technocamps

Greenfoot Ecosystems Session Plan

UNDEB EWROPEAIDD
EUROPEAN UNION

Llywodraeth Cymru
Welsh Government

Cronfa Gymdeithasol Ewrop
European Social Fund

Prifysgol Abertawe
Swansea University

Cardiff University
Prifysgol Caerdydd

Prifysgol Bangor University

Cardiff Metropolitan University
Prifysgol Metropolitan Caerdydd

University of South Wales
Prifysgol De Cymru

i.t.wales

Prifysgol Aberystwyth University

Prifysgol Glyndŵr Wrecsam | Wrexham Glyndŵr University

Workshop Schedule

Introduction - 10 minutes

Pre-Day Questionnaires - 5 minutes

Introduction to Food Chains - 30 minutes

Object-Oriented Programming - 30 minutes

Greenfoot - Producers - 1 hour

Food Chains and Food Webs - 1 hour

Greenfoot - Consumers - 1 hours 20 minutes

Post-Day Questionnaires - 10 minutes

Note: These are estimated times, these will vary between classes, schools etc. so times will need to be adjusted accordingly.

Total - 4 hours 45 minutes

Learning Outcomes

Preparation

1. Ensure all computers have the correct version of Greenfoot installed (2.4.2) and ready to be used.

2. Print out Greenfoot handouts of slides and student workbooks.

Learning Outcomes

1. Good understanding of Ecosystems and Habitat in Science Curriculum.

2. Experience of programming in Greenfoot.

Attendee Prerequisites

1. No previous Greenfoot experience required.

Session Plan Key

In this session plan we use the following colours to differentiate the types of activities:

- **Yellow - Explain.** Teachers should explain the slide/example to the class.
- **Green - Discuss.** Teachers should start an open discussion with the class to get them to feedback some answers/ideas.
- **Purple - Activity.** Students are expected to complete an activity whether it be in their workbooks or on the computer, followed by a discussion of their solutions.
- **Green - Introduction/Conclusion.** The introduction/conclusion is also colour coded green. Teachers should hand out materials in the introduction and conclude the day and collect materials at the end.

Introduction

Begin with introductions, and a brief explanation of the Technocamps programme, before handing out pre-day questionnaires to be filled out by the students and teacher.

Explain: Topics Covered Today

Today, we will be looking at ta topic within Biology called Ecosystems and Habitats. In particular we will learn about: food chains, food webs and energy transfer.

We will also be doing some programming in Greenfoot to create a game based on Ecosystems and Habitats.

Activity: What are Food Chains?

Ask the students the following questions:

- Have you ever heard of food chains before?
- What do you know about food chains?

Give the students 30 seconds in pairs to discuss before asking them to write down their answers in their workbooks. Ask them to explain their answers.

Explain: Food Chains & Organisms

A food chain shows the flow of energy from one living to another.

A habitat the natural home or environment of an animal, plant or other organisms.

Explain that an organisms is an individual animal, plant or a single-celled life form.

Activity: Energy Source

What is the ultimate source of energy for most living things?
Does anyone know what kind of cell this is?
Extension: Do you know what 1-7 are?

Solutions:
Light energy from the Sun
Plant cell
Extension: 1) Mitochondrion, 2) Cytoplasm, 3) Nucleus, 4) Cell wall, 5) Cell membrane, 6) Vacuole, 7) Chloroplast

Activity: What is a Producer?

Students should write down in their workbooks what they think a producer is, as well as an example of a producer.

Explain: Producers

Producers are organisms that produce their own organic nutrients (food).

Introduce algae if not previously mentioned. Algae are a diverse group of aquatic organisms that have the ability to conduct photosynthesis. Seaweed is a type of Algae.

Explain: Object Oriented Programming

Explain the differences between Python which most students will be familiar with and object oriented programming.

It is a way of programming which is slightly different to how we would usually use Python, it is structured differently to Python's sequential way of coding as Java uses Classes and Objects.

Discuss: Classes and Objects

Ask if anyone knows what Classes and Objects are and if they have used them before.

Explain: Classes and Objects

A class is a framework or a blueprint for an object. It contains all the information about an object, features, what it can do etc.

An object can only have features and functions which are defined in its class.

For example think about the class for ice cream. All ice creams are made in the same way and need to be kept in the freezer. They all contains similar ingredients, such as milk, eggs and sugar, but different flavours will contain other ingredients, such as vanilla, cocoa powder, mint, chocolate chips.

Another example is the Student Class. Think about what properties does a student have? What makes a student as student?
Students objects have a name, a age, a gender and a favourite subject.

Activity: Actors and World

A picture of Super Mario is shown on the slides, explain that the classes given in Greenfoot are either Actors or World. Ask students to discuss and write a list of which objects would be of the class Actors and which would be of the World class.
This depends on what these things can do, if they move up and down for example, they cannot be part of the background (world).

For Super Mario, the blue sky, ground blocks, cloud and trees are the background. The mystery block, brick block, Mario and super mushroom are all actors as they all do something and can change/be interacted with.
Ask students whether they think the score, coin counter, level and timer are part of the World class or Actor class.

Explain: What is Greenfoot?

Greenfoot is an introductory visual programming environment using the “Java” programming language. Java is a highly valued language in the Computer Science Industry.

Explain: Greenfoot Version 2.4.2

The version we will be using is Greenfoot Version 2.4.2 in order to make sure we’re all on the same page and have the same methods available to us.

Activity: Greenfoot

Follow the slides on the board to start the game, change the backgrounds appearance and dimensions, create a new Producer class, instantiate objects of the Producer class and add them to the world.

Explain: Greenfoot Coordinate System

Greenfoots coordinate system is slightly different to that used in scratch or in everyday life drawing graphs. The x axis goes from left to right as expected, but the y axis increases as you move down the page.
This is because of how cathode ray tubes worked which were/are used in monitors

Activity: Name the Process

Students should write down in their workbooks what they think the name of the process through which a producer makes food is called.

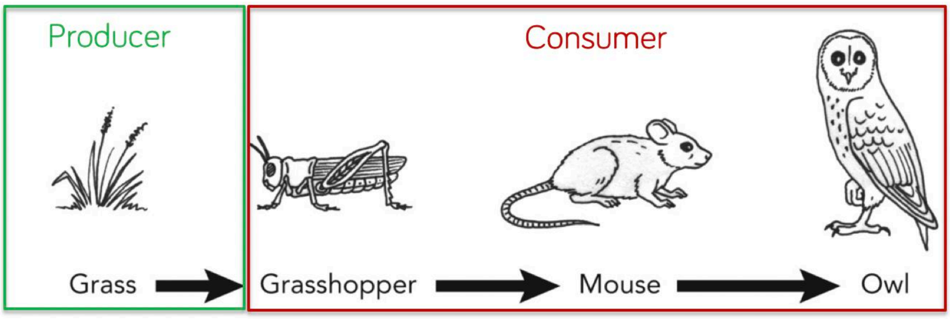
Solution: Photosynthesis, discussed more in depth on the next slide.
Extension: Discuss how this process works.

Explain: Photosynthesis

Producers are organisms that make their own organic nutrients or food - usually using energy from sunlight. This is called photosynthesis. Photosynthesis is the chemical process where plants make glucose (or carbohydrates_ and oxygen from carbon dioxide and water using light energy.

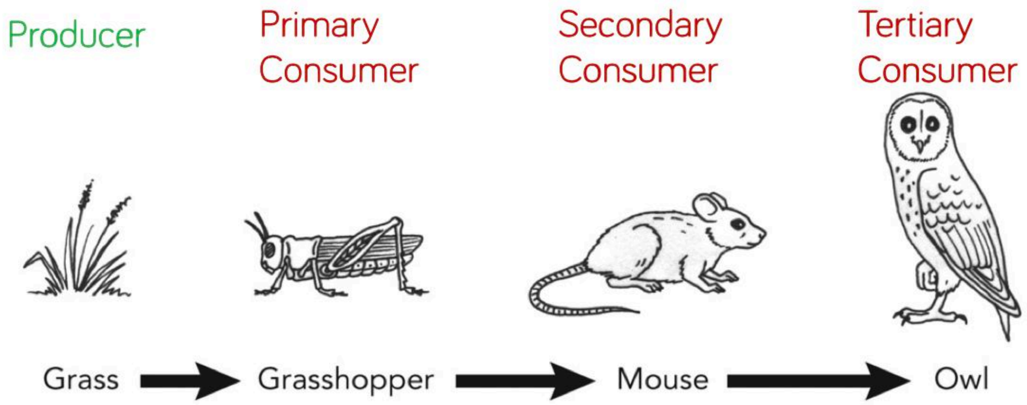
Explain: Consumers

The other organisms in the food chain are consumers, animals that eat plants or other animals, they get all their energy by consuming other organisms.



Explain: Consumer Positions

Consumers in the food chain have different names depending on their position relative to the Producer. These are Primary consumer, Secondary consumer, tertiary consumer and so on. The top end of the food chain is often called an Apex Predator.



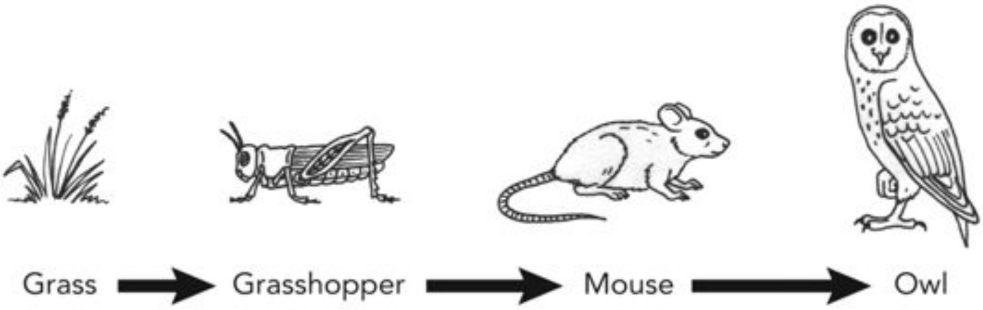
Organism	How it gets its energy
Consumer	Feeding on other organisms
Primary consumer	Eating plants
Secondary consumer	Eating primary consumers
Tertiary consumer	Eating secondary consumers
Herbivore	Eating plants
Carnivore	Eating other animals
Decomposer	Feeding on dead and decaying organisms, and on the undigested parts of plant and animal matter in faeces

Activity: Consumer Positions

Students should match the consumer position name to each animal in the food chain.

Explain: Example Food Chain

In this example, grass is eaten by grasshoppers, grasshoppers are eaten by mice and mice are eaten by owls. The arrows between each organism in the chain always point in the direction of energy flow from the food to the feeder.



Activity: Food Chain

Students should group the organisms on the following slide into producers and consumers as well as herbivores and carnivores.

Then using these groupings they should construct food chains.

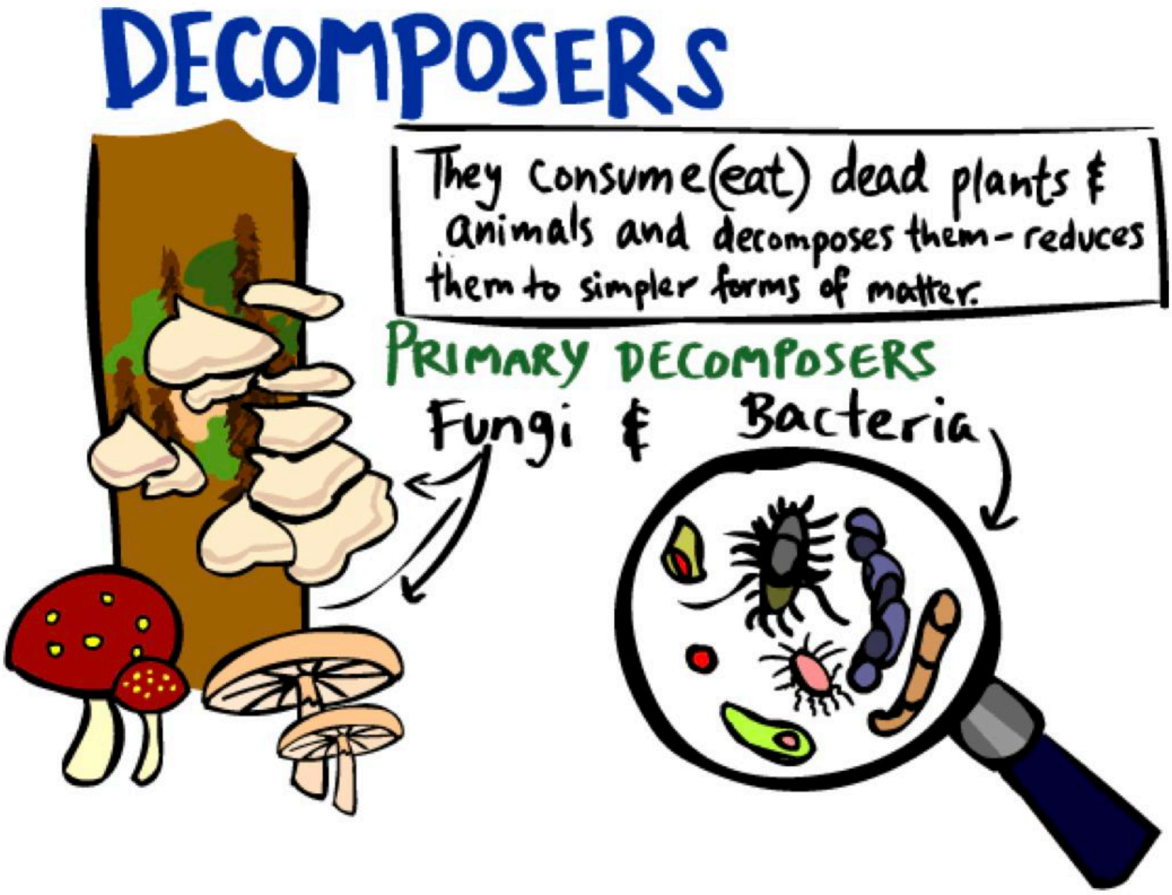
From these food chains the students should identify the primary, secondary and tertiary consumers however this will depend on the specific food chain they have constructed.

Activity: PrimaryConsumer

Students are to create a second Actor subclass called PrimaryConsumer. They should set the image to a mouse and then instantiate two objects of the PrimaryConsumer and get them to display in the middle of your habitat.

Explain: Decomposers

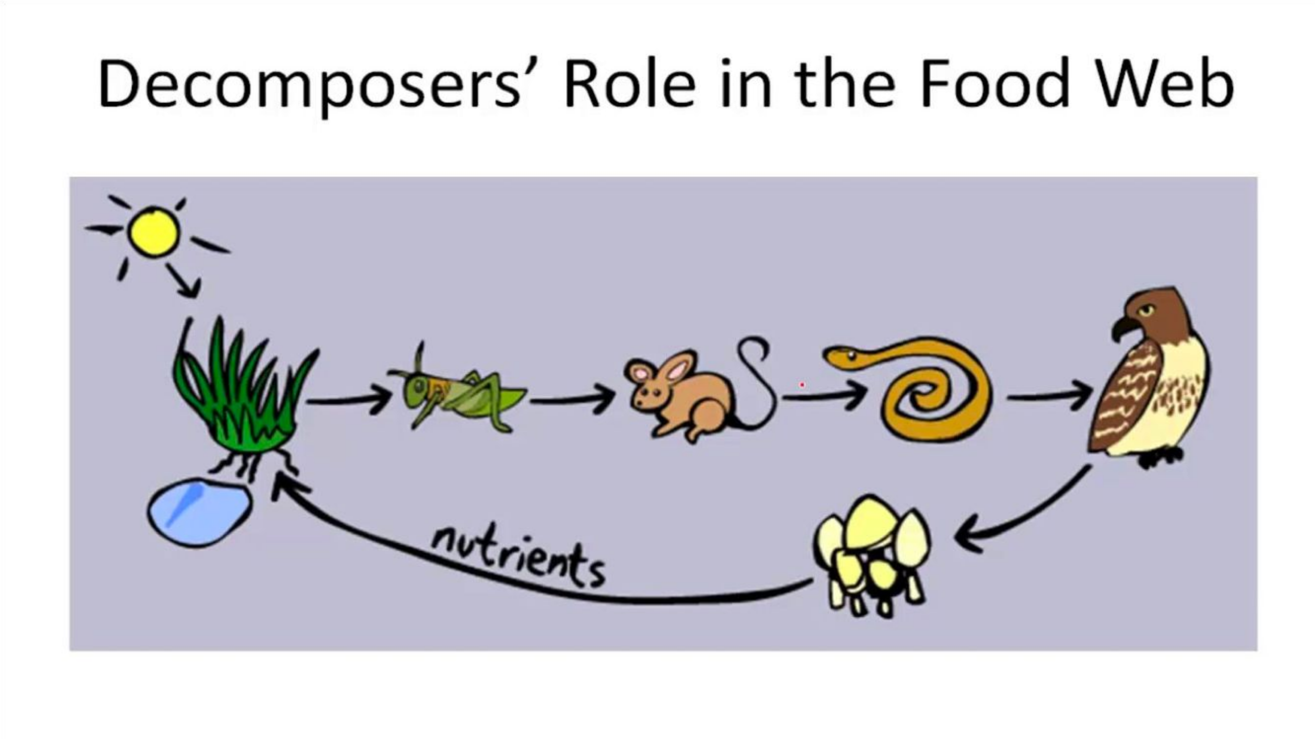
Decomposers are chiefly the micro-organisms such as bacteria and fungi in the form of yeasts and moulds that break down bodies of dead organisms and release compounds that can be used by producers.


DECOMPOSERS
They consume (eat) dead plants & animals and decomposes them - reduces them to simpler forms of matter.
PRIMARY DECOMPOSERS
Fungi & Bacteria

Explain: Decomposers’ Role in the Food Chain

Decomposers are chiefly the micro-organisms such as bacteria and fungi in the form of yeasts and moulds that break down bodies of dead organisms. An example is if any of the consumers shown die, the micro-organisms would break down their bodies into smaller matter for the producers to absorb.

Explain how this provides necessary nutrients for the producers which helps them grow and reproduce, thus causing a chain reaction which helps the consumer chain.

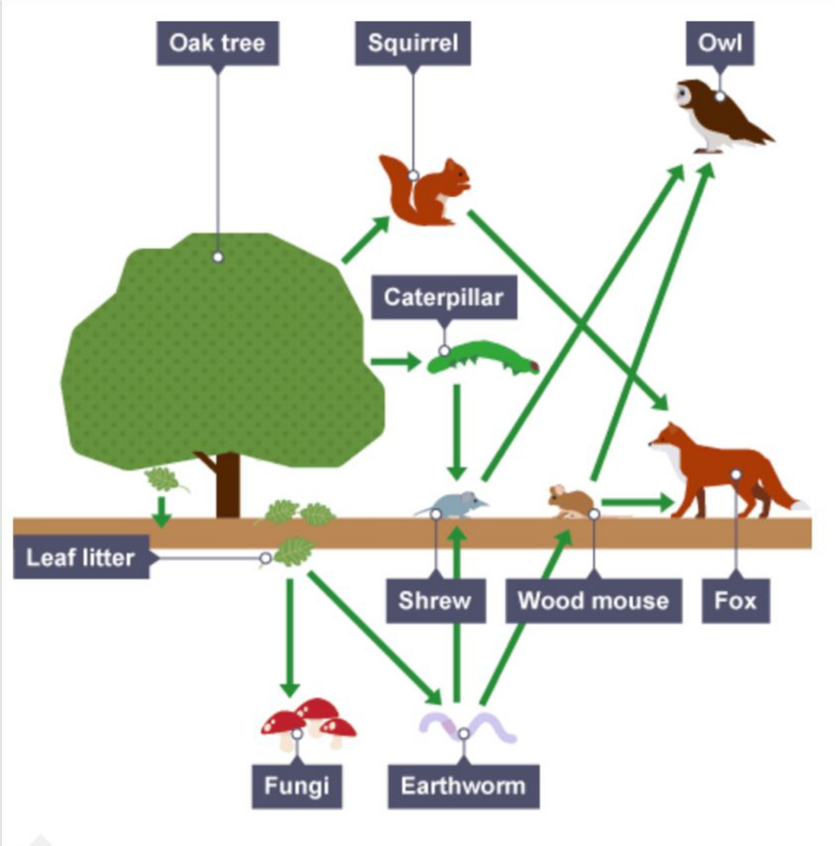


Decomposers’ Role in the Food Web

Explain: What is a Food Web?

A food web is a network of interconnected food chains. It shows the energy flow through part an ecosystems.

An ecosystem is a community of animals, plants and micro-organisms, together with the habitat where they live.

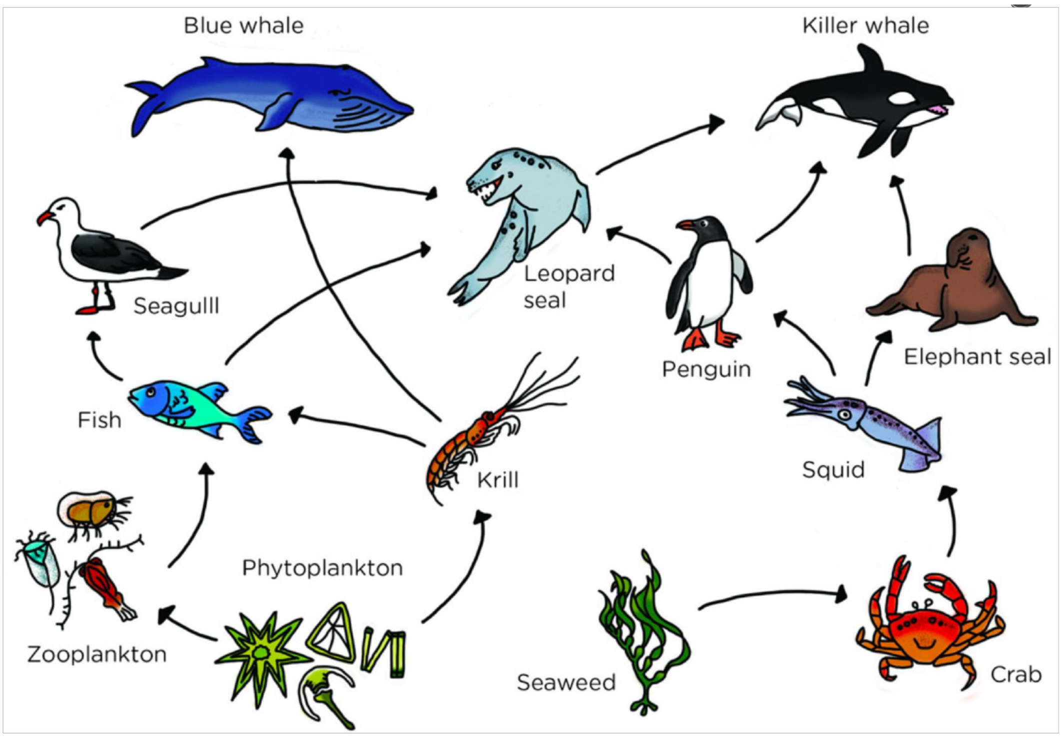


A woodland food web

Activity: Food Web

Students are to use the given food web on the slides to create the following food chains:

- Create a food chain with a length of three
- Create a different food chain with a length of four
- Create the longest food chain possible in this web



Solutions:
(Food chains must start with a producer):

- Any acceptable food chain of length 3 from the food web
- Any acceptable food chain of length 4 from the food web
- Longest food chain possible: phytoplankton, zooplankton, fish, seagull, leopard seal, killer whale. Six in total.

Explain: Energy Transfer

Energy is transferred along food chains from one trophic level to the next. However, the amount of available energy decreases form one trophic level to the next.

Energy Loss:
In a food chain only around 10% of the energy in a trophic level is passed on to the next trophic level. The rest of the energy passes out of the food chain in number of ways:

- It is lost as hear energy
- It is lost through use in life processes such as kinetic energy in movement
- Lost through faeces and remains which are passed to decomposers

Activity: Energy Transfer

Only around 10% of the energy in a trophic level is passed onto the next trophic level. The rest is lost through life processes such as movement, heat and waste.
Lets say we have a producer (mushrooms) which has a total energy of 5000kJ (kilojoules). Students should work out how much energy is passed on to the next trophic level if the mushrooms are eaten. They can work in pairs.

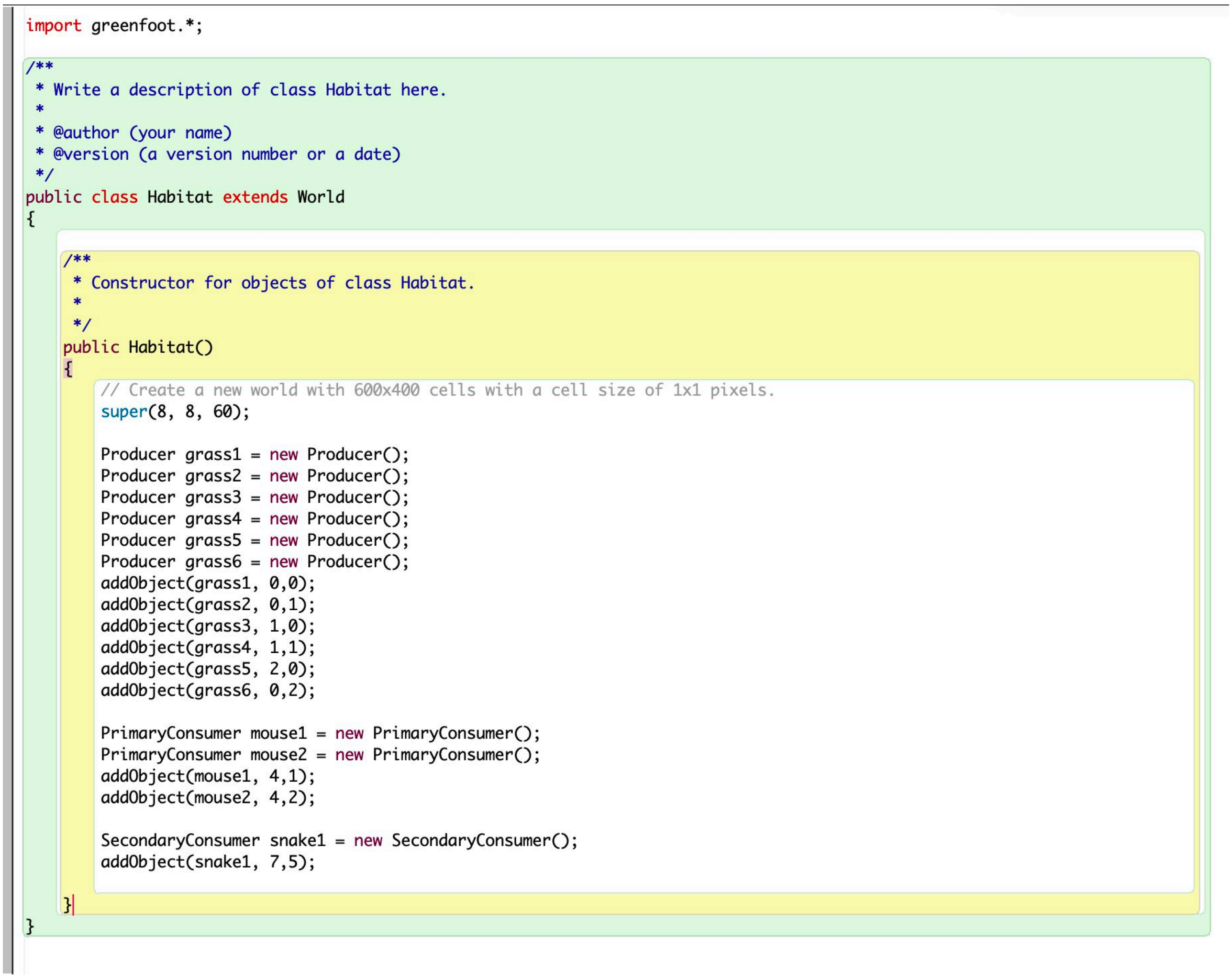
Answer:

The total energy transferred to next level is (10% of 5000)
5000 / 100 * 10 = 500kJ

Activity: Greenfoot Ecosystem

Follow the slides from the presentation to finish off the ecosystem simulation in Greenfoot

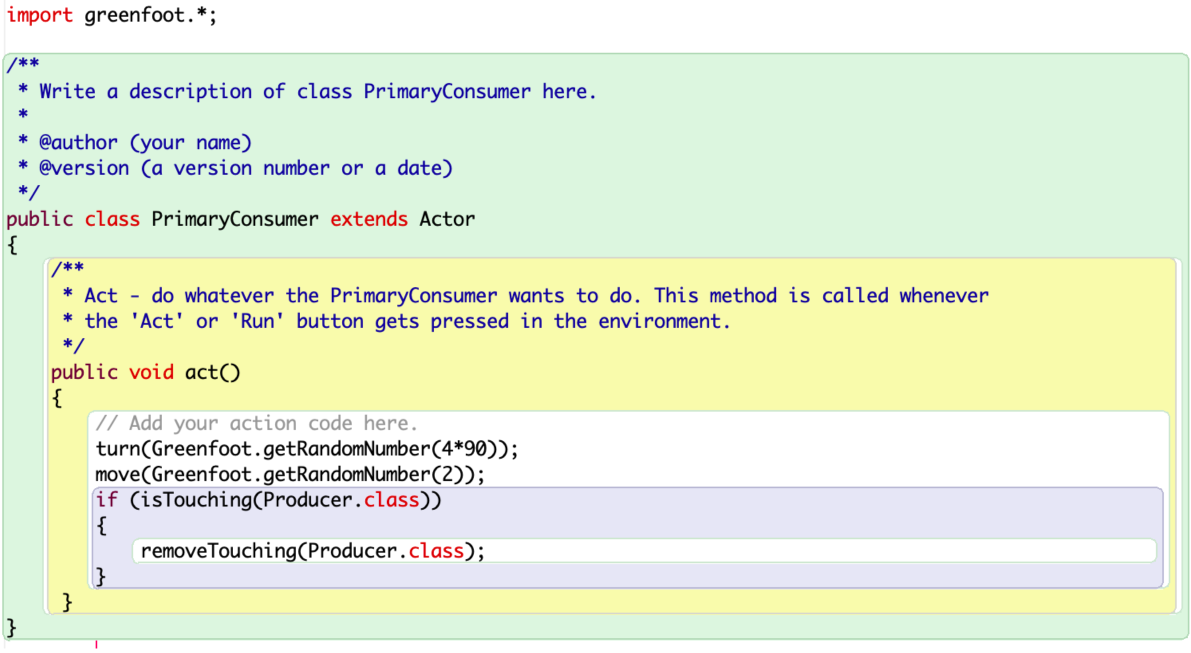
Explain: Full Habitat Code



```java
import greenfoot.*;

/**
 * Write a description of class Habitat here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Habitat extends World
{

    /**
     * Constructor for objects of class Habitat.
     *
     */
    public Habitat()
    {
        // Create a new world with 600x400 cells with a cell size of 1x1 pixels.
        super(8, 8, 60);

        Producer grass1 = new Producer();
        Producer grass2 = new Producer();
        Producer grass3 = new Producer();
        Producer grass4 = new Producer();
        Producer grass5 = new Producer();
        Producer grass6 = new Producer();
        addObject(grass1, 0,0);
        addObject(grass2, 0,1);
        addObject(grass3, 1,0);
        addObject(grass4, 1,1);
        addObject(grass5, 2,0);
        addObject(grass6, 0,2);

        PrimaryConsumer mouse1 = new PrimaryConsumer();
        PrimaryConsumer mouse2 = new PrimaryConsumer();
        addObject(mouse1, 4,1);
        addObject(mouse2, 4,2);

        SecondaryConsumer snake1 = new SecondaryConsumer();
        addObject(snake1, 7,5);

    }
}
```

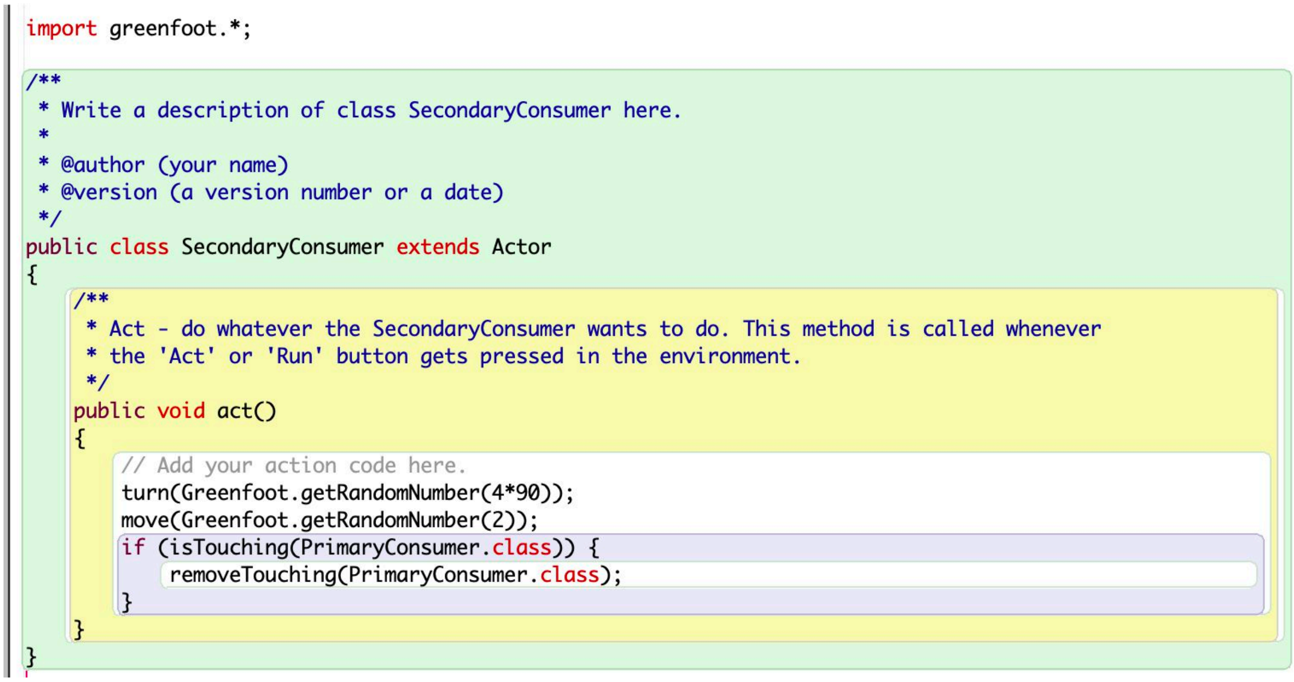
Explain: Full PrimaryConsumer Code

Note: we do not code in the Producer subclass, hence there is no screenshot for it.



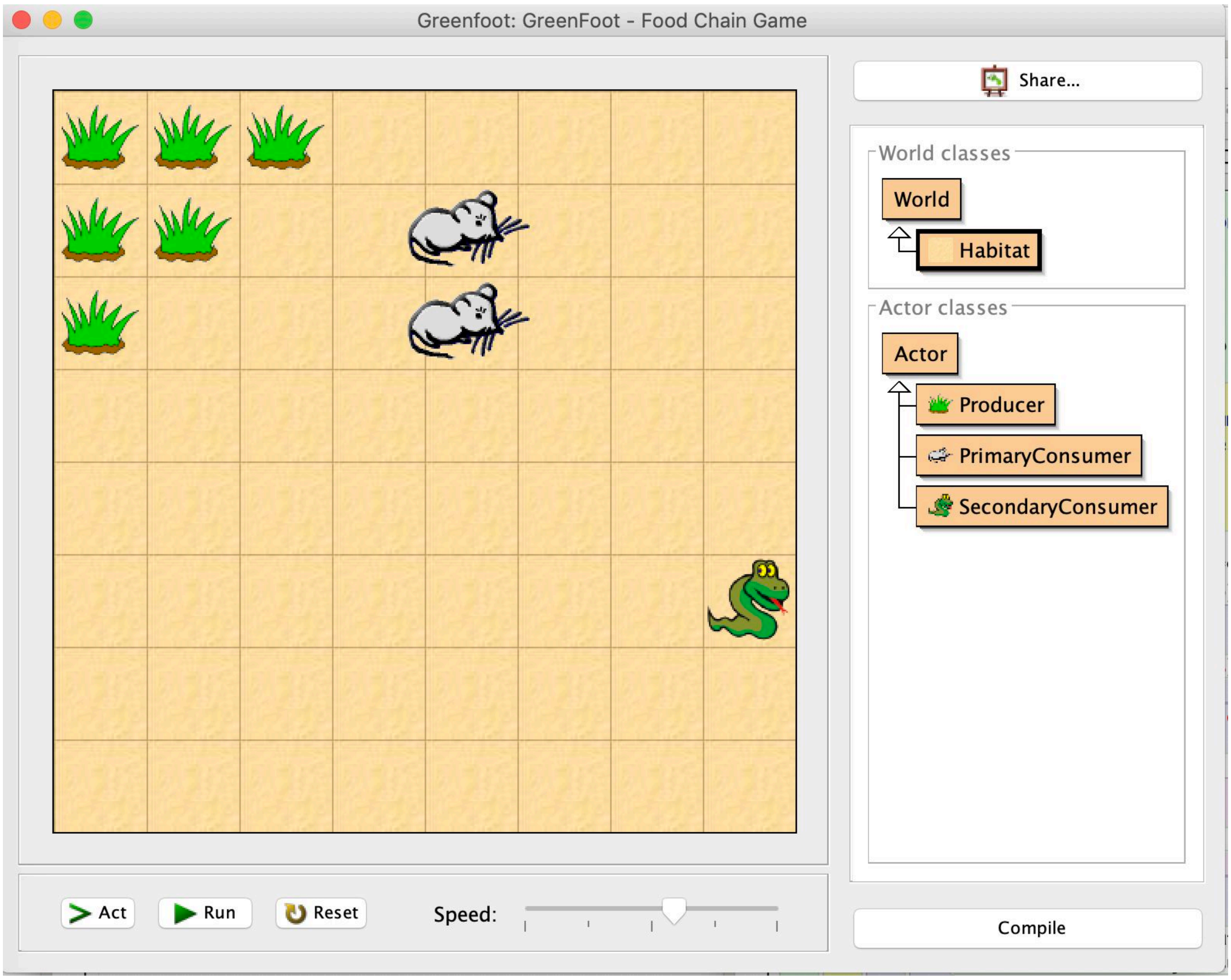
```
import greenfoot.*;

/**
 * Write a description of class PrimaryConsumer here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class PrimaryConsumer extends Actor
{
    /**
     * Act - do whatever the PrimaryConsumer wants to do. This method is called whenever
     * the 'Act' or 'Run' button gets pressed in the environment.
     */
    public void act()
    {
        // Add your action code here.
        turn(Greenfoot.getRandomNumber(4*90));
        move(Greenfoot.getRandomNumber(2));
        if (isTouching(Producer.class))
        {
            removeTouching(Producer.class);
        }
    }
}
```

Explain: Full SecondaryConsumer Code



```
import greenfoot.*;

/**
 * Write a description of class SecondaryConsumer here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class SecondaryConsumer extends Actor
{
    /**
     * Act - do whatever the SecondaryConsumer wants to do. This method is called whenever
     * the 'Act' or 'Run' button gets pressed in the environment.
     */
    public void act()
    {
        // Add your action code here.
        turn(Greenfoot.getRandomNumber(4*90));
        move(Greenfoot.getRandomNumber(2));
        if (isTouching(PrimaryConsumer.class)) {
            removeTouching(PrimaryConsumer.class);
        }
    }
}
```

Explain: End Game View

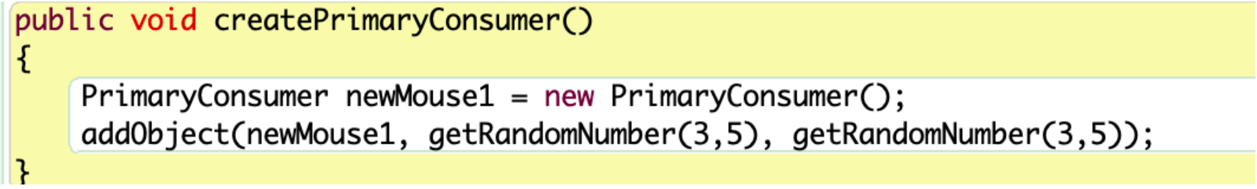


Greenfoot: GreenFoot - Food Chain Game

Explain: Object Interaction

Students can extend their game by adding interactions between objects. That is to say if objects of the same class touch, they can produce offspring. To do this, students need not understand the code but they should understand the concepts it brings.

Students will first need to create this method in their Habitat subclass:



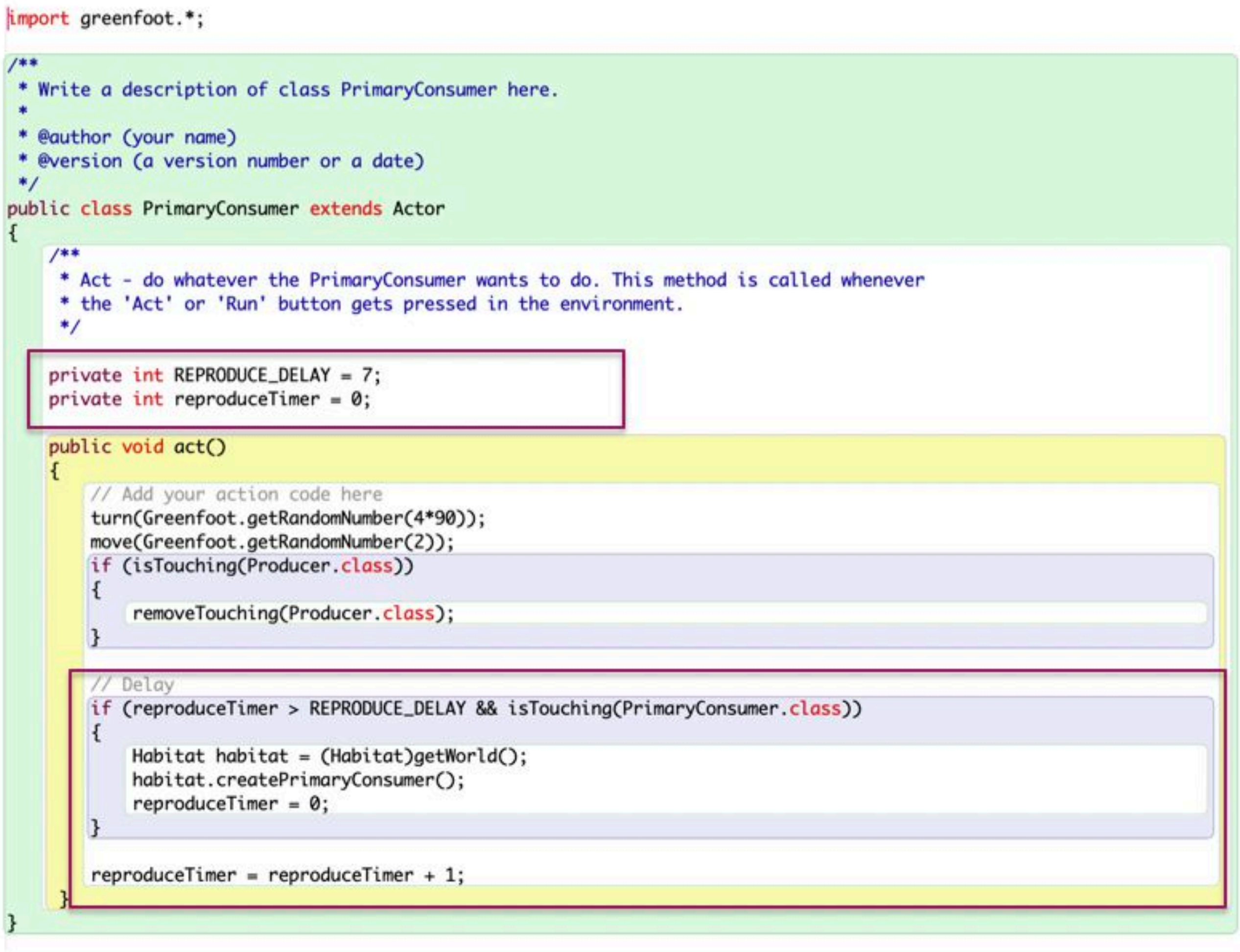
```
public void createPrimaryConsumer()
{
    PrimaryConsumer newMouse1 = new PrimaryConsumer();
    addObject(newMouse1, getRandomNumber(3,5), getRandomNumber(3,5));
}
```

This method when called will create a new PrimaryConsumer object and add it to the world in a random location between 3-4 (x, y) coordinates.

For this you need to make yourself familiar with the Greenfoot method 'getRandomNumber'.

Explain: Object Interaction (Continued)

Next, students need to code in the subclass which will perform the interactions, in this case the PrimaryConsumer.



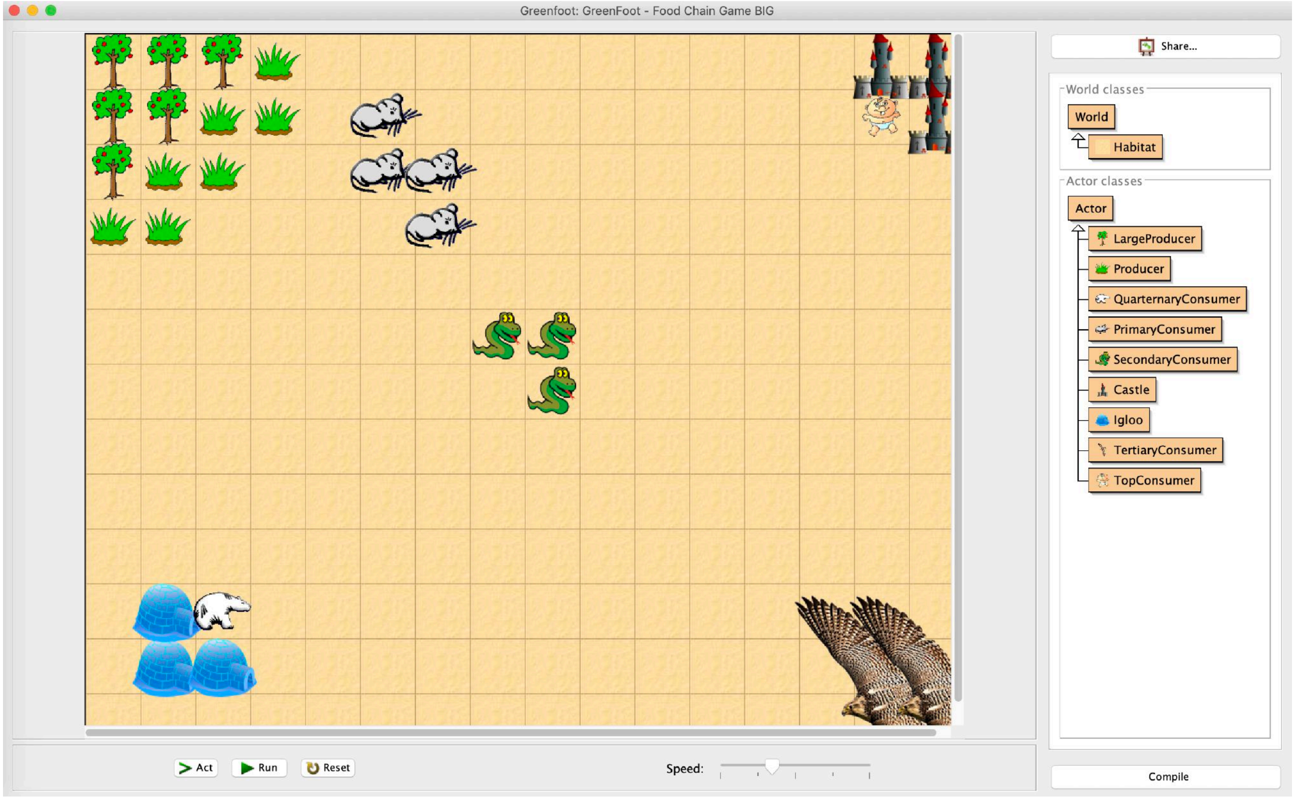
```
import greenfoot.*;

/**
 * Write a description of class PrimaryConsumer here.
 * 
 * @author (your name)
 * @version (a version number or a date)
 */
public class PrimaryConsumer extends Actor
{
    /**
     * Act - do whatever the PrimaryConsumer wants to do. This method is called whenever
     * the 'Act' or 'Run' button gets pressed in the environment.
     */

    private int REPRODUCE_DELAY = 7;
    private int reproduceTimer = 0;
    
    public void act()
    {
        // Add your action code here
        turn(Greenfoot.getRandomNumber(4*90));
        move(Greenfoot.getRandomNumber(2));
        if (isTouching(Producer.class))
        {
            removeTouching(Producer.class);
        }

        // Delay
        if (reproduceTimer > REPRODUCE_DELAY && isTouching(PrimaryConsumer.class))
        {
            Habitat habitat = (Habitat)getWorld();
            habitat.createPrimaryConsumer();
            reproduceTimer = 0;
        }

        reproduceTimer = reproduceTimer + 1;
    }
}
```

Students only need to copy the code inside the purple boxes. Explain that they do not need to fully understand the code but they should understand that it allows the objects to interact producing offsprings by the use of the 'createPrimaryConsumer' method we made in the Habitat subclass.

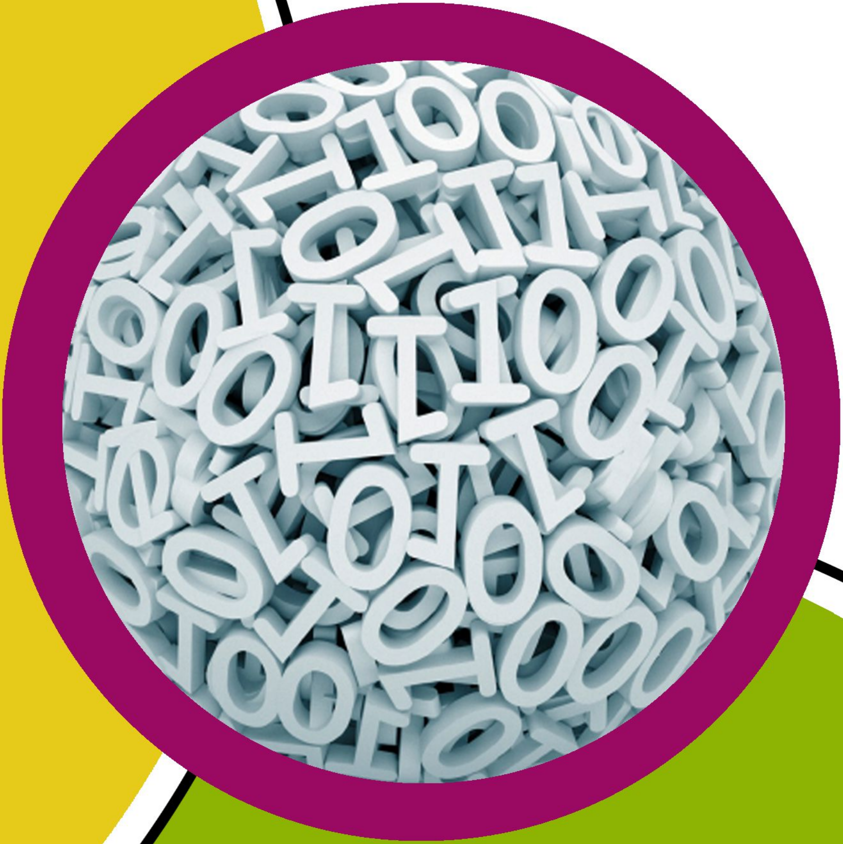
Once added these code sections will produce a game that increases the PrimaryConsumer population. Students can then put the same code snippets in the other classes such as SecondayConsumer etc. A game example with such interactions for various consumer subclasses can be found in this session pack.

Explain: Object Interaction (Continued)

Students can extend their game to create a much larger habitat containing different organisms.



Note: they need only to apply all that they have learned so far to create a much larger game.



technocamps

@Technocamps

Find us on Facebook