Class MyWorld

- java.lang.Object
    - greenfoot.World
        - MyWorld
-

- `public class` **`MyWorld`**
  `extends greenfoot.World`
- Kitchen fever is a simulation set in the kitchen of a busy restaurant. Inspired by the game "Overcooked", the simulation shows the busy work lives of chefs as they cut, cook and plate orders. The two main parent classes are the Ingredients and Chefs classes. However, there are many other classes used in the simulation. These include, the Order, Timer, ProgressBar and Targets classes.
- Having two different types of meat, vegetable and grain ingredients, there are a total of eight dishes the the chefs can make.
- The World class is primarily responsible for creating objects, generating orders and removing the order displays when a dish has been complete. The World also helps keep track of time as it runs at 60 frames per second
- In the parent class Chefs, the child classes are broken up into meat, vegetable, grain, plating and serving chefs. The meat and vegetable chefs are responsible for picking up, cutting and cooking ingredients. The grain chef is responsible for picking up grain ingredients and placing plates on the serving table. The Plating chef checks when the food is cooked and places the food on the plates. Finally, the serving chef takes the plates to serve to the customers.
- The Target class is a parent class that helps with the movement of the chefs. Each subclass represents a station. Each subclass of chef will only go to their designated stations. For instance, the meat chef will only move towards the FoodBox, CuttingBoard, FryingPan and Home Targets.
- The Order class is primarily used to display the orders visually for the viewer. However it also creates the ProgressBar and tells the chefs what ingredients they need to cook.
- The Timer and Progressbar act as an visual indication for when a process has been completed.
- An interesting thing to look out for would be how the animations are done For the animations, a 2D array is used. The array stores the type of animation needed (i.e. move down) and the images associated with that animation. This cuts back on a lot of code because everything is stored in one easily accessible array.
- Another interesting thing to look out for would be the images used. Almost all of background was hand drawn to fit the classic look of the overcooked game. In addition, all of the food images were created using editing software. The most time consuming task was

colouring the images, especially the bricks and tiles as each individual object has its unique look.

- If you don't want to wait for food to burn, press q and watch the chefs panic!
- One of the hardest tasks for this simulation was figuring out how the chefs were going to move. Typically, one would just rotate the character to move towards the target. However, since the simulation was not birds eye view, we had to go about it a completely different way through targets. However, an added benefit of doing it this way is that we actually have walking animations for the chef for all directions. In addition, I think this three dimensional view is one of the main things that help our simulation stand out from the others.
- Sound effects were taken from the youtube channel "SOUND and IMAGE FX" under the playlist "Cooking Sound Effect". Some food images were taken from the game "Minecraft". Background music is from the game "Overcooked". Animation sprites were created using the character sprite generator at "http://gaurav.munjal.us/Universal-LPC-Spritesheet-Character-Generator/". Object list and nanotime code given by Mr.Cohen.
- **Version:**
- October 2018
- **Author:**
- Daniel Tan, Alexander Yee and Zhongchi Li

Constructor Summary

Constructor and Description
MyWorld()

Method Summary

Modifier and Type	Method and Description
void	**act**()
void	**checkButton**(java.lang.String ingredientType, Buttons button1, Buttons button2) Assign the buttons clicked their corresponding ingredients to their order
void	**checkButtons**() Check which of the 2 buttons of each column is clicked
void	**finishedOrder**() Updates the order display by removing the first order and progressbar.
int	**getFrames**() Get the frame that the world is currently at

Order	**getOrders**(int num) Gets the Order objetc

- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail***
- **MyWorld[Show source in BlueJ]**
- public MyWorld()
- ***Method Detail***
- **act[Show source in BlueJ]**
- public void act()
- **Overrides:**
- act in class greenfoot.World
- **checkButton**

-
  ```
  public void checkButton(java.lang.String ingredientType,
                          Buttons button1,
                          Buttons button2)
  ```
- Assign the buttons clicked their corresponding ingredients to their order
- **checkButtons[Show source in BlueJ]**
- public void checkButtons()
- Check which of the 2 buttons of each column is clicked
- **finishedOrder[Show source in BlueJ]**
- public void finishedOrder()
- Updates the order display by removing the first order and progressbar. Shifts all displays one position to the left
- **getFrames[Show source in BlueJ]**

- public int getFrames()
- Get the frame that the world is currently at
- **Returns:**
- int the frame the world is at
- **getOrders[Show source in BlueJ]**
- public Order getOrders(int num)
- Gets the Order objetc
- **Parameters:**
- num - position of the order relative to the display
- **Returns:**
- Order returns the Order object

Class EndScreen

- java.lang.Object
    - greenfoot.World
        - EndScreen
-

- **public class EndScreen**
  extends greenfoot.World
- Version:
- **October 2018**
- Author:
- **Deston**
- ***Constructor Summary***
- Constructor and Description
- **EndScreen()**
- Constructor for objects of class EndScreen.
- ***Method Summary***
- Methods inherited from class greenfoot.World
- **act, 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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait**
- ***Constructor Detail***
- EndScreen[Show source in BlueJ]
- **public EndScreen()**
- Constructor for objects of class EndScreen.

Class Buttons

- java.lang.Object
  - greenfoot.Actor
    - Buttons

- public class **Buttons**
  extends greenfoot.Actor
- This class is a widget that creates buttons for users to input their own orders into the simulation
- **Version:**
- **October 2018**
- **Author:**
- **Zhongchi Li**

Constructor Summary

Constructor and Description
Buttons(int length, int width, java.lang.String image, java.lang.String altImage, boolean stay) The main constructor, accepts the length, width, name of the image files, and whether it will stay as clicked down or not

Method Summary

Modifier and Type	Method and Description
void	**act**() Act, sets paint order and check for clicks
void	**click**() this method checks if button is clicked and sets the images accordingly
void	**disable**() returns if it has been disabled or not
void	**enable**() returns whether it is enabled
boolean	**getClicked**() returns if it has been clicked 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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail***
- **Buttons**

-
  ```
  public Buttons(int length,
                 int width,
                 java.lang.String image,
                 java.lang.String altImage,
                 boolean stay)
  ```
- The main constructor, accepts the length, width, name of the image files, and whether it will stay as clicked down or not
- ***Method Detail***
- **act[Show source in BlueJ]**
- public void act()
- Act, sets paint order and check for clicks
- **Overrides:**
- act in class greenfoot.Actor
- **click[Show source in BlueJ]**
- public void click()
- this method checks if button is clicked and sets the images accordingly
- **disable[Show source in BlueJ]**
- public void disable()
- returns if it has been disabled or not
- **enable[Show source in BlueJ]**
- public void enable()
- returns whether it is enabled
- **getClicked[Show source in BlueJ]**
- public boolean getClicked()

- returns if it has been clicked or not

Class Chefs

- java.lang.Object
  - greenfoot.Actor
    - Chefs

-

- **public abstract class Chefs**
  extends greenfoot.Actor
- **The Chefs class is a Superclass for all the chefs present in the simulation, that controls all the movement, animations and decision making. The Chefs class receives information from the majority of all the other classes to be relayed to all the other chefs. It also controls the Orders that are created from the World, helping the chefs 'see' the next orders that need to be prepare**
- **Contributions by Alexander include: Animations of the chefs Sound effects being played How the chef gets orders Helped fix bugs with movement Created methods checkOrder(),cut(), finishedIngredient(), finishedOrder(), animate(),**
- **Contributions by Daniel include: Chef movement Chef interactions with ingredients**
- **Contribution by Deston and Justin: Creating ingredients within the class How ingredients interacted with chefs**
- Version:
- **October 2018**
- Author:
- **Daniel, Alexander, Justin and Deston**
- ***Field Summary***

Modifier and Type	Field and Description
protected Vegetables	**BroccoliOrBokChoy**
protected Meat	**ChickenOrBeef**
protected boolean	**follow**
protected boolean	**idle**
protected java.lang.String	**ingredientSelect**

protected boolean	isCutting
protected boolean	panic
protected boolean	placedPlate
protected int	speedX
protected int	speedY
protected int	targetX
protected int	targetY
protected int	type

Constructor Summary

Constructor and Description
Chefs()

Method Summary

Modifier and Type	Method and Description
void	addedToWorld(greenfoot.World w)
void	checkOrder(int food)
void	cut()
void	finishedIngredient()
void	finishedOrder()
int	getCuttingTime() returns cutting time
protected void	idleWalk()
protected void	lowerOrder()
protected void	prepare(java.lang.String chefType, java.lang.String meatVegGrain)
void	setCuttingTime(int time) resets cutting time

- void
    - **setPanic()**
    - Makes chef panic, causing the chefs to move randomly at a greater spee

- **Methods inherited from class greenfoot.Actor**
  - **act, 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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait**

- ***Field Detail***
  - BroccoliOrBokChoy[Show source in BlueJ]
    - **protected Vegetables BroccoliOrBokChoy**
  - ChickenOrBeef[Show source in BlueJ]
    - **protected Meat ChickenOrBeef**
  - follow[Show source in BlueJ]
    - **protected boolean follow**
  - idle[Show source in BlueJ]
    - **protected boolean idle**
  - ingredientSelect[Show source in BlueJ]
    - **protected java.lang.String ingredientSelect**
  - isCutting[Show source in BlueJ]
    - **protected boolean isCutting**
  - panic[Show source in BlueJ]
    - **protected boolean panic**
  - placedPlate[Show source in BlueJ]
    - **protected boolean placedPlate**

- speedX[Show source in BlueJ]
  - **protected int speedX**
- speedY[Show source in BlueJ]
  - **protected int speedY**
- targetX[Show source in BlueJ]
  - **protected int targetX**
- targetY[Show source in BlueJ]
  - **protected int targetY**
- type[Show source in BlueJ]
  - **protected int type**

- ***Constructor Detail***
  - Chefs[Show source in BlueJ]
    - **public Chefs()**

- ***Method Detail***
  - addedToWorld
    - **public void addedToWorld(greenfoot.World w)**
    - Overrides:
    - **addedToWorld in class greenfoot.Actor**
  - checkOrder[Show source in BlueJ]
    - **public void checkOrder(int food)**
  - cut[Show source in BlueJ]
    - **public void cut()**
  - finishedIngredient[Show source in BlueJ]
    - **public void finishedIngredient()**
  - finishedOrder[Show source in BlueJ]
    - **public void finishedOrder()**
  - getCuttingTime[Show source in BlueJ]
    - **public int getCuttingTime()**

- returns cutting time

- idleWalk[Show source in BlueJ]

- **protected void idleWalk()**

- lowerOrder[Show source in BlueJ]

- **protected void lowerOrder()**

- prepare

- **protected void prepare(java.lang.String chefType,
  java.lang.String meatVegGrain)**

- setCuttingTime[Show source in BlueJ]

- **public void setCuttingTime(int time)**

- resets cutting time

- setPanic[Show source in BlueJ]

- **public void setPanic()**

- Makes chef panic, causing the chefs to move randomly at a greater spee

Class GrainChef

- java.lang.Object
  - greenfoot.Actor
    - Chefs
      - GrainChef

-
- **public class GrainChef**
  extends Chefs
- Version:
- **October 2018**
- Author:
- **Alexander, Daniel, Justin and Deston**
- *Field Summary*
- Fields inherited from class Chefs
- **BroccoliOrBokChoy, ChickenOrBeef, follow, idle, ingredientSelect, isCutting, panic, placedPlate, speedX, speedY, targetX, targetY, type**
- *Constructor Summary*

Constructor and Description
GrainChef() Constructor

- *Method Summary*

Modifier and Type	Method and Description
void	**act()** Act - do whatever the PlaterChef wants to do.
protected void	**FollowChef()**
void	**updatePlate()** Updates whether the plate has been placed

- Methods inherited from class Chefs
- **addedToWorld, animate, checkOrder, cut, finishedIngredient, finishedOrder, getCuttingTime, idleWalk, lowerOrder, prepare, setCuttingTime, setPanic**
- 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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait**

- *Constructor Detail*

- GrainChef[Show source in BlueJ]

- **public GrainChef()**

- Constructor

- *Method Detail*

- act[Show source in BlueJ]

- **public void act()**

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

- Overrides:

- **act in class greenfoot.Actor**

- FollowChef[Show source in BlueJ]

- **protected void FollowChef()**

- updatePlate[Show source in BlueJ]

- **public void updatePlate()**

- Updates whether the plate has been placed

Class MeatChef

- java.lang.Object
  - greenfoot.Actor
    - Chefs
      - MeatChef

- **public class MeatChef**
  extends Chefs
- Version:
- **October 2018**
- Author:
- **Alexander, Daniel, Deston and Justin**

- ***Field Summary***
- Fields inherited from class Chefs
- **BroccoliOrBokChoy, ChickenOrBeef, follow, idle, ingredientSelect, isCutting, panic, placedPlate, speedX, speedY, targetX, targetY, type**

- ***Constructor Summary***

Constructor and Description
MeatChef()   Constructor for MeatChef class

- ***Method Summary***

Modifier and Type	Method and Description
void	**act()** Act - do whatever the MeatChef wants to do.
protected void	**FollowChef()**

- Methods inherited from class Chefs
- **addedToWorld, animate, checkOrder, cut, finishedIngredient, finishedOrder, getCuttingTime, idleWalk, lowerOrder, prepare, setCuttingTime, setPanic**

- 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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait**
- ***Constructor Detail***
- MeatChef[Show source in BlueJ]
- **public MeatChef()**
- Constructor for MeatChef class
- ***Method Detail***
- act[Show source in BlueJ]
- **public void act()**
- Act - do whatever the MeatChef wants to do. This method is called whenever the 'Act' or 'Run' button gets pressed in the environment.
- Overrides:
- **act in class greenfoot.Actor**
- FollowChef[Show source in BlueJ]
- **protected void FollowChef()**

Class PlatingChef

- java.lang.Object
    - greenfoot.Actor
        - Chefs
            - PlatingChef
- ---
- **public class PlatingChef**
  extends Chefs
- Version:
- **October 2018**
- Author:
- **Alexander, Daniel, Deston and Justin**
- ***Field Summary***
- Fields inherited from class Chefs
- **BroccoliOrBokChoy, ChickenOrBeef, follow, idle, ingredientSelect, isCutting, panic, placedPlate, speedX, speedY, targetX, targetY, type**
- ***Constructor Summary***

Constructor and Description
PlatingChef() Constructor

- ***Method Summary***

Modifier and Type	Method and Description
void	**act()** Act - do whatever the FoodPlatingChef wants to do.
protected void	**FollowChef()**

- Methods inherited from class Chefs
- **addedToWorld, animate, checkOrder, cut, finishedIngredient, finishedOrder, getCuttingTime, idleWalk, lowerOrder, prepare, setCuttingTime, setPanic**
- 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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait**
- ***Constructor Detail***
- PlatingChef[Show source in BlueJ]
- **public PlatingChef()**
- Constructor
- ***Method Detail***
- act[Show source in BlueJ]
- **public void act()**
- Act - do whatever the FoodPlatingChef wants to do. This method is called whenever the 'Act' or 'Run' button gets pressed in the environment.
- Overrides:
- **act in class greenfoot.Actor**
- FollowChef[Show source in BlueJ]
- **protected void FollowChef()**

Class ServingChef

- java.lang.Object
    - greenfoot.Actor
        - Chefs
            - ServingChef

- ---
- **public class ServingChef**
  extends Chefs
- Version:
- **October 2018**
- Author:
- **Daniel**

Field Summary

- Fields inherited from class Chefs
- **BroccoliOrBokChoy, ChickenOrBeef, follow, idle, ingredientSelect, isCutting, panic, placedPlate, speedX, speedY, targetX, targetY, type**

Constructor Summary

Constructor and Description
ServingChef() Constructor

Method Summary

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

- Methods inherited from class Chefs
- **addedToWorld, animate, checkOrder, cut, finishedIngredient, finishedOrder, getCuttingTime, idleWalk, lowerOrder, prepare, setCuttingTime, setPanic**

- 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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait**
- ***Constructor Detail***
- ServingChef[Show source in BlueJ]
- **public ServingChef()**
- **Constructor**
- ***Method Detail***
- act[Show source in BlueJ]
- **public void act()**
- **Act - do whatever the ServingChef 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 VegetableChef

- java.lang.Object
    - greenfoot.Actor
        - Chefs
            - VegetableChef
-

- **public class VegetableChef**
  extends Chefs
- Version:
- **October 2018**
- Author:
- **Alexander, Daniel, Deston and Justin**
- *Field Summary*
- Fields inherited from class Chefs
- **BroccoliOrBokChoy, ChickenOrBeef, follow, idle, ingredientSelect, isCutting, panic, placedPlate, speedX, speedY, targetX, targetY, type**
- *Constructor Summary*

Constructor and Description
VegetableChef() Constructor

- *Method Summary*

Modifier and Type	Method and Description
void	**act()** Act - do whatever the VegetableChef wants to do.
protected void	**FollowChef()**

- Methods inherited from class Chefs
- **addedToWorld, animate, checkOrder, cut, finishedIngredient, finishedOrder, getCuttingTime, idleWalk, lowerOrder, prepare, setCuttingTime, setPanic**
- 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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait**

- ***Constructor Detail***
- **VegetableChef[Show source in BlueJ]**
- **public VegetableChef()**
- Constructor
- ***Method Detail***
- **act[Show source in BlueJ]**
- **public void act()**
- Act - do whatever the VegetableChef wants to do. This method is called whenever the 'Act' or 'Run' button gets pressed in the environment.
- Overrides:
- **act in class greenfoot.Actor**
- **FollowChef[Show source in BlueJ]**
- **protected void FollowChef()**

Class Ingredients

- java.lang.Object
    - greenfoot.Actor
        - Ingredients
- ---
- **public abstract class Ingredients**
  extends greenfoot.Actor
- **The Ingredients class works along with the Chef class to create instances of different types of food such as meat, vegetables, and grains which are used by the chefs. Each instance of meat, vegetables, or grains has its own unique properties to determine how it is being processed (chopped or cooked), and acts appropriately based on which part of the process it is going through. Within the Ingredients class, an interesting aspect to take note of is that many of the methods within the class are called on through the Chef class, thus showing how the two major superclasses within the program interact closely to make the program function smoothly. For example, many methods deal with Greenfoot Images to effectively animate and process food or calculate the time for which food is cooked before it is burnt are stored within the Food class, and are further implemented to work alongside the Chefs after such objects are created in the world. These methods are specifically optimized in order to reduce redundant code and bring out the benefits of Object Oriented Programming due to their interactivity with the Food class. In addition, our class fully utilizes the benefits of Object Oriented Programming because many instances of each object can be created, and they will all work the same in our simulation because of how our methods were generalized. As a result, no matter what type of object or how many of them there are, it will all work seamlessly with our simulation.**
- Version:
- **October 2018**
- Author:
- **Deston Wong, Justin Huynh**
- ***Field Summary***

Modifier and Type	Field and Description
protected greenfoot.GreenfootImage	**altImage**
protected greenfoot.G	**cloud**

reenfootImage	
protected java.lang.String	foodName
protected greenfoot.GreenfootImage	image
protected boolean	isBurning
protected boolean	isChopped
protected boolean	isCooked
protected boolean	isCooking
protected boolean	isPreparing
protected boolean	locationSet
protected greenfoot.GreenfootImage	smoking

- *Constructor Summary*

Constructor and Description
Ingredients()

- *Method Summary*

Modifier and Type	Method and Description
void	act() A method which runs for each instance of Food in the world.
boolean	getChopped() Getter method to determine whether the ingredient is chopped.
boolean	getCooked() Getter method to determine whether the ingredient is fully cooked.

boolean	**getCooking()** Getter method to determine whether the ingredient is cooking.
int	**getCookingTime()** resets cutting time
java.lang.String	**getName()** Getter method to determine the name of the ingredient
void	**PickUp()** Determines the actions of the food when it is picked up, after it has been either chopped or cooked.
void	**Process(int position)** Initializes boolean allowing for animations to occur.
void	**setCookingTime(int time)** Set cooking time

- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait**
- ***Field Detail***
- altImage[Show source in BlueJ]
- **protected greenfoot.GreenfootImage altImage**
- cloud[Show source in BlueJ]
- **protected greenfoot.GreenfootImage cloud**
- foodName[Show source in BlueJ]
- **protected java.lang.String foodName**
- image[Show source in BlueJ]
- **protected greenfoot.GreenfootImage image**
- isBurning[Show source in BlueJ]

- **protected boolean isBurning**
- isChopped[Show source in BlueJ]
- **protected boolean isChopped**
- isCooked[Show source in BlueJ]
- **protected boolean isCooked**
- isCooking[Show source in BlueJ]
- **protected boolean isCooking**
- isPreparing[Show source in BlueJ]
- **protected boolean isPreparing**
- locationSet[Show source in BlueJ]
- **protected boolean locationSet**
- smoking[Show source in BlueJ]
- **protected greenfoot.GreenfootImage smoking**
- *Constructor Detail*
- Ingredients[Show source in BlueJ]
- **public Ingredients()**
- *Method Detail*
- act[Show source in BlueJ]
- **public void act()**
- A method which runs for each instance of Food in the world. Repeatedly checks if ingredients are cooked or chopped so they can act according to actions of the Chef class.
- Overrides:
- **act in class greenfoot.Actor**
- getChopped[Show source in BlueJ]
- **public boolean getChopped()**
- Getter method to determine whether the ingredient is chopped.
- Returns:
- **boolean True if ingredient is chopped**
- getCooked[Show source in BlueJ]

- **public boolean getCooked()**
- Getter method to determine whether the ingredient is fully cooked.
- Returns:
- **boolean True if ingredient is fully cooked**
- getCooking[Show source in BlueJ]
- **public boolean getCooking()**
- Getter method to determine whether the ingredient is cooking.
- Returns:
- **boolean True if ingredient is being cooked**
- getCookingTime[Show source in BlueJ]
- **public int getCookingTime()**
- resets cutting time
- getName[Show source in BlueJ]
- **public java.lang.String getName()**
- Getter method to determine the name of the ingredient
- Returns:
- **String The name of the current ingredient**
- PickUp[Show source in BlueJ]
- **public void PickUp()**
- Determines the actions of the food when it is picked up, after it has been either chopped or cooked. Is called upon throughout methods in the Chef class.
- Process[Show source in BlueJ]
- **public void Process(int position)**
- Initializes boolean allowing for animations to occur. Is called upon throughout methods in the Chef class.
- Parameters:
- **position - Position of the ingredient relative to which pan it is going on**
- setCookingTime[Show source in BlueJ]
- **public void setCookingTime(int time)**
- Set cooking time
- Parameters:

- **int - How long the ingredient takes to cook**

Class Counter

- java.lang.Object
    - greenfoot.Actor
        - Counter
-

- public class **Counter**
  extends greenfoot.Actor
- Updates the counter class image whenever the Grain Chef obtains noodles or rice.
- **Version:**
- October 2018
- **Author:**
- Justin Huynh
- ***Constructor Summary***

Constructor and Description
Counter()

- ***Method Summary***

Modifier and Type	Method and Description
void	**act**() After 50 frames have passed, image will return to the empty countertop
void	**getNoodles**() Updates counter image with noodles
void	**getRice**() Updates counter image with an open rice pot
void	**returnImage**() Returns image to original countertop (no noodles or rice)

- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail***
- **Counter[Show source in BlueJ]**
- public Counter()
- ***Method Detail***
- **act[Show source in BlueJ]**
- public void act()
- After 50 frames have passed, image will return to the empty countertop
- **Overrides:**
- act in class greenfoot.Actor
- **getNoodles[Show source in BlueJ]**
- public void getNoodles()
- Updates counter image with noodles
- **getRice[Show source in BlueJ]**
- public void getRice()
- Updates counter image with an open rice pot
- **returnImage[Show source in BlueJ]**
- public void returnImage()
- Returns image to original countertop (no noodles or rice)

Class Meat

- java.lang.Object
    - greenfoot.Actor
        - Ingredients
            - Meat

-

- public class **Meat**
  extends Ingredients
- **Version:**
- October 2018
- **Author:**
- Justin Huynh
- ***Field Summary***
- **Fields inherited from class Ingredients**
- altImage, cloud, foodName, image, isBurning, isChopped, isCooked, isCooking, isPreparing, locationSet, smoking
- ***Constructor Summary***
    - **Constructor and Description**
    - **Meat**(java.lang.String foodName)
- ***Method Summary***
- **Methods inherited from class Ingredients**
- act, getCooked, getCooking, getCookingTime, getName, PickUp, Process, setCookingTime
- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail***

- **Meat**

- public Meat(java.lang.String foodName)

Class Vegetables

- java.lang.Object
    - greenfoot.Actor
        - Ingredients
            - Vegetables
- ---
- public class **Vegetables**
  extends Ingredients
- **Version:**
- October 2018
- **Author:**
- Justin Huynh
- ***Field Summary***
- **Fields inherited from class Ingredients**
- altImage, cloud, foodName, image, isBurning, isChopped, isCooked, isCooking, isPreparing, locationSet, smoking
- ***Constructor Summary***
- **Constructor and Description**
- **Vegetables**(java.lang.String foodName)
- ***Method Summary***
- **Methods inherited from class Ingredients**
- act, getCooked, getCooking, getCookingTime, getName, PickUp, Process, setCookingTime
- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail*****Vegetables**

- public Vegetables(java.lang.String foodName)

Class Order

- java.lang.Object
    - greenfoot.Actor
        - Order

-

- public class **Order**
  extends greenfoot.Actor
- **Version:**
- October 2018
- **Author:**
- Alexander

- ***Constructor Summary***

Constructor and Description
Order(java.lang.String[] ingredients)

- ***Method Summary***

Modifier and Type	Method and Description
void	**addedToWorld**(greenfoot.World w)
void	**changeLocation**()
java.lang.String[]	**getIngredients**()

- **Methods inherited from class greenfoot.Actor**
- act, 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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- ***Constructor Detail***
- **Order**
- public Order(java.lang.String[] ingredients)
- ***Method Detail***
- **addedToWorld**
- public void addedToWorld(greenfoot.World w)
- **Overrides:**
- addedToWorld in class greenfoot.Actor
- **changeLocation[Show source in BlueJ]**
- public void changeLocation()
- **getIngredients[Show source in BlueJ]**
- public java.lang.String[] getIngredients()
    -

Class Plate

- java.lang.Object
    - greenfoot.Actor
        - Plate
-

- public class **Plate**
  extends greenfoot.Actor
- When a chef interacts with a plate, their cooked ingredient will be drawn onto the plate. When the plate is complete, it can be sent out to the customer.
- **Version:**
- October 2018
- **Author:**
- Deston Wong, Justin Huynh

- ***Constructor Summary***

Constructor and Description
Plate()

- ***Method Summary***

Modifier and Type	Method and Description
boolean	**checkIfComplete**() Method to check if the plate has been completed
int	**getPosition**()
void	**placePlate**() Place plate at a location 40 pixels to the right
void	**setFood**(java.lang.String foodName) Adds the image of the given food onto the plate
void	**setPosition**(int position)

- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- ***Constructor Detail***
- **Plate[Show source in BlueJ]**
- public Plate()
- ***Method Detail***
- **checkIfComplete[Show source in BlueJ]**
- public boolean checkIfComplete()
- Method to check if the plate has been completed
- **Returns:**
- boolean True if plate has all components
- **getPosition[Show source in BlueJ]**
- public int getPosition()
- **placePlate[Show source in BlueJ]**
- public void placePlate()
- Place plate at a location 40 pixels to the right
- **setFood**
- public void setFood(java.lang.String foodName)
- Adds the image of the given food onto the plate
- **setPosition[Show source in BlueJ]**
- public void setPosition(int position)

Class ProgressBar

- java.lang.Object
    - greenfoot.Actor
        - ProgressBar

-

- **public class ProgressBar**
  extends greenfoot.Actor
- Version:
- **October 2018**
- Author:
- **Alexander Yee**

- *Constructor Summary*

Constructor and Description
ProgressBar(double width, double height)

- *Method Summary*

Modifier and Type	Method and Description
void	**changeLocation()** Shift the par to the left
void	**resize()** Adds one level of progress

- 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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait**

- *Constructor Detail*

- ProgressBar[Show source in BlueJ]
- **public ProgressBar(double width,**
  double height)
- ***Method Detail***
- changeLocation[Show source in BlueJ]
- **public void changeLocation()**
- Shift the par to the left
- resize[Show source in BlueJ]
- **public void resize()**
- Adds one level of progress

Class Sliders

- java.lang.Object
    - greenfoot.Actor
        - Sliders

- public class **Sliders**
  extends greenfoot.Actor
- The sliders class is a widget that allows the user to change certain time lengths up to a range in this simulation.
- **Version:**
- Oct 2018
- **Author:**
- Zhongchi Li with the help of Gevater_Tod4711

Constructor Summary

Constructor and Description
Sliders(int min, int max, int initial, int which) The main constructor, accepts a minimum, maximum and inital vaue, and an int to indicate which slider it is

Method Summary

Modifier and Type	Method and Description
void	**act**()
void	**addedToWorld**(greenfoot.World world) add the hand to the world
int	**getCurX**() returns current x value
Sliders.SliderHand	**getHand**() returns the assisting hand object
int	**getInitial**() returns initial value
int	**getMaxX**() returns maximum x value
int	**getMinX**() returns minimum x value
int	**getRatio**() returns the ratio between the length of the image and the range of the value

- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail***
  - **Sliders[Show source in BlueJ]**

  -
    ```
    public Sliders(int min,
                   int max,
                   int initial,
                   int which)
    ```
  - The main constructor, accepts a minimum, maximum and inital vaue, and an int to indicate which slider it is
- ***Method Detail***
  - **act[Show source in BlueJ]**
  - public void act()
  - **Overrides:**
  - act in class greenfoot.Actor
  - **addedToWorld**
  - public void addedToWorld(greenfoot.World world)
  - add the hand to the world
  - **Overrides:**
  - addedToWorld in class greenfoot.Actor
  - **getCurX[Show source in BlueJ]**
  - public int getCurX()
  - returns current x value
- **getHand[Show source in BlueJ]**
- public Sliders.SliderHand getHand()

- returns the assisting hand object

- **getInitial[Show source in BlueJ]**

- `public int getInitial()`
- returns initial value

- **getMaxX[Show source in BlueJ]**

- `public int getMaxX()`
- returns maximum x value

- **getMinX[Show source in BlueJ]**

- `public int getMinX()`
- returns minimum x value

- **getRatio[Show source in BlueJ]**

- `public int getRatio()`
- returns the ratio between the length of the image and the range of the value

Class Targets

- java.lang.Object
  - greenfoot.Actor
    - Targets
-

- public class **Targets**
  extends greenfoot.Actor
- Write a description of class Targets here.
- **Version:**
- October 2018
- **Author:**
- Daniel
- ***Field Summary***

Modifier and Type	Field and Description
protected boolean	**hasPlate**
protected boolean	**isFree**
protected int	**speed**

- ***Constructor Summary***

Constructor and Description
Targets()

- ***Method Summary***

Modifier and Type	Method and Description
protected boolean	**getFree**()
protected boolean	**getPlate**()
protected void	**move**()
protected void	**moveAround**()
protected void	**setFreeFalse**()
protected void	**setFreeTrue**()

protected void	**setPlateFalse**()
protected void	**setPlateTrue**()

- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Field Detail***
- **hasPlate[Show source in BlueJ]**
- protected boolean hasPlate
- **isFree[Show source in BlueJ]**
- protected boolean isFree
- **speed[Show source in BlueJ]**
- protected int speed
- ***Constructor Detail***
- **Targets[Show source in BlueJ]**
- public Targets()
- ***Method Detail***
- **getFree[Show source in BlueJ]**
- protected boolean getFree()
- **getPlate[Show source in BlueJ]**
- protected boolean getPlate()
- **move[Show source in BlueJ]**
- protected void move()

- **moveAround[Show source in BlueJ]**
- protected void moveAround()
- **setFreeFalse[Show source in BlueJ]**
- protected void setFreeFalse()
- **setFreeTrue[Show source in BlueJ]**
- protected void setFreeTrue()
- **setPlateFalse[Show source in BlueJ]**
- protected void setPlateFalse()
- **setPlateTrue[Show source in BlueJ]**
- protected void setPlateTrue()

Class CuttingBoard

- java.lang.Object
    - greenfoot.Actor
        - Targets
            - CuttingBoard

-

- public class **CuttingBoard**
  extends Targets
- Write a description of class target3 here.
- **Version:**
- October 2018
- **Author:**
- Daniel

- ***Field Summary***
- **Fields inherited from class Targets**
- hasPlate, isFree, speed

- ***Constructor Summary***
- **Constructor and Description**
- **CuttingBoard**()

- ***Method Summary***

Modifier and Type	Method and Description
void	**act**() Act - do whatever the target3 wants to do.

- **Methods inherited from class Targets**
- getFree, getPlate, move, moveAround, setFreeFalse, setFreeTrue, setPlateFalse, setPlateTrue

- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail***
- **CuttingBoard[Show source in BlueJ]**
- public CuttingBoard()
- ***Method Detail***
- **act[Show source in BlueJ]**
- public void act()
- Act - do whatever the target3 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 Dishrack

- java.lang.Object
    - greenfoot.Actor
        - Targets
            - Dishrack
-

- public class **Dishrack**
  extends Targets
- Write a description of class Dishrack here.
- **Version:**
- October 2018
- **Author:**
- Daniel
- ***Field Summary***
- **Fields inherited from class Targets**
- hasPlate, isFree, speed
- ***Constructor Summary***
    - **Constructor and Description**
    - **Dishrack**()
- ***Method Summary***

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

- **Methods inherited from class Targets**
- getFree, getPlate, move, moveAround, setFreeFalse, setFreeTrue, setPlateFalse, setPlateTrue
- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail***
- **Dishrack[Show source in BlueJ]**
- public Dishrack()
- ***Method Detail***
- **act[Show source in BlueJ]**
- public void act()
- Act - do whatever the Dishrack 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 FoodBox

- java.lang.Object
    - greenfoot.Actor
        - Targets
            - FoodBox

-

- public class **FoodBox**
  extends Targets
- Write a description of class target2 here.
- **Version:**
- October 2018
- **Author:**
- Daniel
- ***Field Summary***
- **Fields inherited from class Targets**
- hasPlate, isFree, speed
- ***Constructor Summary***
- **Constructor and Description**
- **FoodBox**()
- ***Method Summary***
- **Methods inherited from class Targets**
- getFree, getPlate, move, moveAround, setFreeFalse, setFreeTrue, setPlateFalse, setPlateTrue
- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail***
- **FoodBox[Show source in BlueJ]**

- public FoodBox()

Class FoodPickup

- java.lang.Object
    - greenfoot.Actor
        - Targets
            - FoodPickup

- public class **FoodPickup**
  extends Targets
- Write a description of class FoodPickup here.
- **Version:**
- October 2018
- **Author:**
- Daniel

- ***Field Summary***
- **Fields inherited from class Targets**
- hasPlate, isFree, speed

- ***Constructor Summary***
- **Constructor and Description**
- **FoodPickup**()
- Constructor

- ***Method Summary***
- **Methods inherited from class Targets**
- getFree, getPlate, move, moveAround, setFreeFalse, setFreeTrue, setPlateFalse, setPlateTrue
- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- ***Constructor Detail***

- **FoodPickup[Show source in BlueJ]**
- `public FoodPickup()`
- Constructor

Class FryingPan

- java.lang.Object
    - greenfoot.Actor
        - Targets
            - FryingPan
-

- public class **FryingPan**
  extends Targets
- Write a description of class FryingPan here.
- **Version:**
- October 2018
- **Author:**
- Daniel

- ***Field Summary***
- **Fields inherited from class Targets**
- hasPlate, isFree, speed
- ***Constructor Summary***
  - **Constructor and Description**
  - **FryingPan**()
- ***Method Summary***
- **Methods inherited from class Targets**
- getFree, getPlate, move, moveAround, setFreeFalse, setFreeTrue, setPlateFalse, setPlateTrue
- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail***
- **FryingPan[Show source in BlueJ]**

- public FryingPan()

Class Home

- java.lang.Object
    - greenfoot.Actor
        - Targets
            - Home

-

- public class **Home**
  extends Targets
- Write a description of class target1 here.
- **Version:**
- October 2018
- **Author:**
- Daniel

- ***Field Summary***
- **Fields inherited from class Targets**
- hasPlate, isFree, speed
- ***Constructor Summary***

Constructor and Description
Home()

- ***Method Summary***

Modifier and Type	Method and Description
void	**act**() Act - do whatever the target1 wants to do.

- **Methods inherited from class Targets**
- getFree, getPlate, move, moveAround, setFreeFalse, setFreeTrue, setPlateFalse, setPlateTrue
- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail***
- **Home[Show source in BlueJ]**
- public Home()
- ***Method Detail***
- **act[Show source in BlueJ]**
- public void act()
- Act - do whatever the target1 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 RiceAndNoodles

- java.lang.Object
    - greenfoot.Actor
        - Targets
            - RiceAndNoodles
-

- public class **RiceAndNoodles**
  extends Targets
- Write a description of class RiceAndNoodles here.
- **Version:**
- October 2018
- **Author:**
- Daniel

- ***Field Summary***
- **Fields inherited from class Targets**
- hasPlate, isFree, speed

- ***Constructor Summary***

Constructor and Description
RiceAndNoodles()

- ***Method Summary***

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

- **Methods inherited from class Targets**
- getFree, getPlate, move, moveAround, setFreeFalse, setFreeTrue, setPlateFalse, setPlateTrue

- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail***
- **RiceAndNoodles[Show source in BlueJ]**
- public RiceAndNoodles()
- ***Method Detail***
- **act[Show source in BlueJ]**
- public void act()
- Act - do whatever the RiceAndNoodles 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 ServeryWindow

- java.lang.Object
    - greenfoot.Actor
        - Targets
            - ServeryWindow

- public class **ServeryWindow**
  extends Targets
- Write a description of class target5 here.
- **Version:**
- October 2018
- **Author:**
- Daniel

- ***Field Summary***
- **Fields inherited from class Targets**
- hasPlate, isFree, speed

- ***Constructor Summary***

Constructor and Description
ServeryWindow()

- ***Method Summary***

Modifier and Type	Method and Description
void	**act**() Act - do whatever the target5 wants to do.

- **Methods inherited from class Targets**
- getFree, getPlate, move, moveAround, setFreeFalse, setFreeTrue, setPlateFalse, setPlateTrue

- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail***
- **ServeryWindow[Show source in BlueJ]**
- public ServeryWindow()
- ***Method Detail***
- **act[Show source in BlueJ]**
- public void act()
- Act - do whatever the target5 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 Timer

- java.lang.Object
    - greenfoot.Actor
        - Timer

-

- public class **Timer**
  extends greenfoot.Actor
- Acts as a countdown from a desired time to zero seconds. Designed to show when something can be used again. Multiple instances can be created as long as the trigger keys are different.
- **Version:**
- October 2018
- **Author:**
- Alexander Yee with help from Jordan Cohen

- ***Constructor Summary***

Constructor and Description
Timer(double time, int r, int g, int b, double width, double height) Creates an ability bar with full customization, allowing the user to change countdown time, colour and size.

- ***Method Summary***

Modifier and Type	Method and Description
void	**act**() Counts in real time and checks when to decrease the ability bar height.
boolean	**isDone**() Indicated when timer 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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail***
- **Timer[Show source in BlueJ]**
-
```
public Timer(double time,
             int r,
             int g,
             int b,
             double width,
             double height)
```
- Creates an ability bar with full customization, allowing the user to change countdown time, colour and size.
- **Parameters:**
- time - The countdown time of the ability in seconds. Recommended to use less than 40 seconds
- key - The keyboard input that will begin the countdown. Must be one character or a single input
- r - The red component of colour. Ranges from 0 - 255
- g - The green component of colour. Ranges from 0 - 255
- b - The blue component of colour. Ranges from 0 - 255
- width - The width of the ability bar in pixels. Recommended to use between 1 and 100
- height - The height of the ability bar in pixels. Recommended to use between 30 and 100
- ***Method Detail***
- **act[Show source in BlueJ]**
- public void act()
- Counts in real time and checks when to decrease the ability bar height. This method is called whenever the 'Act' or 'Run' button gets pressed in the environment.
- **Overrides:**
- act in class greenfoot.Actor
- **isDone[Show source in BlueJ]**

- public boolean isDone()
- Indicated when timer is finished
-

Class Washing

- java.lang.Object
    - greenfoot.Actor
        - Washing
-

- public class **Washing**
  extends greenfoot.Actor
- Write a description of class Washing here.
- **Version:**
- October 2018
- **Author:**
- Alexander
- ***Constructor Summary***
    - **Constructor and Description**
    - **Washing**()
- ***Method Summary***

Modifier and Type	Method and Description
void	**act**()
void	**addedToWorld**(greenfoot.World w)
void	**animation**()

- **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, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
- ***Constructor Detail***
- **Washing[Show source in BlueJ]**
- public Washing()

- ***Method Detail***
- **act[Show source in BlueJ]**
- public void act()
- **Overrides:**
- act in class greenfoot.Actor
- **addedToWorld**
- public void addedToWorld(greenfoot.World w)
- **Overrides:**
- addedToWorld in class greenfoot.Actor
- **animation[Show source in BlueJ]**
- public void animation()