AP Computer Science Principles: 2020 Create Task
Student Sample D

3a.

The video illustrates the beginning of our digital air hockey game. The purpose of our program is to provide enjoyable entertainment to solitary players. Many players want to be able to play video games by themselves but find it impossible due to the design of the game, however our program allows players to easily play a two person game, with one person. The input demonstrated is the location of the mouse (x coordinate and y coordinate) which is passed to two different methods depending on whether the mouse is on the right half or left half of the screen. If the mouse is on the left half, the mouse position is sent to the method that controls the paddle on the left side, allowing it to move up and down on the screen as demonstrated, and vice versa for the right paddle. The output is the visible interface of the game.

3b.

```
if(puckX <= 45 && puckY >= 150 && puckY <= 250)
{
  scores[0]++;
  reset();
  print("left");
}

if(puckX >= 755 && puckY >= 150 && puckY <= 250)
{
  scores[1]++;
  reset();
  print("right");
}
```

```
text(scores[1], 20, 40);
text(scores[0], 780, 40);
```

The image on the left shows the first code segment which demonstrates how the list “scores” is updated. It keeps track of both the score of the left-side player and the score of the right-side player in the first and second indices of the list. The image on the right shows the second code segment, which demonstrates how the “scores” list is displayed using the text method which is built into processing. We pass the score held in the list corresponding to the particular player to the specific positions on the left and right sides of the screen. Without using this list we would have to use variables to hold these and it would be more difficult to increment the score.

3c.

Method lpy:

```
int lpy(int y)
{
  if(mouseX < 400 && y < 360 && y > 40)
  {
    leftPaddleY = y;
  }
  return(leftPaddleY);
}
```

Call to method lpy:

```
rect(40, lpy(2 * mouseY - 200), 5, 80);
```

Method, lpy, captured above demonstrates the incrementation of the y coordinate of the left paddle. My partner and I developed this program to limit the movement of the left paddle to only when the mouse is on the left side of the screen. We pass the integer value y, which holds the y intercept of the mouse. Next, we use selection to limit the left paddle's movement up and down by only allowing the leftPaddleY variable equal the mouse's y coordinate if it is less than 360 and greater than 40. The other condition is that the mouse's x coordinate is less than 400, checking that the mouse is on the left side of the screen.

3d.

One path through the code discussed in 3c is for the mouse's y position to be less than 40, which is meant to test when the y position would not meet the requirements for the condition listed in the if statement. This would cause the program to skip over the assignment of the y coordinate of the left paddle to the y coordinate of the mouse, so the paddle wouldn't move.

The second path through the code would be if the mouse's y coordinate was 50, which is meant to test when the y position would meet the requirements of the if statement and the y coordinate of the mouse was assigned to the y coordinate of the left paddle, allowing the paddle to move in the indicated direction up or down.