**WEEK 3 – Flying Carpet Game**

**Unit:** Spatial Sense

**Grade:** Intermediate (7-8)

**Curriculum Expectation**   
describe and perform translations… on a Cartesian plane

**SEL:** see themselves as capable math learners, and strengthen their sense of ownership of their learning, as part of their emerging sense of identity and belonging

**Activity  
1)** Students will use Scratch for this activity to make a game more fun through ‘tinkering’ with the code. Go to: scratch.mit.edu/projects/179382052/ or scratch.mit.edu and search MissLauren and find *Magic Carpet Game* **2)** Play the game according to the instructions. It should be very easy. Come up with a list of things that make the game easy. Now how could you change the game to make it more difficult and more fun?   
**3)** Click *See Inside* to see the code of the game. See the Scratch cheat sheet down below to understand what the code is saying.   
**4)** Students will change aspects of the code of the game to make it more difficult   
**5)** When done, try using other features such as changing the Sprite’s costume, colour, background etc.   
**Note:** If computer access is unavailable, students can run an unplugged version of this game by using tape or string to plot a Cartesian plane on the ground and use the unplugged code (found below) to have their ‘sprite’ move to set location on the Cartesian Plan to catch a ball rolled to that location to collect points. The important part is that students are interacting with all four quadrants of the cartesian plane. See activity sheet below for instructions and example.

**Check for Understanding**   
I understand how placement of Sprites occurs on the Cartesian plane in the Scratch game   
I understand the importance of each piece of ‘code’  
I am able to understand how to change code to make the game function differently

**Materials**   
Recording sheet (attached below), access to Scratch or space to program unplugged ‘sprites’

Unplugged Coding: This activity can be done to compliment or replace the online Scratch activity if access to a computer is not available. Students will plot out a cartesian plane on the ground from -5 to 5 on the x-axis and y-axis. Students will draw or stack these blocks to create their own ‘code’ to program their sprite to move. Once the sprite has arrived at a location, you will determine the coordinates which the ball should be rolled to for the sprite to catch. All four quadrants of the cartesian plane should be used. An operator must be connected to a specific movement to make that movement occur.

Commands for Sprite

Ball Sprite Action

Movement

Operators

Roll Ball to (\_\_\_, \_\_\_)

Change x by -1

When I say ‘Down’

When I say ‘Left’

When I say ‘Right’

When I say ‘Up’

Change x by 1

Left

Down

Up

Right

Change y by 1

Change y by -1

**Example**

Sprite starts at (0,0)

When I say ‘Right’

When I say ‘Up’

Change x by 1

Change y by 1



Up



Up



Right

Roll Ball to (4,3)

Right

Roll Ball to (1,2)

Up

Right





Right



|  |  |  |
| --- | --- | --- |
| What makes the game easy? (*hint*: think of the different characteristics (speed, size etc.) of each the bat, the carpet and the cloud) | How could you change this to make the game more difficult? | How can you adjust the code to make the game harder? |
|  |  |  |

**Scratch Introduction Cheat Sheet**

Scratch uses a Cartesian plane (ie. an x and y-axis) from approximately -250 to 250 on the x-axis (horizontal) and 250 to -250 on the y-axis (vertical).

A screenshot of a computer screen

Description automatically generatedAn event (the thing that has to happen to trigger something else to happen) must be connected to a movement for it to occur when the action (ie pressing the right button pushes moves the sprite right)

A screenshot of a computer screen

Description automatically generatedChange a sprites location by using the blue code block “change x/y by \_\_\_”

A screenshot of a computer screen

Description automatically generatedChanging the x value by a positive value moves the sprite right, changing by a negative value moves a sprite from left across the Cartesian plane. Making the value greater increases the speed

A screenshot of a computer screen

Description automatically generatedChanging the y value by a positive value moves the sprite up and changing by a negative value moves the sprite down the Cartesian plane. Making the value greater increases the speed

A screenshot of a computer screen

Description automatically generatedIf you are having difficulty understanding what the code means try writing it out in plain language to understand it better, you can do this by reading the code, or running code to see what it does

For example, this piece of the bat’s code can translate as:

-When I click the green flag

-Always keep doing until the code finishes

-Start at a random position on the y-axis

-Start at 250 on the x-axis (the far right of the Cartesian Plane)

-For 168 times do:

-move the sprite 3 spaces to the left (3x168=504-across   
the whole Cartesian plane

-repeat forever.

This code is telling the bat sprite to start at a random spot and then fly across the screen at a set speed until it reaches the far side, then repeat it.

Size can be changed through the ‘size’ area.

Direction the sprite is facing can be changed through the ‘direction’ area.