Game Plan Instructions

The purpose of your Game Plan is break down your intended JavaScript Game into manageable, known, chunks.

This serves many purposes, including:

- allowing me to assess and approve your undertaking

- sparing you from having to maintain a mental map of the game as a whole

- giving you a roadmap towards successful completion

It should contain the following:

- A high-level description of what your game will be.

(Settlers of Cataan is a 2-4 player strategy game... etc.)

- A high level/rough design statement

(Old west themed, sepia-toned, fixed layout @ 1600px x 900px, etc.)

- A play-by-play description of the user experience

(Each player enters their name. Game play begins with each player placing 2 settlements on the game board, etc.)

- A list of the (anticipated) game elements

(The equivalent of the pieces in a board game... what's in the box.)

- A list of the user actions

(Enter player name, place a piece on the board, move a piece, submit a guess, quit game, play again etc.)

- A list of the system actions (functions)

(Capture and display the player name(s), switch screens, start/stop timer, score points, switch player etc.)

In terms of how to organize your game plan, the goal is to separate out the UX, design concept and the functionality:

Play-by-play should be written from the perspective of what a user will see and do, from the beginning of the experience right up until the end. It should not include any information on how you will implement it, just WHAT the game experience is.

Separate from that, I want you to list out WHAT the individual elements of the game will be (visual elements, audio, etc. - these will ultimately be your HTML elements). Give your elements meaningful names. (You may well end up using those names as the ids and/or classes for your HTML).

Separate, yet again, I want you to list out WHAT each bit of functionality is that your game will need to have in order to make that user experience happen. Give each bit of functionality a meaningful name. (Again, you might end up using these names as the names of your actual functions).