Project 2

Due: May 13rd by 11:59 EDT

# Background

Your team project is to make a web replica of the original game Oregon Trail. (found here: <https://archive.org/details/msdos_Oregon_Trail_The_1990>)

# Assignment

The project is to make this game web-based using JavaScript, PHP, AJAX, and MySQL. How you use each will be up to the team design, but at the very least, to keep the high score should be using MySQL. There will be a ton of programming just the overall game logic. This should be taken care of by JavaScript, PHP, etc…

In one departure from the original, there is an option in the game to hunt for food. This would be difficult to render in JavaScript, so to simplify have the leader of the group go fishing and give random chances on if a fish is caught, and how heavy it is.

Second departure “can” be the graphics. They do not have to be the same hideous graphics of the DOS game. You can update, but should be in the same theme.

Third, for those that were brand new to the game, some helpful hint (especially when buying or just starting to travel) would be suggested.

# Code Requirements

If you play the game long enough, almost everything can be stored in objects. Use the DB sparingly, for example storing your high score.

Random also plays an important and significant role in this game. Make sure your team can fully identify where.

Finally, nothing should be “pre-bundled” package. This should be a custom creation. If there is a question, consult with your instructor.

# Grade Breakdown

[Video Presentation](#_Video_Presentation) …….……………….……………….……………….……………….……………….……………… 5%

[Documentation](#_Documentation) ……………….……………….……………….……………….……………….………….……………….15%

[Ease of Use](#_Ease_of_Use) ……………….……………….……………….……………….……………….……………….…………………25%

Attractiveness (no explanation needed) ………………….……….……………….……………….…………..20%

[Code](#_Code) ……………….……………….……………….……………….……………….……………………………………….….25%

Team Evaluation ………………………………….……………….……………….……………….……………………....20%

[Thoughtful Add-ons](#_Thoughtful_Add-ons)……………….……………….……………….……………….……………….……………………+10%

# Table Top Presentation

The team will demonstrate their project to all students. The team will be assigned a table where they will demonstrate their projects on the 3rd floor of ITE to student, faculty and staff that walk by. The table will need to be constantly manned throughout the day, by at least one person on the team. (need to reword). How nice your demonstration and table is will be graded by your instructor and volunteering staff.

# Documentation

You must include a documentation file named “README” (with appropriate extension, .docx, .pdf, etc). The file **must** **be editable**. Optionally, a PDF version can be included as well. Your documentation should include **text and screenshots** of your project.

The documentation must contain, **in order**:

1. Introduction of Team or self

a. Names

b. Emails

c. Roles of each member

2. Location of Project

a. How do I (instructor) get to your project

3. Project Description

# Ease of Use

Your project will be graded based on a visual, informational, logical, and mechanical diagnostic examination. Ease of use will be graded by the client. Use a [Website Evaluation Rubric (from Uni. of AZ)](http://www.u.arizona.edu/~awkarlow/files/Rubric.pdf) as a guide. I suggest that you use the “Larry Rule” (coined by Lupoli) and have someone else (aka parent) to try and see how they like the client before you make the final touches.

# Code

Standard coding practices from CMSC 201-341 apply. All code must be original and custom-written. There are many nice “additions” that are easy to download and install (we’ve seen a lot of them), but you don’t learn from it. **IF**, you decide to use aesthetic and/or coding designs from other websites, **make sure you cite them in the documentation and in your commented code**. Failure to do this will result in a zero in this portion of your grade.

How your CSS/JavaScript images were included and organized will be reviewed. All code must be submitted as instructed in the main project documentation. All submitted code must follow the rubric below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Condensed  Conclusion | Very Good | Good | Okay | Less than Okay |
| Code Organization | -Meaningful file/folder/directory usage  -Encapsulation of functionality code utilized  - Very well thought out separation to specific parts of code | -Adequate file/folder/directory usage  -Separation of code into various folders used  - Logical pathways to specific parts of code | -Lack of separation for code’s functionalities  -Ideas were present for coding placement | - Non-meaningful file/folder/directory usage  - Code appears forced together without much care  - No thought process made into separating code based on its functionality |

# Thoughtful Add-ons

The thoughtful add-ons is the wildcard portion of your rubric. This is not a requirement. However, based on your effort and how much the add-ons help your overall application, reasonable points will be added to your overall score. The add-ons should be well-documented in both your documentation and presentation on video. There is a maximum to the amount of points that you can receive for your thoughtful add-ons. (Please reference the rubric on how many points are available.)

# Submitting your project

You should submit the following files within a .zip file:

* proj2.html
* proj2.css
* proj2.js
* video.txt – file with URL link to your video (YouTube)

Starting from the same directory as your .zip file:

cp proj2.zip /afs/umbc.edu/users/s/l/slupoli/pub/cs443/USERNAME/Project2

or

submit cs433\_lupoli proj2.zip (check that this is available)