

GAME STATE



What is our GOAL for this MODULE?

We used our knowledge of functions, loops, operators and variables to create different behaviors in different conditions.

What did we ACHIEVE in the class TODAY?

- Stored the state of a game in a variable.
- Displayed different information on the screen according to the state of the game.
- Used conditional programming and logical operators to assign different behaviors to the objects in the game depending on the state of the game.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Game states
- Logical operators
- Conditional programming

How did we DO the activities?

In a game, there is a change of state, for example, Start, Play, and End.

1. The first state (Serve state) is when the ball is at the center and the user needs to press "Space" to serve the ball.

```
1 //create the ball, playerPaddle and computerPaddle as sprite objects
2 var ball = createSprite(200,200,10,10);
3 var playerPaddle = createSprite(380,200,10,70);
4 var computerPaddle = createSprite(10,200,10,70);
5
6 var gameState = "serve";
7
8 function draw() {
9   //clear the screen
10  background("white");
11
12  //place info text in the center
13  if (gameState === "serve") {
14    text("Press Space to Serve",150,180);
15  }
16
17  //make the player paddle move with the mouse's y position
18  playerPaddle.y = World.mouseY;
19
```

2. The second state (Play state) is when the play starts, and the ball is in motion.
Specific instruction: If the user presses space and the game is in SERVE state, then serve the ball.

```
30 //create edge boundaries
31 //make the ball bounce with the top and the bottom edges
32 createEdgeSprites();
33 ball.bounceOff(topEdge);
34 ball.bounceOff(bottomEdge);
35 ball.bounceOff(playerPaddle);
36 ball.bounceOff(computerPaddle);
37
38
39 //serve the ball when space is pressed
40 if (keyDown("space") && gameState === "serve") {
41   serve();
42   gameState = "play";
43 }
```

3. Change the GameState variable back to "serve" state inside the condition 'when the ball crosses the screen'.

```
38
39 //serve the ball when space is pressed
40 if (keyDown("space") && gameState === "serve") {
41     serve();
42     gameState = "play";
43 }
44
45
46 //reset the ball to the centre if it crosses the screen
47 if(ball.x > 400 || ball.x < 0) {
48     reset();
49     gameState = "serve";
50 }
51
```

4. Add a scoring system and the condition to increase the player or computer score when the opposite paddle misses hitting the ball.

```
50
51
52 //reset the ball to the centre if it crosses the screen
53 if(ball.x > 400 || ball.x < 0) {
54
55     if (ball.x > 400){
56         computerScore = computerScore + 1;
57     }
58     if (ball.x < 0){
59         playerScore = playerScore + 1;
60     }
61
62     reset();
63     gameState = "serve";
64 }
```

5. The third state (Game over) is when the player or computer scores 5 points.

```

56 //reset the ball to the centre if it crosses the screen
57 if(ball.x > 400 || ball.x < 0) {
58
59     if(ball.x > 400) {
60         compScore = compScore + 1;
61     }
62
63     if(ball.x < 0) {
64         playerScore = playerScore + 1;
65     }
66
67     reset();
68     gameState = "serve";
69 }
70
71 if (playerScore === 5 || compScore === 5){
72     gameState = "over";
73     text("Game Over!",170,160);
74     text("Press 'R' to Restart",150,180);
75 }
76
  
```

6. The game then ends and the player needs to press "R" to restart the game.

```

71 if (playerScore === 5 || compScore === 5){
72     gameState = "over";
73     text("Game Over!",170,160);
74     text("Press 'R' to Restart",150,180);
75 }
76
77 if (keyDown("r") && gameState === "over") {
78     gameState = "serve";
79     compScore = 0;
80     playerScore = 0;
81 }
82
83 drawSprites();|
84 }
  
```

What's next?

We are going to add sounds and animation to the game.

Extend Your Knowledge

Refer to the following links for some more information about game states:

1. [Understanding "Game State"](#)
2. [Game Design Concepts](#)