Name(s)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Period \_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
|  | **Activity Guide - Variables Unplugged** |  |

**The Activity**

You will work with your partner to connect variable names and their values based on different programs.

**Setup**

**▢ The Board:** A piece of paper split into 3 sections: Labels, Values, and Workspace

**▢ Variable Name and Value Cards:** Small pieces of paper

**Rules**

1. Name cards must always have one and only one connector
2. You may not create the same name card twice
3. You must create a new value card for every “=” command
4. Value cards can only have one number written on them.
5. Make sure you are looking at the ‘Values’ section on your Board whenever you have to recalculate a variable

**Commands**

Run the programs in order by following the steps for each command.

**Step 1: Find or create the name card and connector**

1. If the command starts with “var” (indicating a variable is being defined), **create a new** name card with the variable name. For example, if Line 1 says:  **var width = 10**, then you should create a name card with the name ‘**width**’. (This name card can be re-used).
2. Place the name card in the ‘Labels’ section of your Board.

**Step 2: Find or calculate the number for your value card**

1. To the right of each variable (after the “=”) is a value. If this value is already a single number, write this value on a new value card and place the value card in the ‘Values’ section of your Board. Place a connector between the variable and its value.
2. If the value of any variable must be calculated or re-calculated from a previous command, use the ‘Workspace’ area at the bottom of the Board to do the calculation before writing that number on a new value card and placing it in the ‘Values’ section of your Board. Replace the value cards as needed when variable values are recalculated.

**Step 3: Fill in the ending state of your program**

Once the program is finished, record the Ending State for all your variables.

**Try It Out!**

Run the programs below. To help you along in the first two programs, check out the before and after examples to the right which show how different commands run.



|  |  |  |
| --- | --- | --- |
| **Program 1** | |  |
| 01 | var width = 10 |
| 02 | var height = 20 + width |
| **Ending State** | |
| **width:** | |
| **height:** | |
|  |  |

|  |  |  |
| --- | --- | --- |
|  | |  |
| **Program 2** | |
| 01 | var size = 30 |
| 02 | size = 60 |
| 03 | size = size + 5 |
| **Ending State** | |
| **size:** | |
|  | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Program 3** | | | 01 | var age = 11; | | 02 | var height = 60; | | 03 | age = age + 1; | | 04 | height = height + 5; | | **Ending State** | | | **age:** | | | **height:** | | | |  |  | | --- | --- | | **Program 4** | | | 01 | var xPosition = 100 | | 02 | var yPosition = xPosition | | 03 | xPosition = yPosition + 30 | | 04 | yPosition = yPosition + 50 | | **Ending State** | | | **xPosition:** | | | **yPosition:** | | |

**Sprite Properties**

You can change your sprite properties in the same way that you change your variables, which allows you to control how the sprite moves across the screen.

To keep track of a sprite’s properties, you’ll need a sprite card. The sprite card goes in the values section, connected to a variable name. On the bottom of the card, list the properties that you want to keep track of. **You will use one connector for each property.**



In the example to the right, the card keeps track of the x and y properties of the sprite, which tell the computer where to place it on the screen.

## Creating a Sprite Card

Every time you see the `createSprite` command, you’ll need to create a new sprite card with the properties you want to keep track of.

## Using Sprite Properties



You can use your sprite property values the same way you used your variable values.

## Drawing Your Sprite

Whenever you see the `drawSprites` command, draw your sprite on the grid according to its x and y coordinate values.

You can use a smiley face for your sprite’s animation.

You can make your own sprite cards or cut out the ones at the end of this activity guide.

|  |  |  |
| --- | --- | --- |
|  |  | |
| **Program 5** | |
| 01 | var smiley = createSprite(100,200); |
| 02 | smiley.setAnimation(“smileyFace”); |
| 03 | drawSprites(); |
| 05 | smiley.x = smiley.x + 50; |
| 06 | drawSprites(); |
|  |  |

## Try it Out!

Run these programs with your partner. Don’t forget to draw your sprites when you see the `drawSprites` command.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Program 6** | | | 01 | var smiley = createSprite(); | | 02 | smiley.setAnimation(“smiley”); | | 03 | smiley.x = 50; | | 04 | smiley.y = 100; | | 05 | drawSprites(); | | 06 | smiley.x = smiley.x + 50; | | 07 | drawSprites(); | | 08 | smiley.x = smiley.x + 50; | | 09 | drawSprites(); | | 10 | smiley.x = smiley.x + 50; | | 11 | drawSprites(); | | **Ending State** | | | **smiley.x** | | | **smiley.y** | | | How did the sprite move across the grid in Program 6?  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Program 7** | | | 01 | var smiley = createSprite(); | | 02 | smiley.setAnimation(“smiley”); | | 03 | smiley.x = 200; | | 04 | smiley.y = 300; | | 05 | drawSprites(); | | 06 | smiley.y = smiley.y - 30; | | 07 | drawSprites(); | | 08 | smiley.y = smiley.y - 30; | | 09 | drawSprites(); | | 10 | smiley.y = smiley.y - 30; | | 11 | drawSprites(); | | **Ending State** | | | **smiley.x** | | | **smiley.y** | | | How did the sprite move across the grid in Program 7?  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

------------------------------------------------- Cut out the sprite cards below ------------------------------------------------------------

