the PACK

Activity Guide

INTRODUCTION TO THE PACK

The Pack Game

The Pack is a digital game created by NYSCI and Design I/O and inspired by NYSCI's groundbreaking large scale interactive ecosystems exhibit, Connected Worlds. The game encourages learners to use Computational Thinking (CT) to solve problems they encounter in the futuristic world of Algos, where healthy ecosystems have faltered and where water and food are scarce. The player must get help from the creatures of Algos to help restore the environment.

Throughout the game, players must find food sources to attract their own Pack of Algos creatures. These creatures have various functions (like digging, moving, grabbing, and bumping) that can help the player collect seeds and bring water to dry areas. Creatures can be combined to form algorithms that perform complex tasks and overcome challenges. Players' use of these algorithms is the key to finding all of the seeds and restoring harmony to Algos!

The Pack Supplemental Activities Guiding Principles

The Pack game and supplemental activities are designed to support middle schoolers in engaging with Computational Thinking concepts and practices while reinforcing basic scientific reasoning. According to Jeanette Wing, Computer Science professor at Columbia University, Computational Thinking can be defined as "the thought processes involved in formulating a problem and expressing its solution in a way that a computer—human or machine—can effectively carry out." CT is quickly becoming an essential 21st century skill that children will need to become active participants in our growing digital age. Beyond just coding, the practices of CT habits of mind enable children to articulate a problem and logically break it down into parts, predict what might happen in the future, create a series of steps to solve a problem, explore cause and effect relationships, and analyze how their actions and the actions of others can impact a situation.

All these skills are especially critical in building learners' capacity to do STEM, to understand complex phenomena, and to reason about the workings of the natural and built environment.

The Pack game and supplemental activities help learners explore foundational concepts of CT. Students engage in reflective discourse about how they use CT to solve problems in the game, in the classroom, and in their everyday lives. The target concepts are:

Decomposition: Defining a problem and breaking that problem into smaller, more manageable parts.

Pattern Recognition and Generalization: Using prior experiences to find patterns within a problem and remixing and reusing prior solutions to solve novel problems.

Pattern Recognition and Generalization: Using prior experiences to find patterns within a problem and remixing and reusing prior solutions to solve novel problems.

Abstraction: Identifying relevant information and removing unnecessary details.

Algorithms: Developing a sequence of rules or instructions to perform a task or solve a problem.

Debugging: Evaluating your solution and iterating on your design as needed.

Helping learners explore these big ideas in an engaging open-world digital game affords a unique opportunity to introduce CT as a powerful approach that learners can use to systematically solve problems and make decisions in class and in their lives. Open-world games are particularly well suited for supporting this kind of thinking; they allow children to test their abilities, experiment, invent, look at problems from multiple points of view and, most importantly, fail and learn from their mistakes in a self-motivating environment where they can try alternate solutions. This active approach to scientific problem solving directly supports state and national science and computer science standards.

INTRODUCTION TO THE PACK SUPPLEMENTAL ACTIVITIES

The supplemental activities encourage players to reflect on their gameplay through the lens of Computational Thinking and to explore how solving problems in the game connects to the ways they can solve problems in class and beyond. The supplemental activities can be implemented in a classroom, as part of an after-school program, or even as an activity for families to do together at home.

Getting Started with The Pack

Exploring Algos	*Game Journal* *Part 1* The facilitator will introduce the game and walk players through Level 1 as a whole group. *Part 2* The facilitator will debrief the gameplay experience and work with players to create a crowdsourced list of Tips and Tricks to encourage reflection about problem solving in the game. Players will document their approaches to solving problems in a Game Journal.
Learning Objective(s)	To get acquainted with playing *the Pack.*

Exploring Computational Thinking (CT)

Activity	Target CT Concepts
Three of a Kind Players will identify three methods for collecting food.	Overall introduction to CT and review of key vocabulary
Making It Work Players will create algorithms for several scenarios to practice a CT approach to solving problems.	Overall exploration of CT
Stay Hydrated Players will find a dry zone and extend the nearest body of water enough to cross the dry zone.	Algorithms Debugging
Who Put That There? Players will use Pattern Recognition to Decompose a novel problem.	Generalization Pattern Recognition Decomposition
Make and Break It Players will design an algorithm for a problem of their choice, then break that algorithm before passing it along to a partner to debug it.	Debugging
Seed Trio Players will design a solution for gathering three seeds in a given scenario.	Decomposition Abstraction

INTRODUCTION TO THE PACK SUPPLEMENTAL ACTIVITIES

Culminating Activities

Activity	Learning Objective(s)
How'd He Do It? Players will analyze the pros and cons of various algorithms in a specific problem context.	To analyze the viability of multiple solutions to a problem given the context.
Map Gaps Players will identify problems or shortcomings of the map feature in *the Pack* and suggest a redesign to be more useful to players.	To apply CT strategies in a design-based challenge.
Level Up Players will design the next level of the game.	To design a new level of *the Pack* in order to reflect on and synthesize experiences from gameplay.

Depending on how you are implementing the activities, you might choose to have learners complete the worksheets as homework, and then dedicate class time to sharing and reflective discussion. Activities are designed to take one or two class periods depending upon whether players are completing the worksheets in class or as homework. While you can select which activities to use with your learners, we encourage you to read through all the activities. In particular, the Game Journal and Three of a Kind activities provide a strong foundation for further exploration and gameplay.

Here are some important things to keep in mind:

Encourage learners to play as much of the game as possible. We recommend learners have as much experience playing the game as time will allow prior to completing the "Exploring Computational Thinking" lessons. It takes most learners two to four hours to play through all levels at least once. Having time to explore the game environment provides learners with more opportunities to use CT, leading to richer peer-to-peer reflective discussion and sharing of problem-solving approaches. Though players may progress through the game at different rates, the game is structured to provide opportunities to engage in CT concepts and practices at every level.

Highlight the decision-making and problem-solving aspects of the game. Introducing the game as a way of solving problems and thinking systematically provides a good basis for learners to reflect on their thinking throughout the game. Introduce the game as an opportunity for learners to think like scientists who figure out how to solve problems.

Facilitate reflective discussions. The supplemental activities are structured reflection touchpoints that enable the facilitator to lead discussions about what learners did, how they went about approaching problems, and to make connections to how this kind of systematic thinking is critical to solving problems in science, computer science and everyday life.

INTRODUCTION TO THE PACK SUPPLEMENTAL ACTIVITIES

Have limited time? Try	With additional time, add in	With additional time, add in
Game Journal Parts 1 and 2 Three of a Kind Level Up	Seed Trio Make and Break It Who Put That There How'd He Do It?	Making It Work Stay Hydrated Map Gaps

The Pack Facilitator Guides

Each activity has an accompanying Facilitator Guide designed to help educators lead learners through reflective conversations to make connections between their gameplay and the ways that they approach problems in other contexts. The guide for each activity follows the same basic structure.

FACILITATOR GUIDE STRUCTURE

Minimum Game Level: This section describes the minimum level a learner should have experienced prior to doing the activity.

Purpose: This section describes what learners will accomplish in this activity and how they will do it.

Background Information for Facilitators: This section describes the central concept(s) for the lesson as well as broader implications.

Activity In Action: This section provides tips for introducing the activity to learners and talking points for transitioning from one activity to the next.

Reflection Questions: The questions in this section build from the questions included on the student worksheet. This includes questions that help learners take the central concepts they explored in the game and connect them to the types of problems they solve in the classroom and their everyday lives.

What To Look For: This section provides ideas for how to recognize learners' understanding and use of the target concepts.

Sample Solutions: This section provides examples of viable responses.

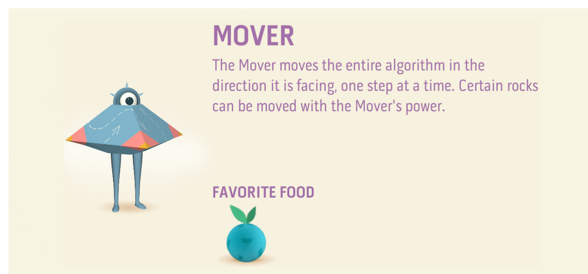
Differentiation: This section provides ideas for scaffolding the activity and providing additional challenges for all kinds of learners.

While the guides provide a basic structure for how an educator might implement each challenge, there is also flexibility for educators to add their own questions or additional activities.

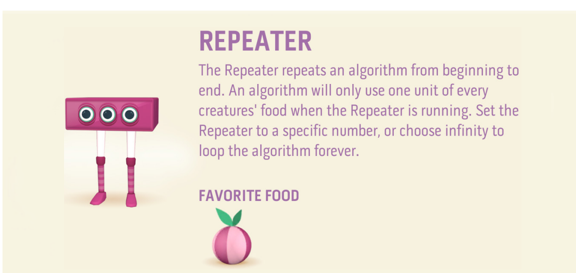
QUICK GUIDE

Welcome to the world of Algos! You must collect seeds to progress through each level of the game, and you will need to use your Pack of creatures to solve the challenges that you encounter along the way. No two game maps are the same. A more thorough Game Guide is available for your reference, but playing the game is the best way to learn about the various features, functions, and ways to navigate the game world.

The Digger is the first creature you will find to join your Pack. The Digger is able to dig waterways, dig up buried fruit, and dig for seeds.



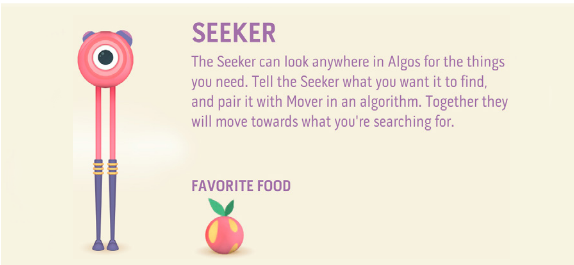
The Mover is the second creature you will find. The Mover can be used to push rocks or move an algorithm from one place to another in the game.



The Repeater is the third creature you will find. The Repeater can be used with other creatures in order to repeat their functions without using more fruits.

The Bumper is the fourth creature you will find. The Bumper can be used to smash rocks and knock down fruits from trees.

QUICK GUIDE



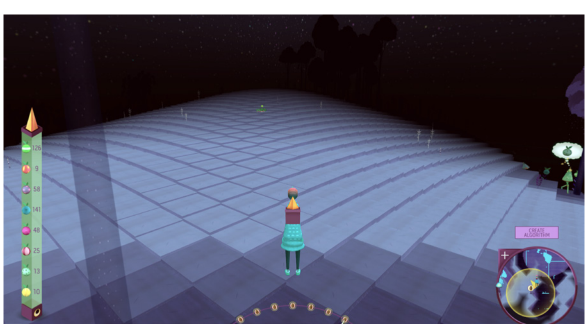
The Seeker is the fifth creature you will find. The Seeker can be used to find different fruits, seeds, and water throughout the world of Algos.

The Grabber is the sixth creature you will find. The Grabber can be used to collect different types of fruits, seeds, and water.

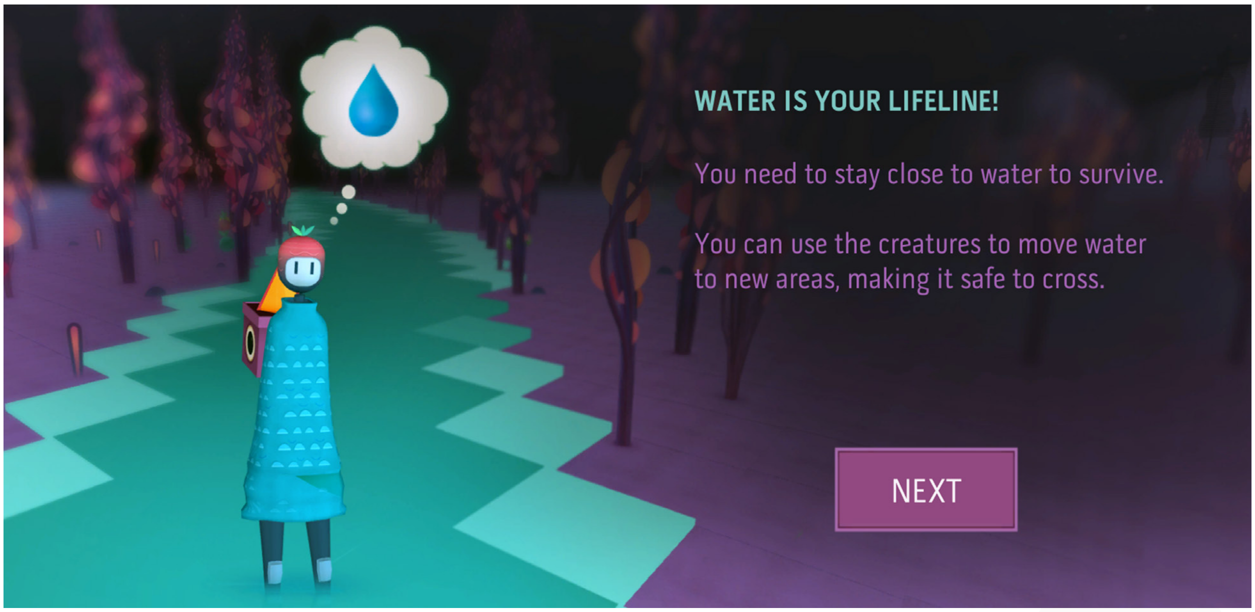
The Builder is the seventh creature you will find. The Builder can be used to reach fruits high up in trees, build mountains, and reach seeds high in the air.

The Dropper is the final creature you will find. The Dropper can be used to release water or fruits that your Avatar has collected throughout Algos.

Many of the problems you will encounter in Algos stem from dealing with dry zones. You cannot walk through dry zones for more than a few seconds at a time. Ignoring the on-screen prompt to find water when traversing a dry zone will cause your motion to slow and the field of view to become black. Eventually you will be forced to return to the edge of the dry zone (see images below).

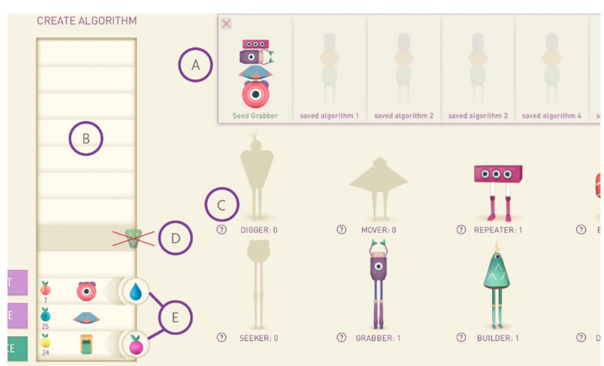


QUICK GUIDE



As you navigate the world of Algos and combine creatures to create algorithms, it will be important for you to document your work. While the Game Journal and worksheets encourage you to record your ideas and reflections, screenshots are also a great option for helping you share your gameplay with your teacher and your classmates. These can include screenshots of algorithms in action as well as screenshots of the "Create Algorithm" screen.

Algorithm in Action



A) Saved algorithms B) Algorithm Editor C) Creature information button D) Removing a creature from the algorithm E) Player selected item creature interacts with

Facilitator Guide
Exploring Algos Game Journal

Purpose

Using the Game Journal, students will document their gameplay, including their approaches to solving problems in the game, the goals and rules they discover, features they use and obstacles they encounter. The journal is designed to help students reflect on a central idea of Computational Thinking (CT)—that there are often different ways of solving the same problem. Use the journals as a tool for surfacing problems students encounter in the game and to compare and share different approaches to solving these problems. This can result in a class-generated Tips and Tricks guide that students can continue to add to throughout the lessons.

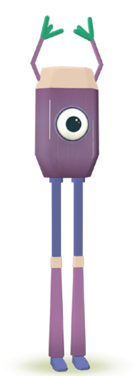
Background Information for Facilitators

In Part 1, you will walk your students through the first level of the game. While the in-game tutorial provides an overview of how to perform basic actions, it does not fully support the player in exploring how to overcome new obstacles that arise, describe the functions of Pack creatures, nor describe the narrative or the "why" of the game; these are things that must be discovered through gameplay. Playing collaboratively will reveal more about the game than having students play individually.

Following Part 1, students will play the game on their own and document their gameplay using the Game Journal. This gameplay can be done as a homework assignment or in class, but students will need at least an hour of play time.

In Part 2 of the lesson, students will use their Game Journals to reflect on their experience, share what they have discovered, and generate a class Tips and Tricks Guide that you can continue to reference and add to in future lessons.

On average, it takes students two to four hours to complete the game. While some students may complete the game as part of this lesson, it will be important for you to provide additional gameplay time before moving on to the next activities.



The Pack is designed to be a judgment-free zone where taking risks and trying things out is part of learning how to think computationally.

Part 1: Introducing the Game

1. Use a projector or smartboard to load a new game and demonstrate how to complete the in-game tutorial.

2. When the tutorial is complete, continue playing until you have collected three seeds to complete the first level. This initial walk-through of the first level of *The Pack* will provide a common entry point for non-gamers and seasoned gamers alike.

a. If you feel your students require more scaffolding, describe what you are doing as you play.

b. If you feel your students require less scaffolding, have them tell you what to do and explain why as you play.

3. After you have played through Level 1 as a class, ask students to assist you in making an entry in your Game Journal to model the type of information they should be documenting on their own. This should include meeting new characters, discovering new features, and combining the creatures to design different solutions to the problems they will encounter in-world. Be sure to review the whole journal so students are familiar with each section. Explain that as a class, you will be creating a "Tips and Tricks" guide, so they should be sure to record anything they think might help another player.

4. Ask students "What do you think is the goal of this game?" Share that one of the goals is to collect enough seeds at each level to unlock new habitats in Algos. Your students will know they have unlocked all the habitats when the dormant friend located in the center of the map awakens.

Student Free Play

It is time for students to play on their own. You should expect students to vary in how quickly they progress through the game but for this free play session, encourage them to play at least through Level 3. This can be assigned as homework or completed in class.

Facilitation Tips

- Make sure students know how to access the game on the device they are using, especially if they will be expected to complete the Game Journal as homework.
- Have students complete a journal entry at least three times during their gameplay: once in the beginning when they are learning to play, again after they have met more creatures while playing on their own, and once more at the end of their gameplay.
- Note that students will progress through the game at different rates; some might get all the way to the end in a short time while others might still be on early levels of the game even after some time. That is OK. Sharing tips and tricks is possible at every level of the game and the journal is a tool for all of the students to reflect on the problem-solving that they are doing.
- Encourage students to share mistakes they have made and how they addressed them as well as their successful tips for solving problems. It is often in the mistakes where the most interesting discussions can happen.
- You are welcome to change the way the journal is structured to make it more or less scaffolded. For example, you could simplify it by just asking for general observations for each level of the game. The key is for students to focus on how they went about solving problems they thought were important in the game.

Part 2: Game Journal Reflection

After students have had time to play, either in class or as homework, come back together as a whole class and use the questions in the Game Journal to help students reflect on what they have discovered and how they have played so far. In addition to the journals, you can add additional questions, like the following:

a. What differences do you notice between levels?

b. Based on what you have seen on the "Create an Algorithm" screen or what you have already heard about this word, what do you think an Algorithm is?

c. What questions do you still have about the game?

As students share out, document their ideas on chart paper or digitally to create a class set of "Tips and Tricks" for future reference. Not every student will have discovered every aspect of the game. This activity works best when students discover game features and mechanisms on their own and share out through conversation and reflection.

Facilitation Tips

- Refer to the Game Guide to help you answer any student questions about the game, including creature names and functions. We do not recommend that you give the Game Guide to students, as it may limit their exploration and ability to discover things on their own.
- This activity might give you an indication of which students have a strong understanding of how to play the game and which might need more scaffolding. Use your observations of students' gameplay and completed journal entries to inform how you scaffold future lessons.

The Pack
Game Journal

People can play *the Pack* and tackle the same problems (like finding fruit or seeds) in many different ways. This Game Journal has some questions to get you started in recording what you discover and the steps you took to solve problems, but you can add other things like useful creatures, tips for making things happen in the game, and timesaving tricks you came up with. Use the Game Journal at least three times throughout your gameplay: make an entry in the beginning when you are just learning to play, sometime later after you have collected more seeds or creatures, and at the end of your gameplay. We all might play the game in different ways, and we will work together to create a class Tips and Tricks guide using what we learn from all our Game Journals.

Level: ______________________________________

What was a tricky problem you were trying to solve on the level you just played?
Any problem you come across in the game can have a tricky aspect to it—so sharing what you saw is helpful!

What did you need to pay attention to about the situation?
Some things matter to solving the problem and some do not make any difference. What parts mattered in this problem?

What are some of the different parts of the problem you were solving?
Describe the different things you had to think about to solve it. What things had to get done, and in what order?

Were there tools or things you have used before that helped here?
What did you use? Pack creatures? Smaller algorithms you used before combined with new creatures?

What worked? Which parts of your solution did what you expected?

What did not work the way you wanted it to? Was there anything you did not expect? What would you change to fix this issue?

Use this space for any other things you want to record that might be helpful to yourself or other players.

Level: ______________________________

What was a tricky problem you were trying to solve on the level you just played?
Any problem you come across in the game can have a tricky aspect to it—so sharing what you saw is helpful!

What did you need to pay attention to about the situation?
Some things matter to solving the problem and some do not make any difference. What parts mattered in this problem?

What are some of the different parts of the problem you were solving?
Describe the different things you had to think about to solve it. What things had to get done, and in what order?

Were there tools or things you have used before that helped here?
What did you use? Pack creatures? Smaller algorithms you used before combined with new creatures?

What worked? Which parts of your solution did what you expected?

What did not work the way you wanted it to? Was there anything you did not expect? What would you change to fix this issue?

Use this space for any other things you want to record that might be helpful to yourself or other players.

Level: ______________________________

What was a tricky problem you were trying to solve on the level you just played?
Any problem you come across in the game can have a tricky aspect to it—so sharing what you saw is helpful!

What did you need to pay attention to about the situation?
Some things matter to solving the problem and some do not make any difference. What parts mattered in this problem?

What are some of the different parts of the problem you were solving?
Describe the different things you had to think about to solve it. What things had to get done, and in what order?

Were there tools or things you have used before that helped here?
What did you use? Pack creatures? Smaller algorithms you used before combined with new creatures?

What worked? Which parts of your solution did what you expected?

What did not work the way you wanted it to? Was there anything you did not expect? What would you change to fix this issue?

Use this space for any other things you want to record that might be helpful to yourself or other players.

Facilitator Guide
Three of a Kind

Minimum Game Level

This activity is intended for players that have played Level 2 and up. Level 1 is a continuation of the tutorial. In Level 2 students will begin to get greater variation in gameplay experiences.

Purpose

Students will look for three ways to get food in order to recognize that there can be multiple ways to solve a problem, and to reflect on the ways in which they use Computational Thinking (CT) to arrive at their solutions.

Background Information for Facilitators

This activity challenges students to gather food. If you have completed the first level together with your students and they have spent some time playing independently, your students will already have some great ideas about how to do this. This activity does not require all students to be on the same level of the game. In fact, a variety of perspectives and contexts are conducive to creating multiple solutions to the same problem.

This activity helps to introduce your students to CT concepts and provides a context for highlighting where these concepts come up in students' gameplay. The most important aspects of debriefing this activity will be comparing and contrasting solutions and identifying ways that students used CT to complete the activity. This will be a good opportunity for discussions about how there can be different ways to accomplish similar goals. The more experiences your students have using CT strategies, the easier it will be for them to generalize them to other subjects in the classroom and beyond.

Activity in Action

The gameplay portion of this activity should be prefaced with a conversation introducing Computational Thinking. Your students will likely be familiar with the concepts of CT but may not know the vocabulary.

1. Ask your students to think of a situation where they had to solve a complex problem. This could be a problem from class, from their lives, or you could give students a challenge to do in the moment (e.g., create a paper airplane that can fly ten feet). Have students turn and talk with a partner or invite a few volunteers to share their examples with the whole class.

2. Go over the “Introduction to Computational Thinking” vocabulary sheet with your students. You might wish to display chart paper with each of the CT components in your classroom so students can revisit them over time.

3. As you go over the target concepts, ask your students to start to identify where each element of CT might come up in the context of the problem they just discussed.

4. Give students time to record their ideas on the vocabulary sheet, either individually or in small groups.

5. After giving students time to work, elicit a few examples to review as a class. It is OK if students are not able to identify every component of CT for every example.

6. Wrap up by reviewing the Three of a Kind worksheet to ensure that students understand the activity.

7. Have your students complete the worksheet either as homework or in class.

Reflection Questions

After students have had time to complete the worksheet, engage in a debrief discussion. Use these additional questions to help students further reflect on their gameplay and ways that they think computationally. Encourage students to explain their responses and solutions to this activity. They might not use newly learned vocabulary right away, but if you point it out in context, they will create their own connections between the concepts and their respective terms.

- How did you solve this challenge? Be sure to share what types of food you gathered.
- Did anyone gather the same types of food in a different way?
- After learning how others completed the activity, which way do you think was the best way to get food? Why?
- Do you think your classmates would agree with you on the way you think is best?
- Is there anything you tried that did not work?
- Is there something you did for getting one type of food that helped for getting a different type of food? What did you do?
- What are some examples of Decomposition, Pattern Recognition and Generalization, Abstraction, Algorithms, or Debugging in this activity?
- Why might there be different responses about which solution players preferred in this activity?
- Was there a "wrong" way to solve the problem? Why or why not?
- Did anyone notice any patterns in how you were approaching this task? (Students may not notice patterns at this point, but it may be useful to ask this question to build up students' capacity to recognize patterns over the course of the challenges.)

Highlight the idea that there are often multiple solutions to a problem and encourage students to provide their own personal examples of different solutions to the same problem.

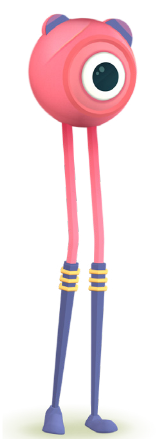
What To Look For

As you go over the responses to the questions on the student worksheet, there are several things to look for which will help you highlight ways that your students already think computationally.

The most obvious CT concept in this activity may be Algorithms since that is what students are asked to create. However, having multiple solutions to similar problems also provides the opportunity to compare different ways students have understood and broken down a problem (Decomposition) and used prior knowledge to help them determine which factors to consider important and which factors are inconsequential (Pattern Recognition and Abstraction). The following are some potential conversation points that can be connected to CT concepts.

For Decomposition, listen for students listing what they might need to get different types of food. For example, a student may mention that if food is in a tree, they will need a Bumper. This is evidence that they are considering the smaller problems or tasks that are required to ultimately get food in three different ways.

For Pattern Recognition and Generalization, listen for when students describe something that they already know from previous experiences playing *the Pack*. For example, a student might describe digging up food from the ground to get a bundle of five pieces of food instead of or in addition to bumping that same type of food from a tree. When students recognize a pattern and apply their prior knowledge, they are generalizing a solution to a new context.



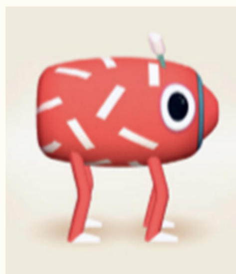
For Abstraction, listen for examples of students listing what must be done to complete the activity. This will likely be very similar to Decomposition in this activity. The clearest distinction will be if they also describe the factors they considered unimportant for completing the task. This will likely come up when they compare and contrast different solutions.

For Debugging, listen for students analyzing and iterating on unsuccessful algorithms or solutions.

Sample Solutions

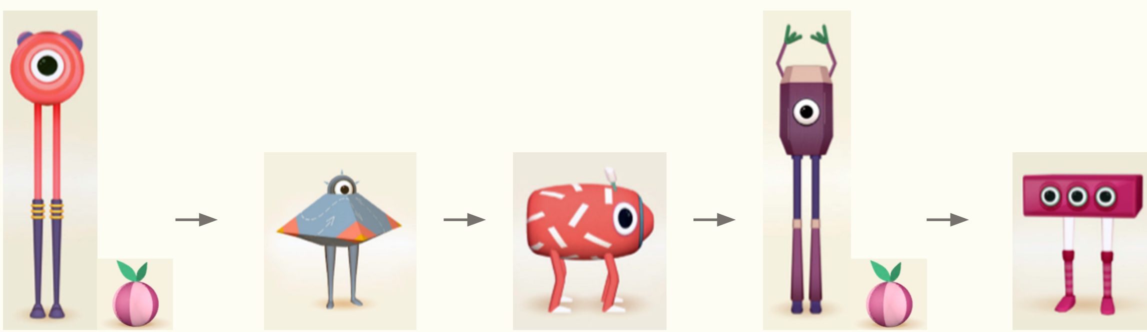
Students might describe some of the following examples for gathering food:

- *Walk over food located on the ground.*
- *If the target food is in a tree, then use the algorithm:*



→ *Walk over food.*

- *If the target food is far away, then use the algorithm:*



Differentiation

For students requiring an added challenge you can provide an extra constraint by asking students to minimize time, food or effort expended on a task. (How "effort" is defined and quantified may be open for debate.)

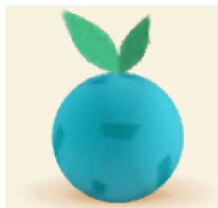
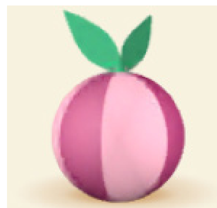
For students requiring more scaffolding for this activity, you can direct them to focus on one or two specific fruits and have a class discussion about which creatures might help them in this task.

Name ______________________________ Date ______________

Vocabulary Words	Definitions	How does this apply to the complex problem you just thought of?
Decomposition	Defining a problem and breaking that problem into smaller, more manageable parts.	
Pattern Recognition and Generalization	Using prior experiences to find patterns within a problem and remixing and reusing prior solutions to solve novel problems.	
Abstraction	Identifying relevant information and removing unnecessary details.	
Algorithms	Developing a sequence of rules or instructions to perform a task or solve a problem.	
Debugging	Evaluating your solution and iterating on your design as needed.	

Name ______________________ Date ______________

Find three different methods to collect food. Use the boxes below to record the different solutions you discover. Be sure to include a picture when possible.

Algorithm #1: Which fruit did you get? How did you get it?

Algorithm #2: Which fruit did you get? How did you get it?

Algorithm #3: Which fruit did you get? How did you get it?

What tips would you give a person who has never played this game to help them accomplish this task?

What were some of the most important factors you considered when deciding which creatures to use when getting food?

Which solutions were your favorite to use? Why?

Would you use your favorite solution to gather all the different types of food? Why or why not?

Compare your favorite solution with a friend. How are your methods similar? Different?

Facilitator Guide
Making It Work

Minimum Game Level

This activity is intended for players that have played Level 5 and up, since a Builder is part of the solution for Scenario 3.

Purpose

Students will create Algorithms for several scenarios to practice a Computational Thinking (CT) approach to solving problems.

Background Information for Facilitators

The focus of this activity is to highlight the importance of systematically approaching a problem when creating an algorithm. One of the common pitfalls students face when solving complex problems is to rush to a solution before having a complete grasp of the context. The process established on the worksheet (Decomposing and Abstracting a problem and looking for familiar patterns before creating an algorithm) is one that can serve as a template for all problem-solving scenarios in the classroom and beyond. This activity complements and builds on Three of a Kind as it once again asks students to explore all aspects of CT as a problem-solving strategy.

Activity in Action

1. Invite students to revisit what they know about how they use CT in their everyday lives, citing examples. It may help to revisit their work from the Three of a Kind lesson.

2. Review the Making it Work worksheet and emphasize that students will be using their CT skills to complete the activity.

3. Students can complete the worksheet in class or as homework.

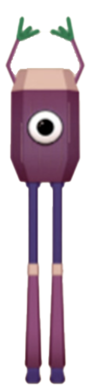
Reflection Questions

After students complete the worksheet, use these additional questions to help them further reflect on how they approached each scenario and ways that CT was used to complete the activity. Encourage students to explain their responses and solutions to this activity using CT vocabulary.

- What are some specific ways that you were able to use CT to complete this activity?
- Did anyone find any of the scenarios unfamiliar? How would you describe the difference between your analysis for the unfamiliar scenario compared to your analysis for a familiar one?
- How did you decide what information from each scenario was unimportant?
- Describe how making an algorithm in *the Pack* is similar to making an algorithm in class.
- Describe how making an algorithm in *the Pack* is different from making an algorithm in class.
- Describe other situations, in school or in life, where you feel it is important to create a game plan before diving into a solution.

What To Look For

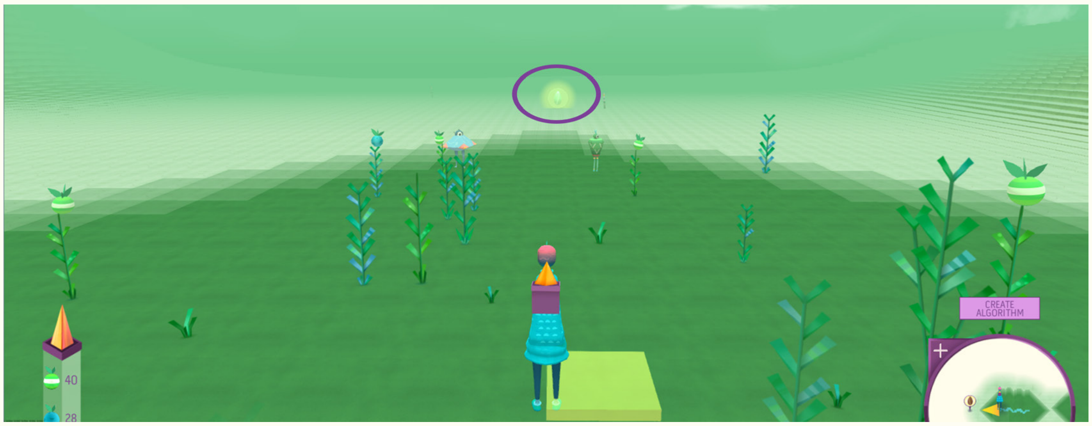
As students complete the worksheet for this activity, it will be interesting to note how final algorithms will differ from student to student. Each student's approach will depend on their experience playing the game and their priorities around what they consider to be a "good" algorithm. Despite any differences in the final algorithms, you should expect responses to show how students are using the context to understand the problem, including their assessment of what is or is not relevant to the solution. Responses should also show evidence of consideration of the smaller tasks that must be completed in order to carry out the final algorithm.



Sample Solutions

The following is a possible solution for Scenario 1. For students that are less detailed in their responses, be sure to ask clarifying and probing questions during the debrief.

Scenario 1: Analysis



1. What choices do I have here? What am I deciding between?
I only have the Digger and Mover here so I can either use them to extend the water that's behind me or I can walk towards the seed to see if I can reach it before passing out. If I use the creatures, I will use food supply and it might take more time. If I can reach the seed without passing out, it might be the fastest way to get it. I will try to walk to the seed.

2. What matters here? What does not matter here?
I don't think the seed is deep enough into the dry zone for the dry zone to matter. Since I'm not planning to use the creatures, I don't care how much food I have in my backpack. I also don't really care about the water source (how much, how far, etc.). What is important is how far I can walk into the dry zone without passing out.

3. What do I know about this situation? What tools do I need?
I know that if I walk too deep into a dry zone I will pass out, but I also know that I can walk a certain distance into the dry zone before that happens. If that doesn't work, I also know that I can extend the water source toward the dry zone which will require using a Digger > Mover creature algorithm about 5 – 8 times. That also means I will need at least 5 – 8 of each kind of food.

My algorithm for this is:
Step 1: Walk towards the seed.
Step 2: Touch it.

Differentiation

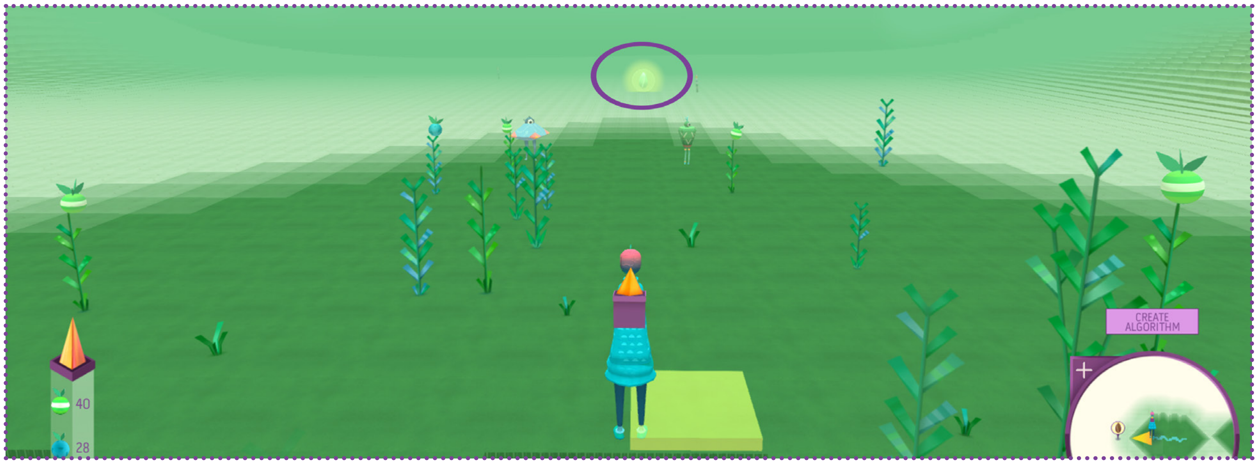
For students who may need some help getting started, do the first scenario in small groups or as a whole class.

Name

Date

Create an algorithm to get the seed for each scenario below.

Scenario 1: Getting Started



1. What choices do I have here? What am I deciding between?

2. What matters here? What does not matter here?

3. What do I know about this situation? What tools do I need?

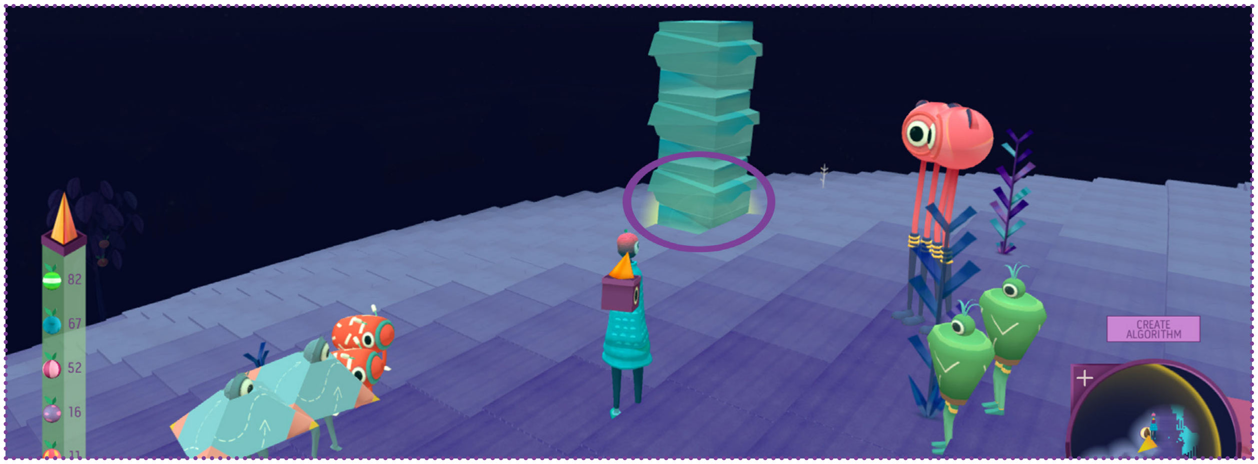
My algorithm for this is:

Name

Date

Create an algorithm to get the seed for each scenario below.

Scenario 2: Getting Started



1. What choices do I have here? What am I deciding between?

2. What matters here? What does not matter here?

3. What do I know about this situation? What tools do I need?

My algorithm for this is:

Name

Date

Create an algorithm to get the seed for each scenario below.

Scenario 3: Getting Started

1. What choices do I have here? What am I deciding between?

2. What matters here? What does not matter here?

3. What do I know about this situation? What tools do I need?

My algorithm for this is:

Name

Date

Were there any scenarios above that were not familiar to you? Which? How did you answer the Getting Started questions for unfamiliar scenarios?

How did the Getting Started questions help you create your algorithms?

Facilitator Guide
Stay Hydrated

Minimum Game Level

This activity is intended for players that have played Level 2 and up. Based on the in-game tutorial, students will be familiar with the task of extending a body of water towards a dry zone.

Purpose

Students will extend the nearest body of water enough to cross a dry zone to consider how the context of a problem affects the solution.

Background Information for Facilitators

This activity allows students to explore how the context of a problem (different constraints and parameters) affects the design of the solution. The fact that no two students will have the same experience completing the activity, despite similar goals, will help illustrate this. While everyone will be familiar with extending a body of water from the in-game tutorial, the terrain between a given dry zone and its closest body of water will vary greatly from game to game. This is because the terrain in *the Pack* is generated each time a new adventure is started. Some students will have an easy time of it while others will encounter a variety of problems based on the terrain. Discussing how students react to these problems will illustrate both the effect of context and the underlying priorities students apply when designing a solution to a problem.

Activity in Action

1. Start by asking students about an algorithm they use regularly — for example, an algorithm for lab safety, the process for solving a complex math equation, or the steps they take to get ready for school.

2. Ask "Does this algorithm work all the time?" This activity will serve as an example of how algorithms do not always work if the conditions of the problem vary. Be sure to note both "yes" responses, which will likely be tied to tried and true contexts (like lab procedures), and "no" responses, which will be more dependent on context.

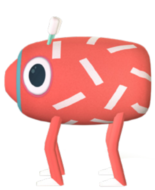
3. Go over the Stay Hydrated worksheet with students to make sure they understand the assignment.

4. Have students complete the worksheet, either as homework or in class, but note that students will need to work with a partner to complete the final worksheet question.

Reflection Questions

After students complete the worksheet, use these additional questions to help them further reflect on their gameplay and ways that they think computationally. Encourage students to explain their responses and solutions to this activity. They may not use newly learned vocabulary right away, but if you point it out in context, they will create their own connections between the Computational Thinking (CT) concepts and their respective terms.

- What were some of the different obstacles you encountered? How did this affect the design of your algorithms?
- Why might you have had a different experience than your partner?
- Did anyone think they had an algorithm that worked but then tried it and found out that it did not work? What did you do to improve it?
- Are any of the above questions things you would consider to be part of Decomposition? Why or why not?
- We might all have the same goal of arriving at school by a certain time. If everyone has the same goal, why do people get here at different times? What are all the factors that might affect your arrival time?
- Imagine you were hired to build a new bridge. What information would you need to know to do a good job? (Students might include things like materials, but probe responses to focus on the context. These are things like location, weather, distance from one end to the opposite end, budget, etc.)
- Think about the examples of algorithms that we brainstormed at the start of the activity. What are some of the reasons an algorithm might work sometimes, but not all the time? What might be different about algorithms that work most or all of the time?

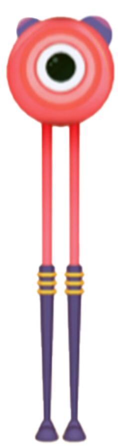


What To Look For

As you go over responses to the questions on the student worksheet and discussion, there are several things to look for that will help you highlight ways that your students already think computationally. While the questions in this activity are intended to provoke conversations around Debugging and Algorithms, you may hear students reflecting on their use of Abstraction and Decomposition as well.

In terms of Abstraction, listen for students' descriptions of things they excluded or did not consider as part of the context that influenced their solution design. For example, students may have ignored the amount of food required for a Bumper if their dry area did not include anything that needed to be bumped.

For Decomposition, listen for students connecting the context of their dry area to any obstacles they faced that required them to solve sub-problems. For example, students might describe running out of food for the Repeater, or encountering a hill that required a more complex algorithm. It may be helpful to remind students that Decomposition means breaking a problem down into smaller, more manageable problems. This includes all the tasks that you must do to ultimately solve the overarching problem. The context dictates what those smaller problems or tasks could be.



Sample Solutions

Student solutions will vary greatly depending on the terrain. Here is an example.

1. Walk to the edge of a body of water that is closest to the dry zone.

2. If there is a hill between the water and the dry zone, estimate how far down you will have to dig to make sure that water can continue to flow. (Gravity may or may not be a factor.)

3. First try the creature algorithm
Digger > Repeater x5 > Mover > Repeater ∞

4. Execute the creature algorithm and follow it to make sure the water continues to flow along with it.

5. If it flows that's it — cross the dry zone!

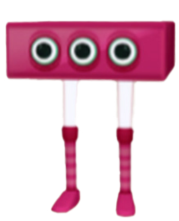
6. If it does not flow, pause the creature algorithm, find the spot where it gets stuck and adjust the first Repeater to dig more or less, based on the terrain and water flow.

7. Keep adjusting the algorithm to maintain the flow of water until you can cross the dry zone.

Differentation

For those students requiring an added challenge, add the requirement that the dry zone be on top of a hill, or at a higher elevation than the source of water. This will require careful planning and several adjustments of any creature algorithm they use for the task.

For those students requiring scaffolding, make sure the dry zone they select is both near a source of water, and as level with the water source as possible. This will ensure that they can use a single creature algorithm without consideration of terrain obstacles. Terrain like this is likely to be found in earlier levels of the game.



Name

Date

Find a dry zone and extend the nearest water source to cross it.

Describe how you did it.

Step 1:

Step 2:

Step 3:

Step 4:

Step 5:

Step 6:

Name

Date

Describe any obstacles you experienced:

Describe any things you tried that did not work:

Name

Date

Now compare your solution and obstacles with a partner. Given the obstacles your partner listed, is there any advice you can give them to improve their solution? Describe below:

Facilitator Guide
Who Put That There?

Minimum Game Level

This activity is intended for players who have reached at least Level 4, although students with more experience playing will have more prior knowledge to pull from.

Purpose

Students will utilize relevant prior knowledge to generalize solutions across problem scenarios.

Background Information for Facilitators

This activity focuses on the Computational Thinking (CT) concepts of Pattern Recognition and Generalization, and especially how they relate to Decomposition.

Pattern Recognition is an essential part of solving novel problems. You can begin to see its importance if you imagine attempting to solve every problem as though you have never seen anything like it before. This is daunting and counterproductive. Any new problem can likely be related to some previous problem and their solutions will share some similarities.

The ability to apply a solution to multiple contexts, both familiar and novel, is a critical part of thinking computationally. This requires identifying the similarities and differences between two contexts and applying or tweaking a known solution when appropriate. Being able to generalize a solution between two different contexts simplifies the task of solving novel problems.

Since this activity is asking students to pull from prior knowledge as a way to focus on the CT concepts of Pattern Recognition and Generalization and does not actually require students to design or test a solution, it does not require the use of the game. Instead, students will rely on prior knowledge to reflect on their problem-solving process. This requires some experience playing the game.

Activity In Action

Before you pass out the worksheet for this activity, begin with a short conversation:

1. Start by asking students to think of a situation where an experience they had in one setting helped them solve a similar problem in a different setting (e.g., being able to transfer what they have learned about one video game to solving problems in a different video game, or being able to apply their approach to solving a single word problem in math class to a logic problem in Computer Science class). You might give students time to reflect individually or in small groups before engaging in a whole class discussion.

2. Note the strategies students describe as they share, and point out when they describe aspects of CT.

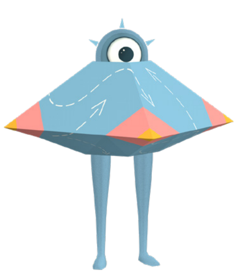
3. Pass out the Who Put That There? worksheets and make sure students understand the activity.

Reflection Questions

After students complete the worksheet, use these additional questions to help them further reflect on the activity:

- Pretend this was your first time playing *the Pack*; do you think you would have been able to solve the problem(s) in the scenarios in the picture? How might your approach to problem solving as a total beginner be different from your approach now?
- What are some algorithms you find yourself returning to again and again while you play? Do you use them to solve the same types of problems? Or can they be used for different things?

  > Has anyone saved or named their favorite algorithms on the Create Algorithm screen? What might be some benefits to saving and naming algorithms?
- How do you think Pattern Recognition and Generalization can be used by scientists? Engineers? Can you think of some specific examples?
- What about athletes? Do they use Pattern Recognition and Generalization? How?

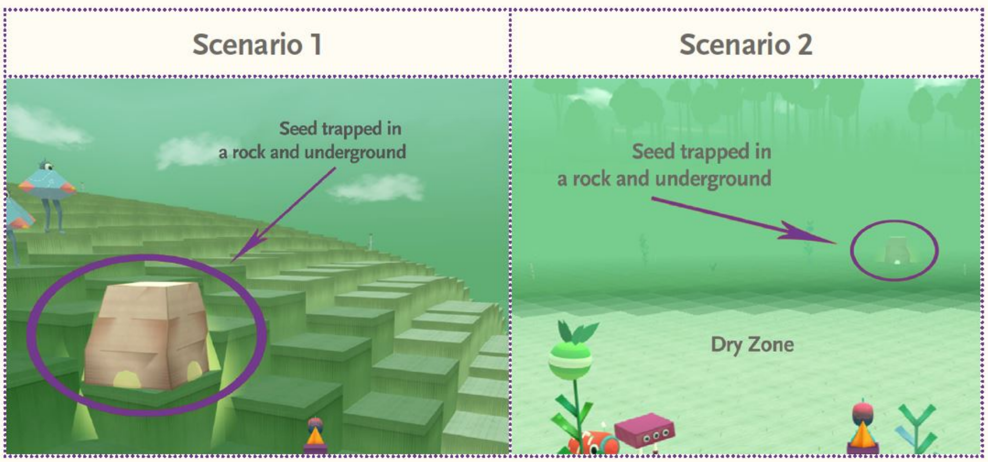


What To Look For

The questions in this activity are intended to provoke conversations around Generalization and Pattern Recognition. As students share out their responses to the worksheet and reflection questions, take note of how students reference their use of prior knowledge, and especially how they pull from prior knowledge to solve problems in unfamiliar situations.

Sample Solutions

In the two scenarios presented below, note that both involve an underground seed encased in stone. The seed in Scenario 2 is beyond a dry zone.



Based on the images presented and questions provided, here are possible responses:

What problem(s) do you need to solve to get the seed in Scenario 1?
I need to get to a seed that is both underground and under a rock.

What problem(s) do you need to solve to get the seed in Scenario 2?
I need to get to a seed that is both underground and under a rock, but I can't just dig it up and walk to it because it's past a dry zone and I can't walk through a dry zone.

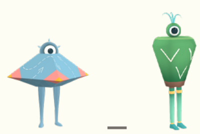
What are some of the differences between the two scenarios?
Scenario 1 has a lot of stepped elevation which is sometimes challenging for the Digger and Mover. The land in Scenario 2 is flat but there is a dry zone that I can't walk through. That means I will

have to either get rid of the dry zone with water or use a creature algorithm that involves the Grabber getting the seed for me.

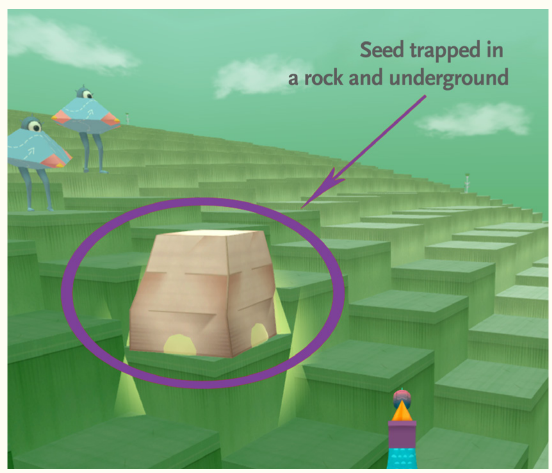
What are some of the similarities between the two scenarios?
Getting the seed in both scenarios requires either bumping or moving the rock and digging underground.

Scenario 1

1. 

2. 

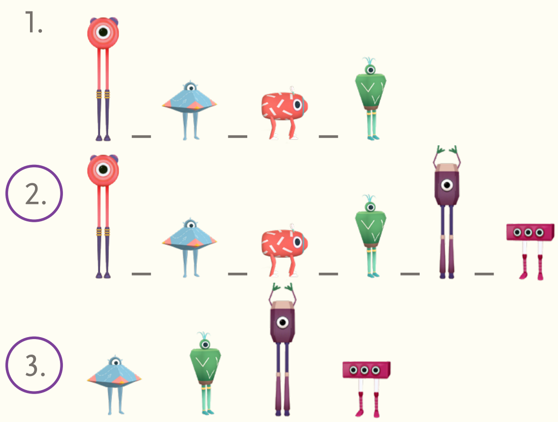
3. 

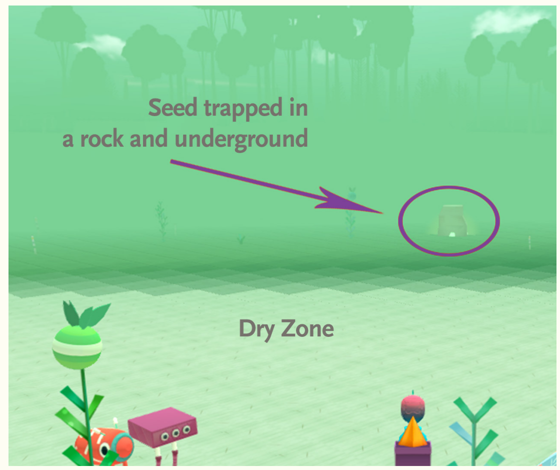


Circle the algorithm(s) that would work to get the seed in Scene 1?
Options 1 and 2 should be circled.

What made you choose these?
Options 1 and 2 would both work in this scene, but Option 3 would get stuck bumping underground.

Scenario 2





Circle the algorithm(s) that would work to get the seed in Scene 2?
Options 2 and 3 should be circled.

What made you choose these?
Options 2 and 3 would both get past all the obstacles. The first option would just get stuck searching back and forth near the seed without getting it or giving me access to it.

How can solving the problem in Scenario 1 help you solve the problem in Scenario 2?
Both scenarios use the same creatures, and the only real difference is getting past the problem of the dry zone. If I can come up with a solution for Scenario 1, all I have to do for Scenario 2 is tweak my solution from Scenario 1 by adding a Grabber and a Repeater.

Describe one experience you have had playing *the Pack* that helped you with this activity.
Answers will vary.

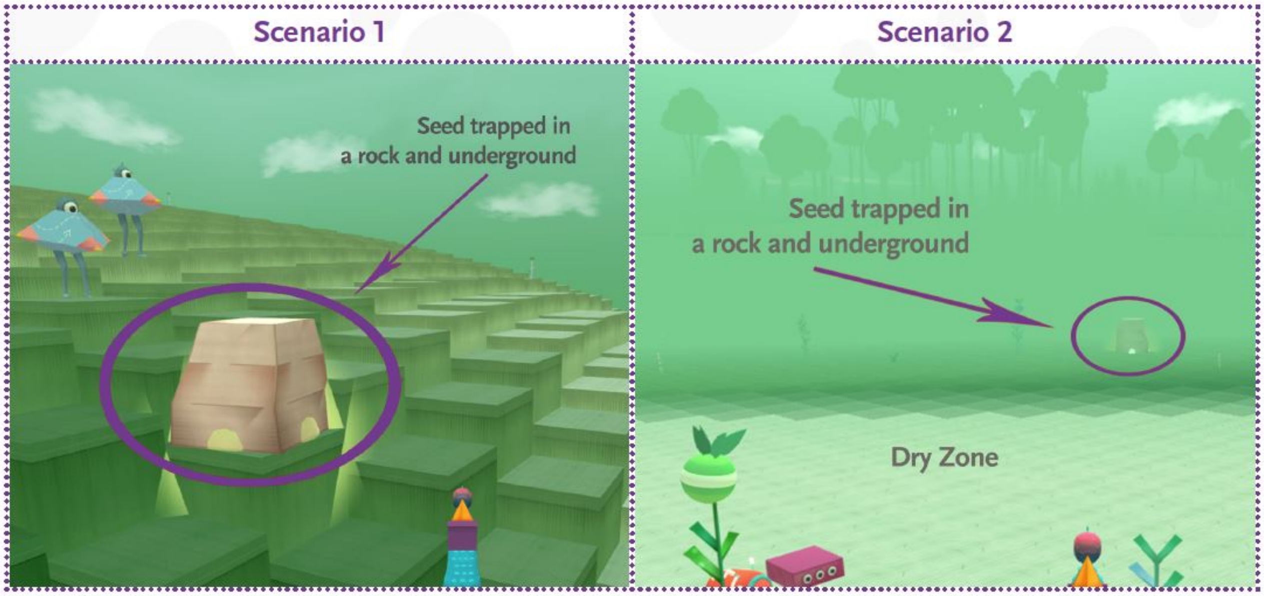
Differentiation

For those students who need scaffolding, you can begin filling out the worksheet as a class or in small groups.

Name

Date

Look at the two Pack scenarios below.



What problem(s) do you need to solve to get the seed in Scenario 1?

What problem(s) do you need to solve to get the seed in Scenario 2?

What are some of the differences between the two scenarios?

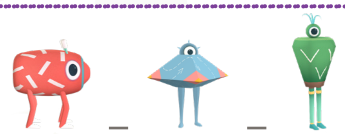
What are some of the similarities between the two scenarios?

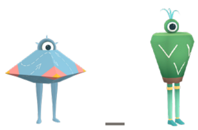
Name

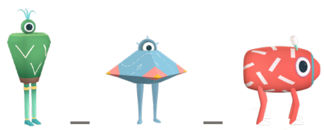
Date

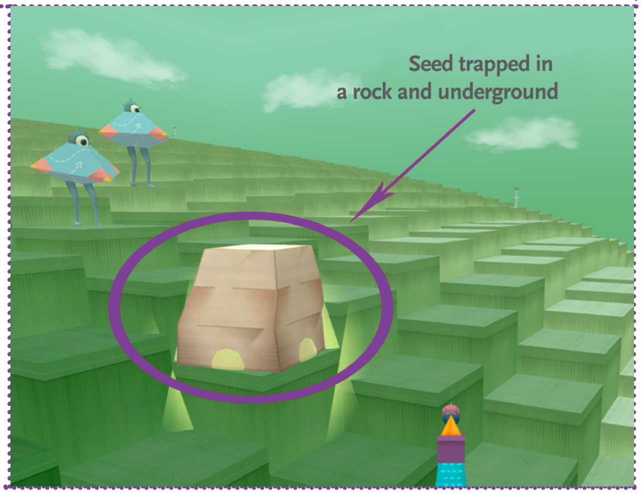
Circle the algorithm(s) that would work to get the seed in Scenario 1.

Scenario 1

1. 

2. 

3. 



What made you choose these?

Name

Date

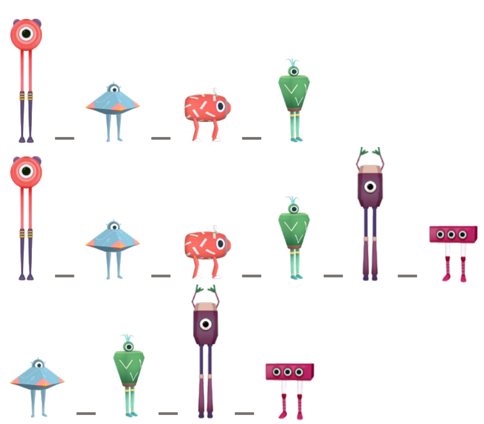
Circle the algorithm(s) that would work to get the seed in Scenario 2.

Scenario 2

1.

2.

3.



What made you choose these?

Name

Date

How did solving the problem in Scenario 1 help you solve the problem in Scenario 2?

Describe one experience you have had playing *the Pack* that helped you with this activity.

Facilitator Guide
Make and Break It

Minimum Game Level

This activity is intended for players that have played up to Level 4 to allow students to pull from a wide variety of gameplay experiences.

Purpose

Students will Debug a classmate's broken algorithm.

Background Information for Facilitators

A common scenario for computer programmers is debugging code that was created by someone else. This activity echoes that process by asking a student to debug an algorithm created by one of their classmates. It provides a first-hand illustration of the importance of context when debugging an algorithm and highlights the critical role of communication when collaborating with others.

Activity in Action

You can introduce this activity by inviting students to share how they have already used debugging in their gameplay. Were there times when an algorithm they created did not work the way they thought it would? What approach did they take to fixing it?

Review the Make and Break It worksheet and highlight that this challenge has four parts.

1. The design of an algorithm and its context
   - Provide students with time to brainstorm ideas for the problem that their algorithm will address.
2. The breaking of that algorithm
   - Remind students that they should break their algorithm by strategically changing one thing about it to stump their partners. However, "stumping" does NOT include changing the variable for the Seeker or Grabber (if they are using those creatures).
3. Debugging that broken algorithm
   - Students should now switch algorithms with a friend so that they are debugging an algorithm that is not their own.
   - Partners cannot ask algorithm designers for clues or communicate with them about the algorithm.
4. Analysis of the debugged algorithm
   - Partners return the algorithm they debugged back to the original owner.
   - The worksheet includes questions to help students reflect on the experience of debugging. If time allows, students can meet with their partners to discuss the experience and their approach to debugging each other's algorithms.

You may wish to have students complete Parts 1 and 2 and submit their worksheets to you before assigning student pairs for Parts 3 and 4. Students can use *the Pack* game for reference as they complete the worksheet.

Debrief and Discussion

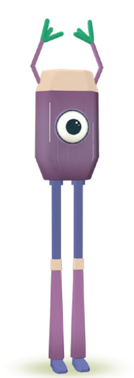
After students complete the worksheet, use these additional questions to help them further reflect on their gameplay:

- Did you debug the algorithm in a way that still met the original purpose? Or did the way you debug cause the algorithm to do something different than what was originally intended? Is this still debugging?
- What questions would you have asked the algorithm designer if you could? How did not being allowed to communicate with the original algorithm designer impact your ability to debug the algorithm?
- Can you think of a time in class or in life when you had to figure out why something was not working? How did you figure out where the problem was?
- Is your approach to debugging similar or different in different contexts (e.g., math class, science class, or playing a game)? Why or why not?

What To Look For

As your students complete the worksheet, look for examples of how they assess whether a solution either works or does not work. While they may not be able to accurately debug the algorithm to achieve the original solution, students should be able to explain their debugging choices. You should also hear students citing elements of the context of the problem in discussion of their responses during debrief of the activity.

You might also see both algorithm designers and debuggers spending a significant amount of time in *the Pack* trying out algorithms and using those algorithms in different areas of the game to test it in a variety of situations or locations.



Sample Solutions

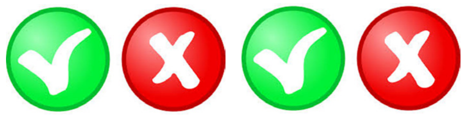
Your students should generate a variety of solutions for this challenge, so there will not be a clear set of sample solutions.

Differentiation

It is likely that your students will work at different speeds. You might consider separating students into groups based on how you anticipate they will work. For your fastest students, perhaps consider having them create algorithms for two different problem contexts.

Name ______________________ Date ____________

Create a creature algorithm that accomplishes a specific task and then break it so that it does not work. Challenge a partner to debug the algorithm so that it accomplishes the original task.



Use the boxes below to document your work.

Part 1

Your Algorithm
Create an algorithm and include a picture or description in the box below.

Purpose and Conditions of Your Algorithm
What does your algorithm do and under what conditions does it work?

Name

Date

Part 2

Broken Algorithm

Now break your algorithm by changing one thing about it. What does it look like now?

Did you decide to break your algorithm by removing a creature, adding a creature, or rearranging the creatures? How did this end up "breaking" the algorithm?

Now, draw or describe your broken algorithm on the next page where it says, "Broken Algorithm." You will give that sheet to a partner so that they can debug your broken algorithm.

Algorithm creator name

Date

Debugging partner name

Part 3

Broken Algorithm (completed by algorithm creator)
Draw or describe the broken algorithm and describe what it is supposed to do.

Debugged Algorithm (completed by debugging partner)
Draw or describe the “fixed” algorithm.

Now that you have debugged the algorithm, pass this page back to the original owner of the broken algorithm.

Name

Date

Part 4

While trying to debug your partner's algorithm, how did you try to figure out what was not working?

How do you know that you "fixed" your partner's algorithm?

Facilitator Guide
Seed Trio

Minimum Game Level

This activity is intended for players that have played Level 5 and up. Level 5 is the level in which you can have the Builder join your Pack, which is necessary for gathering any floating seeds.

Purpose

Students will explore the concepts of Decomposition and Abstraction as they design a solution for a given scenario.

Background Information for Facilitators

The focus of this activity is to highlight Decomposition and Abstraction in the process of solving a problem. When considering the scenario, students must use the image provided on the worksheet and their prior experience with the game to determine the important factors affecting their solutions.

Decomposing a problem and isolating, or Abstracting, the important factors and ignoring the irrelevant factors are often overlapping activities. Looking at the image, a student might begin by noting the actions that would need to be taken to get to the seeds, then determining whether they have the resources to take those actions. As they try out their solutions, they may have to reassess their actions, alternating between Decomposition and Abstraction.

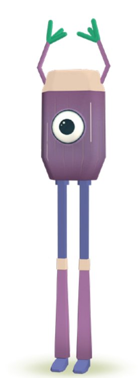
Activity in Action

1. Ask students, "When have you had a big problem that you needed to break down into smaller parts so that you could solve it?" and, "Were there any things you had to ignore to solve that problem, maybe because they were distracting or did not help you solve the problem?"

2. If your students have trouble providing examples, encourage them to think of recent examples of multi-step problems they have been asked to solve in class, or even examples from gameplay.

3. Review the Seed Trio worksheet to ensure students understand the scenario. The image makes it clear to students that two of the seeds are in the ground and one is floating. What may not be as clear is that the floating seed is over a dry zone, as indicated on the map.

4. Have students complete the activity either as homework or in class.



Reflection Questions

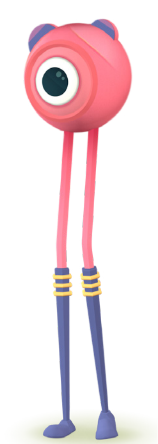
To begin the discussion, have students share their final algorithms. You can facilitate the share-out using whatever method works best for your class. For example, this could be done in small groups or with volunteers presenting to the whole class. After students complete the worksheet, use these additional questions to help them further reflect on the activity:

- What was the very first thing you considered when starting this activity?
- Did everyone start this way? Why or why not?
- Did anyone finish this activity in a way that was surprising to someone else? How so?
- What is the first thing you do when confronted with a complex problem?

> Do you always break the problem down into smaller parts? Why or why not?

- How do you decide what factors are important when trying to solve a problem?
- Can you think of any examples of things you have had to ignore to solve a problem, maybe because they were distracting or did not help you solve the problem?

What To Look For



Because Computational Thinking (CT) is a problem-solving strategy and not a linear set of steps, your students may start to notice points of overlap between the core CT concepts. For example, Decomposing a problem into smaller parts may look a lot like the steps of an algorithm. If you see students struggling to categorize something as Decomposition or Abstraction or an algorithm itself, use this as an opportunity to highlight that CT concepts work together. While it can be helpful to reflect on each component in isolation, in reality the boundaries might not be so clear-cut, and that is OK!

Sample Solutions

The following is one possible way students may describe their thinking. They may not provide as much logic behind their steps as in the example, but your debrief questions should prompt them to provide those details.

What smaller problems do you need to solve to get the three seeds?

- *I need to make sure I have enough food for any creatures I would use (only the Dropper's food is low).*
- *I will be making separate algorithms with a Digger and a Builder because two seeds are underground, and one is floating.*
- *Looking at the map I can see that the floating seed is in a dry zone, so I'll have to extend water into that dry zone, meaning I'll need the Digger, Mover and Repeater.*

Algorithm

Step 1: *I start off by digging up the buried seeds since I can just walk near them and just place and run the Digger on each buried seed before releasing the Digger.*

Step 2: *Next, I look at the map to figure out where the water area closest to the floating seed is and go to it.*

Step 3: *I use the Digger, Mover, Repeater algorithm with the Repeater set to infinity so that I can run it until it hydrates enough of the dry zone for me to walk near the floating seed.*

Step 4: *I locate the shadow under the floating seed to place the Builder, Repeater algorithm and set the Repeater to infinite and run it.*

Step 5: *Finally, I stop the Builder, Repeater algorithm when it reaches the seed and I walk to grab it.*

What factors did you find to be unimportant? Why?

1. I didn't care about the lack of Dropper food. I didn't plan on using the Dropper.

2. I also didn't care about the lack of Grabber food since I plan to get the seeds by walking over them.

3. The food growing closest in the scenario was unimportant since there are plenty of those in the backpack.

Differentiation

For students requiring an added challenge, have them pretend that they are starting this activity with no food. This way, they will have the added challenge of leaving the area to get food before finding the area with the three seeds again to then solve the problem of gathering each one. This added challenge will require the use of the map, an often underused feature of the game.

You could also ask students to come up with multiple solutions to this problem.

For students requiring scaffolding, have them work in teams of three (if possible). They can split the task of figuring out how to get each seed and then teach each other how to get each seed.

For additional scaffolding, you could begin the activity as a whole class before having students complete it on their own.

Name ______________________ Date ____________

There are three seeds in the scenario above.

What smaller problems do you need to solve to get the three seeds?

Name

Date

Algorithm

My steps to get all three seeds are: (Note: use however many steps you need)

1.

2.

3.

4.

5.

6.

7.

8.

Circle all the things in the picture that helped you decide what to include in the above algorithm.

What factors did you find to be unimportant in designing your algorithm? Why?

Facilitator Guide

How'd He Do It?

Minimum Game Level

This activity is intended for players that have played to Level 4 and up as the solutions include the use of the Grabber which is available starting on Level 4.

Purpose

Students will analyze the viability of multiple solutions to a problem given the context.

Background Information for Facilitators

Solving this challenge requires students to apply all the Computational Thinking (CT) strategies they have learned so far. Students are given a problem to solve as well as the context in which it must be solved. Using prior knowledge, they must Decompose the problem to Abstract the important factors they will ultimately use to make their decision about which algorithm was used in the scenario presented. Students are also Generalizing the use of the creature algorithms as solutions since previous use of these same algorithms may have been in a different context from what is presented on the worksheet. An important part of this challenge is how students explain the rationale for their choices. All the solutions presented have the potential to work, but they all require additional actions to be taken and factors to be considered.

Students should use the worksheet to document their reasoning. This activity prompts students to go beyond the general idea that context matters to actually analyze a specific context to determine how and why it matters. A helpful way to think about it is that they are not just choosing the solution they think is best or makes the most sense, but rather they are becoming an expert on the matter, capable of teaching someone else how to identify the best solution.

Activity in Action

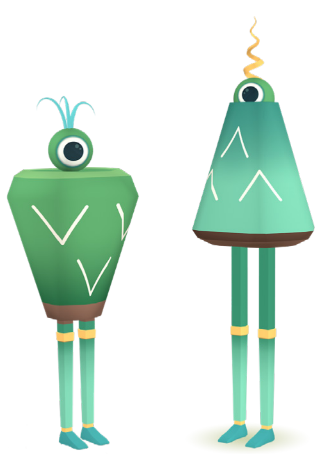
You can start off this activity by finding out what sort of experiences your students have had analyzing the pros and cons of a choice they had to make. They may require some prompts so consider the following as a conversation starter:

"How do you decide how to allot your time between doing homework, spending time with friends, or playing video games (or other favorite or important activity)?"

"How do you weigh the pros and cons, and analyze the possible outcomes to make a final decision?"

Transition to the worksheet by reminding students that while we have learned that there are multiple ways to solve any given problem, we must analyze the context before deciding how we will solve that problem.

Review the worksheet to make sure students understand the activity. Feel free to have students use *the Pack* game to test their ideas. Students can complete the worksheet in class or as homework.



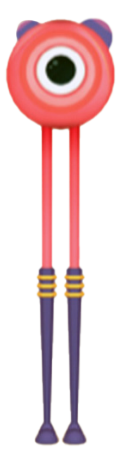
Reflection Questions

Once your students have completed their own analysis, it is equally important to give them the opportunity to compare and contrast their responses with their classmates' responses. After hearing from their peers some students may find that there are factors they had not considered.

After students complete the worksheet, use these additional questions to help them further reflect on their analysis:

- Are you absolutely sure that you considered everything that could possibly affect your choice for this challenge, or was there more you would do if you had more time?
- Did you have everything you needed to make a decision, or was there additional information that would have helped you decide which algorithm Ray used?
- Why do you think we came up with different answers?

  > Do you think we had different answers because we have different priorities, or do you think we had different answers because we interpreted the challenge in different ways?
- Have you ever made an important decision that you think you would have gone differently if you had been given different information at the start?
- Can you perform an analysis if you do not have all the information related to a problem? Why or why not?
- How would you teach someone to perform an analysis of a problem? Is there an algorithm for it?



What To Look For

An important thing to look for in this activity is the quality of the students' analysis of solutions. Students will likely have plenty of opinions to share based on prior knowledge, but it should be evident that they are not just basing their responses on opinion. If any solutions are unfamiliar, students should test them in-game to perform a thorough analysis. Since each player is playing on a different map of Algos generated by the game, they will not be able to totally recreate the same layout as in the picture. Encourage students to look for areas that are similar to the scenario from the worksheet when testing out each algorithm.

Sample Solutions

Below is a chart of things your students might consider to inform their responses to the first set of questions.

Solution #	Purpose of Algorithm	Context/Constraints/Trade-offs
1	This algorithm extends a body of water closer to the dry zone.	**Pros** *This solution permanently removes the obstacle. Once a body of water has been extended near the seed, the dry zone is quenched, and the player can traverse the area with no further action required.* **Cons** *The map of Algos shows that the player is not near a body of water. This means that they must first arrive at the nearest body of water to begin the difficult task of digging a path to the seed. They may also need to supervise the algorithm in action and modify it as dictated by the terrain.*
2	This algorithm leads the player to fruit for the Grabber creature.	**Pros** *This solution supplies the fruit needed to create an algorithm that would allow the player to use the Grabber to gather the seed without entering the dry zone. It takes the player directly to the desired fruit so that there is no searching or wandering involved.* **Cons** *This solution requires many steps, including the design of two algorithms to accomplish the task. It also has the potential to send the player far from the seed depending on how far the nearest required fruit is growing. Additionally, the player must revisit the location of the seed and then design another algorithm utilizing the Grabber to gather the seed.*
3	This algorithm enters the dry zone without the avatar and gathers the seed for the player.	**Pros** *This solution would enable the player to use an algorithm to grab the seed without entering the dry zone. This would be the fastest way to accomplish the task.* **Cons** *Because this solution requires a fruit that the player does not have, there are extra steps involved that nullify the advantage of being the fastest solution. Additionally, the player will have to either wander around looking for the needed fruit or create an additional algorithm to seek it out.*

There are many possible answers to this challenge. The important thing is that this student has performed a thorough analysis of the context and provided a rationale for their choice.

Based on your analysis, which algorithm did Ray use?	Why do you think this?
I think Ray used Algorithm #3: *Seeker > Mover > Grabber > Repeater*	*Looking at the picture for the challenge, I noticed the location was at the edge of the map, so I don't need to be in this area for anything but the seed. Therefore, I didn't need to bring water into it, so the first algorithm choice is not needed. I will probably need the Grabber for more than just this situation, so I should get more Grabber fruit anyway. Now that I have seen the seed, it will show up on my map, so I can always find it again later. I will wander around looking for food for the Grabber, and in the meantime, I can handle any other obstacles or tasks that I face along the way. Even though I'm leaving the seed for later, this saves me time in the long run, so I think this is the choice Ray would make.*

Is it possible he could have used one of the other ones? Why or why not?

Again, there are many possible answers to this challenge. All three algorithms are viable, so students should reference the pros and cons they outlined above.

Differentiation

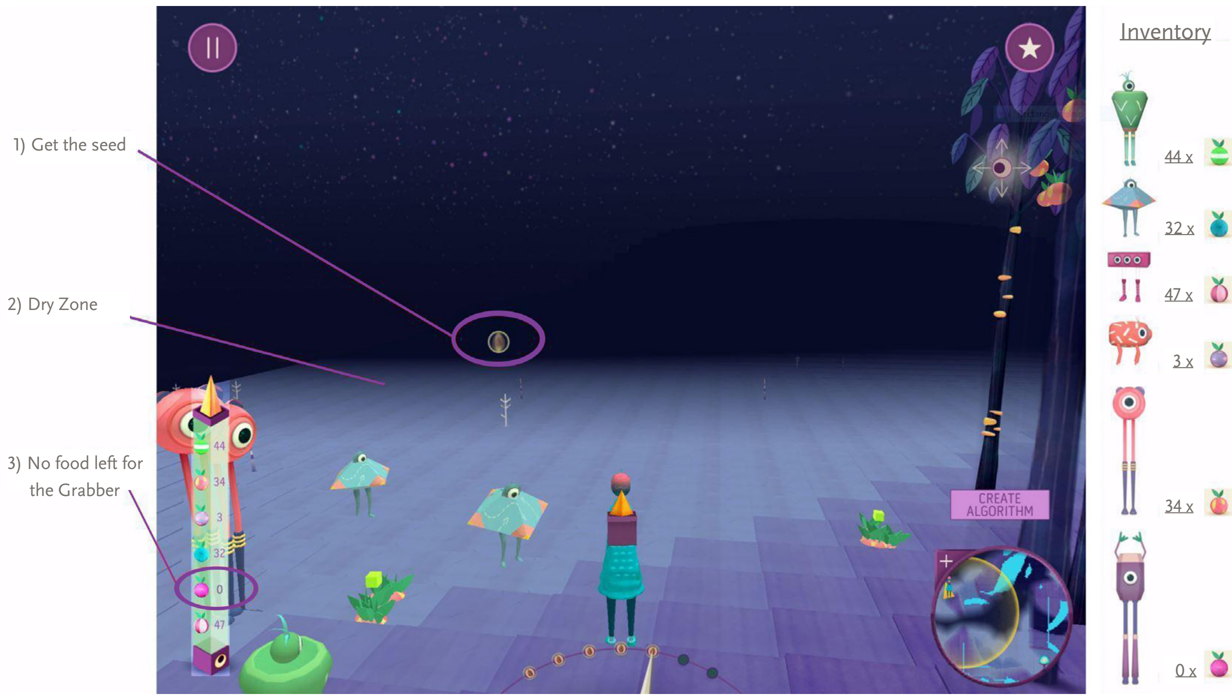
For those students requiring an added challenge, have them assume that there is no food of any kind in the player's backpack. This creates a greater number of details to Decompose from the larger problem and a larger number of factors to consider in selecting an appropriate response.

For those students requiring scaffolding, have them ignore the fact that there is no Grabber food left in the player's backpack. With this detail omitted, Algorithm #3 is the clear choice, and they will not have to consider ways to acquire food in addition to acquiring the seed.

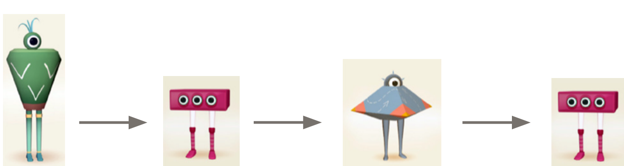
Name

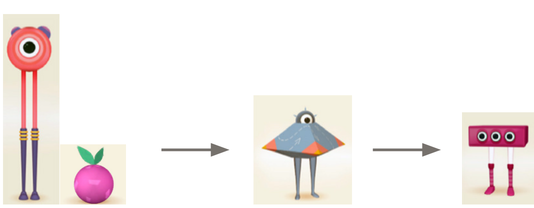
Date

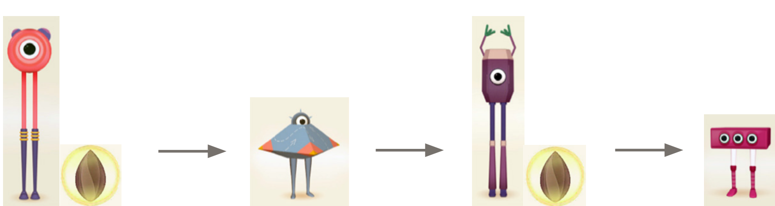
Examine the image below:



Ray loves playing *the Pack*. He successfully collected the seed in the scenario above using one of the algorithms below.

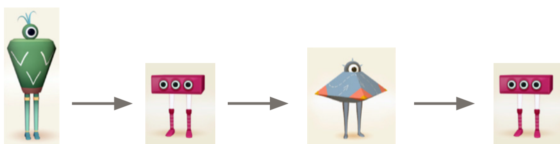
1) 

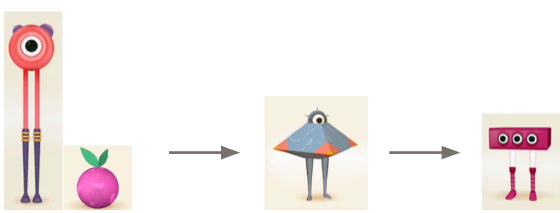
2) 

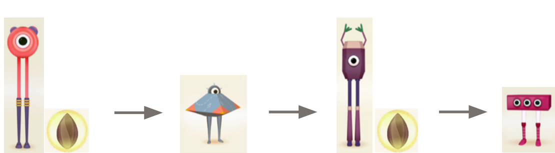
3) 

Name ______________________ Date ____________

Analyze each of the algorithms based on the context. What are the pros and cons of each algorithm given the context pictured in the scenario?

Algorithm #1	Pros and Cons of this Algorithm
	

Algorithm #2	Pros and Cons of this Algorithm
	

Algorithm #3	Pros and Cons of this Algorithm
	

Name

Date

Based on your analysis, which algorithm did Ray use? Why do you think this?

Is it possible he could have used one of the other ones? Why or why not?

Facilitator Guide
Map Gaps

Intended Level

This activity is intended for players that have played to Level 4 and up, to allow students to have enough familiarity with the map feature of the game.

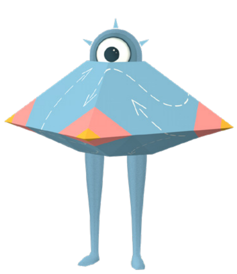
Purpose

Students will identify shortcomings of the map feature in *the Pack* and use Computational Thinking (CT) to suggest a redesign that would be more useful to players.

Background Information for Facilitators

This activity is a design-based challenge that is intended to benefit from the use of CT strategies. The problem presented (the map feature having shortcomings) has multiple parts or layers. It is also a novel problem, in that students are often consumers of a game and rarely if ever tasked with designing or redesigning games. CT strategies are best suited to unfamiliar, multi-step problems that can benefit from being broken down into smaller parts or tasks. As expert players of the game, students will have a wealth of prior knowledge to draw from as they consider ways to improve upon it.

Unlike many problems presented in a typical classroom scenario, this problem is undefined, and the student must first discern the boundaries of the problem that they will address. This is a common feature of real-life problems. Structuring problems in this way will give students practice identifying and applying CT strategies across multiple domains.



Activity in Action

1. Start by asking what students think of the map feature of the game and prompt students to note the reasoning behind their responses. You may have students that barely use the map, and others that find it a key aspect of their gameplay.

2. Review the worksheet to make sure students understand the activity and have them complete the worksheet either in class or as homework.

Reflection Questions

After students complete the worksheet, use these additional questions to help them reflect further. As you facilitate the discussion, encourage students to explain their responses.

- How did you decide what needed to be redesigned?

  > How did you use CT concepts in the process of redesigning?

- Do you feel like your redesign ideas are specific to *the Pack*, or could they be used to redesign map features in other games?
- What makes a game feature useful?
- Can you think of something from your everyday life you would want to improve the design of?

  > What information would you need to help you make an improved design?

  > How would you use CT in this scenario?

What To Look For

As you go over responses to the questions on the student worksheet, there are several things to look for that will help you highlight ways that your students have used CT strategies. At this point, they will likely be using the CT vocabulary and have a deeper understanding of the concepts that the CT vocabulary describes. As CT is meant to be generalized from an in-game context to the classroom and beyond, you should note when students are explicitly using CT in other contexts. Make sure to point this out to them in the moment; this will help them realize how the strategy can be useful as well as help them identify when to use it.

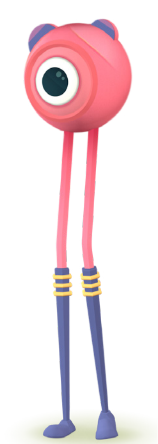
In terms of their redesign, listen for examples of all the tasks they have accomplished in order to complete the ultimate goal of a redesign. These smaller tasks are evidence of Decomposition. Likewise, listen for anything that they have deliberately excluded or ignored as a factor in their redesign as this would be evidence of Abstraction. Any redesign that makes sense to students based on what they know about the game can be taken as evidence of Pattern Recognition. The redesign should work with features of the game that are already in place; this can only be accomplished through the use of prior knowledge. This activity does not require an algorithm as a final product, but the steps taken by any student can be confirmed through conversation.

Sample Solutions

As this is a design-based challenge, solutions can and should vary greatly and will depend on the problem being addressed.

Differentiation

For students requiring an added challenge, have them address the trade-offs presented by their designs. You can also have students form groups and assess each other's individual map redesigns to see if they can create an even more refined version of the map feature.



Name

Date

Some players feel that the map in *the Pack* is not very useful. Redesign the map to be more useful to players.

Why do you think some players might say that the map feature is not very useful?

Name

Date

What changes would you make to the map to improve it? Sketch out anything you need to show visually.

Continue showing improvements on the back of this page if needed.

Name

Date

How would these changes make the map more useful to players?

Facilitator Guide
Level Up

Minimum Game Level

This activity is intended for players that have played all levels of *the Pack*.

Purpose

Students will design a new level of *the Pack* in order to reflect on and synthesize experiences from their gameplay.

Background Information for Facilitators

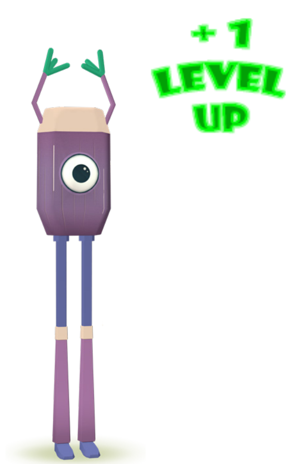
This challenge provides students with the opportunity to synthesize all the skills that were covered in previous activities. Students must use everything they have learned about the game to create a new game level with new environmental conditions that present new problems for a player to solve, as well as a new creature that will help the player overcome the new challenges.

Activity in Action

Introduce this challenge by telling students that they will now have the chance to design the next level of the game. This includes imagining a new Pack creature, this creature's favorite food, how and where that food grows, and obstacles on this level that can benefit from the new creature.

Review the Level Up worksheet and emphasize that a key part of this activity is to imagine the context (terrain, etc.) for the level that they will be designing. Remind students that the context will inform the new obstacle, and the new creature and algorithm(s) that will be used to tackle this obstacle. It may be helpful to have students spend some time brainstorming examples of past obstacles, either as a class or in small groups. Examples of past obstacles include scarce water, floating seeds, and large hills, all of which required new creatures.

Have students complete the worksheet, either as homework or in class.



Reflection Questions

After students complete the worksheet, use these additional questions to help them further reflect on their designs for the new level:

- How did the different terrains in the previous levels impact how you designed the terrain for this new level?
- What would happen if any of the previous creatures were introduced at a level other than the level at which you first met them? Why do you think they were introduced in the order they were?
- Compare your new level with a friend. What similarities or differences do you notice?

The following questions encourage your students to think beyond *the Pack* and consider ways that Computational Thinking (CT) can help solve problems in life. Recognizing CT as a problem-solving strategy, as well as being able to identify the situations that can benefit from the use of such a strategy, empowers your students to solve complex problems and take ownership of the experience. You may want to point out to your students that at this point, they are experts on *the Pack* and can apply the same type of thinking to other contexts!

- Reflect on your experience playing *the Pack*. Are there any CT strategies you used when playing *the Pack* (and designing a new level!) that you can apply to your life? If so, what?
- Has playing *the Pack* helped you identify algorithms in your everyday life? Can you give an example?
- Think about your gameplay experience. What are some aspects of gameplay that you feel you have gotten really good at (e.g., building algorithms, exploring the world of Algos, documenting your work, moving water, etc.)?

What To Look For

Students may want the ability to move creatures diagonally, to add conditions or parameters to algorithms, or even to be able to fly within Algos. Encourage students to think creatively and generate their own ideas.

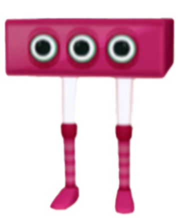
Also, keep a lookout for “new” creatures that actually perform a task that can already be done by a combination of existing creatures. If you notice that students are proposing a new creature whose job can be completed by an algorithm made up of existing creatures, prompt them to think about some function that might make their creature unique.

Sample Solutions

As this is a design-based challenge, solutions can and should vary greatly and will depend on the problem being addressed by the design.

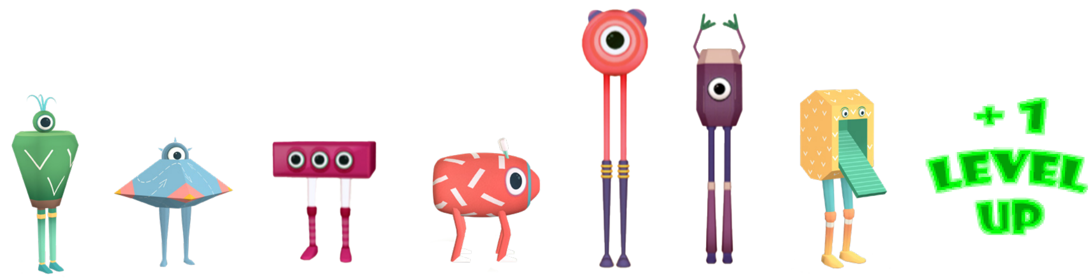
Differentiation

For students requiring scaffolding, encourage them to think about how the different Pack creatures were introduced at each level and how their different functions were specific to the new obstacles presented at that level. Additionally, you can have students focus their design efforts on the creature and the problem that the creature’s abilities address. This omits consideration of the fruit, which is to some degree cosmetic. You can also have students work in groups.



Name ______________________ Date ____________

Design the next level of the game.



Be sure to include:

- A description of the terrain
- A description of the new Pack creature that you will meet and the new fruit the creature needs.
- A problem specific to this new level and an example of how the new creature could be combined with the other Pack members to solve this problem.

Terrain Describe the landscape that is specific to this new level.	**Fruit** Describe the fruit specific to this new level. Where is it located? What does it look like?

Name

Date

New Obstacle

What new obstacle(s) are specific to this new level?

New Pack Member

What new creature will join your Pack? What special function will this new creature have that will meet the challenges of this new level?

Possible Algorithms

Give an example of how this creature could be combined with the other creatures to overcome the new obstacle(s).

Name

Date

OPTIONAL Use this space to draw your new level and creature!

Name

Date

How is your new Pack creature different from any other creatures or combinations of creatures?

Reflect on your overall experience with *the Pack*. How did your past gameplay influence what you designed for the new level?

The Pack program was supported by the US Department of Education (Award No. U411C190044), the National Science Foundation (Award No. 1543144), and the JPB Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the aforementioned funders.