**LEVEL 1: Understanding the provided code blocks**

Think of this code like a set of instructions. In order for the game to play correctly, we need to make sure the instructions deliberately say what we would like to happen.

|  |  |
| --- | --- |
| 1 | This tells us when our code should start running. In this case, once the green flag has been clicked it will move on to the next step. |
| 2 | This directs our sprite (our astronaut dog) to the starting position on the lower left side of the screen |
| 3 | This directive informs the computer that we want it to follow all of the included instructions until our sprite (our astronaut dog) has reached position 168 on the x-axis, where the red flag on the screen is. Notice how this block encases block sets 4-6 and there is an arrow directing up at the bottom. |
| 4 | This is an if, else statement. We will ask the computer to do certain actions IF a certain input is found. If it is not (ELSE), another action will occur.  Notice that next to the if, there is a hexagonal block missing. **We will need to put a TM block here.** |
| 5 | When the computer recognizes the input in the if statement, our sprite will change his costume to dot-away. This is a version of him where he is facing away from us. It also has the option for the sprite to move a certain number of steps. **A number can be input into the white oval if we want the sprite to move in this case**. |
| 6 | When the computer doesn’t recognize the input in the if statement, it will move to this action. Our sprite will change his costume to dot-forward. This is a version of him where he is facing toward us. It also has the option for the sprite to move a certain number of steps. **A number can be input into the white oval if we want the sprite to move in this case.** |
| 7 | After the sprite has reached the red flag, it should say “Yay!” and then move 0 steps (stop). |

Graphical user interface, text, application, chat or text message

Description automatically generated

7

6

5

4

3

2

1

**LEVEL 2: Understanding the provided code blocks**

Think of this code like a set of instructions. In order for the game to play correctly, we need to make sure the instructions deliberately say what we would like to happen.

|  |  |
| --- | --- |
| 1 | This tells us when our code should start running. In this case, once the green flag has been clicked it will move on to the next step. |
| 2 | This directs our sprite (our astronaut dog) to the starting position on the lower left side of the screen |
| 3 | This directive informs the computer that we want it to follow all of the included instructions until our sprite (our astronaut dog) has reached position 168 on the x-axis, where the red flag on the screen is. Notice how this block encases block sets 4-6 and there is an arrow directing up at the bottom. |
| 4 | This is an if, else statement. We will ask the computer to do certain actions IF a certain input is found. If it is not (ELSE), another action will occur.  Notice that next to the if, there is a hexagonal block missing. **We will need to put a TM block here.** |
| 5 | This action will make the sprite say “Yay!” and then move 0 stops (stop). |
| 6 | This action will make our sprite change his costume to dot-away. This is a version of him where he is facing away from us. It also has the option for the sprite to move a certain number of steps. **A number can be input into the white oval if we want the sprite to move in this case**. |
| 7 | This action will make our sprite change his costume to dot-forward. This is a version of him where he is facing toward us. It also has the option for the sprint to move a certain number of steps. **A number can be input into the white oval if we want the sprite to move in this case.** |

A picture containing application

Description automatically generated

2

1

3

4

Text, application, chat or text message

Description automatically generated

6

5

7

**LEVEL 3: Understanding the provided code blocks**

Think of this code like a set of instructions. In order for the game to play correctly, we need to make sure the instructions deliberately say what we would like to happen.

A picture containing graphical user interface

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | This tells us when our code should start running. In this case, once the green flag has been clicked will it move onto the next step. | 2 | This operator will allow us to compare a value to the one provided (168). We can compare our steps or our position on the x- or y-axis. |
| 3 | This will allow our sprite to move forward (positive numbers) or backward (negative numbers). | 4 | This will allow our sprite to move forward (positive numbers) or backward (negative numbers). |
| 5 | This will allow our sprite to move forward 0 steps or stop. | 6 | This directs our sprite (our astronaut dog) to the starting position on the lower left side of the screen. |
| 7 | This will allow us to pull the current position of our sprite on the x-axis. | 8 | This is an if, else statement. We will ask the computer to do certain actions IF a certain input is found. If it is not (ELSE), another action will occur.  Notice that next to the if, there is a hexagonal block missing. **We will need to put a TM block here.** |
| 9 | This directive informs the computer that we want it to follow all of the included instructions until a certain point. This could be until the sprite reaches a certain position on the screen or even when the training model recognizes a certain image. Notice how this block can encase other blocks or block sets and there is an arrow directing up at the bottom. | 10 | This action will make the sprite say “Yay!” |
| 11 | This action will make our sprite change his costume to dot-away. This is a version of him where he is facing away from us. | 12 | This action will make our sprite change his costume to dot-forward. This is a version of him where he is facing toward us. |

12

11

10

9

8

7

6

5

4

3

2

1