Game States and Groups

What is our GOAL for this MODULE?

In this lesson, we learned defined behavior of different objects of the game based on GameState “PLAY” & “END”

What did we ACHIEVE in the class TODAY?

- Used JavaScript objects to save different types of data in key-value format.
- Created two new game states - **PLAY** and **END**.
- Assigned different game behavior for the different states.
- Grouped similar game objects/sprites in a group and assigned the same behavior to all of them.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- JavaScript objects [JSON]
- Groups of similar objects
- Game states and assigning different behaviors

How did we DO the activities?

1. Practice JavaScript objects. Apart from an array, it is another method to store multiple data in a single variable:
    - First, we declare a variable name, set it equal to empty parentheses. e.g. **var Student = { };**
    - Add data name (key) and their value between **{ }** separated by comma:

```
var Student = {
  name: "Sammy",
  class: 7,
  roll_no:21,
  favorite_subject: "coding",
  marks : [30,35,40,50]

};
```

    - Access each data from this variable using a dot eg. variablename.key name:

```
var Student = {
  name: "Sammy",
  class: 7,
  roll_no:21,
  favorite_subject: "coding",
  marks : [30,35,40,50]

};

function setup() {

  createCanvas(400, 400);
  console.log(Student.name)
  console.log(Student.class)
  console.log(Student.favorite_subject)

}
```

2. Write one more code using JavaScript object to draw circle of different size & color:

```
var ball = {
  x:20,
  y:30,
  r:30,
  xspeed:0,
  yspeed:0,
  color:["blue","red","green","purple"],

};
```

```
function draw()
{
  background(220);
  circle(ball.x,ball.y,ball.r);
  fill(ball.color[0]);

}
```

3. Add **Group** functionality to group similar objects into a single group (Cloud and obstacle(cactus)). Using group properties program the behavior of all the objects in a single stroke:

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

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

  console.log("Hello" + 5);

  score = 0;
}

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

4. 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
    obstacleGroup.add(obstacle);
  }
}
```

```
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;

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

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

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

5. Add a variable that will hold the value of the game state. The game state could be **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;
```

6. Add an **if** and **else if** condition inside the **draw()** function:

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

  if(gameState === PLAY){


  }
  else if (gameState === END) {

  }
```

7. Add behaviors inside the game state:

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

  if(gameState === PLAY){


  }
  else if (gameState === END) {

  }
```

8. Move the ground, in **PLAY** state, stop the movement in **END** state:

```
//displaying score
text("Score: "+ score, 500,50);
score = score + Math.round(frameCount/60);

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

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

9. 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(frameCount/60);

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

10. Reset the ground during play:

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

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

    //jump when the space key is pressed
    if(keyDown("space")&& trex.y >= 100) {
        trex.velocityY = -13;
    }
```

11. Make trex jump only during the play state:

```
  //move the ground
  ground.velocityX = -4;
  //scoring
  score = score + Math.round(frameCount/60);

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

  //jump when the space key is pressed
  if(keyDown("space")&& trex.y >= 100) {
      trex.velocityY = -13;
  }

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

}

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

12. Make the invisible ground support the Trex at all times:

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

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


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


  drawSprites();
}
```

13. Spawn the cloud and the obstacles In **PLAY** state:

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

//spawn obstacles on the ground
spawnObstacles();
```

14. Write code to **END** the game when the Trex collides with the obstacles/ cactus:

```
if(obstaclesGroup.isTouching(trex)){
    gameState = END;
}
}

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

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

15. Give '**0**' 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 learn to fix the bugs in the game.

Extend Your Knowledge:

Learn and experiment with Groups.
https://studio.code.org/docs/gamelab/createGroup/#:~:text=Creates%20a%20new%20group%20and,all%20the%20%22enemy%22%20sprites