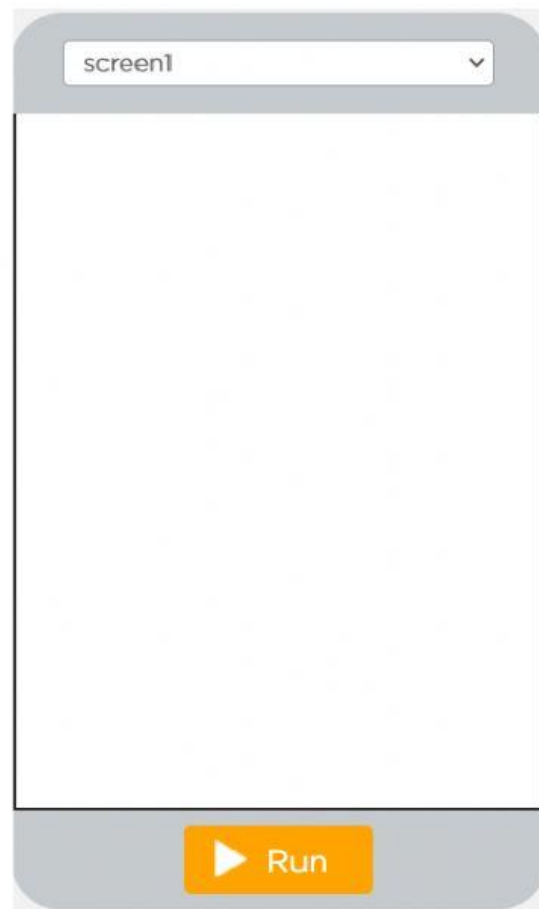


ඩී පී එඩුකේෂන් කෝඩින් පාසල කෝඩින් වෙසක් කලාපය - 2023



“x,y position” ඇසුරින් චලන රචා හඳුම



☒ Make my own

ආරම්භ කරන්න



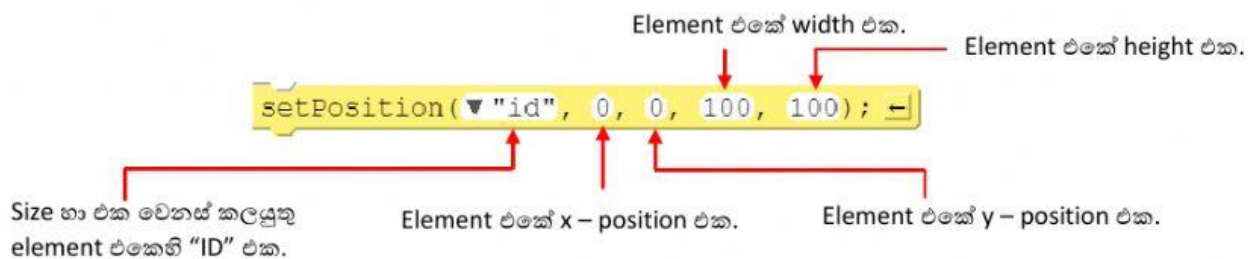
**Coding
School**

එකම image එකක x හෝ y, position එක වෙනස් කරමින්, ආලෝක රටාවක් නිර්මාණය කරන ආකාරය බලමු.

- ❖ මුලින් ම “image2forYposition.png” හා “image1.png” images add කරගෙන පහත පරිදි Properties සාදා ගන්න.

එකම image එකක x හෝ y, position එක වෙනස් කරමින්, ආලෝක රටාවක් නිර්මාණය කරන ආකාරය බලමු.

- ❖ Size හා position එක coding වලින් වෙනස් කිරීම සඳහා UI controls වල ඇති “setPosition()” block එක use කළ යුතුයි.



- ❖ imageA වල, y-position පමණක් වෙනස් කරමින් සිරස් ව පහලට ගමන් කරන ලෙසද, imageB වල x- position පමණක් වෙනස් කරමින් තිරස් ව දකුණේ සිට වමට ගමන් කරන ලෙස ද ආලෝක රටා නිර්මාණය කරන්න.

- ❖ Design වලදී imageA වල y-position එක 0px ලෙස ද, imageB වල x-position 270px ලෙස ද ලබා දී ඇත.

- ❖ imageA සිරස් ව පහලට ගමන් කිරීම එක් අවස්ථාවකදී y-position 40px කින් වෙනස් වන ලෙස ගනිමු.

- ❖ Screen එකෙහි height එක 450px බැවින්, imageA පහල ම අවස්ථාවේදී y-position 400px වේ. එමනිසා y-position එක 40px බැගින් වෙනස් කළ හැකි අවස්ථා ගණන 11කි.
- ❖ imageA ට ඇති අවස්ථා ගණන 11 ක් බැවින්, imageB ද අවස්ථා 11 ක දී තිරස් ව දකුණේ සිට වමට ගමන් කළ යුතුය.
- ❖ Screen එකහි width එක 320px බැවින් imageB වමට ගමන් කළ හැකි උපරිම අවස්ථාවේදී x-position එක 0px වේ. එමනිසා අවස්ථා 11 ක දී x-position එක වෙනස් විය යුතු බැවින් එක් අවස්ථාවක දී x-position එක 27px වලින් වෙනස් කරන්න.
- ❖ එම අවස්ථා වලදී පහත පරිදි position සකස් කරමු.

	imageA : y-position (px)	imageB : x-position (px)
1 වන අවස්ථාව	0	270
2 වන අවස්ථාව	40	243
3 වන අවස්ථාව	80	216
4 වන අවස්ථාව	120	189
5 වන අවස්ථාව	160	162
6 වන අවස්ථාව	200	135
7 වන අවස්ථාව	240	108
8 වන අවස්ථාව	280	81
9 වන අවස්ථාව	320	54
10 වන අවස්ථාව	260	27
11 වන අවස්ථාව	400	0

- ❖ දැන් “count”, “result” ලෙස variable දෙකක් නිර්මාණය කර අගය බිංදුව ලෙස ලබා දෙන්න.

```
var count = 0;
var result = 0;
```

- ❖ ඉන්පසු “timedLoop” ඇතුළේ “count” variable එකට 1ක් එකතු කරන්න. “timedLoop” එකට 100ms ලෙස ලබා දෙන්න.
- ❖ එවිට සෑම තත්පර භාගයක් තුළදී ම “timedLoop” එක call වෙයි. ඒ නිසා count variable එකෙහි අගය සෑම තත්පර භාගයක් තුළදී ම එකකින් වැඩි වෙයි. (0, 1, 2, 3, ...)
- ❖ මෙහිදී අවස්ථා 11 ක් නිසා count එක, 11 න් බෙදා ඉතිරිය result variable එකට assign කර ගන්න. ඕනෑම සංඛ්‍යාවක් 11 න් බෙදූ විට ඉතිරි වන්නේ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 යන සංඛ්‍යා අතරින් එකක් වේ.

- ❖ දැන් result variable එකෙහි අගය අනුව පහත පරිදි images වල position සකස් කරමින් බ්ලේබ් රටාව නිර්මාණය කරන්න. (සම්පූර්ණ code කොටස පහත පරිදි වේ.)

```

var count = 0;
var result = 0;
timedLoop(100, function() {
    result = count + 11;
    if (result == 0) {
        setPosition(▼"imageA", 275, 0, 50, 50); -|
        setPosition(▼"imageB", 270, 408, 50, 50); -|
    }
    else if (result == 1) {
        setPosition(▼"imageA", 275, 40, 50, 50); -|
        setPosition(▼"imageB", 243, 408, 50, 50); -|
    }
    else if (result == 2) {
        setPosition(▼"imageA", 275, 80, 50, 50); -|
        setPosition(▼"imageB", 216, 408, 50, 50); -|
    }
    else if (result == 3) {
        setPosition(▼"imageA", 275, 120, 50, 50); -|
        setPosition(▼"imageB", 189, 408, 50, 50); -|
    }
    else if (result == 4) {
        setPosition(▼"imageA", 275, 160, 50, 50); -|
        setPosition(▼"imageB", 162, 408, 50, 50); -|
    }
    else if (result == 5) {
        setPosition(▼"imageA", 275, 200, 50, 50); -|
        setPosition(▼"imageB", 135, 408, 50, 50); -|
    }
    else if (result == 6) {
        setPosition(▼"imageA", 275, 240, 50, 50); -|
        setPosition(▼"imageB", 108, 408, 50, 50); -|
    }
    else if (result == 7) {
        setPosition(▼"imageA", 275, 280, 50, 50); -|
        setPosition(▼"imageB", 81, 408, 50, 50); -|
    }
    else if (result == 8) {
        setPosition(▼"imageA", 275, 320, 50, 50); -|
        setPosition(▼"imageB", 54, 408, 50, 50); -|
    }
    else if (result == 9) {
        setPosition(▼"imageA", 275, 360, 50, 50); -|
        setPosition(▼"imageB", 27, 408, 50, 50); -|
    }
    else {
        setPosition(▼"imageA", 275, 400, 50, 50); -|
        setPosition(▼"imageB", 0, 408, 50, 50); -|
    }
    -| +|
    showElement(▼"imageA");
    showElement(▼"imageB");
    count = count + 1;
});

```