Projektarbete v.44 (Projektgrupp 2 Linköping) - Hänga gubbe spel

En beskrivning av arbetet under projektet med fokus på HUR ni arbetat agilt.

En reflektion över ert arbetssätt.

- Hur fungerade samarbetet i teamet?
- Vad tar ni med er för lärdomar till nästa projekt?
- Vad skulle ni vilja gjort annorlunda?

Beskrivning av resultatet.

- Hur långt kom ni med färdigställandet av produkten?
- Vad finns kvar att göra?
- Vidareutveckling av produkten (om ni fick fortsätta jobba vidare på den)

Reflektion över resultatet på produkten.

- o Finns det något körbart program?
- o Hur blev koden? Bra, mindre bra, strukturerad, rörig, ...?

Länk till GitHub-repo:

- https://github.com/Filipanderssondev/Project-v44.Hangman
- Trello tavla: https://trello.com/b/QohcSGzZ/projektgrupp-2-v44

Twist ider:

- 2 spelare
- man får ej gissa på vokaler
- När man har alla konsonanter i ordet måste man gissa på hela ordet

Format:

1. Krav
    - User stories is marked in blue
        - Tasks are marked in red

SRS

Overview:

- This game is a two person-only co-op multiplayer game of hangman with a twist. The twist is that you can't guess vowels, only consonants and when all the consonants are guessed (before 12 wrong guesses have been made) you can only guess on the entire word
- Definition of a round: A round is when an incorrect letter or word is guessed. This means that depending on word length a players may have more that 12 guesses if they guess correctly
- This game consists of two levels:
    - Level 1: Guessing of consonants, 12 combined errors = game over
    - Level 2: Guessing of the word, 12 combined errors = game over
    - (Same of 12 wrong guesses limit, not a new batch of 12 guesses)

1. User Interface:

- The game must be controlled entirely through console input/output, with clear prompts and instructions for the players.

2. Game Setup:

- The game must start by allowing two players to enter their names.
- The game consists of 2 levels, in level 1 the players may only guess a consonant until there are no consonants left.
- In level 2 the game should inform the players when all the consonants are “visible” and aks the to guess a whole words from then onwards
- The game ends if someone guesses the correct word.

3. Guessing:

- In level 1: Each player must guess one consonant
    - As a player i want to enter a consonant, so i can have a chance of guessing the right word
        - **Task 3.1.1:** Prompt the player to input a consonant
        - **Task 3.1.2** Implement input functionality, allow the player to input a character
        - **Task 3.1.3:** Store the input value in a variable
        - **Task 3.1.4:** Output the variable with the stored input value
- In level 2: Each player must guess a word
    - As a player i want to enter a word, so i can have a chance of guessing the right word
        - **Task 3.2.1:** Prompt the player to enter a word as a guess.
        - **Task 3.2.2:** Implement input functionality, allow the player to input a word.

        - **Task 3.2.3:** Store the word in a variable
        - **Task 3.2.4:** Output the variable with the stored input value

4. Guessing Display:

- The game should display the correct and wrong guess (both letters and words)
    - Level 1: As a player i want see a display of which consonants that have already been guessed
        - **Task 4.1.1:** Call on the function that stored previous guesses/letters
        - **Task 4.1.2:** Display the “correct letters” instead of the blank spaces
        - **Task 4.1.3:** Display the “incorrect letters” in a line under the hidde word / blank spaces
    - Level 2: As a player i want see a display of which words have been guessed
        - **Task 4.2.1:** Call on the function that stores the previous guesses words
        - **Task 4.2.2:** Display the “correct words” instead of the blank spaces
        - **Task 4.2.3:** Display the “incorrect words” in a line under the hidden word / blank spaces
- The game should show the progress with a visual hangman
    - As a player i want a visual representation of a hangman to see the game progress
        - **Task 4.3.1**: Implement a ASCII (- ,|) characters to visualize the process of hangman.
        - **Task 4.3.2:** Use a condition that checks if the guess is incorrect if true the hangman should receive an additional body part

5. Validations

- In level 1: The game must prevent a player from guessing a **consonant** that has already been guessed.
    - Level 1: As a player i want to know when I enter an already used **consonant**, so i do not waste my guess
        - **Task 5.1.1**: Implement a checker to verify that character hasn't been used.
        - **Task 5.1.2**: Output message to inform player of error
        - **Task 5.1.3:** Reprompt the player for another character
- In level 2: The game must prevent a player from guessing a **word** that has already been guessed.
    - Level 2: As a player i want to know when i enter an already used **word**, so i do not waste my guess
        - **Task 5.2.1**: Implement a checker to verify that a word hasn't been used
        - **Task 5.2.2**: Output message to inform player of error
        - **Task 5.2.3.:** Reprompt the player for another word
- In level 2: The game must prevent player from guessing to long/too short words
    - Level 2: As a player, I want to know when i have guessed a word that's not the size of the hidden word, so i do not waste my guess
        - **Task 5.3.1**: Compare the word length with the hidden word for character match

        - **Task 5.3.2**: Output message to inform player of error
        - **Task 5.3.3.:** Report the player for another word
- In level 1: The game should print an error message if the player enters anything but consonants
    - Level 1: As a player I want the game to print an error message if I enter an invalid character, so I do play the game correctly
        - **Task 5.4**.1 Implement a function that validates if the input is a consonant. (The game should accept only lowercase)
        - **Task 5.4.2**: Output message to inform player of error
        - **Task 5.4.3.:** Reprompt the player for a new letter
- In level 2: The game should print an error message if player enters a word with anything but letters
    - Level 2: As a player I want the game to print an error message if I enter a word that includes an invalid character, so I do play the game correctly
        - **Task 5.5.1**: Implement a function that validates if the input only consists of letters. (The game should accept only lowercase)
        - **Task 5.5.2**: Output message to inform player of error
        - **Task 5.5.3.:** Reprompt the player for a new word

6. Game Flow:

- The game must manage the turn order and inform players when it is their turn.
    - As a player i want to know when it is my turn, so i know when it's my turn to act
        - **Task 6.1.1**: Develop a mechanism to alternate turns between Player 1 and Player 2. Output players name when it is their turn
- The game should notify the players when switching from level 1 to level 2
    - As a player i want to know when the switch from level 1 to level 2 happens, so I know what I should guess (letter or word)
        - **Task 6.2.1**: output the change from “level 1” to “level 2” “All the consonants are guessed, you should guess a word cosisting if x letters now”
        - **Task 6.2.2**: The input that is accepted should only be words as per wors tasks in validations
- The game should allow a player to quit at any time, terminating the game early if necessary.
    - As a player I want to be able to quit the game at anytime, if I get bored.
        - **Task 6.3.1**: Implement a quit command
- After 12 rounds, the game must display the correct word if neither player guessed it.

7. Multiplayer:

- The game must have 2 players, with each player taking their turn in sequence.
    - As a player i want to play with another person, this will make it more interesting

        - **Task 7.1.1**: Implement Two player mode. Set up player registration. Create a function to register two players.
        - **Task 7.1.2**: Allowing each to enter a unique name. Output the name of each player
- Players can choose their own names.
    - As a player I want to be able to enter my name, so that the game experience is more personalized
        - **Task 7.2.1**: Implement a customization for names instead of defaults

8. Game Rules:

- In the beginning of the game the rules of hangman should be displayed with the two levels.
    - As a player i want to be presented with the rules of the game, so i know how to play
        - **Task 8.1.1:** Design and write the rules. Clear and concise explanation of the rules, games objective and special conditions
        - **Task 8.1.2:** Display the rules to the player before the game begins
        - **Task 8.1.3:** Prom player to press enter in order to continue to the game

//Implement rules display function. Code a function that displays the game rules to players at the start of game or on request

9. Game End:

- The game must detect when the word has been correctly guessed and end the game
    - As a player, I want to know when the correct answer has been given, so I know that I have won.
        - **Task 9.1.1**: Compare the input and the hidden word
        - **Task 9.1.2**: If the guess is correct go to **Task 9.2.1**
- The game must announce the winner based on who guessed the right word.
    - As a player, I want the game to display “[playersname] Won!”, so that the winner is clearly identified
        - **Task 9.2.1:** Implement an output like “[playersname] Won!”, display the winners name if one player guesses the word correctly.
        - **Task 9.2.2**: Ask the players if they want to play again
- If the man is hanged and nobody has guessed the correct word, the game should display “Game over”and show the correct word
    - As a player I want it displayed when the game is over and show me correct word, so I know what the hidden word is and that the game has ended
        - **Task 9.3.1**. Implement a condition that cheks if wrog guesses is 12, then proceed with below
        - **Task 9.3.2**: Implement a output “Game over”
        - **Task 9.3.3** If maximum wrong guesses has been made fulfilled to show hidden word and display the full visualization