**Variable Declaration:**

1. Int points = 0;

Declaration an integer variable ‘points’ and initializes it with the value. This variable is used to keep track of the player’s score.

1. Float ball\_x = 100;

Initializes a floating-point variable ‘ball\_x’ with an initial value of 100. This variable likely represents the x-coordinate of the ball’s position on the game screen.

1. Float ball\_y = 50;

Initializes a floating-point variable ‘ball\_y’ with an initial value of 50. This variable likely represents the y-coordinate of the ball’s position on the game screen.

1. Float speed\_x = 10;

Initializes a floating-point variable ‘speed\_x’ with an initial value of 10. This variable probably represents the horizontal speed of the ball in the game.

1. Float speed\_y = 5;

Initializes a floating-point variable ‘speed­\_y’ with an initial value of 5. This variable probably represents the vertical speed of the ball in the game.

1. Boolean gameStart = false;

Initializes a Boolean variable ‘gameStart’ with an initial value of ‘false’. This variable is used to control the game’s start state.

**If changing ‘ball\_x’ and ‘ball\_y’ values:**

1. Increasing ‘ball\_x’ : increasing ‘ball\_x’ will shift the initial position of the ball to the right on the x-axis.This means the ball will start farther to the right on the screen when the game begins.
2. Decreasing ‘ball\_x’ : decreasing ‘ball\_x’ will shift the initial position of the ball to the left on the x-axis. The ball will start closer to the left side of the screen.
3. Increasing ‘ball\_y’ : increasing ‘ball\_y’ will shift the initial position of the ball upwards on the y-axis. The ball will start lower down on the screen.
4. Decreasing ‘ball\_y’ : decreasing ‘ball\_y’ will shift the initial position of the ball downwards on the y-axis. The ball will start higher up on the screen.

**If changing ‘speed\_x’ and ‘speed\_y’ values:**

1. Increasing ‘speed\_x’ : increasing ‘speed\_x’ will cause the ball to move faster horizontally (along the x-axis) in the game.
2. Decreasing ‘speed\_x’ : decreasing ‘speed\_x’ will cause the ball to move slower horizontally
3. Increasing ‘speed\_y’ : increasing ‘speed\_y’ will cause the ball to move faster vertically (along the y-axis) in the game.
4. Decreasing ‘speed\_y’ : decreasing ‘speed\_y’ will make the ball move slower vertically.

**If changing ‘points’ variable:**

1. Increasing ‘points’ : Increasing the ‘points’ variable will directly increase the player’s score in the game.The player will earn more points as the variable valule goes up.
2. Decreasing ‘points’ : Decreasing the ‘points’ variable is typically not recommended in a game context as it might reduce the player's score, but adjusting the points negatively could represent a penalty system in some game scenarios.

**Float Distance :**

1. Abs function is used to calculate the absolute value of a number, ensuring that the result is a positive value, regardless of the sign of the expression within it.
2. Distance is a variable that stores the absolute distance between the x-coordinate of the mouse (‘mouseX’) and the x-coordinate of the ball (‘ball\_x’).
3. When distance is less than 100, points will increase and will print out +1.

**Else part:**

1. ‘speed\_y += 1;’ increments the vertical speed ‘speed\_y’ by 1 unit.This implies that when the conditions within the ‘if’ statement are not met, the vertical speed will increase by 1.It is a acceleration.
2. ‘ball\_x = ball\_x + speed\_x’ and ‘ball\_y = ball\_y + speed\_y’ update the positions of the ball in the x and y direction based on the current speed values. This repositions the ball for the next frame or iteration of the game loop.

This part of the code seems to define the behaviour of the ball when it doesn’t meet the condition where the distance between the ball and the mouse is less than 100 units. In this case, it increases the vertical speed of the ball by 1 unit and then updates the position of the ball both horizontally and vertically based on the updated speeds.