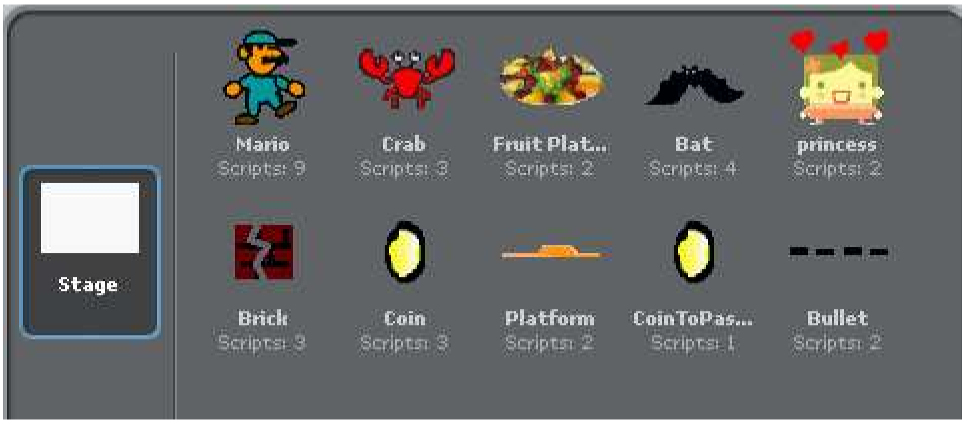
Scratch Programming Lesson 15

In this lesson, we will wrap up the Mini Mario game by making it scroll, among other things. Also, please go to http://shallwelearn.com to download MiniMarioLesson13.sb from Scratch Programming Lesson 13.

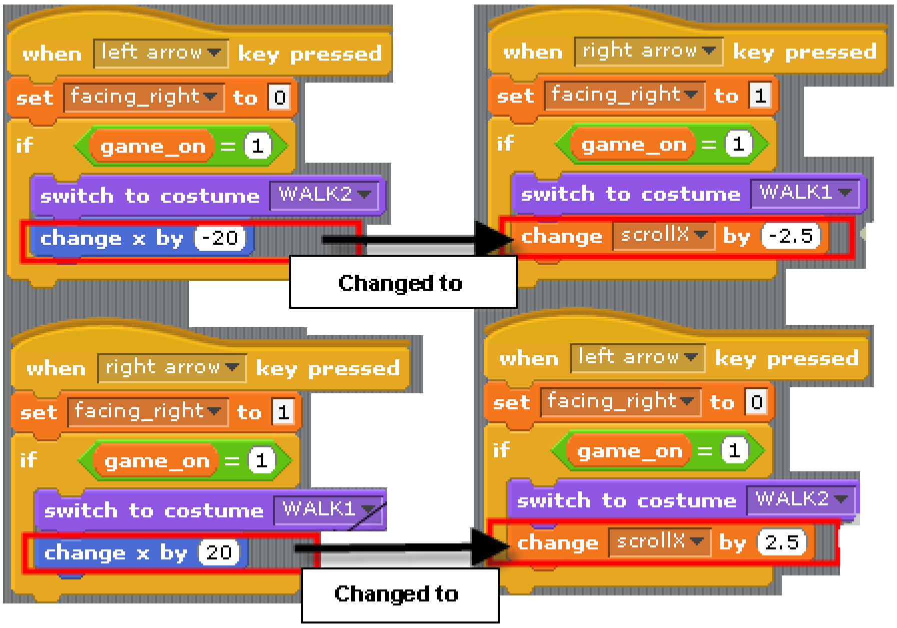
This is the snapshot of the sprite list when the project is opened.



Step 1: Create a Variable to Represent Scrolling Amount

Create a global variable called scroll to represent the amount of scrolling. When Mario moves to the right, the scrolling amount decreases, and when it moves to the left, the scrolling amount increases. If you are unfamiliar of how scrolling works in Scratch or in general, please refer to Shall We Learn Scratch Programming Lesson 14.

Delete “**change x by …**” and replace with “**chang scollX by -2.5**”.



```
when [left arrow] key pressed
set [facing_right] to 0
if <(game_on) = 1>
    switch to costume [WALK2]
    change x by -20
```

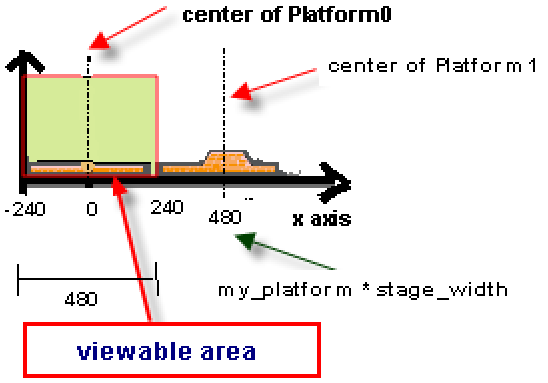
```
when [right arrow] key pressed
set [facing_right] to 1
if <(game_on) = 1>
    switch to costume [WALK1]
    change [scrollX] by -2.5
```

```
when [right arrow] key pressed
set [facing_right] to 1
if <(game_on) = 1>
    switch to costume [WALK1]
    change x by 20
```

```
when [left arrow] key pressed
set [facing_right] to 0
if <(game_on) = 1>
    switch to costume [WALK2]
    change [scrollX] by 2.5
```

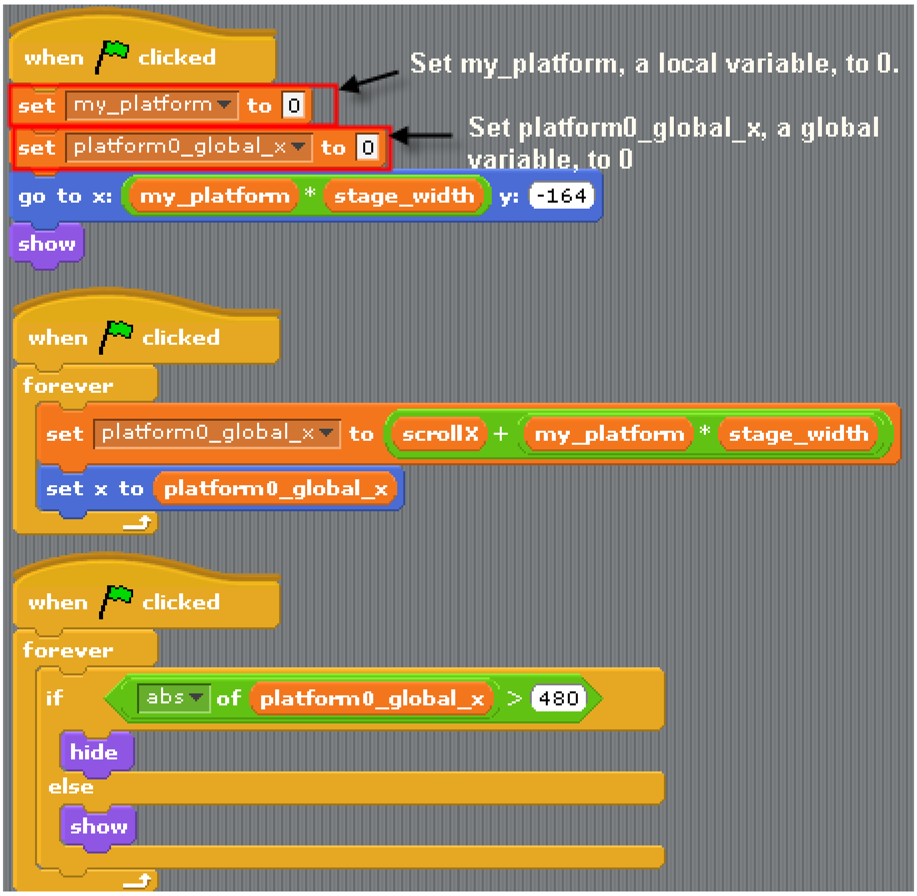
Step 2: Make Platforms Scroll

I am going to split **Platform** sprite into two sprites: **Platform0** and **Platform1**.
I will line these two platform sprites side-by-side. When the game starts, **Platform0** is located at x = 0 or 0 * the stage width (=480). Platform1 is at x = 480 or 1 * the stage width.



stage_width = 480
my_patform for Platform0 = 0
my_platform for Platform1 = 1

Rename **Platform** to **Platform0**. Delete all scripts from **Platform0** and add the following scripts, which are very similar to what we built in Lesson 14, except that I added a global variable call **platform0_global_x**.

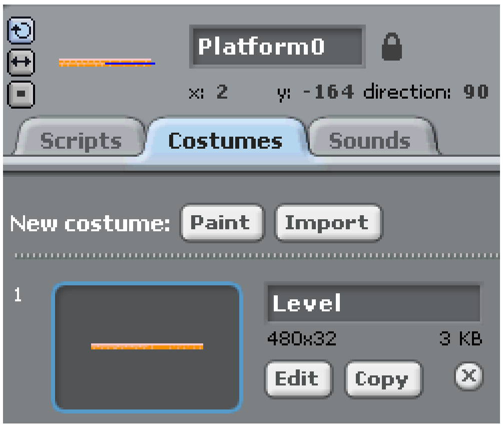


```
when 🏁 clicked
set my_platform to 0
set platform0_global_x to 0
go to x: (my_platform * stage_width) y: -164
show

when 🏁 clicked
forever
    set platform0_global_x to (scrollX + (my_platform * stage_width))
    set x to platform0_global_x

when 🏁 clicked
forever
    if ((abs of platform0_global_x) > 480)
        hide
    else
        show
```

Set my_platform, a local variable, to 0.

Set platform0_global_x, a global variable, to 0

Then make a copy of **Platform0** and name the copy as **Platform1**. Delete costume **Level2** from **Platform0** and rename costume **Level1** to **Level** and delete costume **Level1** from **Platform1** and rename costume **Level2** to **Level**.

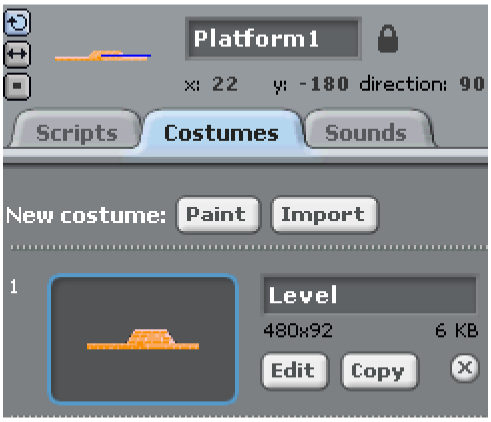


Platform0

x: 2 y: -164 direction: 90

1 Level

480x32 3 KB



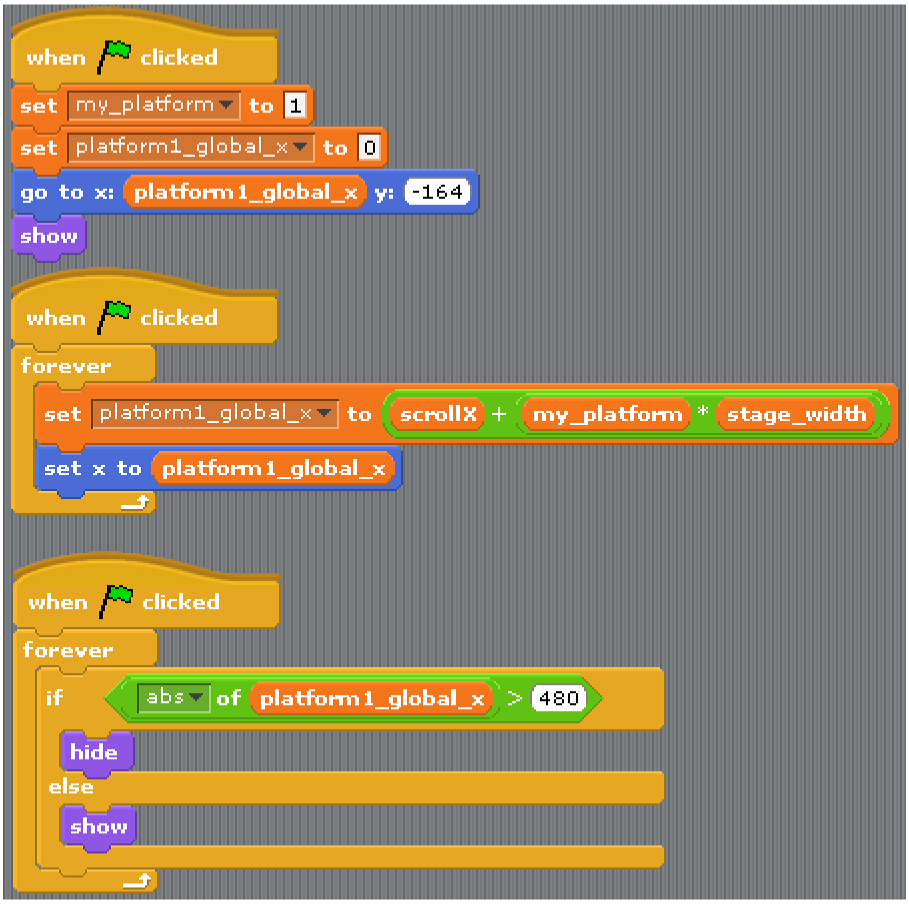
Platform1

x: 22 y: -180 direction: 90

1 Level

480x92 6 KB

Then, as shown below, for **Platform1**, set my_platform to 1 when game starts. Then create a new global variable called **platform1_global_x** and place it in the place of **platform0_global_x**.

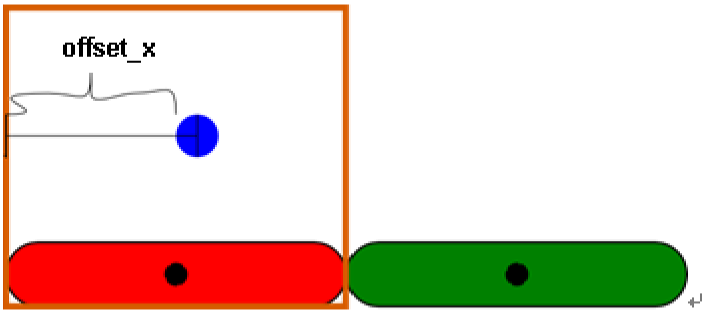


```
when clicked
set my_platform to 1
set platform1_global_x to 0
go to x: platform1_global_x y: -164
show

when clicked
forever
    set platform1_global_x to (scrollX + (my_platform * stage_width))
    set x to platform1_global_x

when clicked
forever
    if (abs of platform1_global_x) > 480
        hide
    else
        show
```

TEST TIME: Now test the game. You can scroll but there is an obvious problem. Both the brick and two coins keep following Mario so Mario could never get to them. We will fix it.

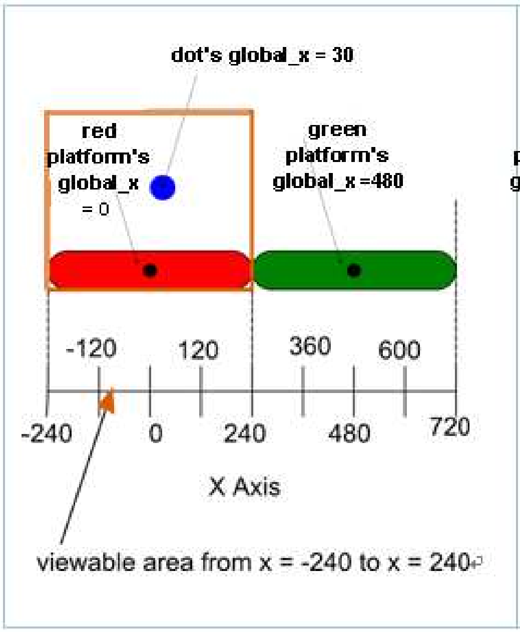
*Step 3: Modify the **Brick** and **CoinToPass** to Stop Them from Following **Mario***

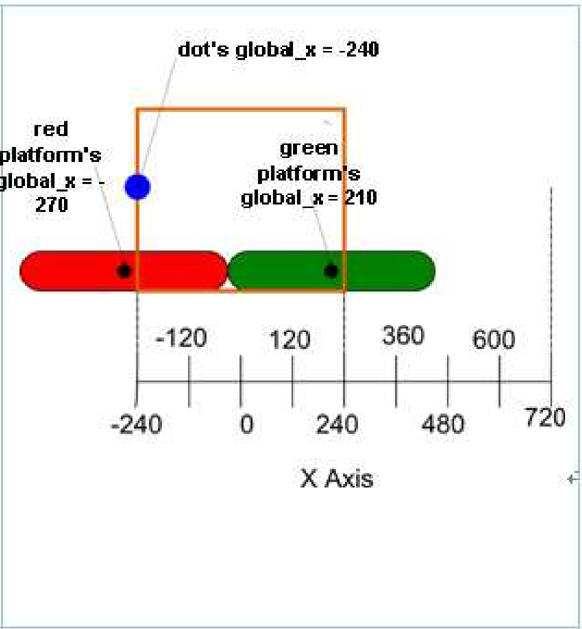
To keep the brick and coins from following **Mario**, we need to make them scroll, just like the platform sprites. We will copy the scripts pertaining to scrolling from platform sprites to both Brick and Coin sprite. But we need to make two changes. The first is to add a local variable call *offset_x*; this variable stores the distance from the left boundary of the platform to the sprite.



The location of the dot sprite will be

$$STARTING\_POSITION_{DOT} = PLATFORM\_INDEX * STAGE\_WIDTH + OFFSET\_X$$
$$POSITION_{DOT} = STARTING\_POSITION_{DOT} + SCROLL\_X_{PLATFORM}$$


dot's global_x = 30
red platform's global_x = 0
green platform's global_x =480
-120
120
360
600
-240
0
240
480
720
X Axis
viewable area from x = -240 to x = 240

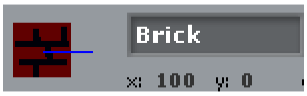

dot's global_x = -240
red platform's global_x = -270
green platform's global_x = 210
-120
120
360
600
-240
0
240
480
720
X Axis

The second change is to change the show/hide boundary from STAGE_WIDTH (=480) to STAGE_WIDTH/2 (=240). The graph below shows the snapshot of the stage before and after the scrolling to the right by 270 (scrollX = -270).

$$\begin{aligned}\text{STARTING\_POSITION}_{\text{DOT}} &= \text{STARTING\_ POSITION}_{\text{RED\_PLATFORM}} + \text{OFFSET\_X}\\ &= \text{PLATFORM\_INDEX} * \text{STAGE\_WIDTH} + \text{OFFSET\_X}\\ &= 0 * 480 + 30 = 30\end{aligned}$$

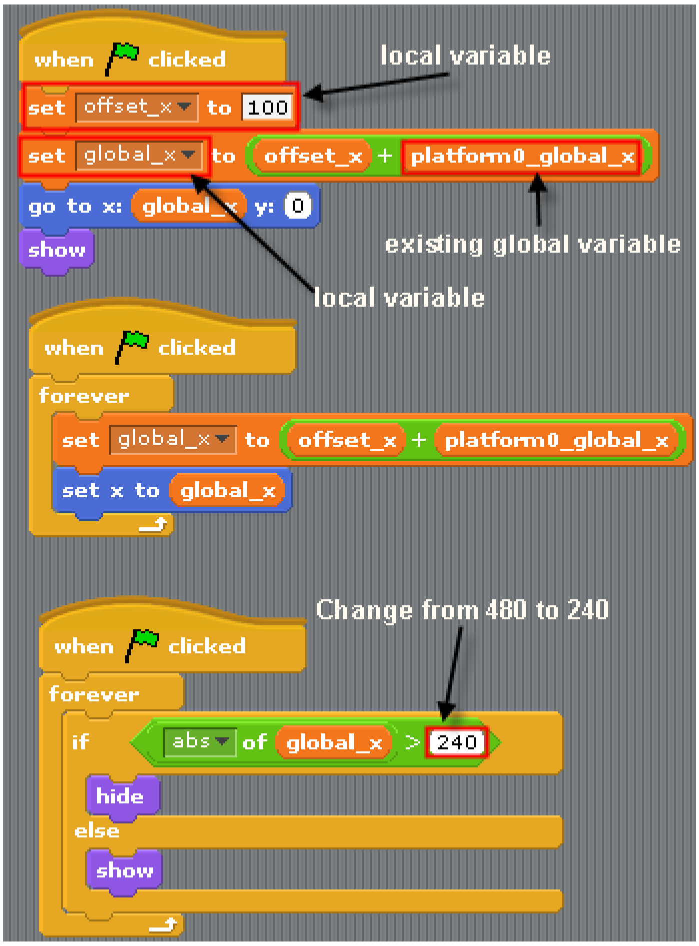
$$\begin{aligned}\text{POSITION}_{\text{DOT}} &= \text{POSITION}_{\text{RED\_PLATFORM}} + \text{OFFSET\_X}\\ &= \text{STARTING\_POSITION}_{\text{RED}} + \text{SCROLL\_X}_{\text{PLATFORM}} + \text{OFFSET\_X}\\ &= 0 * 480 + (-270) + 30 = 30\end{aligned}$$

The snapshot below shows **Brick's** script blocks relevant to scrolling.



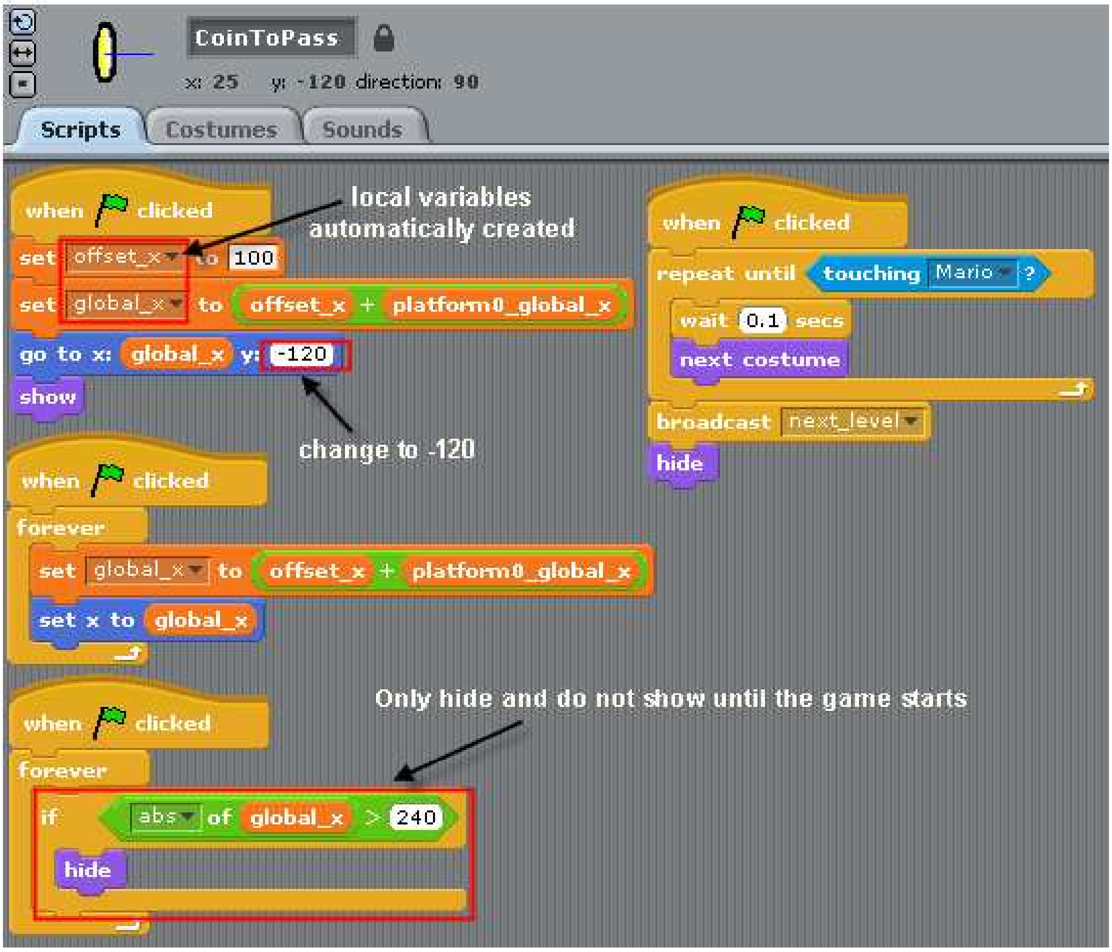
Brick

x: 100 y: 0



```
when ⚑ clicked
set offset_x to 100
set global_x to (offset_x + platform0_global_x)
go to x: global_x y: 0
show

when ⚑ clicked
forever
    set global_x to (offset_x + platform0_global_x)
    set x to global_x

when ⚑ clicked
forever
    if (abs of global_x) > 240
        hide
    else
        show
```

Next, we will change **CoinToPass** sprite. First remove all motion-related scripts and copy all scrolling related scripts from **Brick** to **CoinToPass**.



```
when [flag] clicked
set offset_x to 100
set global_x to (offset_x + platform0_global_x)
go to x: global_x y: -120
show

when [flag] clicked
forever
    set global_x to (offset_x + platform0_global_x)
    set x to global_x

when [flag] clicked
forever
    if abs of global_x > 240
        hide

when [flag] clicked
repeat until touching Mario ?
    wait 0.1 secs
    next costume
broadcast next_level
hide
```

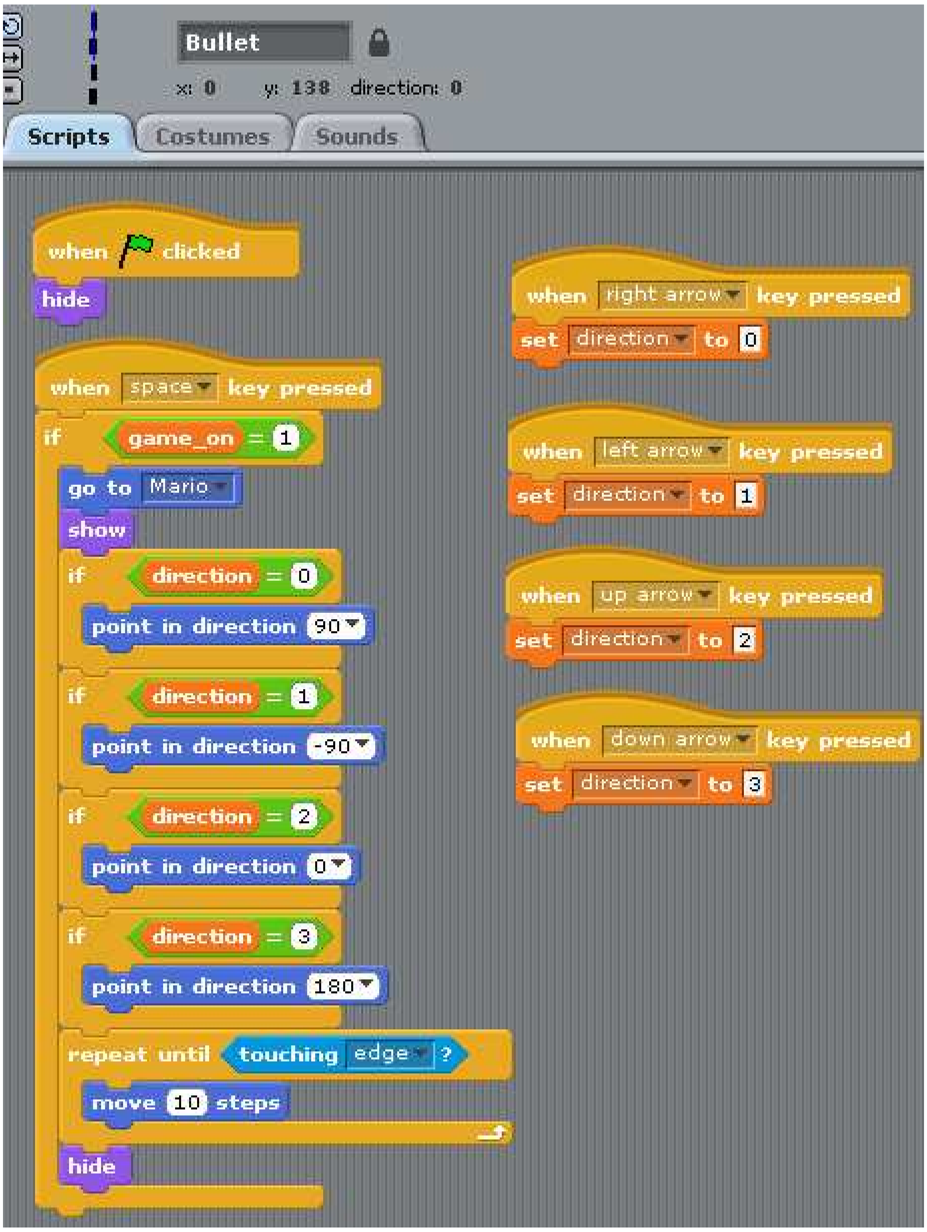
local variables automatically created

change to -120

Only hide and do not show until the game starts

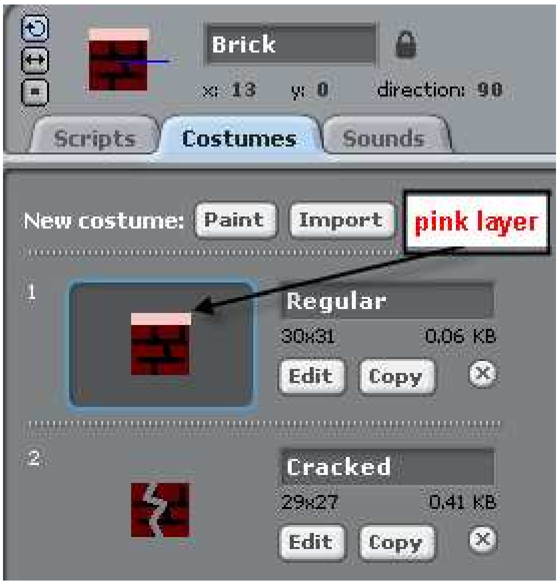
*Step 4: Upgrade the **Bullet** and Turn **Brick** to a Mini Platform*

If you would like, you can upgrade the **Bullet** so that you can shoot from all four directions. Below is the updated script for the **Bullet**.

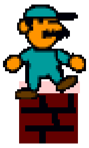


```
when [flag] clicked
hide

when space key pressed
if game_on = 1
    go to Mario
    show
    if direction = 0
        point in direction 90
    if direction = 1
        point in direction -90
    if direction = 2
        point in direction 0
    if direction = 3
        point in direction 180
    repeat until touching edge ?
        move 10 steps
    hide

when right arrow key pressed
set direction to 0

when left arrow key pressed
set direction to 1

when up arrow key pressed
set direction to 2

when down arrow key pressed
set direction to 3
```

To turn the **Brick** into a mini platform, just add a top pink layer (the same pink used to create the top layer of the **Platform0** and **Platform1** sprites).



Then **Mario** can stand on the **Brick** as well.



This concludes Lesson 15, the last lesson of Mini Mario Game series. I hope you have enjoyed this series and leave with enough know-how to make your own platform games!