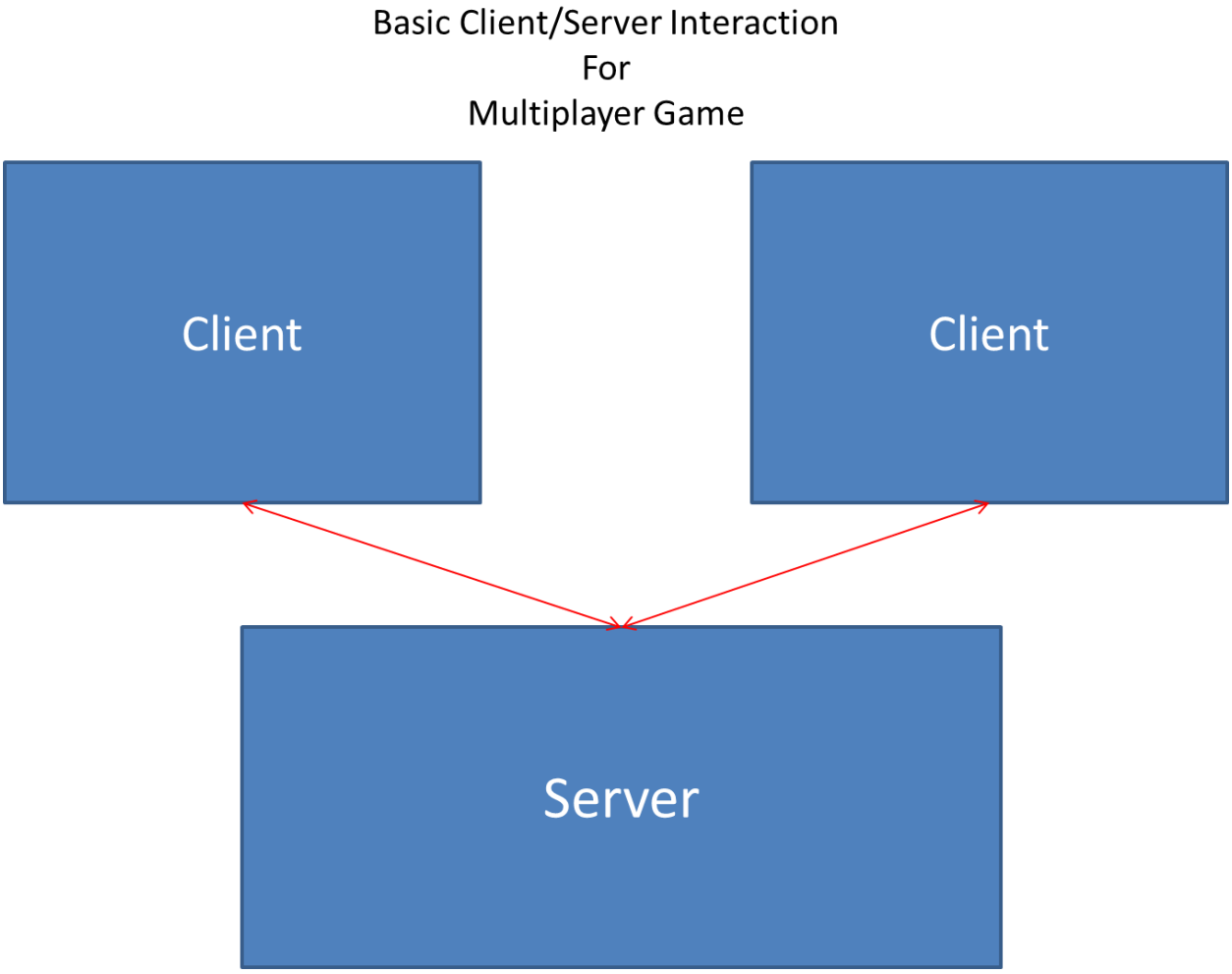
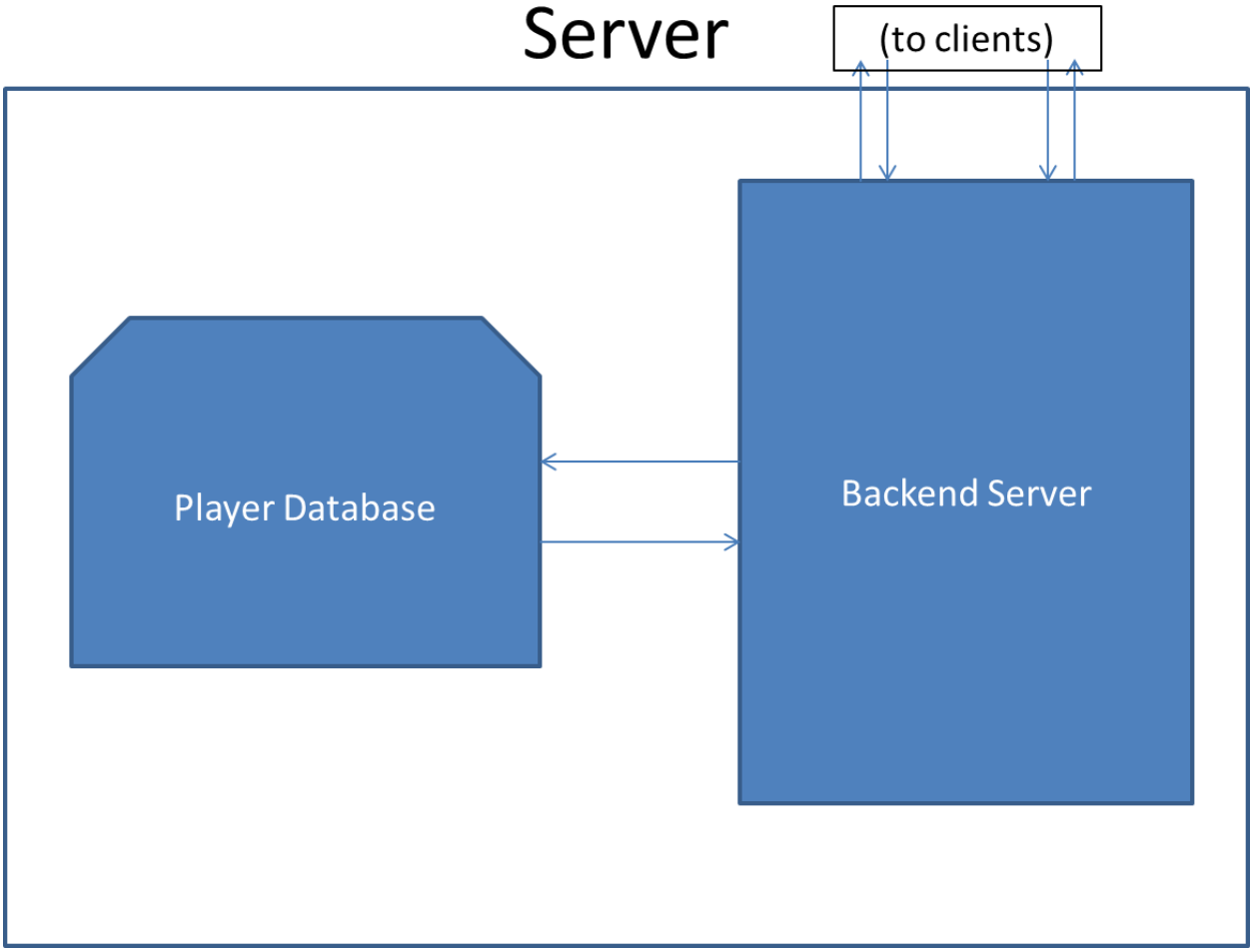
Hunt, Hill, Langham, Lonsway
CSCE 315-503

Space Escape Functional Specification Annotations

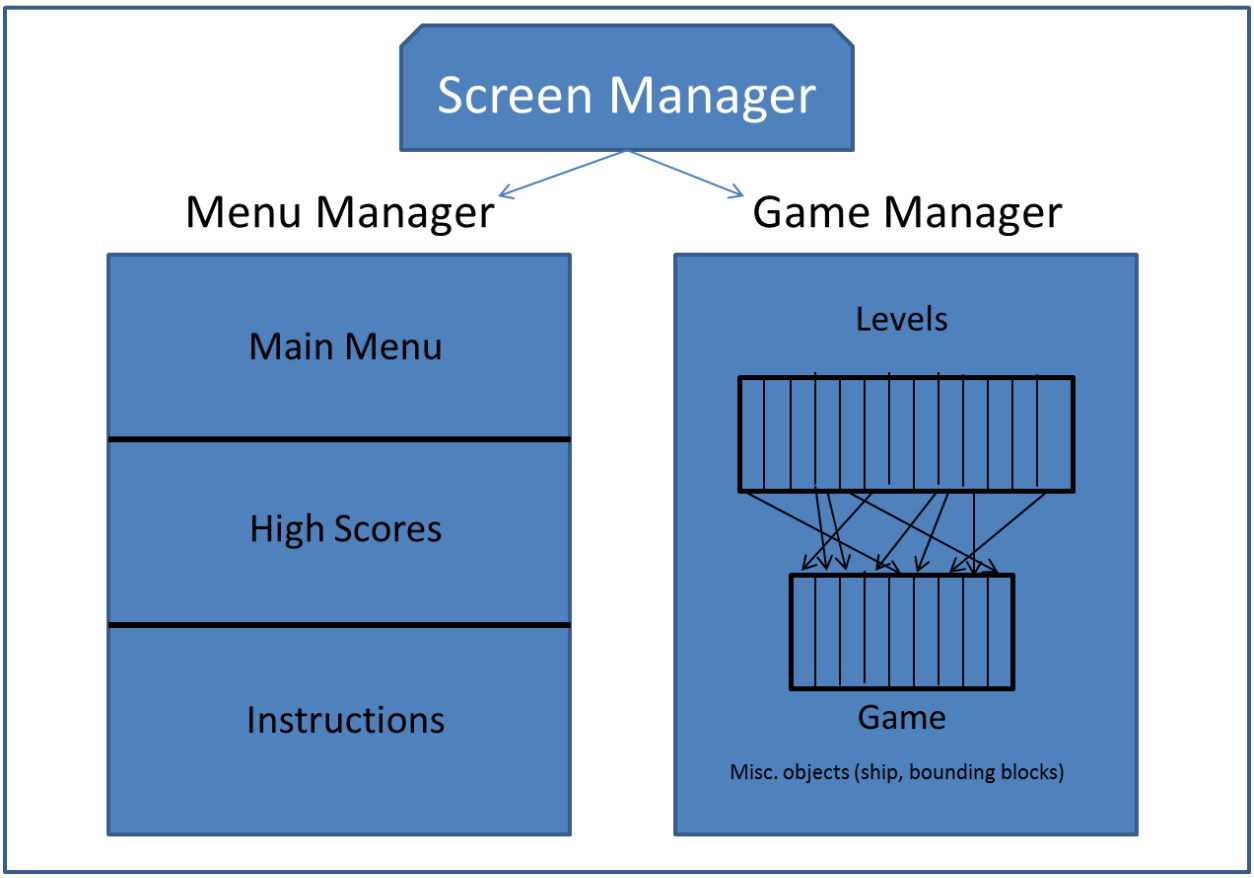


The client and server interact through Index.html. Index.html communicates with the server to hold the names, records, and passwords of people who have played, as well as conveying the current state of the game to all players. Server.js is where the server will keep track of how many people are waiting to play, and match waiting players together into a two-player race. This will all be done through socket.io, and information will be sent as JSON objects.

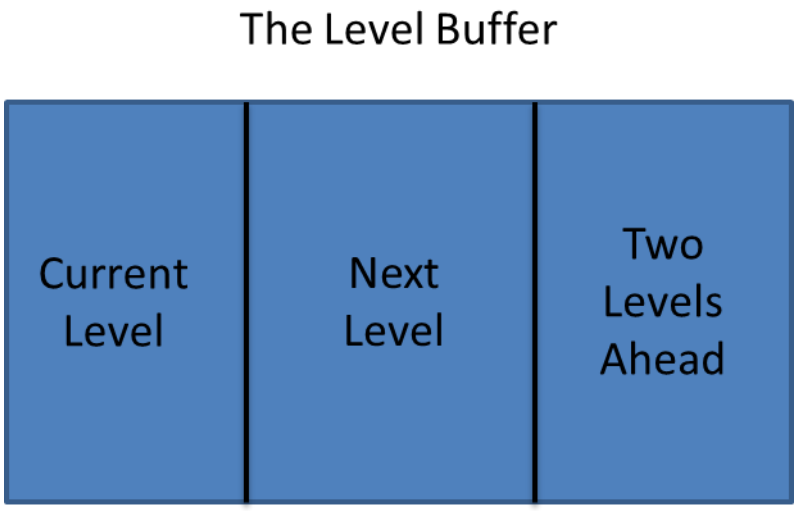


Server.js is where all the information that is relayed to the client is stored, as well as a database of all the players who have played the game. The player database holds a collection of player objects, which are defined in Player.js. The player objects hold usernames, passwords, and high scores for each game mode. The server then communicates the list of high scores to MenuManager.js, and sends information about a player's opponent to GameManager.js and Index.html.

Client

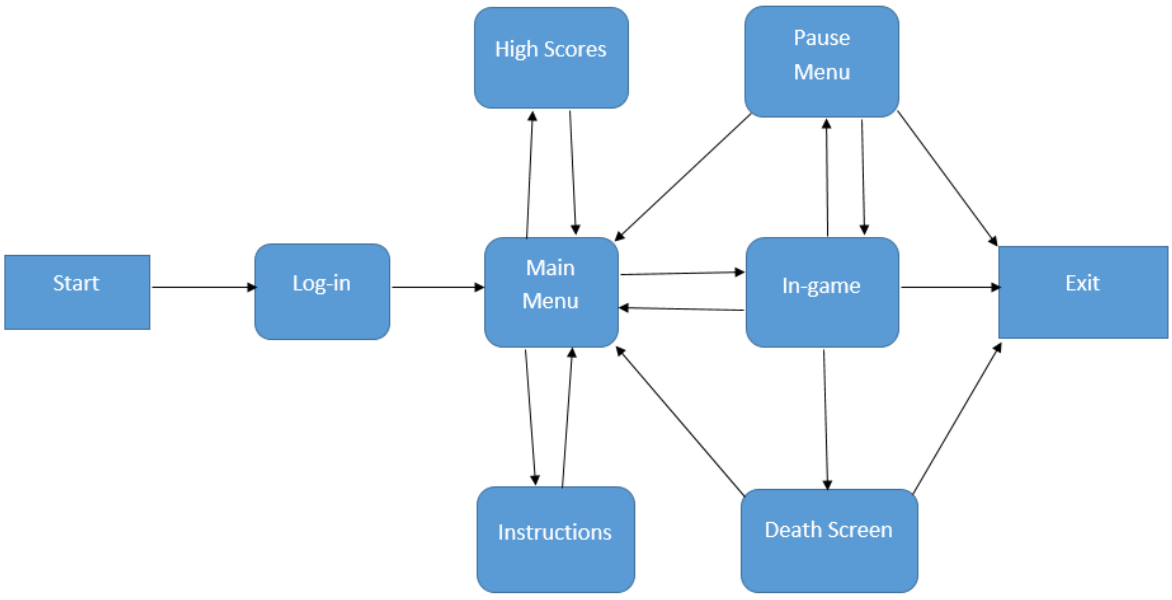


The screen manager is kept in Index.html, which is what the client sees. Information from both MenuManager.js and GameManager.js are sent to Index.html to display to the client, depending on what the client requests. MenuManager.js holds different menus for the Main Menu, Instruction Page, and High Scores list, which will be communicated through Server.js. GameManager.js displays the game to the client, and holds an array of Levels that are passed through Level.js. The array is created by randomly selecting levels from Level.js, and GameManager.js puts them in a buffer. The buffer is then loaded into the game three screens at a time.

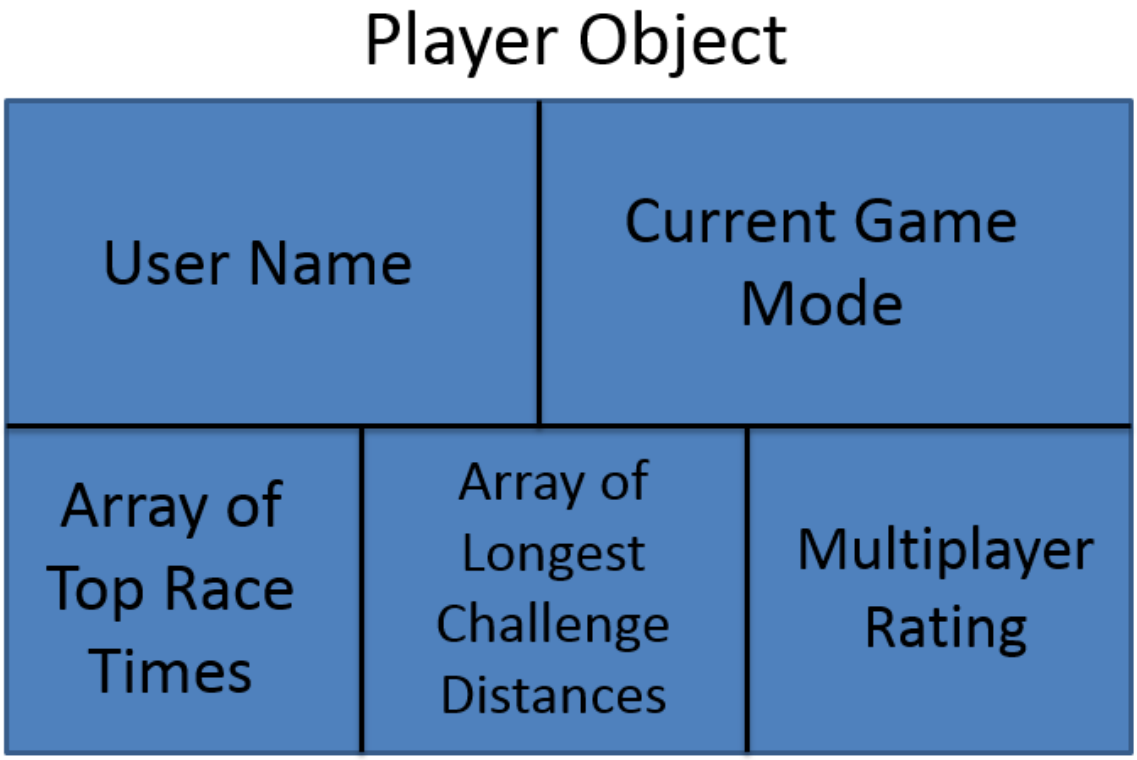


The level buffer is stored in GameManager.js. Depending on which mode is selected, the level buffer is either set at the start of the game, or is randomly generated as the game progresses. The levels are stored individually in Level.js, using a series of blocks that are defined in Block.js. When the levels are randomly generated, the buffer is designed to not repeat levels. The buffer will always start with a blank screen at the start of the game, and the finish line at the end.

Flowchart:



The flowchart above shows the basic layout of our game. The log-in is handled in index.html. The main menu, high scores, and instructions are created and maintained in MenuManager.js. Next, the actual game along with its pause menu and death screen are handled in Game.js and GameManager.js. All of these screens (except the log-in part) are written onto a basic graphics canvas passed in from index.html.



This is a data structure defined in server.js. The purpose of this object type is to hold all of the information for a specific player. The server then maintains an array of these player objects, one for each player. A player object consists of the player's user name, current game mode, array of top ten time trial/race times (one for each difficulty), array of top 10 challenge mode scores, and the player's multiplayer rating. The user name is set at the player's first log-in, the game mode is set every time the user starts a new game, and the other three are updated after every game.