**General Overview:**

It’s a two-player game where you play as a paddle and your job is to stop the ball from touching the bottom of the screen. If it does, you loose and the other player wins. When the ball hits the top of the board the speed of the ball increases by 15% until someone loses. Also saves the number of bounces as your score if you win.  
  
**Networked Objects:**

* **Ball:**
  + One of the clients will spawn the ball and it will tell the server whether to pass the turn off to another player or if a player misses the ball and loses when it’s their turn.
* **Paddles:**
  + Each client gets their own paddle and they can move it around a given space. However, they won’t be able to hit the ball until it’s their turn.

**Server Authority:**

The server will be responsible for detecting if a ball is being hit by a player. It’s also responsible for picking which client gets to spawn the ball. The ball will send messages to the server on if to pass a turn to another player or end the game. The ball will pass a turn to another player when the ball hits the top of the board. If the ball hits the bottom of the board, the player that had the turn during that run will lose. The ball increases its speed on the client’s side because of sync issues related to server latency.

**RPCS:**

**Client to Server:**

* PlayerIsReady();
  + Gets called by the Client when it connects to the server.
  + If 2 players are connected and ready the ball will spawn, and the game will start.
* IAmTheBall(int id);
  + Gets called by one of the Clients that spawned the ball.
  + Meant to keep track of the ball’s net id and the Client’s “Sending Id”.
* PassID(int netid);
  + Gets called from Clients own paddles to tell the server what their netId’s are.
* PassTurnFromBall();
  + Gets called from the Ball when it hits the top of the board.
  + Calls PassTurn() on all clients, passing an Id of the player that has that turn.
* EndTheGame(int netid, int bounces);
  + Gets called from the Ball when it hits the bottom of the board.
  + Calls DeclareWinner() on all clients, passing a net Id of the player that lost.
* TransmitMessage(string message);
  + Sends a chat message to the server that was typed out by a client.

**Server to Client(s):**

* UpdateServerStatus(string message);
  + Updates a piece of the Client’s UI to display a server message.
* PassTurn(int id);
  + Updates the material on your paddle to indicate which turn it is.
* SpawnBall();
  + Gets called from the server who randomly picks a host to spawn it.
* UpdateName(string name);
  + Assigns a name to the player when one connects.
* UpdateChatLog(string name, string message);
  + Updates the chat log contained on every client to display a message and a name that was sent from some other client.
* DeclareWinner(int id, int bounces);
  + Display a message on if you’re the winner or not depending on if you net id is the same net id that was pass in. If not then you’re the winner.
  + If you won the number of wall bounces you made counts as your score.
* SetPiece(int type);
  + Gets called from the server to assign a “Piece” to a player.