In general,

1. We need more comments and documentation on how our code works.
    a. Some methods need to be more specific on what they do, or just have comments in general

2. There are small errors in the code that could cause compilation errors.
    a. The string variables are compared by ==, not by .equal() function which may cause an error depending on what compiler you use.

Within Enemy.java,

3. There is poorly structured code with the constructor for the class.
    a. There two constructors,
        i. One constructor has no parameter and creates its own Point variable for usage. This constructor is not used by anyone.
        ii. The other constructor has a parameter for Point and instantiates this.pos as that point. This constructor is used.
    b. To refactor, I deleted the useless constructor.

4. moveToPlayer() does too much work and has a lot of duplicated code.
    a. There are 8 cases for an enemy to chase the player, and all of the cases are coded in moveToPlayer() function. This can be simplified by creating a new function that gets a current position as an instance. Thus creates a new function called moveHelp(int x, int y), which gets a distance as an instance.
    b. Also, the code included both codes that move to a specific direction (i.e moveUp();) and changing the sprites to regarding direction (i.e direction=”up”). Since it was quite inefficient, I included changing sprites code in the moving function, which was in character class. In this way it not only simplifies the code, but also increases the cohesion.

5. attackPlayer() is a useless function

    a. attackPlayer() serves no purpose because it is not used by anyone. It's function is also redundant because all of the work is done already in the tick() method, and therefore doesn't need to be called.

6. We have variables called XPosition and YPosition that keep getting initialized whenever getXPosition() or getYPosition is called and we want to have fields for these variables
    a. Make a public variable that can be called from outside, and change the variables' name from "newX" and "newY" to "currentX" and "currentY" in order to be more easily understood.

Within Player.java

7. The player class has high coupling and low cohesion.
    a. Especially in the playerTick method, making one small change causes multiple errors in other classes, and there is some unrelated code in this method and class that could be somewhere else to make Player more cohesive.
    b. Creating some functions in Background and Board to allow for a tighter coupling. Removed some functions in Player to make the class more cohesive.