Systems Programming Project - Phase II - Garrett Stallsmith

1. What about the paddle? When the ball is in the other process, do you want to allow your player to move the paddle?
    a. When the ball is in the other process, I do want the player to move the paddle.
    b. I believe that maintaining the interactivity even while the ball is on the other side of the court will be a better gameplay experience.
2. What does your program do when the ball is in your court? What does the program do when the ball is in the other court? How does your program make this transition?
    a. I am only drawing one side of the court in each process. This means the ball will not always be visible.
    b. To fix this, I am adding a ball indicator that will follow the vertical position of the ball.
    c. This way the player can keep tabs on where the ball is coming from without my program drawing both sides of the court.
3. How does the ball travel from court to court? What does the other process need to know about the ball? What do you need to know from the other process when it sends the ball back?
    a. Thankfully, I have taken a vector approach to the ball movement. The ball position and its magnitude are handled separately.
    b. Once the ball touches the ‘middle’ court it’s x and y value will be translated to the other side of the court.
    c. It should be easy to access the position of the ball as simple integers.
4. How do you keep score? Now read the RFC to see how the SPPBTP helps make sense of the problems raised by the preceding questions. This program is different from some standard Internet client-server pairs. This program starts off as a clear client-server pair, but then settles into a pair of symmetric programs as they pass the ball back and forth. Both sides can send a ball over the net, and both sides can miss the ball.
    a. Since both processes need to hold game state they will both act as client and server. That is to say sending and receiving information.
    b. In the event a ball is missed, I will have a separate function that updates the game state for both processes.

c. With this approach, I will be able to keep track of the game state in both processes.