CoderDojo

Scratch Lesson Plan!

1. First we need to position Scratch. To make him fall we have to change the y direction, like gravity.

```
when [flag] clicked
go to x: 0 y: 0
change y by -7
```

2. To make him fall continuously we must add the forever command.

```
when [flag] clicked
go to x: 0 y: 0
forever
    change y by -7
```

3. To change Scratch's size add 'set size'

```
when [flag] clicked
go to x: 0 y: 0
set size to 45 %
forever
    change y by -7
```

4. Now that he's falling we need something for him to land on. First draw a box sprite like this

5. Now we have to add conditions for the colour of the box when Scratch touches it

```
when [flag] clicked
wait until touching color [blue] ?
```

6. Create a variable called 'Jump' (for all sprites) and untick the tickbox

7. Now set jump to 20

```
when [flag] clicked
wait until touching color [blue] ?
set Jump to 20
```

8. We must tell it what jump does, which is change Scratch in the y direction. As well we want 'Jump' to decrease as Scratch jumps on the same box.

```
when [flag] clicked
wait until touching color [blue] ?
set Jump to 20
change y by Jump
change Jump by -1
```

9. We want Scratch to keep jumping so we must add the forever command.

```
when [flag] clicked
forever
    wait until touching color [blue] ?
    set Jump to 20
    change y by Jump
    change Jump by -1
```

10. To repeat this action we must use repeat until so it can decrease jump over time

```
when [flag] clicked
forever
    wait until <touching color [ ]?>
    set Jump to 20
    repeat until <Jump = 0>
        change y by Jump
        change Jump by -1
```

11. We now have to name this action when Scratch touches a box

```
when [flag] clicked
forever
    wait until <touching color [ ]?>
    broadcast Sprites Down
    set Jump to 20
    repeat until <Jump = 0>
        change y by Jump
        change Jump by -1
```

12. Now we have Scratch moving up and down!! We now have to get him to move left and right. We need a variable called 'MoveLR'.

13. The value for MoveLR is 0.

```
when [flag] clicked
set MoveLR to 0
```

14. Because we want these commands to last for the whole game we need to add a forever command and tell it to move in the x direction.

```
when [flag] clicked
set MoveLR to 0
forever
    change x by MoveLR
```

15. Now we have to tell it what happens when the left key is pressed.

```
when [flag] clicked
set MoveLR to 0
forever
    change x by MoveLR
    if <key left arrow pressed?> then
    else
```

16. Limit the amount of movement

```
when [flag] clicked
set MoveLR to 0
forever
    change x by MoveLR
    if <key left arrow pressed?> then
        if <MoveLR > -10> then
    else
```

17. Every time we click the left arrow we want Scratch to move -1 in the x direction.

```
when [flag] clicked
set MoveLR to 0
forever
    change x by MoveLR
    if < key [left arrow] pressed? > then
        if < MoveLR > -10 > then
            change MoveLR by -1
    else
```

18. Now we have to tell it what happens when the right arrow is pressed.

```
when [flag] clicked
set MoveLR to 0
forever
    change x by MoveLR
    if < key [left arrow] pressed? > then
        if < MoveLR > -10 > then
            change MoveLR by -1
    else
        if < key [right arrow] pressed? > then
        else
```

19. Repeat the same as the left arrow, but now we are changing in the +x direction.

20. Now for an added challenge we want Scratch to drift to the right by himself so we will add a bit each time in the +x direction.

```
when [flag] clicked
set MoveLR to 0
forever
    change x by MoveLR
    if < key [left arrow] pressed? > then
        if < MoveLR > -10 > then
            change MoveLR by -1
    else
        if < key [right arrow] pressed? > then
            if < MoveLR < 10 > then
                change MoveLR by 1
        else
```

```
when [flag] clicked
set MoveLR to 0
forever
    change x by MoveLR
    if < key [left arrow] pressed? > then
        if < MoveLR > -10 > then
            change MoveLR by -1
    else
        if < key [right arrow] pressed? > then
            if < MoveLR < 10 > then
                change MoveLR by 1
        else
            set MoveLR to (MoveLR + 0.01)
```

21. To stop Scratch from hitting the edge we are going to send him to the left when he goes too far right.

```
when [flag] clicked
set MoveLR to 0
forever
    change x by MoveLR
    if < key [left arrow] pressed? > then
        if < MoveLR > -10 > then
            change MoveLR by -1
    else
        if < key [right arrow] pressed? > then
            if < MoveLR < 10 > then
                change MoveLR by 1
        else
            set MoveLR to (MoveLR + 0.01)
    if < x position > 240 > then
        set x to -240
```

22. And now the same when he goes too far left!

```
when [flag] clicked
set MoveLR to 0
forever
    change x by MoveLR
    if < key [left arrow] pressed? > then
        if < MoveLR > -10 > then
            change MoveLR by -1
    else
        if < key [right arrow] pressed? > then
            if < MoveLR < 10 > then
                change MoveLR by 1
        else
            set MoveLR to (MoveLR + 0.01)
    if < x position > 240 > then
        set x to -240
    if < x position < -240 > then
        set x to 240
```

23. Now we want scratch to disappear when he falls to the bottom so first we have to make him visible

```
when [flag] clicked
forever
    show
```

24. Where he will disappear will be when he goes past y=-180

```
when [flag] clicked
forever
    show
    if < y position < -180 > then
        hide
```

25. We want to call the point where scratch disappears 'Game Over'.

```
when [flag] clicked
forever
    show
    if < y position < -180 > then
        hide
        broadcast Game Over
```

26. Now we can create a sprite that tells us 'GAME OVER'

GAME OVER

27. Now with this sprite we can position and hide it until it is game over.

```
when [flag] clicked
go to x: 20 y: 0
hide
```

28. Once 'GAME OVER' is broadcasted we can show the sprite.

```
when I receive Game Over ▼
show
```

29. Now we want the box sprite to move down as Scratch bounces on it. We don't, however want it to move down if Scratch is too low, below -20y.

```
when I receive Sprites Down ▼
if  ( y position ▼ of Sprite1 ▼ ) > -20  then
```

30. Now to get it to move down we need to change the y direction and make it glide

```
when I receive Sprites Down ▼
if  ( y position ▼ of Sprite1 ▼ ) > -20  then
    glide 1 secs to x: ( x position ) y: ( y position - 10 )
```

31. Now we can look at the box and first position it.

```
when ⚑ clicked
show
go to x: 124 y: -133
```

32. Now we want to hide the box after it reaches a certain position, -170y

```
when ⚑ clicked
show
go to x: 124 y: -133
forever
    wait until ( y position < -170 )
```

33. Now add the 'hide'.

```
when ⚑ clicked
show
go to x: 124 y: -133
forever
    wait until ( y position < -170 )
    hide
```

34. We want the box go to the top of the screen, y= 180, but have a slightly different x position so we add a random number between -25 and 25 to the original position.

```
when ⚑ clicked
show
go to x: 124 y: -133
forever
    wait until ( y position < -170 )
    hide
    go to x: ( 124 + pick random -25 to 25 ) y: 180
```

35. When it is at this position we can 'show'.

```
when ⚑ clicked
show
go to x: 124 y: -133
forever
    wait until ( y position < -170 )
    hide
    go to x: ( 124 + pick random -25 to 25 ) y: 180
    show
```

36. We want the box to move slightly down from the top of the screen when it first appears so Scratch can jump on it straight away. For this we add a glide to a position -100 from the y 180.

```
when [flag] clicked
show
go to x: 124 y: -133
forever
    wait until < y position < -170 >
    hide
    go to x: 124 + pick random -25 to 25 y: 180
    show
    glide 2 secs to x: x position y: y position - 100
```

37. We can now duplicate the box, make 4 more and scatter them around the screen making sure to change the starting x and y position of all the boxes.

38. If we want to add a score variable, as in how many time Scratch has jumped, we must first create a variable 'Score'.

39. Set score to 0 at the beginning.

```
when [flag] clicked
set Score to 0
```

40. When Scratch touches a box, 'Sprites Down', change the score by 1.

```
when I receive Sprites Down
change Score by 1
```

41. To set a high score create another variable called 'Highscore'. When Highscore < Score we want to set Highscore to Score.

```
when [flag] clicked
forever
    if < Highscore < Score > then
        set Highscore to Score
```

42. As an added challenge, we can make a box slide to the left and right. First click on the box you want to move and add a forever command changing the x direction.

```
when [flag] clicked
forever
    change x by -1
```

43. Now we have to set the limit of the amount of spaces we want to move, I have picked 60 so we repeat moving the box 60 times.

```
when [flag] clicked
forever
    repeat 60
        change x by -1
```

44. Now we can do the same for the other direction and it will move back and forth!

45. Now you should have something that looks like this!