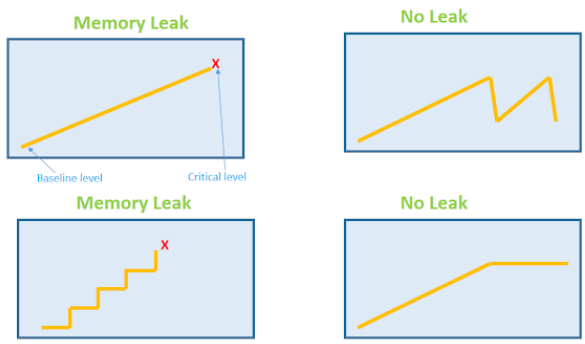
Memory Leak and Switch Statements



What is our GOAL for this MODULE?

Solve the memory leak problem

What did we ACHIEVE in the class TODAY?

- Corrected the memory leak problem in the code
- Used switch statements to randomly spawn different kinds of obstacles in the game.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Switch statement
- depth()
- Correcting memory leak

How did we DO the activities?

1. Change the depth of the clouds to be the same as the T-Rex and then increase the depth of the T-Rex by **1**. This will ensure that T-Rex has a higher depth than the clouds:

```
function spawnClouds() {
  //write code here to spawn the clouds
  if (frameCount % 60 === 0) {
    cloud = createSprite(600,300,40,10);
    cloud.addImage(cloudImage)
    cloud.y = Math.round(random(280,320));
    cloud.scale = 0.4;
    cloud.velocityX = -3;

    //adjust the depth
    cloud.depth = trex.depth
    trex.depth = trex.depth + 1;
  }
}
```

2. Spawn different kinds of obstacles on the way in the T-Rex runner game. Assign a lifetime to each cloud variable that is getting created.

   (**Formula**: **Time=Distance/Speed**; **400/3=134**)

```
function spawnClouds() {
  //write code here to spawn the clouds
  if (frameCount % 60 === 0) {
    cloud = createSprite(600,100,40,10);
    cloud.y = Math.round(random(10,60));
    cloud.addImage(cloudImage);
    cloud.scale = 0.5;
    cloud.velocityX = -3;

    //assign lifetime to the variable
    cloud.lifetime = 200;

    //adjust the depth
    cloud.depth = trex.depth;
    trex.depth = trex.depth + 1;
  }
}
```

3. Create an empty function called **spawnObstacles()** and use it inside the **draw()** function:

```
  trex.collide(invisibleGround);

  //spawn the clouds
  spawnClouds();

  //spawn obstacles on the ground
  spawnObstacles();

  drawSprites();
}

function spawnObstacles(){

}
```

4. Create an obstacle sprite for every 60 frames or so. Give the obstacle the same velocity as the ground. The obstacles need to move with the ground:

```
  //spawn the clouds
  spawnClouds();

  //spawn obstacles on the ground
  spawnObstacles();

  drawSprites();
}

function spawnObstacles(){
 if (frameCount % 60 === 0){
   var obstacle = createSprite(400,365,10,40);
   obstacle.velocityX = -6;
 }
}

function spawnClouds() {
  //write code here to spawn the clouds
```

5. Generate and store a random number between **1** to **6**. Use string concatenation to randomly assign different obstacle animations for the obstacle sprites:

```
if (frameCount % 60 === 0){
  var obstacle = createSprite(400,365,10,40);
  obstacle.velocityX = -6;

  // //generate random obstacles
  var rand = Math.round(random(1,6));
  switch(rand) {
    case 1: obstacle.addImage(obstacle1);
            break;
    case 2: obstacle.addImage(obstacle2);
            break;
    case 3: obstacle.addImage(obstacle3);
            break;
    case 4: obstacle.addImage(obstacle4);
            break;
    case 5: obstacle.addImage(obstacle5);
            break;
    case 6: obstacle.addImage(obstacle6);
            break;
    default: break;
  }

  //assign scale and lifetime to the obstacle
  obstacle.scale = 0.5;
  obstacle.lifetime = 300;
 }
}
```

- Scale the obstacles by half and give them a lifetime:

```
function spawnObstacles(){
 if (frameCount % 60 === 0){
   var obstacle = createSprite(400,365,10,40);
   obstacle.velocityX = -6;

    // //generate random obstacles
    var rand = Math.round(random(1,6));
    switch(rand) {
      case 1: obstacle.addImage(obstacle1);
              break;
      case 2: obstacle.addImage(obstacle2);
              break;
      case 3: obstacle.addImage(obstacle3);
              break;
      case 4: obstacle.addImage(obstacle4);
              break;
      case 5: obstacle.addImage(obstacle5);
              break;
      case 6: obstacle.addImage(obstacle6);
              break;
      default: break;
    }

    //assign scale and lifetime to the obstacle
    obstacle.scale = 0.5;
    obstacle.lifetime = 300;
 }
}
```

What's next?

Building collisions with the obstacles and using game states.

Extend Your Knowledge:

You can read more about the different functions of **p5.play** by exploring the examples in the following link:
https://molleindustria.github.io/p5.play/examples/index.html?fileName=animation.js