Class Arrow

java.lang.Object
    greenfoot.Actor
        Arrow

```
public class Arrow
extends greenfoot.Actor
```

Arrow is a simple Greenfoot Actor used in the class PopUp It displays a triangle that points up, down, left, or right.

Version:

2 (15/04/2020)

Author:

Evan Ng

Constructor Summary

Constructors

Constructor	Description
`Arrow()`	Main constructor - sets all the variables to default values.
`Arrow(int width, int height, int borderWidth, int orientation)`	Constructor that takes four ints - sets the dimensions of the arrow, the border, and the direction it points.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`setTransparency(int transparency)`	Sets the transparency of the image.

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

Arrow

```
public Arrow()
```

Main constructor - sets all the variables to default values. Mainly for testing purposes

Arrow

```
public Arrow(int width,
             int height,
             int borderWidth,
             int orientation)
```

Constructor that takes four ints - sets the dimensions of the arrow, the border, and the direction it points.

Parameters:

`width` - Length of base of arrow

`height` - Height of arrow, distance from base to tip of arrow

`borderWidth` - Thickness of border

`orientation` - Direction arrow points, 0 = down, 1 = left, 2 = up, 3 = right

Method Detail

setTransparency

```
public void setTransparency(int transparency)
```

Sets the transparency of the image.

Parameters:

`transparency` - Transparency of the image. A value in the range 0 to 255. 0 is completely transparent and 255 is completely opaque.

Class Bug

```
java.lang.Object
    greenfoot.Actor
        Pest
            Bug
```

```
public class Bug
extends Pest
```

Bug is a Greenfoot Actor that moves around the screen. Bug flies randomly within a given boundary within a given speed miniumum and limit. It also reduces nearby customers' satisfaction.

Version:

1.3 (15/03/2020)

Author:

Evan Ng

Field Summary

Fields inherited from class Pest

`botherRange, botherRate, changeFrequency, count, dead, image, imageHeight, imageWidth, isTargeted, maxSpeed, maxXpos, maxYpos, minSpeed, minXpos, minYpos, shrinkSpeed, xSpeed, ySpeed`

Constructor Summary

Constructors

Constructor	Description
`Bug()`	Main constructor - sets all the variables to default values.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`act()`	Act - do whatever the Bug wants to do.

`void`	`die()`	Bug will start dying.
`protected void`	`shrinkUntilDelete()`	When Bug dies, Bug will shrink until it reaches 1 pixel wide and 1 pixel tall.

Methods inherited from class Pest

`botherCustomer, changeSpeed, getDead, getIsTargeted, setIsTargeted, stayInLimit`

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

Bug

```
public Bug()
```

Main constructor - sets all the variables to default values.

Method Detail

act

```
public void act()
```

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

Overrides:

`act` in class `greenfoot.Actor`

shrinkUntilDelete

```
protected void shrinkUntilDelete()
```

When Bug dies, Bug will shrink until it reaches 1 pixel wide and 1 pixel tall. It then removes itself. The bigger shrinkSpeed is, the faster Bug will shrink. However, shrinkSpeed must be greater than 1.

Specified by:

`shrinkUntilDelete` in class `Pest`

die

```
public void die()
```

Bug will start dying. It stops moving, shrinks, and deletes itself. Before shrinking, it grows to 1.5 times its size for an instant for a nicer death animation (using a fundamental animation principle: "anticipation").

Specified by:

`die` in class `Pest`

Class Button

java.lang.Object
greenfoot.Actor
Button

```
public class Button
extends greenfoot.Actor
```

Button is a simple Greenfoot Actor that displays a line of text in a rectangular display which can have rounded corners.

Version:

2 (23/04/2020)

Author:

Evan Ng

Constructor Summary

Constructors

Constructor	Description
`Button()`	Main constructor - sets all the variables to default values.
`Button(int width, int height, int cornerRadius, int borderThickness, int fontSize, greenfoot.Color backColor, greenfoot.Color borderColor)`	Constructor that takes five ints and two Colors - for adjusting size and colors of the display: width and height of the display, the border, and font size.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`setText(java.lang.String text)`	Sets the text.
`void`	`update(int width, int height, int cornerRadius, int borderThickness, int fontSize, greenfoot.Color backColor, greenfoot.Color borderColor)`	Updates the size and colors of the display

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

Button

```
public Button()
```

Main constructor - sets all the variables to default values. This is mostly used for testing purposes.

Button

```
public Button(int width,
              int height,
              int cornerRadius,
              int borderThickness,
              int fontSize,
              greenfoot.Color backColor,
              greenfoot.Color borderColor)
```

Constructor that takes five ints and two Colors - for adjusting size and colors of the display: width and height of the display, the border, and font size.

Parameters:

`width` - Width of display

`height` - Height of display

`cornerRadius` - Radius of curve on the corners

`borderThickness` - Width of border

`fontSize` - Size of the text

`backColor` - Color of background

`borderColor` - Color of border

Method Detail

setText

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

Sets the text. The text will be displayed as a single line.
In order to center the text, add spaces to the string. Add spaces to the front of the text if the text is displayed too far the left and vice versa if it is displayed too far to the right.
Ex: popUp.setText("TEXT WORDS ");
Note: Text cannot be centered automatically because Greenfoot cannot accurately determine the widths of texts (especially when using different fonts).

Parameters:

`text` - String that will be displayed at the top of the pop up

update

```
public void update(int width, int height, int cornerRadius, int borderThickness,
int fontSize, greenfoot.Color backColor, greenfoot.Color borderColor)
```

Updates the size and colors of the display

Parameters:

`width` - Width of display

`height` - Height of display

`cornerRadius` - Radius of curve on the corners

`borderThickness` - Width of border

`fontSize` - Size of the font

`backColor` - Color of background

`borderColor` - Color of border

Class Client

java.lang.Object
    greenfoot.Actor
        Client

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

Client is the superclass for critic and client. Client, once spawned, will claim a seat, and walk a determined path towards their seat. Once seated, the Client waits for food. If they wait too long, they leave without paying. If they get their food in time, they eat for as long as eatSpeed determines, pay, and leave in a determined path. Raymond has completed most of this class, Evan has made minor adjustments.

Version:

5 (14/05/2020)

Author:

Raymond Zhang

Field Summary

Fields

Modifier and Type	Field	Description
`protected int`	`eatSpeed`	
`protected int`	`eatTimer`	
`protected Food`	`food`	
`protected greenfoot.GreenfootImage`	`image`	
`protected int`	`moveSpeed`	
`protected int`	`patience`	
`protected HealthBar`	`patienceBar`	
`protected boolean`	`reachedSeat`	
`protected boolean`	`receivedFood`	
`protected double`	`satisfaction`	
`protected Seat`	`seat`	
`protected int`	`serviceSpeed`	
`protected int`	`tableNumber`	

Constructor Summary

Constructors

Constructor	Description
`Client(int patience, int eatSpeed)`	A constructor that takes two ints - determines how patient a client is and how fast they will eat.

Method Summary

All Methods Instance Methods Abstract Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`act()`	Act - do whatever the Client wants to do.
`void`	`addedToWorld(greenfoot.World w)`	Adds the patience bar to the Client and updates it
`void`	`addFood()`	Adds a Food object for the customer to "eat"
`boolean`	`atWorldEdge()`	Detects if Client is at the edge of the world.
`void`	`eat()`	When called, the client will eat.
`void`	`enterWorld()`	When called, Client will enter the world by moving.
`void`	`findSeat()`	This is the path the Client must follow to reach their Seat.
`abstract void`	`finishMeal()`	Abstract method that is called once the client finishes their meal.
`int`	`getPatience()`	Gets the patience of the Client
`boolean`	`getReachedSeat()`	Returns whether or not Client reached their seat.
`boolean`	`getReceivedFood()`	Returns whether or not CLient received food from the chef.
`int`	`getSatisfaction()`	Returns the satisfcation of the Client.
`Seat`	`getSeat()`	Gets the seat of which the client claims
`void`	`leaveStore()`	The pathway of which client will leave the store.
`void`	`setPatience(int patience)`	Sets the patience of the Client

void	`setReceivedFood(boolean receivedFood, greenfoot.World w)`	Determines whether or not Client received food from the chef.
void	`setSeat(Seat s)`	Sets the seat of which the client claims

Methods inherited from class greenfoot.Actor

`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

satisfaction

```
protected double satisfaction
```

patienceBar

```
protected HealthBar patienceBar
```

patience

```
protected int patience
```

tableNumber

```
protected int tableNumber
```

seat

```
protected Seat seat
```

reachedSeat

```
protected boolean reachedSeat
```

receivedFood

```
protected boolean receivedFood
```

serviceSpeed

```
protected int serviceSpeed
```

moveSpeed

```
protected int moveSpeed
```

eatSpeed

```
protected int eatSpeed
```

eatTimer

```
protected int eatTimer
```

food

```
protected Food food
```

image

```
protected greenfoot.GreenfootImage image
```

Constructor Detail

Client

```
public Client(int patience,
              int eatSpeed)
```

A constructor that takes two ints - determines how patient a client is and how fast they will eat.

Parameters:

`patience` - How long a client can wait.

`eatSpeed` - How fast a client can eat.

Method Detail

finishMeal

```
public abstract void finishMeal()
```

Abstract method that is called once the client finishes their meal. (Pay and/or change star rating).

act

```
public void act()
```

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

Overrides:

`act` in class `greenfoot.Actor`

setSeat

`public void setSeat(Seat s)`

Sets the seat of which the client claims

Parameters:

`s` - The seat the Client will sit at.

getSeat

`public Seat getSeat()`

Gets the seat of which the client claims

Returns:

Seat The seat the Client will sit at.

addedToWorld

`public void addedToWorld(greenfoot.World w)`

Adds the patience bar to the Client and updates it

Overrides:

`addedToWorld` in class `greenfoot.Actor`

Parameters:

`w` - The World Client is in.

enterWorld

`public void enterWorld()`

When called, Client will enter the world by moving.

eat

```
public void eat()
```

When called, the client will eat. Adds food if Client starts eating. Counts down based on eatSpeed and will finish the customer's meal once done.

atWorldEdge

```
public boolean atWorldEdge()
```

Detects if Client is at the edge of the world.

leaveStore

```
public void leaveStore()
```

The pathway of which client will leave the store. Must be called continuously so Client can follow through.

getSatisfaction

```
public int getSatisfaction()
```

Returns the satisfcation of the Client. This value is used to determine how much Client will pay and how much Critic will change the star rating.

Returns:

int How satisfied client is. 1 is least satisfied. 3 is most.

setReceivedFood

```
public void setReceivedFood(boolean receivedFood, greenfoot.World w)
```

Determines whether or not Client received food from the chef.

Parameters:

`receivedFood` - Whether or not Client received food

`w` - The World Client is in.

getReceivedFood

```
public boolean getReceivedFood()
```

Returns whether or not CLient received food from the chef.

Returns:

boolean Wheter or not Client received food

setPatience

```
public void setPatience(int patience)
```

Sets the patience of the Client

Parameters:

`patience` - How much Client can wait.

getPatience

```
public int getPatience()
```

Gets the patience of the Client

Returns:

int How much Client can wait.

getReachedSeat

```
public boolean getReachedSeat()
```

Returns whether or not Client reached their seat.

Returns:

boolean Wheter or not Client reached their seat (true if reached the Seat)

addFood

```
public void addFood()
```

Adds a Food object for the customer to "eat"

findSeat

```
public void findSeat()
```

This is the path the Client must follow to reach their Seat. Must be called continuously for Client to follow through.

Class CloseButton

java.lang.Object
    greenfoot.Actor
        CloseButton

```
public class CloseButton
extends greenfoot.Actor
```

CloseButton is a Greenfoot Actor used in the class PopUp. This is just a simple X.

Version:

1 (11/04/2020)

Author:

Evan Ng

Constructor Summary

Constructors

Constructor	Description
`CloseButton()`	Main constructor - sets all the variables to default values.

Method Summary

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

CloseButton

```
public CloseButton()
```

Main constructor - sets all the variables to default values.

Class Cook

java.lang.Object
    greenfoot.Actor
        Cook

```
public class Cook
extends greenfoot.Actor
```

Cook will make food at given intervals. If there is not customer to give the food to, Cook will wait until there is. Cook will display the progress of the "cooking". Images from https://iconscout.com.

Version:

5 (14/05/2020)

Author:

Amy Cheung, Evan Ng

Constructor Summary

Constructors

Constructor	Description
`Cook()`	Main constructor - Sets all the variables for cook and sets cookSpeed to a default value.
`Cook(int cookSpeed)`	Constructor that takes on int - will change cookSpeed.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`act()`	Act - do whatever the Cook wants to do.
`void`	`setLeaving(boolean leaving)`	Sets whether or not Cook is leaving the restaurant
`void`	`setReadyToCook(boolean readyToCook)`	Sets whether or not Cook is ready to make food or not.

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

Cook

```
public Cook()
```

Main constructor - Sets all the variables for cook and sets cookSpeed to a default value.

Cook

```
public Cook(int cookSpeed)
```

Constructor that takes on int - will change cookSpeed.

Parameters:

`cookSpeed` - How fast the cook can make food.

Method Detail

act

```
public void act()
```

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

Overrides:

`act` in class `greenfoot.Actor`

setReadyToCook

```
public void setReadyToCook(boolean readyToCook)
```

Sets whether or not Cook is ready to make food or not. Cook is not ready if Cook is leaving or entering the restaurant.

Parameters:

`readyToCook` - Whether or not Cook is ready to cook (true is ready to cook)

setLeaving

```
public void setLeaving(boolean leaving)
```

Sets whether or not Cook is leaving the restaurant

Parameters:

`leaving` - Whether or not Cook is leaving the restaurant (true is leaving)

Class Critic

java.lang.Object
    greenfoot.Actor
        Client
            Critic

```
public class Critic
extends Client
```

Critic will not only pay, but will also change the star rating of the restaurant. Critic pays more than Customer. Images from https://iconscout.com.

Version:

2 (14/05/2020)

Author:

Raymond Zhang

Field Summary

Fields inherited from class Client

`eatSpeed, eatTimer, food, image, moveSpeed, patience, patienceBar, reachedSeat, receivedFood, satisfaction, seat, serviceSpeed, tableNumber`

Constructor Summary

Constructors

Constructor	Description
`Critic(int patience, int eatSpeed)`	Constructor that takes two ints - How long a critic can wait and how fast they can eat

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`act()`	Act - do whatever the Client wants to do.

void	finishMeal()	Will change star rating upon finishing food.

Methods inherited from class Client

addedToWorld, addFood, atWorldEdge, eat, enterWorld, findSeat, getPatience, getReachedSeat, getReceivedFood, getSatisfaction, getSeat, leaveStore, setPatience, setReceivedFood, setSeat

Methods inherited from class greenfoot.Actor

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

Critic

```
public Critic(int patience,
              int eatSpeed)
```

Constructor that takes two ints - How long a critic can wait and how fast they can eat

Parameters:

patience - How long Critic can wait for food to arrive

eatSpeed - How fast Critic can eat

Method Detail

act

```
public void act()
```

Description copied from class: `Client`

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

Overrides:

`act` in class `Client`

finishMeal

```
public void finishMeal()
```

Will change star rating upon finishing food. Will also pay for food.

Specified by:

`finishMeal` in class `Client`

Class Customer

java.lang.Object
    greenfoot.Actor
        Client
            Customer

```
public class Customer
extends Client
```

Customer is a subclass of Client. Customer will appear in one of four images. After eating, Customer will pay the restaurant. Images from https://iconscout.com.

Version:

3 (14/05/2020)

Author:

Raymond Zhang

Field Summary

Fields inherited from class Client

`eatSpeed, eatTimer, food, image, moveSpeed, patience, patienceBar, reachedSeat, receivedFood, satisfaction, seat, serviceSpeed, tableNumber`

Constructor Summary

Constructors

Constructor	Description
`Customer(int patience, int eatSpeed)`	Constructor that takes two ints - How long a critic can wait and how fast they can eat

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`act()`	Act - do whatever the Client wants to do.

`void`	`finishMeal()`	Customer will pay for their meal once they finish it.

Methods inherited from class Client

`addedToWorld, addFood, atWorldEdge, eat, enterWorld, findSeat, getPatience, getReachedSeat, getReceivedFood, getSatisfaction, getSeat, leaveStore, setPatience, setReceivedFood, setSeat`

Methods inherited from class greenfoot.Actor

`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

Customer

```
public Customer(int patience,
                int eatSpeed)
```

Constructor that takes two ints - How long a critic can wait and how fast they can eat

Parameters:

`patience` - How long Critic can wait for food to arrive

`eatSpeed` - How fast Critic can eat

Method Detail

act

```
public void act()
```

Description copied from class: `Client`

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

Overrides:

`act` in class `Client`

finishMeal

```
public void finishMeal()
```

Customer will pay for their meal once they finish it.

Specified by:

`finishMeal` in class `Client`

Class DotCarousel

java.lang.Object
    greenfoot.Actor
        DotCarousel

```
public class DotCarousel
extends greenfoot.Actor
```

DotCarousel is a Greenfoot Actor used in the class PopUp. This is set of dots that are used to indicate which slide the user is on. These slides are represented by dots.
The slide the user is on will be represented by a highlighted dot while the other dots are more transparent. The highlighted dot can be changed.

Version:

1 (12/04/2020)

Author:

Evan Ng

Constructor Summary

Constructors

Constructor	Description
`DotCarousel()`	Main constructor - sets all the variables to default values.
`DotCarousel(int numberOfSlides, int currentSlide)`	Constructor that takes two ints - for setting the number of dots (slides) and the slide currently displayed.
`DotCarousel(int numberOfSlides, int currentSlide, int dotDiameter, int borderWidth)`	Constructor that takes four ints - for setting the number of dots (slides), the slide currently displayed and the size of the dots.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`setCurrentSlide(int currentSlide)`	Adjusts the highlighted dot.

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

DotCarousel

```
public DotCarousel()
```

Main constructor - sets all the variables to default values.

DotCarousel

```
public DotCarousel(int numberOfSlides,
                   int currentSlide)
```

Constructor that takes two ints - for setting the number of dots (slides) and the slide currently displayed.

Parameters:

`numberOfSlides` - Number of dots

`currentSlide` - Which dot is highlighted

DotCarousel

```
public DotCarousel(int numberOfSlides,
                   int currentSlide,
                   int dotDiameter,
                   int borderWidth)
```

Constructor that takes four ints - for setting the number of dots (slides), the slide currently displayed and the size of the dots.

Parameters:

`numberOfSlides` - Number of dots

`currentSlide` - Which dot is highlighted

`dotDiameter` - Diameter of the dot

`borderWidth` - Width of border on dot

Method Detail

setCurrentSlide

```
public void setCurrentSlide(int currentSlide)
```

Adjusts the highlighted dot. Must be greater than 0 and less than or equal to the number of slides.

Parameters:

`currentSlide` - Which dot is highlighted

Class EndWorld

java.lang.Object
greenfoot.World
EndWorld

```
public class EndWorld
extends greenfoot.World
```

This is the end world, which appears at the end of the simulation. This will display a score that is calculated based on the starting and current funds as well as the star rating. Sounds from https://tabletopaudio.com.

Version:

2 (14/05/2020)

Author:

Amy Cheung, Alex Mar, Evan Ng, Raymond Zhang

Constructor Summary

Constructors

Constructor	Description
`EndWorld(int startingFunds, int currentFunds, double starRating)`	Constructor for objects of class EndWorld.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`act()`	Act - do whatever the PopUp wants to do.

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

EndWorld

```
public EndWorld(int startingFunds,
                int currentFunds,
                double starRating)
```

Constructor for objects of class EndWorld. Takes two ints and one double (the starting and current funds, and star rating in order to calculate the score.

Parameters:

`startingFunds` - Money at the start of the simulation

`currentFunds` - Money at the end of the simulation

`starRating` - Star rating at the end of the simulation

Method Detail

act

```
public void act()
```

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

Overrides:

`act` in class `greenfoot.World`

Class Food

java.lang.Object
greenfoot.Actor
Food

```
public class Food
extends greenfoot.Actor
```

This is simply an image of a food item. The size of the image may be changed. Images from https://hotemoji.com.

Version:

1 (13/05/20)

Author:

Evan Ng

Constructor Summary

Constructors

Constructor	Description
`Food()`	Main constructor - sets all the variables to default values.
`Food (int imageWidth)`	Constructor that takes one int - this will change the size of the image based on the given int.

Method Summary

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

Food

```
public Food()
```

Main constructor - sets all the variables to default values. This is simply an image of a random food item.

Food

```
public Food(int imageWidth)
```

Constructor that takes one int - this will change the size of the image based on the given int.

Class HealthBar

java.lang.Object
    greenfoot.Actor
        HealthBar

```
public class HealthBar
extends greenfoot.Actor
```

HealthBar generates and maintains a custom-sized image of a Health Bar in green and red (red being missing HP). Accepts any size - use Constants. NOT FINISHED! Forgive my lack of commenting...

Version:

0.0.1 (2012)

Author:

Jordan Cohen

Constructor Summary

Constructors

Constructor	Description
`HealthBar()`	Main constructor - this is only called by the other Constructors and is not intended to be called directly.
`HealthBar(int inMaxHP)`	Constructor that takes one int - for objects starting with max hit points.
`HealthBar(int inMaxHP, int inCurrHP)`	Constructor that takes in a different value for current and max HP, ideal for when a new health bar is needed for an Actor that doesn't have full HP.
`HealthBar(int inMaxHP, greenfoot.Actor target)`	Constructor takes an int for current and max hitpoints and also takes in an Actor, which this HP Bar will follow - Whenever the Actor moves, so will this hp bar.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`act()`	Act - do whatever the HealthBar wants to do.
`void`	`update`	update Method: Expects new current HP Returns true if HP has changed

`(int newCurrHP)`	(needs an update) Returns false if HP has not changed (to avoid excessive processing)

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

HealthBar

```
public HealthBar()
```

Main constructor - this is only called by the other Constructors and is not intended to be called directly.

HealthBar

```
public HealthBar(int inMaxHP)
```

Constructor that takes one int - for objects starting with max hit points. This will set both current and maximum hit points to the same value.

HealthBar

```
public HealthBar(int inMaxHP,
                 greenfoot.Actor target)
```

Constructor takes an int for current and max hitpoints and also takes in an Actor, which this HP Bar will follow - Whenever the Actor moves, so will this hp bar. If the Actor is removed from the World, this HP bar will destroy itself. NOTE: This is the Constructor used in the Bug simulation

HealthBar

```
public HealthBar(int inMaxHP,
                 int inCurrHP)
```

Constructor that takes in a different value for current and max HP, ideal for when a new health bar is needed for an Actor that doesn't have full HP.

Method Detail

act

```
public void act()
```

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

Overrides:

`act` in class `greenfoot.Actor`

update

```
public void update(int newCurrHP)
```

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

Class HealthBarTwo

java.lang.Object
    greenfoot.Actor
        HealthBarTwo

```
public class HealthBarTwo
extends greenfoot.Actor
```

HealthBar is a Greenfoot Actor that displays an image displaying the current health of an Actor. The bar is green when the Actor has health and red when the Actor is missing health. The hit point number is displayed using black text.

Values can be inputed in the constructor to make the HealthBar customizable to the game or the Actor. Update methods allows the HealthBar to accurately represent the exact hit points of the Actor as the scenario/game progresses; also allowing for color customization.

Version:

1.0.5 March 10, 2020

Author:

Alex Mar, Amy Cheung

Constructor Summary

Constructors

Constructor	Description
`HealthBarTwo()`	Main constructor - this is only called by the other Constructor and is not intended to be called directly.
`HealthBarTwo(int inMaxHP)`	Takes one int and sets max hit points and current hit points to the same value
`HealthBarTwo(int inMaxHP, int inCurHP)`	Takes two values for current and max hit points, allows actor to have a hit point value that is not the max HP
`HealthBarTwo(int inMaxHP, greenfoot.Actor target)`	Takes one int and set max hit points and current hit points to the same value

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description

void	act()	Act - do whatever the HealthBar wants to do.
void	update(int newCurHP)	Updates health bar to accurately represent the current hit points of the Actor as the scenario/game progresses.
void	update(int newCurHP, int newMaxHP)	Changes the color of the health bar.

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

HealthBarTwo

```
public HealthBarTwo()
```

Main constructor - this is only called by the other Constructor and is not intended to be called directly.

HealthBarTwo

```
public HealthBarTwo(int inMaxHP,
                    int inCurHP)
```

Takes two values for current and max hit points, allows actor to have a hit point value that is not the max HP

Parameters:

`inMaxHP` - New max hit points

`inCurHP` - New current hit points

HealthBarTwo

```
public HealthBarTwo(int inMaxHP)
```

Takes one int and sets max hit points and current hit points to the same value

Parameters:

`inMaxHP` - New max hit points

HealthBarTwo

```
public HealthBarTwo(int inMaxHP,
                    greenfoot.Actor target)
```

Takes one int and set max hit points and current hit points to the same value

Assigns a health bar that follows an actor

If actor is no longer in the world, the health bar will also be removed

Parameters:

`inMaxHP` - New max hit points

`target` - An actor that the health bar will be assigned to

Method Detail

act

```
public void act()
```

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

Overrides:

`act` in class `greenfoot.Actor`

update

```
public void update(int newCurHP, int newMaxHP)
```

Changes the color of the health bar.

Parameters:

`newCurHP` - New current hit points

`newMaxHP` - New max hit points

update

```
public void update(int newCurHP)
```

Updates health bar to accurately represent the current hit points of the Actor as the scenario/game progresses.

Parameters:

`newCurHP` - New current hit points

Class HealthInspector

java.lang.Object
    greenfoot.Actor
        HealthInspector

```
public class HealthInspector
extends greenfoot.Actor
```

Health Inspector will enter the restaurant and linger for a while. During this time, Health Inspector will fine the resturant if they see a pest and may reduce the star rating. When Health Inspector spots a rat, they will immediately leave, fine, and reduce star rating by 1.0. Amy completed the majority of this class. Evan made minor adjustments. Images from https://iconscout.com.

Version:

3 (12/05/2020)

Author:

Amy Cheung

Field Summary

Fields

Modifier and Type	Field	Description
`protected HealthBar`	`ratingBar`	

Constructor Summary

Constructors

Constructor	Description
`HealthInspector (int rating)`	Constructor that takes one int - Will determine how lenient/strict the HealthInspector will be.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`act()`	Act - do whatever the HealthInspector wants to do.

void	addedToWorld (greenfoot.World w)	Adds and updates the ratingBar.
void	exit()	Will have the HealthInspector complete their inspection once this method is called.

Methods inherited from class greenfoot.Actor

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

ratingBar

```
protected HealthBar ratingBar
```

Constructor Detail

HealthInspector

```
public HealthInspector(int rating)
```

Constructor that takes one int - Will determine how lenient/strict the HealthInspector will be.

Parameters:

`rating` - How many pests HealthInspetor can take.

Method Detail

addedToWorld

```
public void addedToWorld(greenfoot.World w)
```

Adds and updates the ratingBar.

Overrides:

`addedToWorld` in class `greenfoot.Actor`

Parameters:

`w` - The world HealthInspector is in.

act

```
public void act()
```

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

Overrides:

`act` in class `greenfoot.Actor`

exit

```
public void exit()
```

Will have the HealthInspector complete their inspection once this method is called.

Class Pest

java.lang.Object
    greenfoot.Actor
        Pest

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

This is pest. During the day, pest will randomly move around. Pest will bother nearby Clients and will reduce their patience. Staff will hit them with Weapons, and once hit, Pest will die.

Version:

4 (14/05/2020)

Author:

Evan Ng, Alex Mar

Field Summary

Fields

Modifier and Type	Field	Description
`protected int`	`botherRange`	
`protected int`	`botherRate`	
`protected int`	`changeFrequency`	
`protected int`	`count`	
`protected boolean`	`dead`	
`protected greenfoot.GreenfootImage`	`image`	
`protected int`	`imageHeight`	
`protected int`	`imageWidth`	
`protected boolean`	`isTargeted`	
`protected int`	`maxSpeed`	
`protected int`	`maxXpos`	
`protected int`	`maxYpos`	
`protected int`	`minSpeed`	
`protected int`	`minXpos`	
`protected int`	`minYpos`	
`protected double`	`shrinkSpeed`	
`protected int`	`xSpeed`	

protected int	ySpeed

Constructor Summary

Constructors

Constructor	Description
Pest()	

Method Summary

All Methods Instance Methods Abstract Methods Concrete Methods

Modifier and Type	Method	Description
protected void	botherCustomer()	Pest will decrease customers' satisfaction levels within a certain range.
protected void	changeSpeed()	Changes the speed of Rat at the given changeFrequency.
abstract void	die()	Pest will die once this method is called.
boolean	getDead()	Return whether Pest is dead.
boolean	getIsTargeted()	Returns whether Pest is targeted by a Staff
void	setIsTargeted (boolean isTargeted)	Sets whether Pest is targeted by a Staff
protected abstract void	shrinkUntilDelete()	Will quickly shrink once Pest is dead.
protected void	stayInLimit()	Keeps Pest within the given position limits.

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

minSpeed

```
protected int minSpeed
```

maxSpeed

```
protected int maxSpeed
```

changeFrequency

```
protected int changeFrequency
```

xSpeed

```
protected int xSpeed
```

ySpeed

```
protected int ySpeed
```

maxXpos

```
protected int maxXpos
```

minXpos

```
protected int minXpos
```

maxYpos

```
protected int maxYpos
```

minYpos

```
protected int minYpos
```

botherRange

```
protected int botherRange
```

botherRate

```
protected int botherRate
```

count

```
protected int count
```

dead

```
protected boolean dead
```

shrinkSpeed

```
protected double shrinkSpeed
```

isTargeted

```
protected boolean isTargeted
```

imageWidth

```
protected int imageWidth
```

imageHeight

```
protected int imageHeight
```

image

```
protected greenfoot.GreenfootImage image
```

Constructor Detail

Pest

```
public Pest()
```

Method Detail

changeSpeed

```
protected void changeSpeed()
```

Changes the speed of Rat at the given changeFrequency. The higher changeFrequency is, the less Rat changes speed/direction. The speed is changed to a random magnitude between minSpeed and maxSpeed.

stayInLimit

```
protected void stayInLimit()
```

Keeps Pest within the given position limits. If Pest is outside the given position limits, Pest will change direction to avoid the limit.

botherCustomer

```
protected void botherCustomer()
```

Pest will decrease customers' satisfaction levels within a certain range.

shrinkUntilDelete

```
protected abstract void shrinkUntilDelete()
```

Will quickly shrink once Pest is dead. Will delete itself at the end.

die

```
public abstract void die()
```

Pest will die once this method is called.

getDead

```
public boolean getDead()
```

Return whether Pest is dead.

Returns:

boolean The living state of Pest, true if Pest is dead.

getIsTargeted

```
public boolean getIsTargeted()
```

Returns whether Pest is targeted by a Staff

Returns:

boolean Whether or not pest is targeted by Staff, true if Targeted.

setIsTargeted

```
public void setIsTargeted(boolean isTargeted)
```

Sets whether Pest is targeted by a Staff

Parameters:

`isTargeted` - Wheter or not pest is targeted by Staff, true if Targeted.

Class PointDistributor

java.lang.Object
    greenfoot.Actor
        PointDistributor

```
public class PointDistributor
extends greenfoot.Actor
```

PointDistributor is a simple Greenfoot Actor that displays a number (points) in a Button. The number can be changed using two arrows placed on the left and right sides of the display. The amount of changer per click can be changed. The display is set such that the points have a maximum and minimum value.

Version:

1 (23/04/2020)

Author:

Evan Ng

Constructor Summary

Constructors

Constructor	Description
`PointDistributor()`	Main constructor - sets all the variables to default values.
`PointDistributor(int defaultPoints, int minPoints, int maxPoints, int changeRate, int width, int height, int cornerRadius, int borderThickness, int fontSize, greenfoot.Color backColor, greenfoot.Color borderColor)`	Constructor that takes five ints and two Colors - for adjusting size and colors of the display: width and height of the display, the border, and font size.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`act()`	Act - do whatever the PointDistributor wants to do.
`Button`	`getButton()`	Returns the button.
`int`	`getClicksFromDefault()`	Returns how many times were clicked away from default.

`int`	`getCurrentPoints()`	Returns how many points the display is currently on.
`Arrow`	`getLArrow()`	Returns the left arrow.
`Arrow`	`getRArrow()`	Returns the right arrow.
`void`	`lockArrow(boolean locked, boolean left)`	Locks or unlocks the chosen arrow (left or right)

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

PointDistributor

```
public PointDistributor()
```

Main constructor - sets all the variables to default values. This is mostly used for testing purposes.

PointDistributor

```
public PointDistributor(int defaultPoints,
                        int minPoints,
                        int maxPoints,
                        int changeRate,
                        int width,
                        int height,
                        int cornerRadius,
                        int borderThickness,
                        int fontSize,
                        greenfoot.Color backColor,
                        greenfoot.Color borderColor)
```

Constructor that takes five ints and two Colors - for adjusting size and colors of the display: width and height of the display, the border, and font size.

Parameters:

`defaultPoints` - Default points upon starting/restarting

`minPoints` - Minimum points allowed

`maxPoints` - Maximum points allowed

`changeRate` - How much each arrow click changes points

`width` - Width of display

`height` - Height of display

`cornerRadius` - Radius of curve on the corners

`borderThickness` - Width of border

`fontSize` - Size of the text

`backColor` - Color of background

`borderColor` - Color of border

Method Detail

act

```
public void act()
```

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

Overrides:

`act` in class `greenfoot.Actor`

getCurrentPoints

```
public int getCurrentPoints()
```

Returns how many points the display is currently on.

Returns:

int Amount of points currently.

getClicksFromDefault

```
public int getClicksFromDefault()
```

Returns how many times were clicked away from default. Ex. If default value was 5, current value is 10, and change rate is 5, 1 is returned. Ex. If default value was 6, current value is 0, and change rate is 2, -3 is returned.

Returns:

int Amount of clicks away from default, 0 means at default.

getButton

```
public Button getButton()
```

Returns the button.

Returns:

int Amount of points currently.

lockArrow

```
public void lockArrow(boolean locked, boolean left)
```

Locks or unlocks the chosen arrow (left or right)

Parameters:

`locked` - Whether the chosen arrow is locked or unlocked: true is locked, false is unlocked

`left` - Chosen arrow: true is left arrow, false is right arrow

getLArrow

```
public Arrow getLArrow()
```

Returns the left arrow.

Returns:

Arrow Returns the left arrow.

getRArrow

```
public Arrow getRArrow()
```

Returns the right arrow.

Returns:

Arrow Returns the right arrow.

Class PointsWorld

java.lang.Object
    greenfoot.World
        PointsWorld

```
public class PointsWorld
extends greenfoot.World
```

This is the points world, where points are distributed to change the stats and vary the simulation. Hovering over various parts of the world will allow the user to fully understand what can be changed and how it will change the simulation. Sounds from https://tabletopaudio.com.

Version:

5 (05/05/2020)

Author:

Amy Cheung, Alex Mar, Evan Ng, Raymond Zhang

Constructor Summary

Constructors

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

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`act()`	Act - do whatever the PopUp wants to do.

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

PointsWorld

```
public PointsWorld()
```

Constructor for objects of class PointsWorld. No parameters are taken.

Method Detail

act

```
public void act()
```

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

Overrides:

`act` in class `greenfoot.World`

Class PopUp

java.lang.Object
    greenfoot.Actor
        PopUp

```
public class PopUp
extends greenfoot.Actor
```

PopUp is a Greenfoot Actor that displays text. The text is displayed in a rectangular box. The PopUp displays two types of text: title and body texts.
The title is shown at the top of the display and can be adjusted to be left, center, or right justified. The body text is shown below the title and is left justified. The body has automatic text wrap.
PopUp comes with Arrow, CloseButton, and DotCarousel, all of which can be disabled.
The display also has the option of having a pop up animation.

Version:

9 (23/04/2020)

Author:

Evan Ng

Constructor Summary

Constructors

Constructor	Description
`PopUp()`	Main constructor - sets all the variables to default values.
`PopUp(int width, int height, int cornerRadius, int borderThickness)`	Constructor that takes four ints - a even simpler constructor that does not have a pop up animation and has no accessories (no close button, arrows, or dot carousel) and uses the default Colors.
`PopUp(int width, int height, int cornerRadius, int borderThickness, boolean popsUp, int popUpTime, int numberOfSlides, int currentSlide, greenfoot.Color backColor, greenfoot.Color borderColor)`	Constructor that takes five ints, one boolean, and two Colors - the size and colours of the display background and border can be adjusted.
`PopUp(int width, int height, int cornerRadius, int borderThickness, greenfoot.Color backColor, greenfoot.Color borderColor)`	Constructor that takes four ints and two Colors - a simpler constructor that does not have a pop up animation and has no accessories (no close button, arrows, or dot carousel).

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`act()`	Act - do whatever the PopUp wants to do.
`int`	`getCurrentSlide()`	Returns which slide the display is currently on.
`void`	`setBody(java.lang.String text)`	Sets the body text.
`void`	`setTitle(java.lang.String text)`	Sets the title text.

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

PopUp

```
public PopUp()
```

Main constructor - sets all the variables to default values. This is mostly used for testing purposes.

PopUp

```
public PopUp(int width,
             int height,
             int cornerRadius,
             int borderThickness,
             boolean popsUp,
             int popUpTime,
             int numberOfSlides,
             int currentSlide,
             greenfoot.Color backColor,
             greenfoot.Color borderColor)
```

Constructor that takes five ints, one boolean, and two Colors - the size and colours of the display background and border can be adjusted. The corners can also be rounded using cornerRadius. The pop up animation can be enabled or disabled; the time of the animation can also be changed.

Parameters:

`width` - Width of display

`height` - Height of display

`cornerRadius` - Radius of curve on the corners

`borderThickness` - Width of border, set to 0 for no border

`popsUp` - Whether display has a pop up animation

`popUpTime` - Length of time of pop up animation

`numberOfSlides` - Number of dots displayed at bottom

`currentSlide` - Which dot is highlighted

`backColor` - Color of background

`borderColor` - Color of border

PopUp

```
public PopUp(int width,
             int height,
             int cornerRadius,
             int borderThickness,
             greenfoot.Color backColor,
             greenfoot.Color borderColor)
```

Constructor that takes four ints and two Colors - a simpler constructor that does not have a pop up animation and has no accessories (no close button, arrows, or dot carousel).

Parameters:

`width` - Width of display

`height` - Height of display

`cornerRadius` - Radius of curve on the corners

`borderThickness` - Width of border, set to 0 for no border

`backColor` - Color of background

`borderColor` - Color of border

PopUp

```
public PopUp(int width,
             int height,
             int cornerRadius,
             int borderThickness)
```

Constructor that takes four ints - a even simpler constructor that does not have a pop up animation and has no accessories (no close button, arrows, or dot carousel) and uses the default Colors.

Parameters:

`width` - Width of display

`height` - Height of display

`cornerRadius` - Radius of curve on the corners

`borderThickness` - Width of border, set to 0 for no border

Method Detail

act

```
public void act()
```

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

Overrides:

`act` in class `greenfoot.Actor`

setTitle

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

Sets the title text. The text will be displayed as a single line at the top of the pop up.
In order to center the text, add spaces to the string. Add spaces to the front of the text if the text is displayed too far the left and vice versa if it is displayed too far to the right.
Ex: popUp.setTitle("TITLE WORDS ");
Note: Text cannot be centered automatically because Greenfoot cannot accurately determine the widths of texts (especially when using different fonts).

Parameters:

`text` - String that will be displayed at the top of the pop up

setBody

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

Sets the body text. The text will be displayed in multiple lines below the title. It will be displayed left justified.

Parameters:

`text` - String that will be displayed below the title

getCurrentSlide

```
public int getCurrentSlide()
```

Returns which slide the display is currently on.

Returns:

int Slide display is currently showing.

Class PriceChange

java.lang.Object
greenfoot.Actor
PriceChange

```
public class PriceChange
extends greenfoot.Actor
```

Price change will show how much a price has changed. It will move upwards slowly, while slowly growing more transparent, and delete itself at the end.

Version:

1 (07/05/2020)

Author:

Evan Ng

Constructor Summary

Constructors

Constructor	Description
`PriceChange(int money, boolean positive)`	Constructor that takes one int and one boolean.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`void`	`act()`	Act - do whatever the PriceChange wants to do.

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

PriceChange

```
public PriceChange(int money,
                   boolean positive)
```

Constructor that takes one int and one boolean. Determines what is displayed in the text.

Parameters:

`money` - How much money is lost or gained.

`positive` - Determines if there is a positive or negative sign in front of the text, true if positive

Method Detail

act

```
public void act()
```

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

Overrides:

`act` in class `greenfoot.Actor`

Class Rat

java.lang.Object
greenfoot.Actor
Pest
Rat

```
public class Rat
extends Pest
```

Rat appears during the simulation to steal food, and prevent the restaurant from becoming popular and famous. Alex did day time behaviours, Evan did night time. Images from https://hotemoji.com.

Version:

April 24, 2020

Author:

Alex Mar

Field Summary

Fields inherited from class Pest

`botherRange, botherRate, changeFrequency, count, dead, image, imageHeight, imageWidth, isTargeted, maxSpeed, maxXpos, maxYpos, minSpeed, minXpos, minYpos, shrinkSpeed, xSpeed, ySpeed`

Constructor Summary

Constructors

Constructor	Description
`Rat()`	Main constructor - sets all the variables to default values.
`Rat(boolean thief)`	This constructor determines wheter or not the rat will move randomly or if the rat will move along a determined path.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and	Method	Description

Type		
`void`	`act()`	Act - do whatever the Rat wants to do.
`void`	`die()`	Rat will start dying.
`protected void`	`shrinkUntilDelete()`	When Rat dies, Rat will shrink until it reaches 1 pixel wide and 1 pixel tall.

Methods inherited from class Pest

`botherCustomer, changeSpeed, getDead, getIsTargeted, setIsTargeted, stayInLimit`

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

Rat

```
public Rat()
```

Main constructor - sets all the variables to default values. This rat will move randomly within certain limits.

Rat

```
public Rat(boolean thief)
```

This constructor determines wheter or not the rat will move randomly or if the rat will move along a determined path.

Parameters:

`thief` - When true, the rat will go into thief mode and follow a determined path, ultimately stealing food. When false, the rat will simply move randomly

Method Detail

act

```
public void act()
```

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

Overrides:

`act` in class `greenfoot.Actor`

shrinkUntilDelete

```
protected void shrinkUntilDelete()
```

When Rat dies, Rat will shrink until it reaches 1 pixel wide and 1 pixel tall. It then removes itself. The bigger shrinkSpeed is, the faster Rat will shrink. However, shrinkSpeed must be greater than 1.

Specified by:

`shrinkUntilDelete` in class `Pest`

die

```
public void die()
```

Rat will start dying. It stops moving, shrinks, and deletes itself. Before shrinking, it grows to 1.5 times its size for an instant for a nicer death animation (using a fundamental animation principle: "anticipation").

Specified by:

`die` in class `Pest`