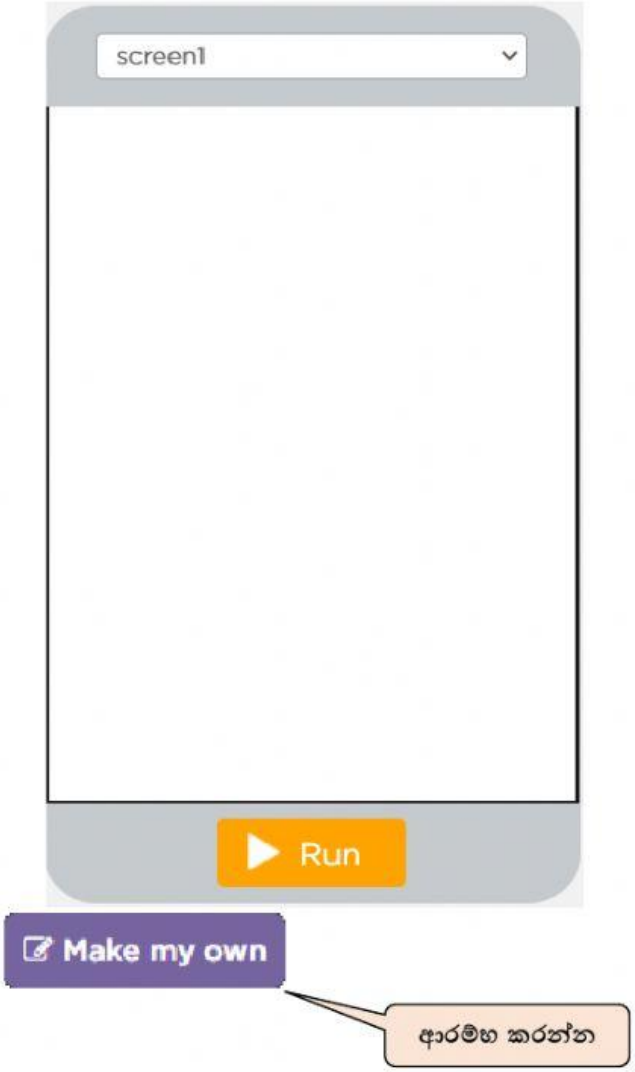
ඩී පී එඩියුකේෂන් කෝඩින් පාසල
කෝඩින් වෙසක් කලාපය - 2023

ආලෝක රටා කීපයක් එක් කිරීමට "function" හදමු



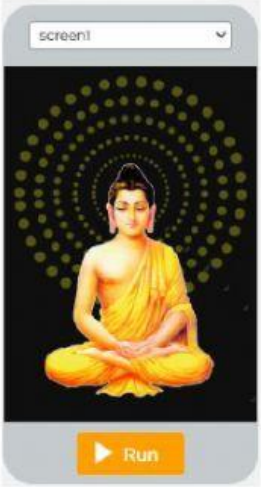
DP EDUCATION Coding School

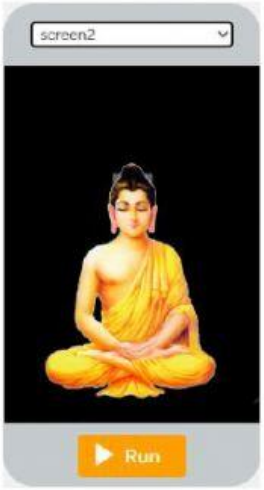
LIVEWORKSHEETS

ආලෝක රටා දෙකක් ඇති screen දෙකක් එක් කර අලංකාර එක් ආලෝක රටාවක් නිර්මාණය කරමු

- ❖ මෙම project එක නිර්මාණය කිරීම සඳහා ඔබට coding මගින් ආලෝක රටා නිර්මාණයට අදාළ පහත ව්‍යාපෘති දෙකෙහි දැනුම අවශ්‍ය වේ.

- ❖ ඔබ විසින් එම ව්‍යාපෘති වලදී නිර්මාණය කරන ලද ආකාරයේම රටා දෙකක් මෙහි දී screen දෙකක් තුළ නිර්මාණය කරගන්න.



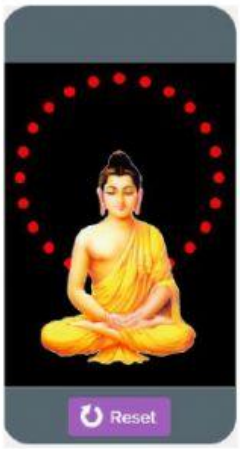


- ❖ Screen1 හි රටාව නිර්මාණයට තුන්වන project එක තුළදී භාවිතා කරන ලද codes භාවිතා කරන්න.
- ❖ Screen2 තුළ ඇති රටාව නිර්මාණයට පස්වන project ඒක තුළදී යොදාගත් codes භාවිතා කරන්න.

LIVEWORKSHEETS

නමුත් ඔබ විසින් වෙන වෙනම code කළ සියළු codes මෙහිදී එකම project එක තුළ සිදු කළ යුතුය. එම නිසා variable නම් කිරීමේදී count, result වැනි variable ව්‍යාපෘති දෙකෙහිදීම භාවිතා කළ නිසා ඒවා සඳහා එකිනෙකට වෙනස් නම් ලබා දෙන්න (count1,count2,result1,result2..)

අවසානයේ screen දෙක තුල පහත රටා දෙක දර්ශණය කිරීම සිදුකරගත යුතුය.



- දැන් අපි බලමු එක් screen එකක ඇති රටාවක් යම්කිසි වේලාවක් දර්ශණය වීමෙන් අනතුරුව අනෙක් screen එකට මාරු වී එයද යම් වේලාවක් දර්ශණය වන හා මෙම ක්‍රියාව දිගින් දිගටම සිදුවීමට අවශ්‍ය කරන code නිර්මාණය කරන ආකාරය.
- ඔබට මතක ඇති ඔබ ඉහත රටා නිර්මාණයේදී ඒවා function තුළ නිර්මාණය කර එම functions 250ms timedloop `timedLoop(250, centerLineFun);` එකක් තුළ call කළා. නමුත් මෙම project එක තුළදී අපි එම call කරන codes වෙනත් ස්ථානයක call කරන බැවින් දැනටමත් ඔබ code කර ඇති කොටසින් ඒවා delete කරන්න. Main ලෙස නිර්මාණය කරන ලද function එකද delete කරන්න. අප විසින් රටා වලට අදාළ function තුන වෙන වෙනම call කරනු ලබයි.

- ප්‍රථමයෙන් වේලාව count කරගැනීම සඳහා variable එකක් නිර්මාණය කරගත යුතුය. `var secs = 0.25;`

- ඉන්පසුව timedloop එකක් add කරගන්න. එය loop විය යුතු වේලාව ලෙස 250 ms ලෙස ලබා දෙන්න.

LIVEWORKSHEETS

❖ දැන් ඒ තුළදී ප්‍රථමයෙන් අපගේ functions තුන call කරගත යුතුය.

```
timedLoop(250, function() {
    centerLineFun();
    pattenFunction();
    resizeImage();
```

දැන් අපගේ රටාවන් Run කිරීමෙන් අනතුරුව screen දෙක තුළ දර්ශණය වේ.

❖ ඉන් පසුව අදාළ වේලාවක් අදාල රටාව පෙන්වීමට හා screen මාරු වීම නිර්මාණය කරගන්නේ කෙසේදැයි බලමු.

❖ Functions call කිරීමෙන් අනතුරුව timedloop එක තුළම පහත ආකාරයෙන් code කරන්න.

```
if(secs < 7)
{
  if (secs == 0.25) {
    setScreen("screen1");
  }
  secs = secs+0.25;
}
else if(secs >= 7 && secs <= 15)
{
  if (secs == 7) {
    setScreen("screen2");
  }
  secs = secs+0.25;
}
else{
  secs = 0.25;
}
});
```

❖ secs ලෙස අප නිර්මාණය කරගත් variable එකෙහි අගය 7 ට වඩා අඩු නම් මෙම කොටස execute වේ. මෙහිදී 6 ට වඩා අඩු වී secs වල අගය 0.25 වූ විගසම screen එක screen1 වලට මාරු වී secs හි අගයට 0.25 ක් ඒකතු කරයි. මෙම ක්‍රියාව එහි අගය 6ට වඩා වැඩි වන තෙක් දිගින් දිගටම සිදු වේ.

❖ Secs variable එකෙහි අගය 6ට වඩා වැඩි වූ විට හා 15 වඩා අඩු වන තෙක් මෙම කොටස execute වේ. මෙහිදී එහි අගය 7 වූ විටම screen ඒක "screen2" වෙත මාරු වී variable එකෙහි අගය 0.25 කින් වැඩි කරයි.

❖ ඉහත තර්ක කිසිවක් සත්‍ය නොවූ විට, එනම් secs variable එකෙහි අගය 15 වඩා වැඩි වූ විට එහි අගය නැවත 0.25 කර timedloop එක නැවත run වේ.

❖ මේ ආකාරයෙන් ඔබට රටා දෙකක් නොනැවතී screen මාරු කිරීම මගින් නිර්මාණය කරගත හැකිය. ඔබ විසින් ඉහත තර්කයන්ගෙ ඇති secs variable එක සමාන විය යුතු අගයන් වෙනස් කිරීම මගින් එක් එක් රටාව දර්ශනය වන වේලාවන් වෙනස් කරගත හැකිය.

LIVEWORKSHEETS