GAME STATES

What is our GOAL for this MODULE?

In this class, we learned to store the state (mode) of a game in a variable and assign different behavior to the objects in the game depending on the state of the game.

What did we ACHIEVE in the class TODAY?

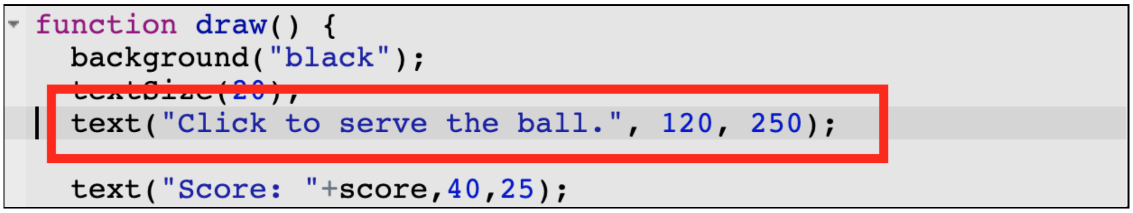
- Added a **text()** instruction near the center which says "Click to serve the ball".
- Created a variable called gamestate (mode) and gave it a starting state (mode) of "serve".
- Changed game state (mode) after the user clicks a Mouse Button
- Coded for the third game state i.e. Game Over mode.
- Detected a keypress on the Space bar and switched the game state between play and pause.
- Displayed a few more text instructions for the player to know if it's a pause state or game over.

Which CONCEPTS/ CODING BLOCKS did we cover today?

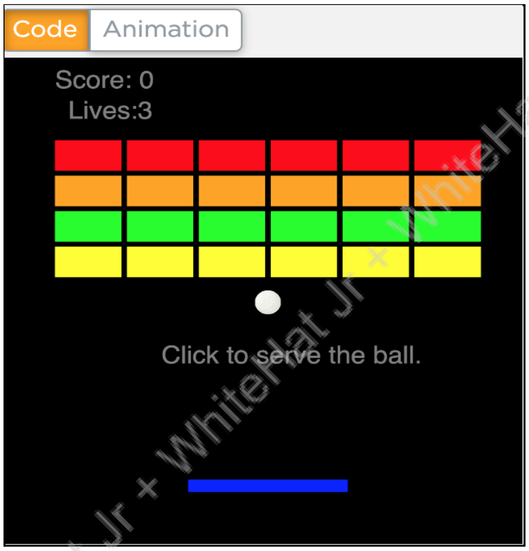
- Use conditional programming.
- Logical operators.
- Add text() instructions on the console.
- How to add game states in the game?

How did we DO the activities?

1. Add a **text()** instruction near the center, which says "**Click to serve the ball**":

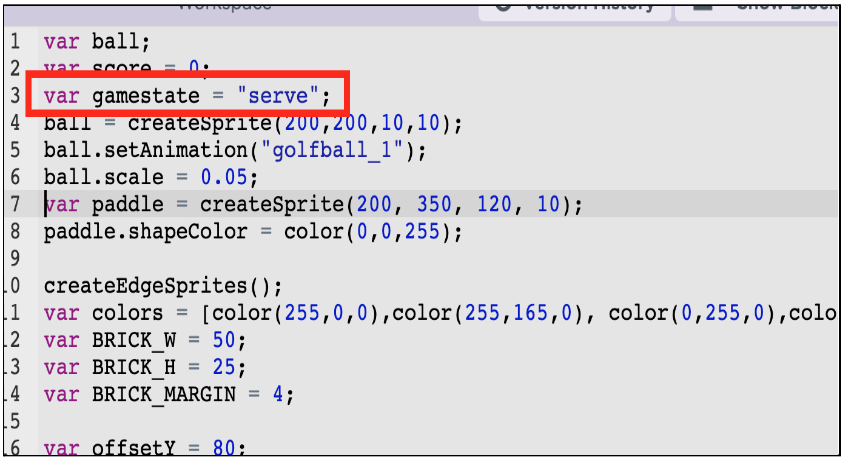


```
function draw() {
  background("black");
  textSize(20);
  text("Click to serve the ball.", 120, 250);

  text("Score:  "+score,40,25);
```



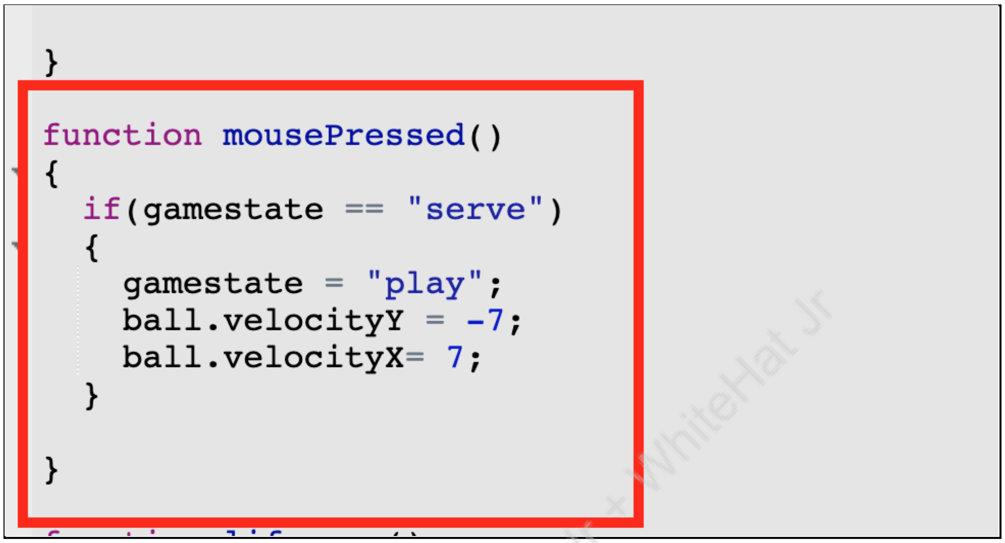
```
Score: 0
 Lives:3

Click to serve the ball.
```

2. Create a variable called gamestate (mode) and give it a starting state (mode) of **"serve"**. Whenever we are storing any text inside a variable, we put it in double quotes **(" ")**.



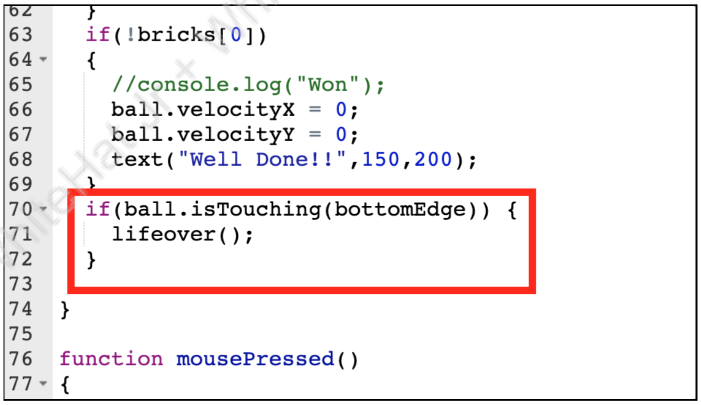
```
var ball;
var score = 0;
var gamestate = "serve";
ball = createSprite(200,200,10,10);
ball.setAnimation("golfball_1");
ball.scale = 0.05;
var paddle = createSprite(200, 350, 120, 10);
paddle.shapeColor = color(0,0,255);

createEdgeSprites();
var colors = [color(255,0,0),color(255,165,0), color(0,255,0),colo
var BRICK_W = 50;
var BRICK_H = 25;
var BRICK_MARGIN = 4;

var offsetY = 80;
```

3. Change game state (mode) after the user clicks a Mouse Button and change the gameState variable to "**play**" after the space key is pressed. Remember that the values inside the variables can change! That's why they are called "variables":



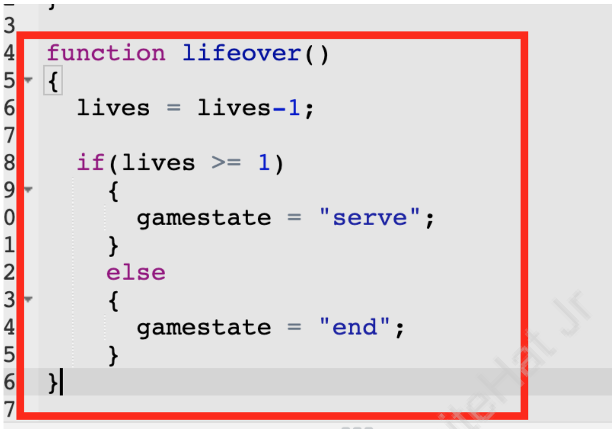
```
}
function mousePressed()
{
  if(gamestate == "serve")
  {
    gamestate = "play";
    ball.velocityY = -7;
    ball.velocityX= 7;
  }

}
```

4. Write a code to detect a touch between the ball and bottom edge of the canvas.



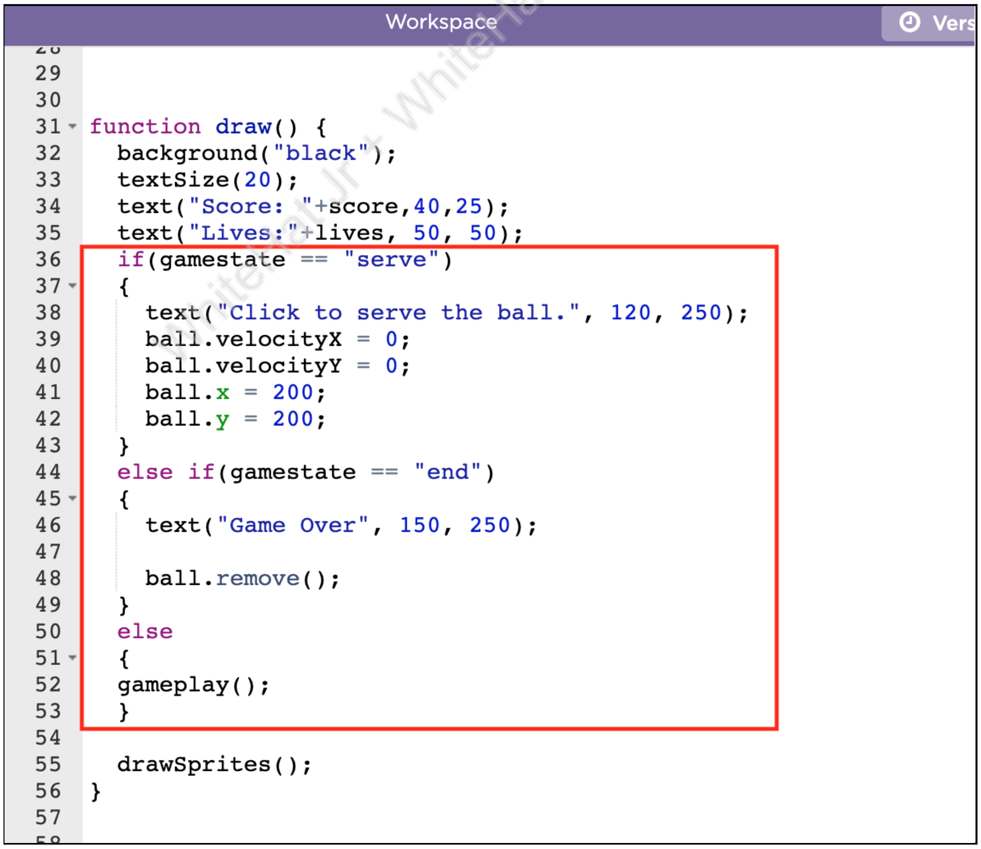
```
  }
  if(!bricks[0])
  {
    //console.log("Won");
    ball.velocityX = 0;
    ball.velocityY = 0;
    text("Well Done!!",150,200);
  }
  if(ball.isTouching(bottomEdge)) {
    lifeover();
  }

}

function mousePressed()
{
```

5. Created a custom function **lifeover()** to reduce the number of lives by **1** when a player hitted by enemy also check how many lives are left and accordingly set the gamestate.



```
function lifeover()
{
  lives = lives-1;

  if(lives >= 1)
    {
      gamestate = "serve";
    }
    else
    {
      gamestate = "end";
    }
}
```

6. Display a few more text instructions for the player to know if it's a serve state or game over state.



```
function draw() {
  background("black");
  textSize(20);
  text("Score: "+score,40,25);
  text("Lives:"+lives, 50, 50);
  if(gamestate == "serve")
  {
    text("Click to serve the ball.", 120, 250);
    ball.velocityX = 0;
    ball.velocityY = 0;
    ball.x = 200;
    ball.y = 200;
  }
  else if(gamestate == "end")
  {
    text("Game Over", 150, 250);

    ball.remove();
  }
  else
  {
  gameplay();
  }

  drawSprites();
}
```

What’s next?

In the next class, we will learn how to increase the complexity of the game and apply Artificial Intelligence to the game.

Extend Your Knowledge:

Bookmark following link: it will be a reference for Inequality operator
https://studio.code.org/docs/gamelab/inequalityOperator/