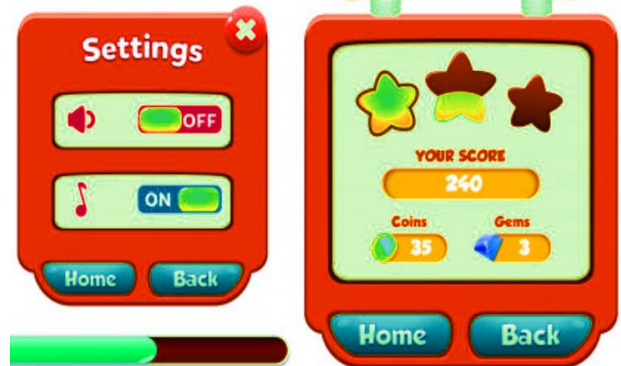


SCOPE OF VARIABLES



What is our GOAL for this MODULE?

We used our knowledge of variables, functions, loops, game states, etc to reset the game and set up a local environment to run the trex code locally.

What did we ACHIEVE in the class TODAY?

- Changed the scope of variable from local to global.
- Reset game when the reset icon is pressed.
- Set up a local environment to run trex code locally.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Scope of variables.
- Changing game state.
- Setting up a local environment.

How did we DO the activities?

We saw the scope of the variable and why we declare them. We declared a message variable in the setup function and tried to log the message in the draw function. We got an uncaught reference error.

```
1  var PLAY = 1;
2  var END = 0;
3  var gameState = PLAY;
4
5  var trex, trex_running, trex_collided;
6  var ground, invisibleGround, groundImage;
7
8  var cloudsGroup, cloudImage;
9  var obstaclesGroup, obstacle1, obstacle2, obstacle3,
10 obstacle4, obstacle5, obstacle6;
11
12 var score;
13 var gameOverImg, restartImg
14 var jumpSound, checkPointSound, dieSound
15 function preload() {
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

onsole Clear

1. Declared the message variable in the setup function.

```
37
38 function setup() {
39   createCanvas(600, 500);
40
41   var message = "This is a message";
42   console.log(message);
43
44   trex = createSprite(50, 380, 20, 50);
45
46   trex.addAnimation("running", trex_running);
47   trex.addAnimation("collided", trex_collided);
48
49
50   trex.scale = 0.5;
51
52   ground = createSprite(200, 380, 400, 20);
53   ground.addImage("ground", groundImage);
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

onsole

your code is already using that name as a variable. You may

2. Logged the message in the draw function.

```
82 }
83
84 function draw() {
85     background(180);
86     console.log(message);
87     //displaying score
88     text("Score: " + score, 500, 50);
89
90
91     if(gameState === PLAY){
92         //move the
93         gameOver.visible = false;
94         restart.visible = false;
95
96         ground.velocityX = -(4 + 3* score/100)
97         //scoring
98     }
```

Console

✖ Uncaught ReferenceError: message is not defined (sketch: line 87)

Thus we learned that variables have a scope and they live and die inside the scope.

```
87
88     ground.velocityX = -4;
89     //scoring
90     score = score + Math.round(getFrameRate()/60);
91
92     if(score>0 && score%100 === 0){
93         checkPointSound.play()
94     }
95
96     if (ground.x < 0){
97         ground.x = ground.width/2;
98     }
99
100     //jump when the space key is pressed
101     if(keyDown("space")&& trex.y >= 362) {
102         trex.velocityY = -12;
103         jumpSound.play();
104     }
105
106     //add gravity
107     trex.velocityY = trex.velocityY + 0.8
108
109     //spawn the clouds
110     spawnClouds();
111
112     //spawn obstacles on the ground
113     spawnObstacles();
```

3. Add a reset icon and add functionality to it.



4. We used the `mousePressedOver()` instruction to detect if the mouse is pressed over the reset sprite and console log something when the mouse is pressed.

```
143
144 //stop trex from falling down
145 trex.collide(invisibleGround);
146
147 if(mousePressedOver(restart)) {
148     console.log("Restart the Game");
149 }
150
151 drawSprites();
152 }
153
154 function spawnObstacles(){
155     if (frameCount % 60 === 0){
156         var obstacle = createSprite(400,365,10,40);
157         obstacle.velocityX = -(6 + score/100);
158     }
```

5. We called a reset() function which resets everything in the game to its original state instead of printing.

```
142
143
144 //stop trex from falling down
145 trex.collide(invisibleGround);
146
147 if(mousePressedOver(restart)) {
148     reset();
149 }
150
151 drawSprites();
152 }
153
154 function reset(){
155 }
156
157
158
```

Console

6. We wrote code for the reset function.

```
< sketch.js Saved: just no
152 }
153
154 function reset(){
155     gameState = PLAY;
156     gameOver.visible = false;
157     restart.visible = false;
158 }
159
160
161
162
163 function spawnObstacles(){
164     if (frameCount % 60 === 0){
165         var obstacle = createSprite(400,365,10,40);
166         obstacle.velocityX = -(6 + score/100);
167
168         //generate random obstacles
```


7. In the previous class we changed the life time of the objects so that they don't disappear. So as long as the obstacles remain the trex will collide with them and the game state will change to END. So we destroy the obstacles when the game is reset.

```
151 drawSprites();
152 }
153
154 function reset(){
155     gameState = PLAY;
156     gameOver.visible = false;
157     restart.visible = false;
158     obstaclesGroup.destroyEach();
159     cloudsGroup.destroyEach();
160 }
161
162
163
164
165 function spawnObstacles(){
166     if (frameCount % 60 === 0){
167         var obstacle = createSprite(400,365,10,40);
```

8. Then we changed the trex collided animation to trex running.

```
154 function reset(){
155     gameState = PLAY;
156     gameOver.visible = false;
157     restart.visible = false;
158     obstaclesGroup.destroyEach();
159     cloudsGroup.destroyEach();
160     trex.changeAnimation("running",trex_running);
161 }
162
163
164
165
166
167
```

9. We moved the condition where we checked the mouse pressed over reset icon inside Game End state to fix the restarting game issue even when the reset icon was invisible.

```
    }  
    else if (gameState === END) {  
        gameOver.visible = true;  
        restart.visible = true;  
  
        if(mousePressedOver(restart)) {  
            reset();  
        }  
  
        ground.velocityX = 0;  
        trex.velocityY = 0  
        //change the trex animation  
        trex.changeAnimation("collided", trex
```

10. We also needed to reset the score.

```
155     drawSprites();  
156 }  
157  
158 function reset(){  
159  
160     gameState = PLAY;  
161     gameOver.visible = false;  
162     restart.visible = false;  
163     trex.changeAnimation("running", trex_running);  
164  
165     obstaclesGroup.destroyEach();  
166     cloudsGroup.destroyEach();  
167     score = 0;  
168  
169  
170 }  
171  
Console
```

11. We updated the score count based on the frame rate to fix the issue of score resetting to 0 temporarily and then starting from the old score.

```
88 if(gameState === PLAY){
89   //move the
90   gameOver.visible = false;
91   restart.visible = false;
92
93   ground.velocityX = -(4 + 3* score/100)
94   //scoring
95   score = score + Math.round(getFrameRate()/60);
96
97   if(score>0 && score%100 === 0){
98     checkPointSound.play()
99   }
100
101   if (ground.x < 0){
102     ground.x = ground.width/2;
103   }
104
105 }
```

What's next?

We'll learn to write code on the local machine.

Extend Your Learning:

To learn more about making games in p5 visit the following link:

<https://creative-coding.decontextualize.com/making-games-with-p5-play/>