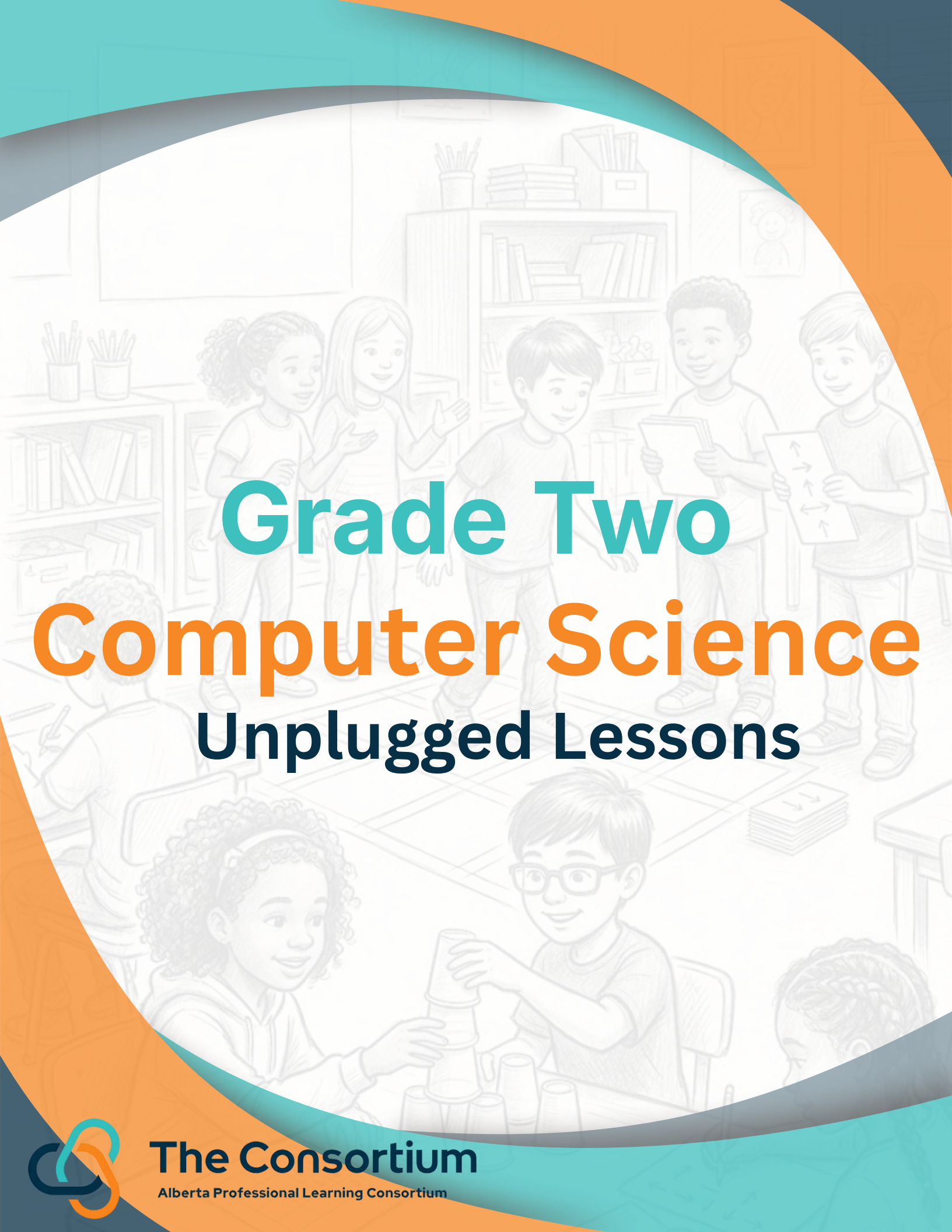


Grade Two Computer Science

Unplugged Lessons

The Consortium
Alberta Professional Learning Consortium

Grade 2 CS

Unit Overview: Level Up! Creative Instruction Design

Organizing Idea: Computer Science: Problem solving and scientific inquiry are developed through the knowledgeable application of creativity, design, and computational thinking.

Guiding Question: How can creativity support design?

Learning Outcome: Students apply creativity when designing instructions to achieve a desired outcome.

Lesson 1: The Creative Curator (Precision & Design)

Focus: Understanding that creativity is the ability to generate something original, and that precise instructions have specific components like verbs, simple language, clear steps, and a starting and stopping point.

Summary: Students will act as museum curators designing a strategy layout. Using a blank 5x5 grid, students will place a board game token at a starting point and draw a playing card to place on the grid as their "exhibit" (the stopping point). They will learn that creativity can be used to design instructions for games and computer programs. Students will work individually or in groups to create three-step to four-step instructions that achieve the desired outcome of navigating to the exhibit, ensuring they use precise words (e.g., "Move", "Turn").

Lesson 2: The Dice Predictor (Reliability)

Focus: Creating three- to four-step instructions and predicting their outcomes.

Summary: Students will continue using their grids, but this time, they will roll a standard 6-sided die to determine how many "obstacles" (small blocks) they must place on their grid path. Working in pairs, one student (the programmer) writes a precise navigation code to get around the obstacles. Before the token is moved, both students must look at the code and predict the outcome of instructions that have three to four steps. They will learn that the reliability of instructions means they consistently lead to the same desired outcome every time the code is run.

Grade 2 CS

Lesson 3: Bug Hunters in the Grid (Testing & Debugging)

Focus: Testing instructions and debugging errors to achieve a desired outcome and increase reliability.

Summary: The teacher will provide groups with a pre-set grid layout (complete with a token, obstacles, and a target playing card) and a pre-written set of instructions that contains an intentional error (a "bug"). Students will test the instructions with three to four steps to verify that the desired outcome is achieved. When the token inevitably crashes or misses the target, students will learn that debugging is the process of identifying and removing errors. They will debug the errors in the set of instructions to reach the target, proving that debugging increases the reliability of the code.

Lesson 4: The Efficiency Loop (Repetition)

Focus: Refining instructions for efficiency by using repetition.

Summary: Students will learn that the efficiency of instructions refers to designing in a way that yields desired outcomes with the least amount of energy, time, or steps. Using a deck of playing cards, a student draws a numbered card (e.g., a 4). They must move their grid token forward that many spaces. Instead of writing "Move Forward" four separate times, students will describe this as a situation in which repetition simplifies instructions. They will refine their instructions to more efficiently achieve the desired outcome by creating a loop (e.g., "Repeat Move Forward 4 times").

Lesson 5: The Collaborative Strategy Game

Focus: Collaborating to design clear three- to four-step instructions, including repetition, to achieve a desired outcome.

Summary: In this final lesson, students apply everything they know about precision, grids, testing, and efficiency. They will learn that collaboration can result in improved ideas, which may enhance creativity and problem solving. In small groups, students will exchange ideas to design a complete, original mini-board game on their grids using dice for movement rules and playing cards for targets. They will exchange ideas to design clear three- to four-step instructions, including repetition, to teach other groups how to navigate their specific game board.

Grade 2 CS Lesson 1

Lesson Title: The Creative Curator
Subject Area: Science (Computer Science)
Grade Level: Grade 2
Duration: 50–60 minutes

Lesson Overview

In this lesson, students learn that creativity can be used to design instructions for games. By treating a blank grid like a new museum layout from a strategy management game, students will use playing cards and board game tokens to design a custom museum map. They will learn that precise instructions have a variety of components, including verbs, simple language, clear steps, and a starting and stopping point.

Learning Objectives: By the end of the lesson, students will be able to:

- ☐ Identify ways creativity is used to design instructions for games.
- ☐ Work individually or in groups to create instructions, using precise words, pictures, or diagrams.
- ☐ Create three-step to four-step instructions that achieve a desired outcome.

Essential Questions:

- ★ How can creativity support design?
- ★ What makes an instruction precise and easy for a computer (or a player) to follow?

Materials & Resources

Category	Item Description
Classroom Tools	Blank 5x5 paper grids (one per group), small board game tokens (or toy cars), and standard decks of playing cards.
Technology	Document camera to model the grid setup.
Files/Handouts	Sets of printed "Arrow Coding Cards" (Forward, Turn Right, Turn Left) and a "Precise Code" recording sheet.

Grade 2 CS Lesson 1

Instructional Procedure

5–10 min | The Hook

Activity Description: Leveling Up our Museum.

Teacher Instructions: Display a blank grid, a board game token, and a playing card under the document camera.

Teacher Script:
"Programmers, today we are going to level up our coding skills by building our own museum strategy board games! Creativity is the ability to generate something original, such as ideas, technology, tools, or products. Today, you are going to use your creativity to design a museum map, and then write the precise code to navigate a curator through your brand new exhibits!"

15–20 min | Direct Instruction

Activity Description: The Anatomy of Precise Instructions.

Teacher Instructions: Explain the required components of a good code. Write the components on the board as a visual checklist.

Teacher Script:
*"When we design our game, we can't just guess where the pieces go. Instructions are designed by using creativity and problem solving. To win the game, your code must be precise! Instructions can be created to be precise, reliable, and efficient to achieve the desired outcome. Precise instructions have a variety of components, including:

1. **Verbs:** Action words like 'Move', 'Turn', or 'Stop'.
2. **Simple language:** Keeping it short and easy to read.
3. **Clear steps:** Numbering them Step 1, Step 2, Step 3.
4. **A starting and stopping point:** We must know exactly where the curator begins and where the exhibit target is.

Let's look at how we will use our tools to set our starting and stopping points today."*

Grade 2 CS Lesson 1

20–30 min | Guided Exploration

Activity Description: The Museum Strategy Grid.

Teacher Instructions:

1. Group students into pairs or threes. Distribute a 5x5 grid, a board game token, arrow cards, and a few playing cards to each group.
2. **Set the Starting Point:** Have students place their token (the "Curator") on the bottom-left square of the grid. This is the required starting point.
3. **Set the Stopping Point (The Exhibit):** Have students draw one playing card and place it anywhere on the top half of the grid. This card represents the stopping point (the new museum exhibit).
4. **Write the Code:** Groups must collaborate to create three-step to four-step instructions that achieve a desired outcome (reaching the playing card). They will place their arrow cards in order (pictures/diagrams) and write the precise words on their recording sheet using verbs and simple language (e.g., "1. Move Forward. 2. Turn Right. 3. Move Forward.").

Teacher Script:
"Work with your team to design your museum board! Place your playing card exhibit on the grid. Then, work in groups to create instructions, using precise words, pictures, or diagrams to move your curator token. Make sure you use three to four steps, and remember your verbs and clear steps!"

10 min | Closure

Activity Description: Evaluating Precision.

Teacher Instructions: Have a few groups share their recording sheets and read their code aloud.

Teacher Script:
"Let's review our codes. Did your instructions include verbs and simple language? Did you clearly mark your starting and stopping point? By working together, your collaboration resulted in improved ideas for your games. Tomorrow, we are going to add obstacles to our museum floors using dice, and we are going to predict the outcomes before we even move our tokens!"

Grade 2 CS Lesson 1

Differentiation

Support/Scaffolding: For students struggling to write the precise words, provide a "word bank" at their table with the required verbs (Move, Turn, Stop) to assist with incorporating simple language. Keep their playing card exhibit closer to the start so they only need a simple three-step code.

Extensions: Challenge advanced groups to draw *two* playing cards for their grid, representing a two-part exhibit tour. They must create a sequence that stops at the first card, pauses, and then continues to the second card.

Assessment

Formative Assessment: Review the "Precise Code" recording sheets during the Guided Exploration to ensure students can create three-step to four-step instructions that achieve a desired outcome. Check that their written instructions include verbs, simple language, clear steps, and a starting and stopping point.

Name: _____

The Creative Curator

🔍 Activity 1: The Anatomy of Precise Code

Precise instructions help our curator reach the exhibit. Draw a line from the coding rule to the example that follows it.

Coding Rule	Example
Use a Verb (Action Word) 🏃	Turn Right
Simple Language 🗣️	Move Forward
Clear Steps 🔢	1. Move Forward
Start and Stop Points 🏁	From the Start to the Exhibit

🎯 Activity 2: Is it Precise?

Instructions must be precise and easy for a player to follow. Look at the two codes below. Circle YES if the code is precise or NO if it is not precise.

Code	Is the code precise?
"Go over there to the card."	**YES** / **NO**
1. Move Forward. 2. Turn Right.	**YES** / **NO**

Name: _____

The Creative Curator

🎨 Activity 3: Precise Code Recording Sheet

Design your museum exhibit on your grid. Then, write your 3-step or 4-step code using precise words and arrows.

My Museum Goal **Draw your goal here:** (Where is the exhibit?)	My Precise Code
	Step 1: ➡️
	Step 2: ➡️
	Step 3: ➡️
	Step 4 (Optional): ➡️

Grade 2 CS Lesson 2

Lesson Title: The Dice Predictor (Reliability)
Subject Area: Science (Computer Science)
Grade Level: Grade 2
Duration: 50–60 minutes

Lesson Overview

In this lesson, students continue to build their strategy board games on a 5x5 grid, introducing standard 6-sided dice to generate random obstacles. Before physically moving their tokens through the museum layout, students will create three-step to four-step instructions and then pause to predict the outcome. They will learn that computers are predictable, and that the reliability of instructions means they consistently lead to the same desired outcome.

Learning Objectives: By the end of the lesson, students will be able to:

- ☐ Create three-step to four-step instructions that achieve a desired outcome.
- ☐ Predict the outcome of instructions that have three to four steps.
- ☐ Explain that the reliability of instructions means they consistently lead to the same desired outcome.

Essential Questions:

- ★ How can we predict what our computer will do before we run the code?
- ★ What makes an instruction reliable?

Materials & Resources

Category	Item Description
Classroom Tools	Blank 5x5 paper grids (one per pair), small board game tokens, standard 6-sided dice, playing cards, and small wooden blocks (for obstacles).
Technology	Document camera.
Files/Handouts	Sets of printed "Arrow Coding Cards" and a "Prediction Protocol" recording sheet.

Grade 2 CS Lesson 2

Instructional Procedure

5–10 min | The Hook

Activity Description: The Unpredictable Die vs. The Predictable Code.

Teacher Instructions: Hold up a 6-sided die and a set of arrow coding cards.

Teacher Script:
"Programmers, if I roll this die, can you predict exactly what number it will land on every single time? (Roll the die). No! A die is random. But what if I show you these three cards: 'Forward, Forward, Turn Right.' Can you predict exactly where my token will go? Yes! Code is not random. When we design our strategy games today, we are going to learn how to predict the outcome of instructions that have three to four steps before we ever touch our game pieces."

15–20 min | Direct Instruction

Activity Description: Reliability and Prediction.

Teacher Instructions: Use the document camera to show a 5x5 grid with a token and a playing card target. Place a block on the grid to act as an obstacle. Lay out three arrow cards next to the grid.

Teacher Script:
*"In a good computer program, the code does exactly what you tell it to do, every single time. We call this reliability. The reliability of instructions means they consistently lead to the same desired outcome.
Let's test this. Look at my grid, and look at my three arrow cards: Forward, Forward, Forward. Before I move my curator token, let's predict the outcome of instructions that have three to four steps. Put your finger on your own desk and trace where my token will go. Will it hit the playing card, or will it hit the block obstacle? (Wait for students to predict). Now, let's test it! (Move the token). You predicted correctly! Because instructions are reliable, we can predict the outcome."*

Grade 2 CS Lesson 2

20–30 min | Guided Exploration

Activity Description: The Dice Predictor Challenge.

Teacher Instructions:

1. Group students into pairs. Distribute a grid, token, die, wooden blocks, playing cards, and arrow cards to each pair.
2. **Setup:** Have students place their token at the start. They roll the die; whatever number appears (e.g., 3), they must place 3 block obstacles randomly on their grid. Then, they place one playing card on an empty square to represent the target exhibit.
3. **Program:** Student A (the Programmer) creates a three- to four-step sequence using the arrow cards to navigate around the blocks toward the playing card.
4. **Predict:** *Before* moving the token, Student B (the Predictor) must look at the code and trace the path with their finger, predicting the outcome.
5. **Test:** Student B moves the token to see if the prediction was correct and if the reliability of the instructions led to the desired outcome.
6. Swap roles and repeat.

Teacher Script:
"It's time to build your museum floors! Roll your dice to see how many obstacles you get. Programmers, lay out your code. But wait! Do not move your token yet. You must stop, point, and predict the outcome of instructions that have three to four steps. Remember, if your code is well-written, its reliability means it will consistently lead to the same desired outcome!"

10 min | Closure

Activity Description: Reflecting on Reliability.

Teacher Instructions: Have students clean up their grids.

Teacher Script:
"Who was able to successfully predict their partner's code today? (Raise hands). Why were you able to predict it? Because the reliability of instructions means they consistently lead to the same desired outcome! Code does exactly what it is written to do. Tomorrow, we are going to look at what happens when there is a mistake in our code. We are going to become Bug Hunters, testing and fixing errors on our grids!"

Grade 2 CS Lesson 2

Differentiation

Support/Scaffolding: If a student struggles to predict from looking at abstract arrow cards, give them a dry-erase marker (if grids are laminated) to physically draw the path the arrows indicate before attempting to move the 3D token.

Extensions: Have advanced students create a "glitch" prediction. Have them intentionally point an arrow at an obstacle, and ask their partner to predict exactly *which* step in the sequence will cause the token to crash.

Assessment

Formative Assessment: Actively observe the pairs during the Guided Exploration. The key assessment checkpoint is ensuring students are pausing to explicitly predict the outcome of instructions that have three to four steps *before* physically moving the token. Listen for them using the word "reliable" when discussing their outcomes.

Name: _____

The Dice Predictor

🎲 Activity 1: Predict the Path

Reliability means instructions consistently lead to the same outcome. Look at the code below. Predict where the curator token will go on the grid.

The Instructions	My Prediction
⬆️ ⬆️ ⬆️	
⬆️ ⬆️ ➡️	
⬅️ ⬆️ ⬆️ ➡️	

🎯 Activity 2: Is it Reliable?

In computer science, code is not random. Circle YES if the outcome is reliable (it happens the same way every time) or NO if it is unpredictable.

Situation	Is it Reliable?
Rolling a 6-sided die to see a number.	**YES** **NO**
Following the code: "Forward, Forward, Turn Right".	**YES** **NO**
Using the same 3-step arrow cards on the same grid.	**YES** **NO**

Name: _____

The Dice Predictor

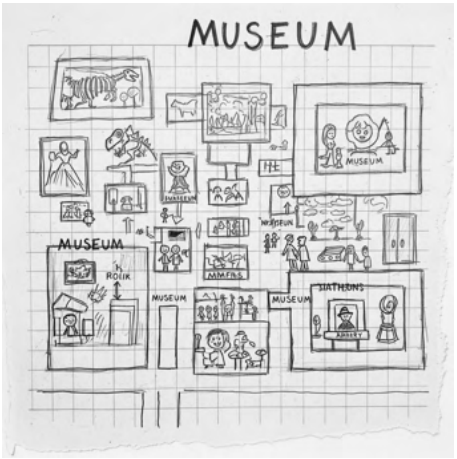
🎨 Activity 3: Be the Programmer

Roll your die to place your block obstacles. Then, write your 3 or 4-step code below.

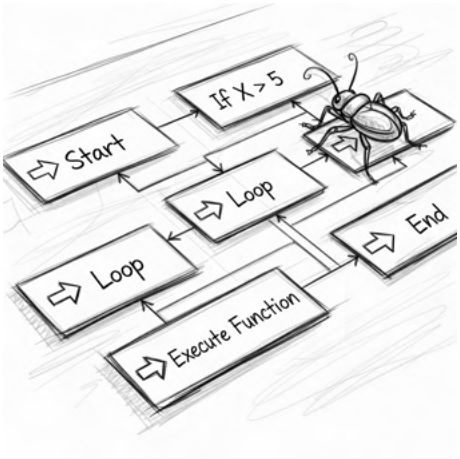
My Museum Goal

My Code (Draw your arrows):

My Partner's Prediction (Where will it land?):



Grade 2 CS Lesson 3



Lesson Title: Bug Hunters in the Grid
Subject Area: Science (Computer Science)
Grade Level: Grade 2
Duration: 50–60 minutes

Lesson Overview

In this lesson, students learn what happens when computer instructions are not perfectly precise. Using their museum strategy grids, students will receive code that contains an intentional mistake. They will learn that debugging is the process of identifying and removing errors in a set of instructions to achieve a desired outcome. By testing and fixing the code, students will understand that debugging can increase the reliability of instructions.

Learning Objectives: By the end of the lesson, students will be able to:

- ☐ Test instructions with three to four steps to verify that a desired outcome is achieved.
- ☐ Identify what a "bug" is in computer programming.
- ☐ Debug any errors in a set of instructions to achieve a desired outcome.

Essential Questions:

- ★ What do we do when our code doesn't work?
- ★ How does fixing errors make our instructions more reliable?

Materials & Resources

Category	Item Description
Classroom Tools	Blank 5x5 paper grids, board game tokens, wooden blocks (obstacles), and standard decks of playing cards.
Technology	Document camera.
Files/Handouts	"Buggy Code" cards (pre-written 3-4 step arrow card sequences that intentionally crash the token into an obstacle or miss the target).

Grade 2 CS Lesson 3

Instructional Procedure

5–10 min | The Hook

Activity Description: The Broken Game.

Teacher Instructions: Set up a grid under the document camera with a token, a block obstacle right in front of it, and a playing card target to the side.

Teacher Script:
"Programmers, I need your help. I bought a new video game, but I think it is broken! The instructions say: 'Move Forward, Move Forward.' Let's try it. (Move the token forward twice, crashing it loudly into the wooden block). Oh no! My curator crashed! The code didn't work. In computer science, an error in our code is called a 'bug'. Today, we are going to become Bug Hunters!"

15–20 min | Direct Instruction

Activity Description: How to Debug.

Teacher Instructions: Explain the definition of debugging and model how to fix the broken code from the hook.

Teacher Script:
*"When a programmer finds a bug, they don't throw the computer away. They fix it! Debugging is the process of identifying and removing errors in a set of instructions to achieve a desired outcome.
Let's look at my broken code again. To fix it, we have to test instructions with three to four steps to verify that a desired outcome is achieved. Step 1 was 'Move Forward'. That caused a crash. So, the error is in Step 1! Let's identify the error and remove it. Instead of moving forward, let's change Step 1 to 'Turn Right'. Now let's test it again. It worked!
By fixing that error, we made sure the code will work perfectly next time. Debugging can increase the reliability of instructions."*

Grade 2 CS Lesson 3

20–30 min | Guided Exploration

Activity Description: The Bug Hunter Challenge.
Teacher Instructions:

1. Group students into pairs and distribute the grids, tokens, playing cards, blocks, and arrow cards.
2. Hand each pair a "Bug Tracker" map (a paper showing exactly where to place their token, their block obstacle, and their playing card target).
3. Hand each pair a "Buggy Code" card. This card features a sequence of 3 to 4 steps that is guaranteed to fail based on the map.
4. Have students physically test instructions with three to four steps to verify that a desired outcome is achieved. They will watch their token fail.
5. Pairs must collaborate to identify the wrong arrow, remove it, and replace it with the correct arrow, working together to debug any errors in a set of instructions to achieve a desired outcome. **Teacher Script:**

"Bug Hunters, your grids are ready! Set up your museum floors exactly like your maps show. Then, run your Buggy Code. When your token crashes or misses the exhibit, don't panic! Work with your partner to identify the bad instruction. Remove it, replace it, and test it again until you debug any errors in a set of instructions to achieve a desired outcome!"

10 min | Closure

Activity Description: Debriefing the Debugging Process.

Teacher Instructions: Have students hold up the arrow card that caused their specific "bug."

Teacher Script:
*"Look at all those errors we removed! Who can tell me what debugging is? (Expected answer: Finding and fixing errors). Exactly! Debugging is the process of identifying and removing errors in a set of instructions to achieve a desired outcome. When we take the bugs out, our games run smoothly every time, which means debugging can increase the reliability of instructions." *

Grade 2 CS Lesson 3

Differentiation

Support/Scaffolding: For students struggling to find the error in a 4-step sequence, have them test the code one single step at a time, pausing after each movement to verify if they are still safe or if that specific step was the bug.

Extensions: Once a pair successfully debugs their given code, challenge them to create their own "Buggy Code" using their arrow cards. They can swap seats with another advanced pair and try to debug each other's custom errors.

Assessment

Formative Assessment: Circulate with a checklist during the Guided Exploration. Note which students can successfully test instructions with three to four steps to verify that a desired outcome is achieved. Specifically, observe if they can independently debug any errors in a set of instructions to achieve a desired outcome without the teacher explicitly pointing out which step was wrong.

Name: _____

Bug Hunter Mission

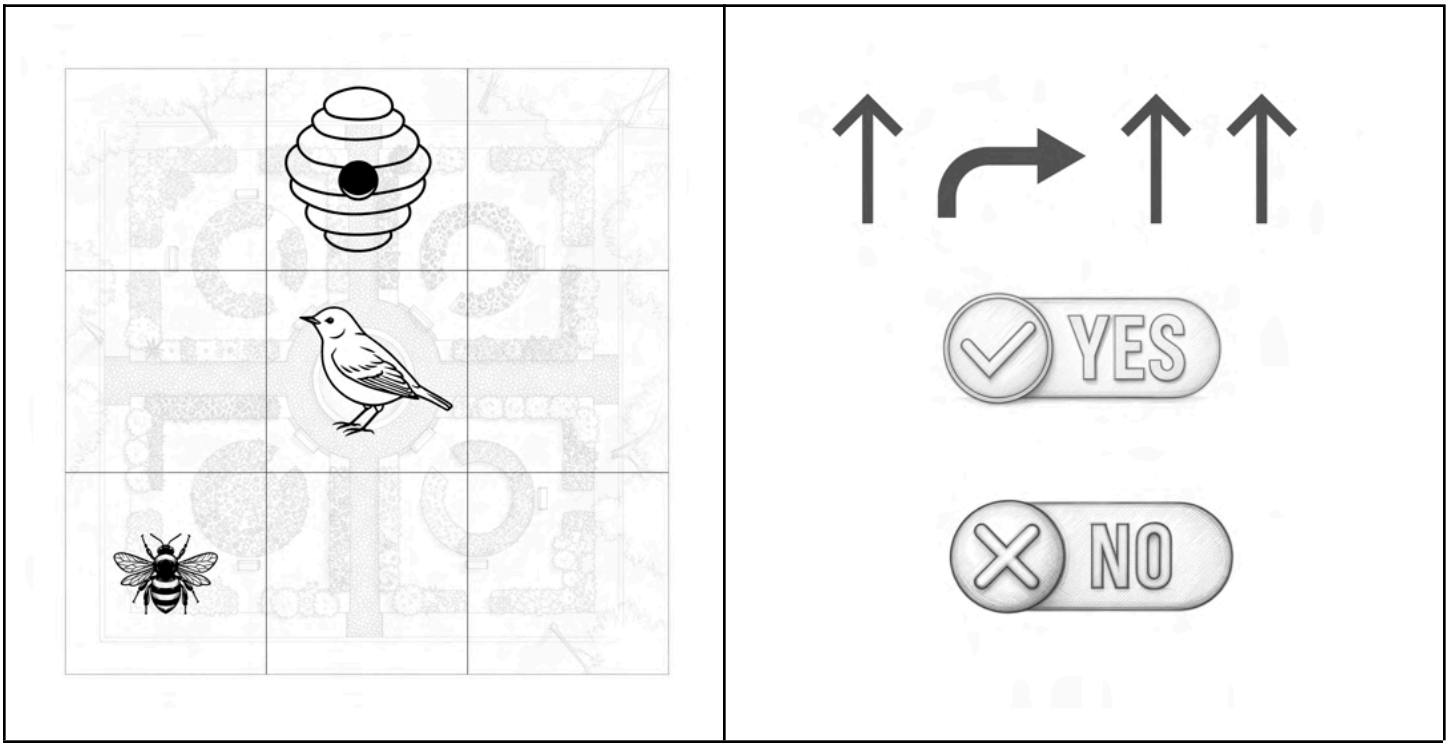
🔍 Activity 1: Bug Hunter Words

In computer science, we have special words for errors and fixing them. Draw a line from the word to what it means.

The Word	What does it mean?
Bug 🐞	Finding and fixing errors in code.
Debugging 🛠️	Making sure instructions work every time.
Reliability ✅	An error in a set of instructions.

🎯 Activity 2: Test the Code

Testing instructions helps us find bugs before they cause a crash. Look at the code below and the grid map. Circle YES if the code works or NO if there is a bug.



Name: _____

Bug Hunter Mission

🎨 Activity 3: Debug the Path

Draw a new path of 3 to 4 steps that avoids the obstacle and reaches the star.

My Debugged Map (Draw your arrows here)

Grade 2 CS Lesson 4

Lesson Title: The Efficiency Loop (Repetition)
Subject Area: Science (Computer Science)
Grade Level: Grade 2
Duration: 50–60 minutes

Lesson Overview

In this lesson, students learn that writing good code isn't just about reaching the target; it's also about saving time and energy. By drawing numbered playing cards, students will navigate long straightaways on their museum grids. They will learn that the efficiency of instructions refers to designing in a way that yields desired outcomes with the least amount of energy, time, or steps. Students will practice using "loops" (repetition) to make their code shorter and faster to write.

Learning Objectives: By the end of the lesson, students will be able to:
Identify that many daily activities include repeated steps.
Describe a situation in which repetition simplifies instructions.
Refine instructions to more efficiently achieve a desired outcome.

Essential Questions:
How can we write instructions faster and with fewer steps?
What does it mean for code to be efficient?

Materials & Resources

Category	Item Description
Classroom Tools	Blank 5x5 paper grids, board game tokens, and standard decks of playing cards (with face cards removed, leaving only numbers 2 through 5).
Technology	Document camera.
Files/Handouts	Sets of printed "Arrow Coding Cards" and a new "Loop Card" (a card with a circular arrow and a blank space to write a number).

Grade 2 CS Lesson 4

Instructional Procedure

5–10 min | The Hook

Activity Description: The Daily Repeat.

Teacher Instructions: Stand at the front of the room and mime brushing your teeth very exaggeratedly, narrating every single brush stroke.

Teacher Script:
"Programmers, watch me get ready for the day. 'Brush up, brush down. Brush up, brush down. Brush up, brush down. Brush up, brush down.' Phew! That took a lot of words! Many daily activities include repeated steps, such as brushing teeth or tying one shoe and then using the same process on the other shoe. What if I just said, 'Brush up and down 20 times'? Would that be faster? Yes! Today we are going to learn how to make our computer code faster by using repetition."

15–20 min | Direct Instruction

Activity Description: Efficiency and Simplification.

Teacher Instructions: Place a grid under the document camera. Put a token at the start and a playing card target 4 squares away in a straight line.

Teacher Script:
*"To get my token to the exhibit, I could use my arrow cards and write: Forward, Forward, Forward, Forward. It works, but it takes a lot of time and steps.
In computer science, we care about efficiency. The efficiency of instructions refers to designing in a way that yields desired outcomes with the least amount of energy, time, or steps.
Instructions may be simplified by repeating steps.
Instead of laying down four arrows, I can use a Loop Card! I will lay down ONE Forward arrow, and next to it, my Loop Card with the number 4. 'Repeat Forward 4 times.' Did I reach the same target? Yes! By using repetition, I was able to refine instructions to more efficiently achieve a desired outcome."*

Grade 2 CS Lesson 4

20–30 min | Guided Exploration

Activity Description: The Playing Card Loop Challenge.

Teacher Instructions:

1. Group students into pairs and hand out the grids, tokens, arrow cards, loop cards, and a small stack of numbered playing cards (2-5 only).
2. One student draws a playing card (e.g., a 5 of Hearts) and places their target that many squares away in a straight line.
3. The other student must write the code two ways. First, the long way (laying down 5 Forward arrows). Second, the efficient way (using 1 Forward arrow and 1 Loop card with a "5").
4. Ask them to look at both sets of code and verbally describe a situation in which repetition simplifies instructions.
5. Have them test the efficient code by moving the token. Swap roles and draw a new playing card.

Teacher Script:
"Time to make our museums efficient! Draw a numbered playing card to see how many spaces away your exhibit is. Programmers, lay out your long code first, then refine instructions to more efficiently achieve a desired outcome by using your Loop cards. When you look at the long code versus the short code, talk to your partner and describe a situation in which repetition simplifies instructions!"

10 min | Closure

Activity Description: Debriefing Efficiency.

Teacher Instructions: Have students clean up their materials.

Teacher Script:
"You were incredibly efficient today! Who can remind the class: does efficiency mean we use *more* steps or *fewer* steps to reach our goal? (Expected answer: Fewer steps!). Yes! The efficiency of instructions refers to designing in a way that yields desired outcomes with the least amount of energy, time, or steps. Tomorrow, in our final lesson, you will work together in groups to design your very own game boards and write efficient, reliable code to beat them!"

Grade 2 CS Lesson 4

Differentiation

Support/Scaffolding: For students who struggle with the abstract concept of the "Loop Card," have them literally stack their identical arrow cards on top of one another and count the pile (e.g., a stack of 4 Forward arrows) to bridge the physical quantity to the written number on the loop.

Extensions: Challenge advanced pairs to draw *two* playing cards and design an L-shaped path. They must use a loop for the first straightaway, an arrow to turn, and a second loop for the second straightaway.

Assessment

Formative Assessment: Circulate the room and listen to pair discussions. Specifically check if students can accurately describe a situation in which repetition simplifies instructions (e.g., "Using the loop is better because I don't run out of forward arrows and it's faster to read"). Verify that they successfully refine instructions to more efficiently achieve a desired outcome using the Loop Card.

Name: _____

The Efficiency Loop

🔁 Activity 1: Spot the Repeat

Draw a line from the activity to the step that gets repeated.

Daily Activity	Repeated Step
Brushing your teeth 🪥	👣 Step, step, step
Walking to the museum 🏛️	🧼 Brush up, brush down
Jumping rope ➰	👏 Clap, clap, clap
Cheering for a friend 📣	🦘 Jump, jump, jump

⏱️ Activity 2: Is it Efficient?

Efficiency means reaching the goal with the least amount of energy, time, or steps. Look at the two codes below. Circle YES if Code B is more efficient than Code A.

Code A (The Long Way)	**Code B (The Loop Way)**	**Is Code B more efficient?**
➡️ ➡️ ➡️ ➡️ ➡️	➡️ + 🔁 5	**YES** **NO**
➡️ ➡️	➡️ ➡️	**YES** **NO**
⬆️ ⬆️ ⬆️ ⬆️	⬆️ + 🔁 4	**YES** **NO**

Name: _____

The Efficiency Loop

✍️ Activity 3: Be the Efficient Programmer

Look at the long code in the box. Your goal is to refine these instructions to make them shorter and faster to write.

The Long Code

➡️➡️➡️➡️

My Efficient Code (Use a Loop Card!) 🔁

Grade 2 CS Lesson 5

Lesson Title: The Collaborative Strategy Game
Subject Area: Science (Computer Science)
Grade Level: Grade 2
Duration: 50–60 minutes

Lesson Overview

In this final unit lesson, students apply everything they have learned about precision, reliability, testing, and efficiency. They will discover that many people, individually or in groups, can create instructions, such as teachers, parents, students, and computer programmers. Working in design teams, students will build a complete, original mini-board game on their grids using dice and playing cards, and then host a classroom tech showcase to share their efficient codes.

Learning Objectives: By the end of the lesson, students will be able to:

- ☐ Explain how collaboration can result in improved ideas, which may enhance creativity and problem solving.
- ☐ Exchange ideas to design clear three- to four-step instructions, including repetition, to achieve a desired outcome.
- ☐ Test and debug a peer's game board.

Essential Questions:

- ★ How do we work together to design a creative game?
- ★ How can we use repetition to make our winning code as efficient as possible?

Materials & Resources

Category	Item Description
Classroom Tools	Blank 5x5 paper grids, board game tokens, standard 6-sided dice, wooden blocks (obstacles), and standard decks of playing cards.
Technology	None required (Unplugged activity).
Files/Handouts	Sets of printed "Arrow Coding Cards", "Loop Cards", and blank "Final Code" recording sheets.

Grade 2 CS Lesson 5

Instructional Procedure

5–10 min | The Hook

Activity Description: The Tech Showcase.

Teacher Instructions: Gather the students on the carpet.

Teacher Script:
"Programmers, you have reached the final level! Think about all the amazing games and apps we use every day. Did one single person build them all alone? No! Many activities at school and in the workplace require creativity and collaboration to improve ideas. Today, you are going to work in design teams to build the ultimate museum strategy game. When we are finished, we are going to turn our classroom into our very own mini Convergence 20.0 technology conference, where you will present your 'Level Up!' game designs for other teams to test!"

15–20 min | Direct Instruction

Activity Description: Exchanging Ideas Safely.

Teacher Instructions: Briefly model how to compromise when designing the game board.

Teacher Script:
*"Before we open our conference, you have to build your exhibit. When you work with your partner today, you must collaborate. Collaboration can result in improved ideas, which may enhance creativity and problem solving. If your partner wants to put the playing card target in the top right corner, and you want it in the top left, how can you solve the problem? (Expected answer: We can talk about it, we can take turns, or we can use two cards).
Exactly! Once your board is built, your team must exchange ideas to design clear three- to four-step instructions, including repetition, to achieve a desired outcome. That means you MUST include a Loop Card in your final code to make it efficient!"*

Grade 2 CS Lesson 5

20–30 min | Guided Exploration

Activity Description: The Collaborative Build & Showcase.

Teacher Instructions:

1. Group students into pairs. Hand out all building materials (grids, tokens, dice, cards, blocks, arrow cards, and loop cards).
2. **The Build:** Give pairs 10 minutes to roll their dice for obstacle placement, choose their playing card targets, and write their final, efficient code using at least one repetition loop.
3. **The Showcase:** Treat the room like a technology conference. Have Pair A leave their code on their desk and walk over to Pair B's desk. Pair B explains their game and Pair A tests the code. If there is a bug, the two teams work together to fix it. Swap so everyone gets to present and test.
   Teacher Script:

"Design teams, begin your builds! Remember to exchange ideas to design clear three- to four-step instructions, including repetition, to achieve a desired outcome. Once your games are tested and working reliably, we will open the showcase and invite the other teams to play your levels!"

10 min | Closure

Activity Description: Unit Celebration.

Teacher Instructions: Bring the class back to the carpet for a final wrap-up of the Grade 2 unit.

Teacher Script:
"I am so incredibly proud of the creativity you showed today. You proved that collaboration can result in improved ideas. Over this unit, you learned how to write precise instructions, predict outcomes, debug errors, and use loops for efficiency! You aren't just instruction followers anymore; you are instruction designers. Give yourselves a massive round of applause for leveling up your computer science skills!"

Grade 2 CS Lesson 5

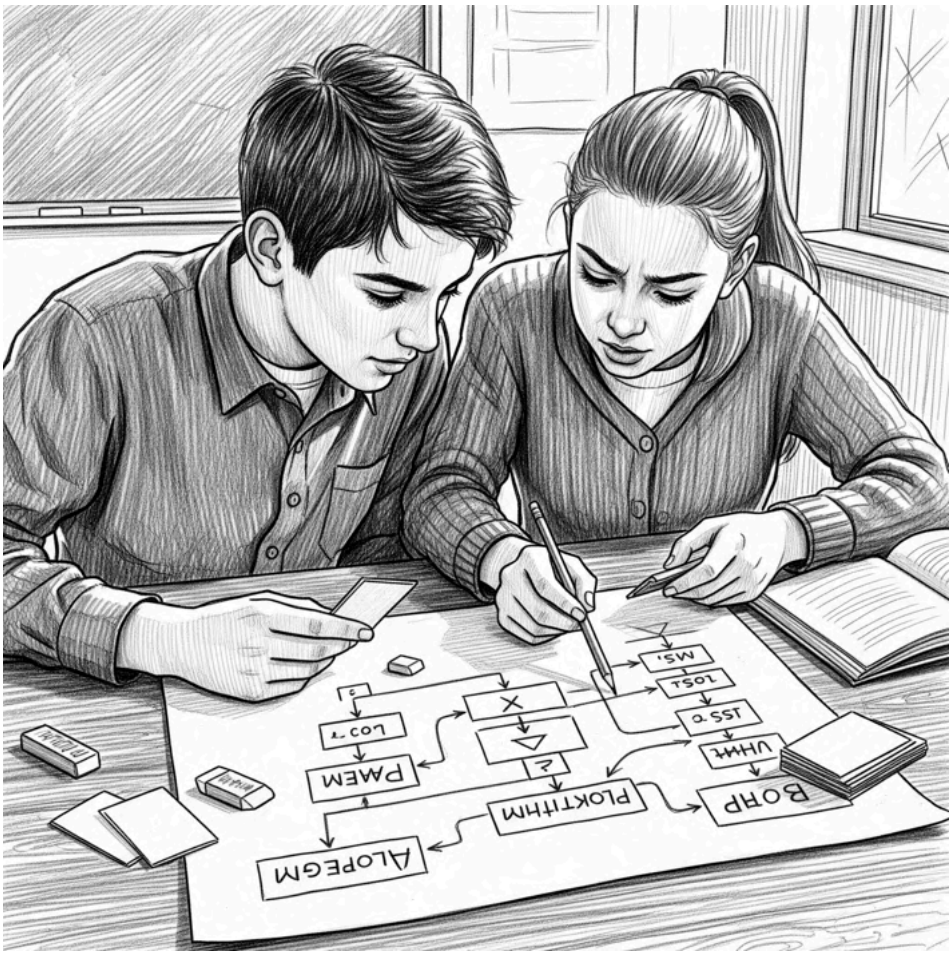
Differentiation

Support/Scaffolding: For pairs who struggle to collaborate without conflict, give them specific, alternating roles: Student A rolls the die and places the blocks, Student B places the playing card. Student A picks the first two arrows, Student B picks the loop card, etc.

Extensions: Challenge advanced pairs to create a "Multiplayer Level." They must design a game board that requires two different tokens to start on opposite sides of the grid and reach the exact same playing card target using two distinct sets of 3-4 step repeating codes.

Assessment

Formative Assessment: Walk the room with an observation clipboard during the showcase. Check that each group was able to successfully exchange ideas to design clear three- to four-step instructions, including repetition, to achieve a desired outcome. Verify that their final code on the recording sheet includes at least one Loop Card to demonstrate efficiency.



Name: _____

Level Up! The Collaborative Strategy Game

🤝 Activity 1: Collaboration Check

Collaboration means working together to improve our ideas. Look at the situations below. Circle the choice that shows collaboration.

Situation	Choice A	Choice B
Your partner wants to use the red token, and you do too!	"I am using it first and you can't have it."	"Let's take turns or find a second red token so we both can play."
You are stuck on where to put an obstacle on the grid.	"I'll decide everything myself since I'm the lead programmer."	"What do you think? Let's roll the dice together to see where it lands!"

🔁 Activity 2: The Loop Challenge

Efficient code uses Loops to repeat instructions. Look at the two codes below. Both move the token 3 spaces to the right. Circle the efficient code that uses a Loop Card.

Code Option A	Code Option B
➡️ ➡️ ➡️	🔁 (Repeat 3 times) ➡️

Name: _____

Level Up! The Collaborative Strategy Game

🎨 Activity 3: Design Your Showcase Level

Draw your 5×5 strategy game board below. Mark your Start, your Obstacles (blocks), and your Goal (playing card).

Name: _____

Level Up! The Collaborative Strategy Game

📜 Activity 4: Final Code

Write the efficient code for your game board using at least one **Loop Card**.

Step 1	
Step 2	
Step 3	
Step 4	
Step 5 (optional)	
Step 6 (optional)	