



Learn with Tayo: Coding Adventures Answer Booklet for parents

Module 1: What is coding?

Worksheet, Level 1 Beginner (suggested 7+) (21 items)

1. **Code puzzles:** 1. blue; 2. yellow; 3. bottom; 4. sprite
2. **Order:** 1. Think; 2. Break it up; 3. Code; 4. Run
3. **True or false:** 1. False; 2. True; 3. True; 4. False
4. **Match up:** Code = B; Algorithm = D; Sprite = A; Program = C
5. **Short answers**

1. What is a sprite?

A character or object on screen that the code moves.

2. Draw or name one thing that runs on code.

Any: a tablet, a game, a robot, traffic lights, a microwave.

3. What is the first step when you get dressed?

Open answer, for example: put on my underwear or socks first.

4. Explain in your own words: Explain what code is, in your own words.

Open answer: the idea explained simply in the learner's own words, with one correct fact.

5. Stretch: Why must instructions for a computer be clear and in order?

Open answer: a clear explanation using one or two facts from the module.

Worksheet, Level 2 Game Builder (suggested 8+) (25 items)

1. **Code puzzles:** 1. `print("Hi");` 2. `Go!` 3. `print("Hi")` (lower-case p); 4. `print("Hi")` (missing bracket); 5. 3

2. **Order:** 1. Think; 2. Break it up; 3. Order; 4. Code; 5. Run

3. Case study

1. What did Tayo forget?

The quote marks around Hello.

2. Write the correct line.

`print("Hello")`

3. Why did it print nil?

Without quotes Lua thinks Hello is a variable name, and it has no value.

4. Key terms: Code = Instructions written for a computer.; Algorithm = A list of clear steps in the right order.; Sprite = A character or object that your code moves.; Program = A finished set of code that does a job.; Block = One instruction you snap together with others.

5. Think it through

1. Explain why an algorithm should be planned before coding.

Planning the steps first makes the order clear, so the code is easier to write and has fewer mistakes.

2. Explain what the Output window is for.

It shows messages from print() so you can see what your code is doing.

6. Longer answers and 7. Extended answer

1. Describe the difference between block code and typed Lua code.

Blocks are dragged and snapped together; Lua is typed words that must be spelled exactly.

2. Write an algorithm of 5 steps for making a drink of squash.

Get a cup, pour some squash, add water, stir, drink.

3. Why do coders use comments?

To leave notes explaining the code to people; the computer ignores them.

4. Explain in your own words: What is the difference between block code and typed code?

Open answer: an accurate explanation in the learner's own words, with supporting detail.

5. Extended answer: 'Anyone can learn to code.' How far do you agree?

Open answer: an introduction, points for and against with examples and numbers from the module, and a reasoned conclusion.

Quiz

1. Code is...

B. A list of instructions for a computer

2. A list of clear steps in order is called an...

B. Algorithm

3. A character that your code moves is a...

A. Sprite

4. Block code runs...

C. From top to bottom

5. Which Lua line shows Hello?

A. print("Hello")

6. Can a computer guess what you mean?

B. No, it only follows instructions

7. Write an algorithm for brushing your teeth, in 5 steps.

Model answer: For example: pick up the brush, put on toothpaste, brush the top teeth, brush the bottom teeth, rinse.

8. Why do you think computers need very exact instructions?

Model answer: Because they cannot guess or fill in missing steps; they only do what the code tells them.

Checkpoints

Open answers. Look for the learner explaining the idea in their own words, with at least one accurate fact or number from the lesson.

Module 2: Moving and sequences

Worksheet, Level 1 Beginner (suggested 7+) (21 items)

1. **Code puzzles:** 1. 3; 2. 3; 3. wait; 4. socks
2. **Order:** 1. Start; 2. Plan; 3. Wait; 4. Test
3. **True or false:** 1. False; 2. True; 3. False; 4. True
4. **Match up:** Sequence = B; Run = D; Motion block = A; Position = C
5. **Short answers**

1. Which block moves the cat right?

Move right.

2. Draw arrows to get the cat to the fish.

Arrows right, right, right, up, up.

3. Why add a wait block?

So we can see each move.

4. Explain in your own words: Explain what a sequence is, in your own words.

Open answer: the idea explained simply in the learner's own words, with one correct fact.

5. Stretch: What happens if you put the steps of a sequence in the wrong order?

Open answer: a clear explanation using one or two facts from the module.

Worksheet, Level 2 Game Builder (suggested 8+) (26 items)

1. **Code puzzles:** 1. `part.Position = Vector3.new(4, 2, 0);` 2. 8; 3. `task.wait(0.5);` 4. 12; 5. `part.Position` (capital P)

2. **Order:** 1. Start; 2. Goal; 3. Plan; 4. Code; 5. Wait; 6. Test

3. Case study

1. Why did Tayo only see the last position?

The lines ran too fast to see.

2. What should he add?

`task.wait(1)` between each move.

3. Write the fixed code for the first two moves.

part.Position = Vector3.new(0,2,0) task.wait(1) part.Position = Vector3.new(0,4,0)

4. Key terms: Sequence = Instructions that run one after another, in order.; Run = To make the computer follow the code.; Motion block = A block that moves or turns a sprite.; Position = Where something is, given as numbers.; Stud = The unit of length in Roblox.

5. Think it through

1. Explain why order matters in a sequence.

Each step builds on the last, so changing the order changes the result.

2. Explain what X, Y and Z mean in Roblox.

X is side to side, Y is up and down, Z is forward and back.

6. Longer answers and 7. Extended answer

1. Describe how to animate a part rising slowly.

Change its Y position a little at a time with a short wait between each change.

2. What is a stud?

The unit of length in Roblox.

3. Explain why Lua is called case sensitive.

Capital and small letters count as different, so Position is not position.

4. Explain in your own words: Why do we sometimes add a wait between moves?

Open answer: an accurate explanation in the learner's own words, with supporting detail.

5. Extended answer: 'A sequence is the most important idea in coding.' How far do you agree?

Open answer: an introduction, points for and against with examples and numbers from the module, and a reasoned conclusion.

Quiz

1. A sequence is...

A. Steps in order

2. To move a sprite we use...

B. Motion blocks

3. In Roblox, Y is...

B. Up and down

4. Roblox measures length in...

B. Studs

5. task.wait(2) waits for...

B. 2 seconds

6. Why add waits?

B. So we can see each step

7. Write the moves to get a sprite 2 squares right and 4 squares up.

Model answer: Move right 2, move up 4 (or up 4 then right 2).

8. Describe a sequence you follow at bedtime.

Model answer: For example: bath, pyjamas, brush teeth, story, sleep.

Checkpoints

Open answers. Look for the learner explaining the idea in their own words, with at least one accurate fact or number from the lesson.

Module 3: Loops

Worksheet, Level 1 Beginner (suggested 7+) (21 items)

1. **Code puzzles:** 1. 3; 2. 4; 3. forever; 4. 4
2. **Order:** 1. Spot; 2. Count; 3. Wait; 4. Test
3. **True or false:** 1. True; 2. False; 3. True; 4. False
4. **Match up:** Loop = B; Repeat = D; Forever = A; Counter = C
5. **Short answers**

1. How many times does 'repeat 5' repeat?
5 times.
2. Draw the shape made by repeat 4: move, turn.
A square.
3. Name something that goes round and round.
Any: a wheel, a clock, a washing machine.
4. Explain in your own words: Explain what a loop is, in your own words.
Open answer: the idea explained simply in the learner's own words, with one correct fact.
5. Stretch: Why is a loop better than copying the same block many times?
Open answer: a clear explanation using one or two facts from the module.

Worksheet, Level 2 Game Builder (suggested 8+) (25 items)

1. **Code puzzles:** 1. 2 3 4 5 6; 2. 10; 3. for i = 1, 3 do print("Hi") end; 4. end; 5. down from 10 to 1
2. **Order:** 1. Spot; 2. Count; 3. Wrap; 4. Wait; 5. Test
3. **Case study**
 1. Why did Studio freeze?
The forever loop had no wait, so it never let the game rest.
 2. What line should he add inside the loop?
task.wait(0.05) or similar.

3. Why does a small wait still look smooth?

It runs about 20 times a second, fast enough to look like smooth spinning.

4. Key terms: Loop = Code that repeats.; Repeat = A loop that runs a set number of times.; Forever = A loop that never stops on its own.; Counter = A number that counts each time round a loop.; for loop = A Lua loop that counts from one number to another.

5. Think it through

1. Explain the difference between a for loop and a while true loop.

A for loop counts a set number of times; while true repeats forever.

2. Explain what the counter i does.

It holds the number of the current turn of the loop and changes each time.

6. Longer answers and 7. Extended answer

1. Describe how to draw a triangle with a repeat loop.

Repeat 3: move forward, turn 120 degrees.

2. Write a loop that counts in fives to 25.

for i = 5, 25, 5 do print(i) end

3. Why is copying code many times a bad idea?

It is long, easy to get wrong, and hard to change later.

4. Explain in your own words: Why must a forever loop in Lua have a wait inside?

Open answer: an accurate explanation in the learner's own words, with supporting detail.

5. Extended answer: 'Loops are the coder's best friend.' How far do you agree?

Open answer: an introduction, points for and against with examples and numbers from the module, and a reasoned conclusion.

Quiz

1. A loop...

A. Repeats code

2. Repeat 4 with move and turn makes a...

B. Square

3. A forever block stops when...

B. You press stop

4. for i = 1, 3 do print(i) end prints...

A. 1 2 3

5. Which word closes a Lua loop?

C. end

6. A while true loop needs...

B. A task.wait()

7. Write a loop that makes a sprite jump 3 times.

Model answer: Repeat 3: jump (or for i = 1, 3 do jump() end).

8. Name two things in real life that loop.

Model answer: For example: the seasons, a washing machine, a song chorus, day and night.

Checkpoints

Open answers. Look for the learner explaining the idea in their own words, with at least one accurate fact or number from the lesson.

Module 4: Making choices

Worksheet, Level 1 Beginner (suggested 7+) (21 items)

1. **Code puzzles:** 1. yes; 2. no; 3. false; 4. stop
2. **Order:** 1. Question; 2. Yes; 3. Code; 4. Test both
3. **True or false:** 1. True; 2. False; 3. True; 4. False
4. **Match up:** Condition = B; True = D; False = A; If = C
5. **Short answers**

1. Finish: If it is dark, then...

Turn on the light (or similar).

2. Colour the light that means stop.

Red.

3. Make an if rule for a cat sprite.

Any: if touching the fish then say yum.

4. Explain in your own words: Explain what a condition is, in your own words.

Open answer: the idea explained simply in the learner's own words, with one correct fact.

5. Stretch: What does else do?

Open answer: a clear explanation using one or two facts from the module.

Worksheet, Level 2 Game Builder (suggested 8+) (25 items)

1. **Code puzzles:** 1. false; 2. if coins == 5 then print("Shop open") end; 3. if score == 10 then;
4. B; 5. elseif

2. **Order:** 1. Question; 2. Yes; 3. No; 4. Code; 5. Test both

3. **Case study**

1. What is wrong?

He used = (set) instead of == (compare).

2. Write the fixed line.

if score == 10 then print("Level up") end

3. Would ≥ 10 be better? Why?

Yes, because the score might jump past exactly 10.

4. Key terms: Condition = A yes-or-no question in code.; True = The answer yes.; False = The answer no.; If = Runs code only when a condition is true.; Else = The plan B when the condition is false.

5. Think it through

1. Explain how elseif helps with a traffic light.

It checks each colour in order and runs only the first true choice.

2. Explain why tests should include the edge case score $= 10$.

Edge cases show whether you chose $>$ or $\geq$ correctly.

6. Longer answers and 7. Extended answer

1. Write an if / elseif / else for a grade: 8+ gold, 5+ silver, else bronze.

if $s \geq 8$ then gold elseif $s \geq 5$ then silver else bronze end

2. What is a Boolean?

A value that is true or false.

3. Give two real machines that use conditions.

Automatic doors, thermostats, street lights.

4. Explain in your own words: What is the difference between $=$ and $==$ in Lua?

Open answer: an accurate explanation in the learner's own words, with supporting detail.

5. Extended answer: 'Choices are what make a program feel smart.' How far do you agree?

Open answer: an introduction, points for and against with examples and numbers from the module, and a reasoned conclusion.

Quiz

1. A condition has an answer that is...

B. True or false

2. If raining then umbrella. It is sunny. Do we take the umbrella?

B. No

3. Which Lua sign compares?

B. $==$

4. score = 5. if score >= 10 then print("Win") end prints...

B. Nothing

5. else runs when...

B. The condition is false

6. A traffic light needs...

A. elseif

7. Write an if rule for what to wear when it is cold.

Model answer: If it is cold then wear a coat, else wear a T-shirt.

8. Why must we test both a yes case and a no case?

Model answer: To check that both paths of the choice work.

Checkpoints

Open answers. Look for the learner explaining the idea in their own words, with at least one accurate fact or number from the lesson.

Module 5: Variables

Worksheet, Level 1 Beginner (suggested 7+) (21 items)

1. **Code puzzles:** 1. 3; 2. start; 3. 2; 4. box
2. **Order:** 1. Name; 2. Start value; 3. Use; 4. Check
3. **True or false:** 1. True; 2. False; 3. True; 4. False
4. **Match up:** Variable = B; Value = D; Set = A; Change = C
5. **Short answers**

1. Draw a box labelled score with 5 inside.
A box labelled score showing 5.
2. What would you call a variable for coins?
coins.
3. Score is 4. Catch 2 stars. What is the score?
6.
4. Explain in your own words: Explain what a variable is, in your own words.
Open answer: the idea explained simply in the learner's own words, with one correct fact.
5. Stretch: Why should we set a variable at the start of a game?
Open answer: a clear explanation using one or two facts from the module.

Worksheet, Level 2 Game Builder (suggested 8+) (25 items)

1. **Code puzzles:** 1. 10; 2. Hi Tayo; 3. local jumpPower = 50; 4. local myScore = 0; 5. 10
2. **Order:** 1. Name; 2. Start value; 3. Change; 4. Use; 5. Check
3. **Case study**
 1. Why does it reset to 0 each time?
The variable is made fresh inside the function every time it runs.
 2. Where should local coins = 0 go?
Outside, at the top of the script, so it is made once.

3. What line adds one coin?

coins = coins + 1

4. Key terms: Variable = A named box that stores a value.; Value = What is stored inside a variable.; Set = Put a new value in a variable.; Change = Add to or take away from a variable.; String = Words or letters inside quote marks.

5. Think it through

1. Explain `coins = coins + 1` in plain words.

Take the old coins value, add one, and store it back as the new coins.

2. Explain the difference between a number and a string.

A number can be used in maths; a string is text inside quotes.

6. Longer answers and 7. Extended answer

1. Describe how a game keeps score using a variable.

Set score to 0 at the start and add to it each time the player earns a point.

2. Write Lua to store a player's name and print a welcome.

local name = "Ama" print("Welcome " .. name)

3. Why does camel case exist?

Names cannot contain spaces, so capitals show where new words begin.

4. Explain in your own words: What makes a good variable name?

Open answer: an accurate explanation in the learner's own words, with supporting detail.

5. Extended answer: 'A game without variables would be boring.' How far do you agree?

Open answer: an introduction, points for and against with examples and numbers from the module, and a reasoned conclusion.

Quiz

1. A variable is like...

A. A named box

2. Set score to 0, then change by 2 twice. Score is...

B. 4

3. `local lives = 3` makes...

B. A variable

4. "Tayo" in quotes is a...

B. String

5. Best name for a score?

B. score

6. coins = 10, coins = coins - 3. Coins is...

B. 7

7. Name three variables a racing game might need.

Model answer: For example: speed, lap, timeLeft.

8. Why is 'x' a poor name for the player's lives?

Model answer: It does not tell anyone what is stored in it.

Checkpoints

Open answers. Look for the learner explaining the idea in their own words, with at least one accurate fact or number from the lesson.

Module 6: Events

Worksheet, Level 1 Beginner (suggested 7+) (21 items)

1. Code puzzles: 1. Hi; 2. when; 3. event; 4. yellow

2. Order: 1. Event; 2. Reaction; 3. Connect; 4. Test

3. True or false: 1. True; 2. False; 3. True; 4. False

4. Match up: Event = B; When block = D; Touched = A; Function = C

5. Short answers

1. What starts when you press the green flag?

The code under the green flag block.

2. Draw what a sprite does when tapped.

Open: for example a cat saying Hello.

3. Name an event at home.

Any: doorbell, light switch, kettle click.

4. Explain in your own words: Explain what an event is, in your own words.

Open answer: the idea explained simply in the learner's own words, with one correct fact.

5. Stretch: Why do programs wait for events?

Open answer: a clear explanation using one or two facts from the module.

Worksheet, Level 2 Game Builder (suggested 8+) (25 items)

1. Code puzzles: 1. Touched; 2. `lava.Touched:Connect(onTouch)`; 3. function (local function name()); 4. `part.Touched:Connect(onTouch)` (use a colon); 5. As many times as you like

2. Order: 1. Event; 2. Reaction; 3. Function; 4. Connect; 5. Test

3. Case study

1. Why did the event fire many times?

Several body parts touched the coin, each firing Touched.

2. What idea stops this?

A debounce: a true/false variable that blocks repeats.

3. What else could he do after one touch?

Destroy the coin or hide it so it cannot be touched again.

4. Key terms: Event = Something that happens, like a tap or touch.; When block = A hat block that starts code when an event happens.; Touched = The Roblox event for when a part is touched.; Function = A named group of code you can run again and again.; Connect = Join an event to a function.

5. Think it through

1. Explain the difference between an event and a function.

An event is something happening; a function is the code that runs in response.

2. Explain why functions save time.

You write the code once and run it many times.

6. Longer answers and 7. Extended answer

1. Describe how to make a door open when a button is touched.

Connect the button's Touched event to a function that moves or hides the door.

2. Write a function called sayHi that prints Hi.

local function sayHi() print("Hi") end

3. Why might hit.Parent be useful?

It tells you which character touched the part.

4. Explain in your own words: What does Connect do in Roblox?

Open answer: an accurate explanation in the learner's own words, with supporting detail.

5. Extended answer: 'Every game is just a lot of events.' How far do you agree?

Open answer: an introduction, points for and against with examples and numbers from the module, and a reasoned conclusion.

Quiz

1. An event is...

A. Something that happens

2. Which block starts code when the flag is clicked?

B. when green flag clicked

3. part.Touched fires when...

B. Something touches the part

4. A function is...

A. A named group of code

5. Connect joins...

B. An event and a function

6. A doorbell press is like a...

B. Event

7. List three events in a game you know.

Model answer: For example: pressing jump, touching a coin, the timer running out.

8. Describe what should happen when a player touches lava.

Model answer: They lose a life or go back to the start.

Checkpoints

Open answers. Look for the learner explaining the idea in their own words, with at least one accurate fact or number from the lesson.

Module 7: Debugging

Worksheet, Level 1 Beginner (suggested 7+) (21 items)

1. **Code puzzles:** 1. right; 2. bug; 3. debugging; 4. test
2. **Order:** 1. Notice; 2. Guess; 3. Test; 4. Repeat
3. **True or false:** 1. False; 2. True; 3. True; 4. False
4. **Match up:** Bug = B; Debugging = D; Error message = A; Line number = C
5. **Short answers**

1. Circle the wrong block: move left (fish is right).

The move left block.

2. What do we do after we fix a bug?

Test again.

3. How can you feel when you find a bug?

Open: proud, happy, like a detective.

4. Explain in your own words: Explain what a bug is, in your own words.

Open answer: the idea explained simply in the learner's own words, with one correct fact.

5. Stretch: Why should you change one thing at a time when debugging?

Open answer: a clear explanation using one or two facts from the module.

Worksheet, Level 2 Game Builder (suggested 8+) (26 items)

1. **Code puzzles:** 1. Missing end; 2. = should be ==; 3. Missing closing quote; 4. local must be lower case; 5. Missing closing bracket

2. **Order:** 1. Notice; 2. Read; 3. Guess; 4. Fix; 5. Test; 6. Repeat

3. Case study

1. What is the bug?

> 5 needs more than 5; he has exactly 5.

2. How could print help?

print(coins) would show 5, so he would see the test was false.

3. Write the fixed condition.

if coins >= 5 then

4. Key terms: Bug = A mistake in code.; Debugging = Finding and fixing bugs.; Error message = A note from the computer saying what went wrong.; Line number = Which line of code has the problem.; Test = Run the code to check it works.

5. Think it through

1. Explain rubber duck debugging.

Explaining your code line by line out loud often helps you spot the mistake.

2. Explain why line numbers in error messages are useful.

They show where to look first, saving time.

6. Longer answers and 7. Extended answer

1. Describe how to use print to find a bug.

Print values at key points and see where they stop being what you expect.

2. What is the difference between an error and a bug that gives no error?

An error stops the code with a message; a silent bug runs but gives the wrong result.

3. Why change one thing at a time?

So you know which change fixed or caused the problem.

4. Explain in your own words: How do error messages help a coder?

Open answer: an accurate explanation in the learner's own words, with supporting detail.

5. Extended answer: 'Debugging is more important than writing new code.' How far do you agree?

Open answer: an introduction, points for and against with examples and numbers from the module, and a reasoned conclusion.

Quiz

1. A bug is...

B. A mistake in code

2. Debugging means...

B. Finding and fixing mistakes

3. Expected 'end' means...

A. You forgot to close a block

4. print helps us...

A. See values while code runs

5. Good debuggers change...

B. One thing at a time

6. Do experts make bugs?

B. Yes, all coders do

7. Describe the steps you use to debug.

Model answer: Notice the problem, read the code or error, guess where it is, change one thing, test again.

8. Why are bugs a good thing to learn from?

Model answer: They teach you how code really works and help you avoid the same mistake later.

Checkpoints

Open answers. Look for the learner explaining the idea in their own words, with at least one accurate fact or number from the lesson.

Module 8: Build a game

Worksheet, Level 1 Beginner (suggested 7+) (21 items)

1. Code puzzles: 1. loop; 2. 1; 3. keys; 4. plan

2. Order: 1. Plan; 2. Code; 3. Test; 4. Share

3. True or false: 1. True; 2. False; 3. True; 4. False

4. Match up: Game design = B; Obby = D; Play-test = A; Anchored = C

5. Short answers

1. Draw your own game idea.

Open drawing of a game.

2. How do you win Catch the Fruit?

Catch enough fruit to reach the target score.

3. Who could play-test your game?

Any: a parent, brother, sister or friend.

4. Explain in your own words: Explain why coders plan a game first.

Open answer: the idea explained simply in the learner's own words, with one correct fact.

5. Stretch: Which coding ideas does Catch the Fruit use?

Open answer: a clear explanation using one or two facts from the module.

Worksheet, Level 2 Game Builder (suggested 8+) (26 items)

1. Code puzzles: 1. Anchored; 2. h.Health = 0; 3. Touched; 4. finish.Touched:Connect(function() print("You win!") end); 5. hit.Parent:FindFirstChild("Humanoid")

2. Order: 1. Plan; 2. Build; 3. Code; 4. Score; 5. Test; 6. Share

3. Case study

1. What does the play-test tell Tayo?

The third jump is too hard and is spoiling the fun.

2. How could he fix it?

Make the gap shorter, or add a checkpoint before it.

3. Why should he listen to feedback?

Players show problems the maker cannot see.

4. Key terms: Game design = Planning how a game works and feels.; Obby = An obstacle course game in Roblox.; Play-test = Letting someone try your game to find problems.; Anchored = A Roblox setting that stops a part falling.; Humanoid = The part of a Roblox character that has health.

5. Think it through

1. Explain how a lava part uses events and conditions together.

Touched is the event; checking for a Humanoid is the condition before setting health to 0.

2. Explain what iterate means for a game designer.

Improve the game in small rounds of testing and fixing.

6. Longer answers and 7. Extended answer

1. Describe the full plan for a mini obby.

Goal, platforms with gaps, lava that resets, checkpoints, finish that says you win.

2. Which variables might an obby need?

Stage, time, deaths, coins.

3. How would you make the obby harder in level 2?

Smaller platforms, moving parts, longer gaps, a timer.

4. Explain in your own words: Why is play-testing important?

Open answer: an accurate explanation in the learner's own words, with supporting detail.

5. Extended answer: 'A game is never really finished.' How far do you agree?

Open answer: an introduction, points for and against with examples and numbers from the module, and a reasoned conclusion.

Quiz

1. What should you do first?

A. Plan

2. The falling fruit uses a...

A. Forever loop

3. Score is stored in a...

A. Variable

4. An obby is...

A. An obstacle course

5. Anchored stops a part from...

A. Falling

6. After play-testing you...

B. Fix and improve

7. Plan your own game: goal, player, rules, how to win.

Model answer: Open answer that includes all four parts.

8. Name four coding ideas from this course and where they are used in a game.

Model answer: Sequences for moves, loops for falling fruit, ifs for catching, variables for score, events for key presses.

Checkpoints

Open answers. Look for the learner explaining the idea in their own words, with at least one accurate fact or number from the lesson.