wordGuessDesign

Overview:

I would create a game that is similar to hangman. It would need to do the following things:

- Get input from the user (Player 1) into a string and then perform the following validation:
  - Only one word (no spaces)
  - No numbers (we will tell the user that words with numbers are off limits)
  - No special characters
- Clear the screen and display:
  - Guess counter (remaining guesses)
  - Eligible letters \n
  - Underscores to represent the number of blank lines (separated by spaces for clarity).
- Allow user to guess a letter and then
  - if right
    - Take from eligible letters
    - Fill in any letters in correct order with blank lines in between and display to user
    - Prompt for next letter
  - if wrong
    - decrement guess counter
    - Take from eligible letters
    - Tell user they got it wrong
    - Prompt for next letter.
- The game ends when the guess counter reaches 0 or the user figures out the correct word

Implementation details:

A/B - Store secret word, can only contain letters

I would use a do-while loop that did the following:
- Prompts user for word with cin and stores it into a string variable
- checks each character with isalpha() and returns error message if isalpha() returns false.
  - Re-prompt in this case
- Clear the screen with newlines (or an OS call)
- Created a char array for letter guesses with a-z

C - Repeatedly ask for single letters

I would have a do-while loop (conditional upon wordGuess == false && numGuesses > 0) that did the following:
- Prompts the user for a char
- checks the char against the guess list. If char has been guessed or is not alpha, use continue keyword to restart loop.
- checks the char against the word. If char exists then update the string that is shown to the user (with underscores) and don't decrement guess counter. Also, remove letter from guess

array. If char doesn't exist, then remove from guess array, decrement guess counter and tell user to try again with the valid letters that they can use.

D - Show correct guesses and word

- This can be done with one string. I would probably just use a cstring and code like this:

```
char cstring[stringToGuess.length()] = '\0'
for(int i = 0; i < stringToGuess.length(); i++) { cstring[i] = '_' }
…
for (int i = 0; i < stringToGuess.length(); i++)
{
        if(guess == stringToGuess.at(i)) { cstring[i] = guess }
}
```

As long as the cstring is initialized only once, the word displayed to the user will be continually updated.

E - Only so many guesses

This one is easy. Just use an int counter and use counter- - to ensure that it counts down. When it reaches 0, the do-while loop terminates.

F - Misc. requirements

- Show the secret word:
  - cout << secretword
- list letters that are missing still
  - Just cout the c-string that i created above. That shows BOTH what they got right and what was still missing.
- Play again:
  - Just put a do-while loop around the whole program in main and prompt for a y/n