Heuristic evaluation – Nielsen’s 10 Usability Heuristics

Nielsen's heuristics

**Visibility of System Status**

This system tackles this issue very efficiently. It has updates every 30 seconds of the left hand of the screen of what is going on in the backend and the current environment status. It informs of the strength of the fire, the warmth of the room. In addition, the backend count of the users resources (in this case, wood). As you gain resources new options become available and they are updated as you reach a certain threshold. In addition to this, there are backend cool downs for each of the buttons, and they are displayed by a slowly fading gray bar on the buttons. When you save the game there is a little alert that pops up in the corner that lets you know an action has been performed

**Match between system and real world**

The system talks in simple terms, it makes it very obvious what each button on the webpage does. The only ambiguity is when the first new tab pops up it’s not really obvious what it does. It also has plain and simple terms about what is going on in the system.

**User control and freedom**

This system does not provide a great way to revert actions, unless you are constantly saving. While the game does have auto saving after set intervals, there is no other way to “undo” an action other than restarting the game.

**Consistency and standards**

The user most definitely does not have to wonder if multiple things have the same meaning. All of the buttons are unique. The different situations that arise are shown in alerts and they are unique situations. Although most of them have similar requirements (giving some sort of resource to do something else) they are all unique in their reward.

**Error Prevention**

The game does not have very much error prevention. That being said, I did not run into any errors while playing around with the program.

**Recognition rather than recall**

Implementing the game with buttons rather than making it a text based typing adventure strongly pushes for recognition rather than recall. It also has a brief history on the left of recent events that have taken place so the user can remember if they have recently done something by seeing it rather than just from having to memorize. The cost of all of the building appears when you scroll over each button so that is also pushing for recognition over recall.

**Flexibility and efficiency of Use**

The program is simple enough that there isn’t really any need for there to be ways for the user to perform frequent actions, other than to stroke the fire, which is the button that is always available unless the fire has gone out.

**Aesthetic and Minimalist Design**

The text that is shown is not all necessary, but it pushing the storytelling of the game further. So while it is not minimalist, it is a great addition.

**Help users recognize, diagnose, and recover from errors**

There was no error reporting that I could find, so I was not able to discover if it would report or help the user discover how to revert the error. Although going through the code I did not find any error catching.

**Help and documentation**

This is the one major flaw of the program is that there is no documentation on the user end. If you are working with the code, there is sufficient documentation that lets you know what does what, but in the application the only way to know what something does is to read the text on it (which is normally good enough) or to try to click it and test it.