Web Development – Mr. Turner

Project – The Crossing

Project Overview

Three humans and three of the deadly blomfargles are trapped on one side of a huge precipice. They have a hover car, which can carry 2 of them across at a time, and have agreed to work together to get everyone across. The trouble is, if the blomfargles ever outnumber the humans on either side, they will eat them. Can the user get all 6 across the precipice without letting the blomfargles eat the humans?

Display

The list below includes the essential elements of the page.

- The page can be laid out either vertically or horizontally, with the precipice separating the two halves of the board.
- 6 images should represent the blomfargles and the humans.
- 1 image should represent the hover car.
- The user will interact with the game by clicking on the humans and blomfargles to move them into and out of the hover car.
- The user can move the hover car from one side of the precipice to the other by clicking on it.
- There should be a log of actions taken by the user.
- There should be a reset button to reset the game.

Functionality

The program will need to keep track of how many humans and blomfargles are on each side of the precipice. It will begin with all 6 on the beginning side.

The program will need to keep track of the hover car's position. It will start on the beginning side.

The program will need to keep track of who is in the hover car. Remember that the hover car can only hold 2 "people", regardless of the human/blomfargle combination.

People inside the hover car count as being on the side of the precipice currently occupied by the hover car.

When the user clicks on one of the people, that person will be moved from the precipice to the hover car or from the hover car to its side of the precipice. If moving someone into the hover car and the hover car is already full (2 people) or not there, the clicked person will not move.

When the user clicks the hover car, it will switch sides of the precipice. The hover car cannot move without a driver, so there must be at least one person in the hover car in order for it to move.

If, at any time, the blomfargles outnumber the humans, the game ends badly because the humans are eaten. Remember that being in the hover car counts as being on the side of the precipice occupied by the hover car so adding to or removing from the hover car doesn't change that balance. Only moving the hover car changes the balance.

If all 6 people get to the far side of the precipice, the game is over and the user has won.

Enhancements

- The Spider
    - On the planet, there is a giant spider that likes to eat humans and blomfargles. It generally keeps to the jungle and it is terrified of the hover car, but there is a chance it may appear.
        - Every time the user clicks, there is a 2% change the spider will appear.
        - The spider will always appear on the side of the precipice ***not*** occupied by the hover car.
        - The spider will eat a random person on that side of the precipice. It doesn't care which. Humans and blomfargles all taste the same.
- New recruits
    - Lots of ships crash land on this planet and there are straggling humans and blomfargles all over the place. One may decide to join the party.
    - Every time the hover car moves, there is a 5% chance that another castaway will hear it and join you.
    - A new recruit will always join the party on the beginning side of the precipice.
    - A new recruit cannot be moved into the hover car on the turn it arrives.
    - If the new recruit should cause the humans to be outnumbered by the blomfargles, nothing will happen until the next time the hover car moves.

Necessary Programming Skills

- Comprehension of the specifications sheet.
- Design Document
    - Figure out the information you need to keep track of.
        - This information will become your global variables.
    - Plan out the individual tasks your program must perform.
        - Think through the steps for each task.
        - Think through the information your task needs (where does it come from?).
        - These will become your functions.
    - Plan out the user interface.
        - You can start with the barest interface, but you should have an idea what you want the final product to look like.
- Managing your variables

  - What’s global, local, and passed through as parameters (hint - this program can use all three)?
- Assigning functions as tasks
  - Does your program sort out the different tasks into their own functions?
- Sequencing
  - Does your program sequence from the user interaction into the necessary functions?
  - Is there an efficiency to your code that flows from the design document?
- An intuitive user experience
  - Is your display appropriate to the program (what’s viewable and what scrolling has to be done)?
  - Is your display adaptable to other resolutions?
  - Is the interface intuitive?