox

```
public LZTextBox(greenfoot.Color textColor,
                 int fontSize,
                 java.lang.String alignment,
                 int borderWidth,
```

```
                int width,
                int height)
```

Control the size of the text box, as well as the its border. Also controls all text customization.

Parameters:

`textColor` - color of the message displayed

`fontSize` - font size of the message

`aligment` - the alignment of the text

`borderWidth` - width of box border

`width` - width of text box

`height` - height of text box

LZTextBox

```
public LZTextBox(int x,
                int y,
                greenfoot.Color textColor,
                int fontSize,
                java.lang.String alignment,
                greenfoot.GreenfootImage image)
```

Allows for a custom image background and allows specific coordinates, which means it will appear before run is clicked.

Parameters:

`x` - the x coordinate of the text box

`y` - the y coordinate of the text box

`textColor` - color of the message displayed

`fontSize` - font size of the message

`aligment` - the alignment of the text

`image` - GreenfootImage of an image used

LZTextBox

```
public LZTextBox(int x,
                int y,
                greenfoot.Color textColor,
                int fontSize,
                java.lang.String alignment,
```

```
                int borderWidth,
                int width,
                int height,
                greenfoot.Color borderColor,
                greenfoot.Color boxColor)
```

Ultimate constructor that allows customization of text color, font size, coordinates, as well as customization for the text box.

Parameters:

`x` - the x coordinate of the text box

`y` - the y coordinate of the text box

`textColor` - color of the message displayed

`fontSize` - font size of the message

`aligment` - the alignment of the text

`borderWidth` - width of box border

`width` - width of text box

`height` - height of text box

`borderColor` - color of the text box border

`boxColor` - color of the text box

LZTextBox

```
public LZTextBox(greenfoot.Color textColor,
                int fontSize,
                java.lang.String alignment,
                greenfoot.GreenfootImage image)
```

Allows for a custom image background. The size of the text box is the same of the image chosen. However, height and width of image can be changed with the update() method.

Parameters:

`textColor` - color of the message displayed

`fontSize` - font size of the message

`aligment` - the alignment of the text

`image` - GreenfootImage of an image used

LZTextBox

```
public LZTextBox(greenfoot.Color textColor,
                 int fontSize,
                 java.lang.String alignment,
                 int borderWidth,
                 int width,
                 int height,
                 greenfoot.Color borderColor,
                 greenfoot.Color boxColor)
```

Ultimate constructor that allows customization of text color, font size, as well as customization for the text box.

Parameters:

`textColor` - color of the message displayed

`fontSize` - font size of the message

`aligment` - the alignment of the text

`borderWidth` - width of box border

`width` - width of text box

`height` - height of text box

`borderColor` - color of the text box border

`boxColor` - color of the text box

Method Detail

act

```
public void act()
```

Act method for text box instances.

Overrides:

`act` in class `greenfoot.Actor`

updateText

```
public void updateText()
```

Changes the displayed text to the next one. Public method so that the user can update text with any key.

update

```
public void update(greenfoot.Color boxColor, greenfoot.Color borderColor,
greenfoot.Color textColor)
```

Updates the colors of the text box. If given a null value for color, then the color will be the same as before.

Parameters:

`boxColor` - the new bg color for the text box

`borderColor` - the new color for the border

`textColor` - the new color for the text

update

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

Updates the location of the text box. Text box can be placed anywhere, including outside the world.

Parameters:

`x` - the new x coordinate

`y` - the new y coordinate

update

```
public void update(int boxWidth, int boxHeight, int borderWidth)
```

Updates size of the text box. If negative values are added, the dimensions will not change at all.

Parameters:

`boxWidth` - new width of the text box

`boxHeight` - new height of the text box

`borderWidth` - new border width of the text box

update

```
public void update(java.lang.String text)
```

Adds another line(s) of text to be displayed on the text box. Use \n to represent a newline.

Parameters:

`text` - string to be displayed

getDisplay

```
public boolean getDisplay()
```

Returns whether or not the text box is displaying.

Returns:

boolean true if displaying currently, else false

Class Mine

java.lang.Object
    greenfoot.Actor
        Building
            Mine

```
public class Mine
extends Building
```

Mine sprite for the mine buildingslot

Version:

2020-10-10

Author:

Leo Foo

Field Summary

Fields inherited from class Building

EMPTY_SPRITE, FARM_SPRITE, HOUSE_SPRITE, MINE_SPRITE, SENTRY_SPRITE, sprite, STORAGE_SPRITE

Constructor Summary

Constructors

Constructor	Description
Mine()	Constructor for the Mine class.

Method Summary

Methods inherited from class Building

_update, destroy, getNearestEvent, getSprite

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, getX, getY, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

Mine

```
public Mine()
```

Constructor for the Mine class.

Class Miner

java.lang.Object
    greenfoot.Actor
        Human
            Miner

```
public class Miner
extends Human
```

Miners work at the mine structure and help collect iron for crafting sentry bullets for protection.

Version:

2020-11-10

Author:

Lucy Zhao

Field Summary

Fields inherited from class Human

`atLocation, BUILDER, BUILDER_SPRITE, BUILDER_WORK_TIME, buildingType, buildSound, chopSound, dead, DEFAULT_HP, DEFAULT_SPEED, enroute, FARMER, FARMER_SPRITE, FARMER_WORK_TIME, FOOD_BIAS, FULL_HUNGER, HOUSE_BIAS, hp, hpBar, hunger, hurtSound, isStarving, isWorking, LUMBERJACK, LUMBERJACK_SPRITE, LUMBERJACK_WORK_TIME, MINER, MINER_SPRITE, MINER_WORK_TIME, mineSound, nearestIndex, offset, SAFETY_TIME, speed, sprite, STARVE_TIME, starveDeathTime, targetBuilding, targetX, targetY, TOTAL_HUMAN_TYPES, type, workBar, xLoc, xVel, yLoc, yVel, ZOMBIE_CHANCE`

Constructor Summary

Constructors

Constructor	Description
`Miner(int xLoc, int yLoc)`	The constructor for the Miner class.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	_update()	Controls the behavior of the miner.
protected void	work()	The work method where the miner gains resources for the human population

Methods inherited from class Human

addHealthBar, checkIsAtLocation, checkRoute, damage, drainFood, getHealthBar, getNearestBuilding, getType, getWorkBar, getX, getY, moveTo, randomZombieChance, setRandomRotation, setVolumes, turnTo

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

Miner

```
public Miner(int xLoc,
             int yLoc)
```

The constructor for the Miner class.

Parameters:

xLoc - the x location

yLoc - the y location

Method Detail

_update

```
public void _update()
```

Controls the behavior of the miner.

Specified by:

`_update` in class `Human`

work

```
protected void work()
```

The work method where the miner gains resources for the human population

Specified by:

`work` in class `Human`

Class ScoreBar

java.lang.Object
    greenfoot.Actor
        ScoreBar

```
public class ScoreBar
extends greenfoot.Actor
```

A simple class that displays the data and statistics of the world.

Version:

2020-11-09

Author:

Lucy Zhao

Constructor Summary

Constructors

Constructor	Description
`ScoreBar(int width, int height)`	Construtor of ScoreBar class.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	`addStat(java.lang.String statName, int statValue)`	Adds a stat to be displayed on the scorebar.
void	`hideStats(boolean hide)`	Method that hides the scorebar.
void	`updateStat(java.lang.String statName, int statValue)`	Updates a specified stat with the right value

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, getX, getY,

intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

ScoreBar

```
public ScoreBar(int width,
                int height)
```

Construtor of ScoreBar class. Takes a width and a height.

Parameters:

`width` - the width of the scorebar

`height` - the height of the scorebar

Method Detail

addStat

```
public void addStat(java.lang.String statName, int statValue)
```

Adds a stat to be displayed on the scorebar.

updateStat

```
public void updateStat(java.lang.String statName, int statValue)
```

Updates a specified stat with the right value

hideStats

```
public void hideStats(boolean hide)
```

Method that hides the scorebar.

Class Sentry

java.lang.Object
    greenfoot.Actor
        Building
            Sentry

```
public class Sentry
extends Building
```

Sentry that attacks nearby zombies

Version:

2020-10-09

Author:

Young Chen

Field Summary

Fields

Modifier and Type	Field	Description
`static greenfoot.GreenfootSound`	`fireSound`	

Fields inherited from class Building

`EMPTY_SPRITE, FARM_SPRITE, HOUSE_SPRITE, MINE_SPRITE, SENTRY_SPRITE, sprite, STORAGE_SPRITE`

Constructor Summary

Constructors

Constructor	Description
`Sentry(int xLoc, int yLoc)`	Constructor for the Sentry class.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	_update()	The update method of the Sentry class.

Methods inherited from class Building

destroy, getNearestEvent, getSprite

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, getX, getY, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Field Detail

fireSound

```
public static final greenfoot.GreenfootSound fireSound
```

Constructor Detail

Sentry

```
public Sentry(int xLoc,
              int yLoc)
```

Constructor for the Sentry class.

Parameters:

`xLoc` - the x location

`yLoc` - the y location

Method Detail

_update

```
public void _update()
```

The update method of the Sentry class. Targets and kills zombies.

Overrides:

`_update` in class `Building`

Class Settings

java.lang.Object
    greenfoot.World
        Settings

```
public class Settings
extends greenfoot.World
```

World for the simulation settings

Version:

2020-11-12

Author:

Young Chen

Constructor Summary

Constructors

Constructor	Description
`Settings()`	Constructor for objects of class Settings.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`act()`	Act method for Settings.

Methods inherited from class greenfoot.World

`addObject, getBackground, getCellSize, getColorAt, getHeight, getObjects, getObjectsAt, getWidth, numberOfObjects, removeObject, removeObjects, repaint, setActOrder, setBackground, setBackground, setPaintOrder, showText, started, stopped`

Methods inherited from class java.lang.Object

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

Constructor Detail

Settings

```
public Settings()
```

Constructor for objects of class Settings.

Method Detail

act

```
public void act()
```

Act method for Settings.

Overrides:

`act` in class `greenfoot.World`

Class Simulation

java.lang.Object
    greenfoot.World
        Simulation

```
public class Simulation
extends greenfoot.World
```

Main world class where the simulation takes place. This project simulates the survival of a group of humans in a post-apocalyptic world. In this simulation, the humans will gather resources to keep themselves alive from starvation and any zombies that approach. There are three resources present: wood, iron, and food. Wood is required to build buildings, and each building requires 15 wood to build. Iron is required to fuel the sentries with ammunition, where each sentry shot costs 3 iron. Food is required to keep all the humans alive. Each human consumes food every update to fill up an internal hunger value, which itself decreases every update too. Once a human's hunger reaches 0, the human will die. All of these resources, with the exception of wood, are generated through buildings. Food is produced on farms, which generate food at a rate proportional to the ratio between farmers and farms. Iron is produced in mines, which, just like the farms, produces iron at a rate proportional to the number of miners. Wood, on the other hand, comes from the trees the lumberjacks cut down. Each tree generates between 5 to 20 wood. There are also three other types of buildings in this simulation, namingly sentries, storage, and houses. Sentries consume the aforementioned amount of iron with each shot fired to protect the humans from zombies, with each shot dealing 20 damage points to their target. Storage buildings set the limit for the maximum amount of each type of resource the humans can have. The starting capacity for no storage buildings is 100, and with each additional storage building built, the capacity is increased by 100. Houses are used to increase the population, and will only do so if the housing capacity is larger than the population. Each house increases the housing capacity by 5, and once the housing capacity has grown larger than the population, the houses will start spawning humans until the housing capacity is reached. Each human required 15 food to spawn, and houses will only spawn a new human if the total food in the simulation is greater than 25. There are four major categories of humans: farmers, miners, builders, and lumberjacks. Farmers seek the nearest empty farm, and bind themselves to it. The same goes for miners, but with mines instead. Builders look for the nearest empty building slot, which are generated when the world is first created, and build the building with the highest demand on it. This demand is calculated by adding the numbers of each type of building, and then taking the type of building with the lowest number. Some of these buildings, such as the farm and storage, have a certain "bias" applied to them so that they're built first since they are vital to the survival of the humans early on. Lumberjacks search for the nearest tree, and chop it down once they get to the tree, which once chopped down generates the aforementioned amount of wood. Since a post-apocalyptic simulation would not be complete without some threats, this simulation has an events superclass containing to subclasses that can harm the humans. The first one is the zombie, which will seek the nearest human and head towards it. If it manages to reach the human, it will deal 50 damage per second to the human, and if it manages to drain the human's entire hit point count of 100, the human will then become a zombie and start to attack other humans. Humans will also sometimes randomly become zombies after a certain period has passed. The second event subclass is the tornado, which is spawned at world creation in the top left corner of the world. The tornado moves in a random direction every few seconds, and if it manages to touch a building, tree, or human, the corresponding entity will be immediately destroyed. Credits for artworks used in the simulation: Statbar Class: MrCohen (https://www.greenfoot.org/users/3111) Sprites: Lucy Zhao Graphics: Lucy Zhao, Young Chen Background image: Scribe (https://opengameart.org/content/topdown-tileset) Main menu soundtrack: https://freesound.org/people/tyops/sounds/484301/ Simulation soundtrack: https://freesound.org/people/frankum/sounds/317363/ End screen soundtrack: https://freesound.org/people/hear-no-elvis/sounds/120899/ Sound effects: Button: https://freesound.org/people/Leszek_Szary/sounds/171520/ Building destruction: https://freesound.org/people/ssierra1202/sounds/391961/ Zombie one:

https://freesound.org/people/Under7dude/sounds/163440/ Zombie two:
https://freesound.org/people/nanity05/sounds/193759/ Human hurt:
https://freesound.org/people/AlineAudio/sounds/416839/ Building:
https://freesound.org/people/zbig77/sounds/244985/ Mining:
https://freesound.org/people/michorvath/sounds/270589/ Chopping:
https://freesound.org/people/14FPanskaSilovsky_Petr/sounds/419928/ Sentry fire:
https://freesound.org/people/Bird_man/sounds/275151/

Version:

2020-10-11

Author:

Young Chen

Field Summary

Fields

Modifier and Type	Field	Description
`static int`	**`EASY`**	
`static int`	**`END_DELAY`**	
`static int`	**`HARD`**	
`static int`	**`NORMAL`**	
`static int[]`	**`startHumans`**	

Constructor Summary

Constructors

Constructor	Description
`Simulation``(int difficulty)`	Contructor of Simulation, creates the world where the simulation takes place.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	**`act`**`()`	Intializes the world variables, as well as checks if the simulation has ended

Methods inherited from class greenfoot.World

addObject, getBackground, getCellSize, getColorAt, getHeight, getObjects, getObjectsAt, getWidth, numberOfObjects, removeObject, removeObjects, repaint, setActOrder, setBackground, setBackground, setPaintOrder, showText, started, stopped

Methods inherited from class java.lang.Object

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

Field Detail

EASY

```
public static final int EASY
```

See Also:

Constant Field Values

NORMAL

```
public static final int NORMAL
```

See Also:

Constant Field Values

HARD

```
public static final int HARD
```

See Also:

Constant Field Values

END_DELAY

```
public static final int END_DELAY
```

See Also:

Constant Field Values

startHumans

```
public static int[] startHumans
```

Constructor Detail

Simulation

```
public Simulation(int difficulty)
```

Contructor of Simulation, creates the world where the simulation takes place.

Parameters:

`difficulty` - the difficulty of the simulation

Method Detail

act

```
public void act()
```

Intializes the world variables, as well as checks if the simulation has ended

Overrides:

`act` in class `greenfoot.World`

Class Start

java.lang.Object
    greenfoot.World
        Start

```
public class Start
extends greenfoot.World
```

Start class controls the starting screen of the simulation.

Version:

2020-11-09

Author:

Young Chen, Lucy Zhao, Leo Foo

Constructor Summary

Constructors

Constructor	Description
`Start()`	

Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
void	`act()`	
static void	`playClick()`	Plays the click noise for buttons
void	`started()`	music becomes disoriented if you press start and reset and press start again.
void	`stopped()`	pauses music if Greenfoot is paused.

Methods inherited from class greenfoot.World

`addObject, getBackground, getCellSize, getColorAt, getHeight, getObjects, getObjectsAt, getWidth, numberOfObjects, removeObject, removeObjects, repaint, setActOrder, setBackground, setBackground, setPaintOrder, showText`

Methods inherited from class java.lang.Object

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

Constructor Detail

Start

```
public Start()
```

Method Detail

stopped

```
public void stopped()
```

pauses music if Greenfoot is paused.

Overrides:

`stopped` in class `greenfoot.World`

started

```
public void started()
```

music becomes disoriented if you press start and reset and press start again. somehow music player isnt connected to greenfoot start stop. so when when scenario starts bgMusic should be null since if its not, the jplayer or javazoom whatever will

Overrides:

`started` in class `greenfoot.World`

act

```
public void act()
```

Overrides:

`act` in class `greenfoot.World`

playClick

```
public static void playClick()
```

Plays the click noise for buttons

Class StatBar

java.lang.Object
    greenfoot.Actor
        StatBar

```
public class StatBar
extends greenfoot.Actor
```

New and Improved Stat Bar (Formerly Health Bar). This stat bar can be set to follow an Actor or stay in one place (see constructors). This stat bar may have customized colors, can hide when at full, and can have a customized border. This class aims to be as flexible as possible, allowing it to be simple to use for beginners (easy 0 or 2 parameter constructor) while also highly flexible for those who want to provide more specific parameters (multiple bars in custom colours and sizes with custom offsets and borders).

Implementation - If using multiple bars, all arrays must be the same size. To optimize the appearance choose a height such that:

```
(height - (borderThickness * 2)) % numBars == 0
```

In other words, after factoring out the border, the size should be evenly divisible by the number of bars, so that all bars end up the same size.

Version Notes:

- Now has a boolean to determine whether it will hide itself when Val is full.
- Now has a set of constructors to allow simple and complex implementation.
- 2.1.0 --> Added a border feature, allows customization of thickness and colour

Version:

2.1.0 - 2020 rewrite

Author:

Jordan Cohen, Mr Cohen

Constructor Summary

Constructors

Constructor	Description
`StatBar()`	Main constructor - A basic constructor that sets default values.
`StatBar(int[] maxVal, int[] currVal, greenfoot.Actor owner, int width, int height, int offset, greenfoot.Color[] filledColor, greenfoot.Color[] missingColor, boolean hideAtMax, greenfoot.Color borderColor, int borderThickness)`	The king of all StatBar constuctors! Takes details for an array of bars, otherwise the same as above.
`StatBar(int maxVal, int currVal,`	A simple constructor for a

greenfoot.Actor owner, int width, int height, int offset)	somewhat customized stat bar.
StatBar(int maxVal, int currVal, greenfoot.Actor owner, int width, int height, int offset, greenfoot.Color filledColor, greenfoot.Color missingColor)	Similar to above, but with the ability to customize colors
StatBar(int maxVal, int currVal, greenfoot.Actor owner, int width, int height, int offset, greenfoot.Color filledColor, greenfoot.Color missingColor, boolean hideAtMax)	Similar to above, but with the ability to have the bar hide when full - for example if you don't want full health bars shown.
StatBar(int maxVal, int currVal, greenfoot.Actor owner, int width, int height, int offset, greenfoot.Color filledColor, greenfoot.Color missingColor, boolean hideAtMax, greenfoot.Color borderColor, int borderThickness)	The most detailed constructor! Can specify a border including thickness and color.
StatBar(int maxVal, greenfoot.Actor owner)	A simple constructor - specify a single value (which will be treated as both current and max for the stat) as well as an owner to follow.
StatBar(int maxVal, greenfoot.Actor owner, greenfoot.Color filledColor, greenfoot.Color missingColor)	A simple constructor for a somewhat customized stat bar.
StatBar(int maxVal, greenfoot.Color filledColor, greenfoot.Color missingColor)	A simple constructor for a somewhat customized stat bar.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
int	**getCurrVal**()	Returns the current value of health of the first bar.
int	**getX**()	Return the x location
int	**getY**()	Return the x location
void	**moveMe**()	For projects where efficiency is more important, DELETE THE ACT METHOD and call this directly instead.

void	**setMaxVal** (int[] maxVal)	
void	**update** (int newCurrVal)	
void	**update** (int[] newCurrVal)	update Method: Expects new current Val Returns true if Val has changed (needs an update) Returns false if Val has not changed (to avoid excessive processing)

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

StatBar

```
public StatBar()
```

Main constructor - A basic constructor that sets default values. Easy to use, not very flexible.

StatBar

```
public StatBar(int maxVal,
               greenfoot.Color filledColor,
               greenfoot.Color missingColor)
```

A simple constructor for a somewhat customized stat bar. If owner is null, just position this object where you want it and it wont move. If owner is not null, this object will follow the owner.

Parameters:

`maxVal` - the maximum value for this stat

`currVal` - the starting value for this stat

`filledColor` - the color to be used to represent the current value

`missingColor` - the color to be used to represent the missing value

StatBar

```
public StatBar(int maxVal,
               greenfoot.Actor owner)
```

A simple constructor - specify a single value (which will be treated as both current and max for the stat) as well as an owner to follow. If you do not want this to follow an Actor, use null for the second parameter.

Parameters:

`maxVal` - The maximum value for this stat, which will also be the starting value for this stat

`owner` - The Actor to follow around. If you do not want to associate this with an Actor, provide null instead.

StatBar

```
public StatBar(int maxVal,
               greenfoot.Actor owner,
               greenfoot.Color filledColor,
               greenfoot.Color missingColor)
```

A simple constructor for a somewhat customized stat bar. If owner is null, just position this object where you want it and it wont move. If owner is not null, this object will follow the owner.

Parameters:

`maxVal` - the maximum value for this stat

`currVal` - the starting value for this stat

`owner` - the Actor that this stat bar will follow (null for DONT FOLLOW). Can be changed to just an Actor if needed

`filledColor` - the color to be used to represent the current value

`missingColor` - the color to be used to represent the missing value

StatBar

```
public StatBar(int maxVal,
               int currVal,
```

```
                greenfoot.Actor owner,
                int width,
                int height,
                int offset)
```

A simple constructor for a somewhat customized stat bar. If owner is null, just position this object where you want it and it wont move. If owner is not null, this object will follow the owner.

Parameters:

`maxVal` - the maximum value for this stat

`currVal` - the starting value for this stat

`owner` - the Actor that this stat bar will follow (null for DONT FOLLOW). Can be changed to just an Actor if needed

`width` - the width of the stat bar

`height` - the height of the stat bar

`offset` - the y-offset for positioning this bar in relation to it's owner

StatBar

```
public StatBar(int maxVal,
                int currVal,
                greenfoot.Actor owner,
                int width,
                int height,
                int offset,
                greenfoot.Color filledColor,
                greenfoot.Color missingColor)
```

Similar to above, but with the ability to customize colors

Parameters:

`maxVal` - the maximum value for this stat

`currVal` - the starting value for this stat

`owner` - the Actor that this stat bar will follow (null for DONT FOLLOW). Can be changed to just an Actor if needed

`width` - the width of the stat bar

`height` - the height of the stat bar

`offset` - the y-offset for positioning this bar in relation to it's owner

`filledColor` - the color to be used to represent the current value

`missingColor` - the color to be used to represent the missing value

StatBar

```
public StatBar(int maxVal,
               int currVal,
               greenfoot.Actor owner,
               int width,
               int height,
               int offset,
               greenfoot.Color filledColor,
               greenfoot.Color missingColor,
               boolean hideAtMax)
```

Similar to above, but with the ability to have the bar hide when full - for example if you don't want full health bars shown.

Parameters:

`maxVal` - the maximum value for this stat

`currVal` - the starting value for this stat

`owner` - the Actor that this stat bar will follow (null for DONT FOLLOW). Can be changed to just an Actor if needed

`width` - the width of the stat bar

`height` - the height of the stat bar

`offset` - the y-offset for positioning this bar in relation to it's owner

`filledColor` - the color to be used to represent the current value

`missingColor` - the color to be used to represent the missing value

`hideAtMax` - set to true to have this statBar hide itself when currVal == maxVal

StatBar

```
public StatBar(int maxVal,
               int currVal,
               greenfoot.Actor owner,
               int width,
               int height,
               int offset,
               greenfoot.Color filledColor,
               greenfoot.Color missingColor,
               boolean hideAtMax,
               greenfoot.Color borderColor,
               int borderThickness)
```

The most detailed constructor! Can specify a border including thickness and color.

Parameters:

`maxVal` - the maximum value for this stat

`currVal` - the starting value for this stat

`owner` - the Actor that this stat bar will follow (null for DONT FOLLOW). Can be changed to just an Actor if needed

`width` - the width of the stat bar

`height` - the height of the stat bar

`offset` - the y-offset for positioning this bar in relation to it's owner

`filledColor` - the color to be used to represent the current value

`missingColor` - the color to be used to represent the missing value

`hideAtMax` - set to true to have this statBar hide itself when currVal == maxVal

`borderColor` - the Color of the border

`borderThickness` - the thickness of the border. This value should be at least 1.

StatBar

```
public StatBar(int[] maxVal,
               int[] currVal,
               greenfoot.Actor owner,
               int width,
               int height,
               int offset,
               greenfoot.Color[] filledColor,
               greenfoot.Color[] missingColor,
               boolean hideAtMax,
               greenfoot.Color borderColor,
               int borderThickness)
```

The king of all StatBar constuctors! Takes details for an array of bars, otherwise the same as above. Note that all arrays must be the same length.

Parameters:

`maxVal` - [] the maximum values for each stat

`currVal` - [] the starting values for each stat

`owner` - the Actor that this stat bar will follow (null for DONT FOLLOW). Can be changed to just an Actor if needed

`width` - the width of the stat bar

`height` - the height of the stat bar

`offset` - the y-offset for positioning this bar in relation to it's owner

`filledColor` - [] the colors to be used to represent the current values

`missingColor` - [] the colors to be used to represent the missing values

`hideAtMax` - set to true to have this statBar hide itself when currVal == maxVal

`borderColor` - the Color of the border

`borderThickness` - the thickness of the border. This value should be at least 1.

Method Detail

moveMe

```
public void moveMe()
```

For projects where efficiency is more important, DELETE THE ACT METHOD and call this directly instead. This allows the statBar object to be reactive, only moving when told, rather than acting each act(). For most projects, and especially for beginners, the act method is easier to manage.

update

```
public void update(int newCurrVal)
```

update

```
public void update(int[] newCurrVal)
```

update Method: Expects new current Val Returns true if Val has changed (needs an update) Returns false if Val has not changed (to avoid excessive processing)

getCurrVal

```
public int getCurrVal()
```

Returns the current value of health of the first bar. Method added by Lucy Zhao.

Returns:

int the current value

setMaxVal

```
public void setMaxVal(int[] maxVal)
```

getX

```
public int getX()
```

Return the x location

Overrides:

`getX` in class `greenfoot.Actor`

Returns:

int the x location

getY

```
public int getY()
```

Return the x location

Overrides:

`getY` in class `greenfoot.Actor`

Returns:

int the x location

Class Storage

java.lang.Object
    greenfoot.Actor
        Building
            Storage

```
public class Storage
extends Building
```

Storage sprite for the storage buildingslot

Version:

2020-10-10

Author:

Leo Foo

Field Summary

Fields inherited from class Building

`EMPTY_SPRITE, FARM_SPRITE, HOUSE_SPRITE, MINE_SPRITE, SENTRY_SPRITE, sprite, STORAGE_SPRITE`

Constructor Summary

Constructors

Constructor	Description
`Storage()`	Constructor for the Storage class.

Method Summary

Methods inherited from class Building

`_update, destroy, getNearestEvent, getSprite`

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, getX, getY, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

Storage

```
public Storage()
```

Constructor for the Storage class.

Class Tornado

java.lang.Object
    greenfoot.Actor
        Event
            Tornado

```
public class Tornado
extends Event
```

A tornado that moves around and damages nearby buildings

Version:

2020-11-04

Author:

Young Chen

Field Summary

Fields inherited from class Event

`damage, DEFAULT_DAMAGE, DEFAULT_HP, hp, METEOR, range, rot, TORNADO, type, xLoc, yLoc, ZOMBIE`

Constructor Summary

Constructors

Constructor	Description
`Tornado(int xLoc, int yLoc)`	Creates a tornado at specified location

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`_update()`	Tornado update method

Methods inherited from class Event

damage, die, getBuildingsWithinRange, getHumansWithinRange, getTreesWithinRange, getType, getX, getY, killNearbyThings

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

Tornado

```
public Tornado(int xLoc,
               int yLoc)
```

Creates a tornado at specified location

Parameters:

`xLoc` - Location in x axis of tornado

`yLoc` - Location in y axis of tornado

Method Detail

_update

```
public void _update()
```

Tornado update method

Specified by:

`_update` in class `Event`

Class Tree

java.lang.Object
    greenfoot.Actor
        Tree

```
public class Tree
extends greenfoot.Actor
```

Tree that lumberjacks can murder

Version:

2020-10-09

Author:

Lucy Zhao, Young Chen

Constructor Summary

Constructors

Constructor	Description
`Tree(int x, int y)`	Constructor for objects of class Tree

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	`_update()`	Updates the tree
protected void	`addedToWorld (greenfoot.World world)`	Makes sure trees don't spawn on top of other trees/buildings
void	`chop()`	Removed chopped down trees and updates wood resources
void	`destroy()`	Removes the tree from the world
boolean	`getTargetStatus()`	Returns the targeted status of the tree
int	`getX()`	Returns the x location
int	`getY()`	Returns the y location
void	`setTargetStatus`	Set the targeted status of a tree

```
            (boolean status)
```

Methods inherited from class greenfoot.Actor

`act, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards`

Methods inherited from class java.lang.Object

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

Constructor Detail

Tree

```
public Tree(int x,
            int y)
```

Constructor for objects of class Tree

Method Detail

_update

```
public void _update()
```

Updates the tree

addedToWorld

```
protected void addedToWorld(greenfoot.World world)
```

Makes sure trees don't spawn on top of other trees/buildings

Overrides:

`addedToWorld` in class `greenfoot.Actor`

chop

```
public void chop()
```

Removed chopped down trees and updates wood resources

destroy

```
public void destroy()
```

Removes the tree from the world

setTargetStatus

```
public void setTargetStatus(boolean status)
```

Set the targeted status of a tree

Parameters:

`status` - true if its targeted by a lumberjack, otherwise false

getTargetStatus

```
public boolean getTargetStatus()
```

Returns the targeted status of the tree

Returns:

boolean true if being targeted, otherwise false

getX

```
public int getX()
```

Returns the x location

Overrides:

`getX` in class `greenfoot.Actor`

Returns:

int the x location of the building

getY

```
public int getY()
```

Returns the y location

Overrides:

`getY` in class `greenfoot.Actor`

Returns:

int the y location of the building

Class Utils

java.lang.Object
    Utils

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

Common math helper methods.

Version:

2020-10-07

Author:

Lucy Zhao, Young Chen

Constructor Summary

Constructors

Constructor	Description
`Utils()`	

Method Summary

All Methods Static Methods Concrete Methods

Modifier and Type	Method	Description
`static int`	`calcDist(int ax, int bx, int ay, int by)`	Calculates the distance of two locations.
`static float`	`getAngleTo(int fromX, int toX, int fromY, int toY)`	Calculates the distance of two locations.

Methods inherited from class java.lang.Object

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

Constructor Detail

Utils

```
public Utils()
```

Method Detail

calcDist

```
public static int calcDist(int ax, int bx, int ay, int by)
```

Calculates the distance of two locations.

Parameters:

`ax` - the first x location

`bx` - the second x location

`ay` - the first y location

`by` - the second y location

Returns:

int the distance

getAngleTo

```
public static float getAngleTo(int fromX, int toX, int fromY, int toY)
```

Calculates the distance of two locations.

Parameters:

`fromX` - the x location to calculate from

`toX` - the x location to calculate the angle to

`fromY` - the y location to calculate from

`toX` - the y location to calculate the angle to

Returns:

float the angle in radians

Class WorldManagement

java.lang.Object
    WorldManagement

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

Class to manage all the world events

Version:

2020-11-10

Author:

Leo Foo, Lucy Zhao, Young Chen

Field Summary

Fields

Modifier and Type	Field	Description
static float	armouryDemand	
static java.util.ArrayList	backgrounds	
static int	BUILDING_PADDING	
static int	BUILDING_SIZE	
static java.util.ArrayList	buildings	
static int	CAM_SPEED	
static float	deltaTime	
static int	difficulty	
static int	EASY	
static float	elapsed	
static java.util.ArrayList	events	
static float	farmDemand	
static float	food	
static int	GRID_SEPARATION	
static int	HARD	
static int	highestDemand	
static float	houseDemand	
static float	housing	

static int	**HUMAN_GAP**
static java.util.ArrayList	**humans**
static float	**iron**
static long	**lastTime**
static int	**LIMITER_TIMER**
static int	**MAX_EVENTS**
static int	**MAX_EVENTS_THRESHOLD**
static int	**MAX_TREES**
static float	**mineDemand**
static int	**NORMAL**
static float	**pop**
static ScoreBar	**scoreboard**
static float	**sentryDemand**
static float	**START_FOOD**
static int	**START_FREEZE_FRAMES**
static float	**START_HOUSING**
static float	**START_IRON**
static float	**START_POP**
static float	**START_STORAGE**
static float	**START_WOOD**
static float	**storage**
static float	**storageDemand**
static float	**threatLevel**
static int	**totalArmoury**
static int	**totalBarracks**
static int	**totalBuilders**
static int	**totalFarm**
static int	**totalFarmers**
static int	**totalHouse**
static int	**totalLumberjacks**
static int	**totalMine**
static int	**totalMiners**

static int	**totalSentry**
static int	**totalStorage**
static int	**totalWood**
static int	**TREE_SPAWN_RATE**
static java.util.ArrayList	**trees**
static int	**TYPES_OF_HUMANS**
static float	**wood**
static Simulation	**world**
static int	**WORLD_SIZE**
static int	**ZOMBIE_SPAWN_RATE**
static int	**zombieSpawnRate**

Constructor Summary

Constructors

Constructor	Description
WorldManagement(int worldWidth, int worldHeight, Simulation world)	Constructor of the WorldManagement Class.

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	**_update**()	Methods that updates the world.
static void	**addEvent**(int eventID, int xLoc, int yLoc)	Adds an event to the world.
static void	**addHuman**(int humanID, int xLoc, int yLoc)	Adds a human instance to the world.
int	**generateOffset**()	Generates a

		random offset for buildings
static java.util.ArrayList<BuildingSlot>	**getBuildings**()	Returns an ArrayList of all existing buildings.
static BuildingSlot	**getBuildingSlot**(int index)	Returns the building slot at a specified index.
static float	**getDeltaTime**()	Returns delta time.
static Event	**getEvent**(int index)	Gets the event at specified index
static java.util.ArrayList<Event>	**getEvents**()	Returns an ArrayList of all existing events.
static float	**getFood**()	Returns the amount of food resources.
static Human	**getHuman**(int index)	Returns a human instance at a specified index from the humans Arraylist
static java.util.ArrayList<Human>	**getHumans**()	Returns a list of all the current human instances
static float	**getIron**()	Returns the amount of iron resources.
static java.util.ArrayList<Tree>	**getTrees**()	Returns an ArrayList of

		all existing trees.
static float	**getWood**()	Returns the amount of wood resources.
static Simulation	**getWorld**()	Returns the current world.
static boolean	**hasHousing**()	
void	**init**()	Initializes the assets of the world.
static void	**playSound**(greenfoot.GreenfootSound sound)	Plays sound effects for the world.
void	**setDifficulty**(int difficulty)	Sets the difficulty
static void	**updateFood**(float val)	Updates the food resource.
static void	**updateIron**(float val)	Updates the iron resource.
static void	**updateWood**(float val)	Updates the wood resource.

Methods inherited from class java.lang.Object

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

Field Detail

WORLD_SIZE

public static final int WORLD_SIZE

See Also:

Constant Field Values

CAM_SPEED

```
public static final int CAM_SPEED
```

See Also:

Constant Field Values

GRID_SEPARATION

```
public static final int GRID_SEPARATION
```

See Also:

Constant Field Values

BUILDING_SIZE

```
public static final int BUILDING_SIZE
```

See Also:

Constant Field Values

BUILDING_PADDING

```
public static final int BUILDING_PADDING
```

See Also:

Constant Field Values

TREE_SPAWN_RATE

```
public static final int TREE_SPAWN_RATE
```

See Also:

Constant Field Values

ZOMBIE_SPAWN_RATE

```
public static final int ZOMBIE_SPAWN_RATE
```

See Also:

Constant Field Values

MAX_TREES

```
public static final int MAX_TREES
```

See Also:

Constant Field Values

START_FREEZE_FRAMES

```
public static final int START_FREEZE_FRAMES
```

See Also:

Constant Field Values

TYPES_OF_HUMANS

```
public static final int TYPES_OF_HUMANS
```

See Also:

Constant Field Values

MAX_EVENTS

```
public static final int MAX_EVENTS
```

See Also:

Constant Field Values

HUMAN_GAP

public static final int HUMAN_GAP

See Also:

Constant Field Values

LIMITER_TIMER

public static final int LIMITER_TIMER

See Also:

Constant Field Values

MAX_EVENTS_THRESHOLD

public static final int MAX_EVENTS_THRESHOLD

See Also:

Constant Field Values

EASY

public static final int EASY

See Also:

Constant Field Values

NORMAL

public static final int NORMAL

See Also:

Constant Field Values

HARD

public static final int HARD

See Also:

Constant Field Values

difficulty

```
public static int difficulty
```

zombieSpawnRate

```
public static int zombieSpawnRate
```

humans

```
public static java.util.ArrayList humans
```

buildings

```
public static java.util.ArrayList buildings
```

trees

```
public static java.util.ArrayList trees
```

backgrounds

```
public static java.util.ArrayList backgrounds
```

events

```
public static java.util.ArrayList events
```

world

```
public static Simulation world
```

scoreboard

```
public static ScoreBar scoreboard
```

deltaTime

```
public static float deltaTime
```

elapsed

```
public static float elapsed
```

lastTime

```
public static long lastTime
```

threatLevel

```
public static float threatLevel
```

armouryDemand

```
public static float armouryDemand
```

farmDemand

```
public static float farmDemand
```

mineDemand

```
public static float mineDemand
```

houseDemand

```
public static float houseDemand
```

sentryDemand

```
public static float sentryDemand
```

storageDemand

```
public static float storageDemand
```

highestDemand

```
public static int highestDemand
```

totalArmoury

```
public static int totalArmoury
```

totalBarracks

```
public static int totalBarracks
```

totalWood

```
public static int totalWood
```

totalFarm

```
public static int totalFarm
```

totalMine

```
public static int totalMine
```

totalHouse

```
public static int totalHouse
```

totalSentry

```
public static int totalSentry
```

totalStorage

```
public static int totalStorage
```

totalFarmers

```
public static int totalFarmers
```

totalMiners

```
public static int totalMiners
```

totalBuilders

```
public static int totalBuilders
```

totalLumberjacks

```
public static int totalLumberjacks
```

START_POP

```
public static final float START_POP
```

See Also:

Constant Field Values

START_FOOD

```
public static final float START_FOOD
```

See Also:

Constant Field Values

START_WOOD

```
public static final float START_WOOD
```

See Also:

Constant Field Values

START_IRON

```
public static final float START_IRON
```

See Also:

Constant Field Values

START_STORAGE

```
public static final float START_STORAGE
```

See Also:

Constant Field Values

START_HOUSING

```
public static final float START_HOUSING
```

See Also:

Constant Field Values

pop

```
public static float pop
```

food

```
public static float food
```

wood

```
public static float wood
```

iron

```
public static float iron
```

storage

```
public static float storage
```

housing

```
public static float housing
```

Constructor Detail

WorldManagement

```
public WorldManagement(int worldWidth,
                       int worldHeight,
                       Simulation world)
```

Constructor of the WorldManagement Class. Use to set up the world

Parameters:

`worldWidth` - Width of world

`worldHeight` - Height of world

`world` - Greenfoot World object

Method Detail

init

```
public void init()
```

Initializes the assets of the world. This includes the buildings, scorebar, and humans.

_update

```
public void _update()
```

Methods that updates the world.

setDifficulty

```
public void setDifficulty(int difficulty)
```

Sets the difficulty

Parameters:

`difficulty` - The new difficulty to set to

generateOffset

```
public int generateOffset()
```

Generates a random offset for buildings

Returns:

int the value of the offset

addHuman

```
public static void addHuman(int humanID, int xLoc, int yLoc)
```

Adds a human instance to the world.

Parameters:

`humanID` - the type of human to be added

`xLoc` - the x location of the human

`yLoc` - the y location of the human

addEvent

```
public static void addEvent(int eventID, int xLoc, int yLoc)
```

Adds an event to the world.

Parameters:

`eventID` - the type of human to be added

`xLoc` - the x location of the human

`yLoc` - the y location of the human

getHuman

```
public static Human getHuman(int index)
```

Returns a human instance at a specified index from the humans Arraylist

Parameters:

`index` - the index of the Arraylist

Returns:

Human the human instance at that index

getHumans

```
public static java.util.ArrayList<Human> getHumans()
```

Returns a list of all the current human instances

Returns:

ArrayList the list containing all human instances

getEvent

```
public static Event getEvent(int index)
```

Gets the event at specified index

Parameters:

`index` - Index of event

Returns:

Event at index

hasHousing

```
public static boolean hasHousing()
```

Returns:

hasHousingSpace

getBuildingSlot

```
public static BuildingSlot getBuildingSlot(int index)
```

Returns the building slot at a specified index.

Parameters:

`index` - the index of the ArrayList

Returns:

BuildingSlot the building slot at that index

getBuildings

```
public static java.util.ArrayList<BuildingSlot> getBuildings()
```

Returns an ArrayList of all existing buildings.

Returns:

ArrayList list of all buildings

getEvents

```
public static java.util.ArrayList<Event> getEvents()
```

Returns an ArrayList of all existing events.

Returns:

ArrayList list of all events

getTrees

```
public static java.util.ArrayList<Tree> getTrees()
```

Returns an ArrayList of all existing trees.

Returns:

ArrayList list of all trees

updateFood

```
public static void updateFood(float val)
```

Updates the food resource.

Parameters:

`val` - amount to update by

updateWood

```
public static void updateWood(float val)
```

Updates the wood resource.

Parameters:

`val` - amount to update by

updateIron

```
public static void updateIron(float val)
```

Updates the iron resource.

Parameters:

`val` - amount to update by

getWood

```
public static float getWood()
```

Returns the amount of wood resources.

Returns:

float the amount of wood

getIron

```
public static float getIron()
```

Returns the amount of iron resources.

Returns:

float the amount of iron

getFood

```
public static float getFood()
```

Returns the amount of food resources.

Returns:

float the amount of food

getWorld

```
public static Simulation getWorld()
```

Returns the current world.

Returns:

Simulation the current world

getDeltaTime

```
public static float getDeltaTime()
```

Returns delta time.

Returns:

float the delta time

playSound

```
public static void playSound(greenfoot.GreenfootSound sound)
```

Plays sound effects for the world. Greenfoot sounds can have problems playing, so using try catch prevents any errors from stopping the simulation.

Parameters:

`sound` - the sound to be played

Class YCWidget

java.lang.Object
    greenfoot.Actor
        YCWidget

```
public class YCWidget
extends greenfoot.Actor
```

A selection widget that can be used to visually select one choice from an array of choices through key presses or any other user input method. This widget has a customisable background colour, a customisable highlight colour, and customisable dimensions. Other features of this class than can be adjusted include parenting to an actor's location, mutable selections, auto hide after a certain period of inactivity, and using showing images as the selection choices.

Note: If setting custom dimensions, the dimension must be divisable by the number of selections to avoid spacing issues.

Version:

2020-10-10

Author:

Young Chen

Constructor Summary

Constructors

Constructor	Description
`YCWidget()`	Create a basic YCWidget with the default elements, colours, and settings.
`YCWidget(java.lang.String[] options)`	Create a basic YCWidget with custom elements and the default colours, and settings.
`YCWidget(java.lang.String[] options, int length, int height)`	Create a basic YCWidget with custom elements and dimensions, and the default colours, and settings.
`YCWidget(java.lang.String[] options, int length, int height, boolean useImage)`	Create a basic YCWidget with custom elements, custom dimensions, and the ability to use images

	to represent each element.
YCWidget(java.lang.String[] options, int length, int height, boolean autoHide, int hideDelay, greenfoot.Actor parent, int xOffSet, int yOffSet)	Create a basic YCWidget that has custom elements, has custom dimensions and is parented to another actor's location with a custom x-axis offset and a custom y-axis offset.
YCWidget(java.lang.String[] options, int length, int height, greenfoot.Actor parent)	Create a basic YCWidget with custom elements, with custom dimensions and that is parented to another actor's location, as well as the default colours, and settings.
YCWidget(java.lang.String[] options, int length, int height, greenfoot.Actor parent, int xOffSet, int yOffSet)	Create a basic YCWidget that has custom elements, has custom dimensions and is parented to another actor's location with a custom x-axis offset and a custom y-axis offset.
YCWidget(java.lang.String[] options, greenfoot.Color backgroundColour, greenfoot.Color highlightColour)	Create a basic YCWidget with custom elements, custom background and highlight background colours, and the default settings.
YCWidget(java.lang.String[] options, greenfoot.Color backgroundColour, greenfoot.Color highlightColour, greenfoot.Color dividerColour, greenfoot.Color regularTextColour, greenfoot.Color highlightTextColour)	Create a basic YCWidget with custom elements, fully custom colours, and the default settings.
YCWidget(java.lang.String[] options, greenfoot.Color backgroundColour, greenfoot.Color highlightColour, greenfoot.Color dividerColour, greenfoot.Color regularTextColour, greenfoot.Color highlightTextColour, int length, int height)	Create a basic YCWidget with custom elements, fully custom colours, custom dimensions and the default settings.
YCWidget(java.lang.String[] options, greenfoot.Color backgroundColour,	Create a basic YCWidget with custom elements,

greenfoot.Color highlightColour, greenfoot.Color dividerColour, greenfoot.Color regularTextColour, greenfoot.Color highlightTextColour, int length, int height, int hideDelay, int xOffSet, int yOffSet, java.lang.String scrollDirection, boolean useImage, boolean autoHide, greenfoot.Actor parent)	fully custom colours, custom dimensions, and fully custom settings.
YCWidget(java.lang.String[] options, greenfoot.Color backgroundColour, greenfoot.Color highlightColour, greenfoot.Color dividerColour, greenfoot.Color regularTextColour, greenfoot.Color highlightTextColour, int length, int height, java.lang.String scrollDirection)	Create a basic YCWidget with custom elements, fully custom colours, custom dimensions, and partially custom settings.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
void	**act**()	Actor act method.
int	**getActive**()	Get the index of the current selected element
java.lang.String	**getActiveName**()	Get the name of the current selected element
int	**getHeight**()	Get the height of the YCWidget object
int	**getWidth**()	Get the width of the YCWidget object
void	**hide**()	Hide the YCWidget
boolean	**isActive**(int index)	Check if an element is selected using the index of the element
boolean	**isActive** (java.lang.String name)	Check if an element is selected using the name of the element
boolean	**isShowing**()	Gets whether or not the YCWidget is showing
void	**prolongShowing**(int time)	Temporarily increases the delay if autohide is on
void	**resetDelay**()	Resets the hide delay if autohide is on
void	**shiftSelect**()	Shift the selection one to the right or

		left, depending on the YCWidget's scroll direction
void	**show**()	Show the YCWidget
void	**switchVisibility**()	Makes the widget visible if hidden and hidden if visible
void	**update**(int xOffset, int yOffset)	Set a new x-axis offset and y-axis offset if the YCWidget has a parent Actor.
void	**update** (java.lang.String[] sections)	Update the number of sections and the elements contained in the sections.
void	**update** (java.lang.String[] sections, int width, int height)	Update the number of sections and the elements contained in the sections, along with the width and height.
void	**update** (java.lang.String value, int index)	Replace one element with a new element.

Methods inherited from class greenfoot.Actor

addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, getX, getY, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Constructor Detail

YCWidget

```
public YCWidget()
```

Create a basic YCWidget with the default elements, colours, and settings.

YCWidget

```
public YCWidget(java.lang.String[] options)
```

Create a basic YCWidget with custom elements and the default colours, and settings.

Parameters:

`options` - String array containing the element names to be added to the YCWidget

YCWidget

```
public YCWidget(java.lang.String[] options,
                greenfoot.Color backgroundColour,
                greenfoot.Color highlightColour)
```

Create a basic YCWidget with custom elements, custom background and highlight background colours, and the default settings.

Parameters:

`options` - String array containing the element names to be added to the YCWidget

`backgroundColour` - The colour of the element box when it is not selected

`highlightColour` - The colour of the element box when it is selected

YCWidget

```
public YCWidget(java.lang.String[] options,
                int length,
                int height)
```

Create a basic YCWidget with custom elements and dimensions, and the default colours, and settings.

Parameters:

`options` - String array containing the element names to be added to the YCWidget

`length` - Size of the YCWidget in the x direction

`height` - Size of the YCWidget in the y direction

YCWidget

```
public YCWidget(java.lang.String[] options,
                int length,
                int height,
                greenfoot.Actor parent)
```

Create a basic YCWidget with custom elements, with custom dimensions and that is parented to another actor's location, as well as the default colours, and settings.

Parameters:

`options` - String array containing the element names to be added to the YCWidget

`length` - Size of the YCWidget in the x direction

`height` - Size of the YCWidget in the y direction

`parent` - The actor that the YCWidget is parented to. Input null if no parent is desired

YCWidget

```
public YCWidget(java.lang.String[] options,
                int length,
                int height,
                greenfoot.Actor parent,
                int xOffSet,
                int yOffSet)
```

Create a basic YCWidget that has custom elements, has custom dimensions and is parented to another actor's location with a custom x-axis offset and a custom y-axis offset.

Parameters:

`options` - String array containing the element names to be added to the YCWidget

`length` - Size of the YCWidget in the x direction

`height` - Size of the YCWidget in the y direction

`parent` - The actor that the YCWidget is parented to in regards to its location. Input null if no parent desired

`xOffSet` - The x-axis offset of the widget in relation to the actor. Input any number if no parent desired

`yOffSet` - The y-axis offset of the widget in relation to the actor. Input any number if no parent desired

YCWidget

```
public YCWidget(java.lang.String[] options,
                int length,
                int height,
```

```
                boolean autoHide,
                int hideDelay,
                greenfoot.Actor parent,
                int xOffSet,
                int yOffSet)
```

Create a basic YCWidget that has custom elements, has custom dimensions and is parented to another actor's location with a custom x-axis offset and a custom y-axis offset.

Parameters:

`options` - String array containing the element names to be added to the YCWidget

`length` - Size of the YCWidget in the x direction

`height` - Size of the YCWidget in the y direction

`autoHide` - Whether or not to automatically hide the YCWidget after a certain period of inactivity

`hideDelay` - The time of which the widget will hide itself in milliseconds if there has been no new key presses and that autohide has been enabled. Input any number of autohide will not be enabled

`parent` - The actor that the YCWidget is parented to in regards to its location

`xOffSet` - The x-axis offset of the widget in relation to the parent. Input any number if no parent desired

`yOffSet` - The y-axis offset of the widget in relation to the parent. Input any number if no parent desired

YCWidget

```
public YCWidget(java.lang.String[] options,
                int length,
                int height,
                boolean useImage)
```

Create a basic YCWidget with custom elements, custom dimensions, and the ability to use images to represent each element.

Parameters:

`options` - String array containing the element names to be added to the YCWidget, or if use image is enabled, the string array containing the image file names of the images to be used contained in the image directory of the greenfoot project to represent each element. If no image is found, then the string representation of the element will be used instead

`length` - Size of the YCWidget in the x direction

`height` - Size of the YCWidget in the y direction

`useImage` - Whether or not to use images to represent elements

YCWidget

```
public YCWidget(java.lang.String[] options,
                greenfoot.Color backgroundColour,
                greenfoot.Color highlightColour,
                greenfoot.Color dividerColour,
                greenfoot.Color regularTextColour,
                greenfoot.Color highlightTextColour)
```

Create a basic YCWidget with custom elements, fully custom colours, and the default settings.

Parameters:

`options` - String array containing the element names to be added to the YCWidget

`backgroundColour` - The colour of the element box when it is not selected

`highlightColour` - The colour of the element box when it is selected

`dividerColour` - The colour of the dividing line in between each element

`regularTextColour` - The colour of the name of each element that has not been selected

`highlightTextColour` - The colour of the name of the element that has been selected

YCWidget

```
public YCWidget(java.lang.String[] options,
                greenfoot.Color backgroundColour,
                greenfoot.Color highlightColour,
                greenfoot.Color dividerColour,
                greenfoot.Color regularTextColour,
                greenfoot.Color highlightTextColour,
                int length,
                int height)
```

Create a basic YCWidget with custom elements, fully custom colours, custom dimensions and the default settings.

Parameters:

`options` - String array containing the element names to be added to the YCWidget

`backgroundColour` - The colour of the element box when it is not selected

`highlightColour` - The colour of the element box when it is selected

`dividerColour` - The colour of the dividing line in between each element

`regularTextColour` - The colour of the name of each element that has not been selected

`highlightTextColour` - The colour of the name of the element that has been selected

`length` - Size of the YCWidget in the x direction

`height` - Size of the YCWidget in the y direction

YCWidget

```
public YCWidget(java.lang.String[] options,
                greenfoot.Color backgroundColour,
                greenfoot.Color highlightColour,
                greenfoot.Color dividerColour,
                greenfoot.Color regularTextColour,
                greenfoot.Color highlightTextColour,
                int length,
                int height,
                java.lang.String scrollDirection)
```

Create a basic YCWidget with custom elements, fully custom colours, custom dimensions, and partially custom settings.

Parameters:

`options` - String array containing the element names to be added to the YCWidget

`backgroundColour` - The colour of the element box when it is not selected

`highlightColour` - The colour of the element box when it is selected

`dividerColour` - The colour of the dividing line in between each element

`regularTextColour` - The colour of the name of each element that has not been selected

`highlightTextColour` - The colour of the name of the element that has been selected

`length` - Size of the YCWidget in the x direction

`height` - Size of the YCWidget in the y direction

`scrollDirection` - The name of the direction of which the next selected element will be. Valid directions: "right", "left"

YCWidget

```
public YCWidget(java.lang.String[] options,
                greenfoot.Color backgroundColour,
                greenfoot.Color highlightColour,
                greenfoot.Color dividerColour,
                greenfoot.Color regularTextColour,
                greenfoot.Color highlightTextColour,
                int length,
                int height,
                int hideDelay,
                int xOffSet,
                int yOffSet,
```

```
                java.lang.String scrollDirection,
                boolean useImage,
                boolean autoHide,
                greenfoot.Actor parent)
```

Create a basic YCWidget with custom elements, fully custom colours, custom dimensions, and fully custom settings.

Parameters:

`options` - String array containing the element names to be added to the YCWidget, or if use image is enabled, the string array containing the image file names of the images to be used contained in the image directory of the greenfoot project to represent each element. If no image is found, then the string representation of the element will be used instead

`backgroundColour` - The colour of the element box when it is not selected

`highlightColour` - The colour of the element box when it is selected

`dividerColour` - The colour of the dividing line in between each element

`regularTextColour` - The colour of the name of each element that has not been selected

`highlightTextColour` - The colour of the name of the element that has been selected

`length` - Size of the YCWidget in the x direction

`height` - Size of the YCWidget in the y direction

`hideDelay` - The time of which the widget will hide itself in milliseconds if there has been no new key presses and that autohide has been enabled. Input any number of autohide will not be enabled

`xOffSet` - The x-axis offset of the widget in relation to the parent. Input any number if no parent desired

`yOffSet` - The y-axis offset of the widget in relation to the parent. Input any number if no parent desired

`scrollDirection` - The name of the direction of which the next selected element will be. Valid directions: "right", "left"

`useImage` - Whether or not to use images to represent elements

`autoHide` - Whether or not to automatically hide the YCWidget after a certain period of inactivity

`parent` - The actor that the YCWidget will be parented to in relation to its location. Input null if no parent is desired.

Method Detail

act

```
public void act()
```

Actor act method.

Overrides:

`act` in class `greenfoot.Actor`

update

```
public void update(java.lang.String[] sections)
```

Update the number of sections and the elements contained in the sections. The new number of sections must divide evenly with the width.

Parameters:

`sections` - The String array with the new elements to be inserted into the YCWidget, or the new file names of the image representations of each element if use image has been enabled

update

```
public void update(java.lang.String[] sections, int width, int height)
```

Update the number of sections and the elements contained in the sections, along with the width and height. The new number of sections must divide evenly with the specified width.

Parameters:

`sections` - The String array with the new elements to be inserted into the YCWidget, or the new file names of the image representations of each element if use image has been enabled

`width` - The new width of the YCWidget. Input -1 to keep original width

`height` - The new height of the YCWidget. Input -1 to keep original height

update

```
public void update(java.lang.String value, int index)
```

Replace one element with a new element.

Parameters:

`value` - The new value of the element

`index` - The index of the new value

update

```
public void update(int xOffset, int yOffset)
```

Set a new x-axis offset and y-axis offset if the YCWidget has a parent Actor.

Parameters:

`xOffset` - new x-axis offset

`yOffset` - new y-axis offset

isActive

```
public boolean isActive(java.lang.String name)
```

Check if an element is selected using the name of the element

Parameters:

`name` - The name of the element

Returns:

Whether the element is active. False if element is not selected and true if element is selected

isActive

```
public boolean isActive(int index)
```

Check if an element is selected using the index of the element

Parameters:

`index` - The index of the element

Returns:

Whether the element is active. False if element is not selected and true if element is selected

getActive

```
public int getActive()
```

Get the index of the current selected element

Returns:

Index of the current selected element. Will return -1 if no element has been selected.

getActiveName

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

Get the name of the current selected element

Returns:

Name of the current selected element. Will return "" if no element has been selected.

hide

```
public void hide()
```

Hide the YCWidget

show

```
public void show()
```

Show the YCWidget

isShowing

```
public boolean isShowing()
```

Gets whether or not the YCWidget is showing

Returns:

Whether or not the YCWidget is showing

prolongShowing

```
public void prolongShowing(int time)
```

Temporarily increases the delay if autohide is on

Parameters:

`time` - Amount of milliseconds the current increase the delay by

resetDelay

```
public void resetDelay()
```

Resets the hide delay if autohide is on

getWidth

```
public int getWidth()
```

Get the width of the YCWidget object

Returns:

Width of widget

getHeight

```
public int getHeight()
```

Get the height of the YCWidget object

Returns:

Height of widget

shiftSelect

```
public void shiftSelect()
```

Shift the selection one to the right or left, depending on the YCWidget's scroll direction

switchVisibility

```
public void switchVisibility()
```

Makes the widget visible if hidden and hidden if visible

Class Zombie

java.lang.Object
    greenfoot.Actor
        Event
            Enemy
                Zombie

```
public class Zombie
extends Enemy
```

Class of enemy that chases down and attacks humans, and if they manage to kill a human, that human turns into a zombie too.

Version:

2020-10-09

Author:

Young Chen

Field Summary

Fields

Modifier and Type	Field	Description
`static greenfoot.GreenfootSound`	**`zombieOne`**	
`static greenfoot.GreenfootSound`	**`zombieTwo`**	

Fields inherited from class Event

`damage, DEFAULT_DAMAGE, DEFAULT_HP, hp, METEOR, range, rot, TORNADO, type, xLoc, yLoc, ZOMBIE`

Constructor Summary

Constructors

Constructor	Description
`Zombie``(int xLoc, int yLoc)`	Create a zombie at specified location

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
void	**_update**()	Zombie update method
void	**die**()	Causes the zombie to die.

Methods inherited from class Enemy

getNearestHuman

Methods inherited from class Event

damage, getBuildingsWithinRange, getHumansWithinRange, getTreesWithinRange, getType, getX, getY, killNearbyThings

Methods inherited from class greenfoot.Actor

act, addedToWorld, getImage, getIntersectingObjects, getNeighbours, getObjectsAtOffset, getObjectsInRange, getOneIntersectingObject, getOneObjectAtOffset, getRotation, getWorld, getWorldOfType, intersects, isAtEdge, isTouching, move, removeTouching, setImage, setImage, setLocation, setRotation, turn, turnTowards

Methods inherited from class java.lang.Object

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

Field Detail

zombieOne

```
public static final greenfoot.GreenfootSound zombieOne
```

zombieTwo

```
public static final greenfoot.GreenfootSound zombieTwo
```

Constructor Detail

Zombie

```
public Zombie(int xLoc,
              int yLoc)
```

Create a zombie at specified location

Parameters:

`xLoc` - Location in x-axis of zombie

`yLoc` - Location in y-axis of zombie

Method Detail

_update

```
public void _update()
```

Zombie update method

Specified by:

`_update` in class `Event`

die

```
public void die()
```

Causes the zombie to die.

Overrides:

`die` in class `Event`