Object Attack

Educator's Guide

Overview

CS Hands-On is a 501(c)(3) nonprofit teaching computational thinking skills through technology-free lessons and activities. This curriculum is built to teach fundamental computer science concepts in an engaging, hands-on way. In this mission, students use public and private properties to play an action-packed game.

- ### Prerequisite Knowledge

  Student should have completed the Conditional Schedule and Object Oriented Fun activity, which introduces the concept of variables and object-oriented programming.

- ### Lesson Details

  At Decomposphere, students will learn to break problems down into smaller parts with Dot. Students will learn the importance of encapsulation with public and private properties to play an action-packed Grid Attack game.

  This lesson was developed for students ages 8 to 13, and can be modified for students of all skills and ages. This lesson takes roughly 30 minutes.

Learning Objectives

- ### Key Question

  How can we assign public and private properties to objects?

- ### Key Terms

  Encapsulation: A conditional statement that triggers a set of instructions when a certain condition is false.

- ### Curriculum Standards

  Students should be able to...

  - Explain why encapsulation is used (Decomposition)
  - Read, write, and interpret public and private properties (Literacy)
  - Use functions and variables to play a Grid Attack game (Creative Arts)

  View standards addressed here

Lesson Plan

- **Materials**
  - Object Attack worksheet (per student)
- **Setup**
  - Hand out a Object Attack worksheet to each student
  - Set up your classroom to form students in groups of 2

ANSWER KEY & LESSON ANNOTATIONS

Decomposphere
Mission 3

Name: _____ Date: _____

Object Attack

Outstanding Objects!

Dot is so excited to see you back at Decomposphere! Today, you'll learn more about objects and how they can keep their properties out of sight. Let's jump right in!

Who let the dogs out?

In object-oriented programming, **encapsulation** allows for the variables and functions of an object to be encapsulated or kept private from the interference of other objects. Each object can have private and public properties.

What's the difference between private and public?

While private properties (variables and functions) cannot be changed or used by other objects, public properties can. Let's take a look at an example with Dot's pet dog, Rocky!

Rocky the Labrador

Variables (attributes Rocky has)		Functions (what Rocky does)	
private breed	**Why private?** We can't control Rocky's breed and age.	**private** bark	**Why private?** We can't control when Rocky barks or sleeps
private age		**private** sleep	
public owner	**Why public?** We can control who owns Rocky	**public** feed	**Why public?** We can control when to feed or play with Rocky
		public play	

CS Hands-On Object Attack 1

Reflect

If we modeled humans as objects, what would some of our private and public properties be?

Ex. Private: Age, name, birthday

Public: Hair color, pets

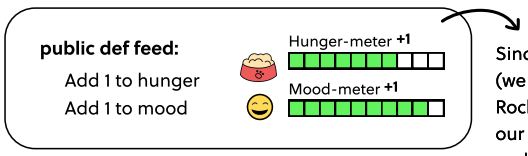
Decomposphere
Mission 3

Why do we use private properties?

In programming, it's essential to make properties like breed, mood, hunger, and energy level private to **protect other objects from changing them**.

How can we change private variables?

We can give our object certain **functions to change these private variables!**
For example, let's take a look at our public function **feed**:



Since our feed function is public (we can decide when to feed Rocky), we can indirectly change our private variables (hunger and mood).

However, some private variables can never be changed.
For example, we cannot change Rocky's breed as a Labrador.

CS Hands-On Object Attack 2

Educator Note

Through public functions, we can change private variables like hunger and mood.

Encourage your students to brainstorm other public functions to change the private variable mood of Rocky the Labrador. (Play, walk, pet)

Decomposphere
Mission 3

Grid Attack!

In this action-packed game, you will be creating your own **player object** with its **private/public variables and functions** to attack an opponent!

Materials

- 2 tokens for you and your friend (This will represent where you are on the board!)
- 1 die

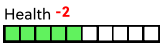
How to Play

- Define your player object's variables by creating a name, age, and health value. Each player starts the game with a health level of 5 (out of 5). Next to each variable and function, circle whether it is private or public! *(Think: Can these properties be directly changed or used by other objects?)*
- Take turns rolling the die with your friend. Use the **walk** function to move your token the number of steps you rolled in any direction(up, down, left, right). Keep track of your activity in your **activity list**.
  (Ex. If you rolled a 6, you can choose to walk 3 steps left and 3 steps up).

  If you are on , use the **eat** function to add 1 point to your health.

  If you are at the **same location** as your opponent, use the **attack** function to subtract 2 points from your opponent's health.

- Winning the game: You win the game when your opponent's health status reaches 0.

CS Hands-On Object Attack 3

Extension
If students finish early, have them change the Health variable to a higher/lower number to play another game. Print as many copies of the activity list and game board as needed.

My Player Object

Variables

private/public Name
(circle one)

private/public Age
(circle one)

private/public Health
(circle one)
5

Functions

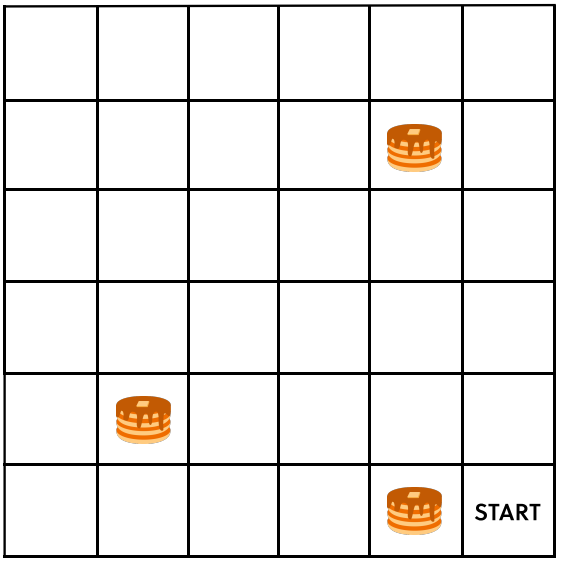
private/public def Walk:
(circle one)
Move the number of steps rolled

private/public def Attack:
(circle one)
Subtract 2 points from opponent's health

private/public def Eat:
(circle one)
Add 1 point to your own health

My Activity List

Function	Health	Function	Health
Ex. Walk left 3 steps	5		



Wrap up & reflect

Group students into pairs and have them discuss the following reflection questions. Afterwards, have students share their ideas as a class.

- Model an insect as an object with its own private/public variables and functions. What properties would it have?

  Ex. Ladybug
  Private variables: Number of legs = 6, Number of wings = 2, Color = Red
  Public variables: Location = Park
  Private functions: Fly, Crawl

- Why is it important to establish different properties as either public or private?

  Ex. When a property is public, we know that it can be changed directly by us or other objects. When the property is private, we know that it can either be changed through a public function or simply cannot be changed at all.