COSC 123
Computer Creativity

Decisions and Loops

Dr. Ramon Lawrence
University of British Columbia Okanagan
ramon.lawrence@ubc.ca

Key Points

1) The **`If/Else`** statement is used to make decisions.

2) A decision requires a condition that consists of relational operators and Boolean functions.

3) A set of statements can be executed multiple times using **`While`** and **`Loop`** statements.

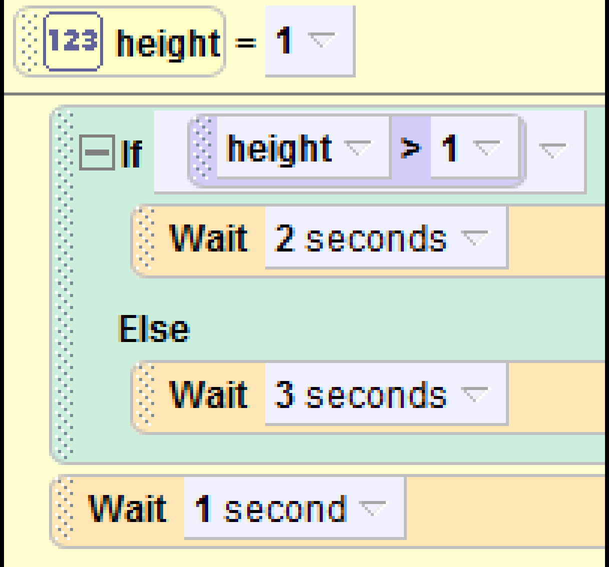
The `If/Else` Statement

Decisions are used to allow the program to perform different actions in certain conditions.

To make a decision we must do two things:

- 1) Determine the ***condition*** in which to make the decision.
- 2) Tell the computer what to do if the condition is true or false.

Example:



```
123 height = 1
If height > 1
    Wait 2 seconds
Else
    Wait 3 seconds
Wait 1 second
```

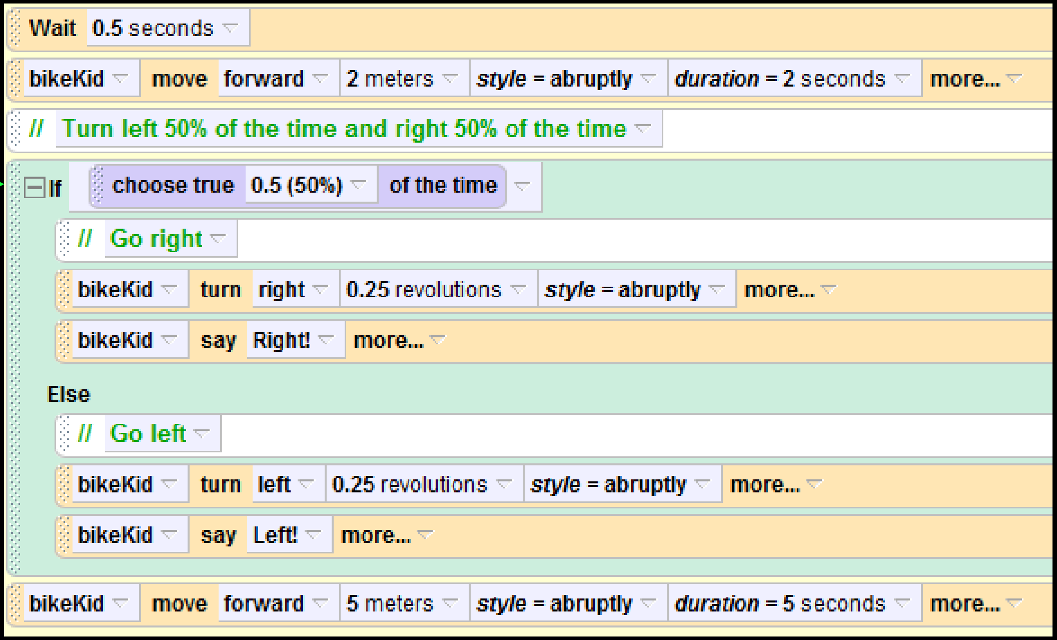
Condition to check: True or False?

Do this if condition is **true**.

Do this if condition is **false**.

Do statement after if in **either case**.

Example `If/Else` Statement Left or Right?



```
Wait 0.5 seconds
bikeKid move forward 2 meters style = abruptly duration = 2 seconds more...
// Turn left 50% of the time and right 50% of the time
If choose true 0.5 (50%) of the time
    // Go right
    bikeKid turn right 0.25 revolutions style = abruptly more...
    bikeKid say Right! more...
Else
    // Go left
    bikeKid turn left 0.25 revolutions style = abruptly more...
    bikeKid say Left! more...
bikeKid move forward 5 meters style = abruptly duration = 5 seconds more...
```

Condition →

Go right statements {

Go left statements {

Demonstration Exercise Decisions

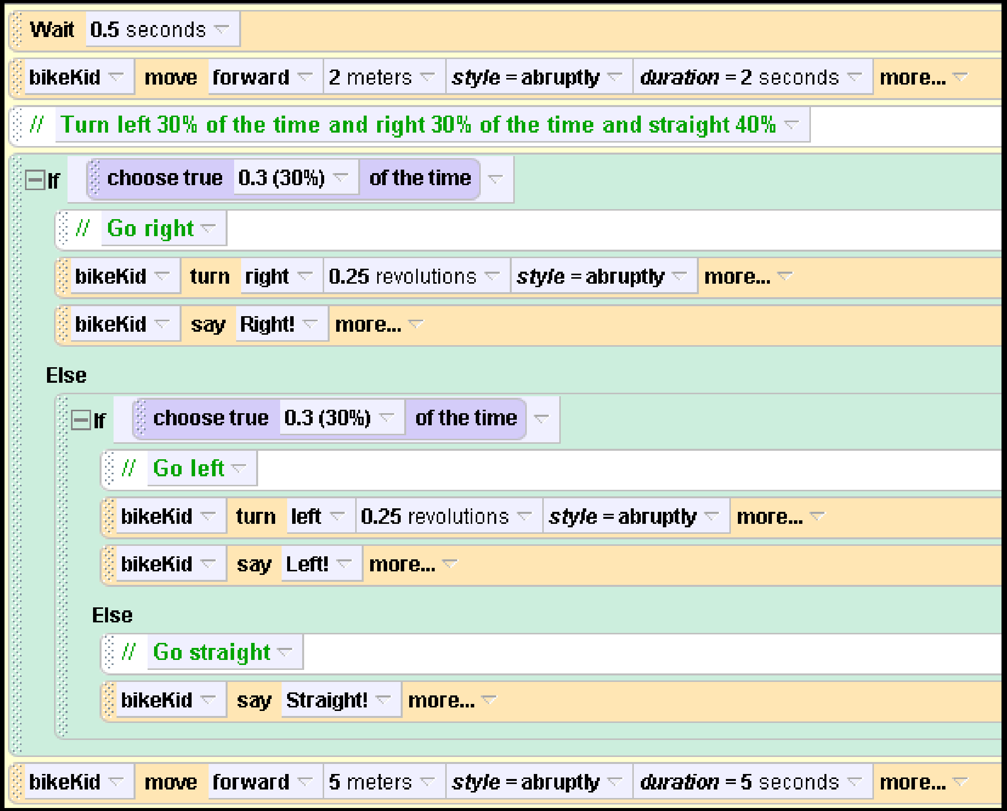
Use **`intersection.a2w`**.

Tasks:

- Play the animation.
- Modify so that the biker turns left 90% of the time.
- Modify so that the bike turn is smoother by moving forward and turning right at the same time.
- Modify so that the biker plays says "Hello" regardless of which direction turned.

Nested `If/Else` Statements More Than Two Possibilities

An `If/Else` statement could contain another ***nested*** `If/Else` statement.



```
Wait 0.5 seconds
bikeKid move forward 2 meters style = abruptly duration = 2 seconds more...
// Turn left 30% of the time and right 30% of the time and straight 40%
If choose true 0.3 (30%) of the time
    // Go right
    bikeKid turn right 0.25 revolutions style = abruptly more...
    bikeKid say Right! more...
Else
    If choose true 0.3 (30%) of the time
        // Go left
        bikeKid turn left 0.25 revolutions style = abruptly more...
        bikeKid say Left! more...
    Else
        // Go straight
        bikeKid say Straight! more...
bikeKid move forward 5 meters style = abruptly duration = 5 seconds more...
```

Nested if statement

Demonstration Exercise Nested Decisions

Use **`intersection2.a2w`**.

Tasks:

- Play the animation.
- Modify so that if the bike goes left he says "whoo hoo" while at the same time spinning around once.
- Modify so that there is a 50% of turning back around if the decision was to go straight.
- Add comments to say what each block of code in your if/else statements does.

Relational Operators

Relational operators are used to compare numeric data:

◆>	- Greater than
◆>=	- Greater than or equal
◆<	- Less than
◆<=	- Less than or equal
◆==	- Equal
◆!=	- Not equal

The Logical Operators

Operators:

`both` **X** `and` **Y** - true if X and Y are true, false otherwise
`not` **X** - true if X is false
`either` **X** `or` **Y** - true if either X or Y or both are true

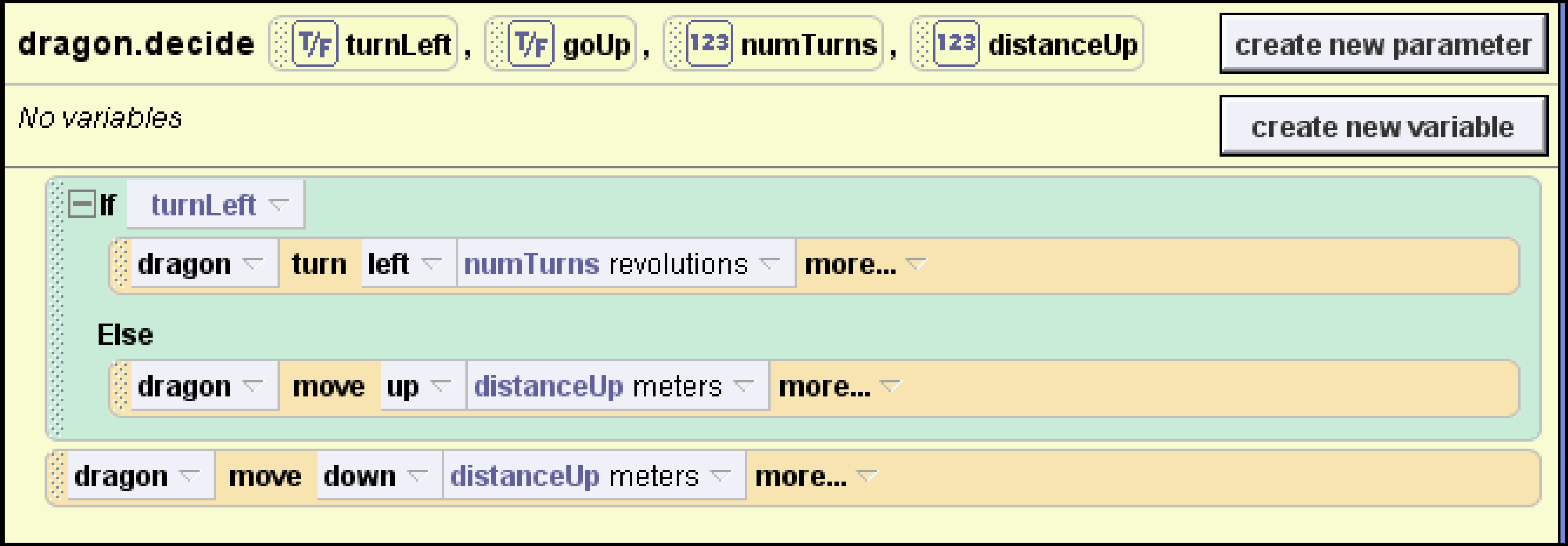
Examples:

- not *(This is COSC 123)*
- both *(This is COSC 123)* AND *(My name is Joe Smith)*
- either *(This is COSC 123)* OR *(My name is Joe Smith)*

Decisions

Question: What is the result of this code if:

- turnLeft = true ; goUp = false ; numTurns = 5 ; distanceUp = 7



```
dragon.decide  T/F turnLeft ,  T/F goUp ,  123 numTurns ,  123 distanceUp
No variables
If turnLeft
    dragon turn left numTurns revolutions more...
Else
    dragon move up distanceUp meters more...
dragon move down distanceUp meters more...
```

A) The dragon goes up 7 meters then down 7 meters.

B) The dragon turns around 5 times.

C) The dragon turns around 5 times and then down 7 meters.

Relational Operators

Question: True or false: a is true and b is false. What is **`both a and b`**?

A) true

B) false

Relational Operators (2)

Question: True or false: a is true and b is false.

What is **`either a or b`**?

A) true

B) false

Decision Exercises

Exercise #1: Turning Boat - Create a water world with a boat.

- Make the boat turn one half turn to the right if a random number between 1 and 100 is even otherwise one half turn to the left if the random number is odd.
- All your code can be in the my first method.

Exercise #2: Turning Zamboni - Create a world with a zamboni.

- Create a method called `turn` for the zamboni that has a parameter called **num**.
- In the `turn` method, decide if **num** is between 50 and 100 inclusive. If it is, turn the zamboni around.
- Test your code with four method calls in my first method with the values 75, 50, 25, 150. Go forward 10 meters before each call.

Repetition using Loops

A ***loop*** allows the programmer to repeat statements.

There are two loops in Alice:

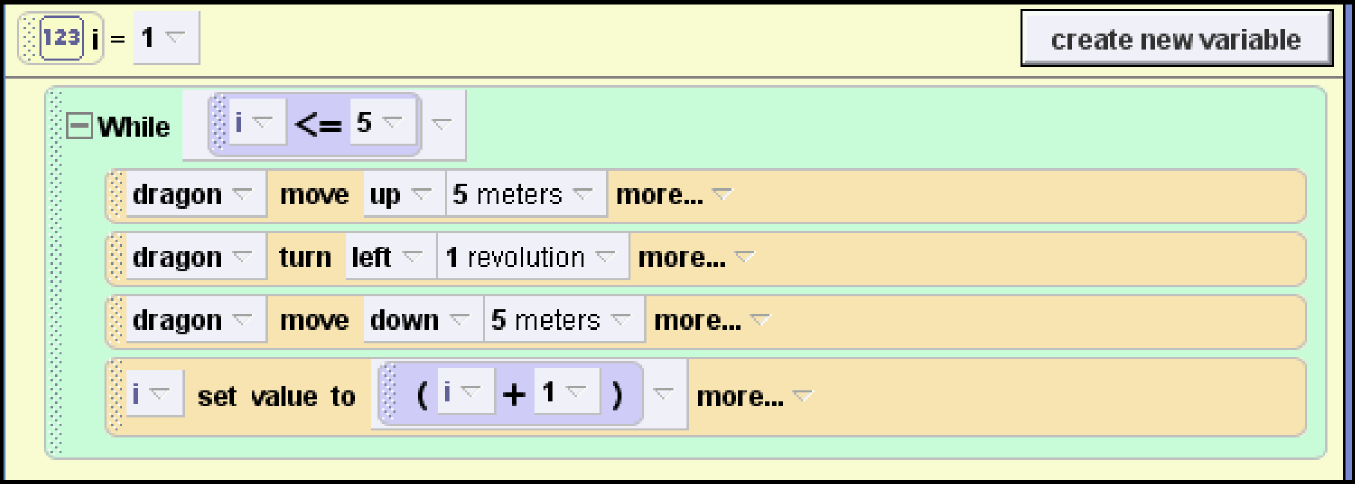
- the **`While`** statement
- the **`Loop`** statement

The `While` Statement

A **`While`** statement executes the statements it contains as long as its condition remains true.

- The condition is checked at the start of the loop and at the start of every loop ***iteration***.
- An infinite loop is a loop whose condition never becomes false, and the loop never ends.

Example:



```
123 i = 1
While   i <= 5
    dragon  move  up  5 meters  more...
    dragon  turn  left  1 revolution  more...
    dragon  move  down  5 meters  more...
    i  set value to  ( i + 1 )  more...
```

Demonstration Exercise While Loop

Use `Collision.a2w`.

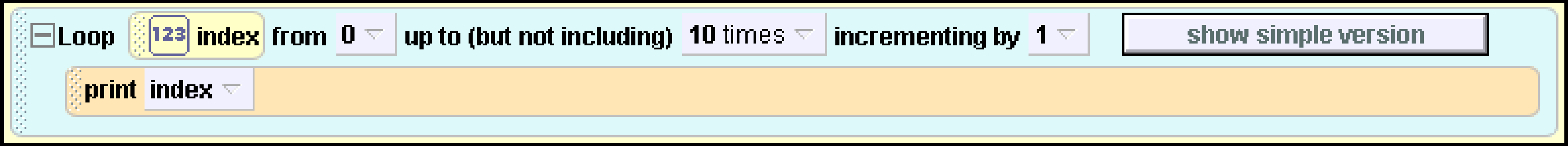
Tasks:

- Play the animation.
- Modify so that the vehicles and camera move twice as fast.
- Modify so that when trucks collide the cement truck says ouch!.
- Modify so that the cement truck's wheel breaks loose and continues to roll down the street.

The Loop Statement

The **Loop** statement allows you to control the exact number of repetitions.

The **Loop** statement uses a condition that tests the value of an integer counter variable and stops when its value reaches a specified end value.



```
Loop 123 index from 0 up to (but not including) 10 times incrementing by 1    show simple version
    print index
```

Demonstration Exercise Decisions

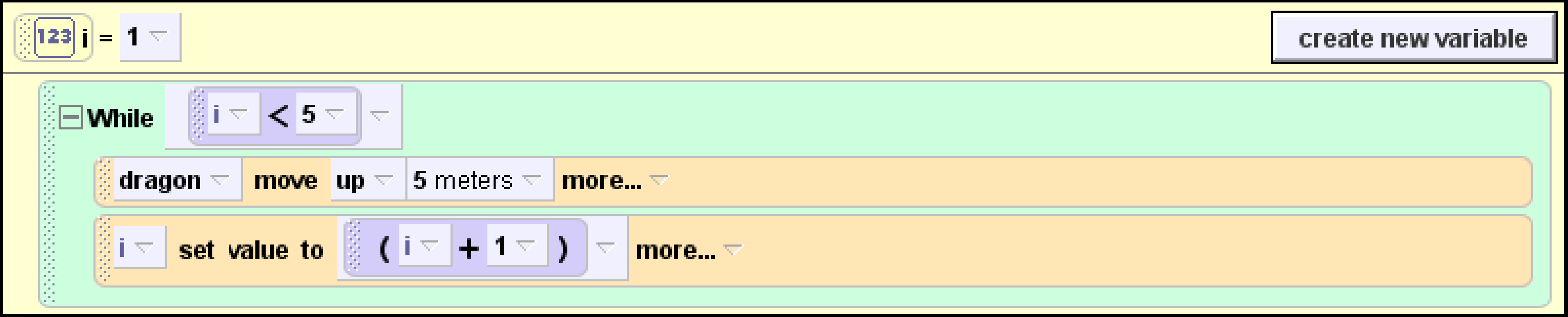
Use `SpeedingCar.a2w`.

Tasks:

- Play the animation.
- Modify so that the it calculates the total distance the car travels during its trip. Print out the result when the car comes to a stop.
- Modify so that the distance is updated while the car is moving. Display the result as a 3D text object in the window.
- Modify so that the speed is also continuously updated.

`While` Statement

Question: How many times does this **`While`** statement execute?



```
123 i = 1                                          create new variable
While i < 5
    dragon move up 5 meters more...
    i set value to ( i + 1 ) more...
```

A) 0

B) 3

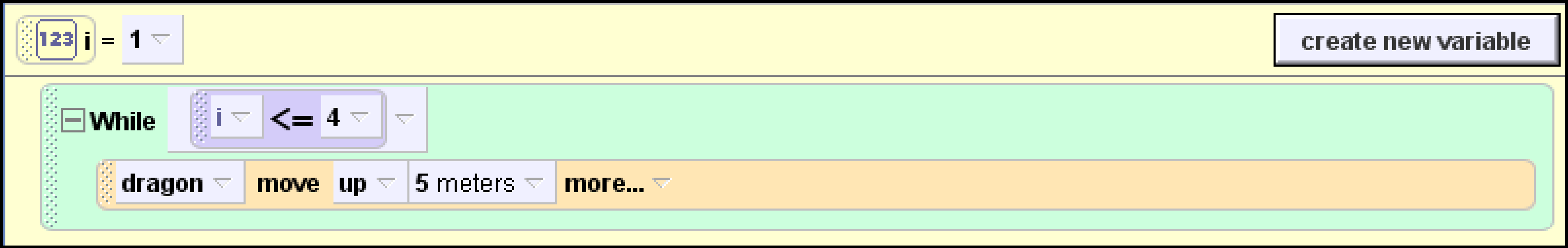
C) 4

D) 5

E) 6

While Statement (2)

Question: How many times does this **While** statement execute?



```
123 i = 1
While i <= 4
    dragon move up 5 meters more...
```

A) 0

B) 3

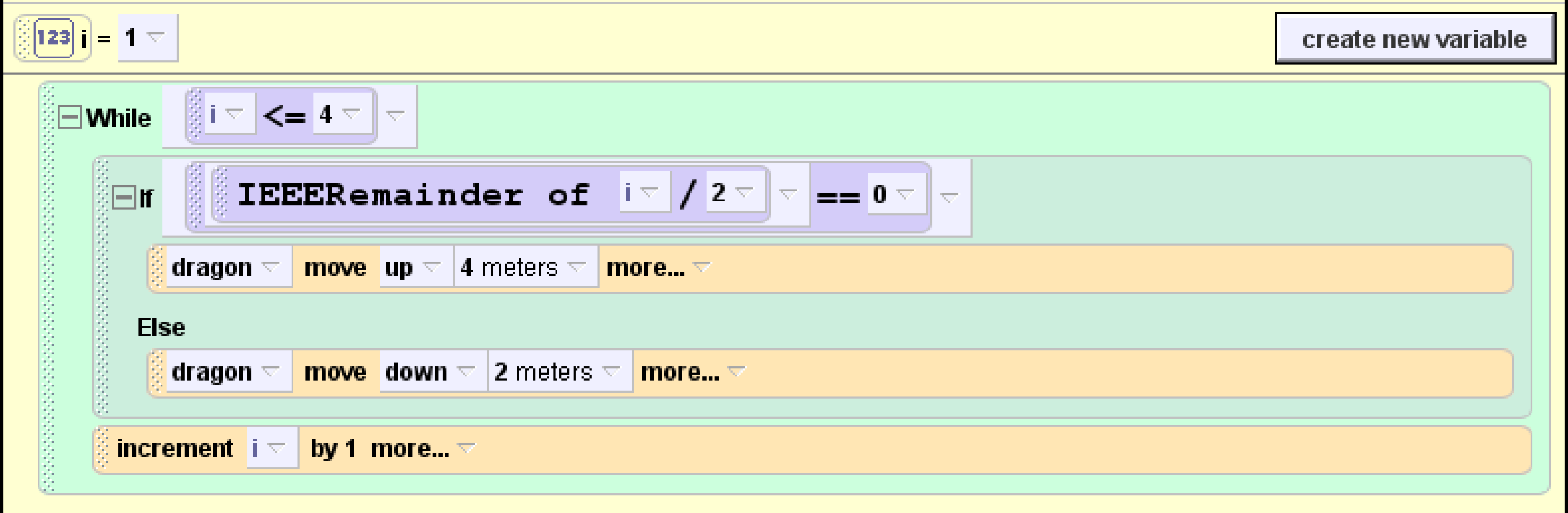
C) 4

D) 5

E) infinite

`While` Statement and Decisions

Question: How far off the ground is the dragon?

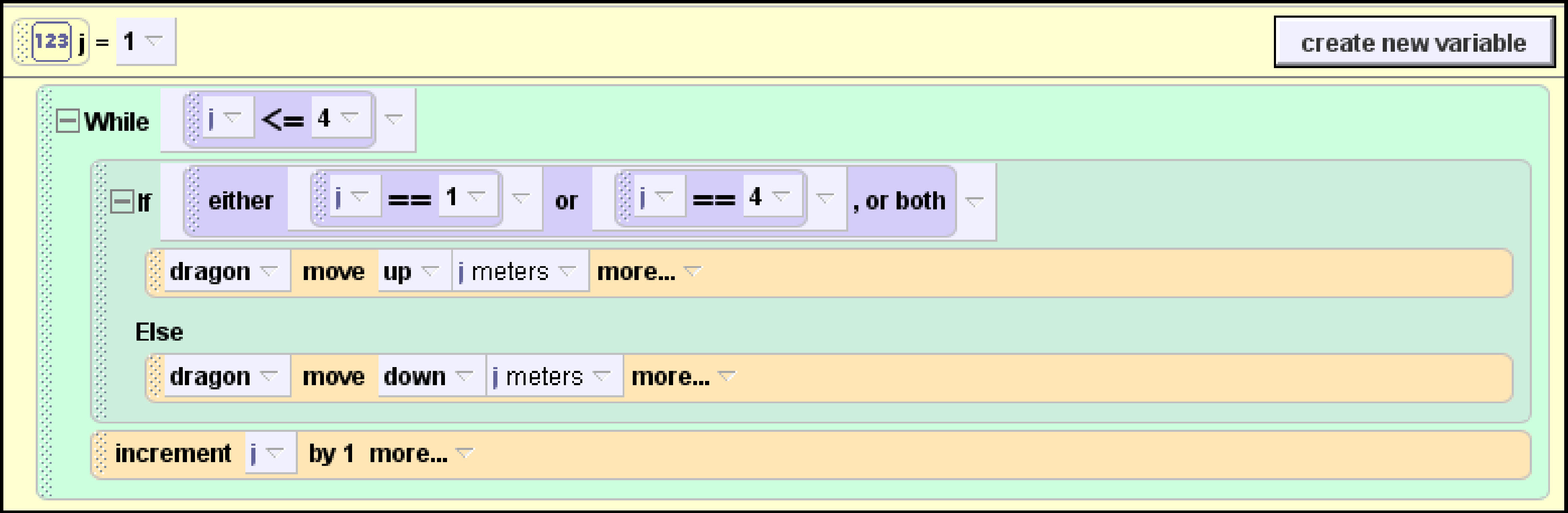


```
123 i = 1
While  i <= 4
    If  IEEERemainder of  i / 2  == 0
        dragon  move  up  4 meters  more...
    Else
        dragon  move  down  2 meters  more...
    increment  i  by 1  more...
```

A) -2 **B)** 0 **C)** 2 **D)** 4 **E)** 8

While Statement and Decisions (2)

Question: How far off the ground is the dragon?

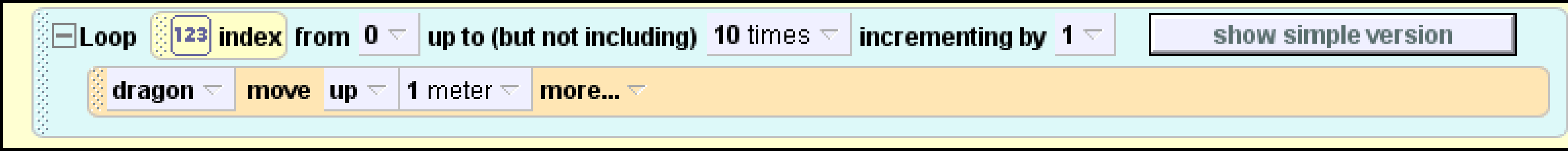


```
123 j = 1
While j <= 4
    If either j == 1 or j == 4 , or both
        dragon move up j meters more...
    Else
        dragon move down j meters more...
    increment j by 1 more...
```

A) -10 **B)** 0 **C)** 2 **D)** 4 **E)** 10

Loop Statement

Question: How many times does this loop execute?



```
Loop 123 index from 0 up to (but not including) 10 times incrementing by 1
    dragon move up 1 meter more...
```

A) 0

B) 9

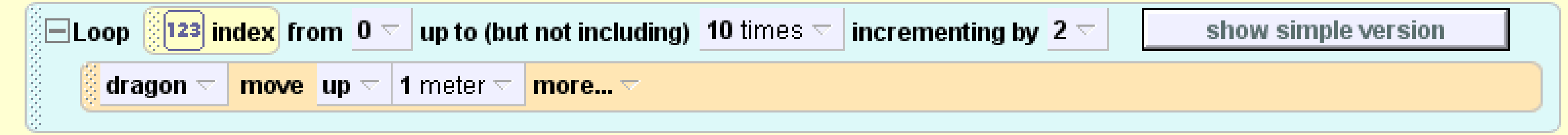
C) 10

D) 11

E) infinite

Loop Statement (2)

Question: How many times does this loop execute?



```
Loop 123 index from 0 up to (but not including) 10 times incrementing by 2
    dragon move up 1 meter more...
```

A) 0

B) 4

C) 5

D) 10

E) infinite

Exercises
Decisions and Loops

Exercise #1: Counting - Create a world with a 3D text object that counts from 1 to 10. Then counts down from 10 to 1.

Exercise #2: Jumping - Create a world where four characters perform jumps in unison five times. Each time give one of the characters a 30% chance to replace a jump with a full turn.

Exercise #3: Bouncing Ball - Create a world where a ball rolls off a table, bounces on the ground, and comes to rest. Decrease the height of the bounce by half each time. Move the ball away from the table slightly each bounce. Stop when the bounce height is small.

Conclusion

Decisions using the **`If/Else`** statement allow for controlling the flow of a program and decide when to execute certain statements.

Repetition of a block of statements can be done using:

- **`While`** statement that executes statements until its condition is false
- **`Loop`** statement that executes statements a specific number of times
- An infinite loop is a loop whose condition never becomes false (the loop never ends).

Decisions using **`If/Else`** statements and repetition using **`While/Loop`** statements can be nested.

Objectives

Key terms: decision, loop, condition

Alice skills:

- Making decisions with **`If/Else`**.
- Conditions: relational operators and Boolean (logical) operators.
- Nested **`If/Else`** decisions.
- Repetition using the **`While`** statement.
- Repetition using the **`Loop`** statement.
- Nested repetition statements.
- Using 3D Text boxes.