```
1  import greenfoot.*;
2  import java.awt.Color;
3  import java.io.*;
4  import java.util.*;
5
6  /**
7   * Germany World
8   *
9   * Candidate Number: gkm599 001242 0082
10  * v. 1.0 Mar 2017
11  */
12 public class Germany extends World
13 {
14     Sprite girl;
15     Human j;
16     Human o;
17     Instruction instruction;
18     Counter interactionCount;
19     Counter foodEaten;
20     String[] foodNames =
21         {"haribo", "apfelschorle", "spaetzle", "kinder", "schnitzel", "c
   urrywurst"};
22     String[] alphaFood;
23     Boolean isShowing;
24
25     public Germany()
26     {
27         super(850, 400, 1);
28         addHumans();
29         addSprite();
30         showInstructions();
31         addCounters();
32         initializeArray();
33         isShowing = false;
34     }
35
36     public void act()
37     {
38         showFood();
39         displayWinMsg();
40     }
41
42     public void addSprite()
43     {
44         girl = new Sprite("sprite.png");
45         addObject ( girl , 30, 210);
46     }
47
48     public void addHumans()
```

```
49      {
50          j = new Human ("jannik", "jannik.png", "jannik.txt");
51          o = new Human ("jonas", "jonas.png", "jonas.txt");
52          addObject (j , 150, 210);
53          addObject (o, 300, 200);
54
55      }
56
57      public void showInstructions()
58      {
59          instruction = new Instruction();
60          addObject (instruction, 290, 200);
61      }
62
63      public void addCounters()
64      {
65          interactionCount = new Counter();
66          addObject (interactionCount, 50, 370);
67
68          foodEaten = new Counter();
69          addObject (foodEaten, 50, 340);
70
71          GreenfootImage bg = getBackground();
72          bg.setColor(Color.WHITE);
73          bg.setFont(new java.awt.Font("Arial", 1 , 18));
74          bg.drawString("Interaction Points", 100, 375);
75          bg.drawString("Number of Foods Eaten", 100, 345);
76
77      }
78
79      public Counter getIntCounter()
80      {
81          return interactionCount;
82      }
83
84      public Counter getFoodCounter()
85      {
86          return foodEaten;
87      }
88
89      public void initializeArray()
90      {
91          List<String> food = new ArrayList();
92          int count = foodNames.length;
93          for (int i = 0; i< count; i++) {
94              food.add(foodNames[i]);
95          }
96          java.util.Collections.sort(food);
97          alphaFood = food.toArray(new String[food.size()]);
```

```
98
99      }
100
101     public void showFood()
102     {
103         boolean isMarket = girl.returnIsMarket();
104         int i = 0;
105         if (isShowing == false && isMarket == true){
106             int count = foodNames.length;
107             int xcoord = 630;
108             int ycoord = 280;
109             while (i < count)
110             {
111                 addObject(new Food((alphaFood[i]), (alphaFood[i]+"png"),
    (alphaFood[i]+"big.png")), xcoord + 85*i, ycoord);
112                 i++;
113                 if (i % 3 == 0) {
114                     ycoord = ycoord + 80;
115                     xcoord = xcoord - (85*3);
116                 }
117
118             }
119             isShowing = true;
120         }
121         else if (isShowing == true && isMarket == false)
122         {
123             removeObjects(getObjects(Food.class));
124             isShowing = false;
125         }
126     }
127
128     public void displayWinMsg()
129     {
130         int fE = foodEaten.getValue();
131         if (fE>=6)
132         {
133             GreenfootImage bg = getBackground();
134             bg.setColor(Color.WHITE);
135             bg.fillRect(130, 25, 340, 80);
136             bg.setColor(Color.GREEN);
137             bg.setFont(new java.awt.Font("Arial", 1 , 70));
138             bg.drawString("YOU WIN!", 140, 100);
139             Greenfoot.stop();
140
141         }
142     }
143 }
144
```

```
1  import greenfoot.*;
2  import java.awt.Color;
3  import java.io.*;
4
5  /**
6   * Human Class, will talk when interacted with.
7   *
8   * Candidate Number: gkm599 001242 0082
9   * v. 1.0 Mar 2017
10  *
11  **/
12 public class Human extends Actor
13 {
14     private String himage;
15     private String name;
16     private String file;
17     private String lineArray [];
18     private int convNumber;
19     public Human (String humanName, String humanImage, String fileName)
20     {
21         this.name = humanName;
22         himage = humanName + ".png";
23         himage = humanImage;
24         setImage (himage);
25         file = humanName + ".txt";
26         file = fileName;
27         try {
28             lineArray = readText();
29         }
30         catch (IOException ioe) {
31             GreenfootImage lineImage = new GreenfootImage("Error: Can't
   Read Text", 16, Color.WHITE, Color.BLACK);
32             Actor text = new TextActor();
33             text.setImage(lineImage);
34             getWorld().addObject(text, 720, 40);
35         }
36     }
37
38     public void act()
39     {
40         //
41     }
42
43     public String[] readText() throws IOException
44     {
45         InputStream input = getClass().getClassLoader().getResourceAsStr
   eam(file);
46         BufferedReader reader = new BufferedReader(new InputStreamReader
   (input));
```

```
47
48          String[] lineArray = new String[10];
49
50          int i = 0;
51          String l = null;
52          while ((l = reader.readLine())!=null){
53              lineArray[i] = l;
54              i = i + 1;
55          }
56
57          return lineArray;
58      }
59
60
61      public String getName()
62      {
63          return name;
64      }
65
66      public int getConvNumber()
67      {
68          return convNumber;
69      }
70
71      public String speak()
72      {
73          convNumber = convNumber + 1;
74          if (convNumber >=10) {
75              return lineArray[0];
76              // after you've cycled through a character's conversation, w
    ill urge you to converse with others.
77          }
78          else {
79              return lineArray[convNumber];
80          }
81      }
82          public void defaultMessage()
83      {
84          if (convNumber == 0 ){
85              Actor speech = new TextActor();
86              speech.setImage("nontspeech.png");
87              getWorld().addObject(speech, 715, 150);
88              GreenfootImage lineImage = new GreenfootImage( "Hey! I'm " +
     name + ". Press 'space' \n to talk to me." , 16, Color.BLACK, new Color
     (0,0,0,0));
89              Actor text = new TextActor();
90              text.setImage(lineImage);
91              getWorld().addObject(text, 710, 60);
92          }
```

```
93        }
94    }
```

```
1  import greenfoot.*;  // (World, Actor, GreenfootImage, Greenfoot and Mou
   seInfo)
2  import java.awt.Color;
3
4  /**
5   * A Counter class that allows you to display a numerical value on scree
   n.
6   *
7   * The Counter is an actor, so you will need to create it, and then add
   it to
8   * the world in Greenfoot.  If you keep a reference to the Counter then
   you
9   * can adjust its value.
10  *
11  * @author Neil Brown and Michael KÃ¶lling
12  * @version 1.0
13  */
14 public class Counter extends Actor
15 {
16     private static final Color transparent = new Color(0,0,0,0);
17     private GreenfootImage background;
18     private int value;
19     private int target;
20
21     public Counter()
22     {
23         this(new String());
24     }
25
26     /**
27      * Create a new counter, initialised to 0.
28      */
29     public Counter(String prefix)
30     {
31         background = getImage();  // get image from class
32         value = 0;
33         target = 0;
34         updateImage();
35     }
36
37     /**
38      * Animate the display to count up (or down) to the current target v
   alue.
39      */
40     public void act()
41     {
42         if (value < target) {
43             value++;
44             updateImage();
```

```
45          }
46          else if (value > target) {
47              value--;
48              updateImage();
49          }
50      }
51
52      /**
53       * Add a new score to the current counter value.  This will animate
54       * the counter over consecutive frames until it reaches the new valu
    e.
55       */
56      public void add(int score)
57      {
58          target = target+ score;
59          updateImage();
60      }
61
62      /**
63       * Return the current counter value.
64       */
65      public int getValue()
66      {
67          return target;
68      }
69
70      /**
71       * Update the image on screen to show the current value.
72       */
73      private void updateImage()
74      {
75          GreenfootImage image = new GreenfootImage(background);
76          GreenfootImage text = new GreenfootImage(" " + value, 22, Color.
    BLACK, transparent);
77
78          if (text.getWidth() > image.getWidth() - 20)
79          {
80              image.scale(text.getWidth() + 20, image.getHeight());
81          }
82
83          image.drawImage(text, (image.getWidth()-text.getWidth())/2,
84                          (image.getHeight()-text.getHeight())/2);
85          setImage(image);
86      }
87  }
88
```

```
1  import greenfoot.*;
2  import java.awt.Color;
3  import java.io.*;
4
5  /**
6   * Sprite to manipulate, walk it around the screen to interact with Huam
   ns and places.
7   *
8   * Candidate Number: gkm599 001242 0082
9   * v. 1.0 Mar 2017
10  */
11 public class Sprite extends Actor
12 {
13     private String sprite;
14     private Boolean isIntersect; //true if in midst of interaction
15     private Boolean isDown;
16     private Human isHuman;
17     private Boolean isMarket;
18     private Boolean hSpeaking;
19
20     public Sprite (String spriteImage)
21     {
22         sprite = spriteImage;
23         isIntersect = false;
24         isDown = false;
25         isMarket = false;
26         hSpeaking = false;
27     }
28
29     /**
30      * Act - do whatever the Sprite wants to do. This method is called w
   henever
31      * the 'Act' or 'Run' button gets pressed in the environment.
32      */
33     public void act()
34     {
35         checkKeyPress();
36         checkForPerson();
37         checkForPlace();
38         humanSpeak();
39     }
40
41     public void checkKeyPress()
42     {
43         int speed = 3;
44         if (Greenfoot.isKeyDown("left") && getX() > 20 )
45         {
46             move(-speed);
47         }
```

```
48          if (Greenfoot.isKeyDown("right") && getX() < 560)
49          {
50              move(speed);
51          }
52          if (Greenfoot.isKeyDown("down") && getY() < 220)
53          {
54              setLocation(getX(), getY()+speed);
55          }
56          if (Greenfoot.isKeyDown("up") && getY() > 200)
57          {
58              setLocation(getX(), getY()-speed);
59          }
60      }
61
62      public void checkForPerson()
63      {
64          isHuman = (Human) getOneIntersectingObject(Human.class);
65          if (isHuman !=null) {
66              if (isIntersect == false) {
67                  isHuman.defaultMessage();
68                  hSpeaking = true;
69              }
70              isIntersect = true;
71          }
72
73          else if (getWorld().getObjects(TextActor.class) != null) {
74              getWorld().removeObjects(getWorld().getObjects(TextActor.cla
    ss));
75              isIntersect = false;
76              hSpeaking = false;
77          }
78      }
79
80      public void showSpeak()
81      {
82          String line = isHuman.speak();
83          String[] words = line.split("-");
84          int height = 50;
85          int count = words.length;
86
87          for (int i = 0; i< count; i++) {
88              GreenfootImage lineImage = new GreenfootImage( words[i] , 16
    , Color.BLACK, new Color (0,0,0,0));
89              Actor text = new TextActor();
90              text.setImage(lineImage);
91              getWorld().addObject(text, 710, height);
92              height = height + 20;
93          }
94      }
```

```
95
96      public void humanSpeak()
97      {
98          if (hSpeaking == true){
99              if ( isDown == false &&  (Greenfoot.isKeyDown("space") == tr
    ue )) {
100                 isDown = true;
101                 int cn = isHuman.getConvNumber();
102
103                 if (cn<10){
104                     givePoints();
105                 }
106                 Actor speech = new TextActor();
107                 speech.setImage("nontspeech.png");
108                 getWorld().addObject(speech, 715, 150);
109                 showSpeak();
110             }
111             if ( isDown == true  && (Greenfoot.isKeyDown("space") == fal
    se )) {
112                 isDown = false;
113             }
114         }
115     }
116
117     public void checkForPlace()
118     {
119         //Market
120         if (getX() < 75 && getX() > 55 && getY() >= 198 && getY() < 203
    && isMarket == false)
121         {
122             getWorld().addObject(new Market(), 715, 200);
123             isMarket = true;
124         }
125         else if (isMarket == true && (getX() < 55 || getX() > 75 || getY
    () >= 203)){
126             isMarket= false;
127             getWorld().removeObjects(getWorld().getObjects(Market.class)
    );
128         }
129     }
130
131     public boolean returnIsMarket()
132     {
133         return isMarket;
134     }
135
136     public void givePoints()
137     {
138         Germany germany = (Germany) getWorld();
```

```
139         Counter ic = germany.getIntCounter();
140         ic.add(6);
141     }
142 }
143
144
```

```
1  import greenfoot.*;
2  import java.awt.Color;
3
4  /**
5   * Displays instruction box at beginning of game.
6   *
7   *  Candidate Number: gkm599 001242 0082
8   * v 1.0 Mar 2017
9   */
10 public class Instruction extends Actor
11 {
12     public Instruction()
13     {
14          GreenfootImage image = new GreenfootImage ("instructionbox.png"
   );
15         image.setColor(Color.BLACK);
16         image.setFont(new java.awt.Font("Arial", 1 , 13));
17         image.drawString("Move the sprite with up, down, left, right arr
   ow keys.", 10 ,40 );
18         image.drawString("Interact with other characters to learn about
   Germany!", 10 ,57 );
19         image.drawString("Earn interaction points by talking to people!"
   , 10 ,74 );
20          image.drawString("Buy ALL food with points to WIN.", 10 ,91);
21          image.setColor(Color.RED);
22          image.drawString("Press 'space' to continue.", 90 ,112);
23         setImage (image);
24
25     }
26     /**
27      * Act - do whatever the Instruction wants to do. This method is cal
   led whenever
28      * the 'Act' or 'Run' button gets pressed in the environment.
29      */
30     public void act()
31     {
32
33        hideInstructions();
34     }
35     public void hideInstructions()
36     {
37         if (Greenfoot.isKeyDown("space"))
38         {
39             getWorld().removeObject(this);
40         }
41     }
42     }
43
44
```

45

Class Food


```
1  import greenfoot.*;  // (World, Actor, GreenfootImage, Greenfoot and Mou
   seInfo)
2  import java.awt.Color;
3  import java.io.*;
4
5  /**
6   * Food items to be displayed in the shop.
7   *
8   * Candidate Number: gkm599 001242 0082
9   * v. 1.0 Mar 2017
10  */
11 public class Food extends Actor
12 {
13     Boolean mouseOver;
14     Boolean isMouseDown;
15     Boolean isShowing;
16     String fImage;
17     String fName;
18     String fileName;
19     String bImage;
20     String[] lineArray;
21     public Food (String foodName, String foodImage, String bigImage){
22         fName = foodName;
23         foodImage = foodName + ".png";
24         fImage = foodImage;
25         setImage(foodImage);
26         bImage = foodName + "big.png";
27         bImage =  bigImage;
28         mouseOver = false;
29         fileName = fName + ".txt";
30         isMouseDown = false;
31
32         try {
33             lineArray = readText();
34         }
35         catch (IOException ioe) {
36             GreenfootImage lineImage = new GreenfootImage("Error: Can't
   Read Text", 16, Color.WHITE, Color.BLACK);
37             Actor text = new TextActor();
38             text.setImage(lineImage);
39             getWorld().addObject(text, 720, 40);
40         }
41     }
42
43     /**
44      * Act - do whatever the Food wants to do. This method is called whe
   never
45      * the 'Act' or 'Run' button gets pressed in the environment.
46      */
```

```
47      public void act()
48      {
49          // display the big image in the pale box when mouse over small i
    mage
50          MouseInfo mouse = Greenfoot.getMouseInfo();
51
52          if (!mouseOver && Greenfoot.mouseMoved(this))
53          {
54              mouseOver = true;
55          }
56          if (mouseOver == true )
57          {
58              // small speech bubble
59              Actor speech = new TextActor();
60              speech.setImage("small speech.png");
61              getWorld().addObject(speech, 710, 90);
62
63              // show large image
64              Actor img = new TextActor();
65              img.setImage(bImage);
66              getWorld().addObject(img, 710, 170);
67
68              // show name;
69              GreenfootImage lineImage = new GreenfootImage(fName, 16, Col
    or.BLACK, Color.WHITE);
70              Actor text = new TextActor();
71              text.setImage(lineImage);
72              getWorld().addObject(text, 710, 200);
73
74              showDescription();
75
76              if (Greenfoot.mouseClicked(this) && isMouseDown == false)
77              {
78                  buyAnItem();
79                  isMouseDown = true;
80              }
81              else if (Greenfoot.mouseClicked(null) && isMouseDown == true
    )
82              {
83                  isMouseDown = false;
84              }
85          }
86          if (mouseOver && Greenfoot.mouseMoved(null) && ! Greenfoot.mouse
    Moved(this))
87          {
88              mouseOver = false;
89          }
90      }
91
```

```
92      private void showDescription()
93      {
94          String line = lineArray[0];
95          String[] words = line.split("-");
96          int height = 50;
97          int count = words.length;
98
99          for (int i = 0; i< count; i++) {
100             GreenfootImage lineImage = new GreenfootImage( words[i] , 16
    , Color.BLACK, new Color (0,0,0,0));
101             Actor text = new TextActor();
102             text.setImage(lineImage);
103             getWorld().addObject(text, 710, height);
104             height = height + 20;
105         }
106
107     }
108
109     public String[] readText() throws IOException
110     {
111         InputStream input = getClass().getClassLoader().getResourceAsStr
    eam(fileName);
112         BufferedReader reader = new BufferedReader(new InputStreamReader
    (input));
113
114         String[] lineArray = new String[3];
115
116         int i = 0;
117         String l = null;
118         while ((l = reader.readLine())!=null){
119             lineArray[i] = l;
120             i = i + 1;
121         }
122
123         return lineArray;
124     }
125
126     private void buyAnItem()
127     {
128         Germany germany = (Germany) getWorld();
129         Counter ic = germany.getIntCounter();
130         Counter fc = germany.getFoodCounter();
131         int wallet = ic.getValue();
132         if (wallet >= 20){
133             ic.add(-20);
134             fc.add(1);
135         }
136
137     }
```

```
138
139 }
140
```