The software architecture of our project would most closely resemble a client-server architecture. This architecture is chosen because the way it is structured, is the game is hosted on Github or a server somewhere, and a user interacts with it. When the user interacts, the server updates based on the mouse movement, or the left or right arrow, which are inputs sent to the game, and the paddle responds to the user input. The application itself does all the calculations based on the mouse movements, or the arrow movements, updates the ball and paddles' positions on the screen to the user. The main script is where the game is played. It is what pushes all the objects to a game array, and where the main animation loop is. Our setup script is where we declared all of our variables from the html in order to manipulate them in javascript, and initialized anything else that needed to be set up with the canvas, mouse movements, and arrow key movements. The main game animation happens inside of main.js, which is executed when the program is run. It updates each object every frame to account for hits on bricks, hits on the paddle, and the bounces off the walls. It also accounts for the ball's velocities and relative position on the screen. All of the objects scale with respect to the screen size.. This allows the player to resize the screen and still have a game that is playable. We plan on hosting it on Github's servers, so the client will access and play it from the url which it is served from.