Introduction of Team

Joshua Rao ([josh1996@tamu.edu](mailto:josh1996@tamu.edu))

Location of Project

My entire project can be played on the projects server that we have been using to host the weekly assignments. The address is <http://projects.cse.tamu.edu/josh1996/OregonTrail/index.html>

The link to my youtube video is - https://www.youtube.com/watch?v=7hJAbMaeyag&feature=youtu.be

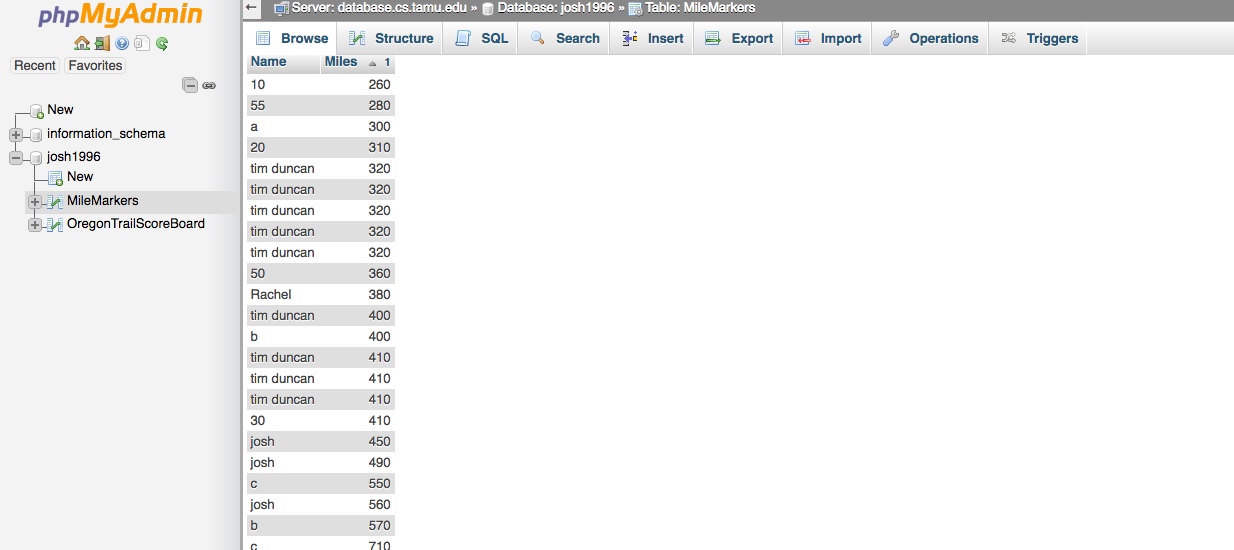
URL – <http://projects.cse.tamu.edu/josh1996/OregonTrail/index.html> (AFTER SUBMISSION)

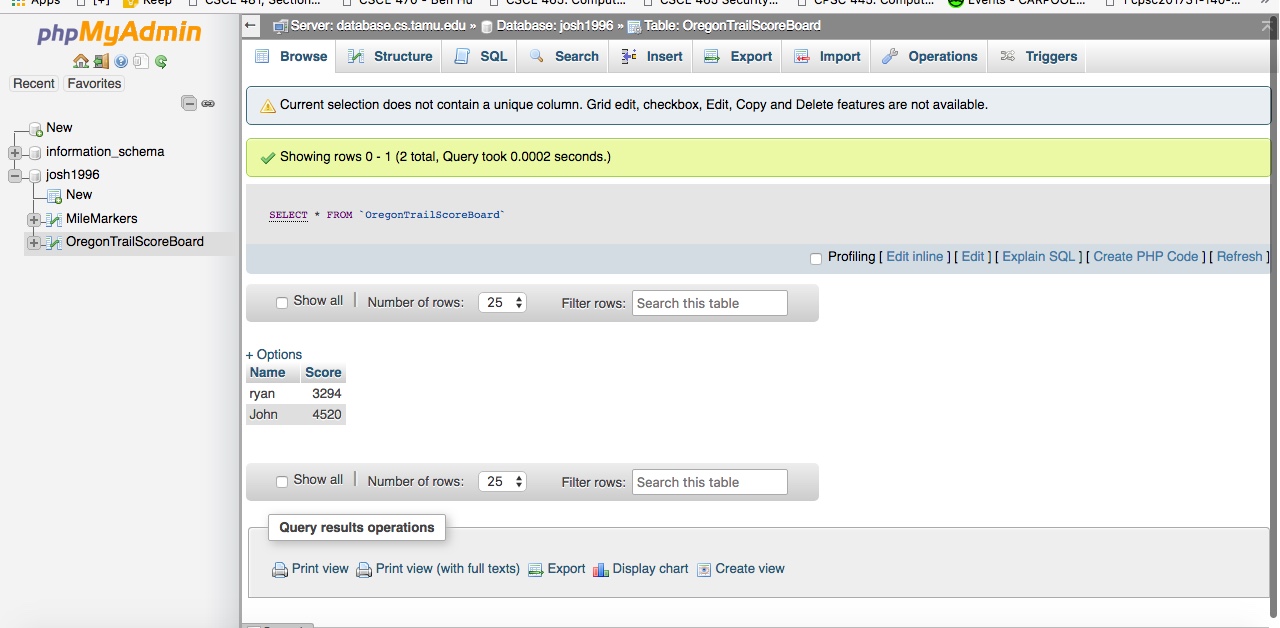
Project Description

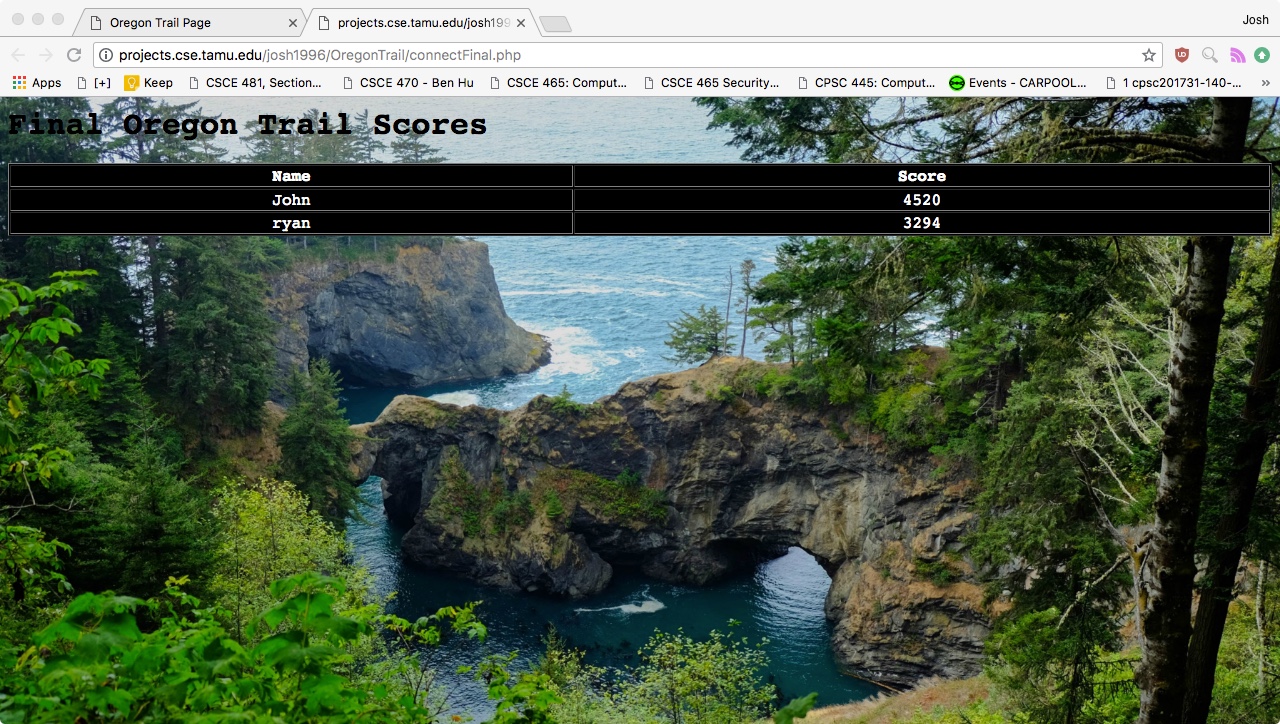
The majority of my project is composed of Javascript functions. The HTML and CSS is used sparingly and just as style or dimension changes for divs, paragraphs, or images that are accessed by Javascript functions. My project is a fairly accurate representation of the original oregon trail game with additional tracking of people who have previously died playing the game as well as a final scoreboard. Both the tracking and scoreboard are accomplished by accusing our TAMU database using the methods that we have learned in class with php. The game can be replayed multiple times and provide vastly different experiences every time. The use of random numbers was crucial in making the game more like the Oregon Trail game that I grew up playing. Everything from getting a disease to picking up fruit is determined by random. Events like contracting a disease or dying are more likely if food is low but these experiences are still random and not guaranteed. The project is composed of one HTML page, two CSS pages, one JS page, four short php scripts for people dying and accessing the final scoreboard, and an images folder. The user has every option available in the actual oregon trail game. Each day is incremented and all of the variables such as food, health, and money are constantly being changed by javascript functions that are working in the background.

Database Setup

I have two databases in my project which are both found under my tamu database account (phpMyAdmin). There is one named “MileMarkers” where every death is submitted through a ‘insert into MileMarkers (Name, Miles) values ('$name', $mile)’ with the appropriate values for ‘$name’ and ‘$mile’. Once someone is dead a javascript function executes a php script which contains the name of the member who died and how many miles they traveled. The php script has an ‘insert’ statement which inserts a name and mile number for the ‘name’ and ‘mile’ columns in the table. Additionally, this database is used to keep track of those who have died in previous plays of the game. The database is constantly accessed every new day and if you are approaching a mile marker where someone has died a gravestone will appear for one day and then disappear. If someone beats the game and reaches Oregon City then they are inserted into the “OregonTrailScoreBoard” database. I have also attached a shot of what the final scoreboard looks like. This scoreboard is displayed by executing a ‘SELECT \* FROM OregonTrailScoreBoard ORDER BY Score DESC’ SQL command.







Languages Used

JavaScript

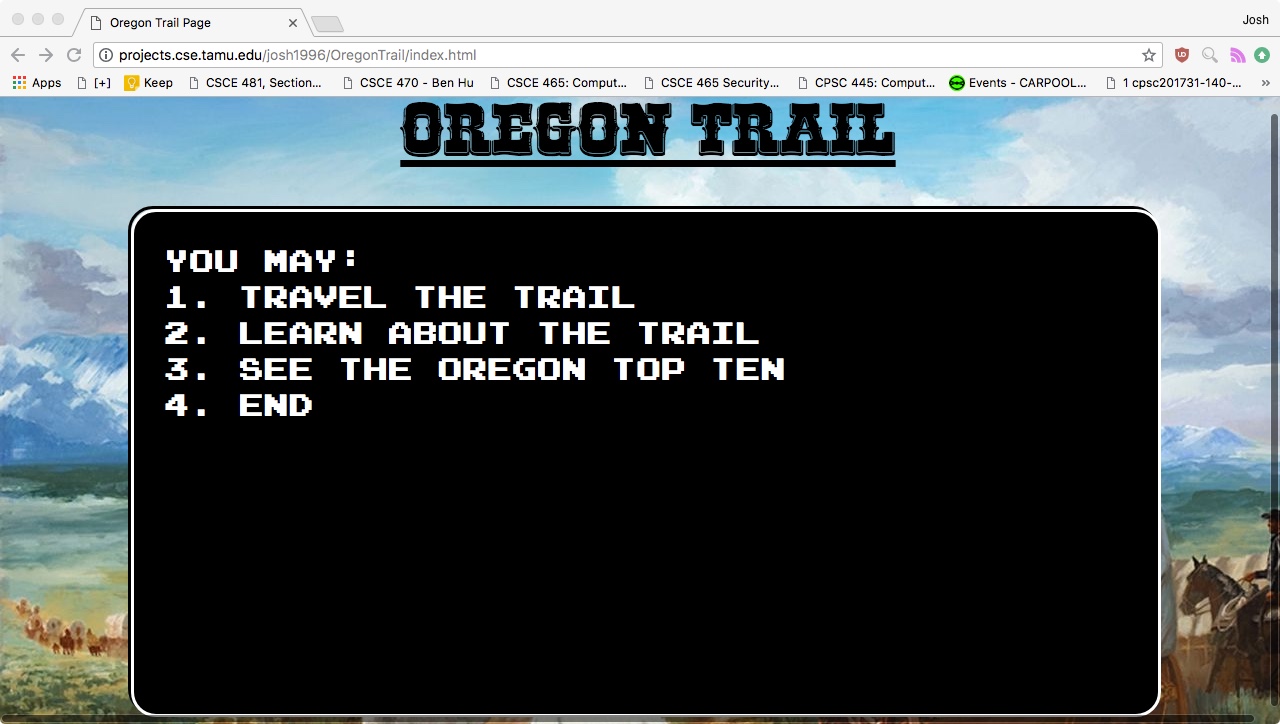
Javascript was used for a majority of the interface and background functions. Everything from the text displayed on the screen to the animations of the oxen moving along the trails and rivers was done using Javascript. The positions of these objects were constantly incremented in order to get the look of it correct. The user also has many things with them such as money, oxen, food, etc. All of these items are contained in javascript objects and the calculations for them was the user gains and loses a number of each is done using javascript functions. Javascript moves the animations along the screen as well as making sure that nothing that wouldn’t happen in real life doesn’t happen in the game. Such as someone traveling with no oxen but they still have a wagon or that someone has negative money.

PHP

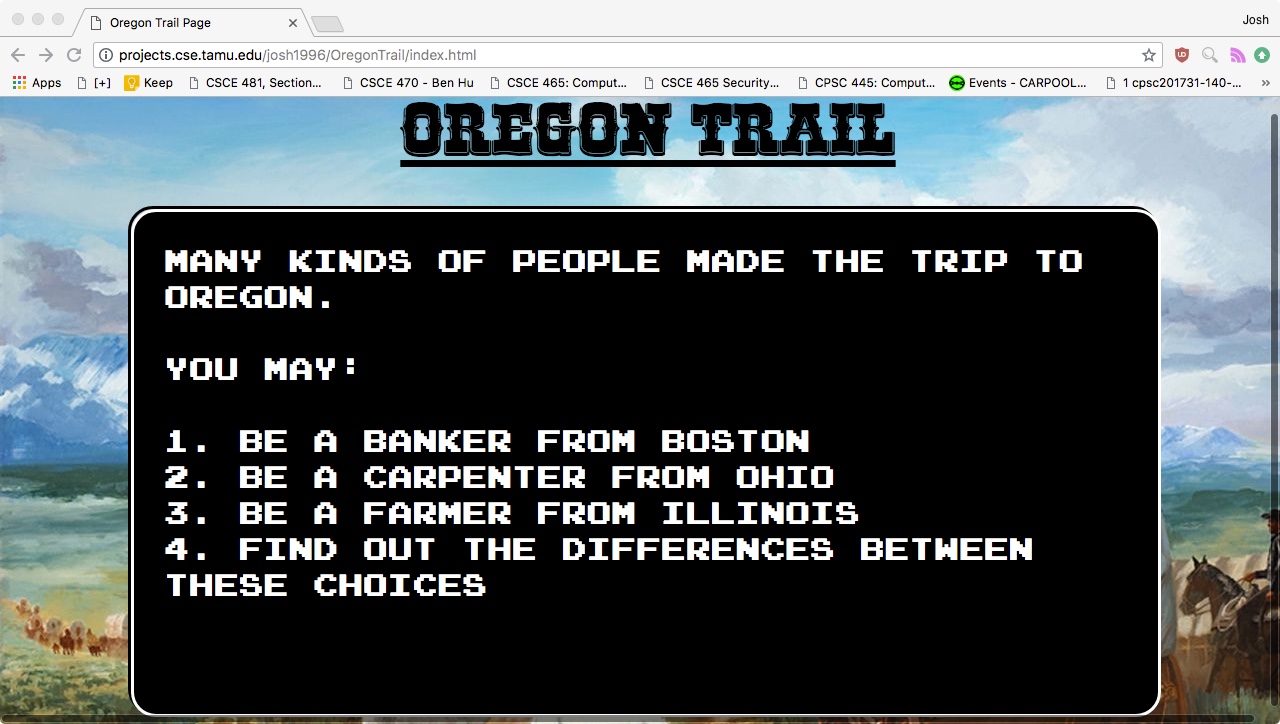
PHP was used to send information to the two databases and get information from them. PHP sends an ‘insert’ statement with a player’s name and how far they got in the game (miles) to the MilesMarker database and sends an ‘insert’ statement to the OregonTrailScoreBoard database with the name of the player and how many points they have earned if they win the game. Additionally, while you are playing the game, PHP constantly checks the database to see if you are about to approach a mile where someone else has died in a previous play of the game. If you are then you are shown a grave indicate a mile marker for the death of someone previously playing the game. The fourth PHP script is used to query the OregonTrailScoreBoard database and show the results ordered by highest score to lowest score.

Slick Sheet

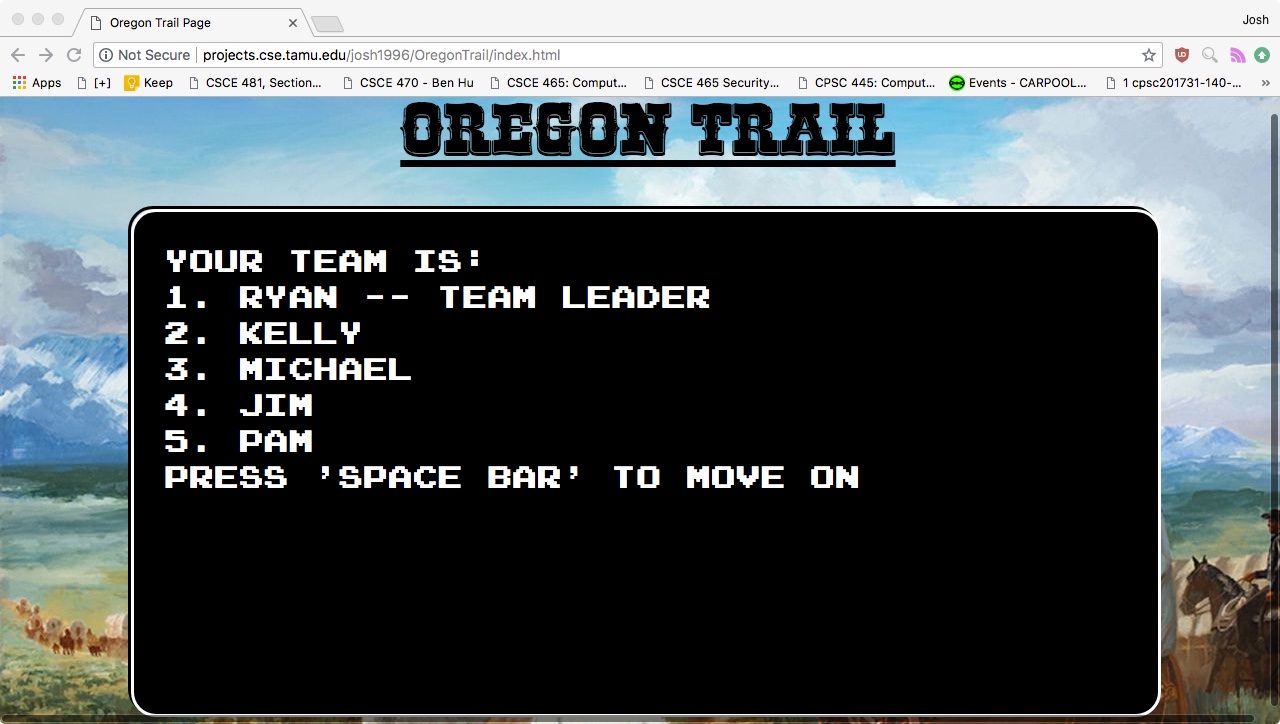
Main Menu - The first page that you see \*\*\*All of these options on each menu will be shown in my video



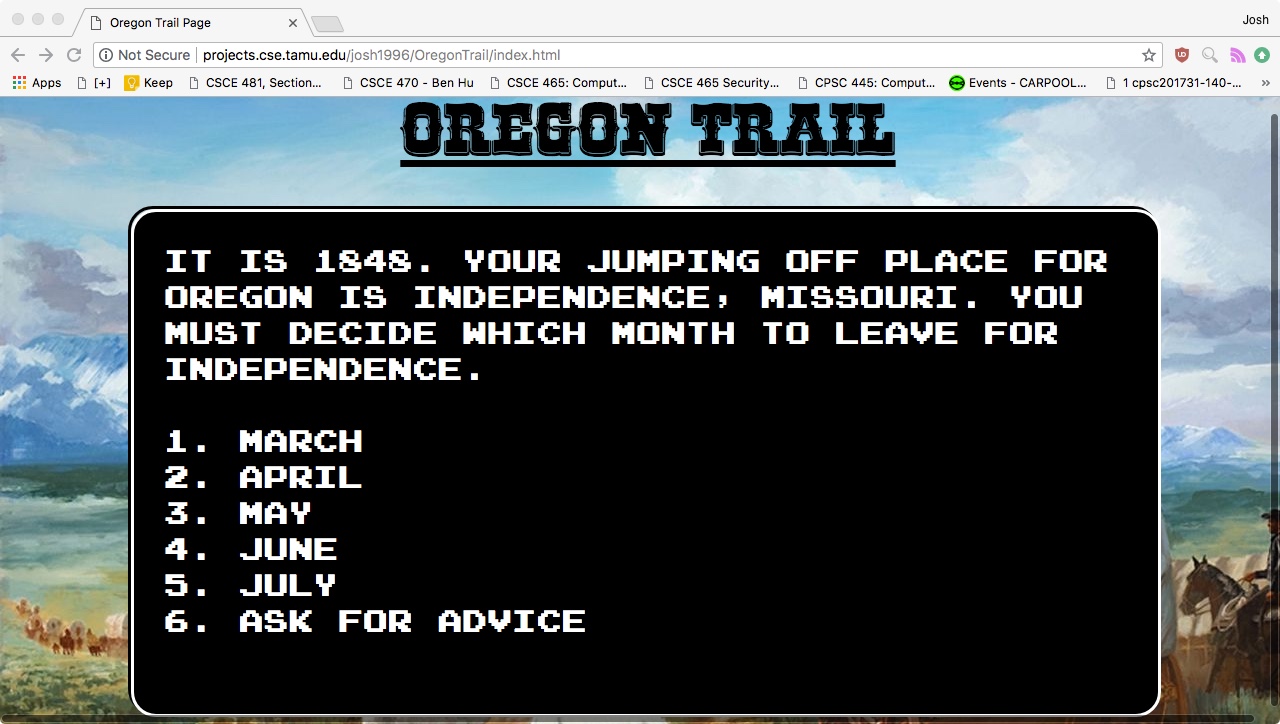
Choosing if you want to be a banker, carpenter, or farmer. Bankers start with $800, carpenters start with $600, and farmers start with $400.



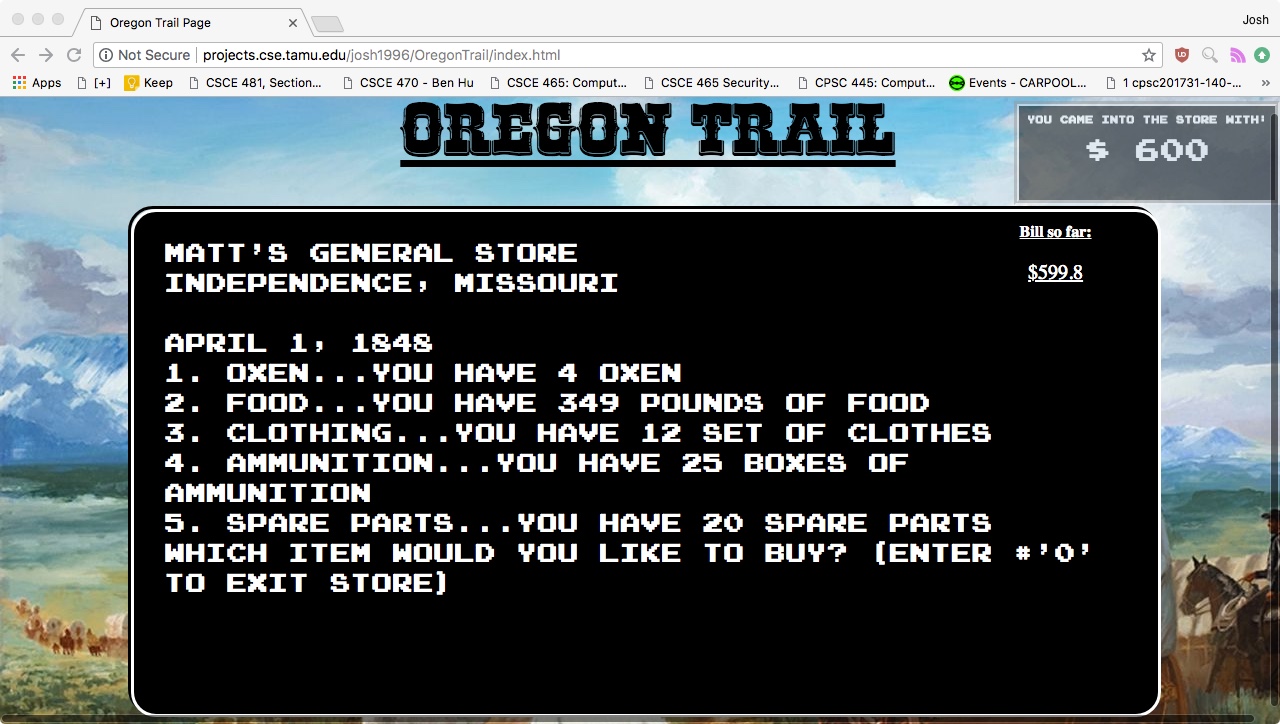
You are shown your team after you personally type in the name of each one.



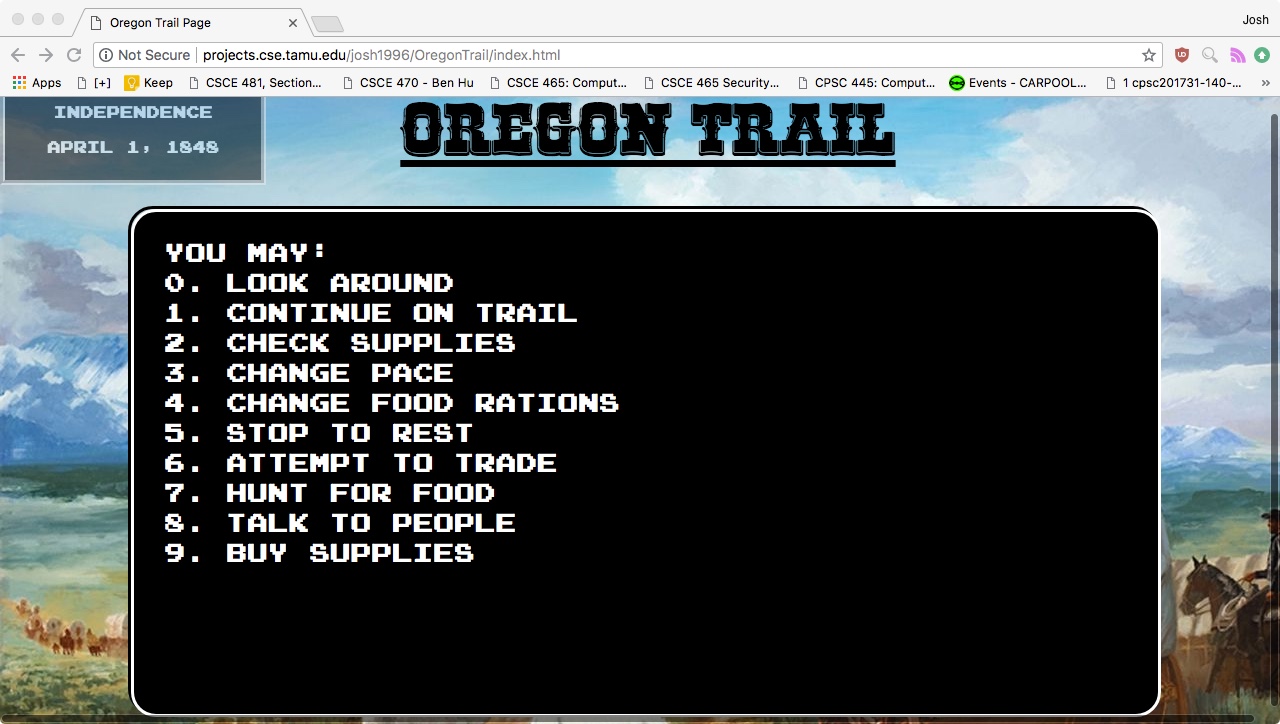
Choosing which month of the year that you want to start the game on



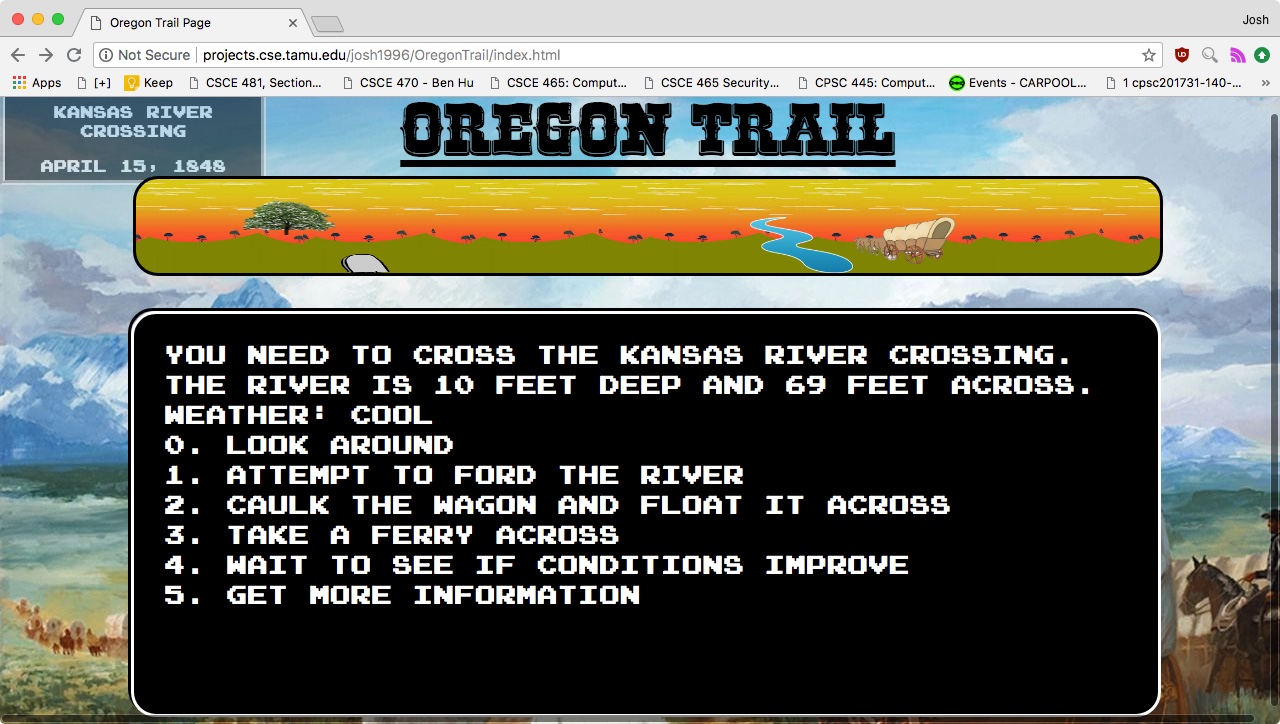
I made some selections from the store at the beginning of the game and show them here. Notice that you are told how much money you walked into the store with and how much your current bill is. The current bill is changed every time you make a purchase.



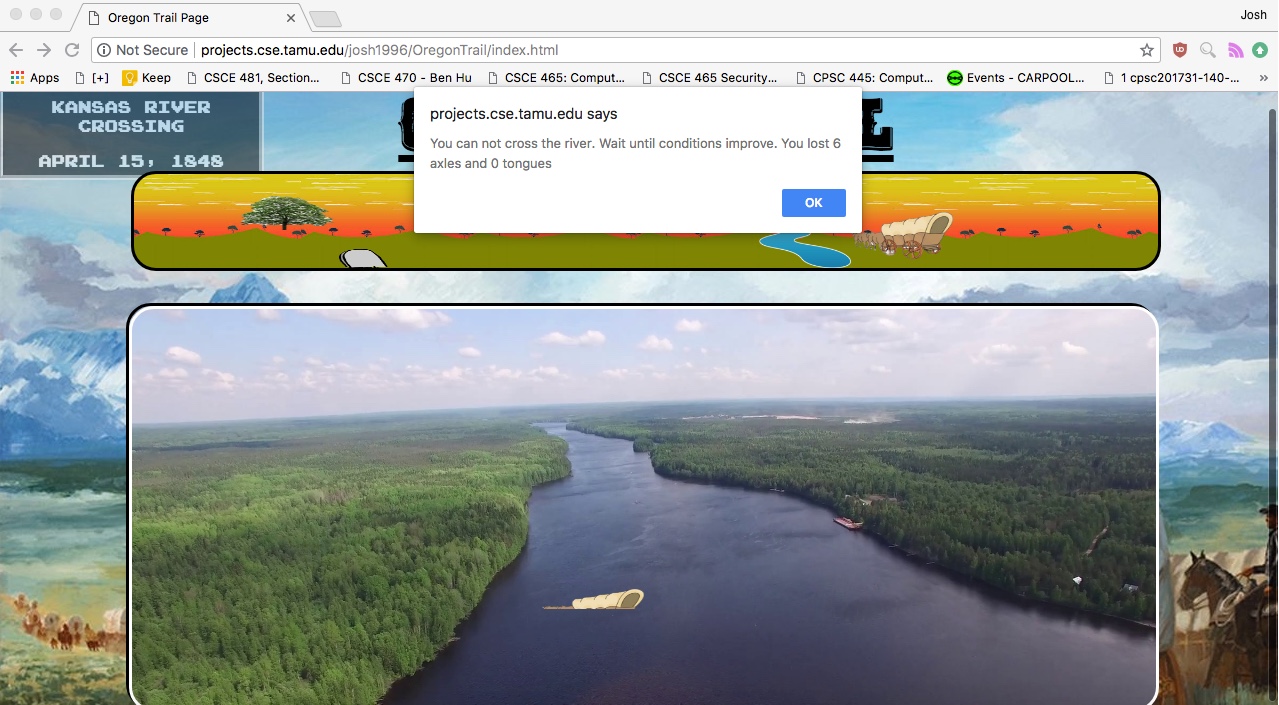
The first day on the oregon trail. The options will be explored in the video.



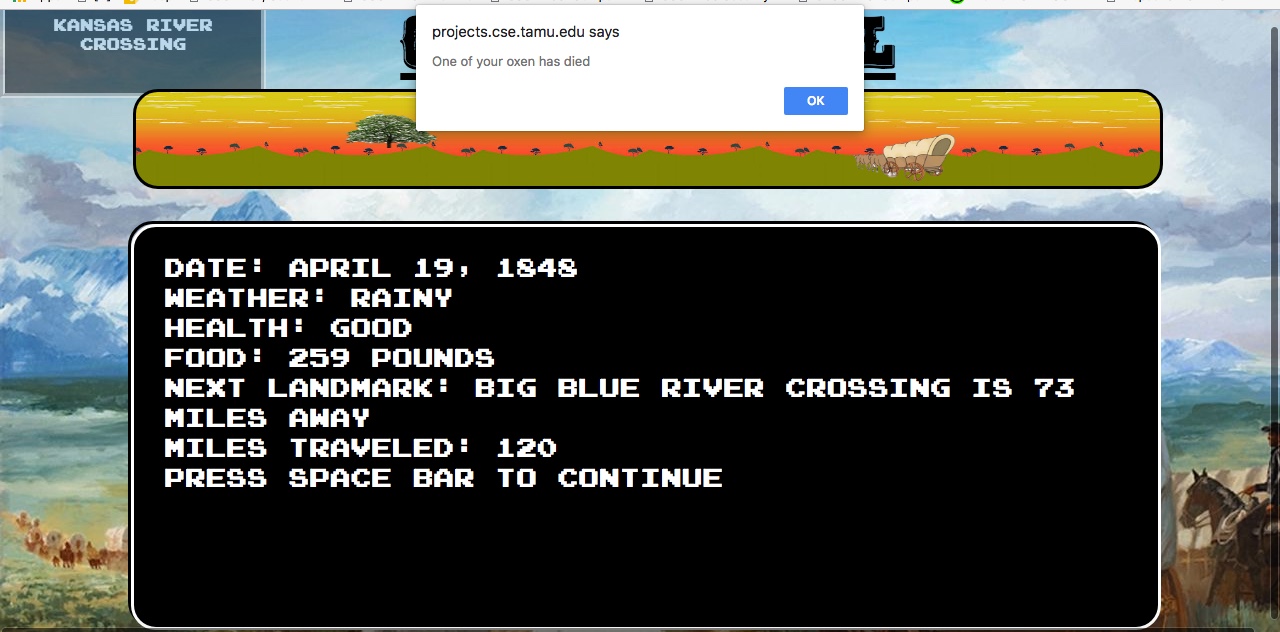
When you press ‘1’ and continue on the trail then you are shown a little animation window which has an oxen with a wagon moving forward. Trees constantly move to the right in order to better show a team moving along the oregon trail and the images for rivers, forts, and natural landmarks approach you when they are an appropriate distance away from you.



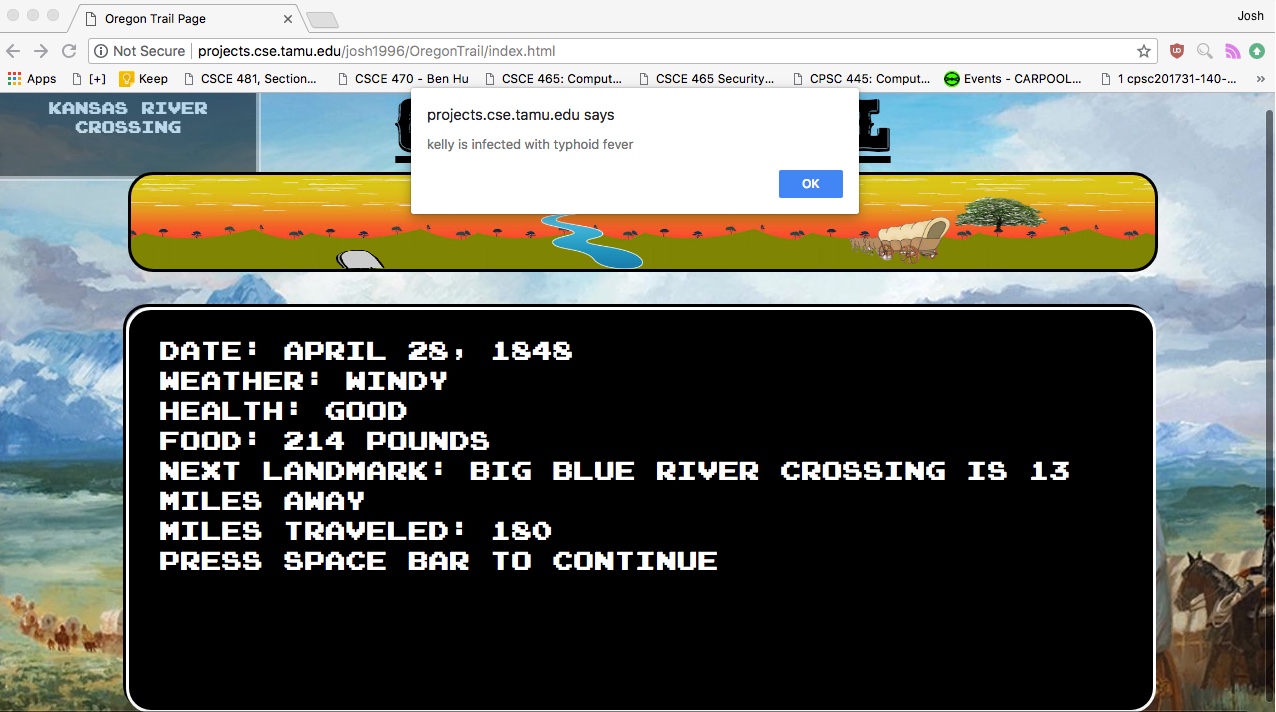
When you cannot cross a river (because of weather or just random bad luck) you are sent back to the page shown in the previous pic. The picture of the wagon that you see here is cut in half at the very end of the crossing in order to illustrate a wagon sinking like in the real game. For the rest of the crossing before this image you are shown a whole wagon so you do not know if you will sink or not.



Just an example of the random things that can happen to you in the game. Other things include gaining fruit, oxen wandering off and losing a few days, getting stuck in a ditch and losing a day, one of your oxen dying, and bad weather causing you to lose random objects such as wheels or food.



You are also notified if a player is sick. After you press ‘OK’ in the picture underneath, you are given the option of resting for 3 days and having the player recover or you could press ‘cancel’ and move on without losing any days but you now have a sick member.



Thoughtful Add-Ons

When you are entering in names and numbers in the beginning menus you cannot enter in any invalid entries. Such as empty spaces for a name or letters for the number of oxen you want. I catch these errors and alert you that you have to submit the correct input or else you cannot move on. The game does not crash when someone does this. Additionally, I have decided to work with onkeypress() functions a great deal with the menus instead of having the user type in the number that they want in a menu. I believe that this is more intuitive and a user would appreciate this. For the end of the game, I have a congratulations screen which shows you that you won if you have $100 and selected to take the toll road. If you can make it past the columbia river then I have the same congratulations screen which moves to your final score. I just made it random depending on how much money you have left if you choose to cross the Columbia River. You may not make it and die at the very end if you choose to cross and then the game randomly decides that you did not make it across such as in the river crossings shown previously in the game. I also used alert boxes throughout the game and I like these because it stops the game and allows the user to understand what is happening. The next landmark image changes depending on if it is a fort, natural landmark, or river.