Nick Greenquist
MVC Documentation

Multi Quest: MVC Functionality

What your site does and what its purpose.

My site is a canvas game called Multi-Quest. It is a multiplayer side scrolling game where players have to travel as far forward as they can, defeating monsters along the way. The site also have a login system. Users can create their own accounts, and when they join a game, their username in-game is the same as their real username. Also, they can view their stats and create weapons. The overall purpose of the site is to be a fun relaxing game.

How you used MVC and which framework you used

I used Model-View-Controller to handle the user account information and also Weapon Information. The MVC code for this site was taken from Domo Maker. The Domo controller essentially was reused to handle the main 'maker' page (in this site it is called the Character Screen). It handled routing Account info to the html pages after login or signup. The weapon page was created to replace the actual Domo logic. It has its own controller and model. The controller handles the creation and deletion of weapons from an account. The model has just a simple name and attack stat. There is also a Game controller. There is no model for game but one might be implemented in the future. The controller simply handles routing the site to the game url and passing in the username variable. The Leader controller again has no model but has a controller that handles routing the site to the leader board page. The controller is responsible for calling a sorted query on the mongo database to return all users in ranked order by their maxDistance.

I used Express to handle organization of the MVC. It helped with routing and structuring the controllers and models.

What you used Mongo for (what did you store/retrieve)

I used Mongo to store data about user accounts and also weapons they make. Account data has more than just a username and password. It also has all changeable stats that their characters would require inside the game. For instance, level, maxDistance, attack, speed, etc. In the future, I will move out that data into a separate Character collection, but for now, the info is stored inside Account.

I also store Weapon data. On the character screen, the user can create and delete weapons for their characters. This right now does not affect game play but I plan on implementing something similar. Users collect exp from the game and will be able to choose to level up their spells, increase stats, or create more powerful weapons. Right now, I simply store and retrieve the weapons when needed.

How you used a templating language and which templating language you used

I used Handlebars for my templating language. It was extremely useful for displaying all the data I retrieved from the database. For example, on the character screen, I display the user’s stats and below that all the weapons they could possibly have. Handlebars made this easy, specifically looping through all the weapons returned by the controller and displaying the variables.

On top of displaying weapons. Handlebars also made it easy to transition relevant variables efficiently for the character screen and leaderboard pages. I could navigate to both seamlessly and displaying the information was made easy with handlebars.