GAME STATES AND GROUPS

What is our GOAL for this MODULE?

We created two game states and assigned different behaviors to them. We also set colliders for all the objects of the game.

What did we ACHIEVE in the class TODAY?

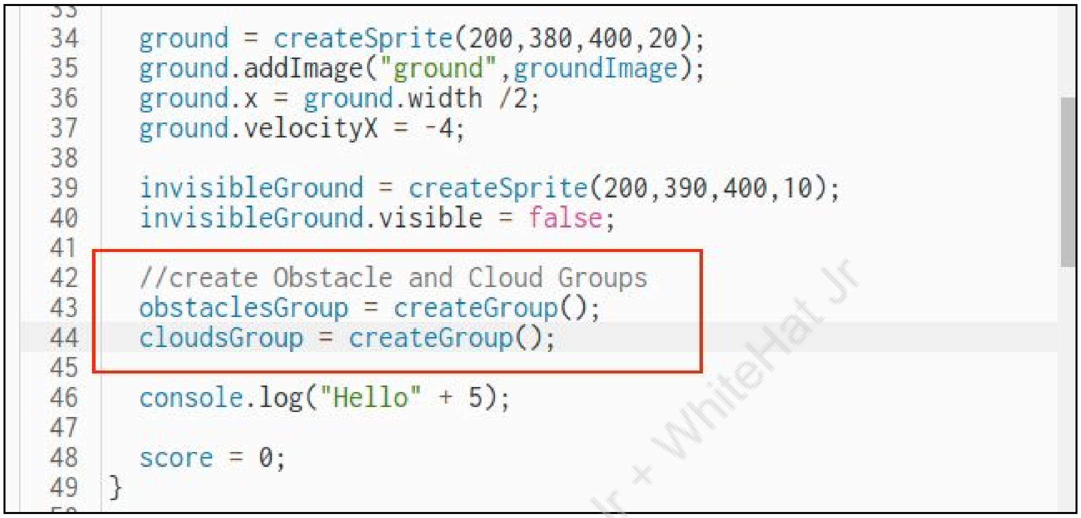
- Created two new game states - PLAY and END.
- Assigned different game behavior for the different states.
- Grouped similar game objects in a group and assigned the same behavior to all of them.
- Created colliders for the Trex and each obstacle.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Group()
- Gamestate
- Colliders

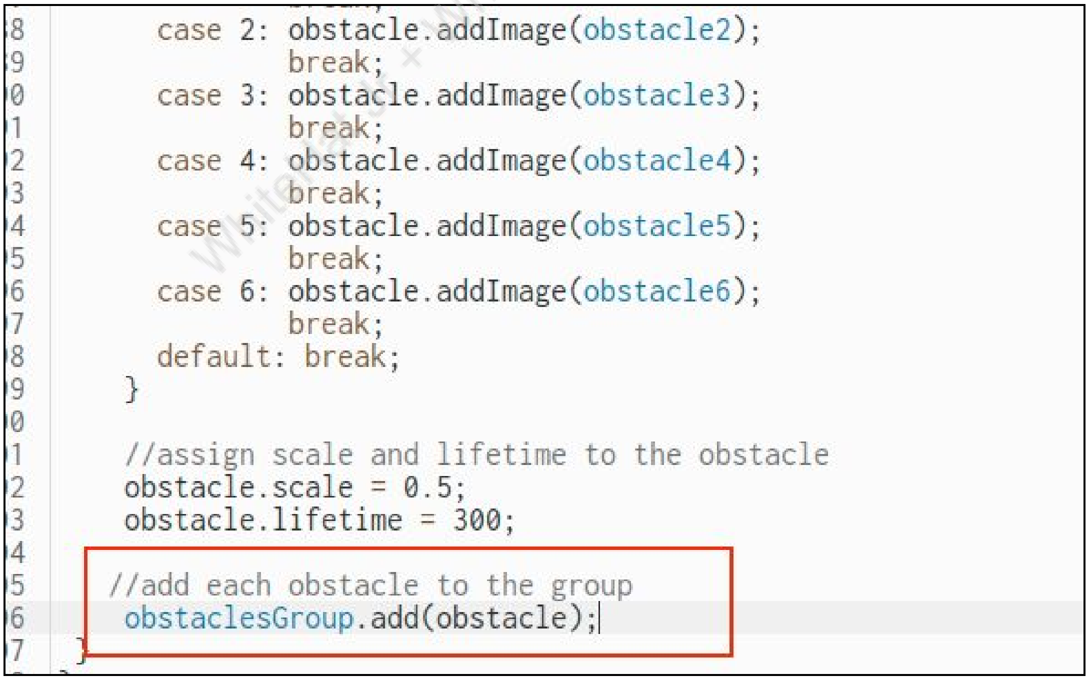
How did we DO the activities?

1. Group all objects into a single group [clouds and obstacles (cactus)].
    - Program the behavior of all the objects in a single stroke using group properties.

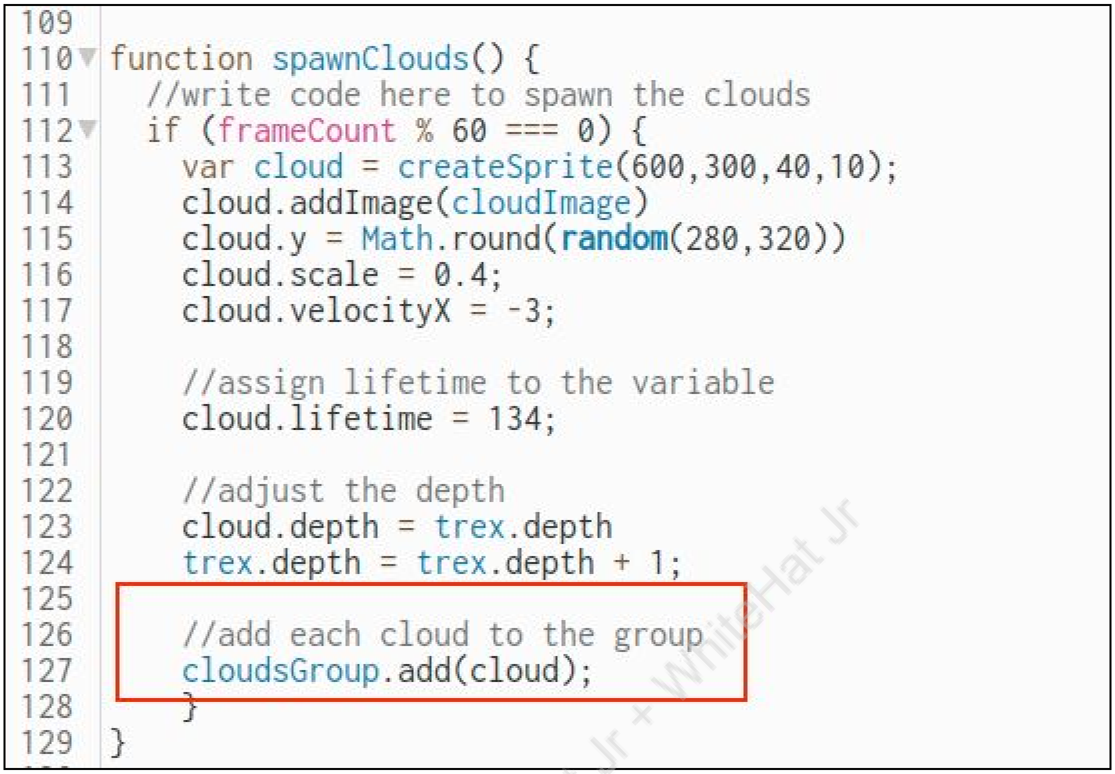


```
  ground = createSprite(200,380,400,20);
  ground.addImage("ground",groundImage);
  ground.x = ground.width /2;
  ground.velocityX = -4;

  invisibleGround = createSprite(200,390,400,10);
  invisibleGround.visible = false;

  //create Obstacle and Cloud Groups
  obstaclesGroup = createGroup();
  cloudsGroup = createGroup();

  console.log("Hello" + 5);

  score = 0;
}
```

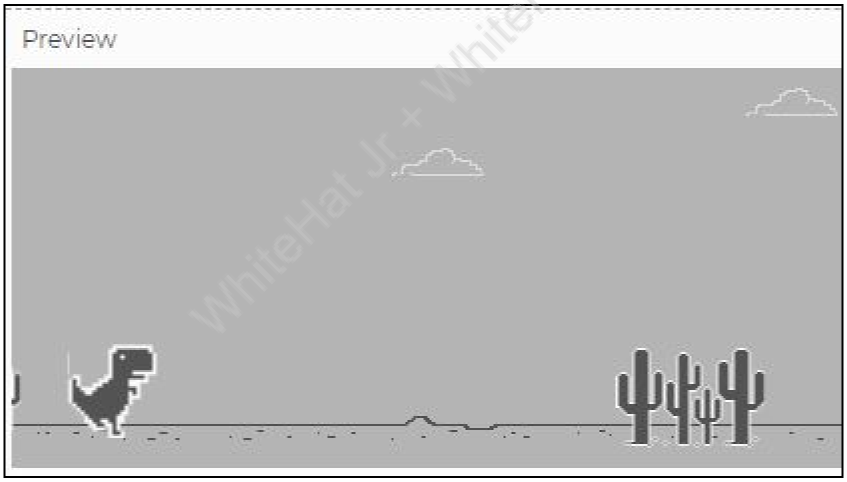
2. Add sprites to the groups.



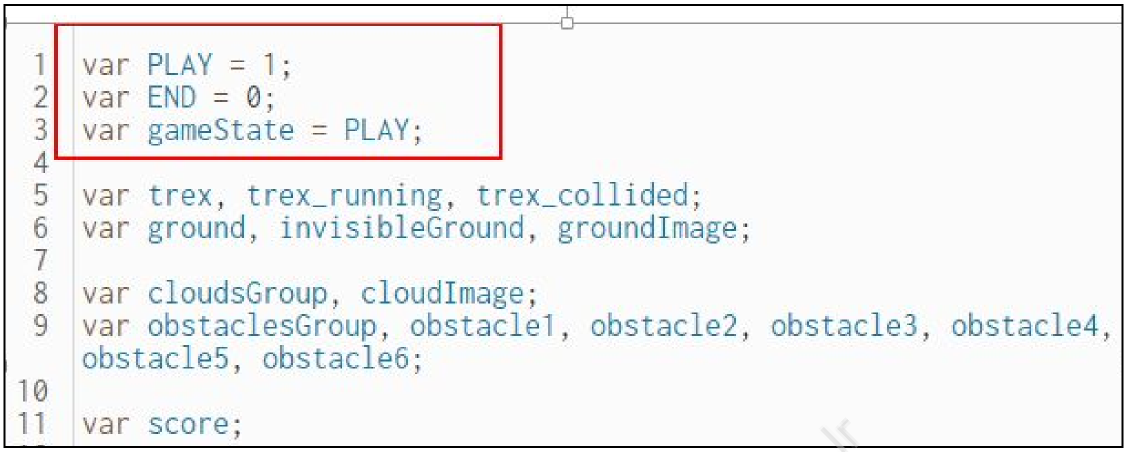
```
      case 2: obstacle.addImage(obstacle2);
              break;
      case 3: obstacle.addImage(obstacle3);
              break;
      case 4: obstacle.addImage(obstacle4);
              break;
      case 5: obstacle.addImage(obstacle5);
              break;
      case 6: obstacle.addImage(obstacle6);
              break;
      default: break;
    }

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

   //add each obstacle to the group
    obstaclesGroup.add(obstacle);
  }
```



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

    //assign lifetime to the variable
    cloud.lifetime = 134;

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

    //add each cloud to the group
    cloudsGroup.add(cloud);
    }
}
```

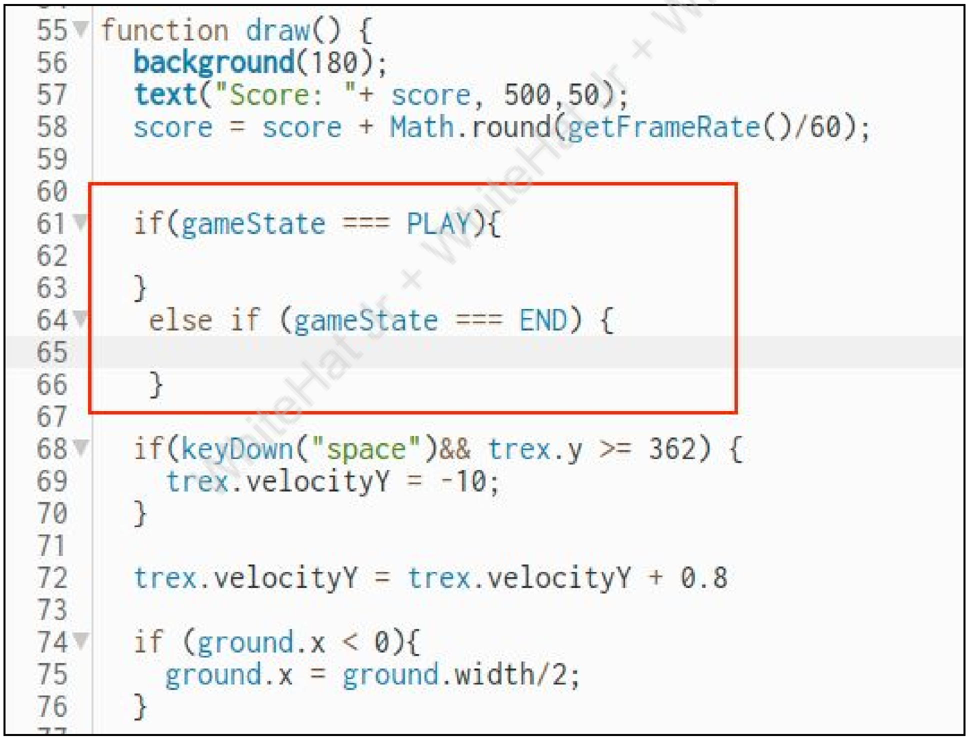


3. Introduce a variable that will hold the game state's value and set it to either PLAY or END.



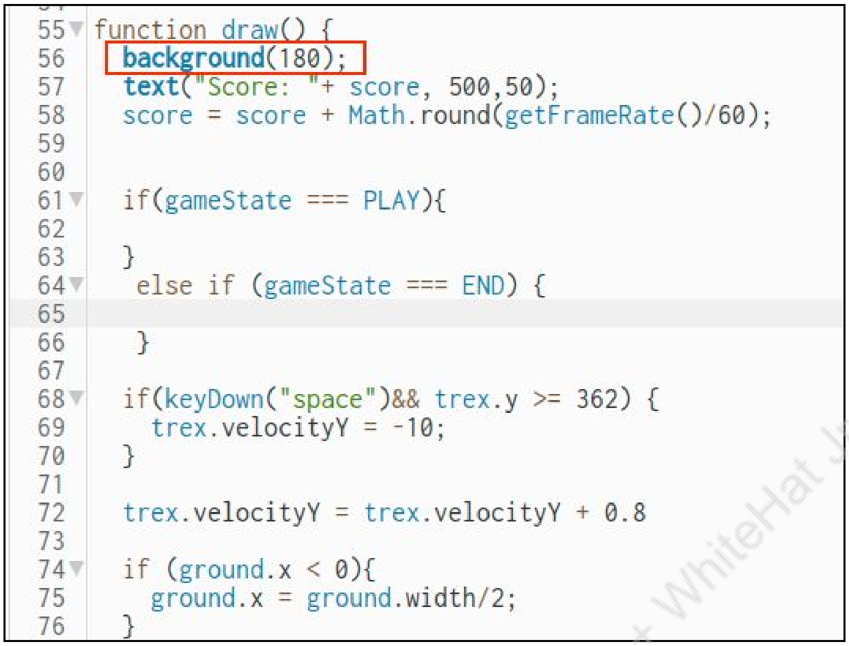
```
var PLAY = 1;
var END = 0;
var gameState = PLAY;

var trex, trex_running, trex_collided;
var ground, invisibleGround, groundImage;

var cloudsGroup, cloudImage;
var obstaclesGroup, obstacle1, obstacle2, obstacle3, obstacle4, obstacle5, obstacle6;

var score;
```

4. Add an **if** and **else-if** condition inside the function **draw()**.



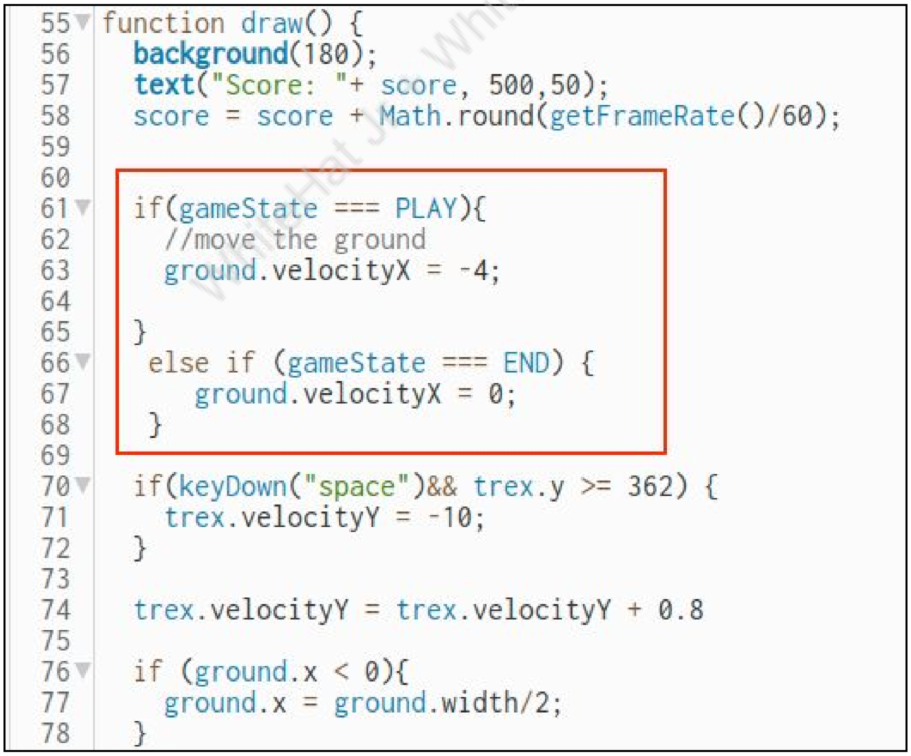
```
function draw() {
  background(180);
  text("Score: "+ score, 500,50);
  score = score + Math.round(getFrameRate()/60);


  if(gameState === PLAY){

  }
   else if (gameState === END) {

   }

  if(keyDown("space")&& trex.y >= 362) {
    trex.velocityY = -10;
  }

  trex.velocityY = trex.velocityY + 0.8

  if (ground.x < 0){
    ground.x = ground.width/2;
  }
```

5. Add behaviors inside gameState.



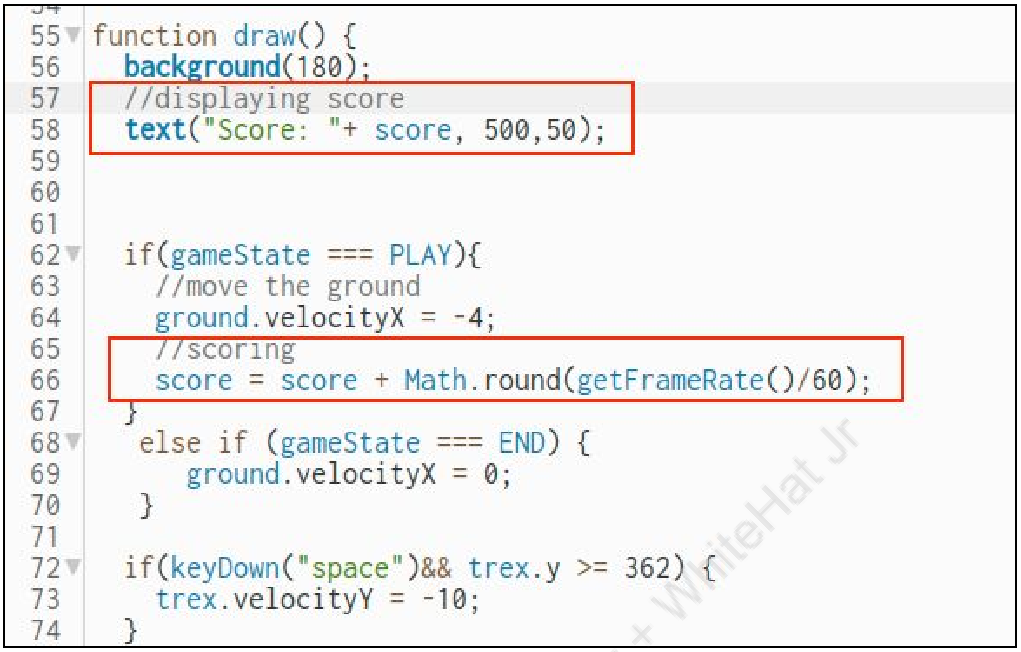
```
function draw() {
  background(180);
  text("Score: "+ score, 500,50);
  score = score + Math.round(getFrameRate()/60);


  if(gameState === PLAY){

  }
   else if (gameState === END) {

   }

  if(keyDown("space")&& trex.y >= 362) {
    trex.velocityY = -10;
  }

  trex.velocityY = trex.velocityY + 0.8

  if (ground.x < 0){
    ground.x = ground.width/2;
  }
```

6. Move the ground in PLAY state and stop the movement in END state.

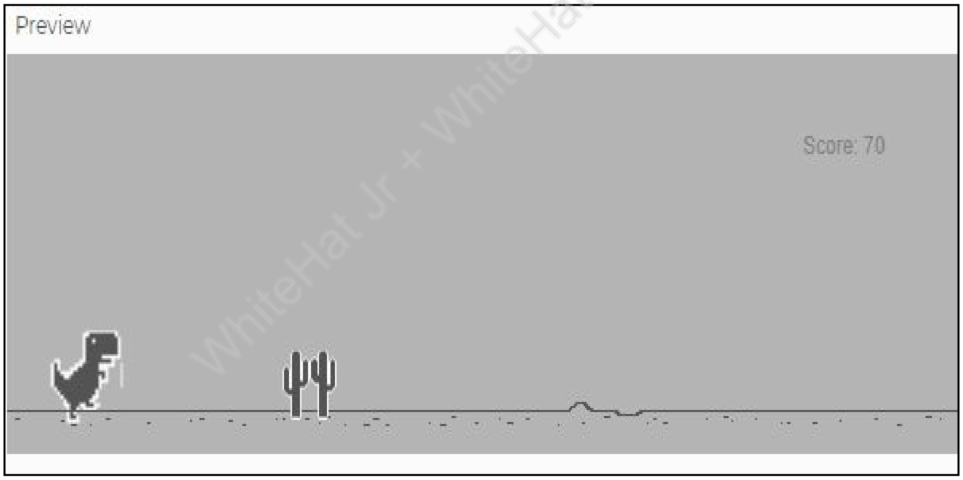


```
function draw() {
  background(180);
  text("Score: "+ score, 500,50);
  score = score + Math.round(getFrameRate()/60);


  if(gameState === PLAY){
    //move the ground
    ground.velocityX = -4;

  }
   else if (gameState === END) {
      ground.velocityX = 0;
   }

  if(keyDown("space")&& trex.y >= 362) {
    trex.velocityY = -10;
  }

  trex.velocityY = trex.velocityY + 0.8

  if (ground.x < 0){
    ground.x = ground.width/2;
  }
```

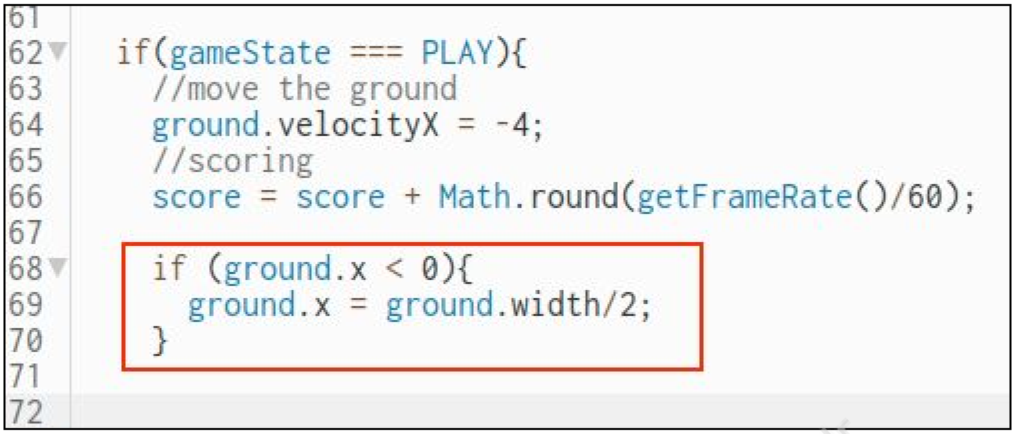
7. Display score at all times.



```
function draw() {
  background(180);
  //displaying score
  text("Score: "+ score, 500,50);



  if(gameState === PLAY){
    //move the ground
    ground.velocityX = -4;
    //scoring
    score = score + Math.round(getFrameRate()/60);
  }
   else if (gameState === END) {
      ground.velocityX = 0;
   }

  if(keyDown("space")&& trex.y >= 362) {
    trex.velocityY = -10;
  }
```

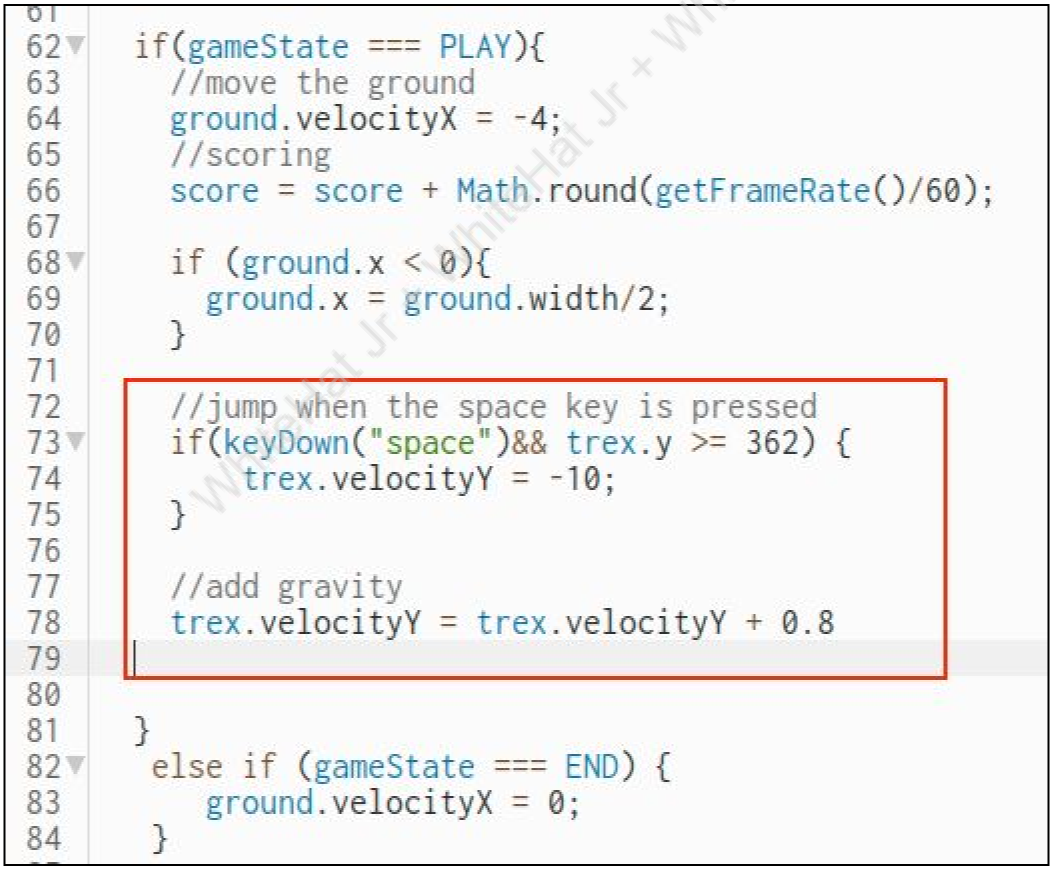


8. Reset ground during PLAY state.



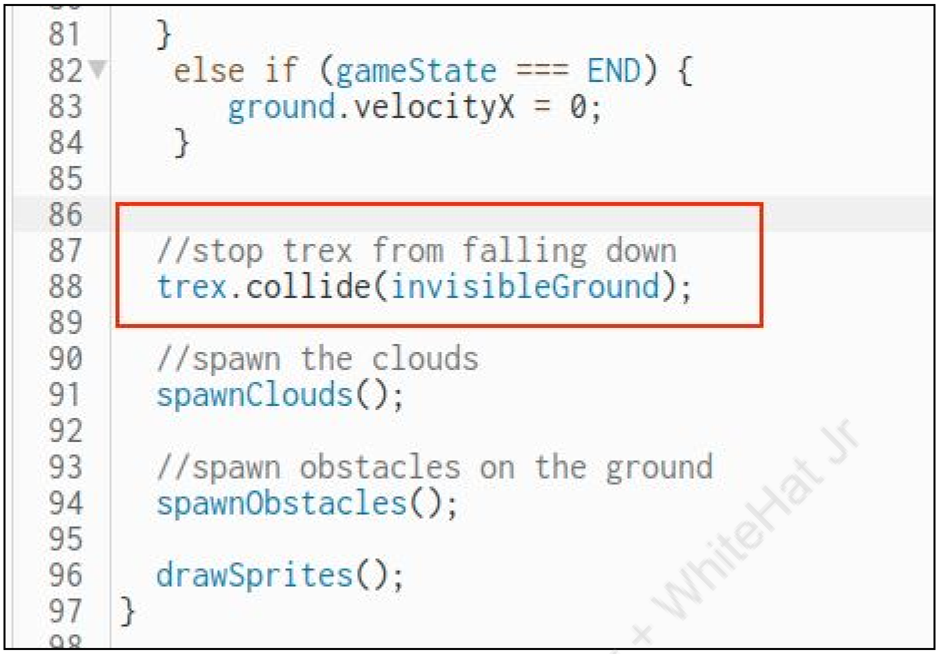
```
  if(gameState === PLAY){
    //move the ground
    ground.velocityX = -4;
    //scoring
    score = score + Math.round(getFrameRate()/60);

    if (ground.x < 0){
      ground.x = ground.width/2;
    }


```

9. Make Trex jump only during the PLAY state.



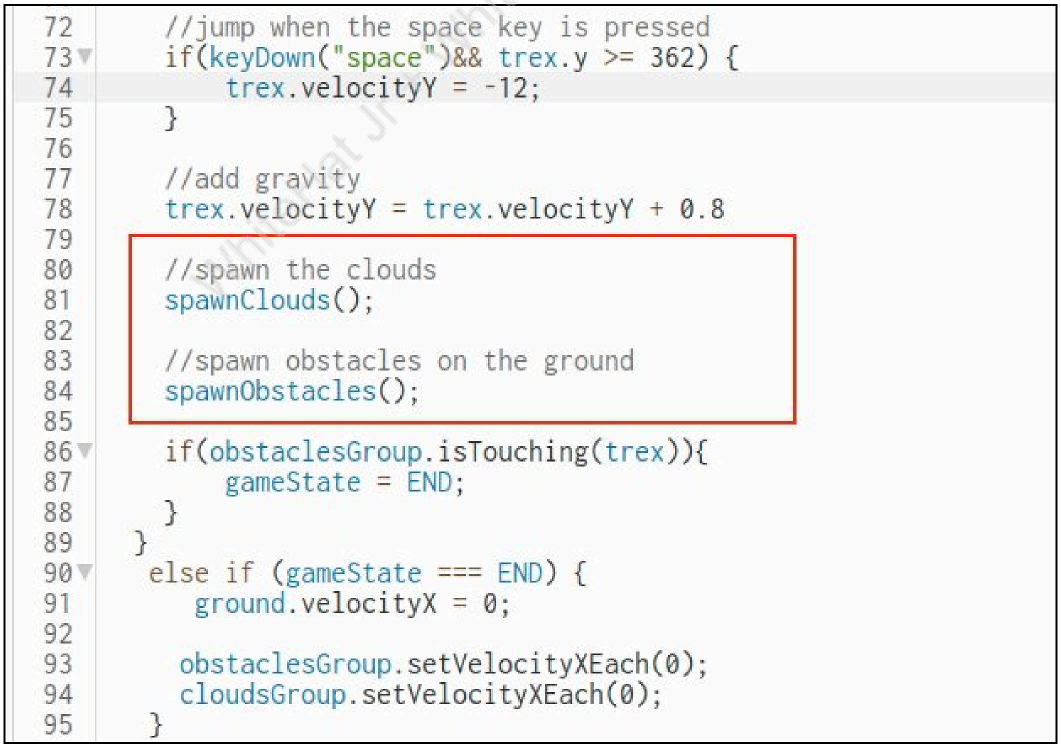
```
  if(gameState === PLAY){
    //move the ground
    ground.velocityX = -4;
    //scoring
    score = score + Math.round(getFrameRate()/60);

    if (ground.x < 0){
      ground.x = ground.width/2;
    }

    //jump when the space key is pressed
    if(keyDown("space")&& trex.y >= 362) {
        trex.velocityY = -10;
    }

    //add gravity
    trex.velocityY = trex.velocityY + 0.8


  }
   else if (gameState === END) {
      ground.velocityX = 0;
   }
```

10. Make the invisible ground support the Trex at all times.



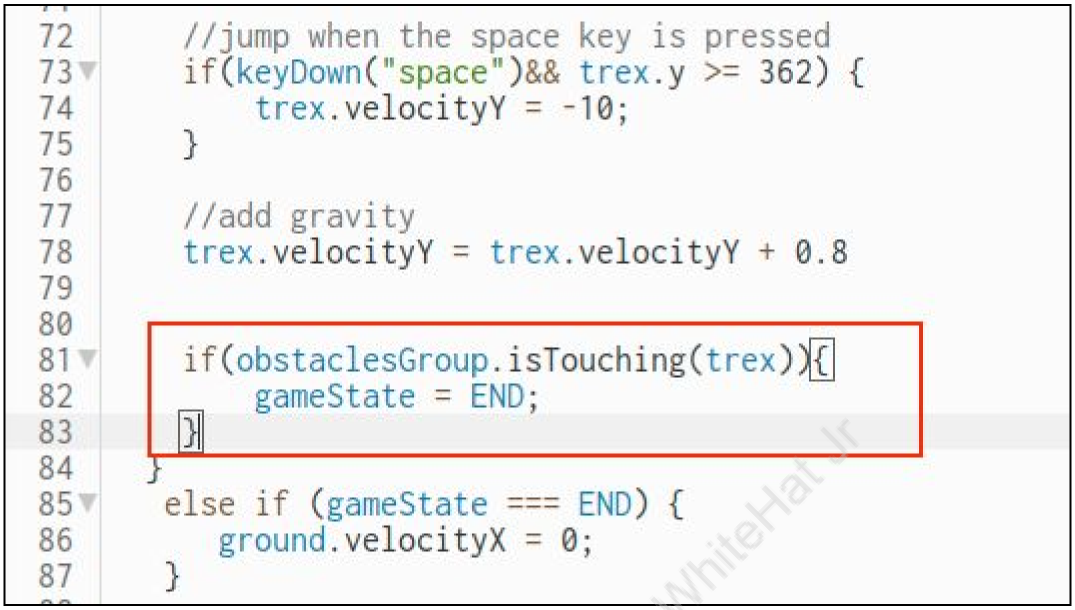
```
  }
   else if (gameState === END) {
      ground.velocityX = 0;
   }


  //stop trex from falling down
  trex.collide(invisibleGround);

  //spawn the clouds
  spawnClouds();

  //spawn obstacles on the ground
  spawnObstacles();

  drawSprites();
}
```

11. Spawn the cloud and the obstacles In PLAY state.



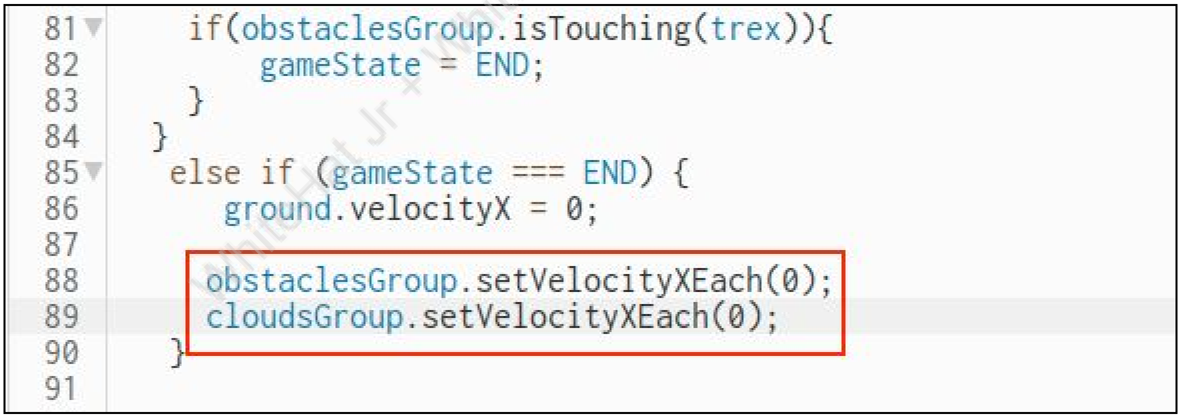
```
    //jump when the space key is pressed
    if(keyDown("space")&& trex.y >= 362) {
        trex.velocityY = -12;
    }

    //add gravity
    trex.velocityY = trex.velocityY + 0.8

    //spawn the clouds
    spawnClouds();

    //spawn obstacles on the ground
    spawnObstacles();

    if(obstaclesGroup.isTouching(trex)){
        gameState = END;
    }
  }
   else if (gameState === END) {
      ground.velocityX = 0;

     obstaclesGroup.setVelocityXEach(0);
     cloudsGroup.setVelocityXEach(0);
   }
```

12. Write code to END the game when the Trex collides with the obstacles/cacti.



```
    //jump when the space key is pressed
    if(keyDown("space")&& trex.y >= 362) {
        trex.velocityY = -10;
    }

    //add gravity
    trex.velocityY = trex.velocityY + 0.8

    if(obstaclesGroup.isTouching(trex)){
        gameState = END;
    }
  }
   else if (gameState === END) {
      ground.velocityX = 0;
   }
```

13. Give zero velocity to all the obstacles and the clouds in the game when the Trex collides with an obstacle.



```
    if(obstaclesGroup.isTouching(trex)){
        gameState = END;
    }
  }
   else if (gameState === END) {
      ground.velocityX = 0;

     obstaclesGroup.setVelocityXEach(0);
     cloudsGroup.setVelocityXEach(0);
   }
```

What's next:

We will fix the bugs present in the game.

Extend Your Knowledge:

1. P5 functions and their examples: Read more about the different functions of p5.play via various examples.